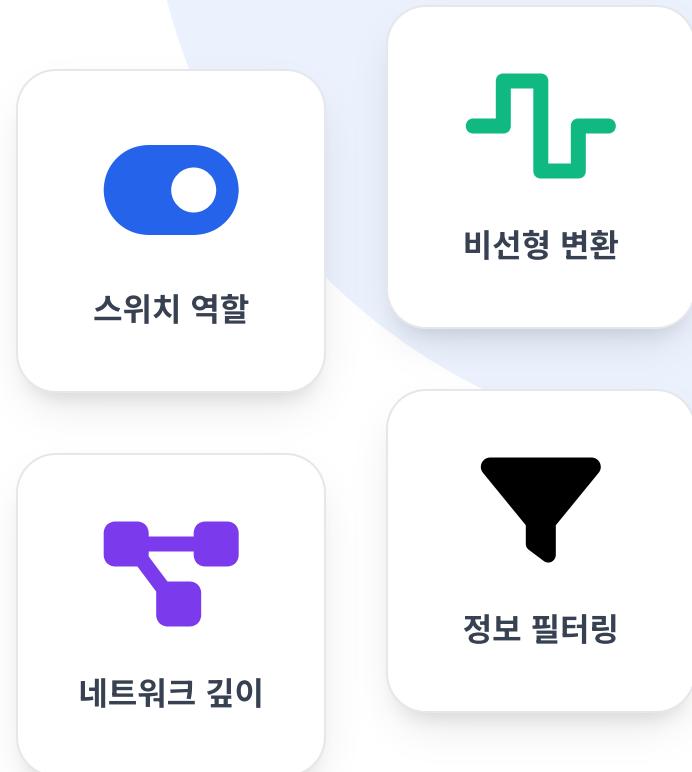


DEEP LEARNING FUNDAMENTALS

신경망의 스위치: 활성화 함수의 역할과 종류

선형 연산의 결과를 비선형으로 변환하는
활성화 함수의 기하학적 의미와
학습 효율에 미치는 영향

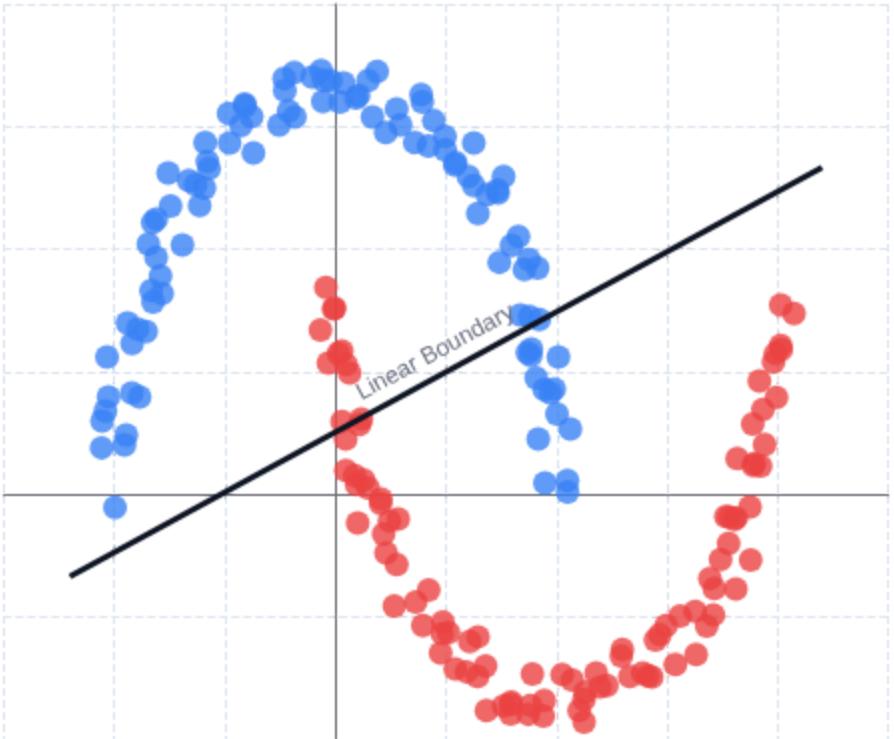


☰ 기하학적 해석

▣ 기울기 소실

⚡ ReLU & 변종

✖ 선형 분리 실패 사례



ⓘ 직선 하나로는 '비선형 패턴(Moons)'을 완벽히 가를 수 없다

INTRODUCTION

왜 $Wx + b$ 만으로는 부족한가?

■ 결정 경계의 한계

선형 모델의 결정 경계는 항상 '직선'이나 '평면'입니다. 세상은 XOR 문제, 나선형(Spiral), 두 개의 달(Moons)처럼 구불구불한 비선형 패턴으로 가득 차 있습니다.



깊이를 늘려도 '직선'

활성화 함수 없이 층만 쌓으면, 수학적으로 행렬의 곱셈이 합쳐져 결국 하나의 선형 변환과 같아집니다.



필요한 것은 '비선형성'

공간을 구부리고 비틀어 복잡한 영역을 구분할 수 있는 마법, 즉 활성화 함수가 필요합니다.

비선형성의 마법: 선형을 쌓아도 선형인 이유

▣ 수학적 직관

2개의 선형 층(Layer)을 쌓았을 때:

$$y = W_2(W_1x + b_1) + b_2$$



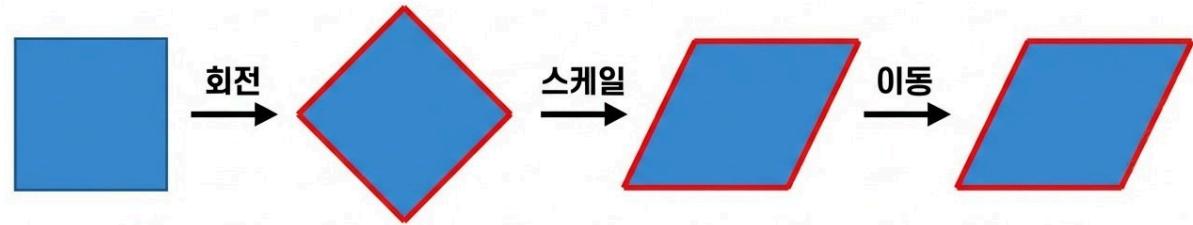
$$y = (W_2W_1)x + (W_2b_1 + b_2)$$



$$y = W'x + b'$$

▲ 결국 또 하나의 선형 함수일 뿐!

기하학적 증명



직선은 계속 직선으로!

직선은 계속 직선이다

아무리 변환해도 모서리가 훠어지지 않음

1

합성 함수의 한계

선형 변환의 합성은 수학적으로 다시 선형 변환이 됩니다. 깊이를 늘려도 표현력은 늘어나지 않습니다.

2

공간 변환의 제약

회전, 스케일, 평행이동만으로는 공간을 '구부릴' 수 없습니다.

활성화 함수(Activation Function)란?

“ 핵심 정의 ”

입력 신호의 총합을 출력 신호로 변환하는 필터 혹은 게이트

뉴런이 입력된 정보(자극)를 받아들였을 때, 이 정보를 다음 뉴런으로 전달
할지 말지, 또는 어떤 강도로 전달할지를 결정합니다.

$$a = \mathbf{f}(\mathbf{Wx} + \mathbf{b})$$

$\mathbf{Wx} + \mathbf{b}$
선형 입력 합



a
변환된 출력



비선형성 (Non-linearity)

단순한 선형 결합만으로는 풀 수 없는 복잡한 문제(XOR 등)를 해결하기 위해, 공간을 '구부리고 비틀어' 표현력을 극대화합니다.



미분 가능성 (Differentiability)

신경망 학습(역전파)을 위해 기울기(Gradient) 계산이 가능해야 합니다. 이는 오차를 줄이는 방향을 알려주는 나침반 역할을 합니다.



출력 분포 재형성

입력값을 0~1 사이의 확률로 변환(Sigmoid)하거나, 음수를 제거(ReLU)하는 등 데이터의 분포를 목적으로 맞게 조절합니다.

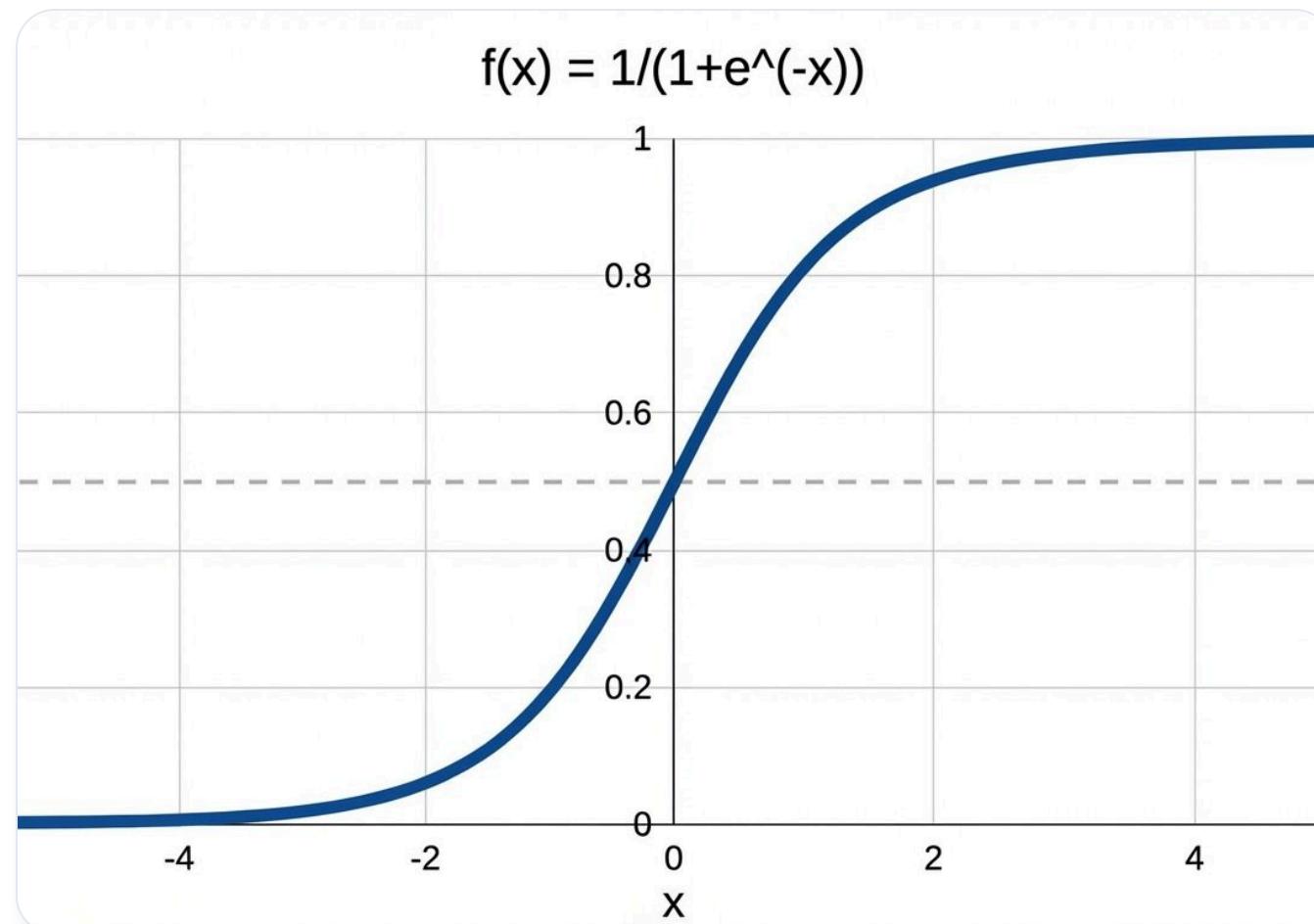


기울기 흐름 조절

역전파 과정에서 오차 신호가 입력층까지 얼마나 잘 전달될지를 결정합니다. 학습의 성패를 좌우하는 핵심 역할을 수행합니다.

시그모이드 (Sigmoid)

가장 고전적이고 직관적인 S자 곡선



MATHEMATICAL FORMULA

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



출력 범위: (0, 1)

출력값이 0과 1 사이로 제한되어 있어 '확률(Probability)'로 해석하기 매우 적합합니다. 이진 분류(Binary Classification)의 출력층에 주로 사용됩니다.



매끄러운 곡선 (Smooth)

모든 지점에서 미분 가능하며, 입력값의 작은 변화가 출력값에 부드럽게 반영됩니다. 신경망 초기 역사에서 가장 널리 쓰였습니다.



도함수의 한계

도함수 $f'(x) = f(x)(1-f(x))$ 의 최댓값은 0.25에 불과합니다. 이는 깊은 망에서 치명적인 '기울기 소실'의 원인이 됩니다.

Sigmoid의 한계: Zero-Centered 아님

! Optimization Problem

✖ 핵심 문제

Sigmoid 함수의 출력값은 항상 **0보다 큰 양수**입니다. (0 ~ 1 범위) 이로 인해 가중치 업데이트가 비효율적으로 일어납니다.

○ 입력이 항상 양수(Positive)

이전 층의 출력(Sigmoid 결과)이 다음 층의 입력으로 들어갈 때, 모든 입력값이 양수가 됩니다.

○ 기울기 부호의 동기화

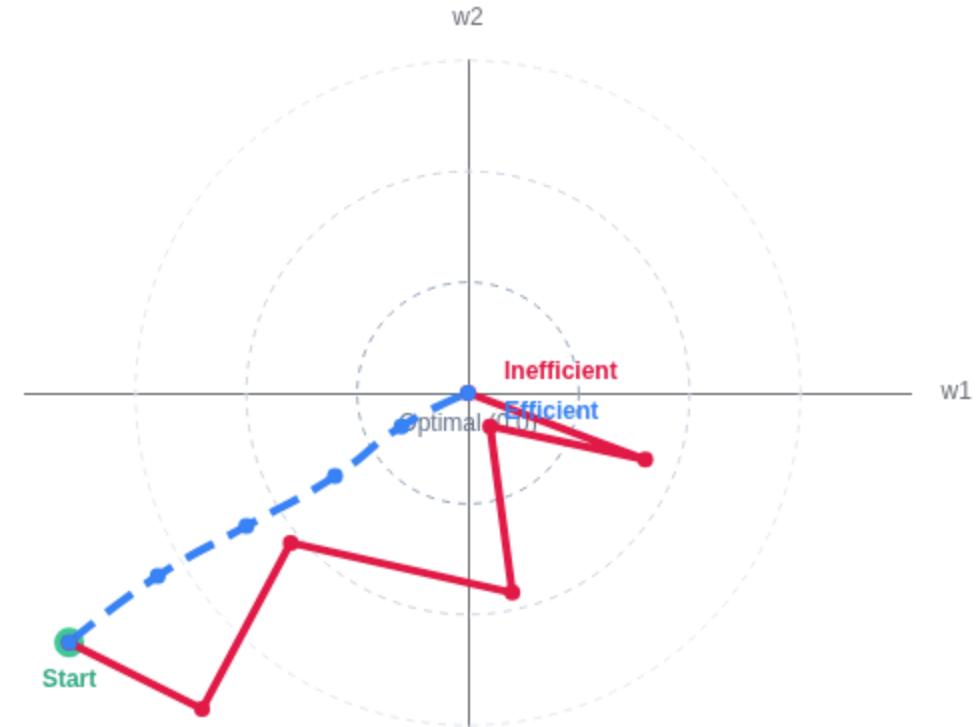
$$\frac{\partial L}{\partial w} = (\frac{\partial L}{\partial a}) \cdot \mathbf{x}$$

입력 x 가 양수이므로, 기울기(Gradient)의 부호는 오차 신호($\frac{\partial L}{\partial a}$)에 의해 결정됩니다. 즉, 모든 가중치가 다 같이 증가하거나, 다 같이 감소합니다.

○ 지그재그(Zigzag) 최적화

원하는 방향으로 바로 가지 못하고, 양수/음수 방향을 오가며 지그재그로 이동하게 되어 학습 속도가 매우 느려집니다.

≠ Gradient Update Path



— Sigmoid (Zigzag Path)

— Zero-Centred Ideal (Direct Path)

Sigmoid의 한계: 기울기 소실

 Gradient Vanishing

❶ 정보의 증발

역전파(Backpropagation) 과정에서 입력층으로 갈수록 **기울기**가 **0에 수렴**하여 학습이 멈추는 현상입니다.

❷ 미분값의 한계 (Maximum 0.25)

$$\max(f'(x)) = \max(f(x)(1-f(x))) = 0.25$$

Sigmoid의 도함수는 최대값이 0.25입니다. 층을 지날 때마다 신호의 크기가 최소 1/4 토막이 납니다.

❸ 연쇄 법칙의 덫 (Chain Rule)

깊은 층을 통과하며 1보다 작은 값들이 계속 곱해집니다.

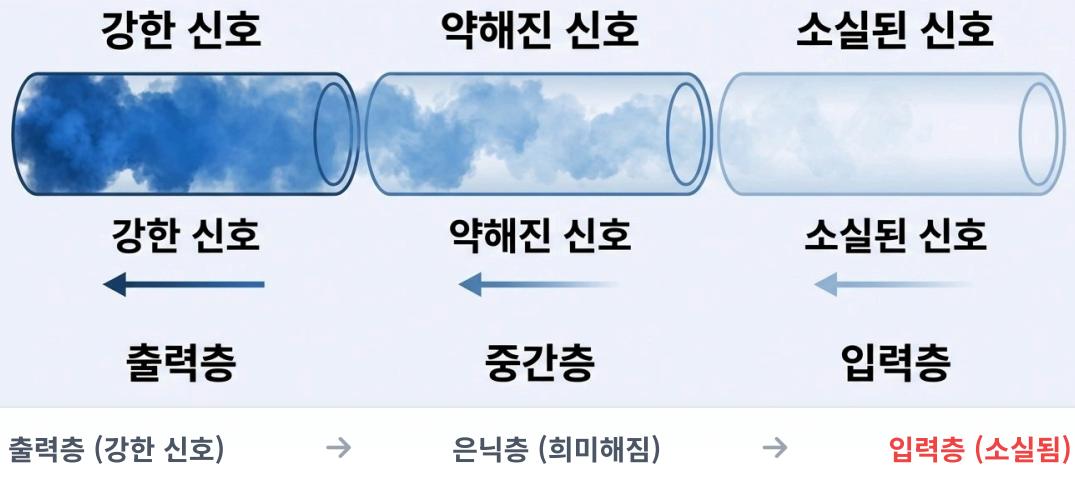
$$0.25 \times 0.25 \times 0.25 \dots \approx 0$$

❹ 결과: 앞단 층의 학습 정지

입력층 근처의 가중치들은 업데이트되지 않아, 깊은 신경망을 쌓는 의미가 사라집니다.

❻ Visual Analogy

기울기 소실 현상 = 정보의 증발

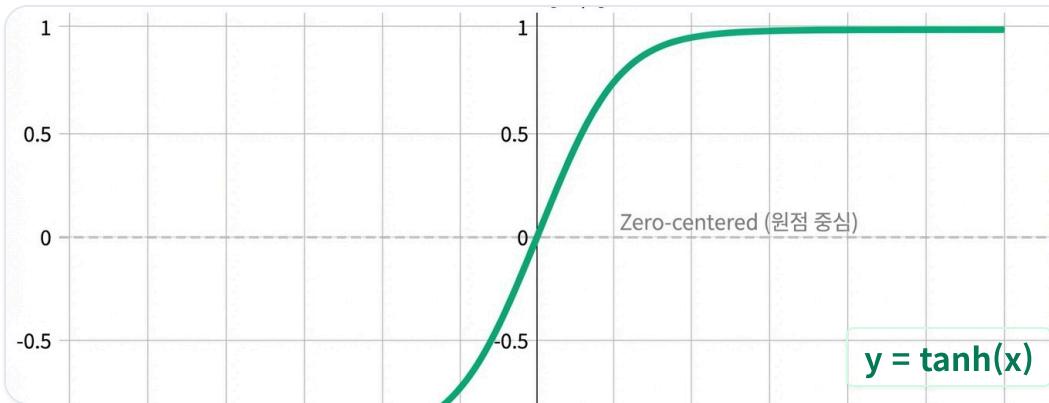


“파이프를 지날 때마다 증기가 새어나가, 끝에는 아무것도 남지 않는 것과 같습니다.”

하이퍼볼릭 탄젠트 (Tanh)



▣ 함수 특성 및 형태



$$\tanh(x) = 2\sigma(2x) - 1$$

※ Sigmoid를 변형하여 크기를 키우고 내린 형태

OUTPUT RANGE

-1 ~ 1

CENTER POINT

0

↔ Sigmoid vs Tanh



Sigmoid

(0, 1)

출력이 항상 양수. 학습 시 파라미터 업데이트가 지그재그(Zigzag)로 비효율적으로 일어남.



Tanh

(-1, 1)

0을 중심으로 양수와 음수 모두 출력. 평균이 0에 가까워져 수렴 속도 개선.

💡 Zero-centered의 중요성

입력 데이터의 중심이 0이 아니면, 역전파 과정에서 기울기가 모두 양수거나 음수가 되어 최적의 경로를 찾아가는 데 시간이 오래 걸립니다. **Tanh**는 이 문제를 해결했습니다.

❗ 주의: 여전히 $|x|$ 가 클 때 기울기 소실(Vanishing Gradient) 문제는 남아있음.

딥러닝의 혁명: ReLU

Rectified Linear Unit

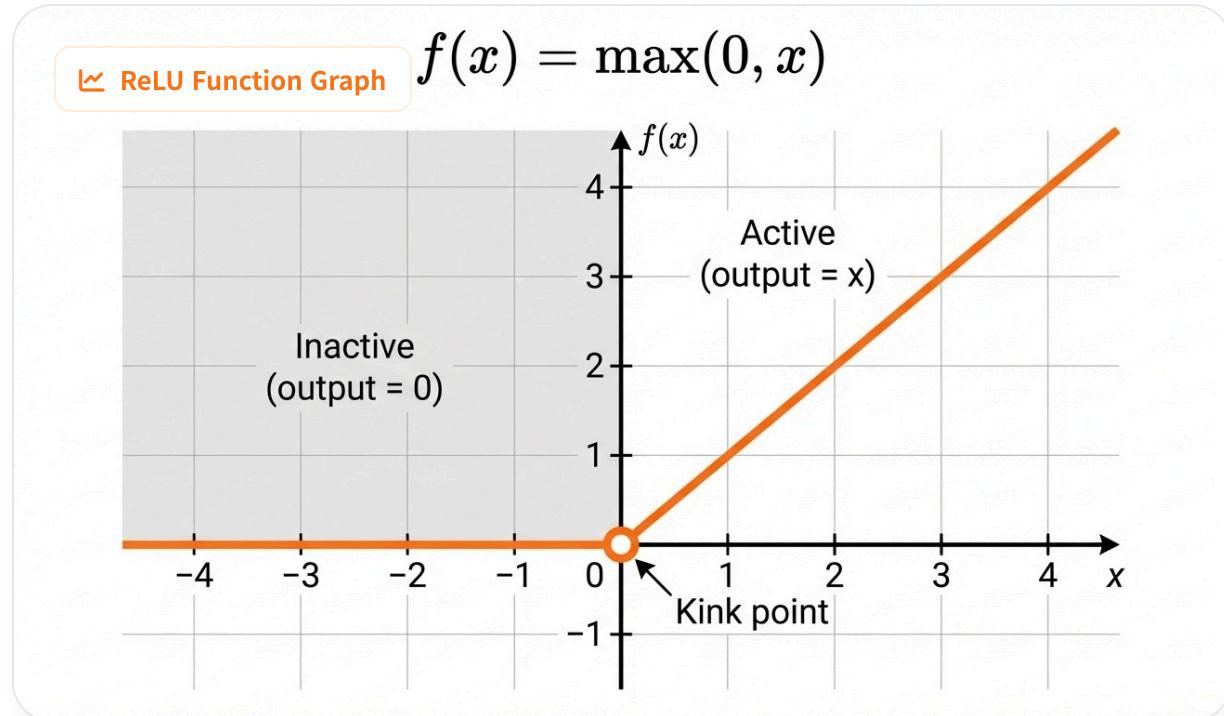
복잡함을 버리고
단순함의 미학을
선택하다.

S자 곡선(Sigmoid, Tanh)이 가진 연산의 복잡함과 기울기 소실 문제를 가장 단순한 '꺾은선' 하나로 해결했습니다.

THE FORMULA

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

⚡ "음수면 0, 양수면 그대로"



연산의 단순함

복잡한 지수 함수(e^x) 계산 없이, 단순한 비교 연산($x > 0?$)만으로 처리되어 학습 속도가 비약적으로 빠릅니다.



희소성 (Sparsity)

음수 입력은 모두 0으로 만들어, 뉴런의 일부만 활성화시킵니다. 이는 뇌의 정보 처리 방식과 유사한 효율성을 가집니다.

ReLU가 왜 빠른가?

Gradient Survival

양수 영역에서
기울기는
절대 죽지 않는다.

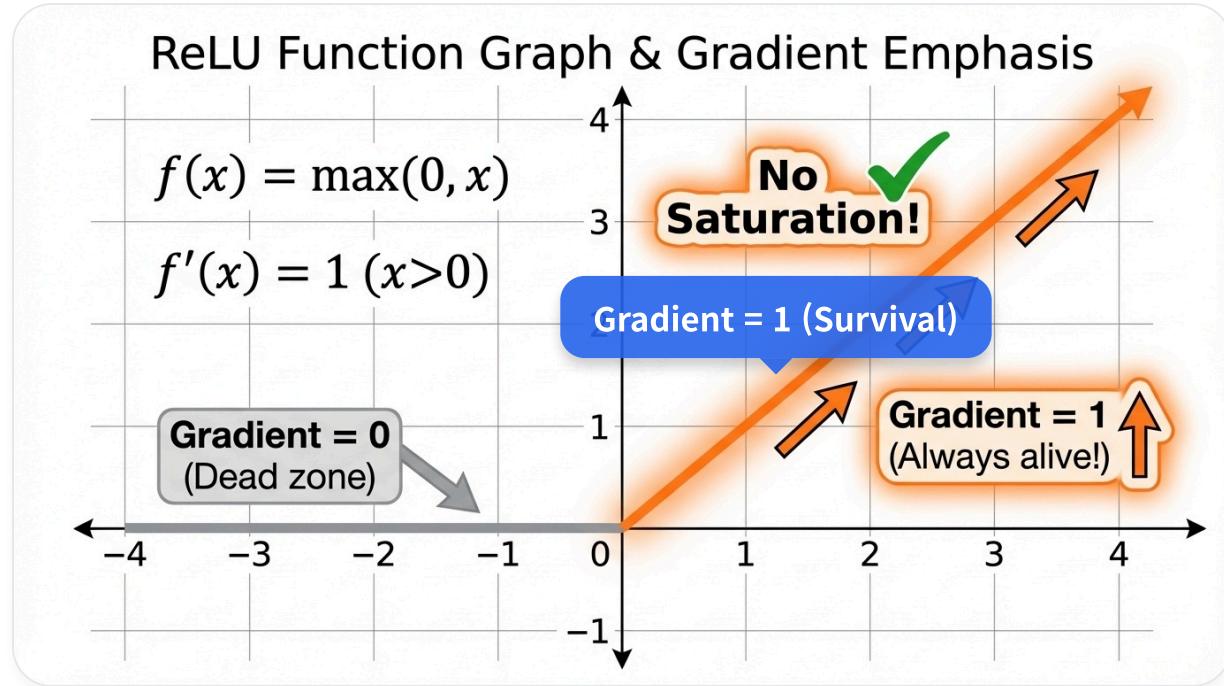
입력이 양수라면 미분값은 항상 1입니다.

이는 깊은 신경망에서도 신호가 약해지지 않고 끝까지 살아서 전달 됨을 의미합니다.

THE DERIVATIVE

$$f'(x) = 1 \quad (\text{if } x > 0)$$

✓ 포화(Saturation) 현상 완벽 제거



기울기 보존 (Gradient Survival)

Sigmoid나 Tanh는 양 끝단에서 기울기가 0에 가까워지지만, ReLU는 양수 구간에서 기울기가 1로 유지되어 학습 신호를 '고속도로'처럼 전달합니다.



최적화 가속

복잡한 지수 연산이 없고, 선형적 특성 덕분에 경사 하강법(Gradient Descent)의 수렴 속도가 기존 함수 대비 약 6배 이상 빠릅니다.

▲ CRITICAL ISSUE

ReLU의 치명적 약점

Dying ReLU 문제



⚡ Signal Lost

'Dying ReLU'



입력 $< 0 \rightarrow$ 출력 = 0 \rightarrow 기울기 = 0 \rightarrow 학습 정지!

Input $x < 0$	\rightarrow	Output 0	\rightarrow	Gradient 0
------------------	---------------	-------------	---------------	---------------

ⓘ 음수 영역에 빠진 뉴런은 더 이상 학습되지 않고 '죽은' 상태가 됩니다.

뉴런이 영원히 잠들어 버리다

학습 도중 가중치가 업데이트되어 뉴런이 음수 입력만 받게 되면, 출력과 기울기가 모두 0이 됩니다. 이후 역전파 시에도 기울기가 전달되지 않아 **가중치가 영원히 업데이트되지 않는 상태**에 빠집니다.

🔍 주요 원인 (CAUSES)

- **높은 학습률 (Large Learning Rate)**
가중치가 큰 폭으로 변해 음수 영역으로 '점프'해버림.
- **데이터 분포의 치우침**
입력 데이터 자체가 음수 위주일 때.

💡 해결책 (SOLUTIONS)

- **Leaky ReLU / ELU 사용**
음수 영역에서도 미세한 기울기를 주어 회생 가능하게 함.
- **Batch Normalization**
입력 분포를 강제로 정규화하여 0 근처로 모음.

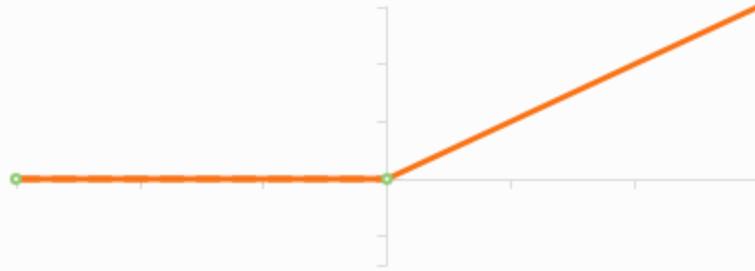
ReLU의 변종들: 음수 영역을 살려라

 SOLUTION FOR
Dying ReLU

STANDARD

ReLU

Rectified Linear Unit



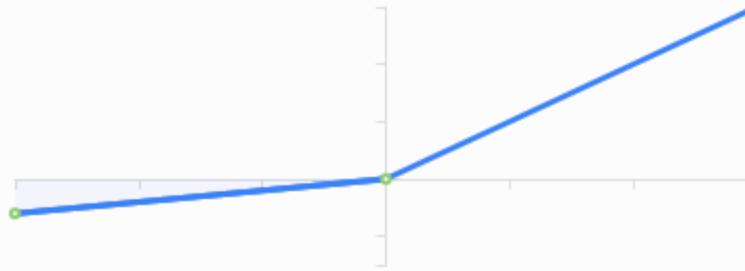
$$\max(0, x)$$

- ✓ 단순함: 연산 속도가 가장 빠름.
- ✓ 희소성: 불필요한 뉴런 비활성화.
- ⚠ Dying ReLU: 음수 입력 시 기울기가 0이 되어 학습 중단 위험.

VARIANT A

Leaky ReLU

Leaky Rectified Linear Unit



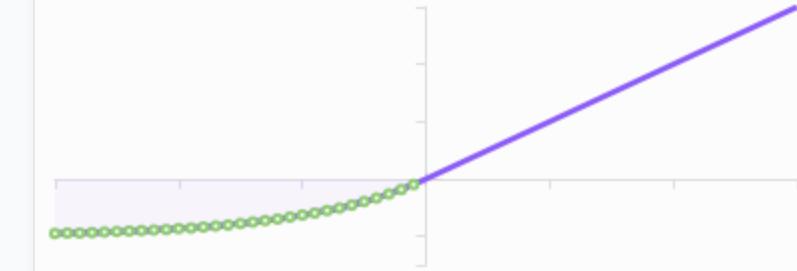
$$\max(0.01x, x)$$

- ✓ 음수 보존: 0이 아닌 작은 기울기(0.01)를 가짐.
- ✓ Dying 방지: 뉴런이 죽지 않고 계속 학습 가능.
- ℹ Parametric: 기울기를 파라미터로 학습 가능(PReLU).

VARIANT B

ELU

Exponential Linear Unit

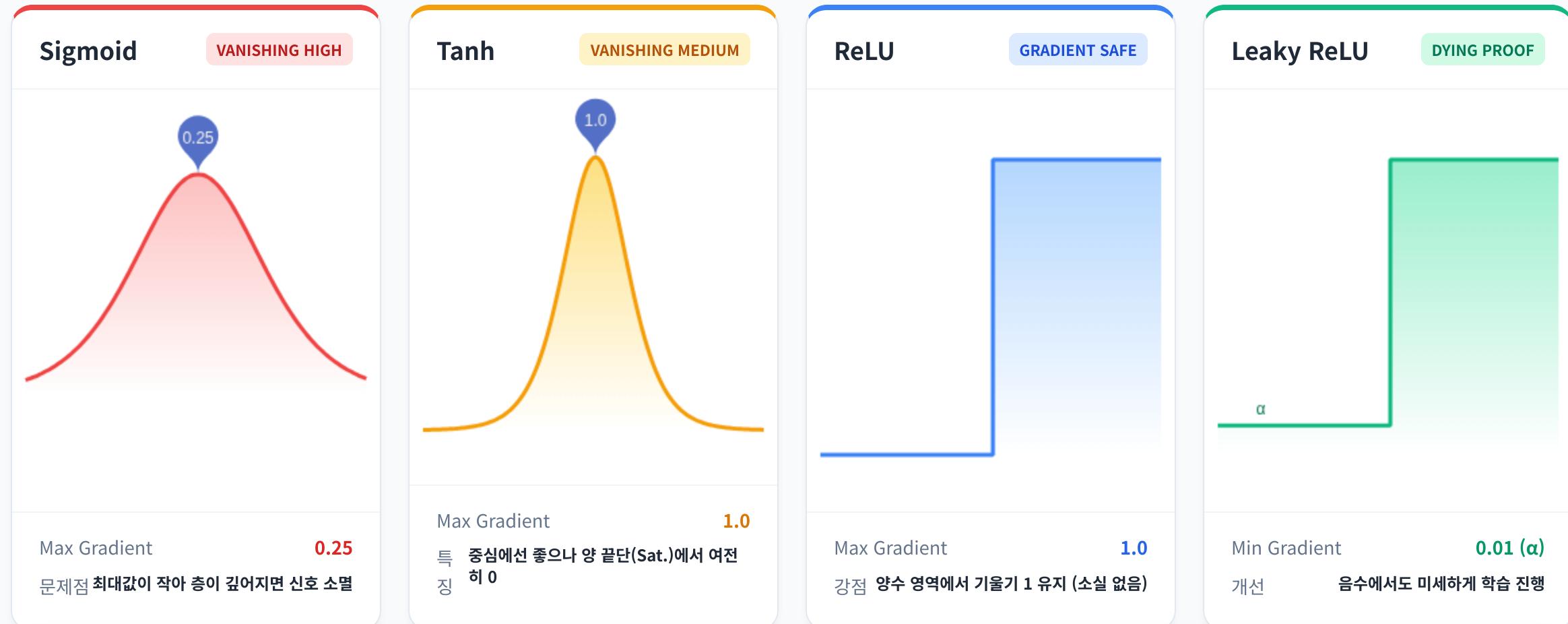


$$x > 0 ? x : \alpha(e^x - 1)$$

- ✓ 부드러움: 미분 불가능 지점 없이 매끄러운 곡선.
- ✓ 노이즈 강함: 음수 영역 포화로 노이즈에 강인함.
- ⚠ 비용: 지수 함수(exp) 계산으로 연산 비용 약간 높음.

함수별 도함수(Derivative) 비교

💡 Why Derivative Matters? 도함수 값 = 학습 속도



핵심 인사이트

도함수의 값이 1에 가까운 구간이 넓을수록 깊은 신경망에서도 학습 정보가 잘 전달됩니다.

RECOMMENDATION

Hidden Layer에는 ReLU 권장

선택 가이드: 어디에 무엇을 쓸까?

Layer Type & Problem Type Matrix



은닉층 (Hidden Layer)

데이터의 특징을 추출하는 내부 레이어

✓ 1순위 (DEFAULT)

대부분의 경우 가장 먼저 시도

계산 효율이 좋고 기울기 소실 문제가 적어, 깊은 신경망에서도 학습이 안정적이고 빠릅니다.

ReLU

➊ 문제 발생 시 (DYING RELU)

뉴런이 죽거나(0 출력) 학습이 정체될 때

음수 영역에서도 미세한 기울기를 전달하여 뉴런을 되살리고 학습의 유연성을 높입니다.

Leaky ReLU / ELU

⚙ 특수 상황 (RNN 등)

순환 신경망(RNN)이나 데이터 중심화가 중요할 때

-1~1 범위로 출력을 유지하여 기울기 폭발을 방지하는 효과가 있습니다.

Tanh



출력층 (Output Layer)

최종 결과를 예측하는 마지막 레이어

PROBLEM TYPE

이진 분류

(Binary Classification)

Sigmoid

Range: [0, 1]

출력을 0과 1 사이의 확률값으로 변환.
(예: 스팸 메일 여부, 합격/불합격)

PROBLEM TYPE

다중 분류

(Multi-class Classification)

Softmax

Sum = 1.0

각 클래스에 속할 확률을 출력하며, 총합이 1이 됨.
(예: 숫자 인식 0~9, 옷 종류 분류)

PROBLEM TYPE

회귀

(Regression)

Identity (Linear)

Range: (-∞, ∞)

활성화 함수 없이 값 그대로 출력.
(예: 집 값 예측, 온도 예측)

핵심 요약 (Summary)



활성화 함수가 신경망 학습에 미치는 결정적 영향



비선형성의 마법

활성화 함수는 선형 공간을 '**구부리고 비틀어**' 복잡한 패턴을 표현할 수 있게 만듭니다. 이것이 없다면 깊은 신경망도 단지 하나의 선형 함수일 뿐입니다.



기울기 소실의 위험

Sigmoid와 Tanh는 입력값이 커질수록 기울기가 0에 수렴하여 '**정보의 증발**'을 초래합니다. 깊은 층까지 학습 신호가 도달하지 못하게 막습니다.



ReLU의 혁신

"양수는 그대로, 음수는 0으로." 가장 단순한 형태로 연산 효율성을 높이고, 양수 영역에서 **기울기를 온전히 보존**하여 딥러닝의 깊이를 열었습니다.



기울기 흐름 (Gradient Flow)

학습의 성패는 역전파 시 **기울기가 끊김 없이 흐르느냐**에 달려 있습니다. 죽은 뉴런(Dying ReLU)을 피하고 적절한 활성화 함수를 선택해야 합니다.



결국 딥러닝 학습은 올바른 활성화 함수를 통해 기울기를 살리는 것입니다.