

Activation Function

13기 권민지 김재겸 이승우 임채현

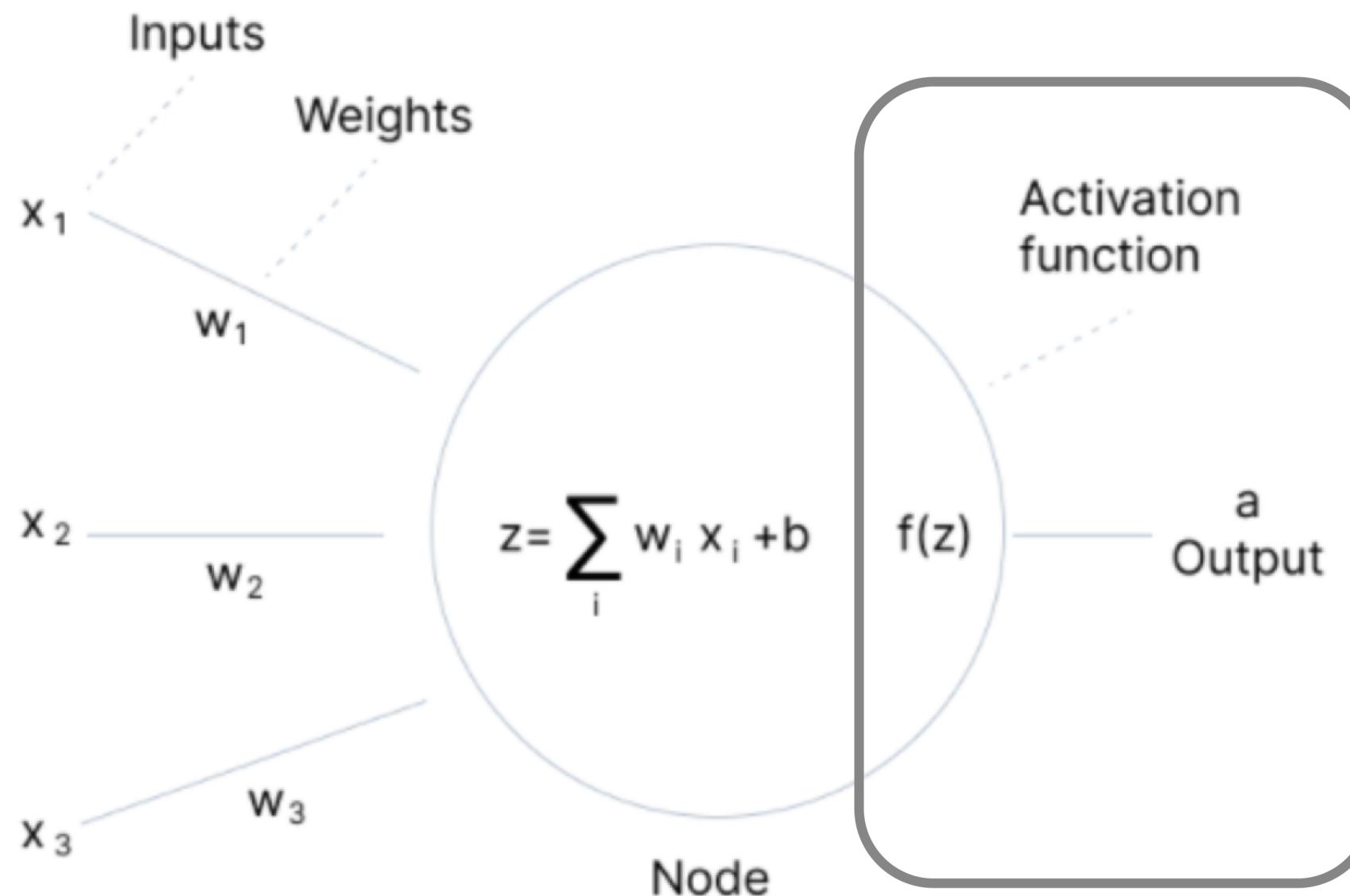
CONTENTS

- 01 ————— **What is AF**
- 02 ————— **Activation Function**
- 03 ————— **Sigmoid / Logistic**
- 04 ————— **Softmax**
- 05 ————— **Tanh**
- 06 ————— **ReLU**
- 07 ————— **leaky ReLU**
- 08 ————— **ELU**
- 09 ————— **SWISS**
- 10 ————— **GELU**
- 11 ————— **How to Choose**

01

What is AF?

가중치 합이 적용된 Input을 다음 레이어로
어떤 Output 값으로 보낼지 결정하는 Function



Activation Function

Binary Step Function

- 임계치를 기준으로 넘으면 출력
- 다중 값 표현 불가 -> 다중 클래스에서 사용 불가
- 기울기가 0 이므로 역전파과정 사용 불가

Non-linear Activation Function

- 입력과 출력 간의 복잡한 관계 표현 가능
- 이미지, 영상, 음성 등 고차원 데이터에 유리
- 선형 데이터를 비선형 데이터로 바꾸는 과정 必

Linear Activation Function

- 선형 함수 : 다중값 출력 가능
- 미분값 상수 : 역전파과정 사용 불가
- 은닉층 무시, 얻을 수 있는 값 제한됨

비선형함수를 사용하는 이유

1. 입력과 관련 있는 미분 값을 얻으므로 역전파 가능
2. 입력 뉴런의 어떤 가중치가 더 나은 예측을 제공할 수 있는지 이해 가능
3. 출력이 여러 레이어를 통과하는 입력의 비선형 조합이 되므로 뉴런의 여러 레이어를 쌓을 수 있음

Activation Function

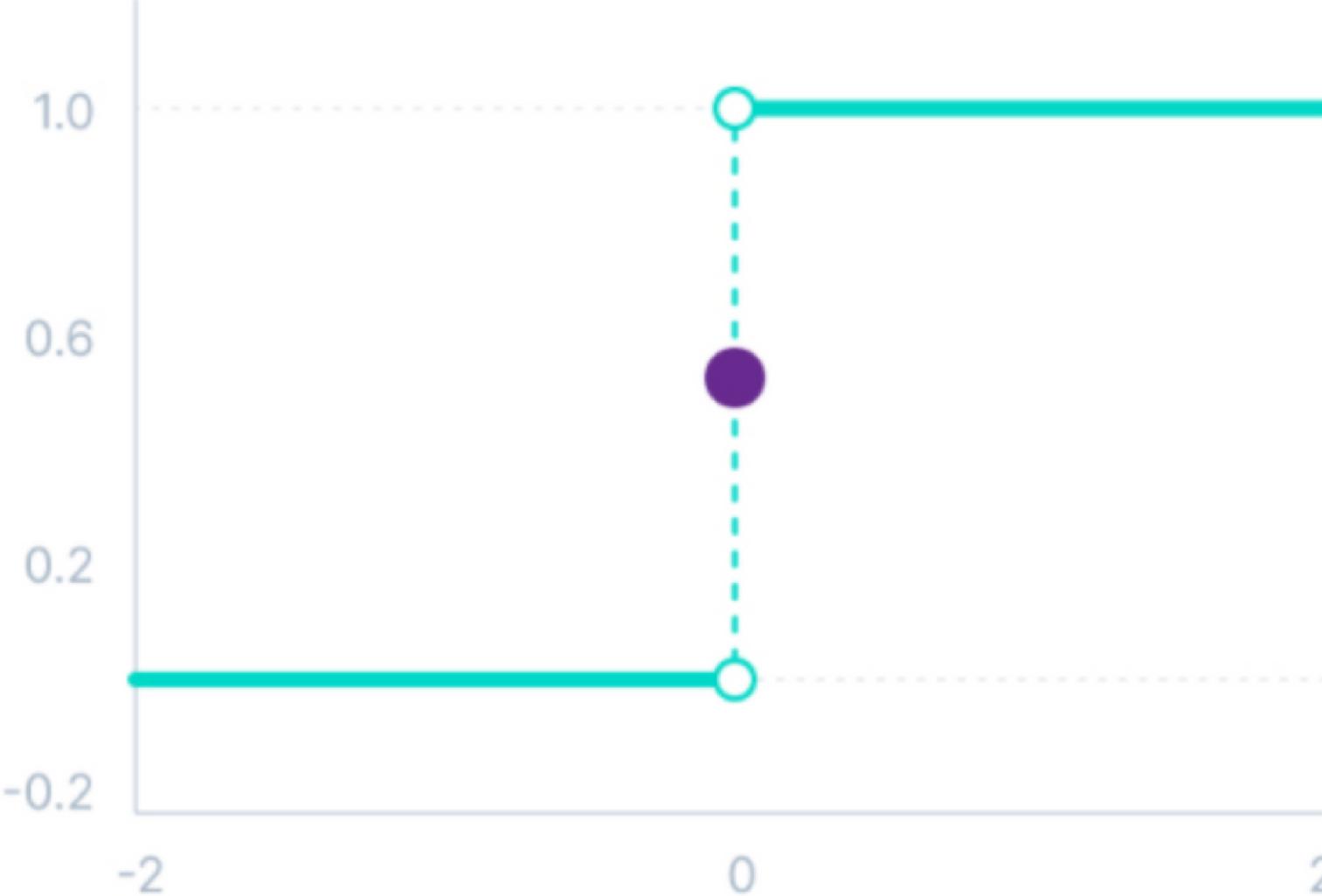
Binary Step Function

- 임계치를 기준으로 넘기기
- 다중 값 표현 불가 -> 다른 활성화 함수 필요
- 기울기가 0 이므로 역전파 과정 불가능

Linear Activation Function

- 선형 함수 : 다중값 출력 가능
- 미분값 상수 : 역전파 과정 가능
- 은닉층 무시, 얻을 수 있는 정보 전부 통과하는 입력의 비선형 조합이 베이어를 쌓을 수 있음

Binary Step Function



$$f(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x, \quad (x < 0) \end{cases}$$

Activation Function

- 단순한 관계 표현 가능
- 고차원 데이터에 유리
- 데이터로 바꾸는 과정 必需

| 유

• 분 값을 얻으므로 역전파 가능

• 중치가 더 나은 예측을 제공할 수 있음

• 통과하는 입력의 비선형 조합이 베이어를 쌓을 수 있음

Activation Function

Binary Step Function

- 임계치를 기준으로 넘으면 출력
- 다중 값 표현 불가 -> 다중 클래스에서 사용 불가
- 기울기가 0 이므로 역전파과정 사용 불가

Non-linear Activation Function

- 입력과 출력 간의 복잡한 관계 표현 가능
- 이미지, 영상, 음성 등 고차원 데이터에 유리
- 선형 데이터를 비선형 데이터로 바꾸는 과정 必

Linear Activation Function

- 선형 함수 : 다중값 출력 가능
- 미분값 상수 : 역전파과정 사용 불가
- 은닉층 무시, 얻을 수 있는 값 제한됨

비선형함수를 사용하는 이유

1. 입력과 관련 있는 미분 값을 얻으므로 역전파 가능
2. 입력 뉴런의 어떤 가중치가 더 나은 예측을 제공할 수 있는지 이해 가능
3. 출력이 여러 레이어를 통과하는 입력의 비선형 조합이 되므로 뉴런의 여러 레이어를 쌓을 수 있음

Activation Function

Binary Step Function

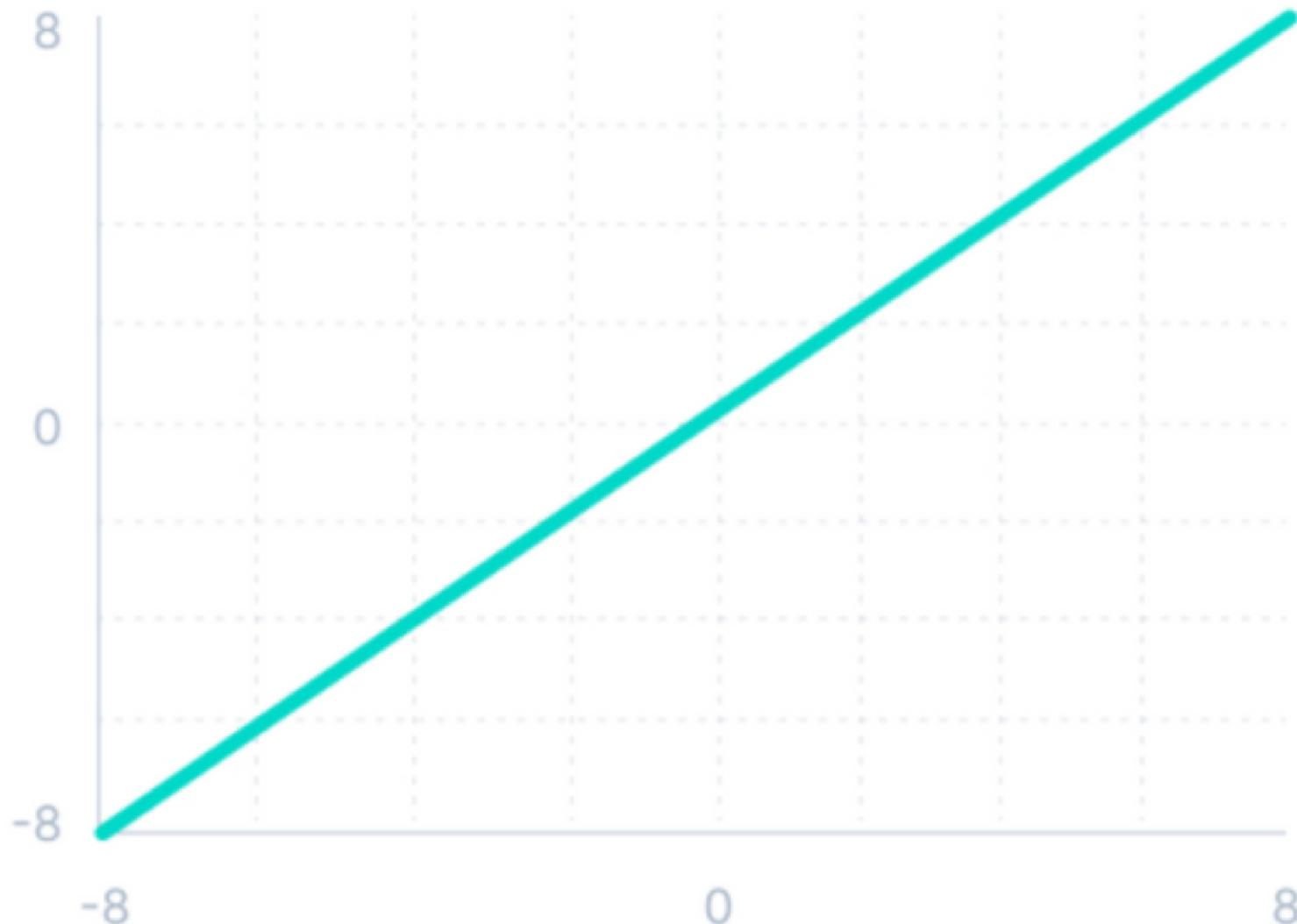
- 임계치를 기준으로 넘으면 1을 출력
- 다중 값 표현 불가 -> 다중 출력 불가
- 기울기가 0 이므로 역전파 과정에서 미분 불가

Linear Activation

- 선형 함수 : 다중값 출력 가능
- 미분값 상수 : 역전파과정에서 미분 가능
- 은닉층 무시, 얻을 수 있는 값은 다

$$f(x) = cx, \quad c \text{ is constant}$$

Linear Activation Function



Activation Function

• 다양한 관계 표현 가능
• 미분 가능
• 다중 출력 가능
• 고차원 데이터에 유리
• 비선형 데이터로 바꾸는 과정 必

이유

• 미분 값을 얻으므로 역전파 가능
• 미분 가능한 경우 더 나은 예측을 제공할 수

• 미분 가능한 경우 더 나은 예측을 제공할 수
• 미분 가능한 경우 더 나은 예측을 제공할 수
• 미분 가능한 경우 더 나은 예측을 제공할 수
• 미분 가능한 경우 더 나은 예측을 제공할 수

Activation Function

Binary Step Function

- 임계치를 기준으로 넘으면 출력
- 다중 값 표현 불가 -> 다중 클래스에서 사용 불가
- 기울기가 0 이므로 역전파과정 사용 불가

Non-linear Activation Function

- 입력과 출력 간의 복잡한 관계 표현 가능
- 이미지, 영상, 음성 등 고차원 데이터에 유리
- 선형 데이터를 비선형 데이터로 바꾸는 과정 必

Linear Activation Function

- 선형 함수 : 다중값 출력 가능
- 미분값 상수 : 역전파과정 사용 불가
- 은닉층 무시, 얻을 수 있는 값 제한됨

비선형함수를 사용하는 이유

1. 입력과 관련 있는 미분 값을 얻으므로 역전파 가능
2. 입력 뉴런의 어떤 가중치가 더 나은 예측을 제공할 수 있는지 이해 가능
3. 출력이 여러 레이어를 통과하는 입력의 비선형 조합이 되므로 뉴런의 여러 레이어를 쌓을 수 있음

Activ

Binary

- 임계치
- 다중 값
- 기울기

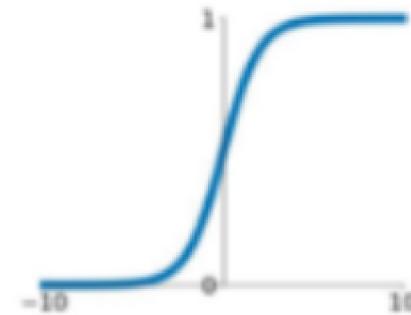
Linea

- 선형 함
- 미분값
- 은닉층

Activation Functions

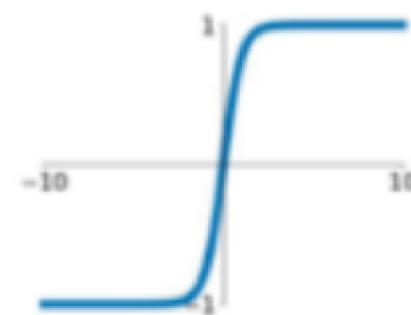
Sigmoid

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



tanh

$$\tanh(x)$$



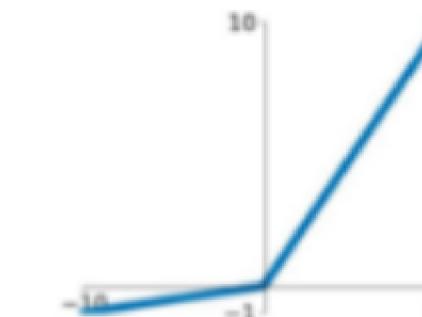
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

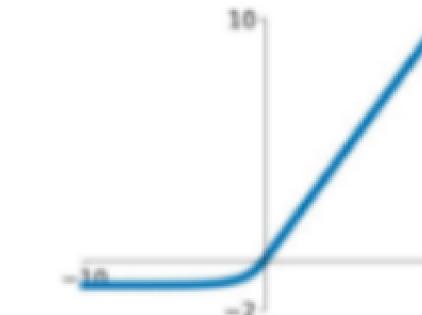


Maxout

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

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Different Activation Functions and their Graphs

03

Sigmoid / Logistic

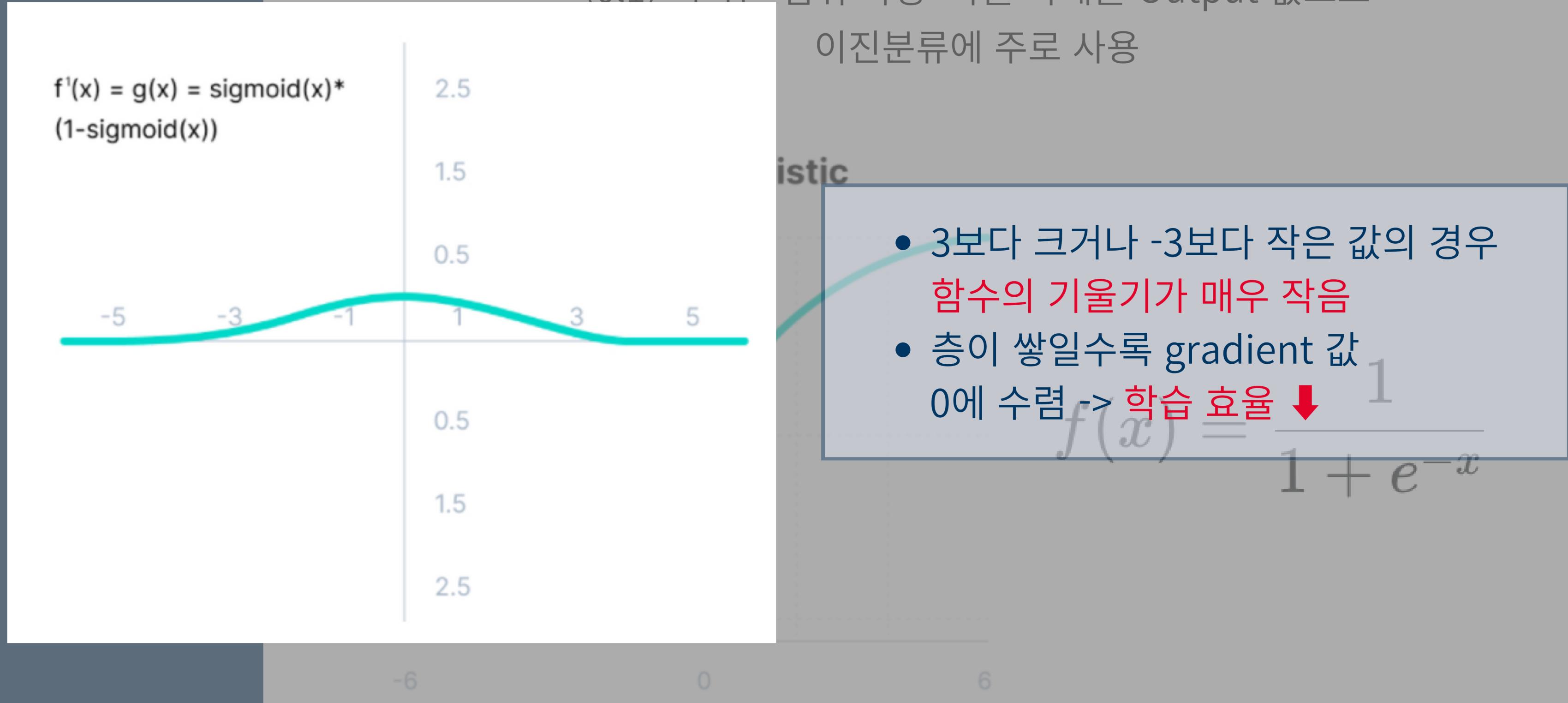
- (0,1) 이라는 범위 특성: 확률 자체를 Output 값으로
- 이진 분류에 주로 사용



03

Sigmoid / Logistic

- (0,1)이라는 범위 특성: 확률 자체를 Output 값으로
이진분류에 주로 사용



04

Softmax

- Logistic 함수의 **다차원 일반화**
- 활성화 함수 마지막 과정으로 활용
- 출력값을 확률로

$$\text{Softmax } \sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

$$\text{Sigmoid } S(x) = \frac{1}{1 + e^{-x}}$$

04

Softmax

- Logistic 함수의 **다차원 일반화**
- 딥러닝에선 주로 활성화 함수 마지막 과정으로 활용
- 출력값을 확률로

Quiz1

Softmax는 0000에 주로 사용된다.

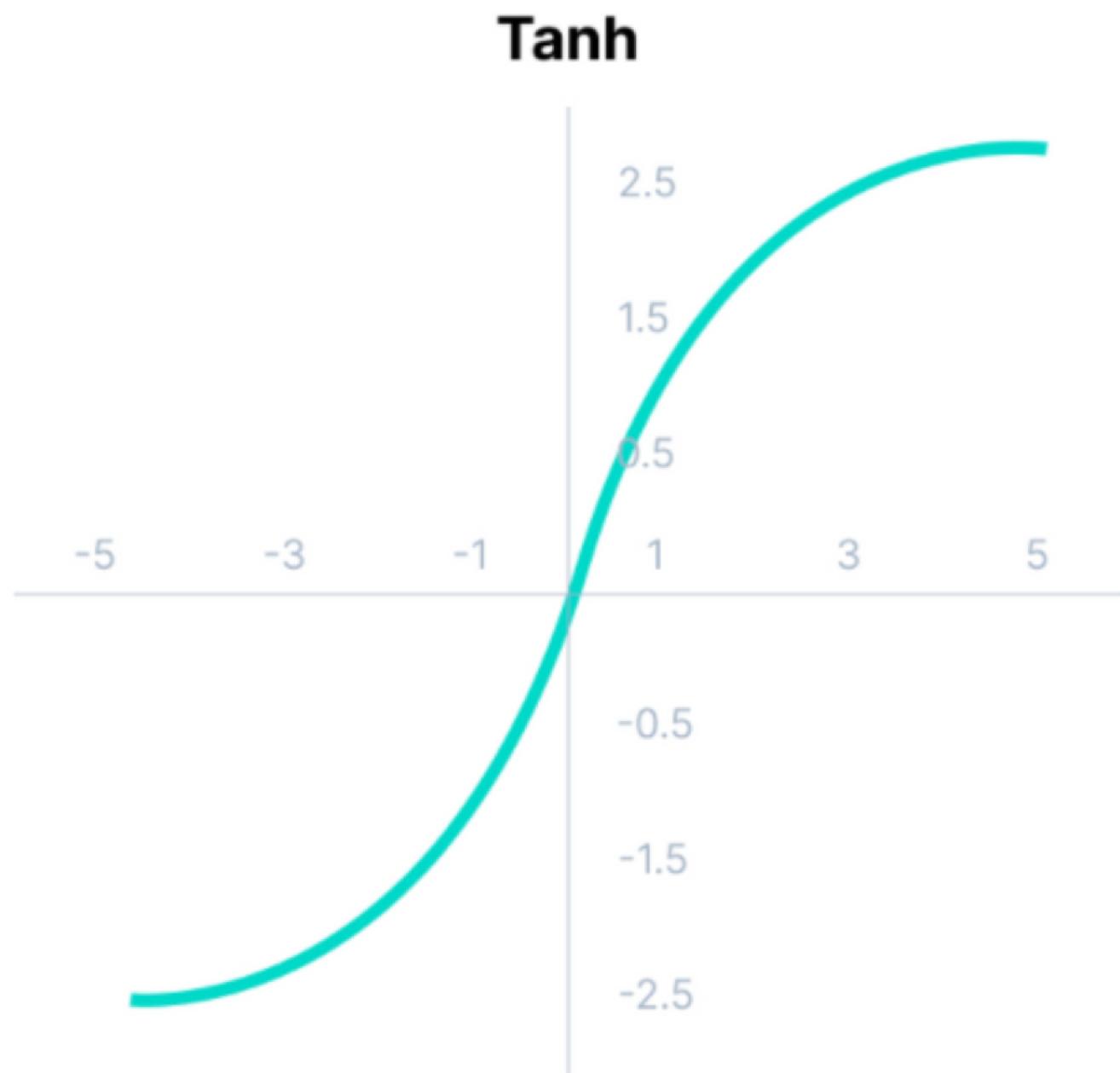
$$\frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

$$\text{Sigmoid } S(x) = \frac{1}{1 + e^{-x}}$$

05

Tanh

- (-1,1) 이라는 범위 특성: 0을 데이터 중심으로 두기에 용이

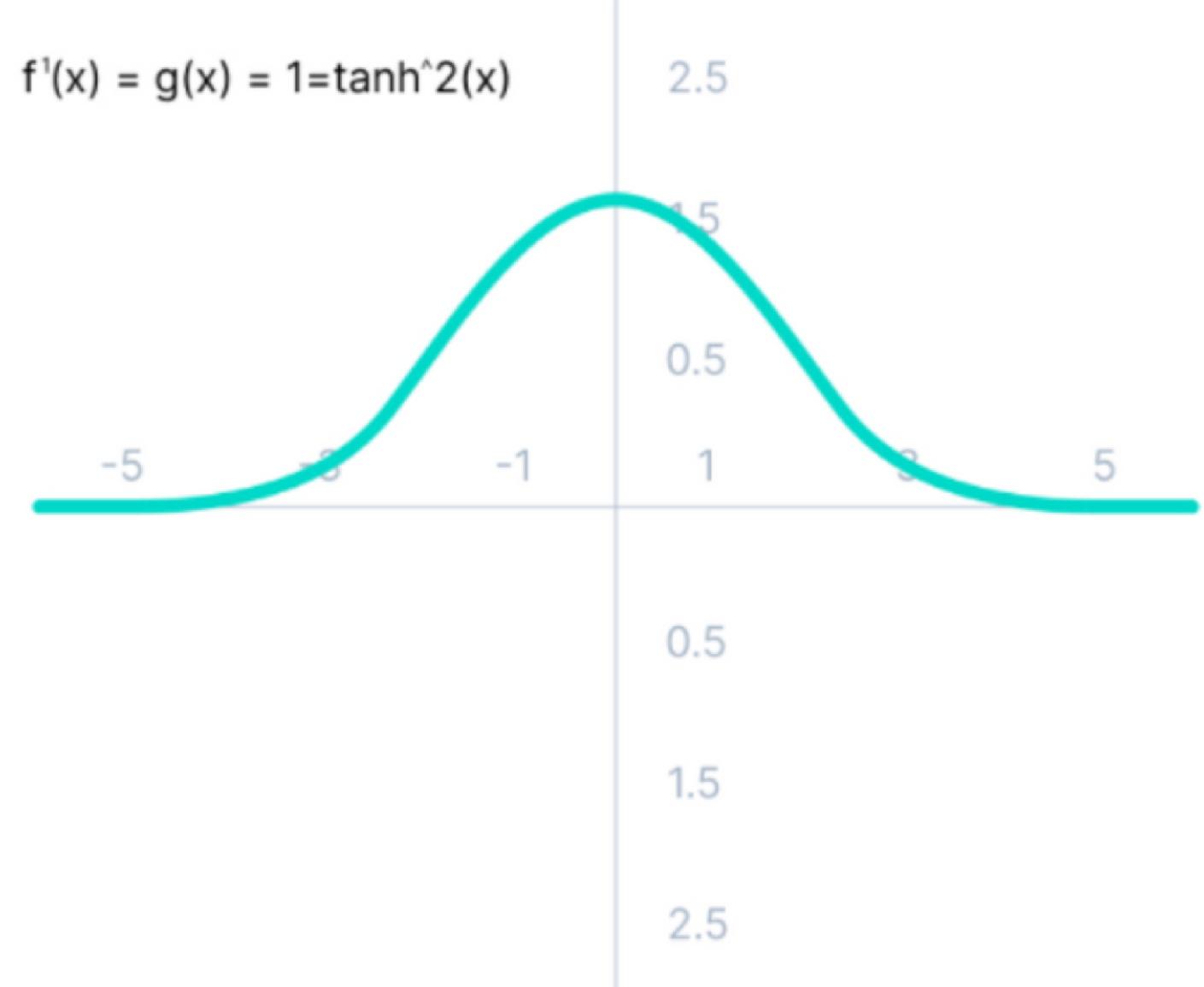


$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

05

Tanh

Tanh (derivative)



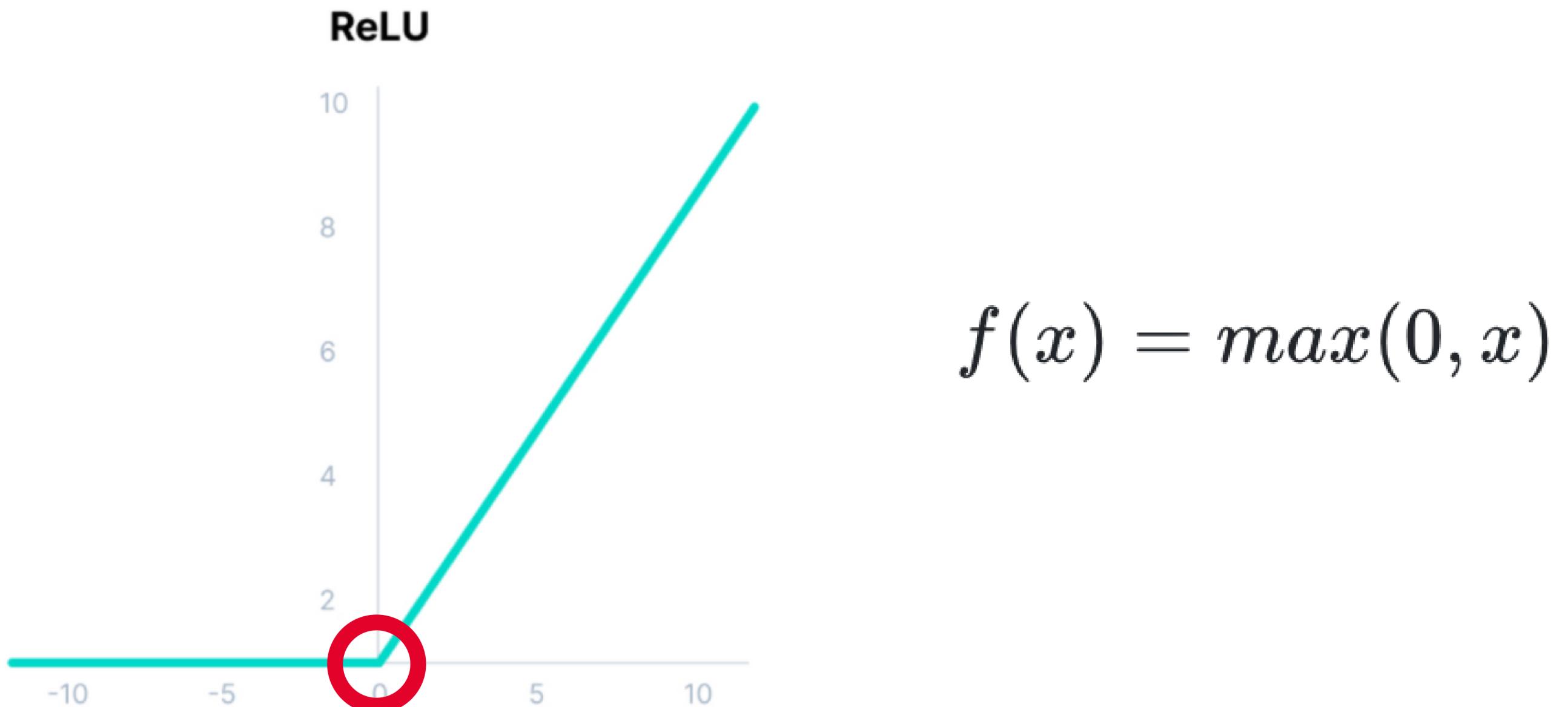
라는 범위 특성: 0을 데이터 중심으로 두기에 용이

- 5보다 크거나 -5보다 작은 값의 경우
함수의 기울기가 매우 작음
- 층이 쌓일수록 gradient 값
0에 수렴 \rightarrow 학습 효율 ↓
- Sigmoid 함수보다 가파른 기울기

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

ReLU

- Rectified Linear Unit
- 선형함수이지만, 미분함수 존재 => 역전파 가능, 효율적 학습 가능



06

ReLU

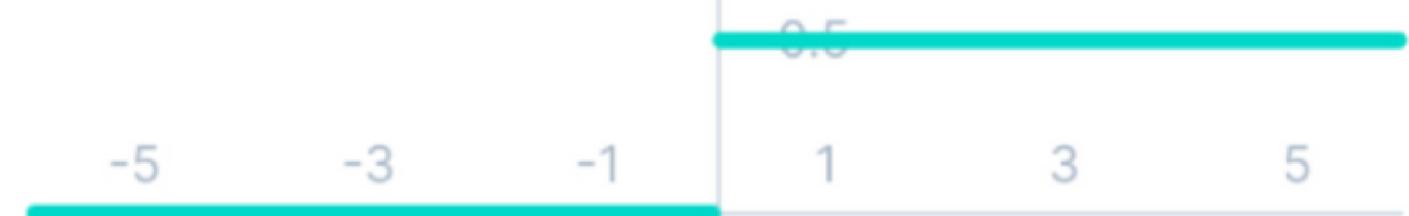
The Dying ReLU problem

$$f^1(x) = g(x) = 1, x \geq 0$$

2.5

Quiz2

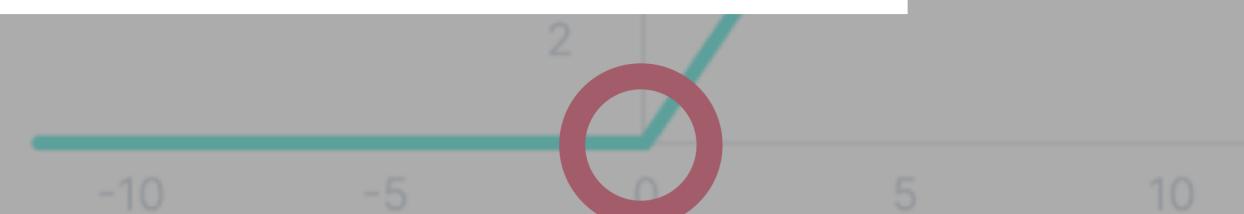
ReLU 함수 음수 입력 값 미분 함수



Rectified Linear Unit

미분함수 존재 => 역전파 가능, 효율적 학습 가능

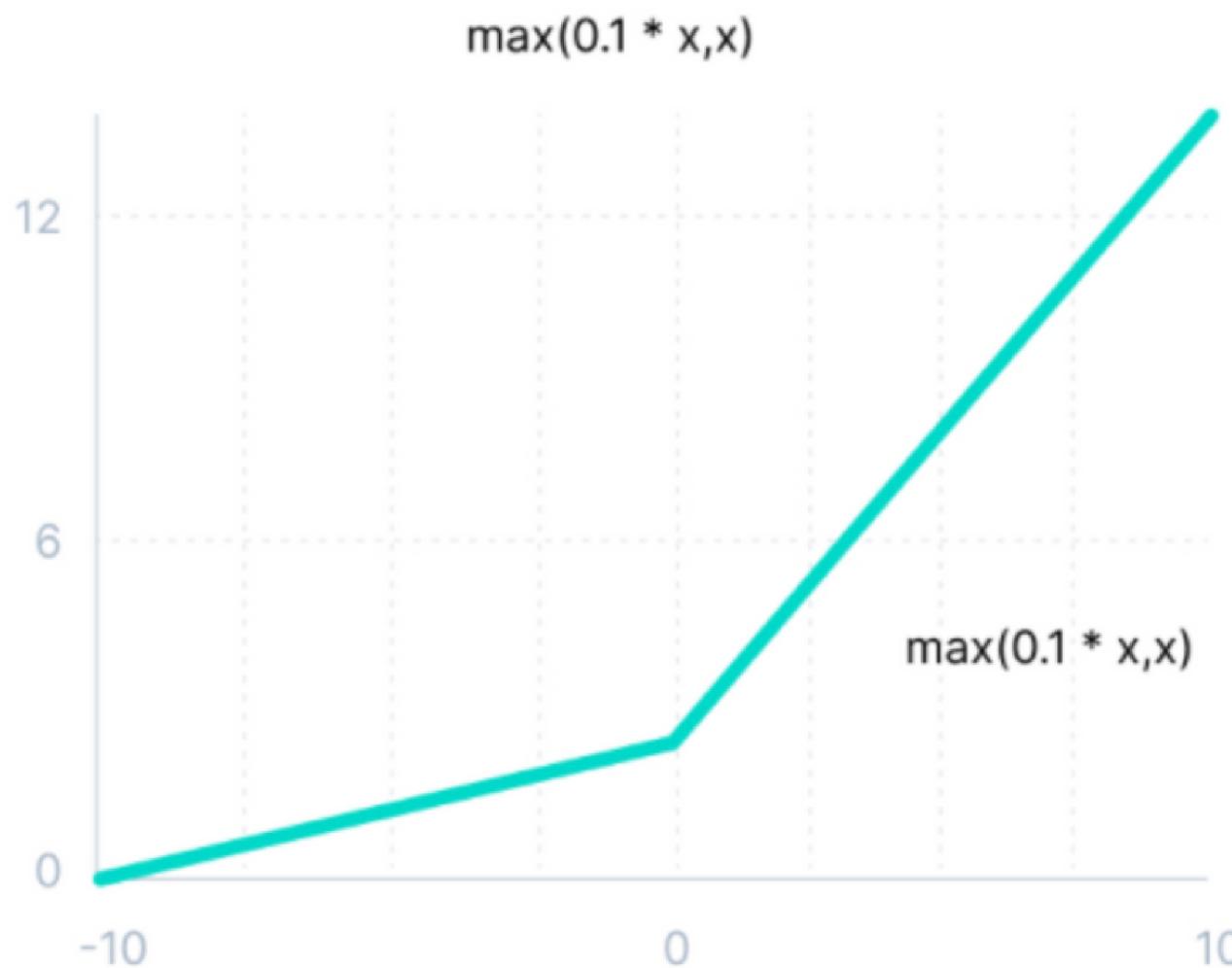
- 특정 뉴런만 활성화: 연산 효율 ↑
- Dying ReLU 문제:
모든 음수 입력 값은 즉시 0 =>
훈련하는 모델의 기능이 감소
- 데이터 셋의 축소가 장점이자 단점



leaky ReLU

- ReLU의 Dying ReLU 문제 해결
- ReLU보다 균형적인 값을 반환

Leaky ReLU



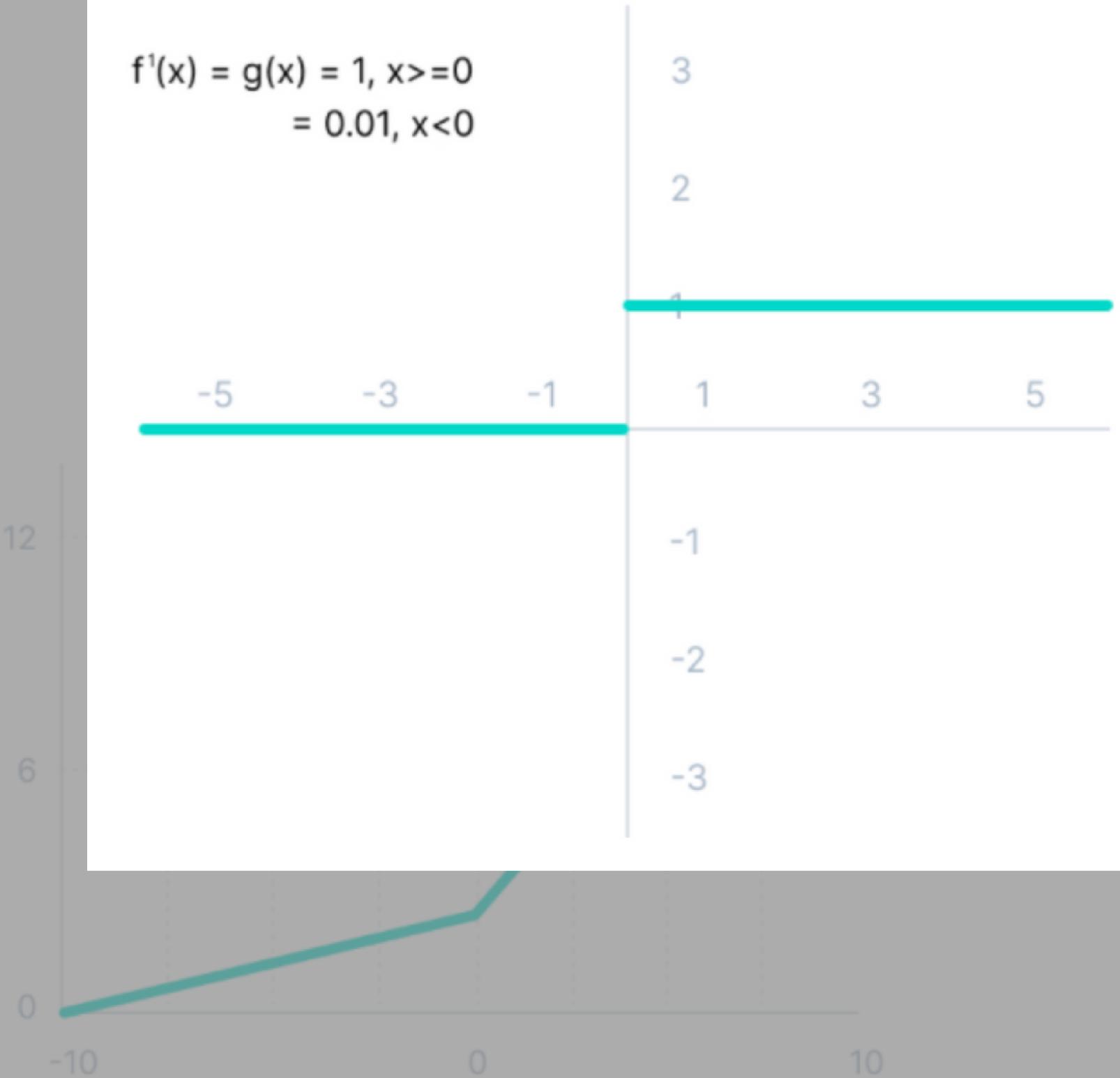
$$f(x) = \max(0.1x, x)$$

07

leaky ReLU

Leaky ReLu (derivative)

$$f'(x) = g(x) = 1, x \geq 0 \\ = 0.01, x < 0$$



문제 해결

값을 반환

$$) = \max(0.1x, x)$$

Recent Research

Recent Research On Activation Function

ELU

Exponential Linear Unit

FAST AND ACCURATE DEEP NETWORK LEARNING BY EXPONENTIAL LINEAR UNITS (ELUS)

Djork-Arné Clevert, Thomas Unterthiner & Sepp Hochreiter

Institute of Bioinformatics

Johannes Kepler University, Linz, Austria

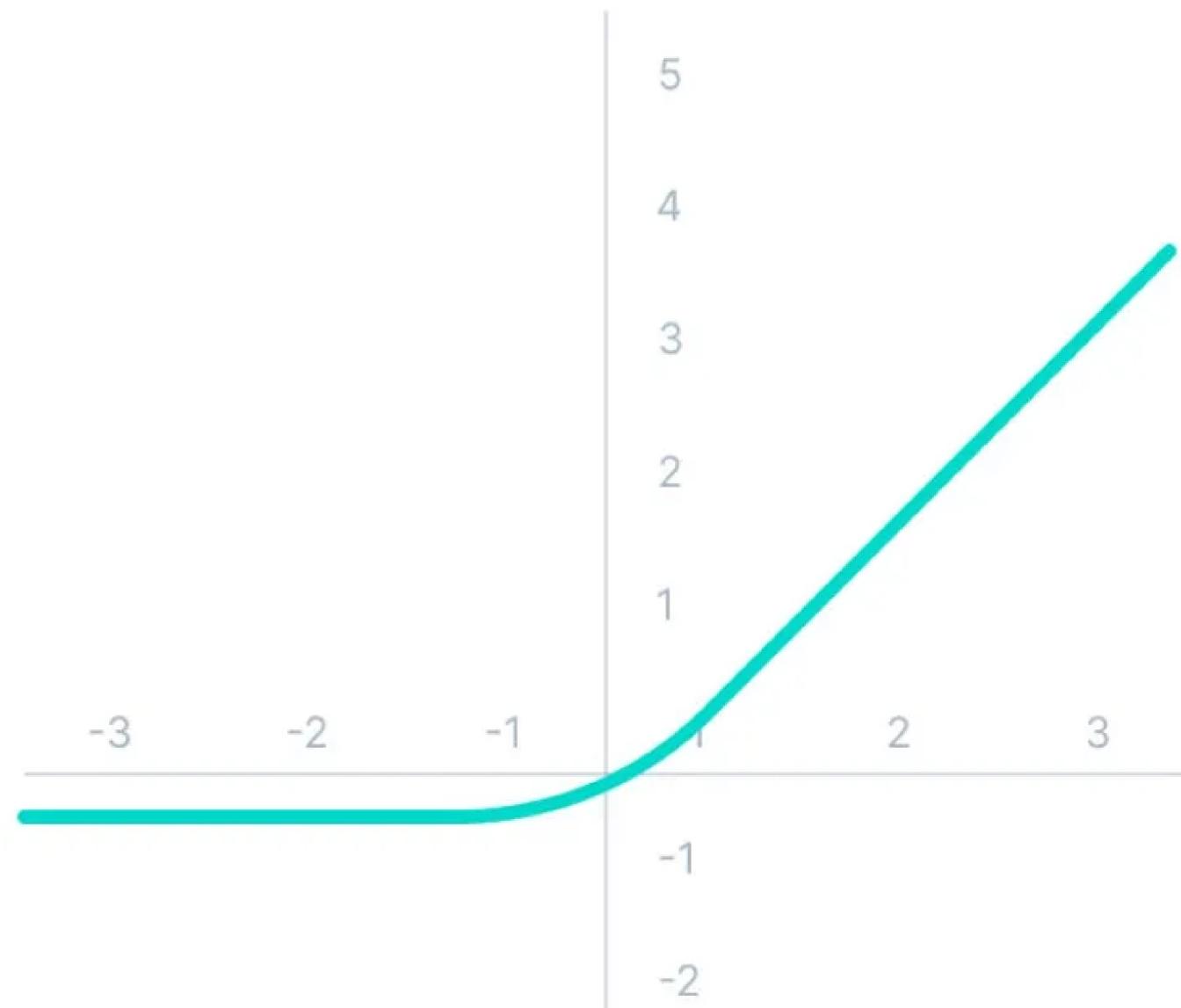
{okko, unterthiner, hochreit}@bioinf.jku.at

ICLR 2016

7027회 인용

ELU

Exponential Linear Unit



variant of ReLU

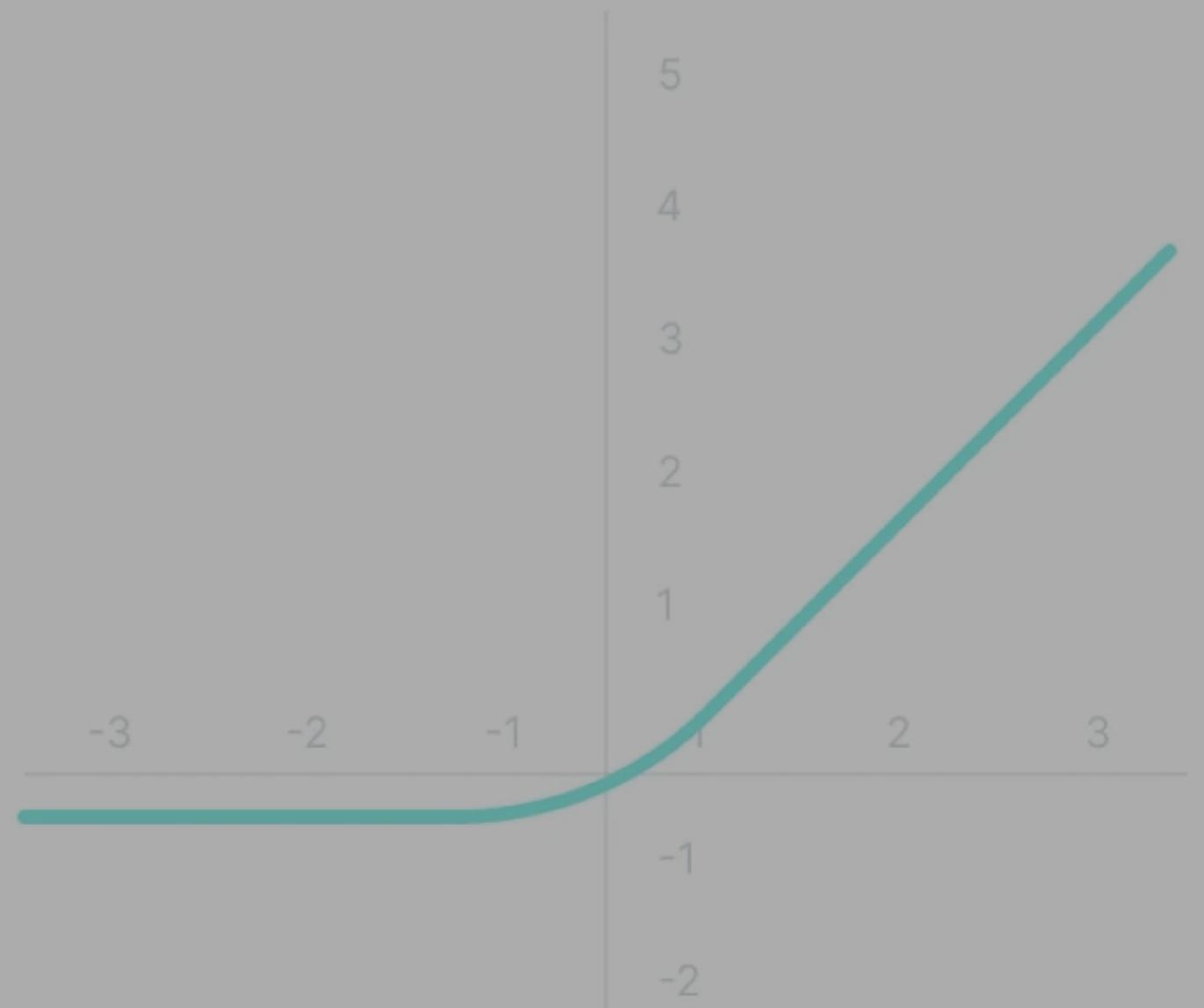
$$\begin{cases} x & \text{for } x \geq 0 \\ \alpha(e^x - 1) & \text{for } x < 0 \end{cases}$$

negative value:log curve

06

ELU

Exponential Linear Unit



variant of ReLU

Quiz3
-∞로 갈때 ELU는?

$$\begin{cases} x & \text{for } x \geq 0 \\ \alpha(e^x - 1) & \text{for } x < 0 \end{cases}$$

negative value: log curve

ELU

Exponential Linear Unit

Advantage vs ReLU

Smooth Transitions

- 음의 입력에 대해 부드러운 비선형성 제공 > 미분을 잘 정의

Negative Values

- dying ReLU 문제 해결

Zero-centered Output

- 출력의 평균이 0에 가까움 > 빠른 훈련 가능

ELU

Exponential Linear Unit

Advantage vs ReLU

Smooth Transitions

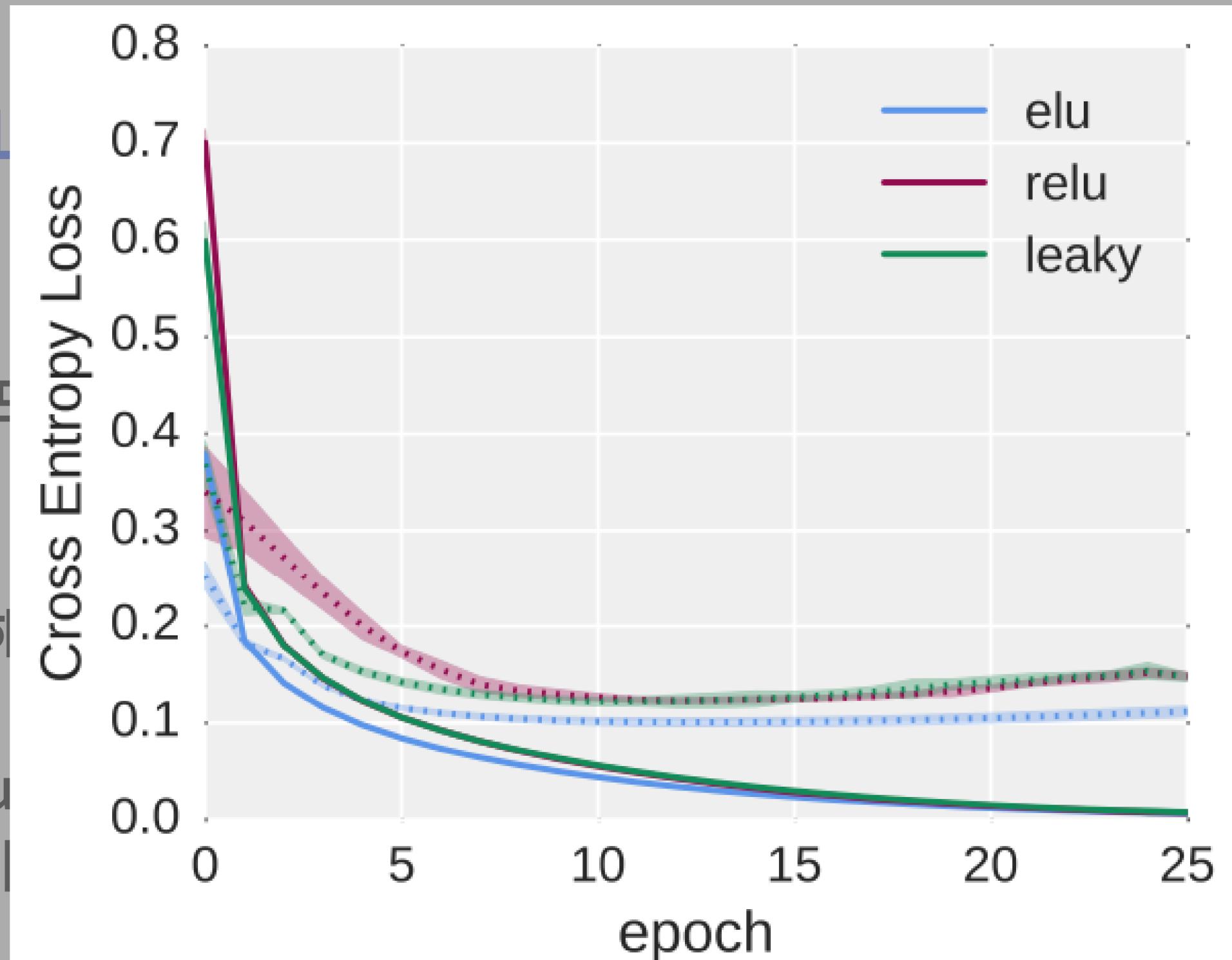
- 음의 입력에 대해 부드러운 전이

Negative Values

- dying ReLU 문제 해결

Zero-centered Output

- 출력의 평균이 0에 가깝다



ELU

Exponential Linear Unit

Limitation of the ELU

Computational Cost

- 음의 input에 지수 함수 계산 > 계산 비용 증가

Hyperparameter α

- α 를 결정하는데 별도의 학습이 필요함

Exploding gradient problem / Vanishing gradient problem

ELU

Exponential Linear Unit

Limitation of the ELU

Computational Cost

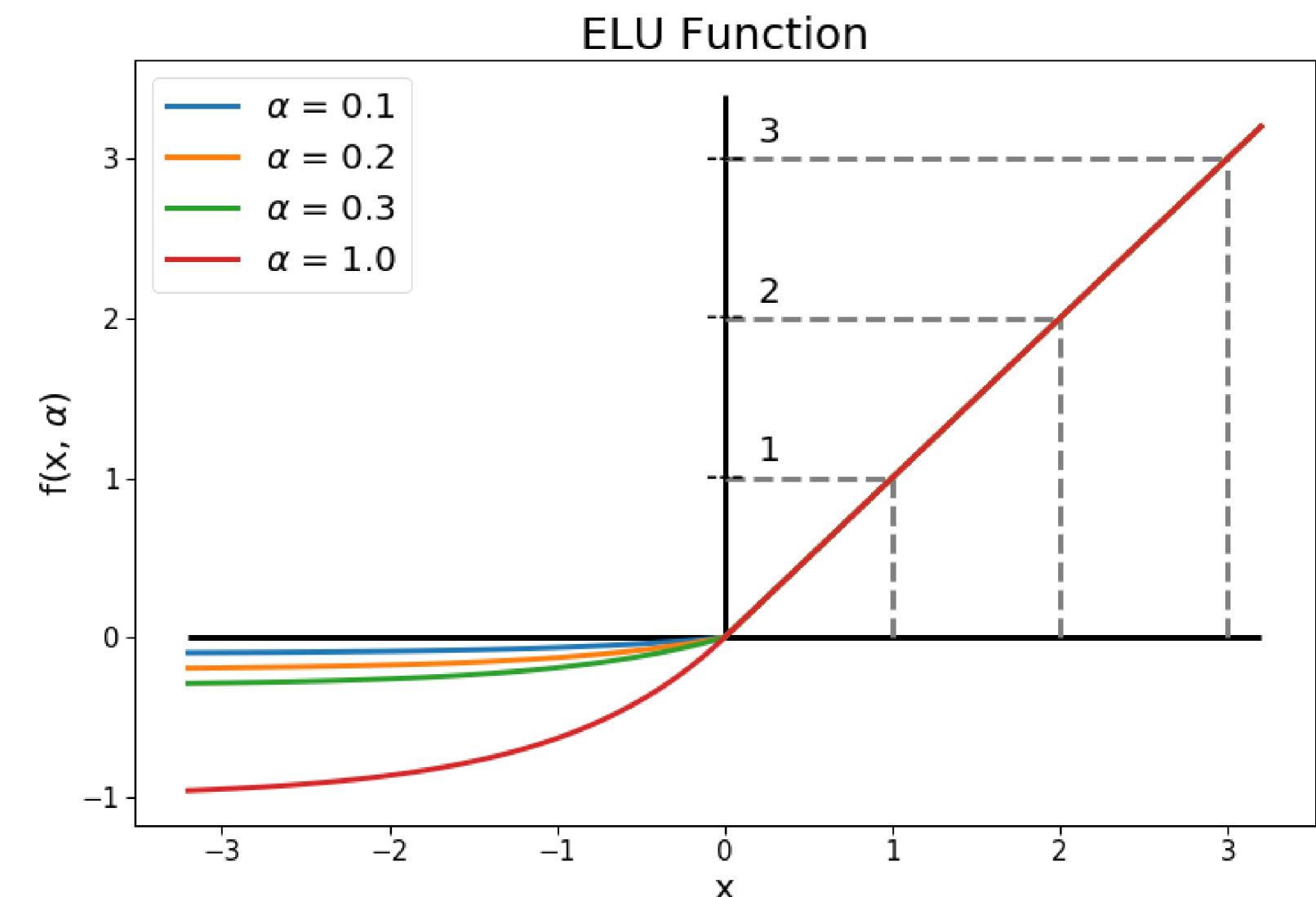
- 음의 입력에 지수 함수 계산

Hyperparameter α

- α 가 음의 input에 대처하는 능력

Exploding gradient problem

- 양의 input에 대해 속도가 빠름



ELU

Exponential Linear Unit

Limitation of the ELU

Computational Cost

- 음의 입력에 지수 함수 계산 > 계산 비용 증가

Hyperparameter α

- α 가 음의 input에 대한 output을 결정하는데 별도의 학습이 필요함

Exploding gradient problem / Vanishing gradient problem

Swish

Self-gated activation function

SWISH: A SELF-GATED ACTIVATION FUNCTION

Prajit Ramachandran*, Barret Zoph, Quoc V. Le

Google Brain

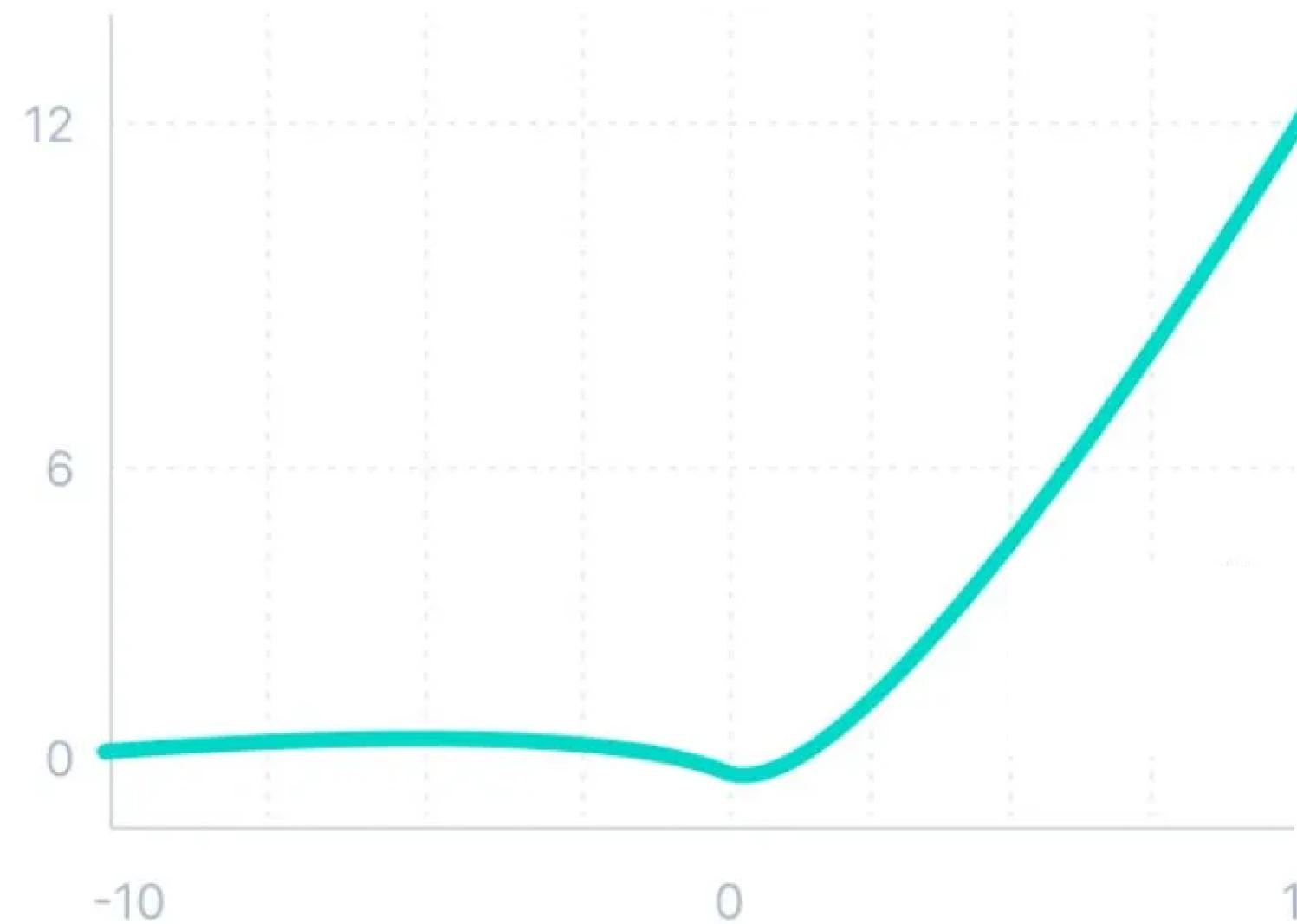
{prajit, barrettzoph, qvl}@google.com

ICLR 2018

3863회 인용

Swish

Self-gated activation function



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

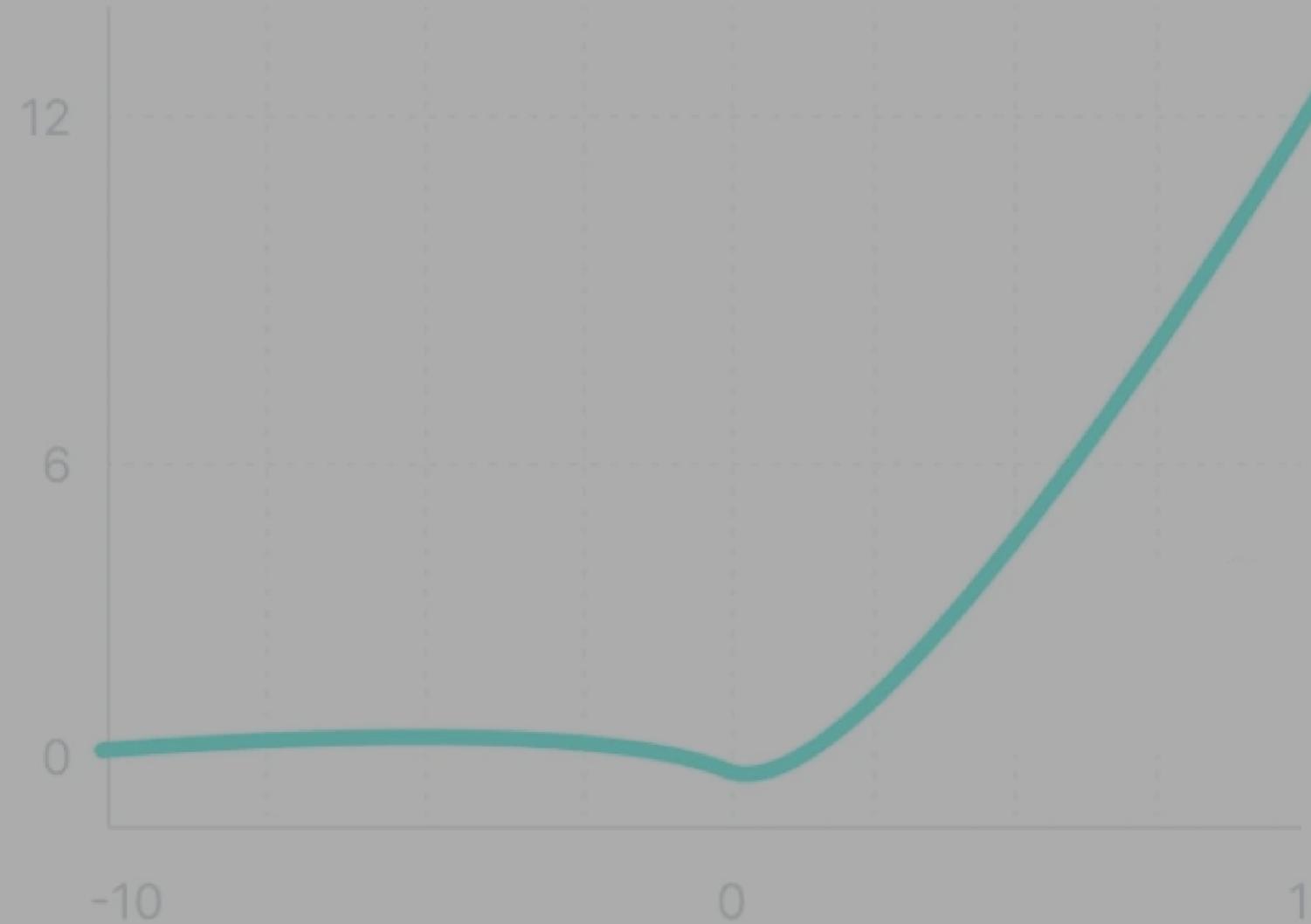
Inspired by use of sigmoid function for gating In LSTMs

Swish

Self-gated activation function

Quiz4

Swish 함수 작성

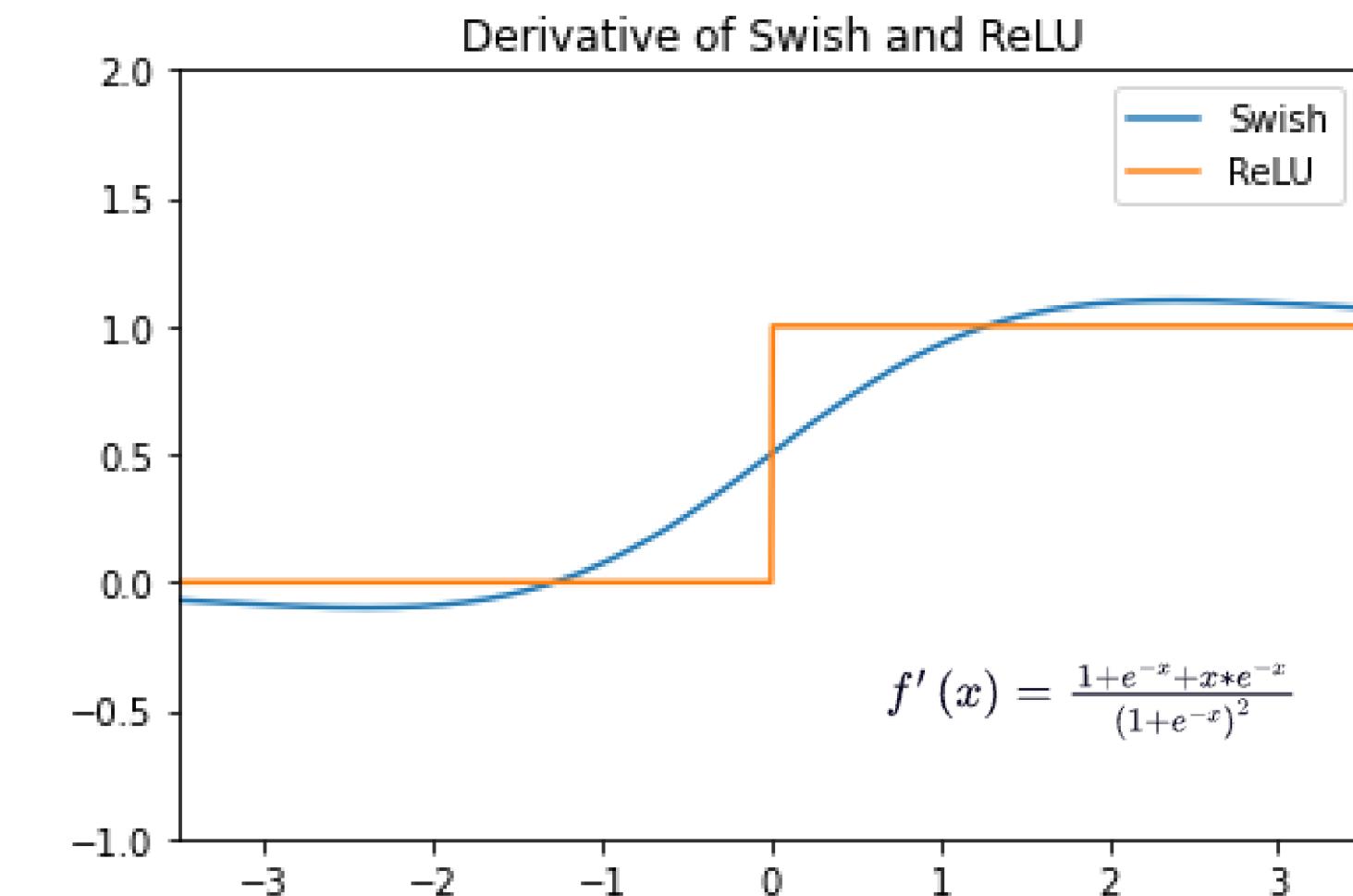
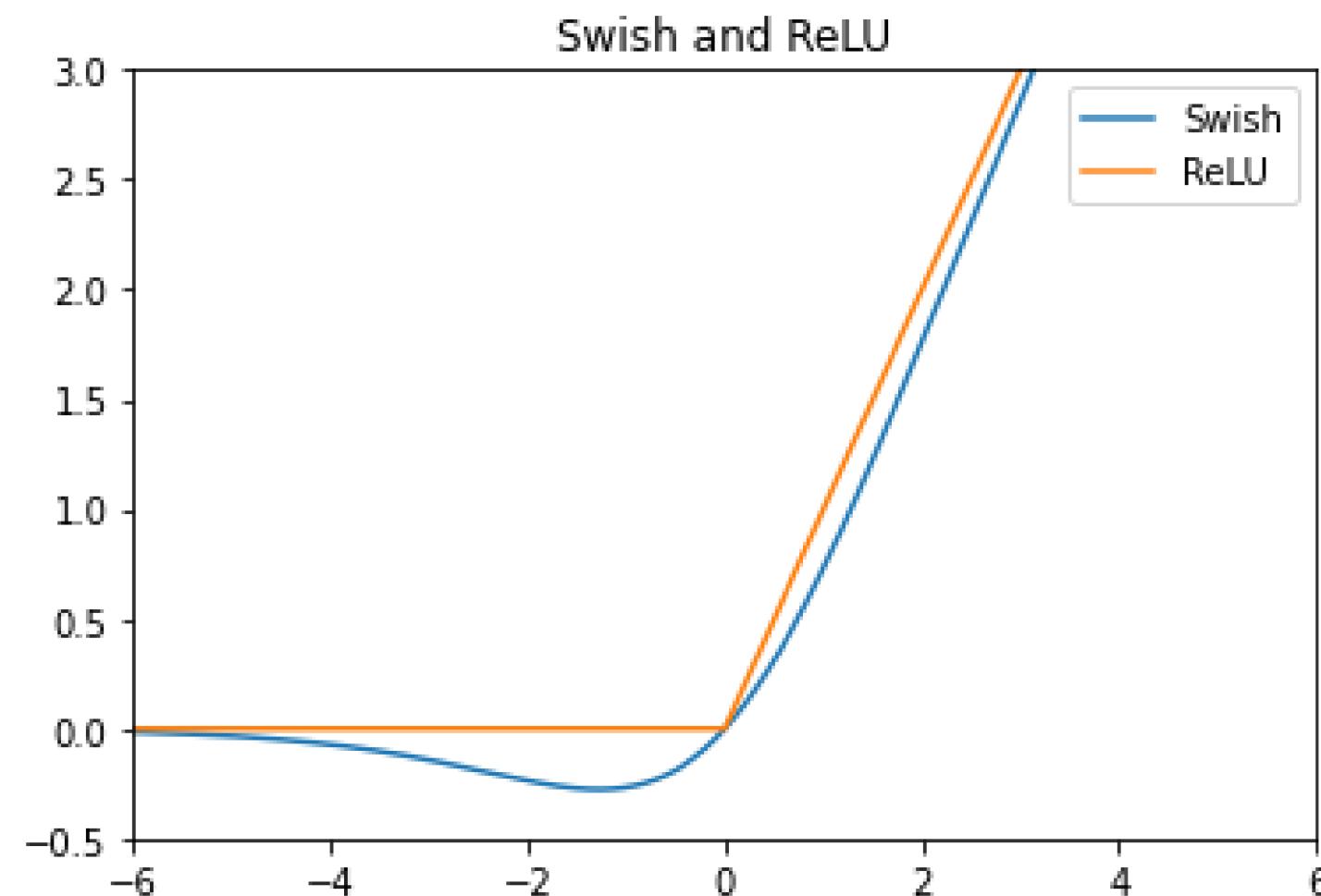


$$f(x) = x \cdot \text{sigmoid}(x)$$

Inspired by use of sigmoid function for gating In LSTMs

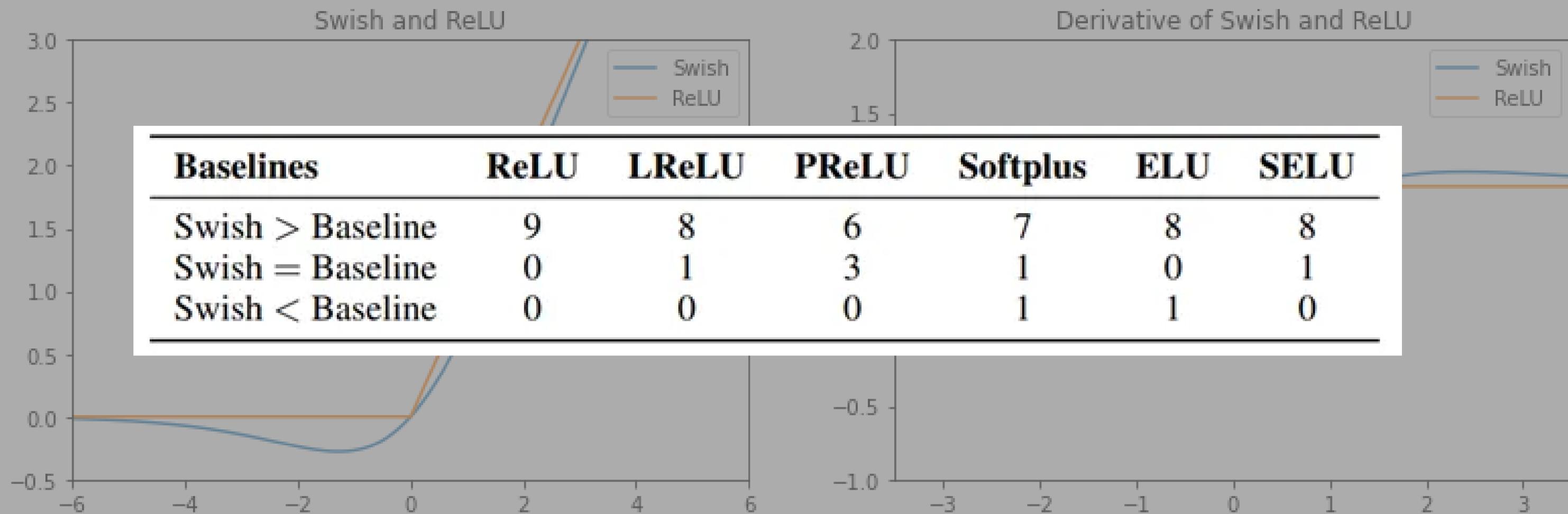
Swish

Self-gated activation function



Swish

Self-gated activation function



Swish

Self-gated activation function

Advantage of the Swish

smooth function

- ReLU/ELU 와 달리 0에서 부드럽게 변화 > 안정적 학습

Non-monotonicity

- 다양한 패턴 학습 가능

Large Negative Values to 0

- model sparsity 유지 > 계산 효율성 증가

Swish

Self-gated activation function

Limitation of the Swish

Computational Cost

- 지수 함수 계산 > 계산 비용 증가

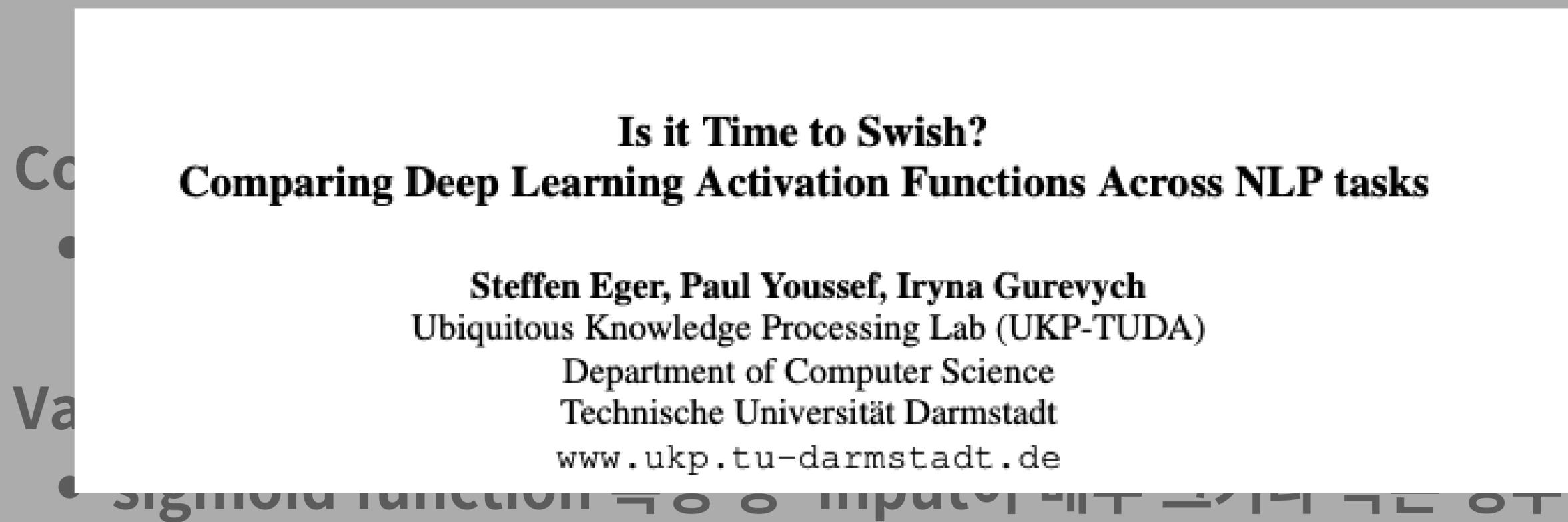
Vanishing gradient problem

- sigmoid function 특성 상 input이 매우 크거나 작은 경우

Swish

Self-gated activation function

Limitation of the Swish



Activation Function in LLMs

What is LLM?

- Large Language Model
- Transformer Architecture 기반
- GPT-3 1750억개, Switch Transformer 1조개 파라미터
- Multimodal Model

Role of Activation Function in LLMs

- 어떤 정보가 중요한지, 통과되어야 하는지, 무시해야 하는지
- 개별 단어뿐만 아니라 문맥, 뉘앙스, 전체 구조를 파악

Activation Function in LLMs

What is LLM?

- Large Language Model
- Transformer Architecture 기반
- G  ChatGPT
- Multimodal Model



Role of Activation Function in LLMs

- 어떤 정보가 중요한지, 통과되어야 하는지, 무시해야 하는지
- 개별 단어뿐만 아니라 문맥, 뉘앙스, 전체 구조를 파악

GELU

Gaussian Error Linear Unit

GAUSSIAN ERROR LINEAR UNITS (GELUS)

Dan Hendrycks*

University of California, Berkeley

hendrycks@berkeley.edu

Kevin Gimpel

Toyota Technological Institute at Chicago

kgimpel@ttic.edu

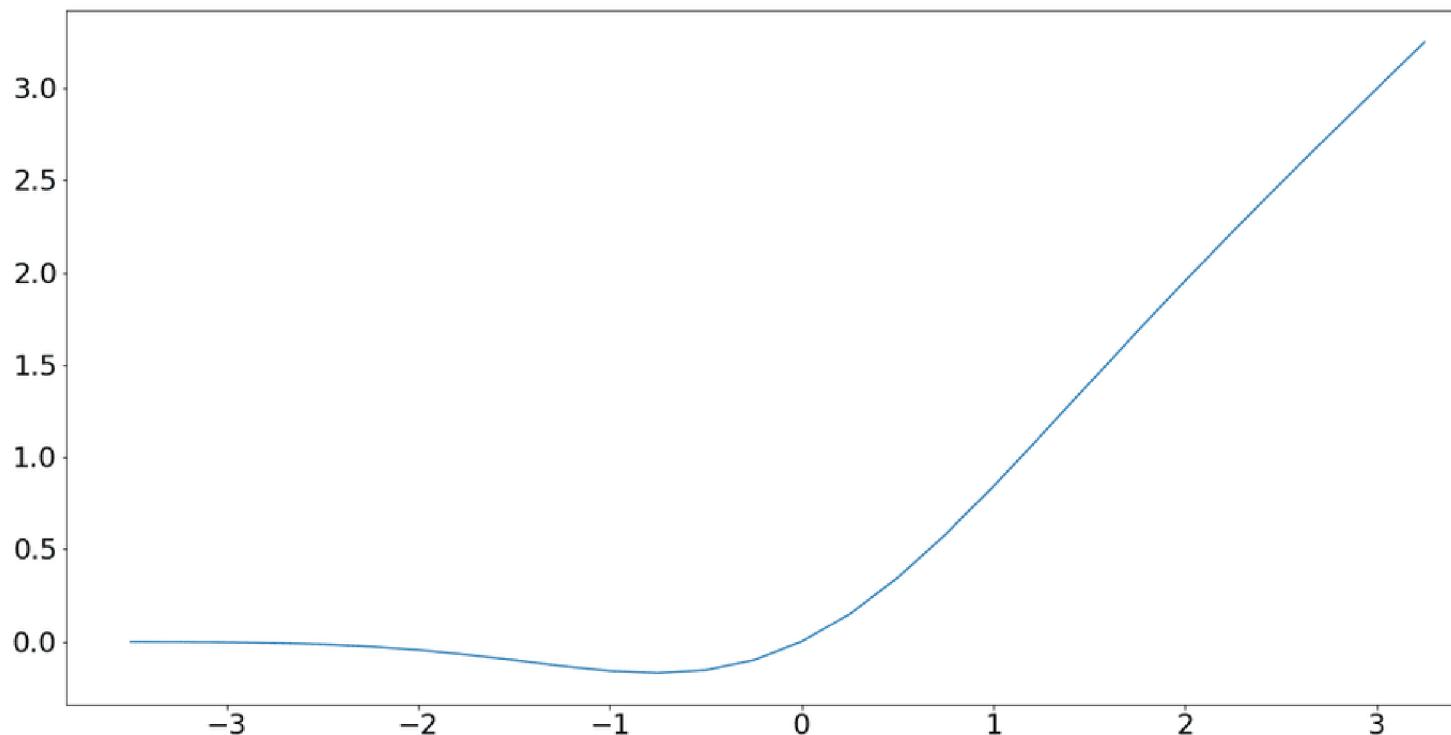
arXiv 2016

5521회 인용

Inspiration from ReLUs, dropout , and zoneout

GELU

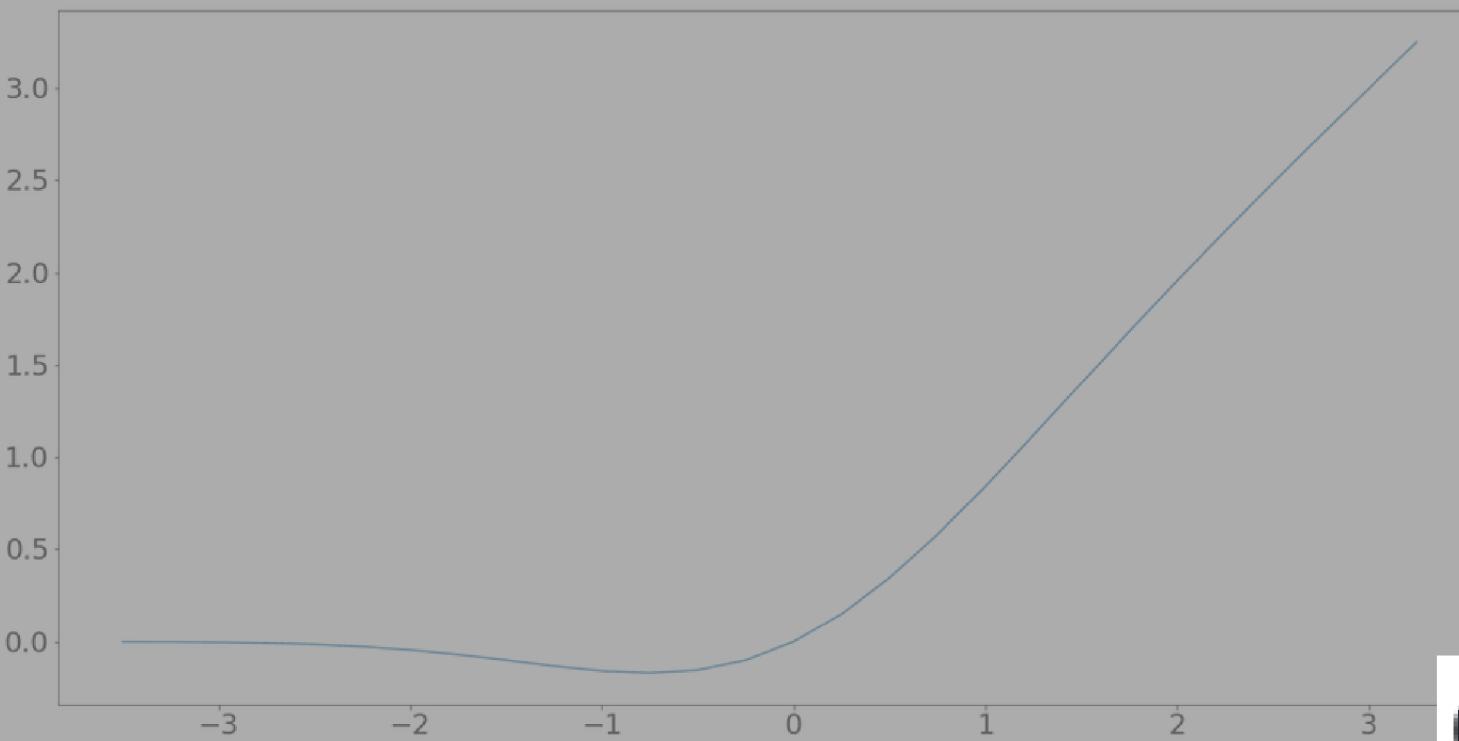
Gaussian Error Linear Unit



$$GELU(x) = x \times CDF(x) = x \times \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{x}{\sqrt{2}} \right) \right)$$

GELU

Gaussian Error Linear Unit



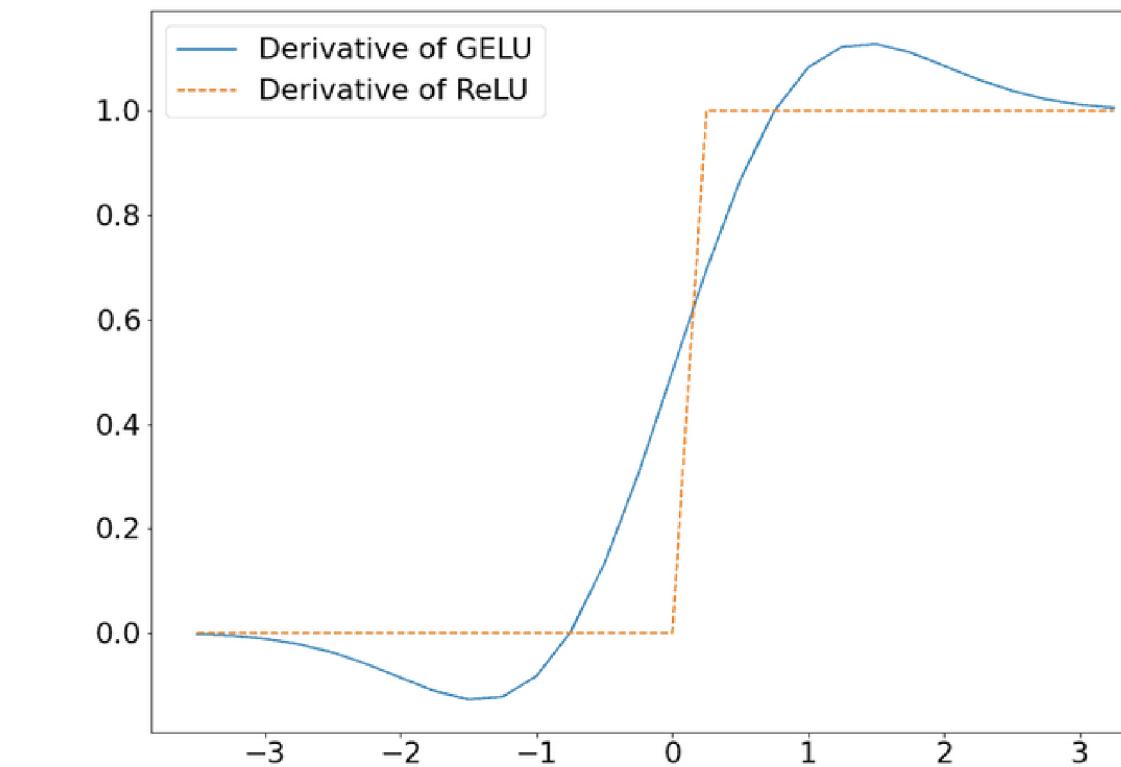
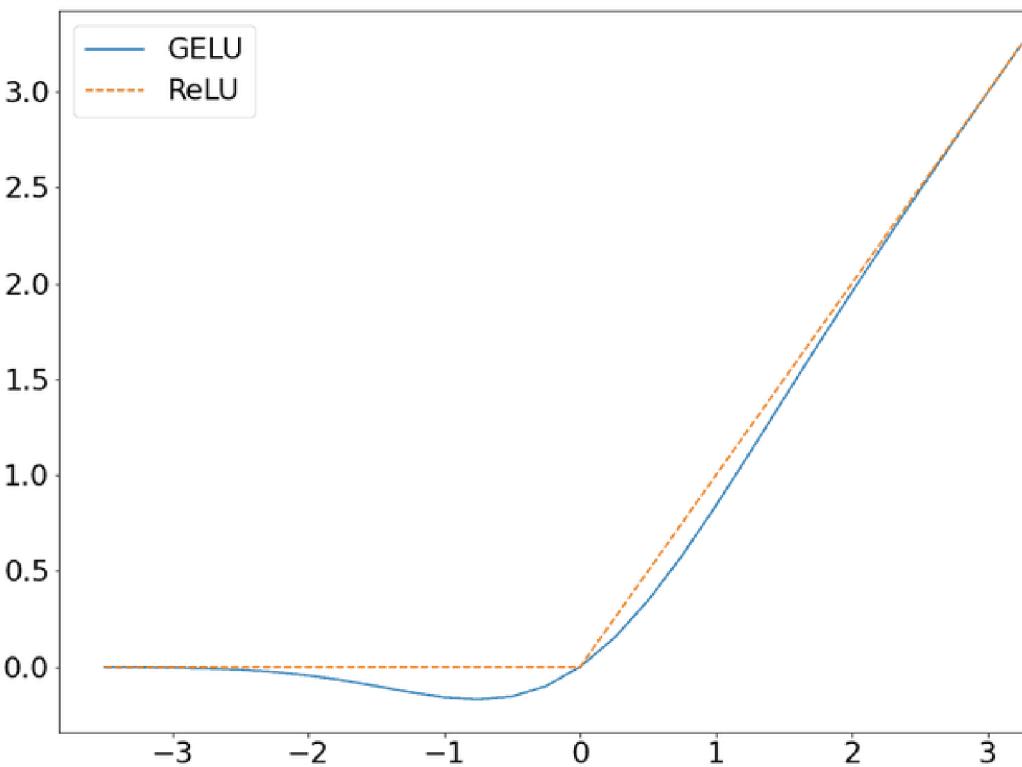
$$GELU(x) = x \times CDF(x) = x \times \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{x}{\sqrt{2}} \right) \right)$$

$$0.5x \left(1 + \tanh \left[\sqrt{2/\pi} (x + 0.044715x^3) \right] \right)$$

$$x\sigma(1.702x)$$

GELU

Gaussian Error Linear Unit

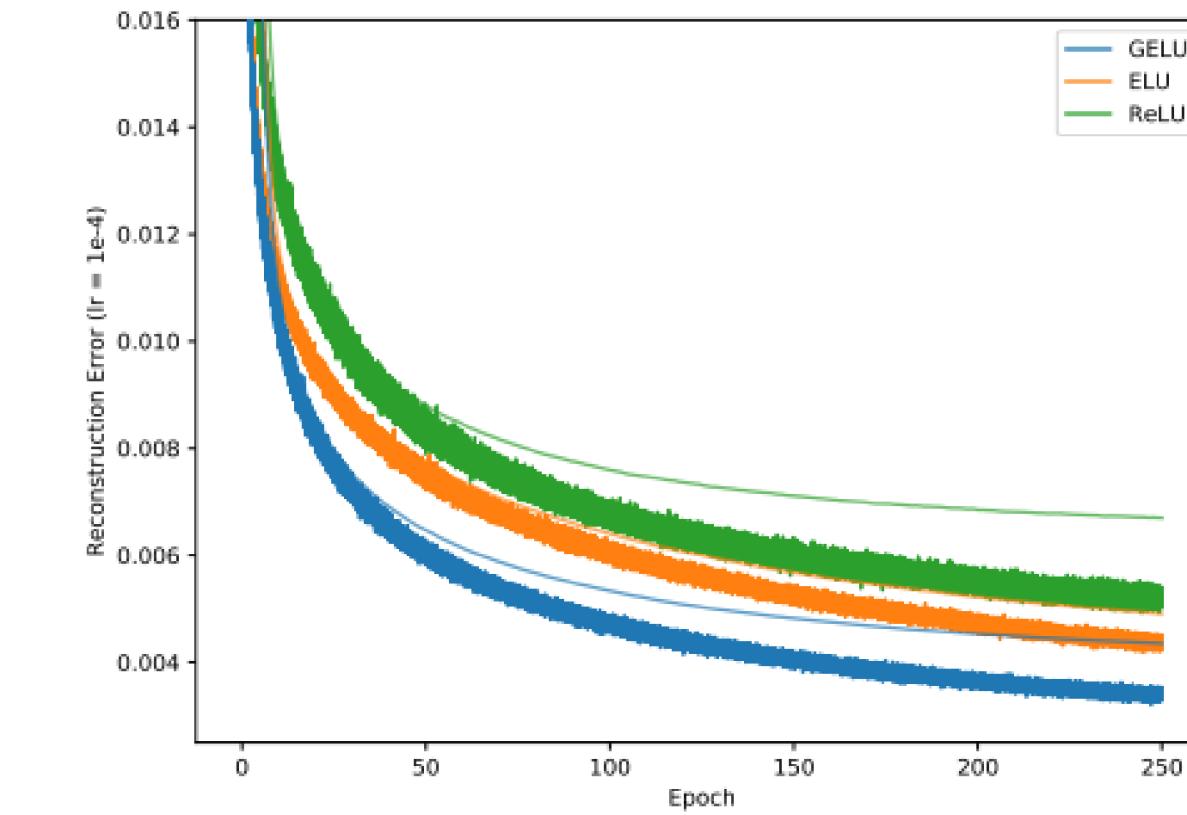
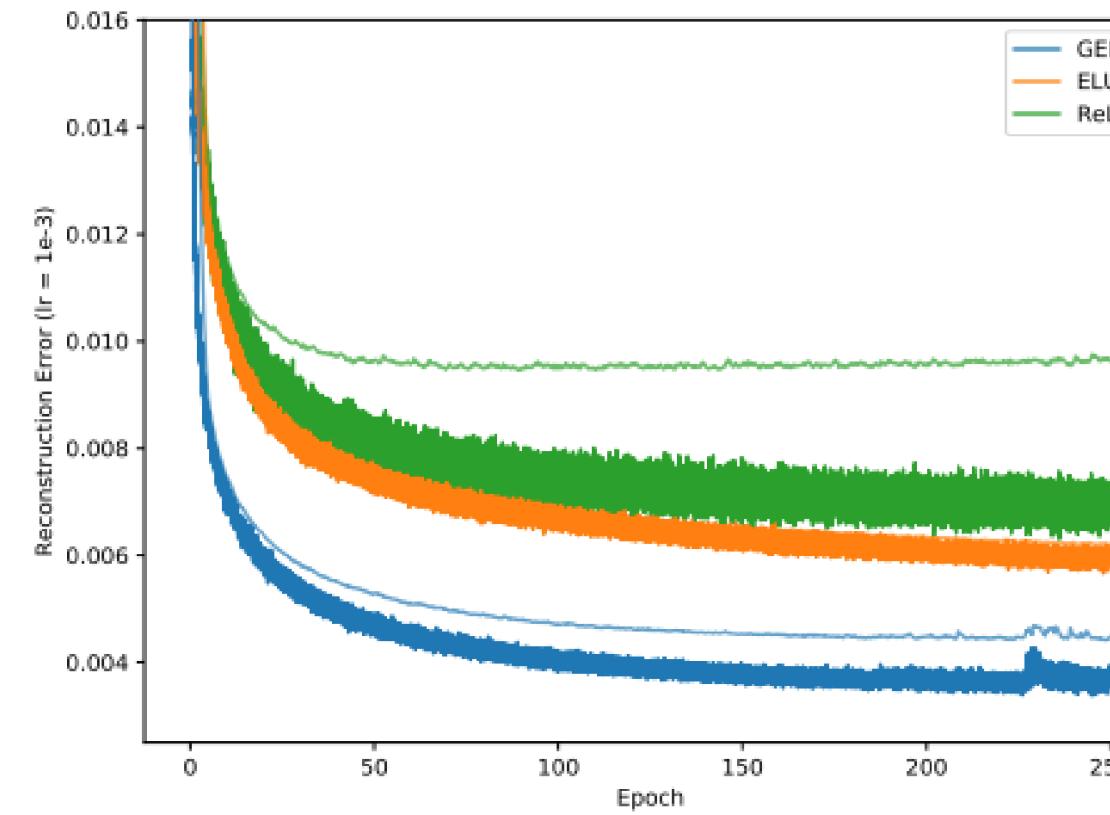


Smooth Non-Linearity

- 부드러운 비선형성 제공 > 미분을 잘 정의 > Vanishing Gradient 해결

GELU

Gaussian Error Linear Unit



Probabilistic Gating Mechanism

- Input 분포에 따른 유동적 활성화
- Input 특성을 더 잘 반영 > 뉴양스를 잘 반영

Transformer 기반 모델에 효과적



LLM Trend



LLM Trend



SwiGLU = Swish + GLU

Gemma

LLM Trend



LLM Trend

LLaMA

∞ Meta

GEGLU = GELU + GLU

LLAMA 2

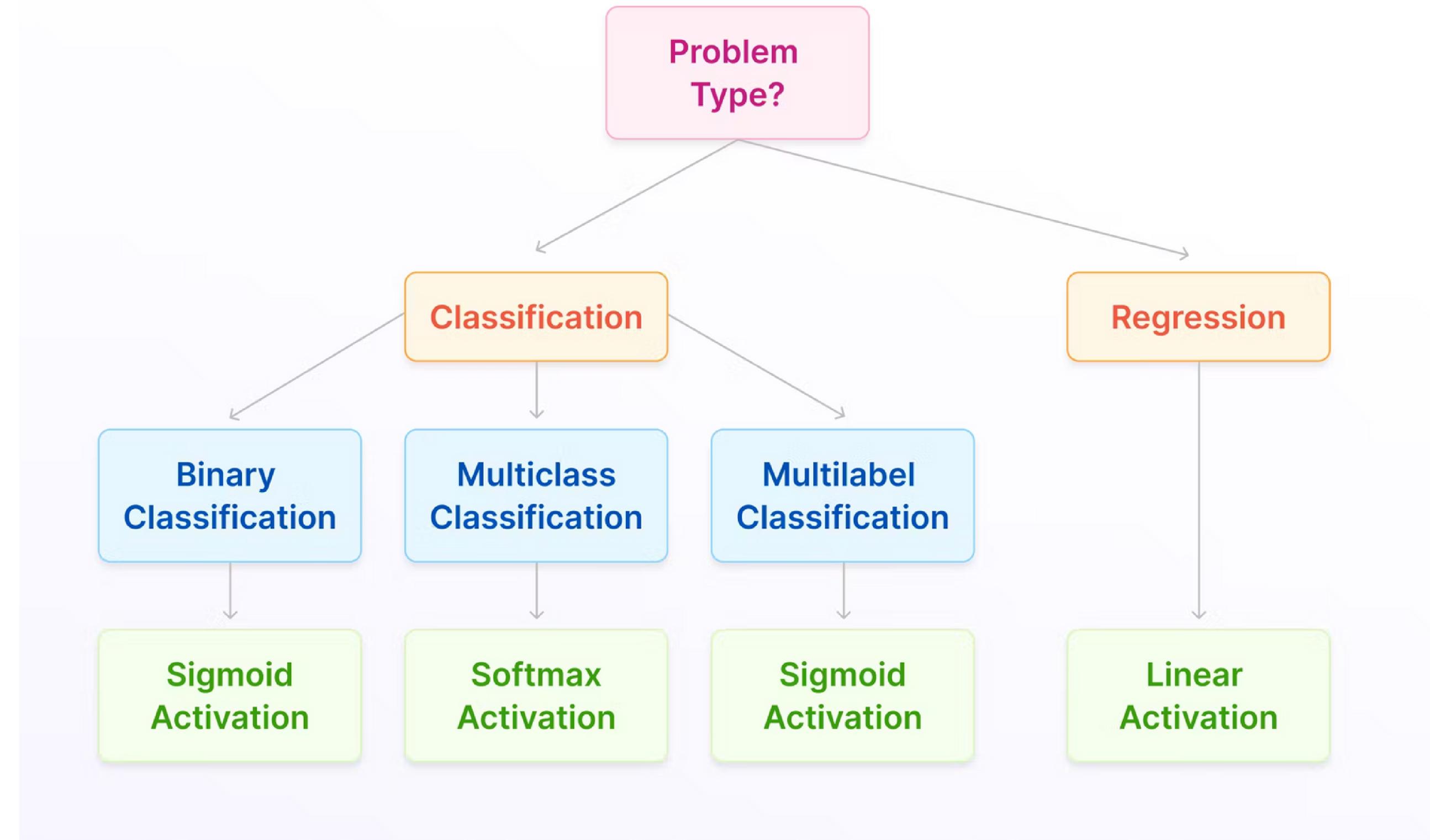


∞ Meta LLAMA 3



Gemma

How to Choose the right Activation Function?



● 참고문헌

- "Parallel Distributed Processing" (1986) - Rumelhart, Hinton, and Williams
"Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters" (1989) - John S. Bridle (Artificial Intelligence Stack Exchange)
"Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit" (2000) - Hahnloser et al.
"Rectified Linear Units Improve Restricted Boltzmann Machines" (2010) - Nair and Hinton
"Rectifier Nonlinearities Improve Neural Network Acoustic Models" (2013) - Andrew L. Maas et al.
"Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)" (2016) - Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter
"Searching for Activation Functions" (2017) - Ramachandran, Zoph, Le
"Gaussian Error Linear Units (GELUs)" (2016) - Dan Hendrycks and Kevin Gimpel
"Is it Time to Swish? Exploring the Time Dimension of Activation Functions" (2021) - Ronen Tamari, Dafna Shahaf.



THANK YOU

- BITAmin 13기 딥러닝 발표 5조
13기 권민지 김재겸 이승우 임채현