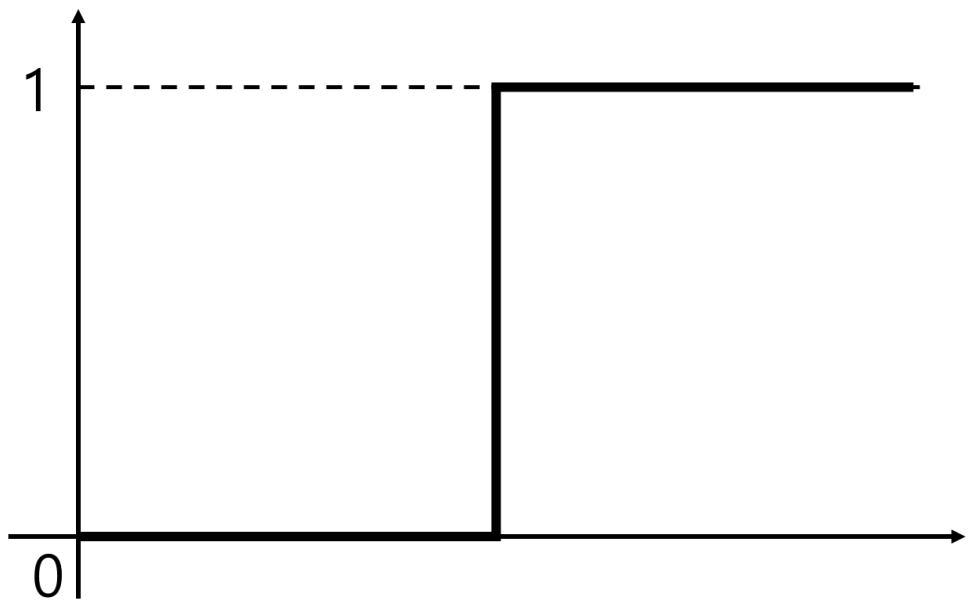
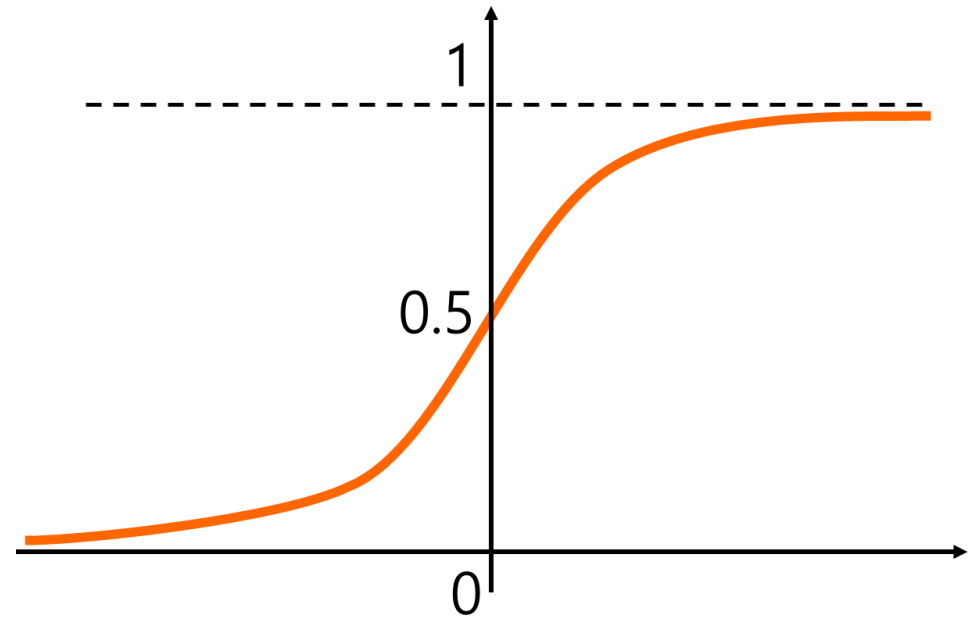
Introduction to 인공 신경망

시그모이드 활성화 함수



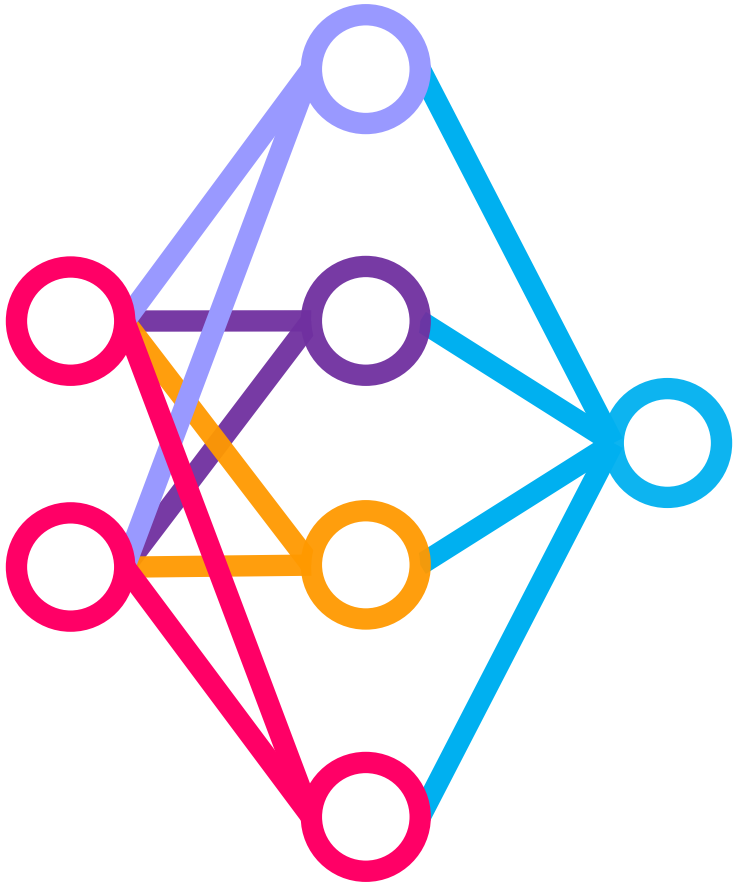


신박AI

안녕하세요 신박AI입니다

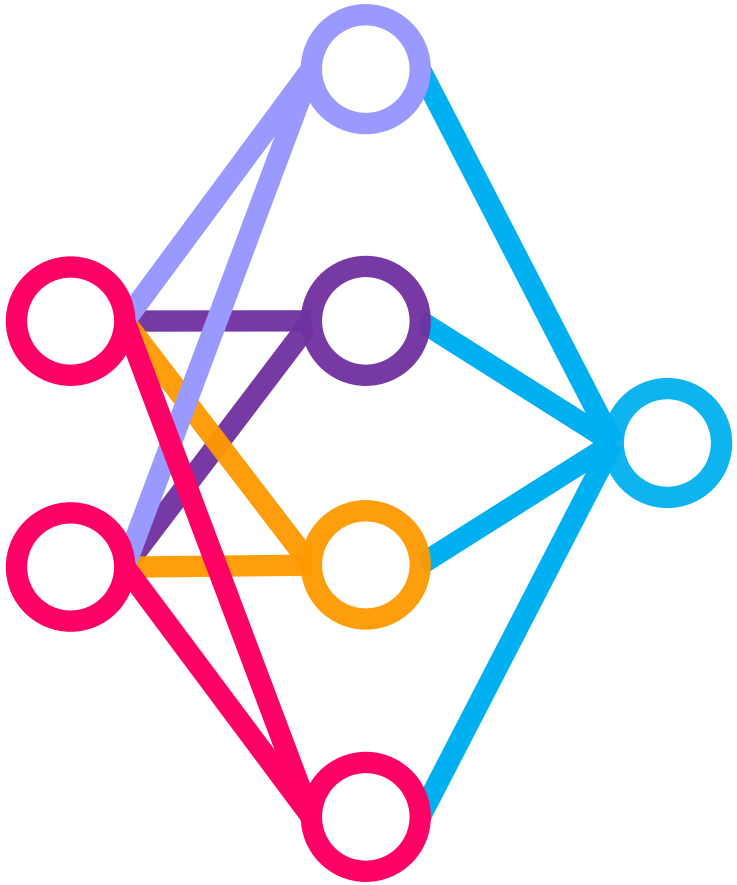
신박AI

지난 영상에서는 퍼셉트론의 한계와 다층신경망을 소개해드렸습니다



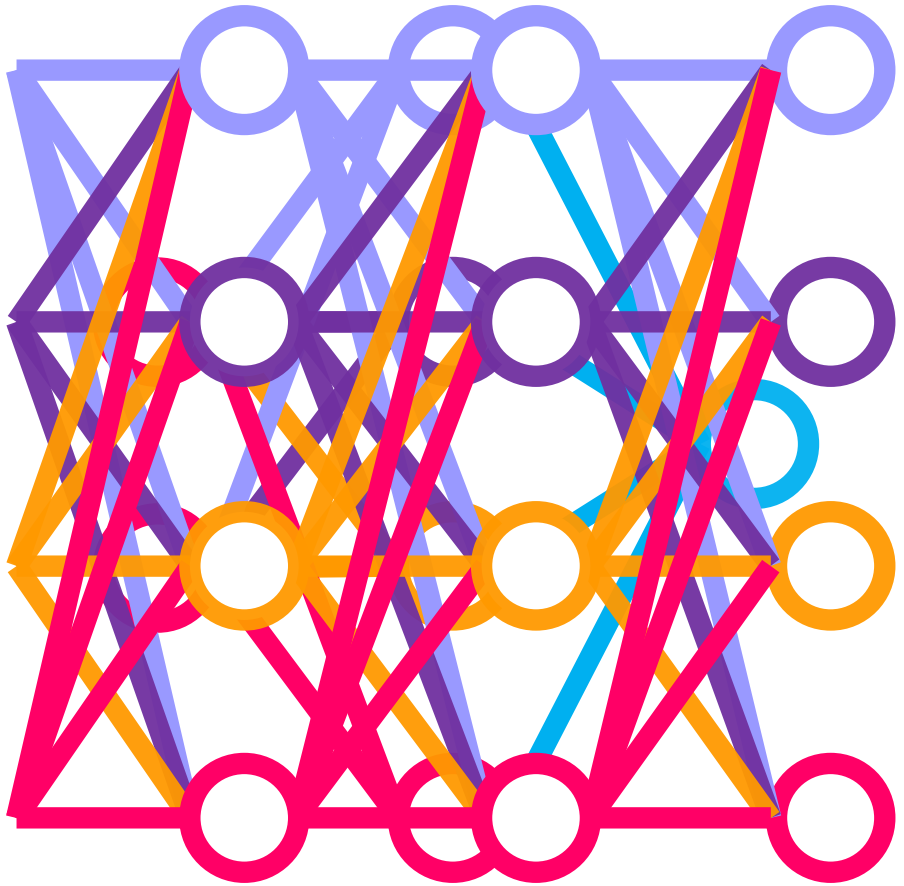
신박AI

다층 신경망은 단층 퍼셉트론에 비해 여러모로 파워풀한 신경망입니다



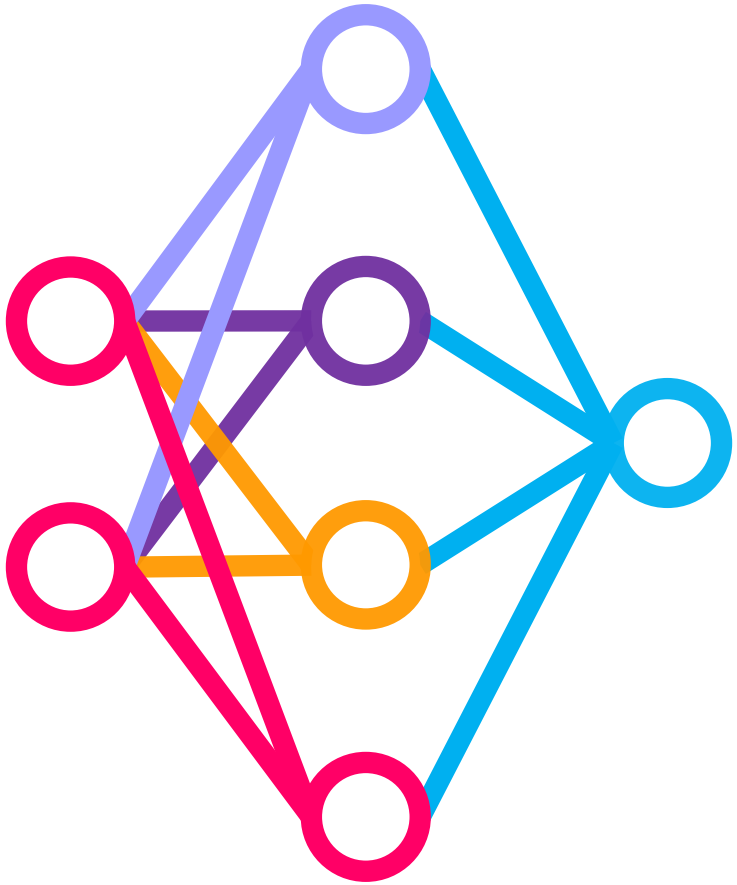
신박AI

다층 신경망이 더욱 발전하여 결국 오늘날 딥러닝까지 이르게 되는데요



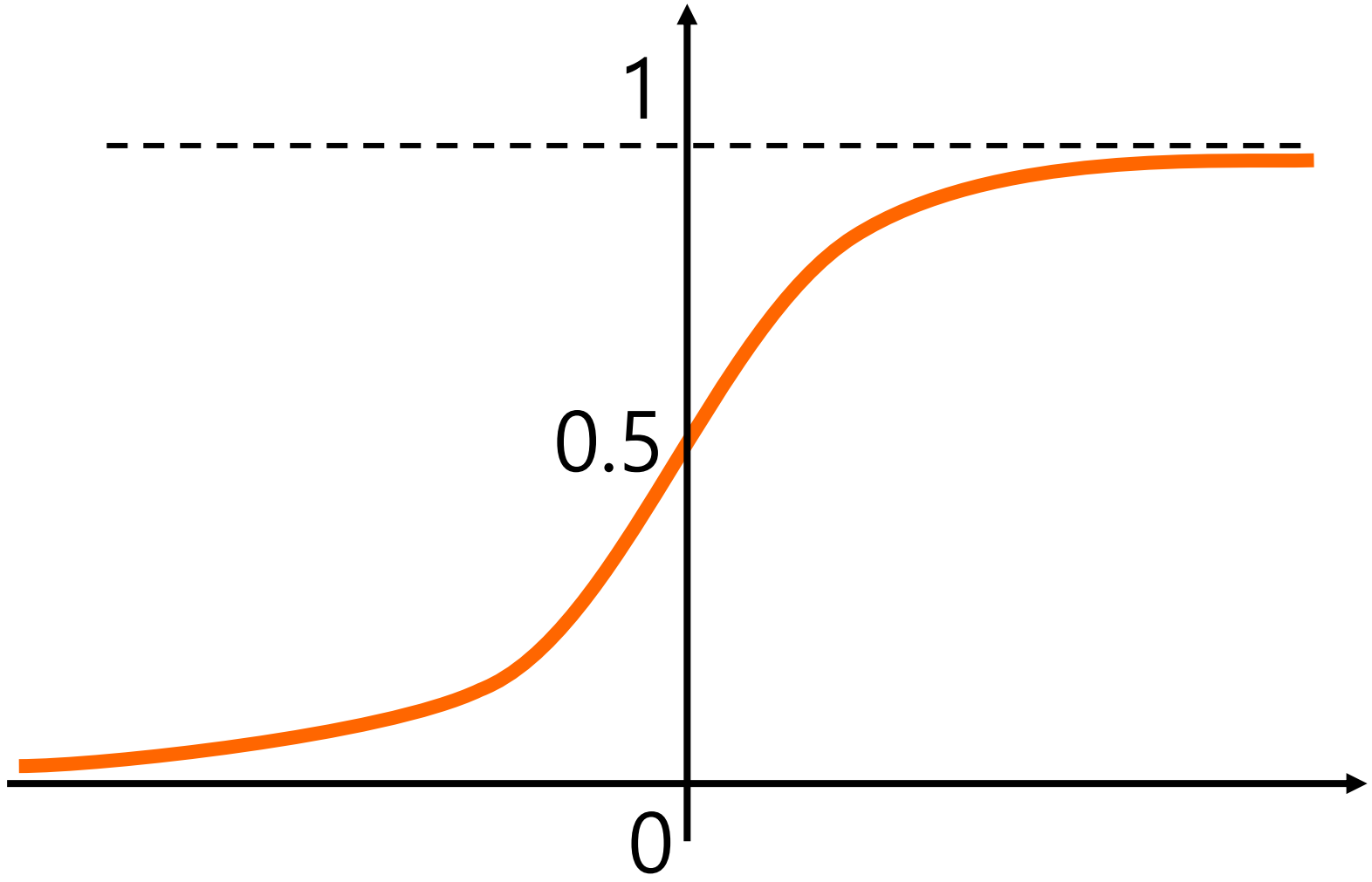
신박AI

다층 신경망은 퍼셉트론에 비하여 많은 점이 달라졌습니다



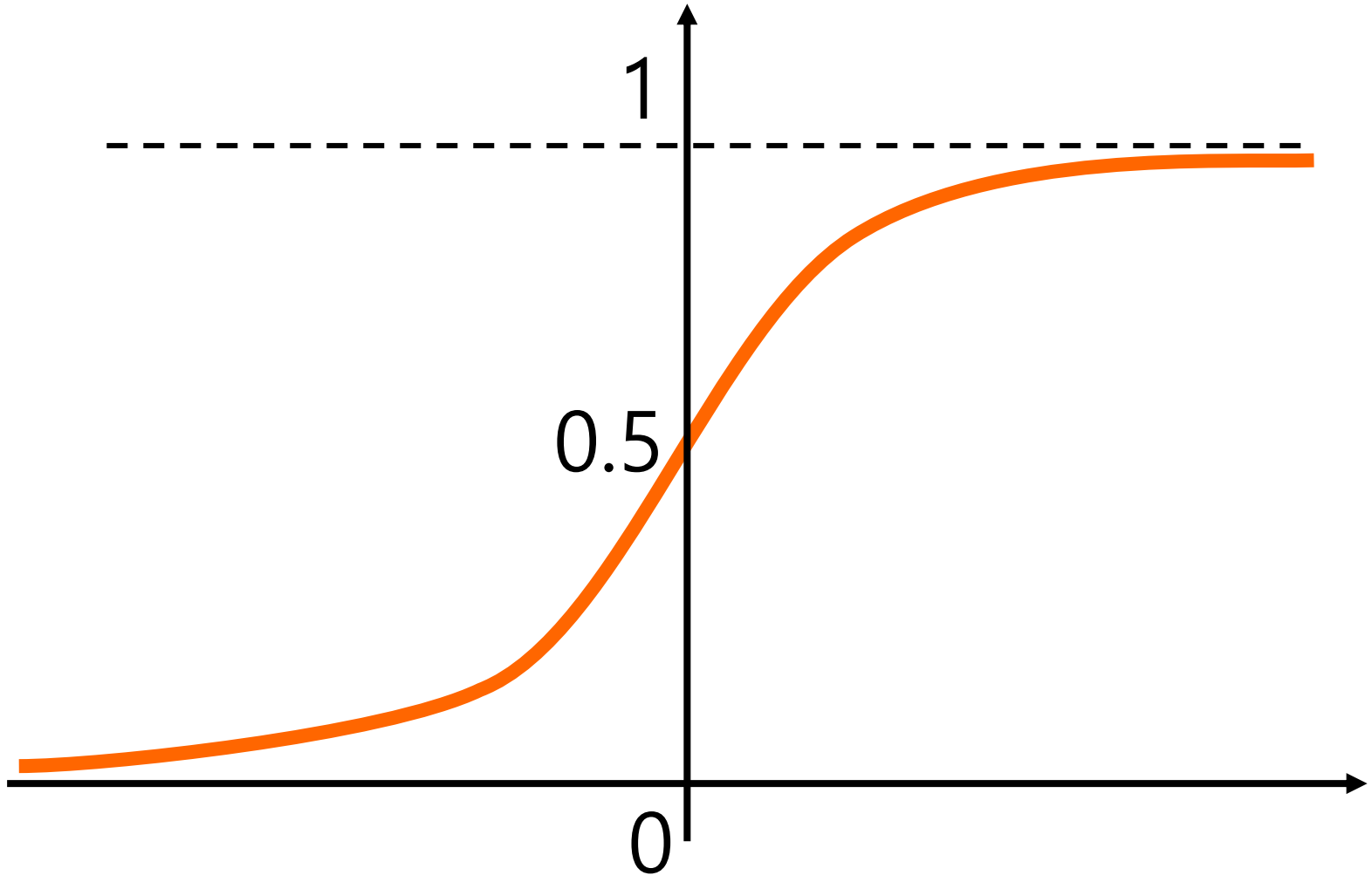
신박AI

그 중에 오늘 함께 배워볼 부분은 활성화 함수입니다



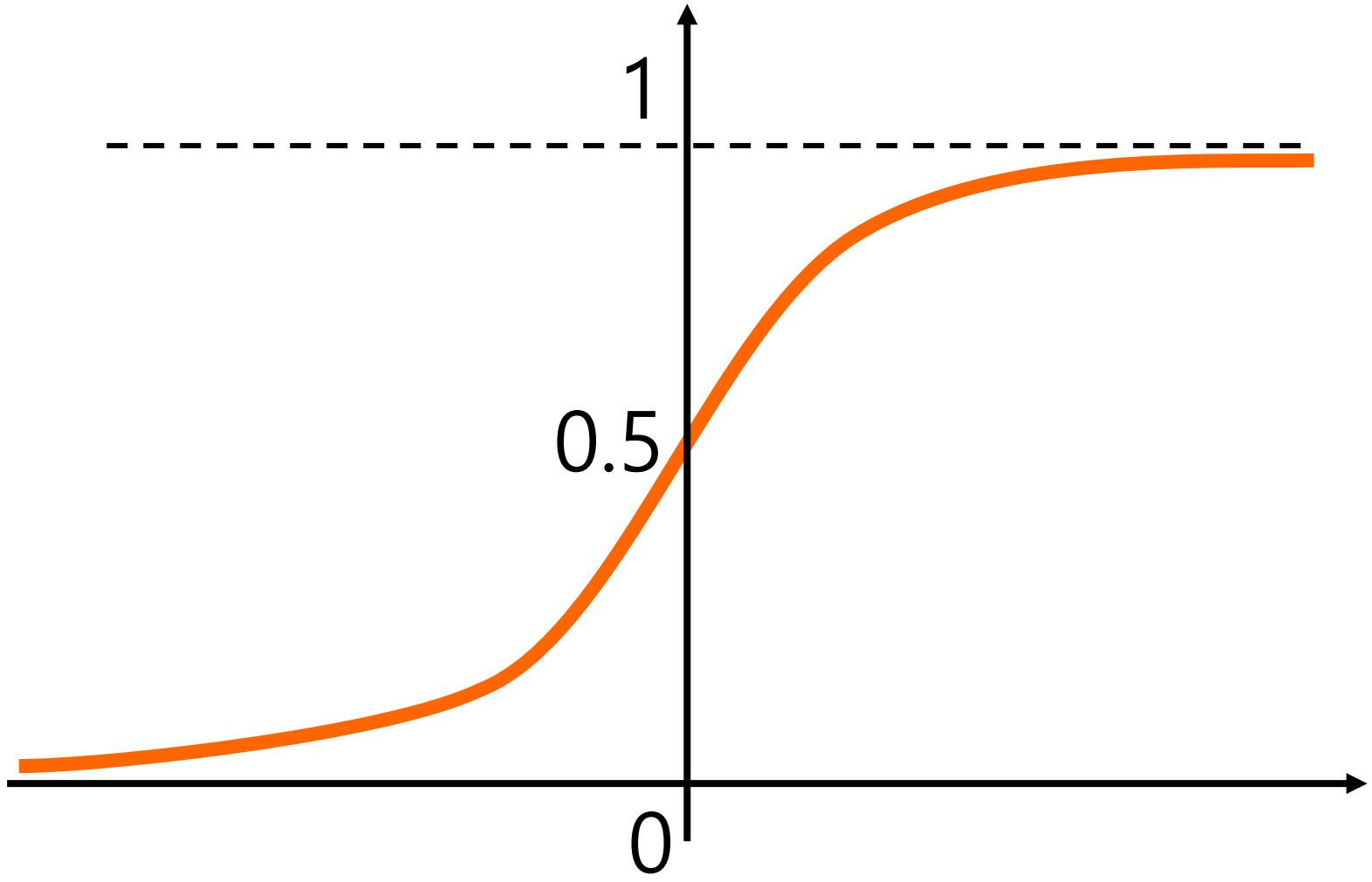
신박AI

오늘 배울 활성화 함수는



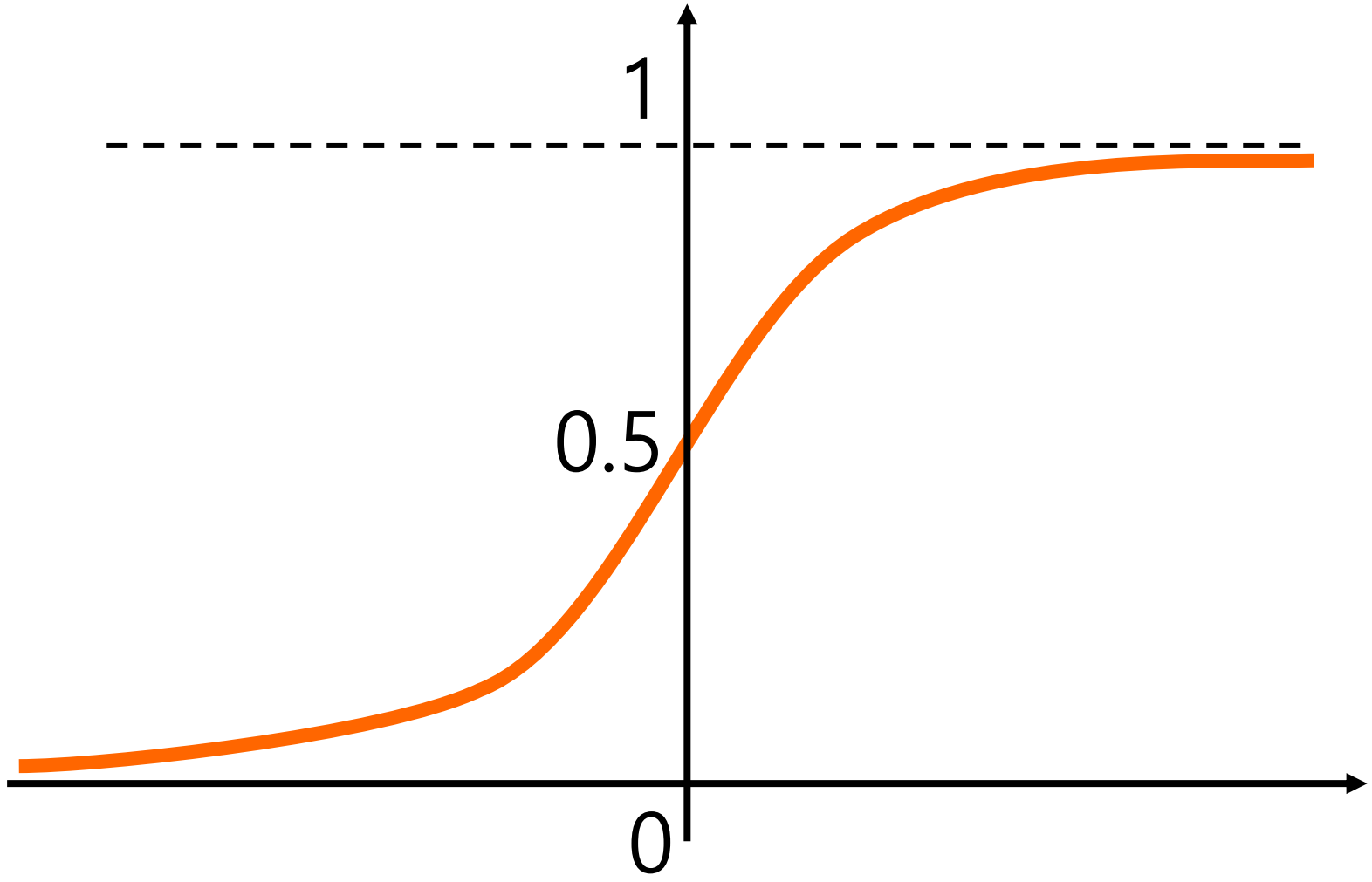
신박AI

오늘 배울 활성화 함수는 어찌보면 상대적으로 단순했던 신경망을



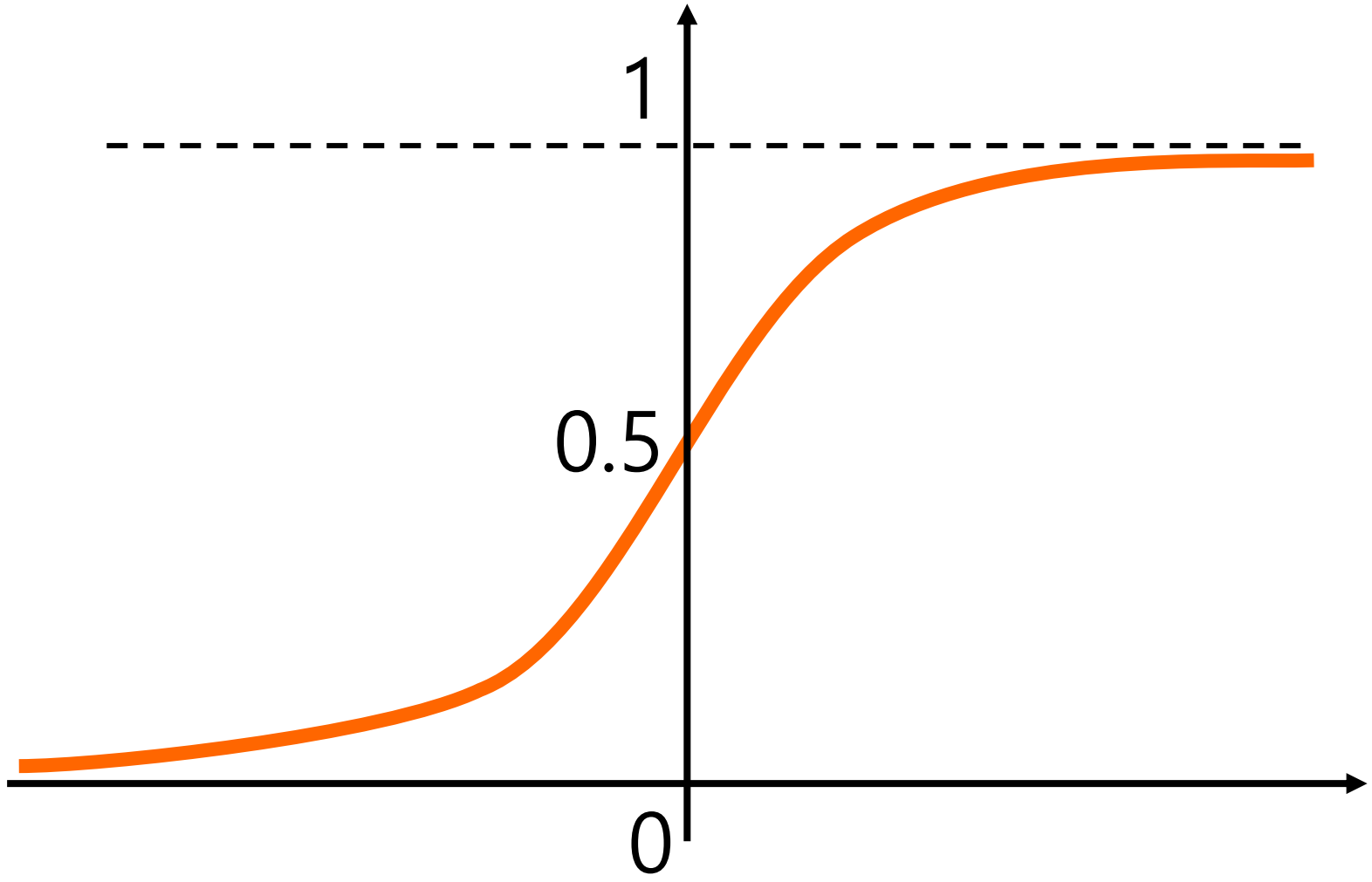
신박AI

딥러닝처럼 복잡하고 다양한 기능을 갖는 신경망으로 발전할 수 있게 만든 중요한 초석이 되었습니다



신박AI

그래서 오늘 활성화 함수에 대해 차근차근 함께 알아보도록 하겠습니다



신박AI

이 채널은 여러분의 관심과 사랑이 필요합니다

신박AI

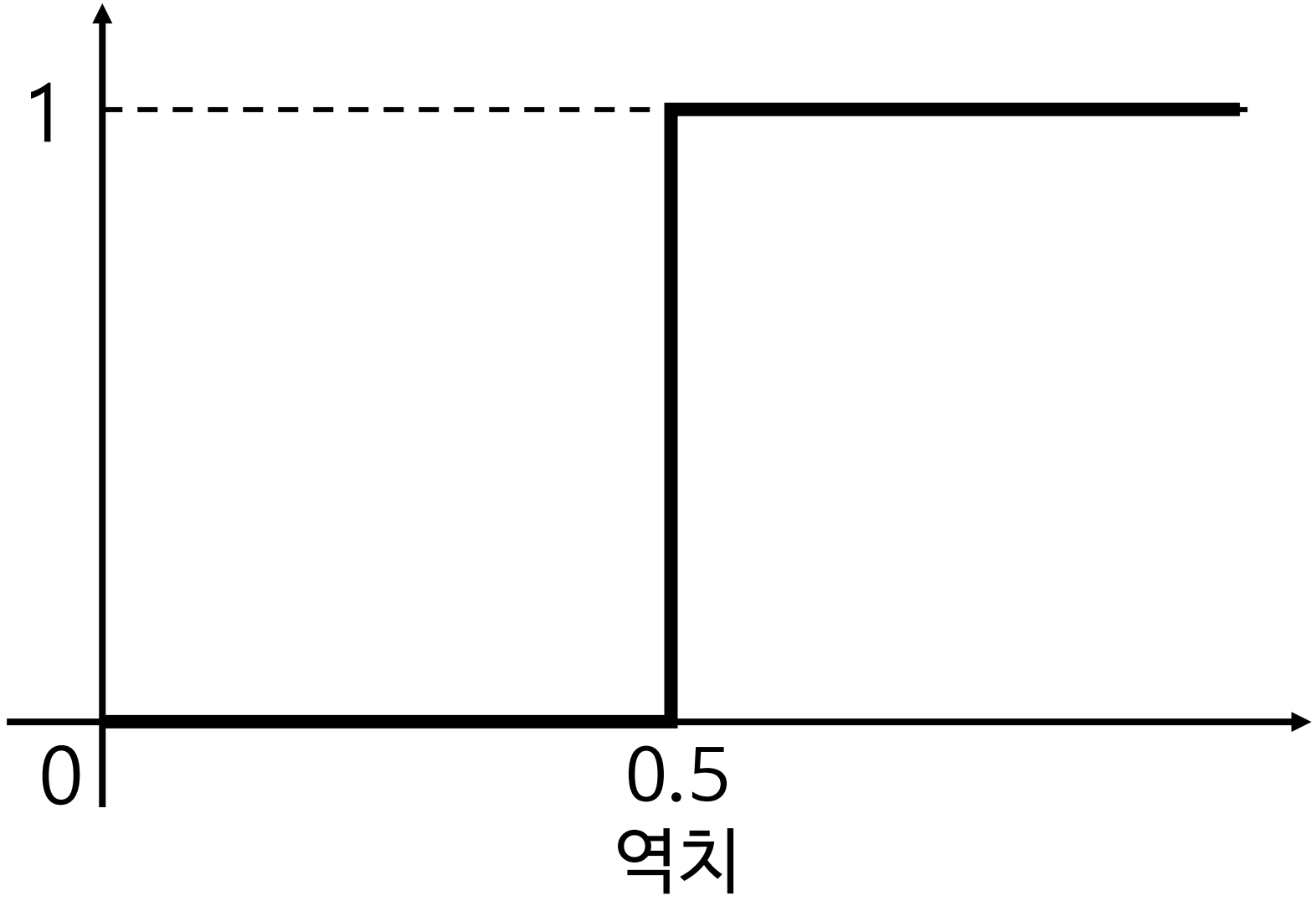
‘좋아요’와 ‘구독’버튼은 강의 준비에 큰 힘이 됩니다!

신박AI

Chapter 1
계단 함수의 한계

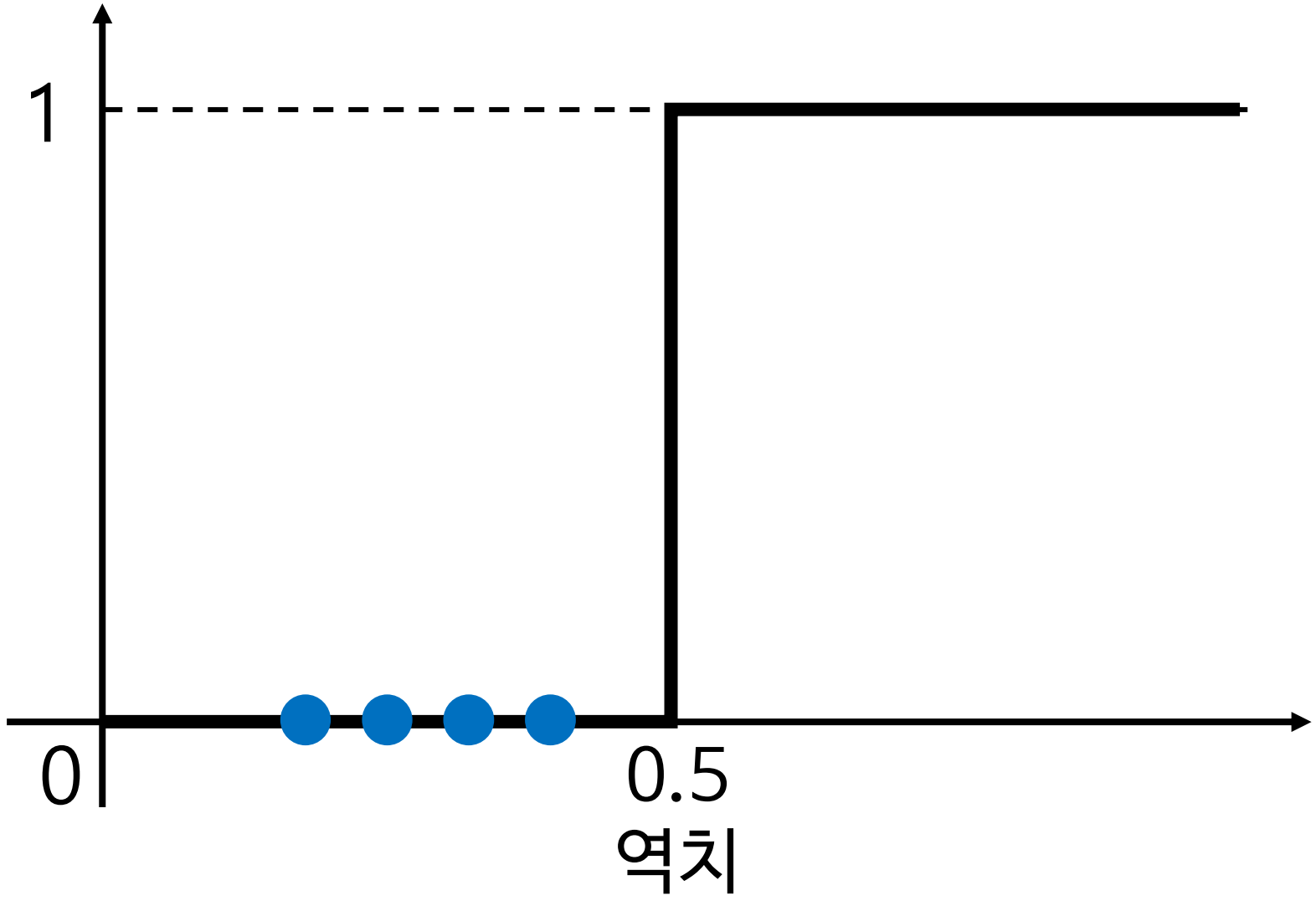
신박AI

지난 영상에서 보여드렸던 활성화 함수는 계단함수 였습니다



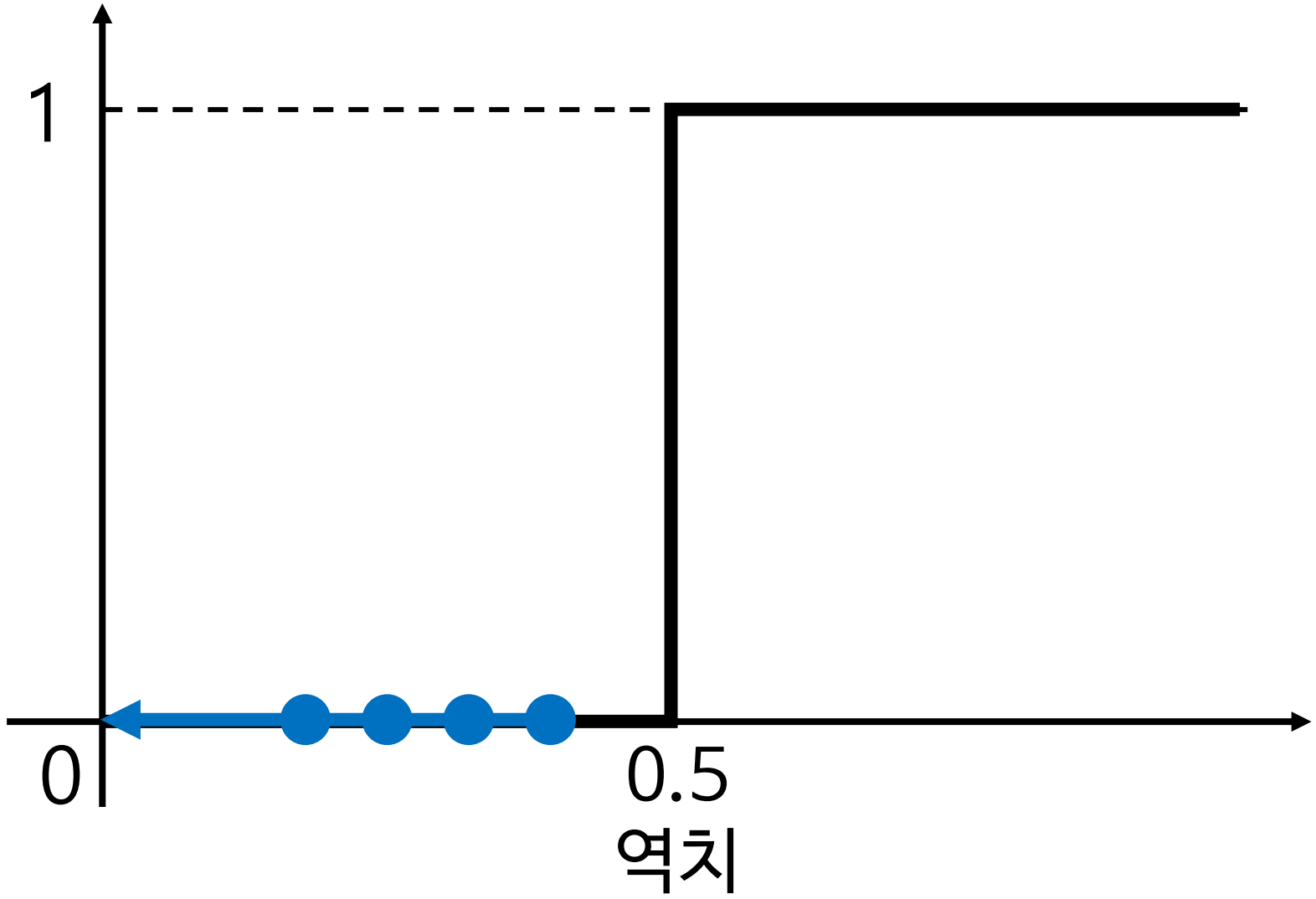
신박AI

계단함수는 함수의 값이 특정값 (역치)이하일 경우는



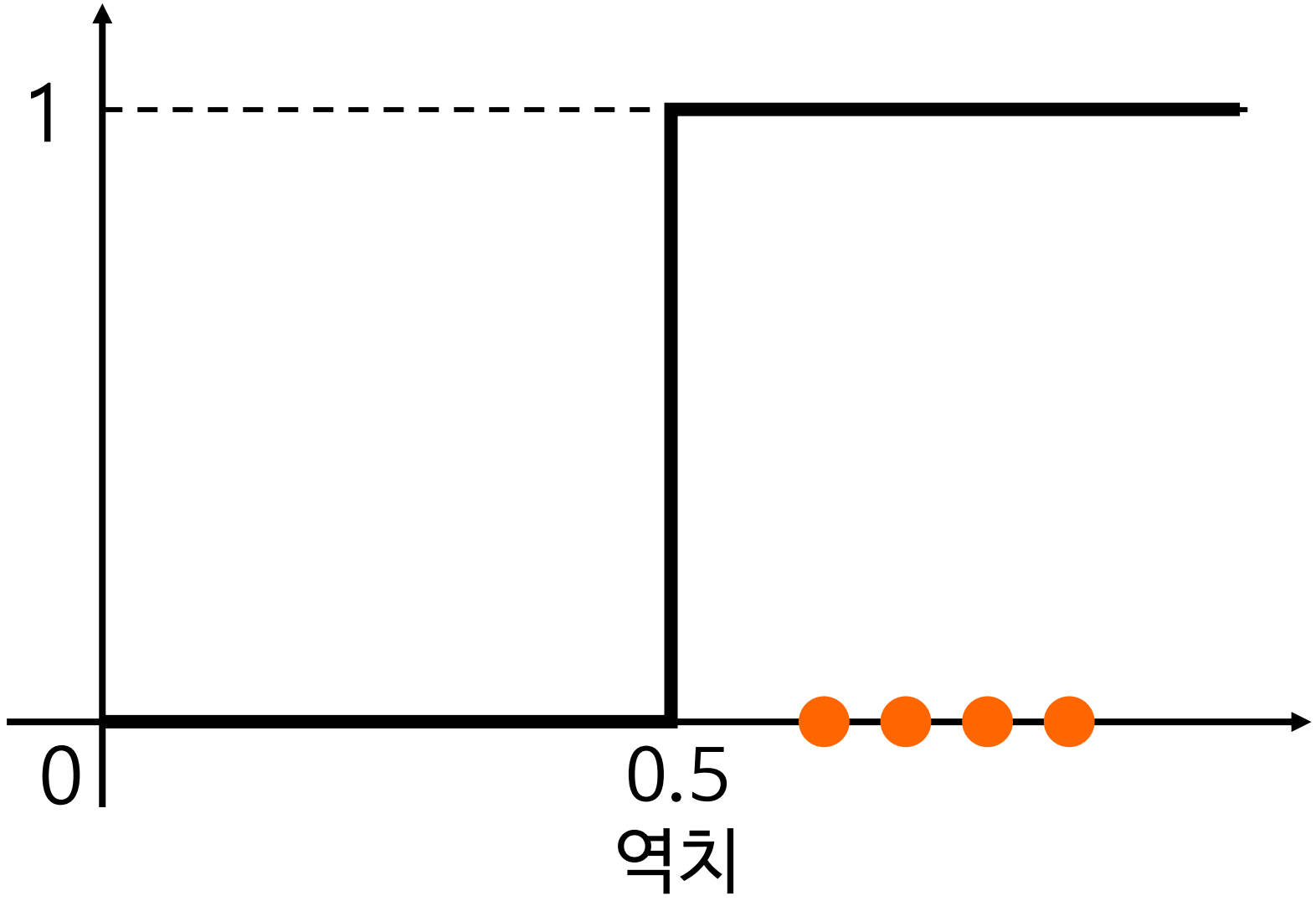
신박AI

계단함수는 함수의 값이 특정값 (역치)이하일 경우는 0을,



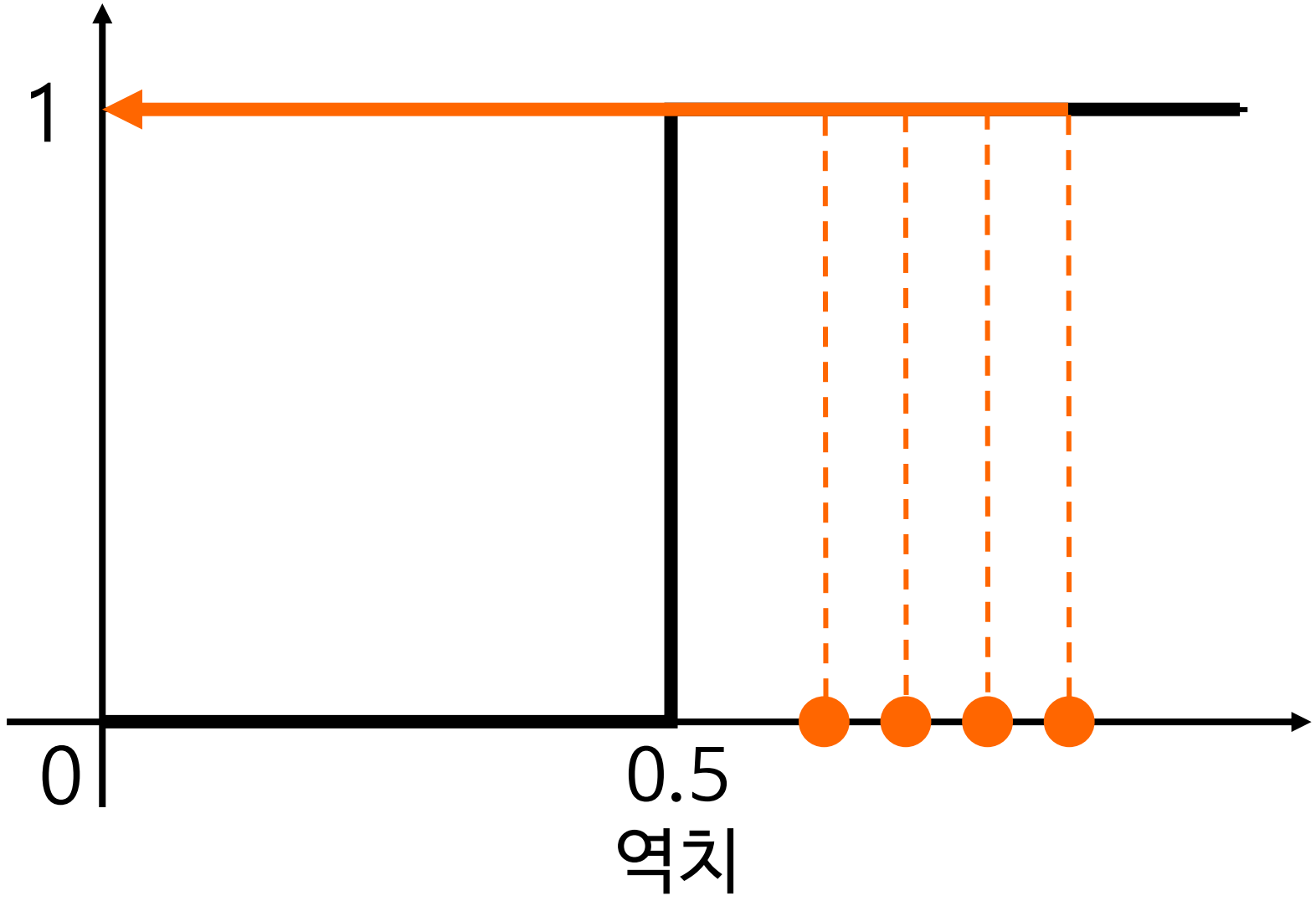
신박AI

계단함수는 함수의 값이 특정값 (역치)이상일 경우는



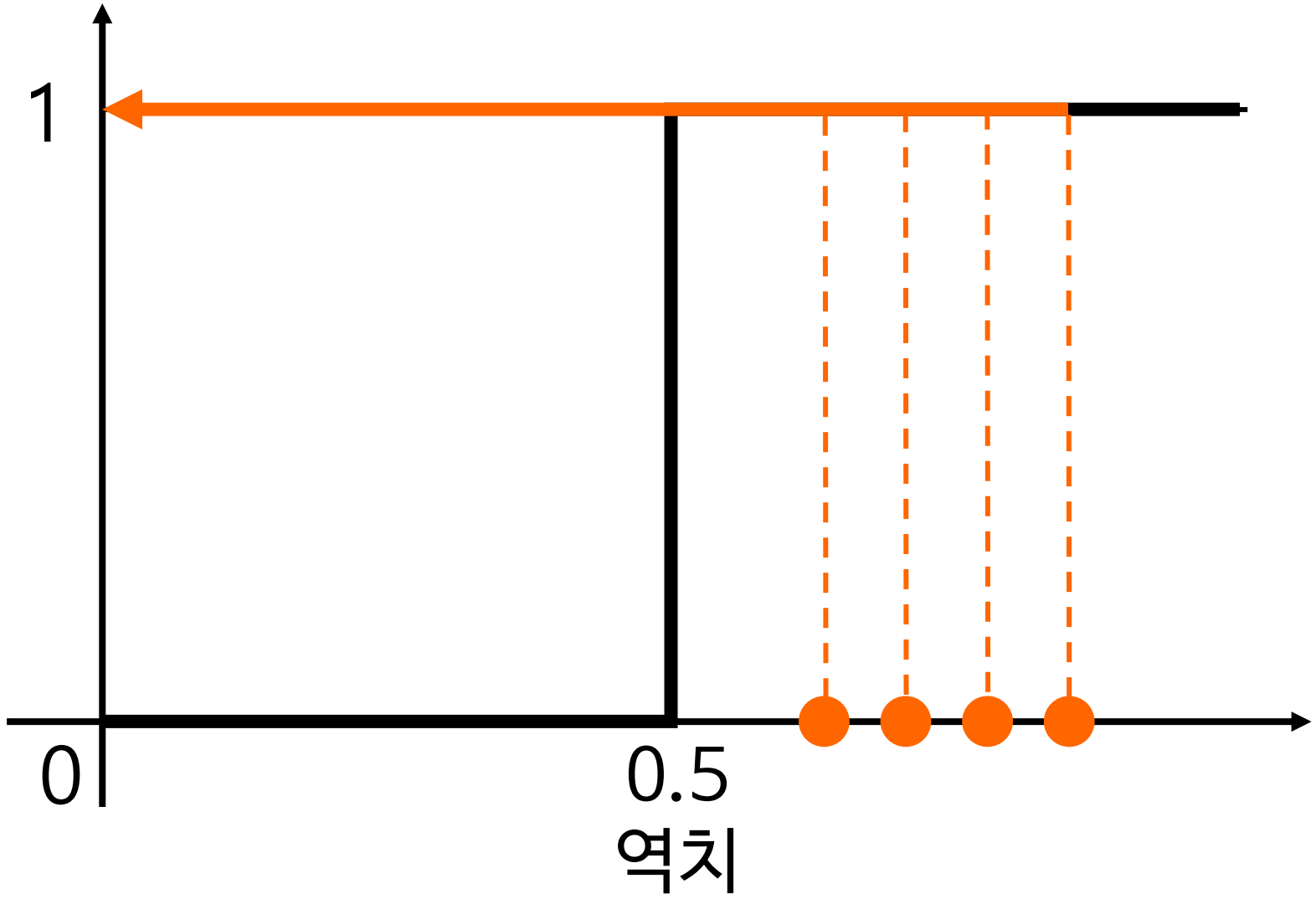
신박AI

계단함수는 함수의 값이 특정값 (역치)이상일 경우는 1을 출력하는



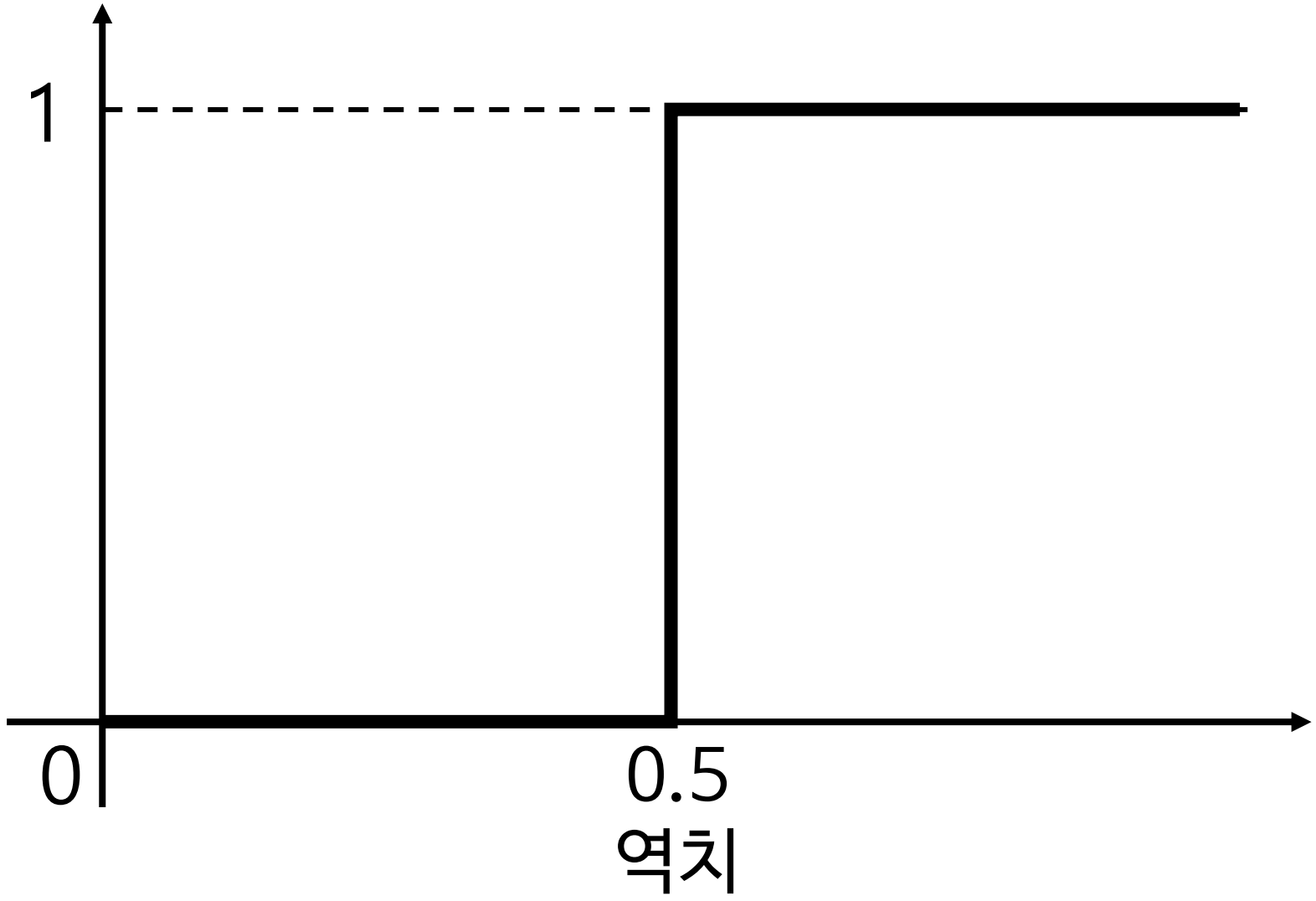
신박AI

계단함수는 함수의 값이 특정값 (역치)이상일 경우는 1을 출력하는 비선형 함수였습니다



신박AI

그런데 이 계단함수는 치명적인 약점이 있었습니다



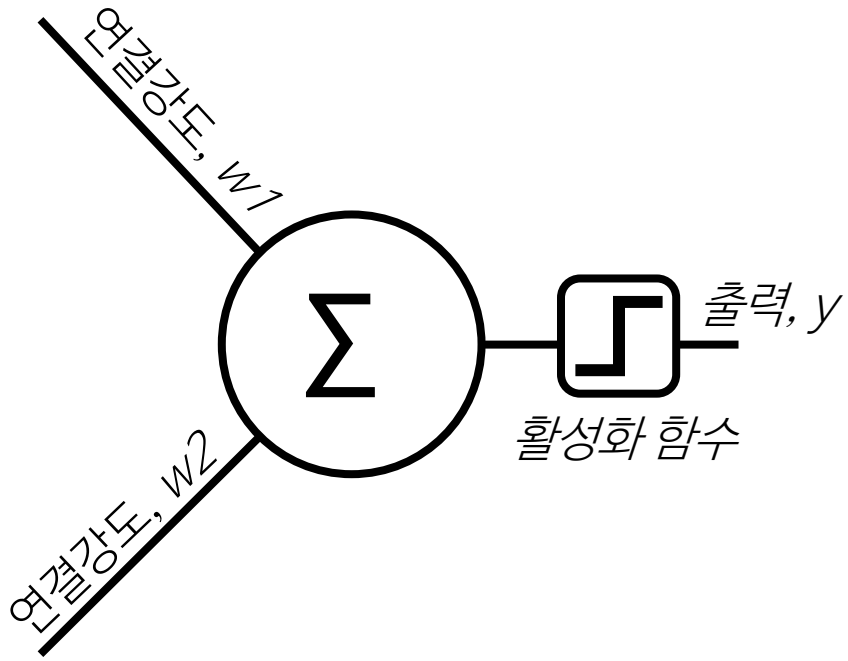
신박AI

이 약점을 설명 드리기 위해 지난 영상 슬라이드를 잠깐 빌려오겠습니다

신박AI

우리는 지난 영상에서 단층 퍼셉트론의 출력값을

x1	x2
0.8	0.6
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

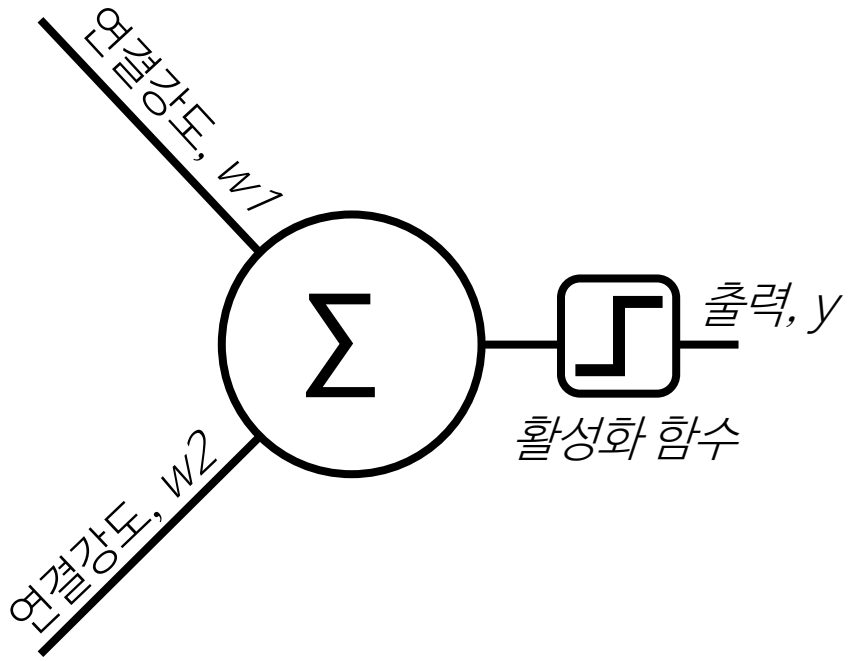


y
1
1
0
0
1
0
0

신박AI

어떻게 계산하는지 살펴 본 바가 있습니다

x1	x2
0.8	0.6
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

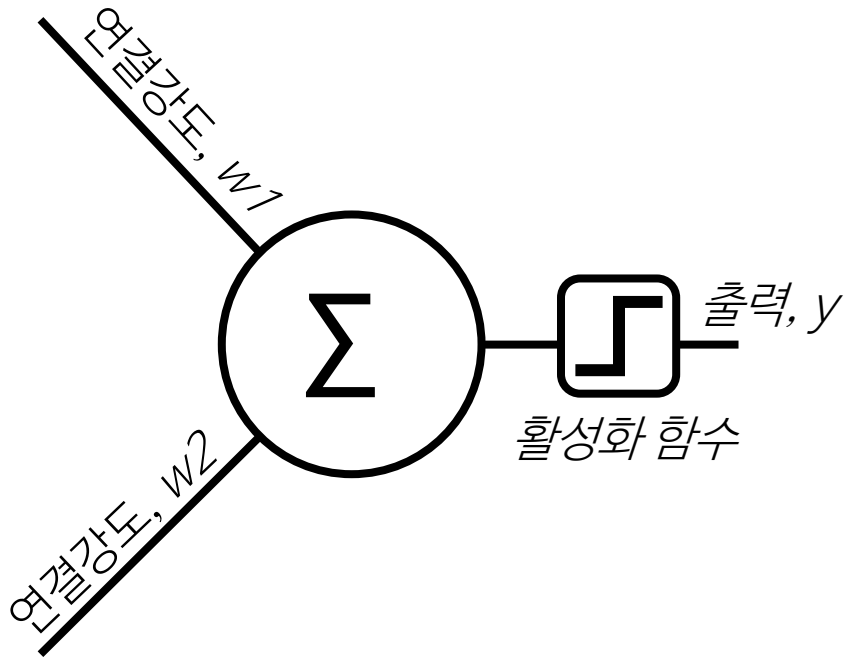


y
1
1
0
0
1
0
0

신박AI

혹시모를 기억의 망각을 대비하기 위해서

x1	x2
0.8	0.6
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

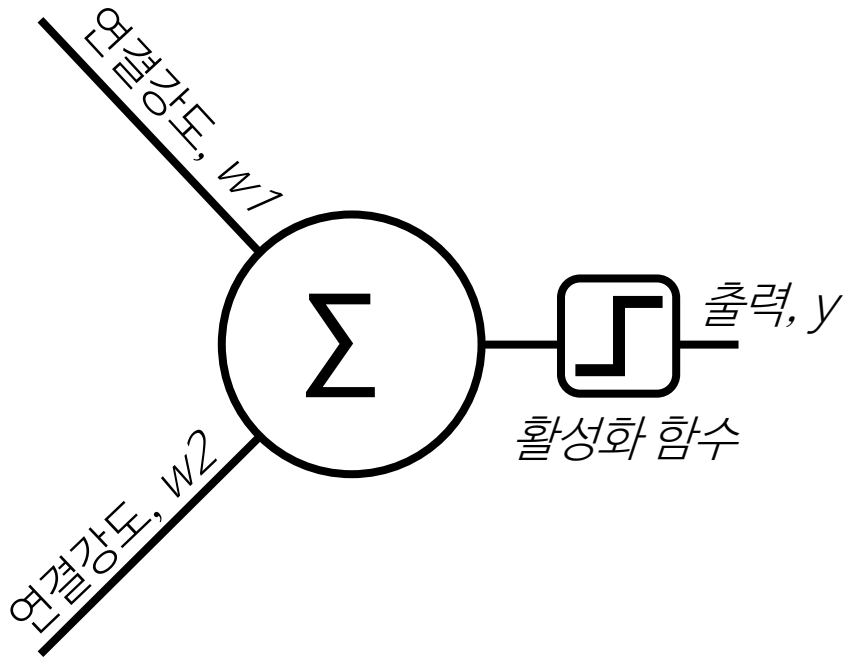


y
1
1
0
0
1
0
0

신박AI

입력값을 넣는 부분부터 다시 살펴보도록 하겠습니다

x1	x2
0.8	0.6
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

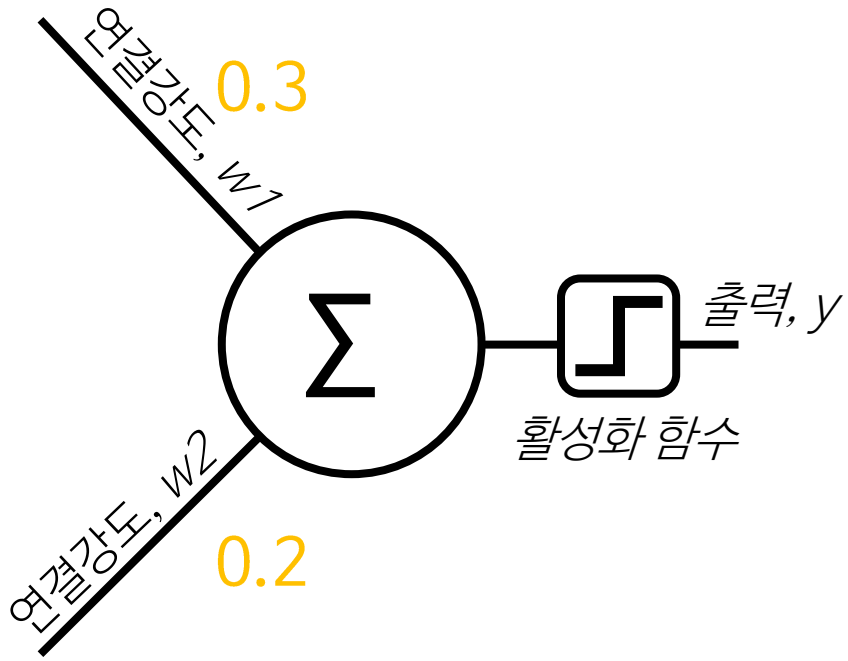


y
1
1
0
0
1
0
0

신박AI

첫번째 스텝은 우선 각 연결강도 값을 랜덤으로 정해줘야 합니다

x1	x2
0.8	0.6
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

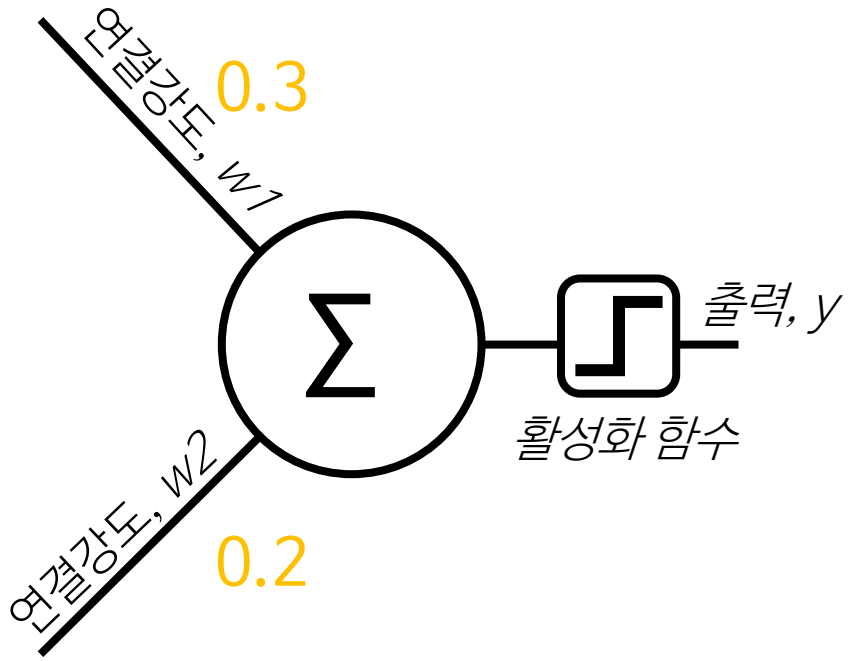


y
1
1
0
0
1
0
0

신박AI

자 이제 이 데이터를 퍼셉트론에 넣어봅시다

x1	x2
0.8	0.6
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

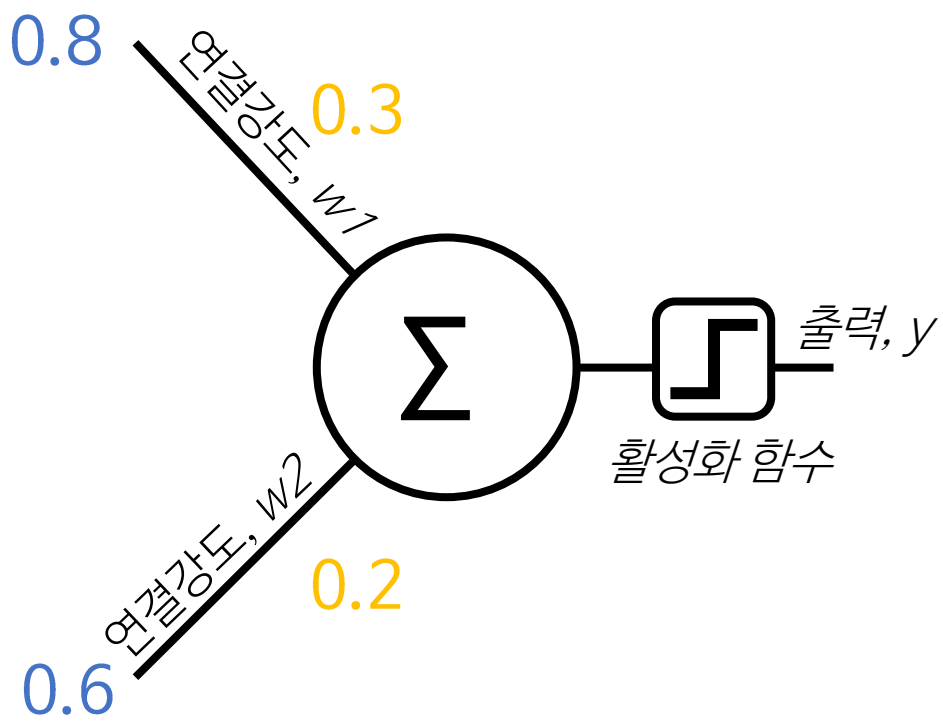


y
1
1
0
0
1
0
0

신박AI

자 이제 이 데이터를 퍼셉트론에 넣어봅시다

x1	x2
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

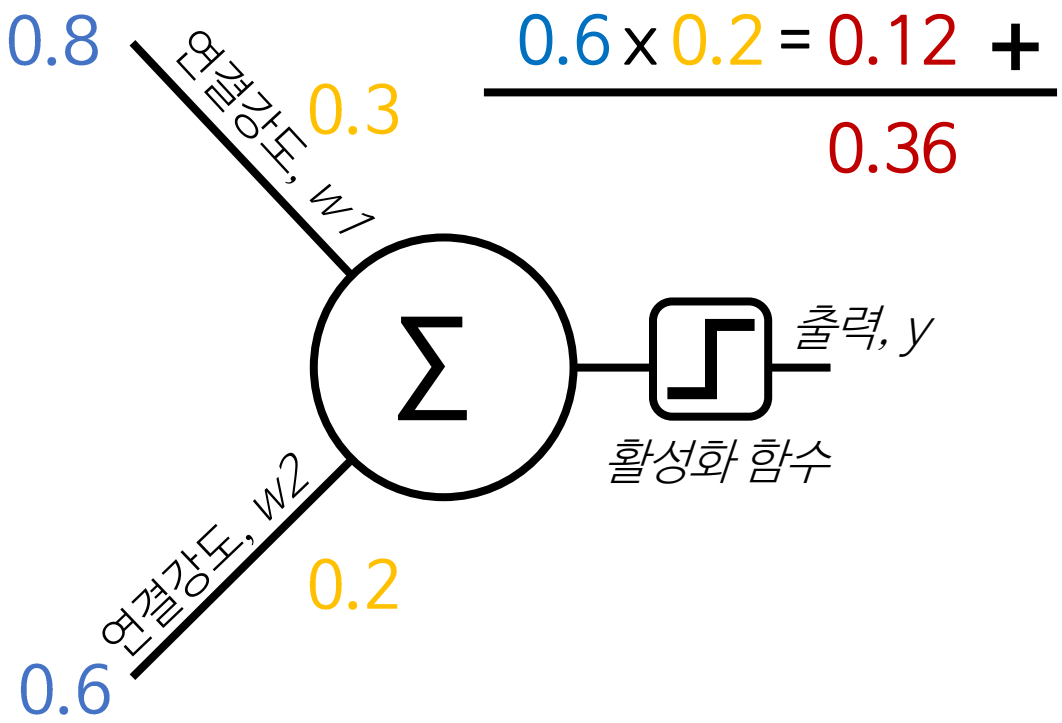


y
1
1
0
0
1
0
0

신박AI

각 연결강도와 입력값의 곱들의 합을 구하고..

x1	x2
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

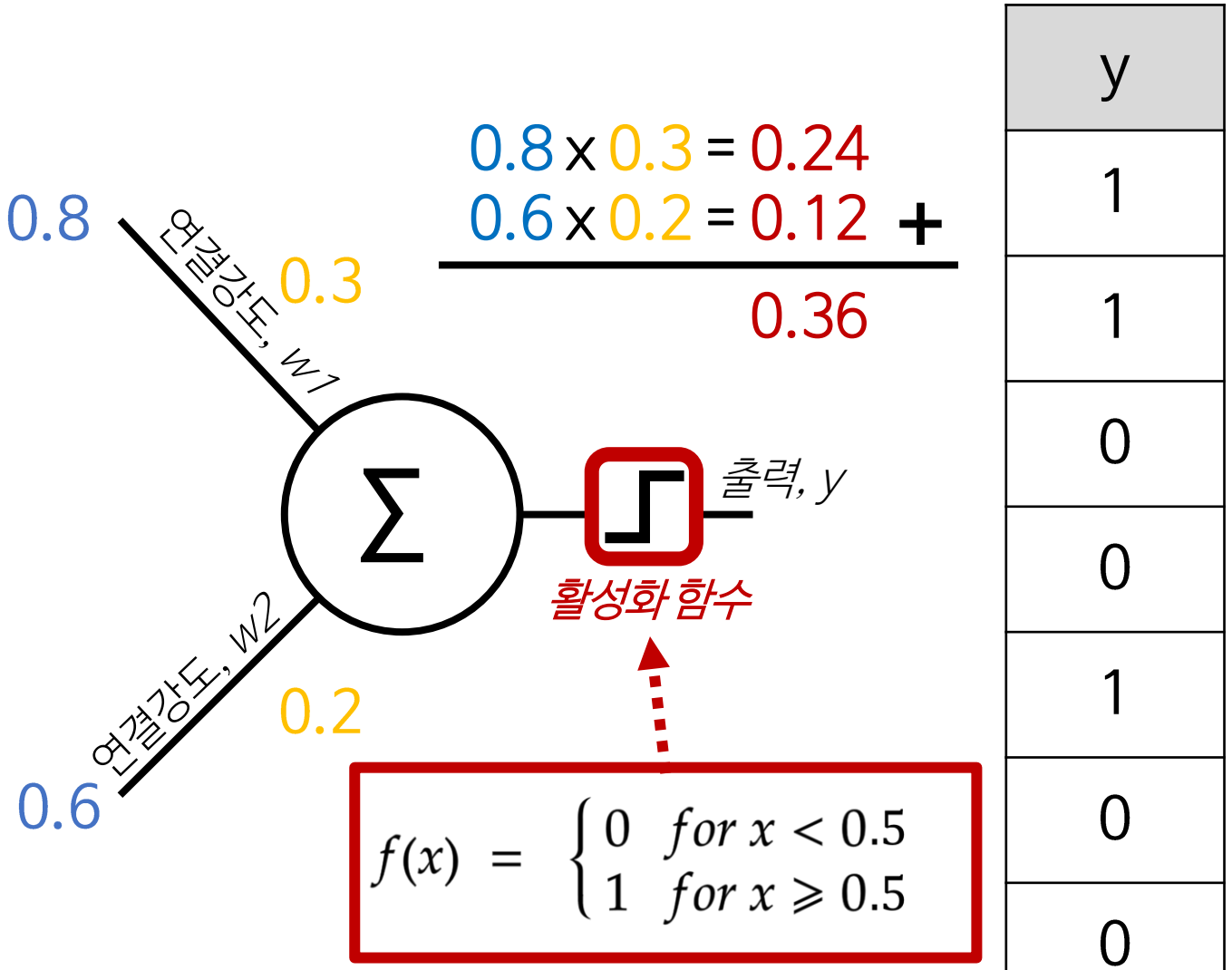


y
1
1
0
0
1
0
0

신박AI

그 합을 계단함수로 정의된 활성화 함수에 넣으면..

x1	x2
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

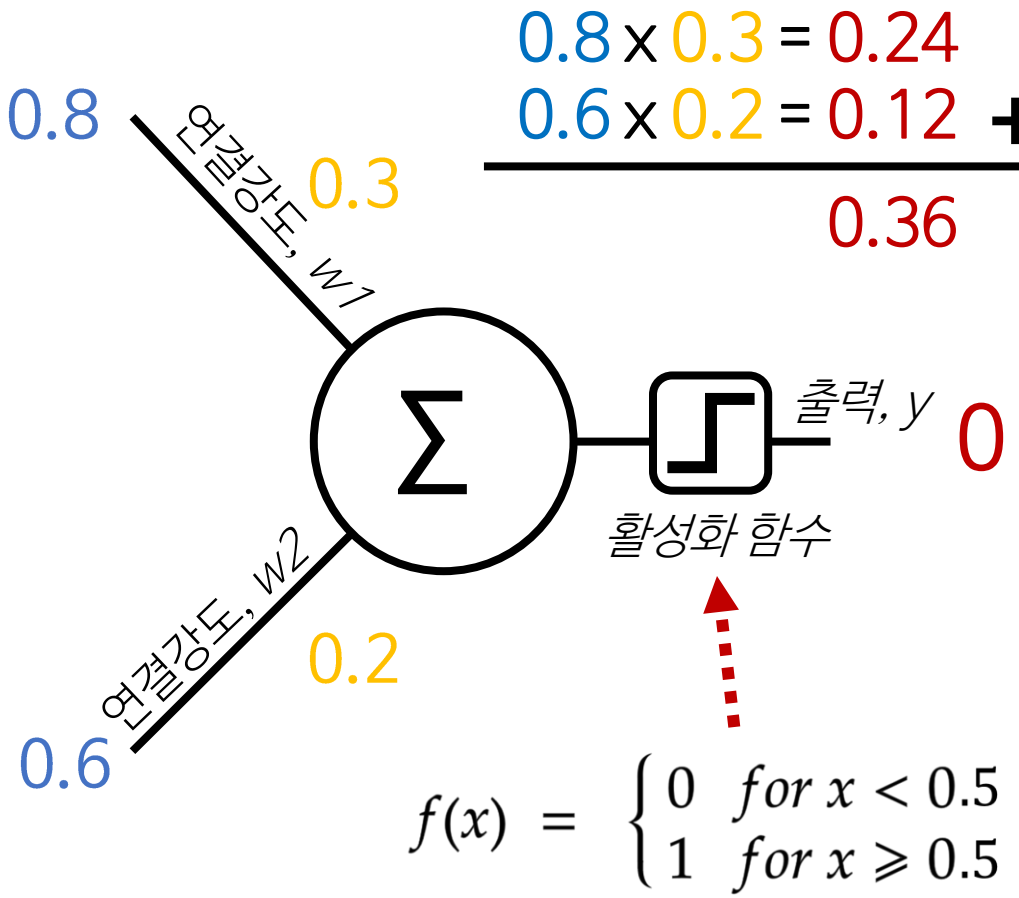


y
1
1
0
0
1
0
0

신박AI

퍼셉트론의 출력이 0이 되는 것을 지난 영상에서 본 바가 있습니다

x1	x2
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

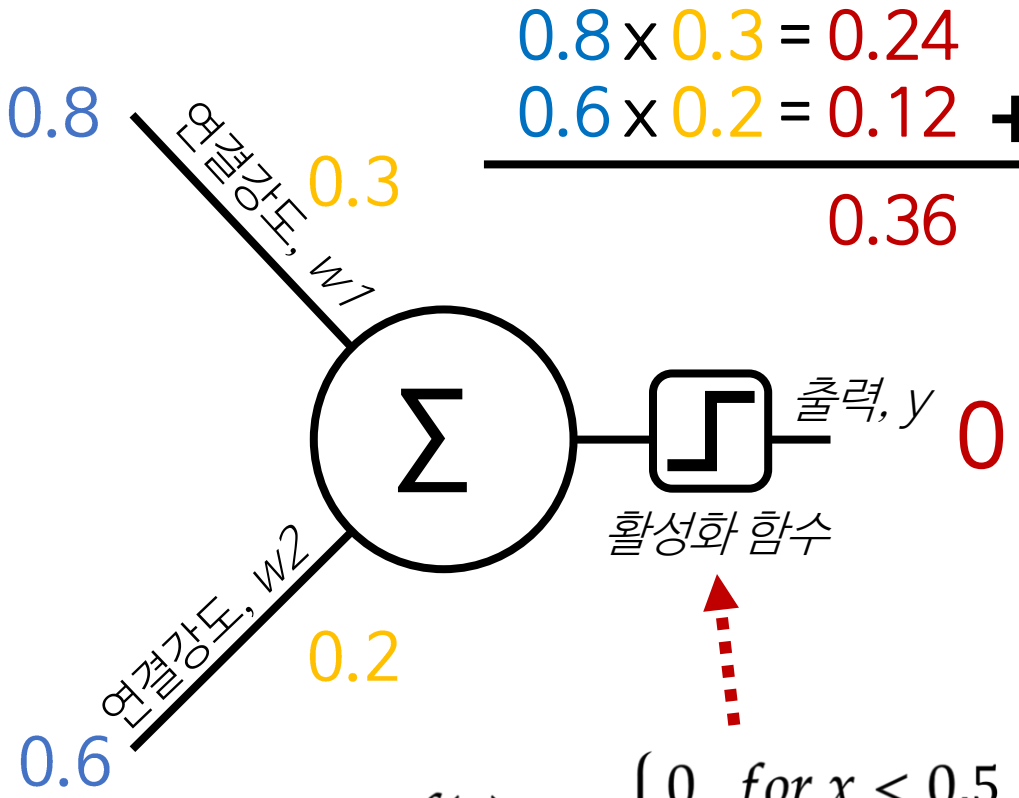


y	y'
1	0
1	
0	
0	
1	
0	
0	

신박AI

그런데 말입니다

x1	x2
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

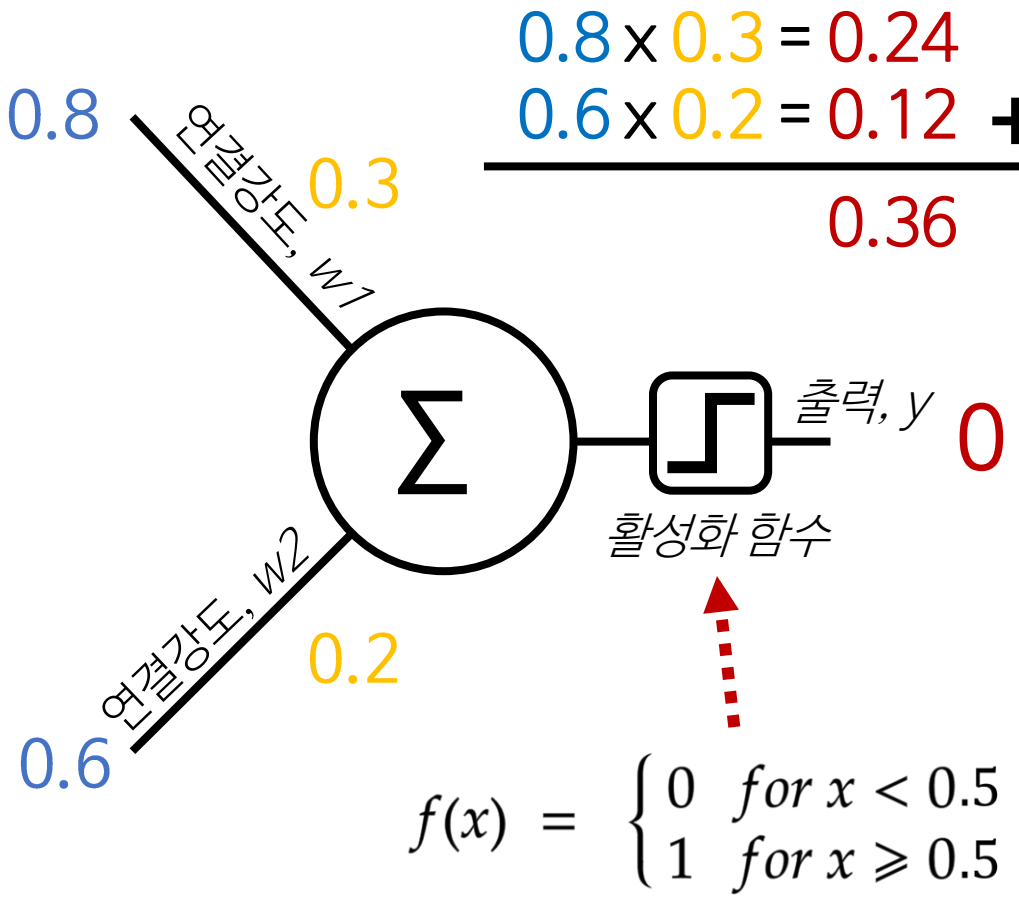


y	y'
1	0
1	
0	
0	
1	
0	
0	

신박AI

단층 퍼셉트론의 학습알고리즘에 의해서..

x1	x2
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

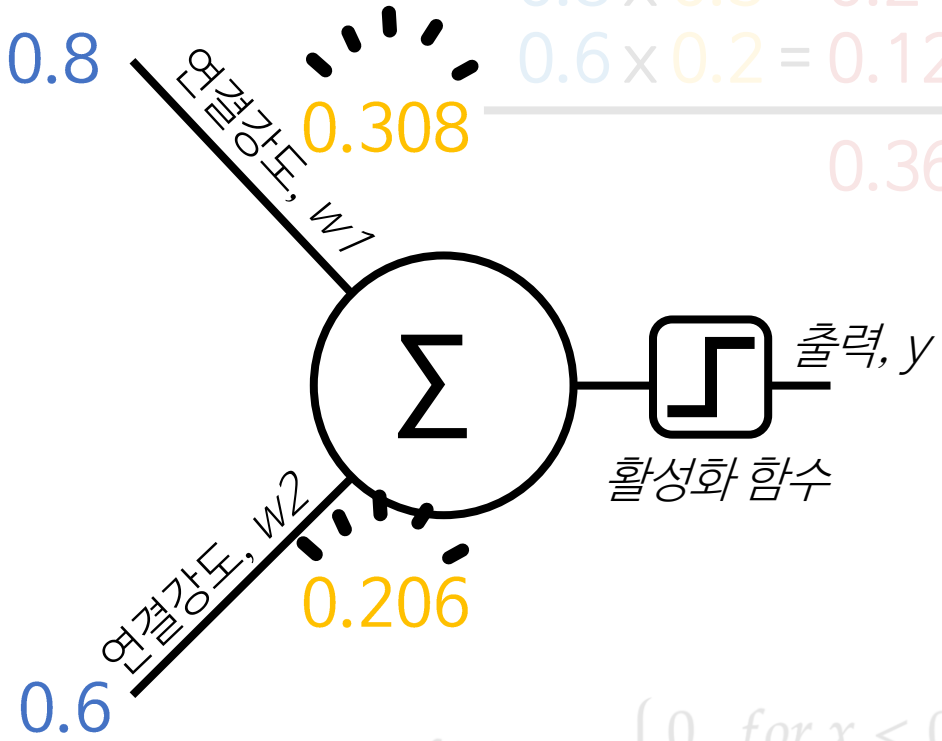


y	y'
1	0
1	
0	
0	
1	
0	
0	

신박AI

이렇게 새로운 연결강도로 업데이트가 된다 하더라도

x1	x2
0.6	0.9
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2



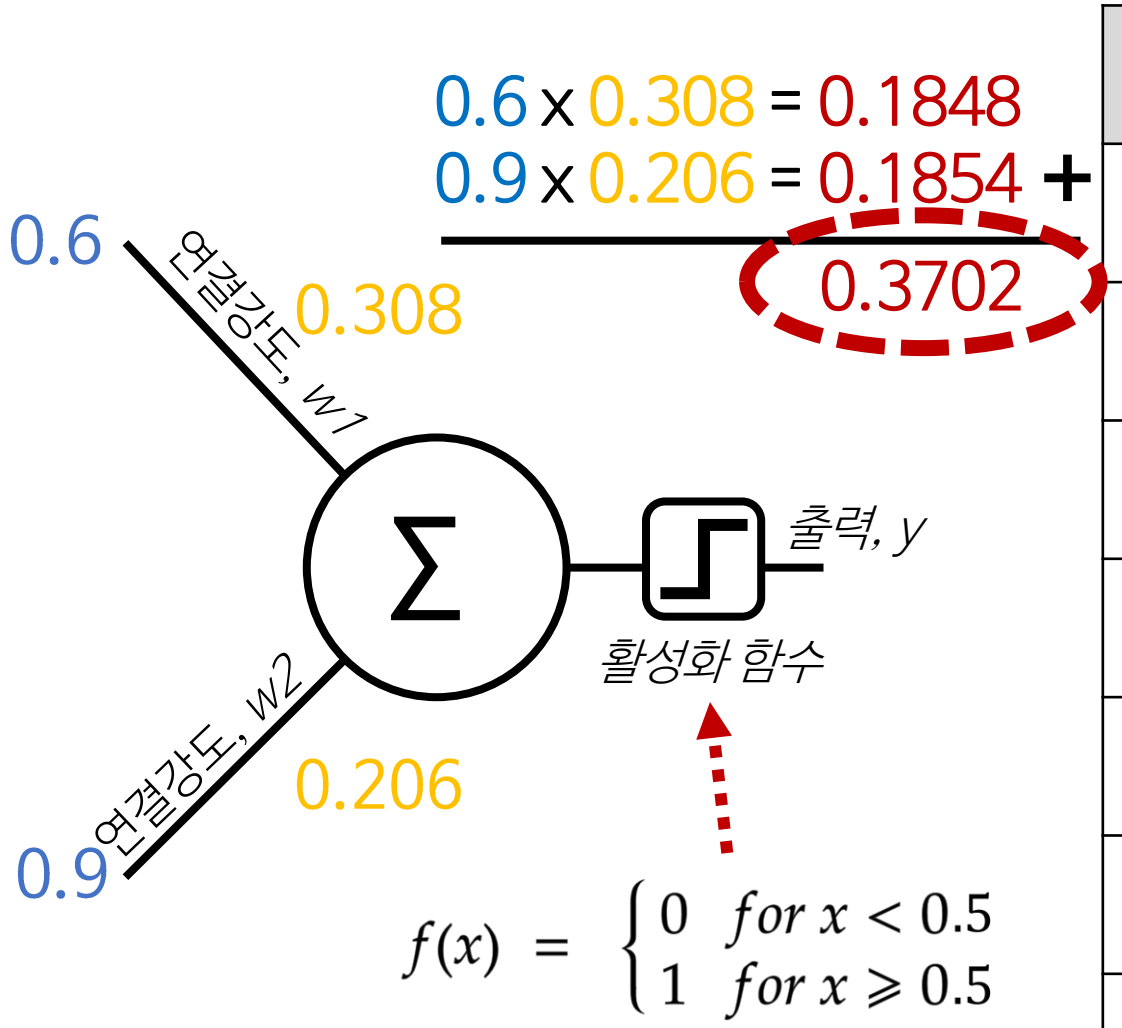
$$0.8 \times 0.3 = 0.24$$
$$0.6 \times 0.2 = 0.12 \; +$$
$$0.36$$

$$f(x) = \begin{cases} 0 & for\ x < 0.5 \\ 1 & for\ x \geqslant 0.5 \end{cases}$$

y	y'
1	0
1	
0	
0	
1	
0	
0	

그래서 계산값이 좀더 커진다 하더라도

x1	x2
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

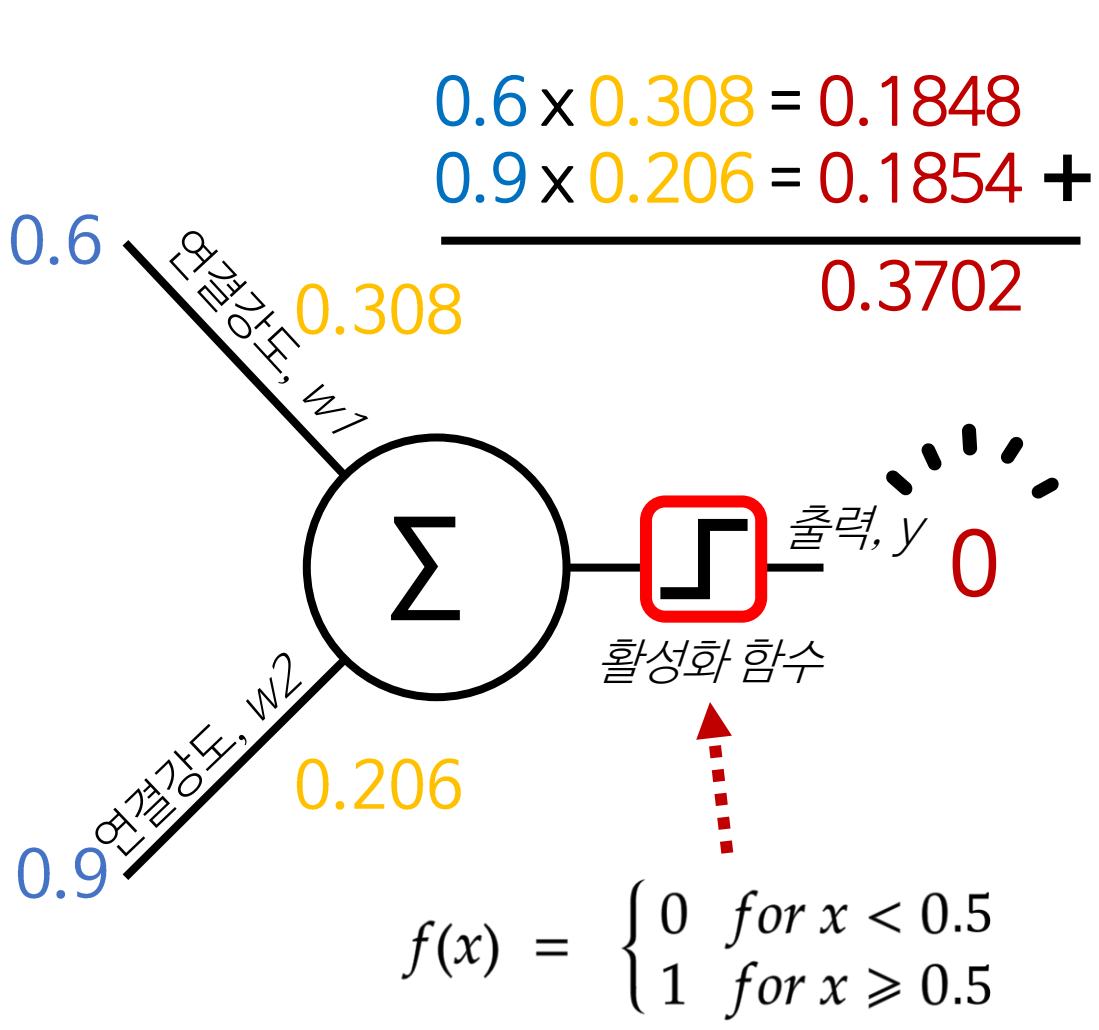


y	y'
1	0
1	
0	
0	
1	
0	
0	

신박AI

다시 활성화 함수에 의해 출력값은 0이 되는 것을 볼수가 있습니다

x1	x2
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

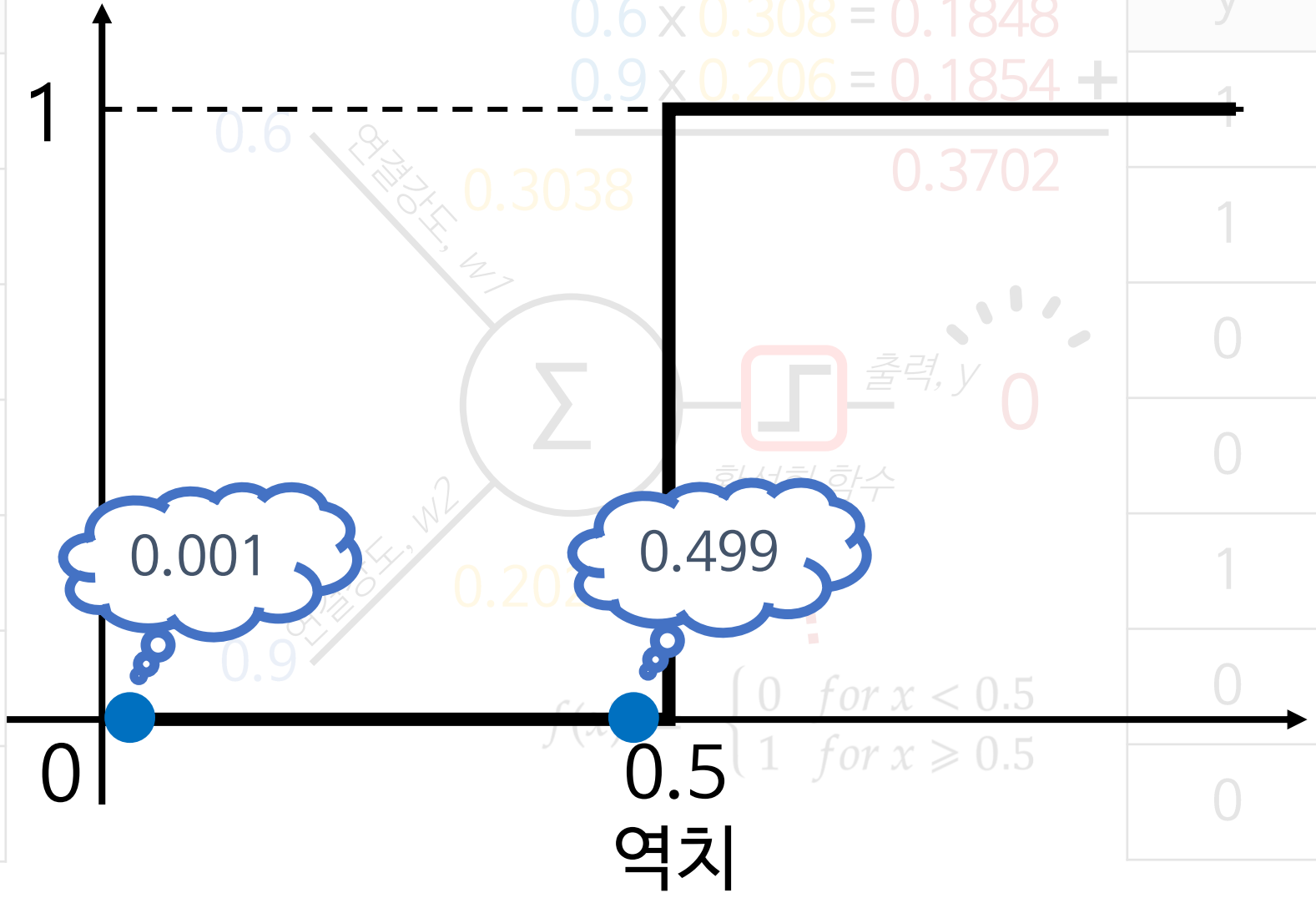


y	y'
1	0
1	0
0	
0	
1	
0	
0	

신박AI

극단적인 경우 퍼셉트론의 계산값이 0.001이든, 0.499든 상관없이

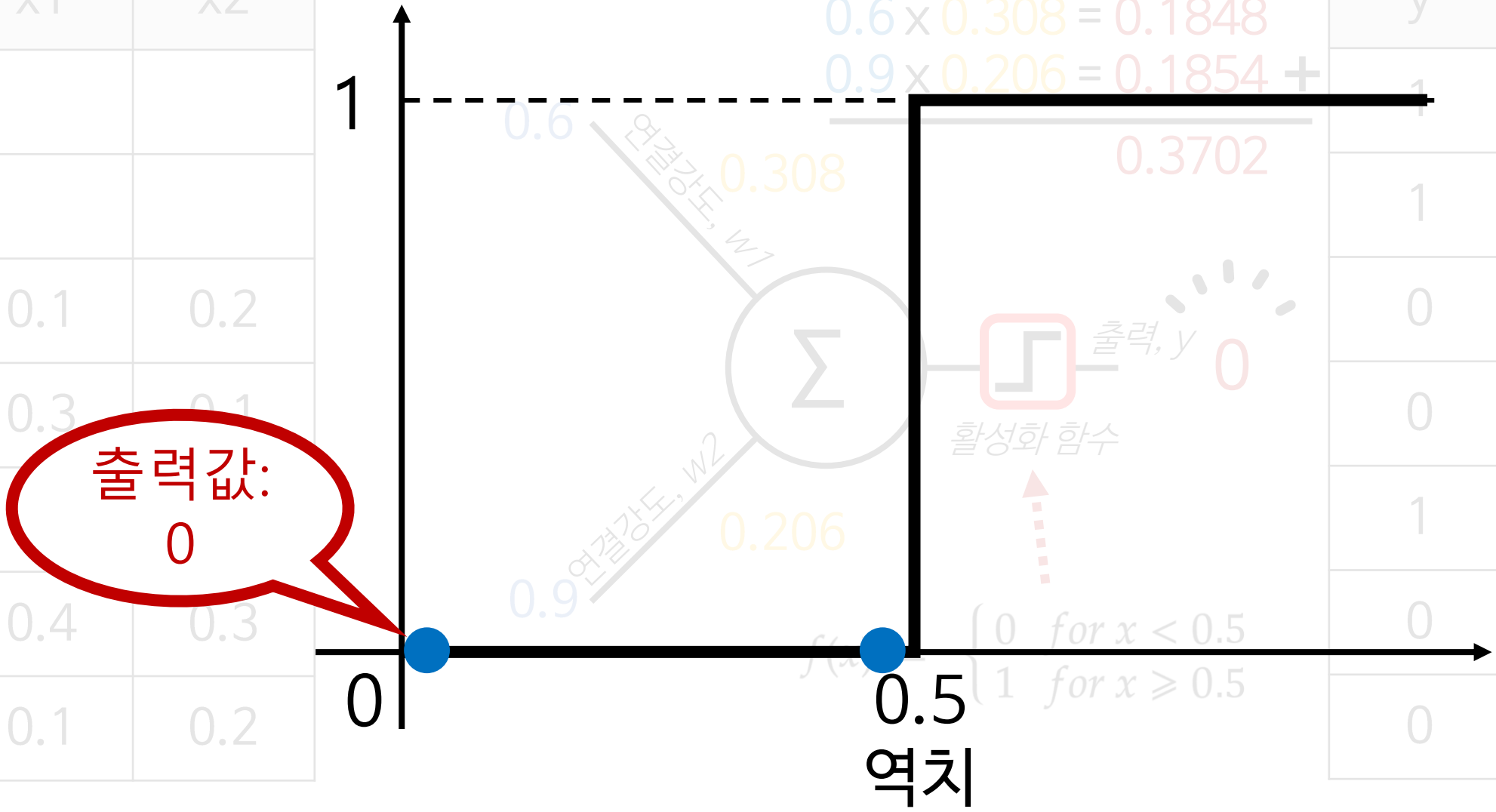
x1	x2
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2



y
1
1
0
0
1
0
0

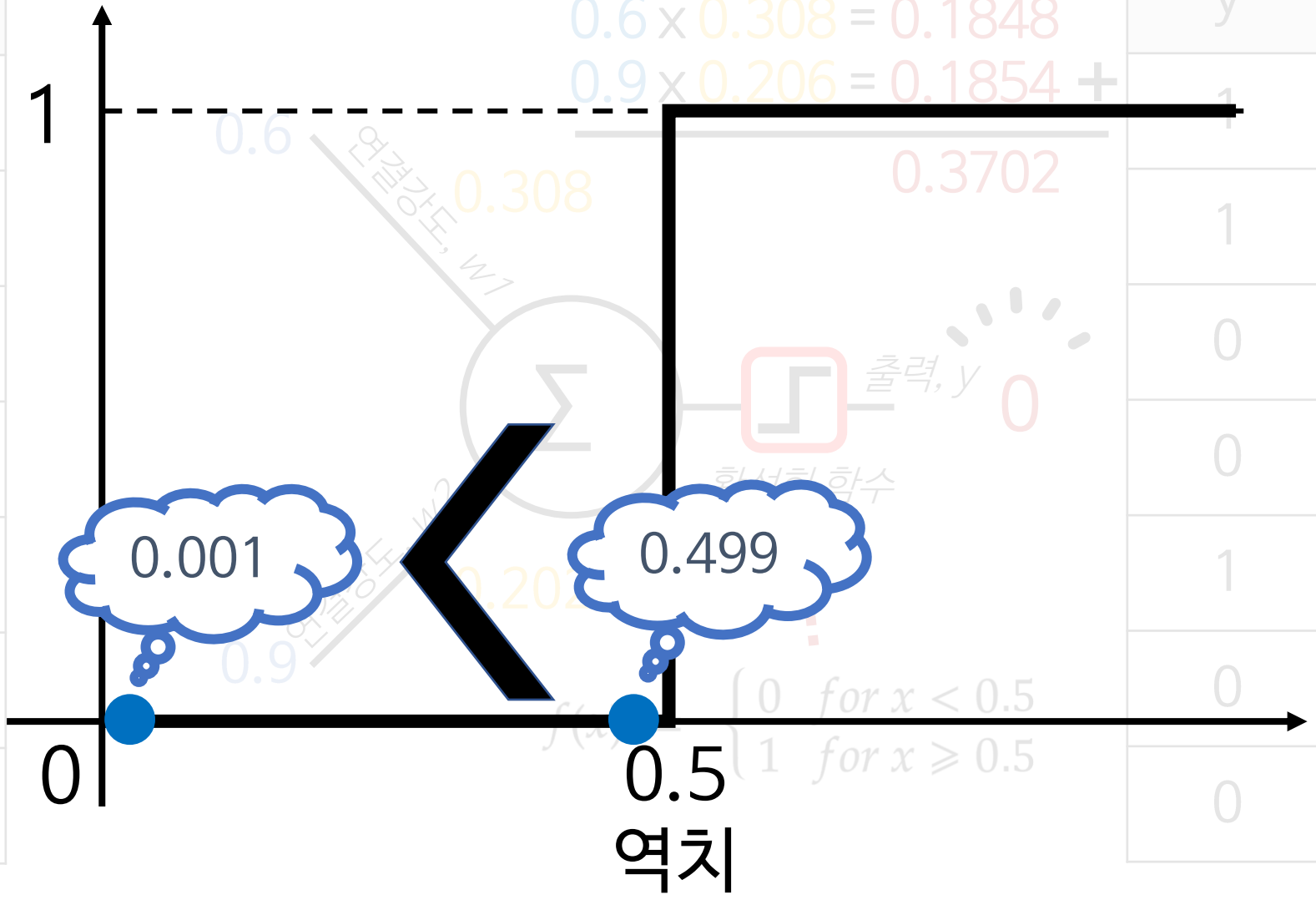
y'
0
0

출력값은 0이 됩니다.



분명 0.499는 0.001보다 더 잘 예측하긴 한건데,

x1	x2
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

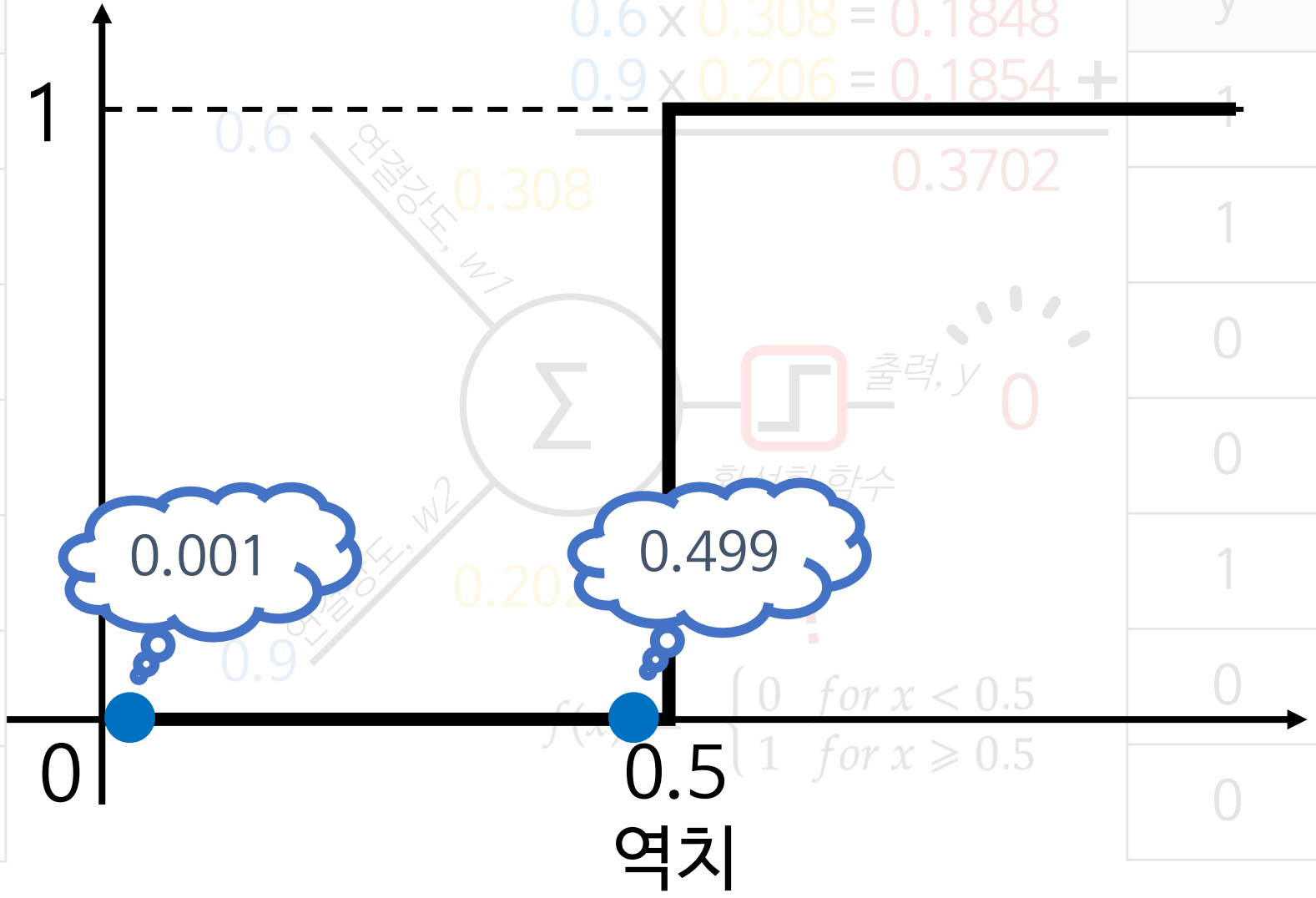


y
1
1
0
0
1
0
0

y'
0
0

그런 내부적인 차이는 무시한채 그저 똑같이 오차가 1이 되어버렸습니다

x1	x2
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2

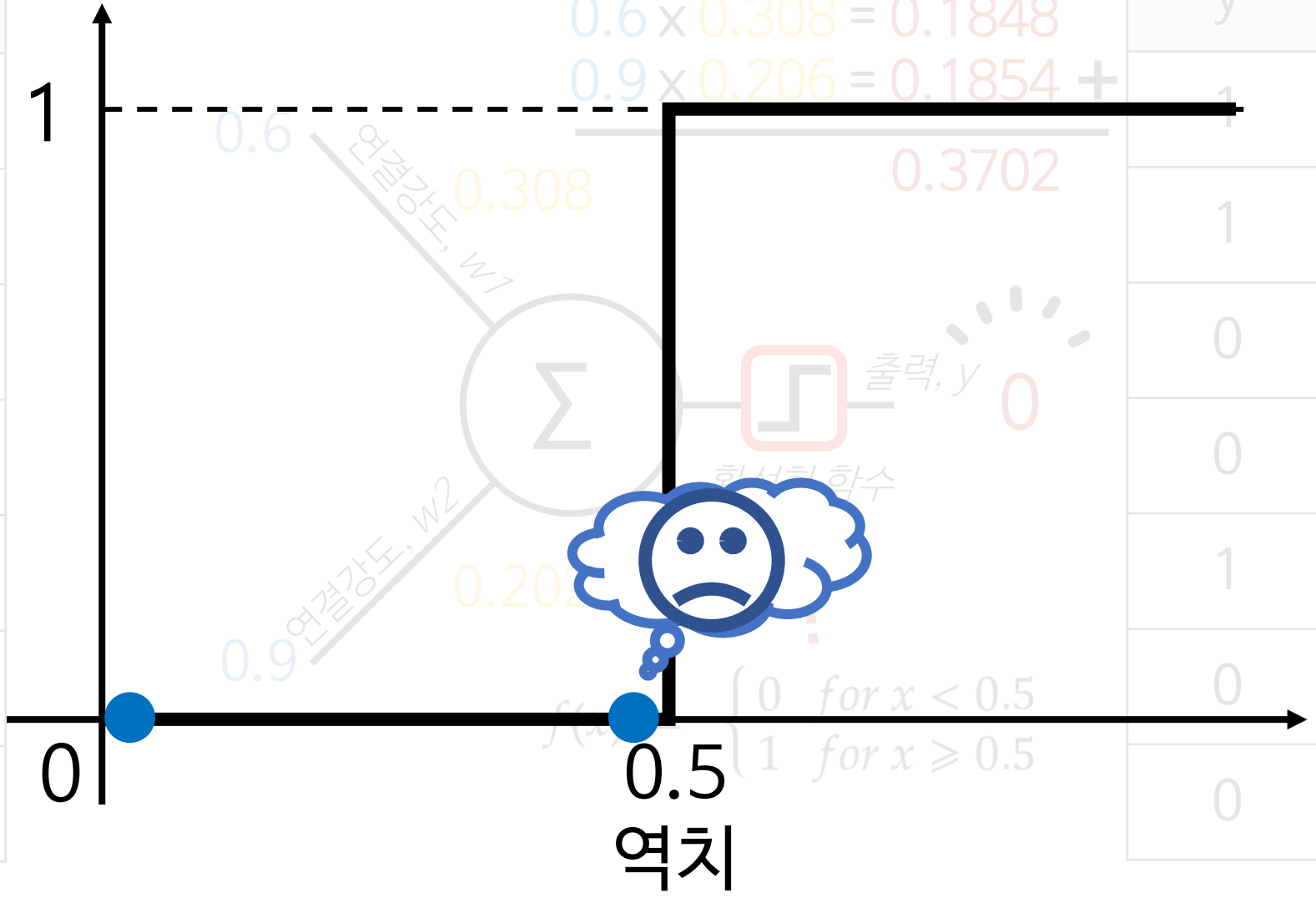


y
1
1
0
0
1
0
0

y'
0
0

0.499입장에서는 환장할 노릇입니다

x1	x2
0.1	0.2
0.3	0.1
0.6	0.6
0.4	0.3
0.1	0.2



y
1
1
0
0
1
0
0

y'
0
0

그것은 마치 어떤 학교에 무자비한 교수가 있는데,

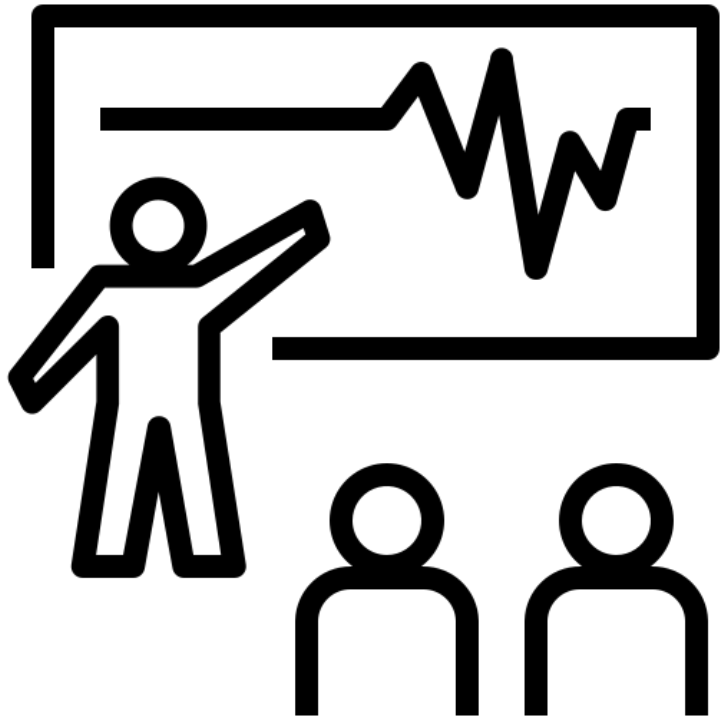


image: Flaticon.com

공부 하나도 안하고 0점을 받은 학생이나

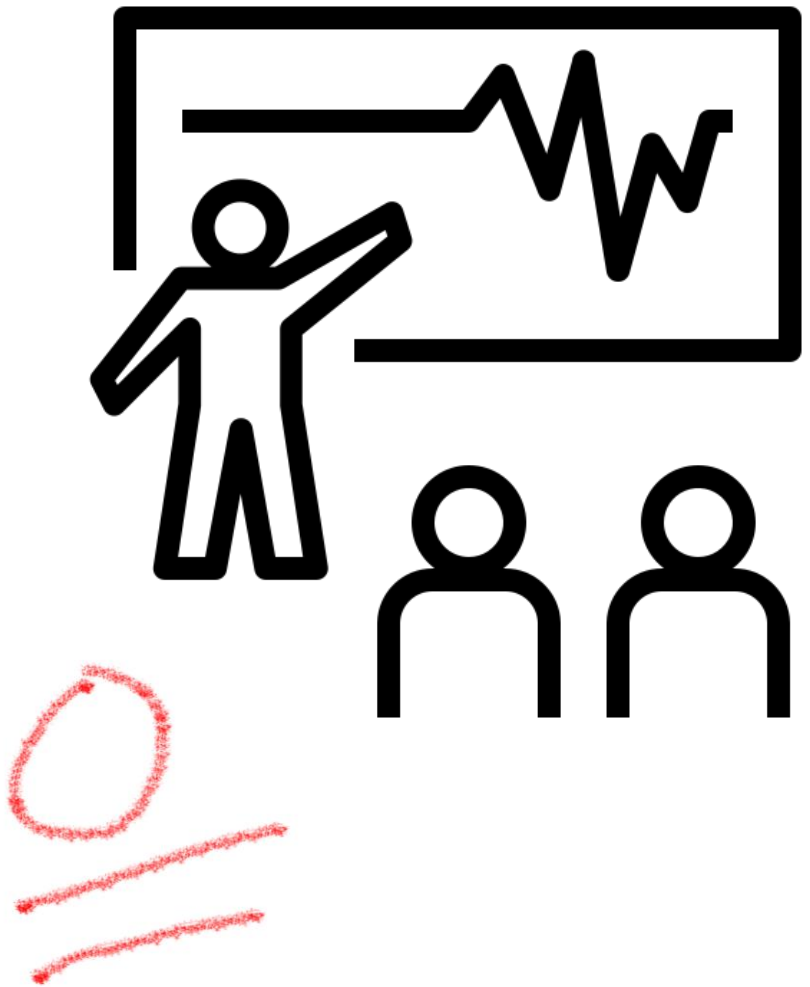


image: Flaticon.com

그래도 열심히 했으나 실수를 하는 바람에 59점을 받은 학생을

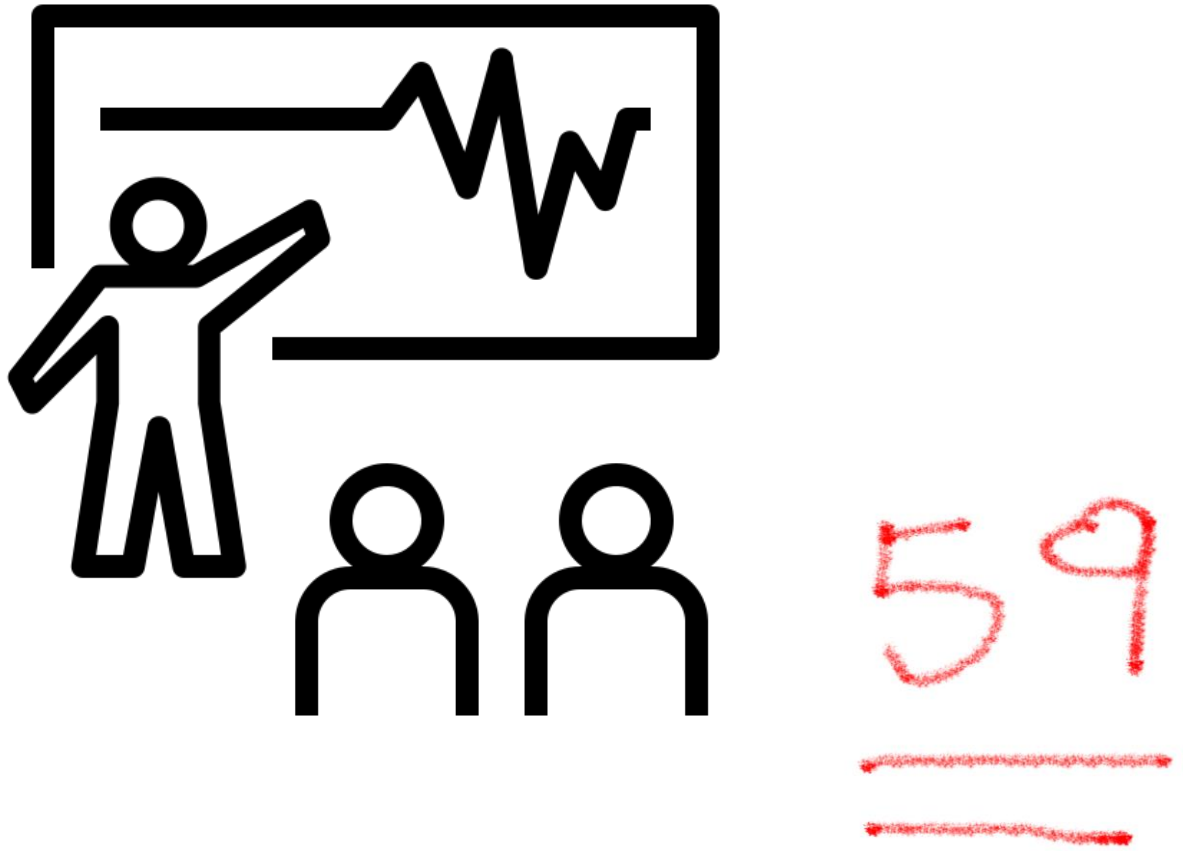


image: Flaticon.com

무조건 다 F를 주는 그런 무자비한 처사라고나 할까요?

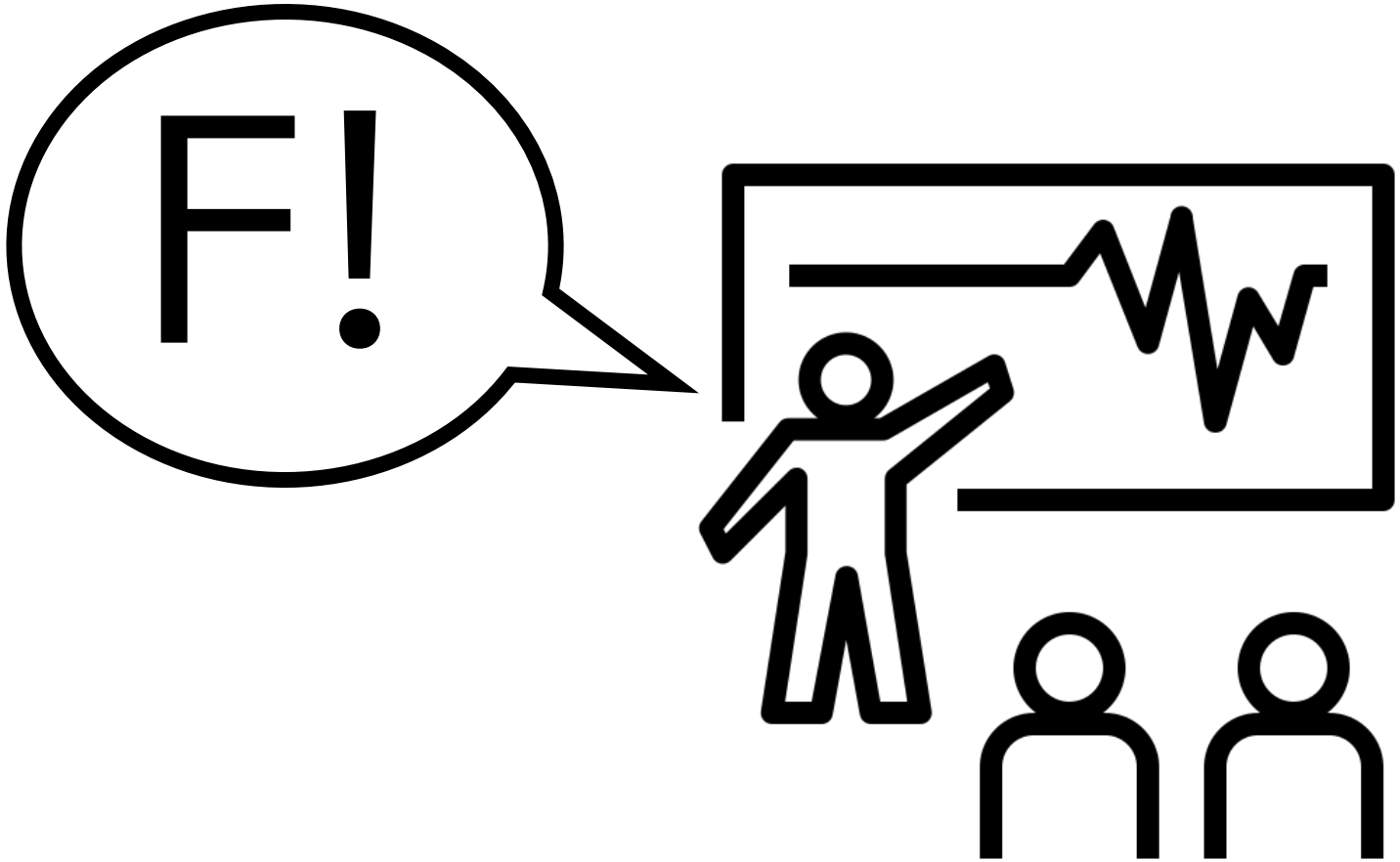


image: Flaticon.com

이래서는 학습할 맛이 나지 않겠죠?

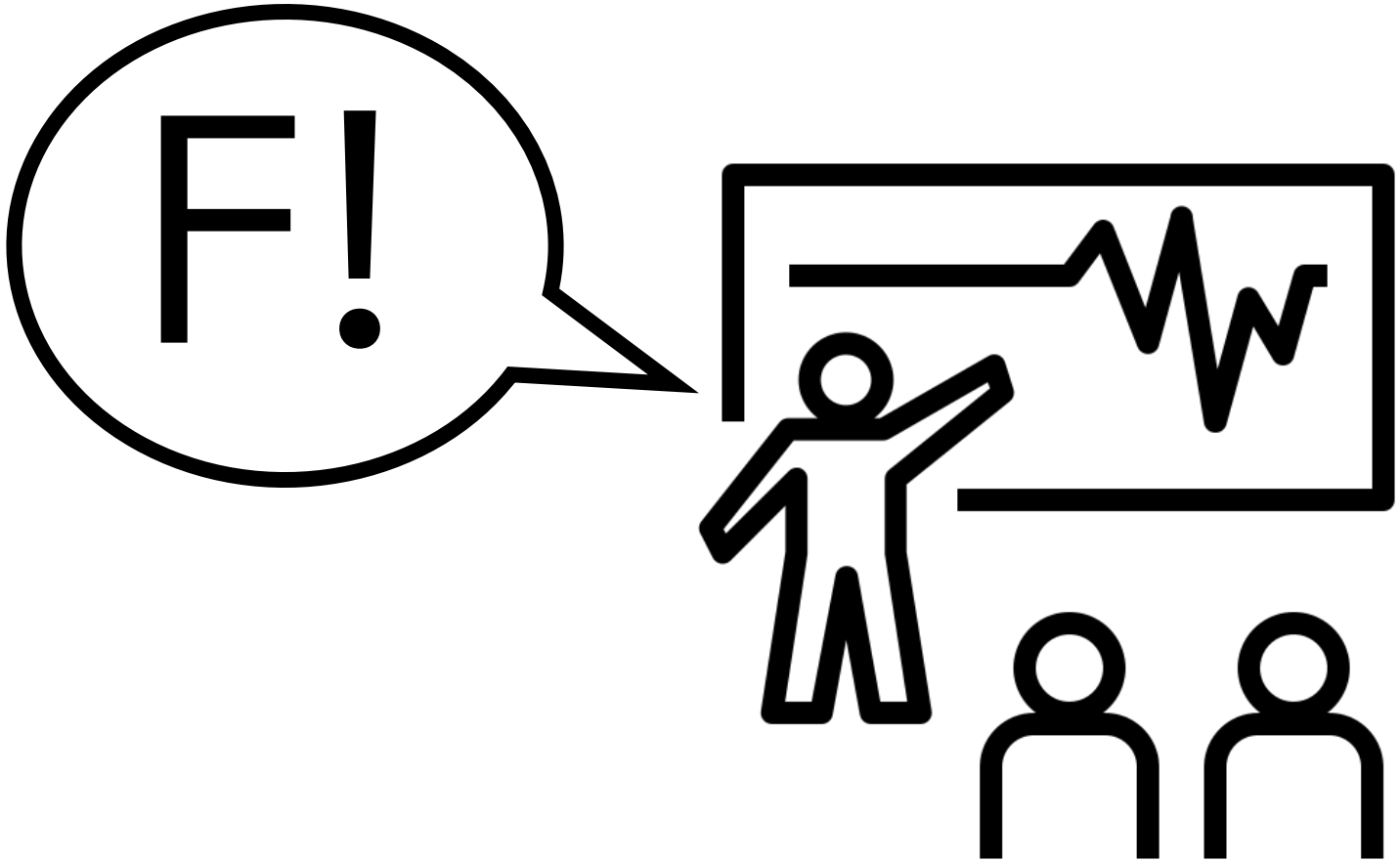
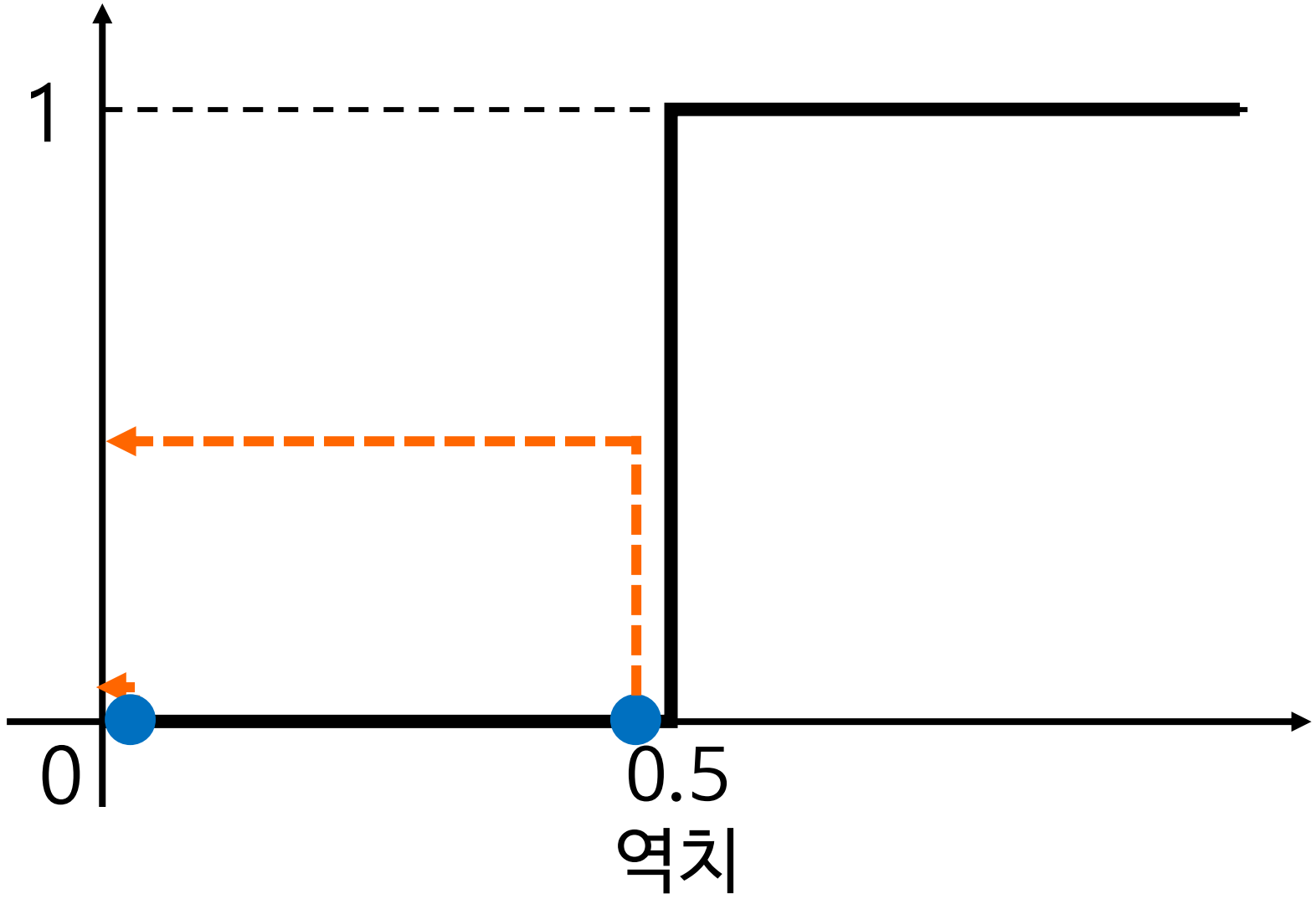


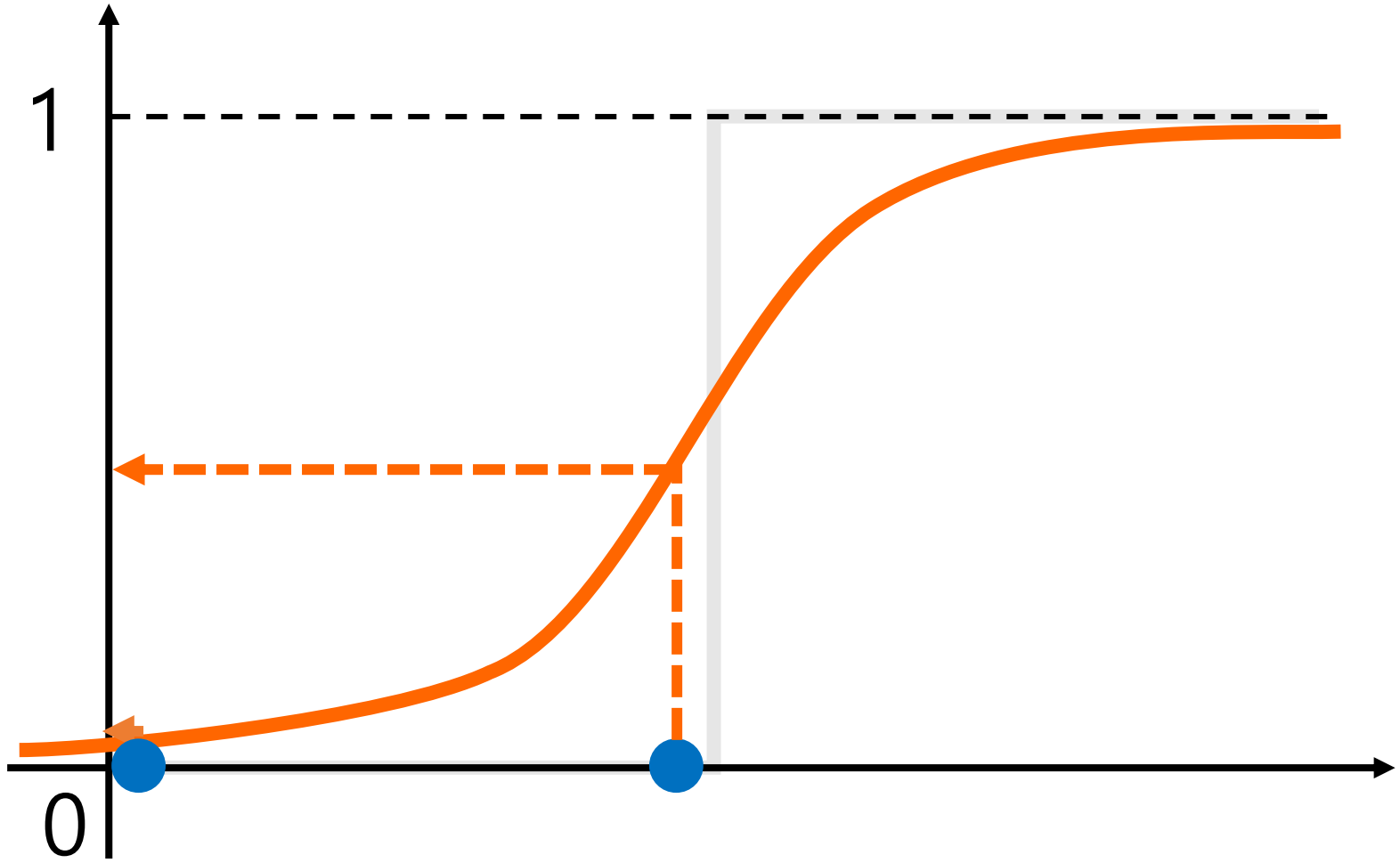
image: Flaticon.com

그렇다면 학습의 효율성을 위해 내부값의 차이를 오차에 반영할 수는 없을까요?



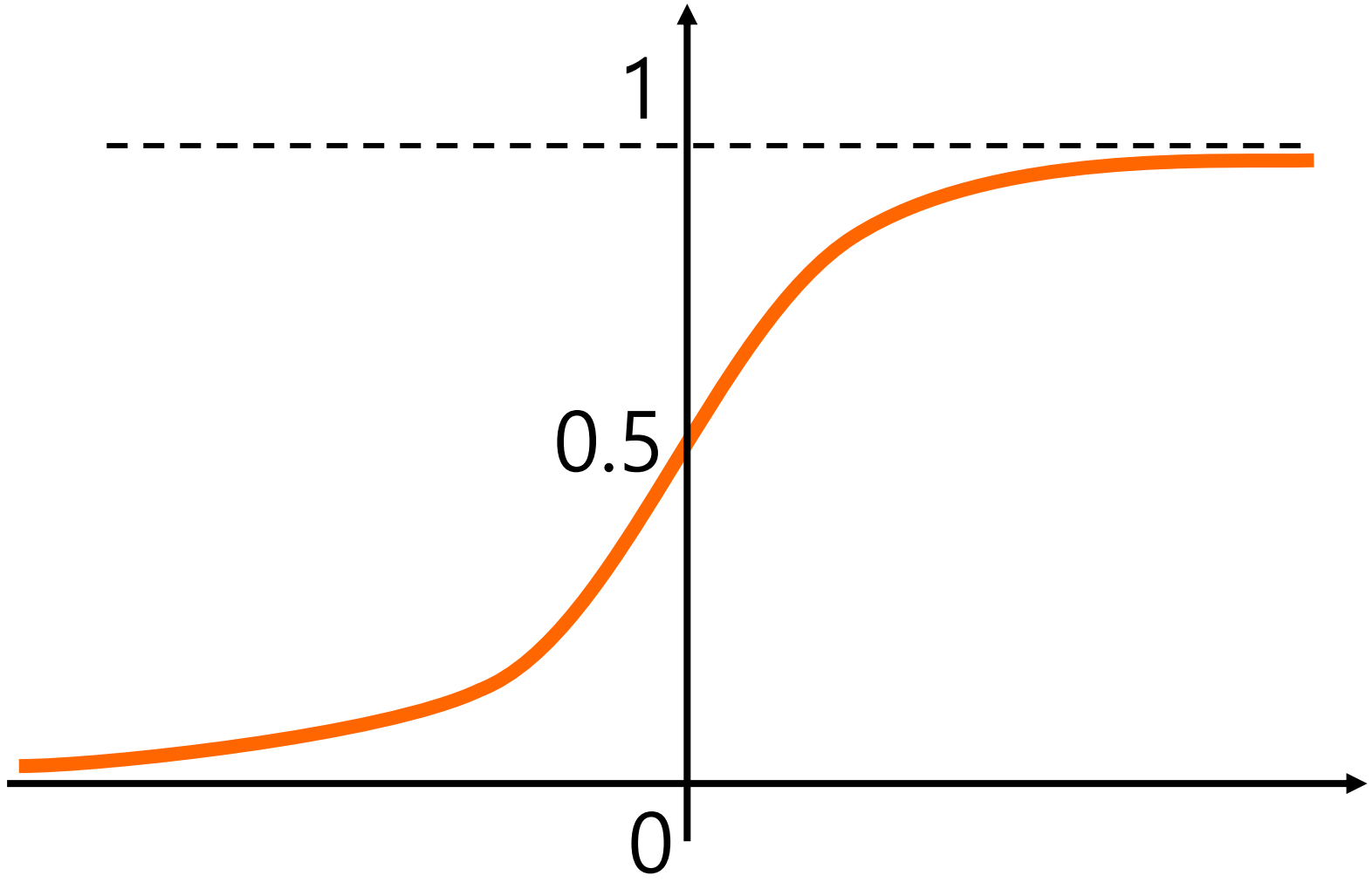
신박AI

그래서 다음과 같은 '형태'의 활성화 함수가 필요한 것입니다



신박AI

이런 형태의 활성화 함수 중 대표적인 것이 시그모이드 함수입니다

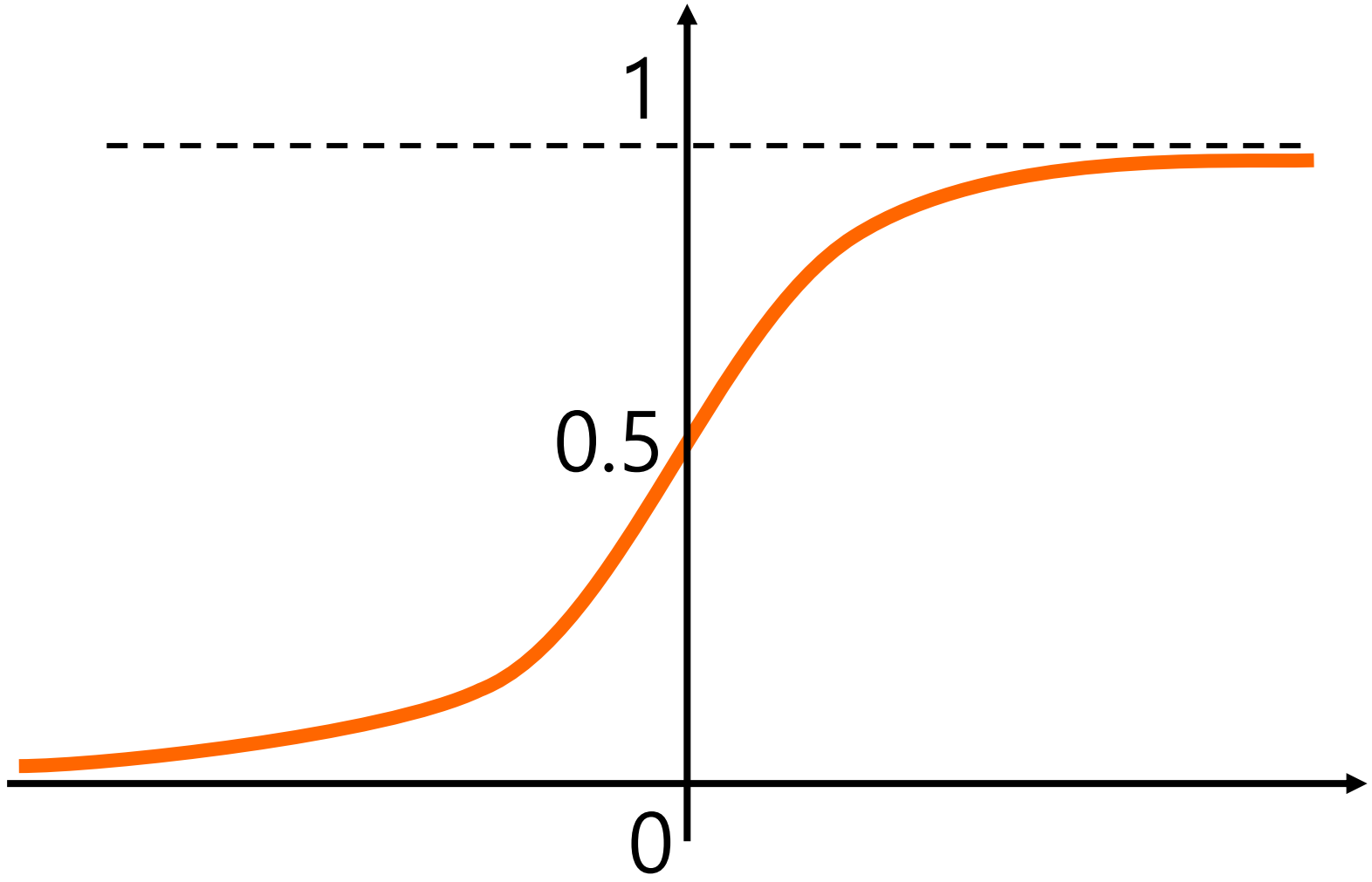


신박AI

Chapter 2
시그모이드 (sigmoid)함수

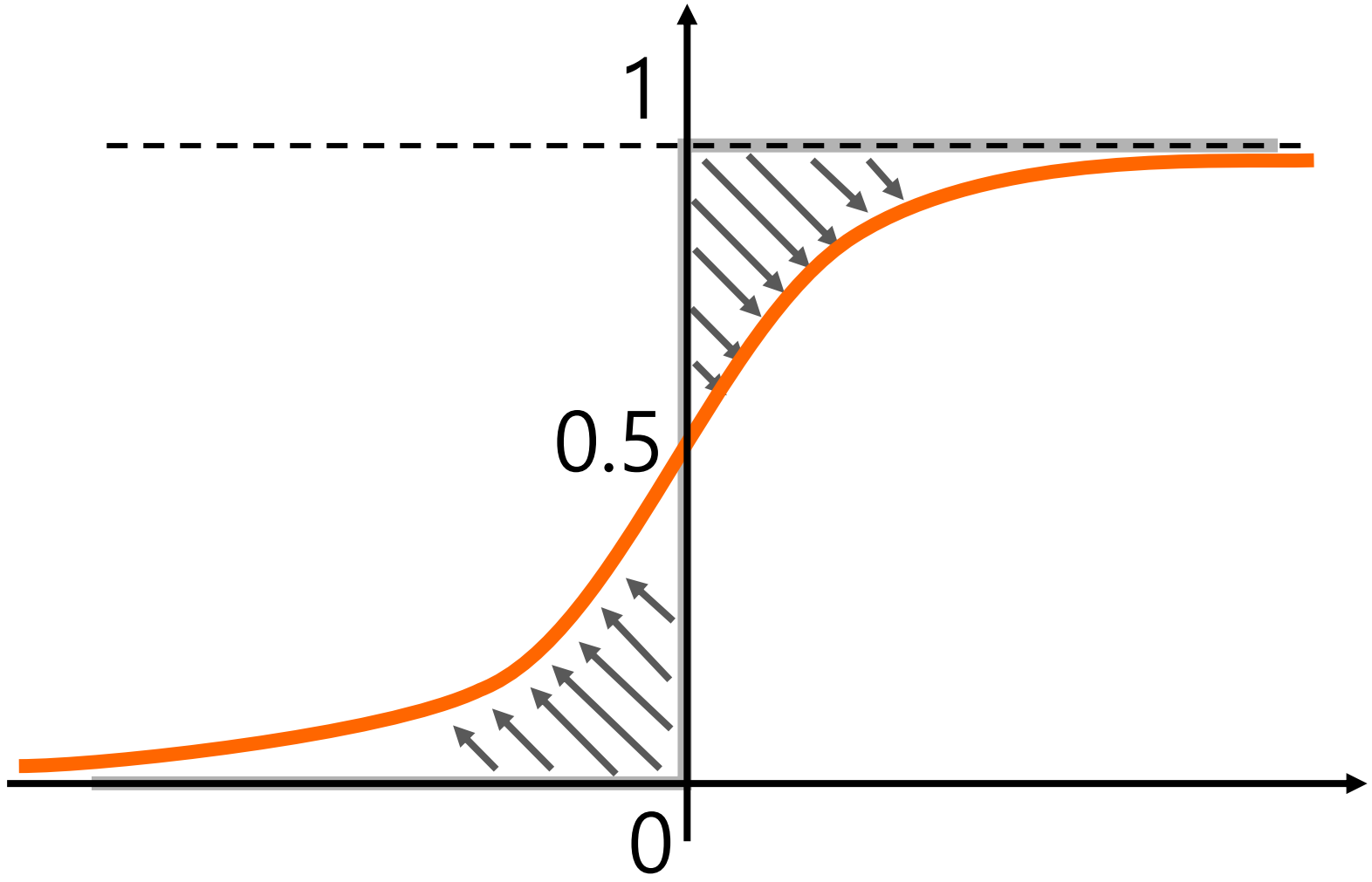
신박AI

시그모이드 함수는 아래와 같이 S자 형태의 함수로서



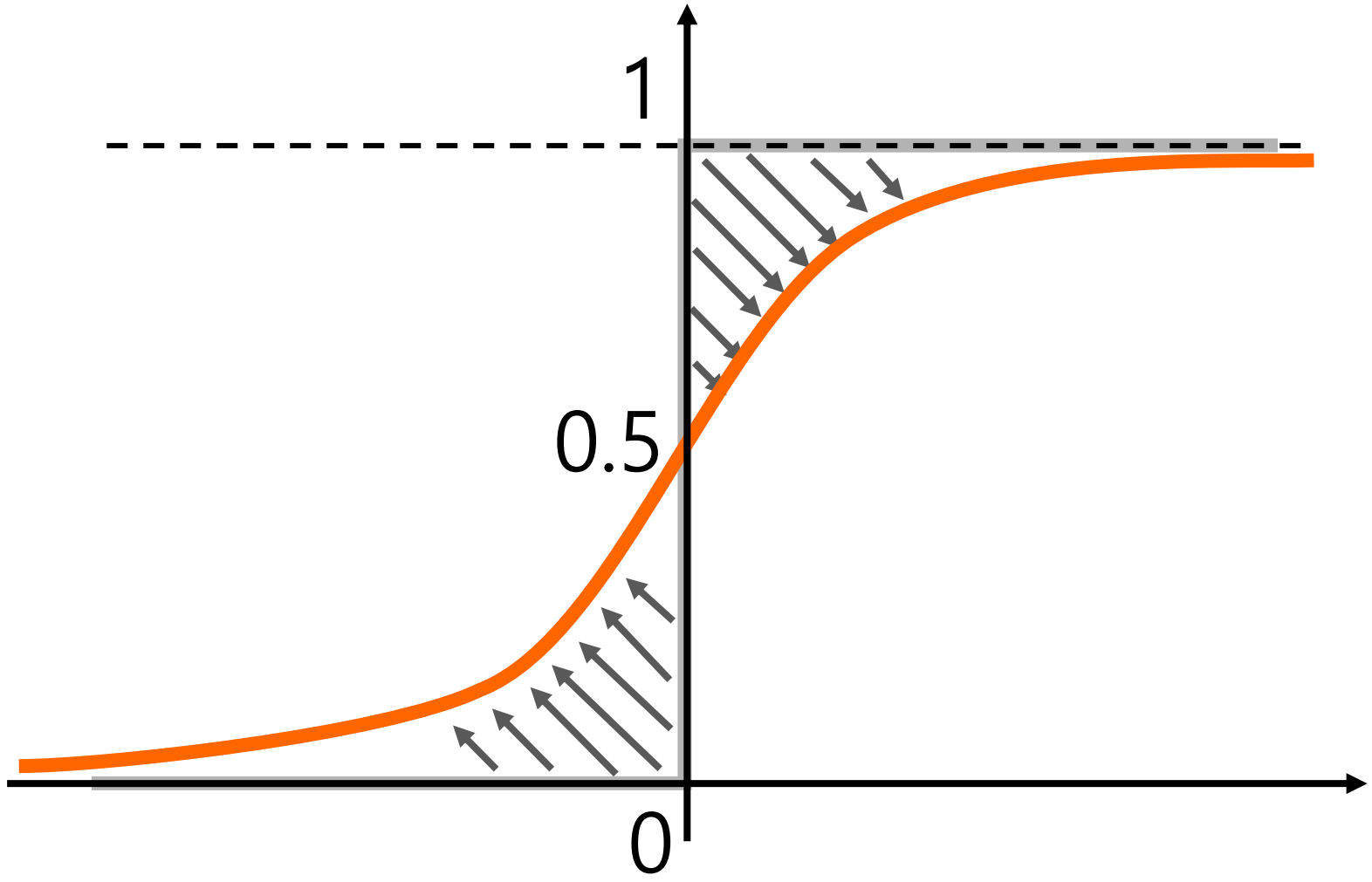
신박AI

계단함수를 좀 더 부드럽게 한 것 같이 생겼습니다



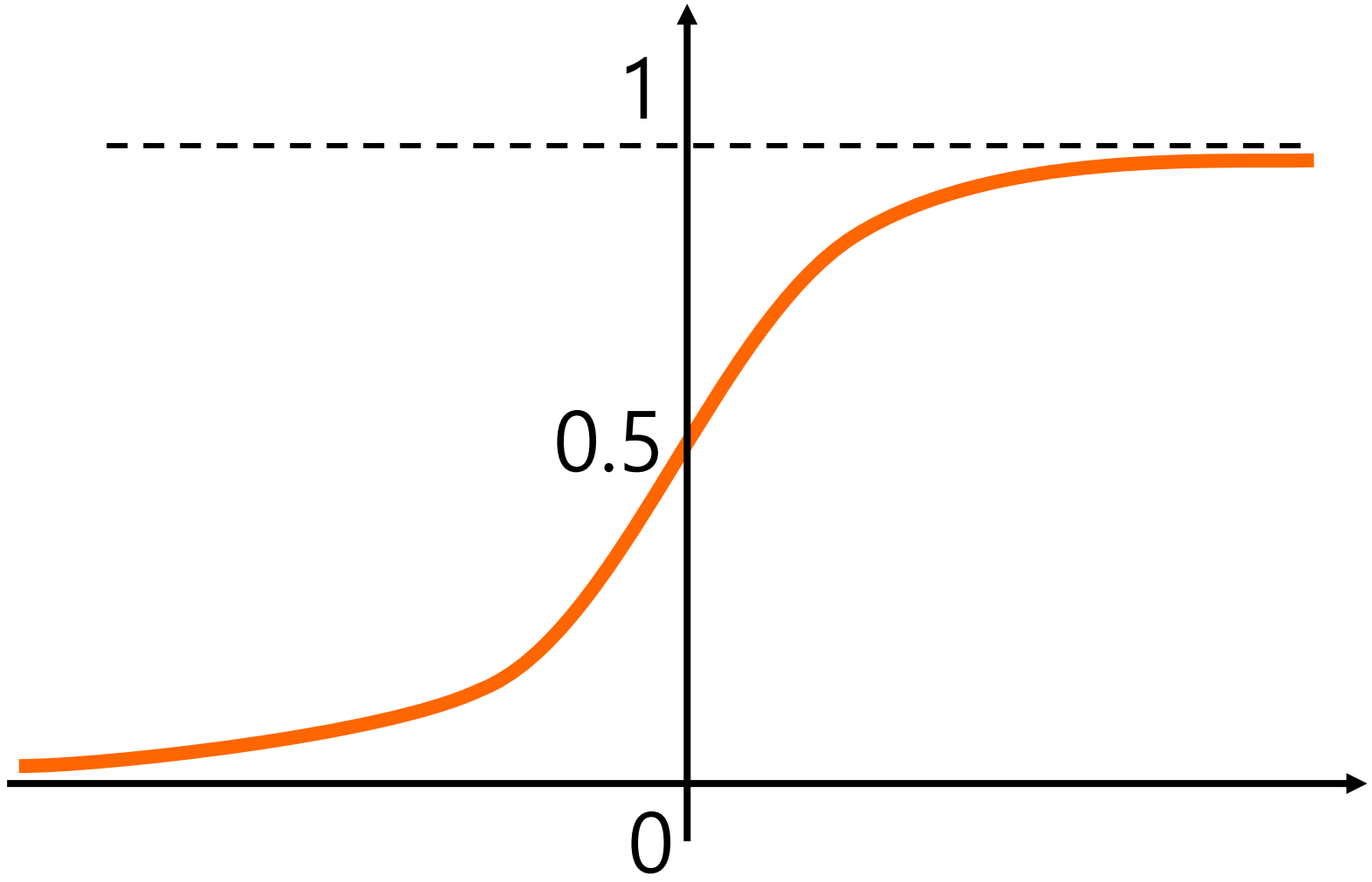
신박AI

모난 부분을 좀 부드럽게 한 것 뿐인 것 같은 이 차이가



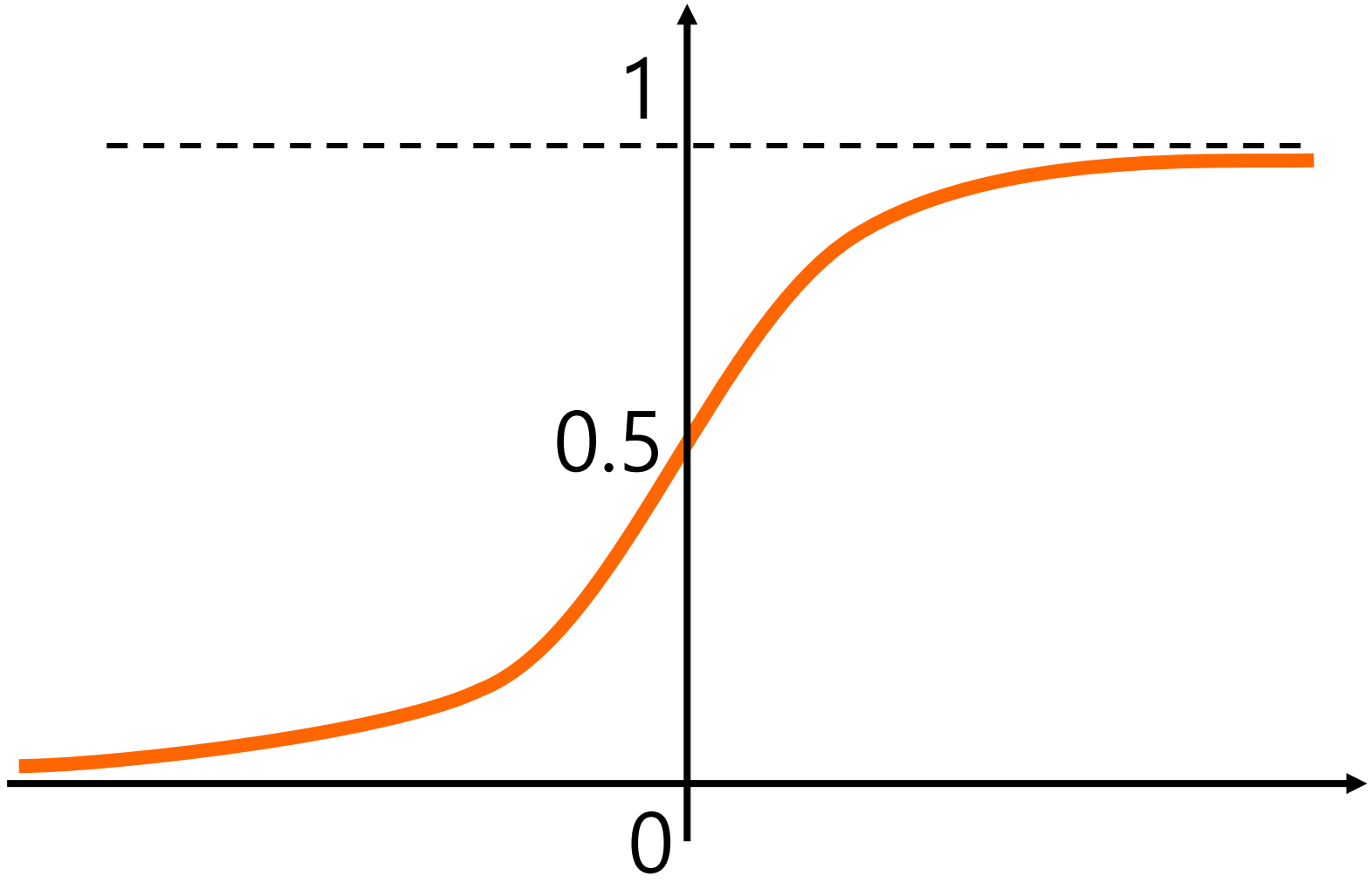
신박AI

다층 신경망의 시대를 열어주는 결정적인 혁신이 됩니다



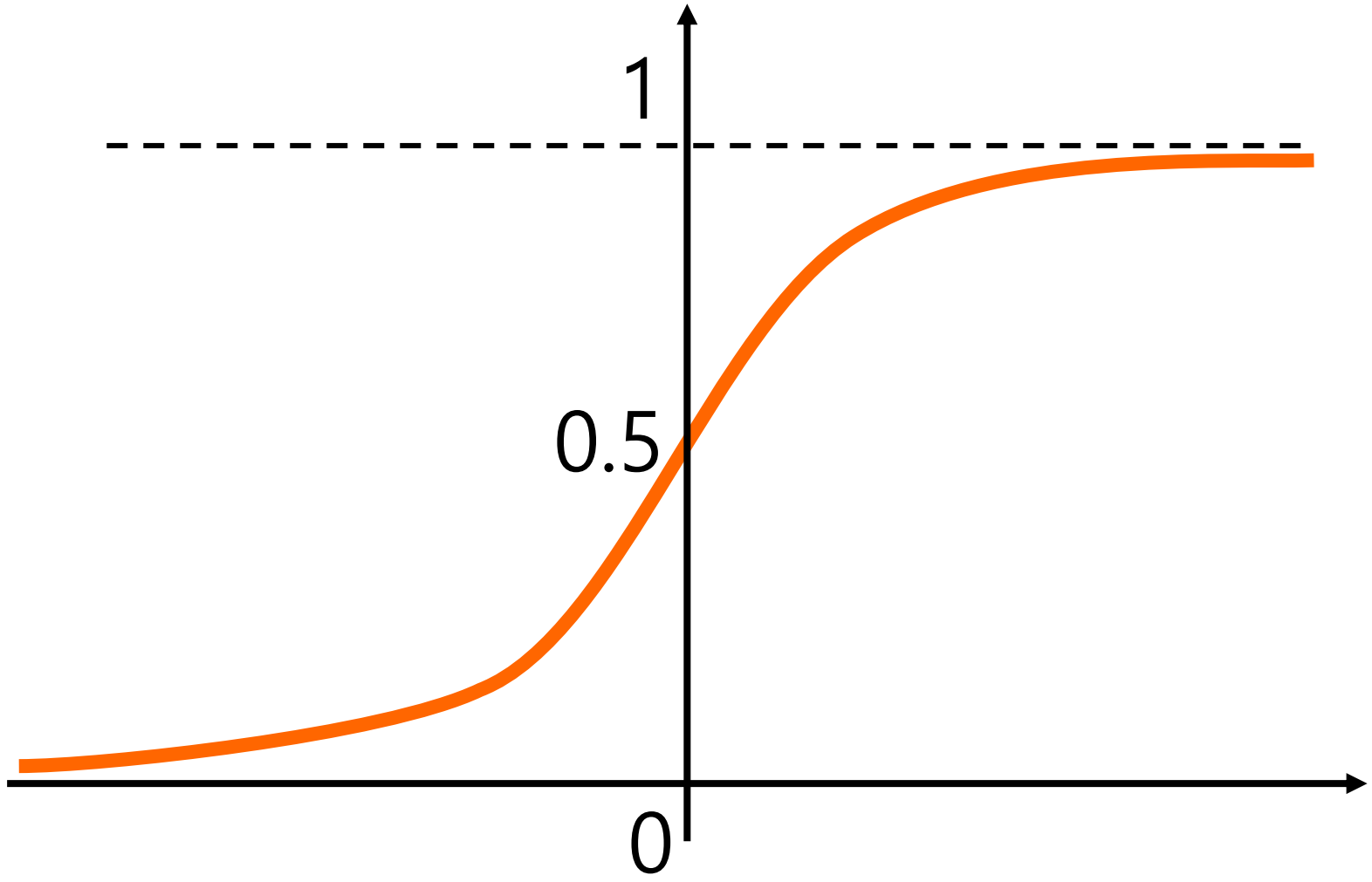
신박AI

앞으로 나올 영상에서 더 자세하게 다루겠지만,



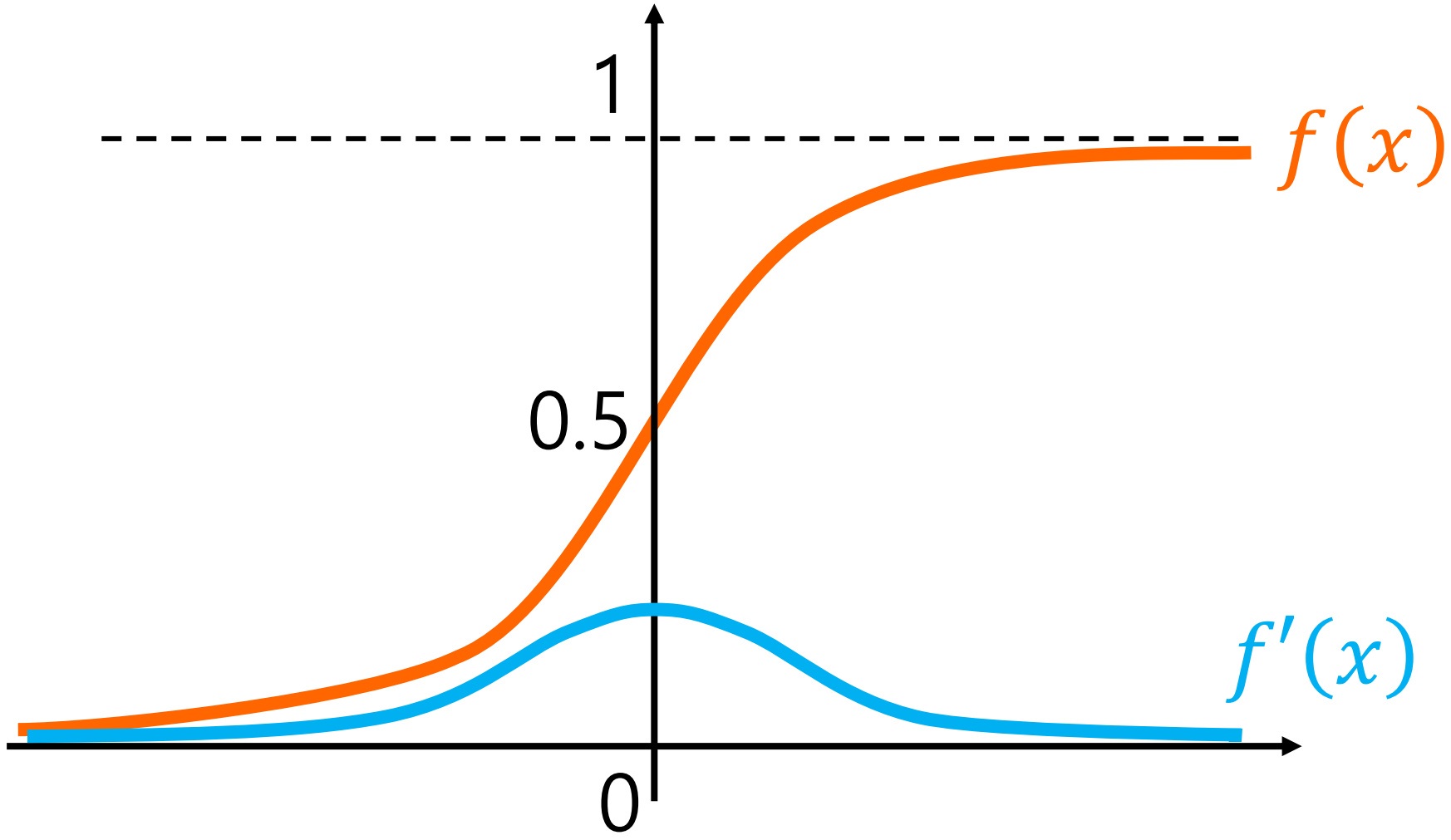
신박AI

이 작은 차이가 결정적 차이를 가져오게 된 이유에는,



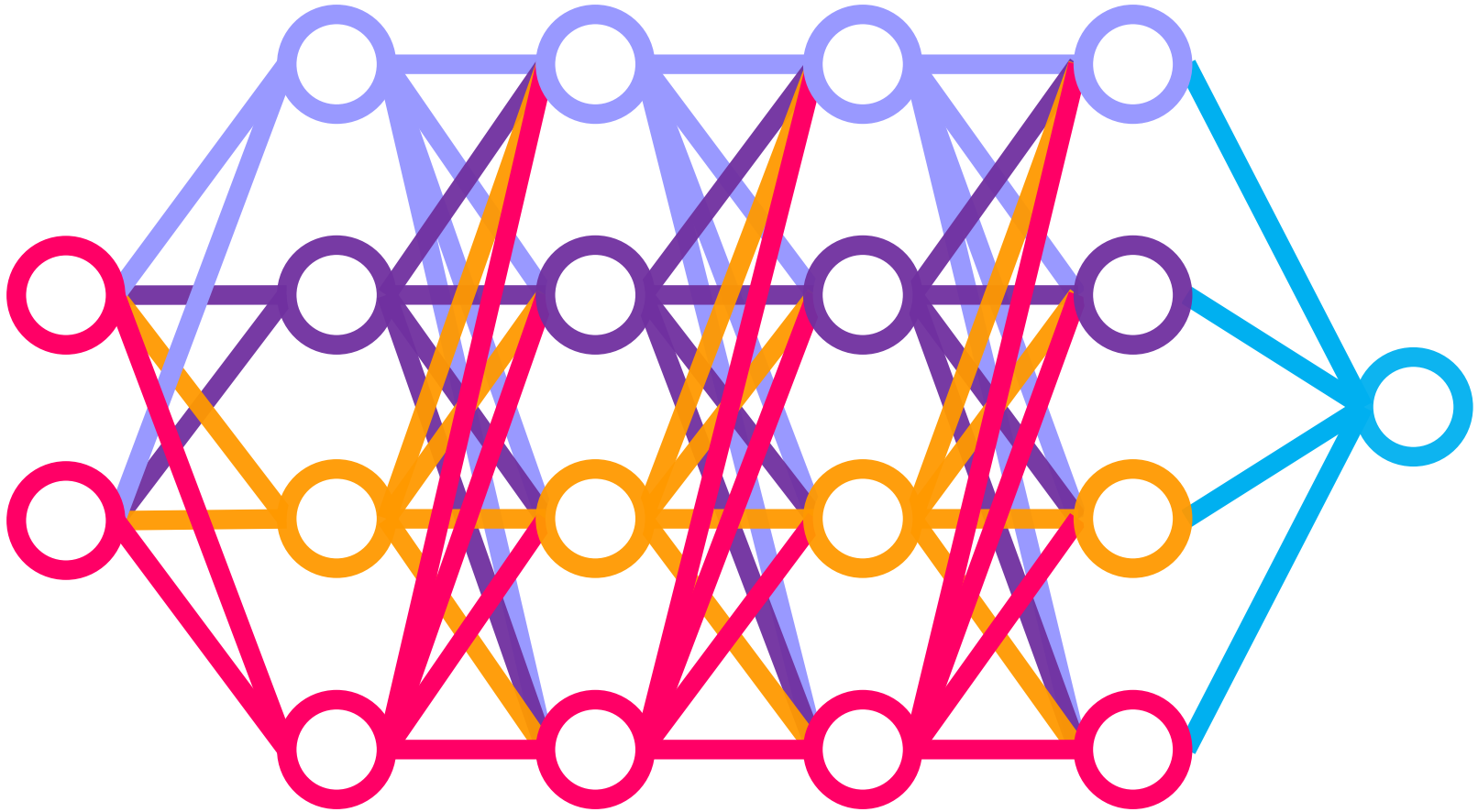
신박AI

시그모이드 함수는 미분가능한 differentiable 함수 이기 때문입니다



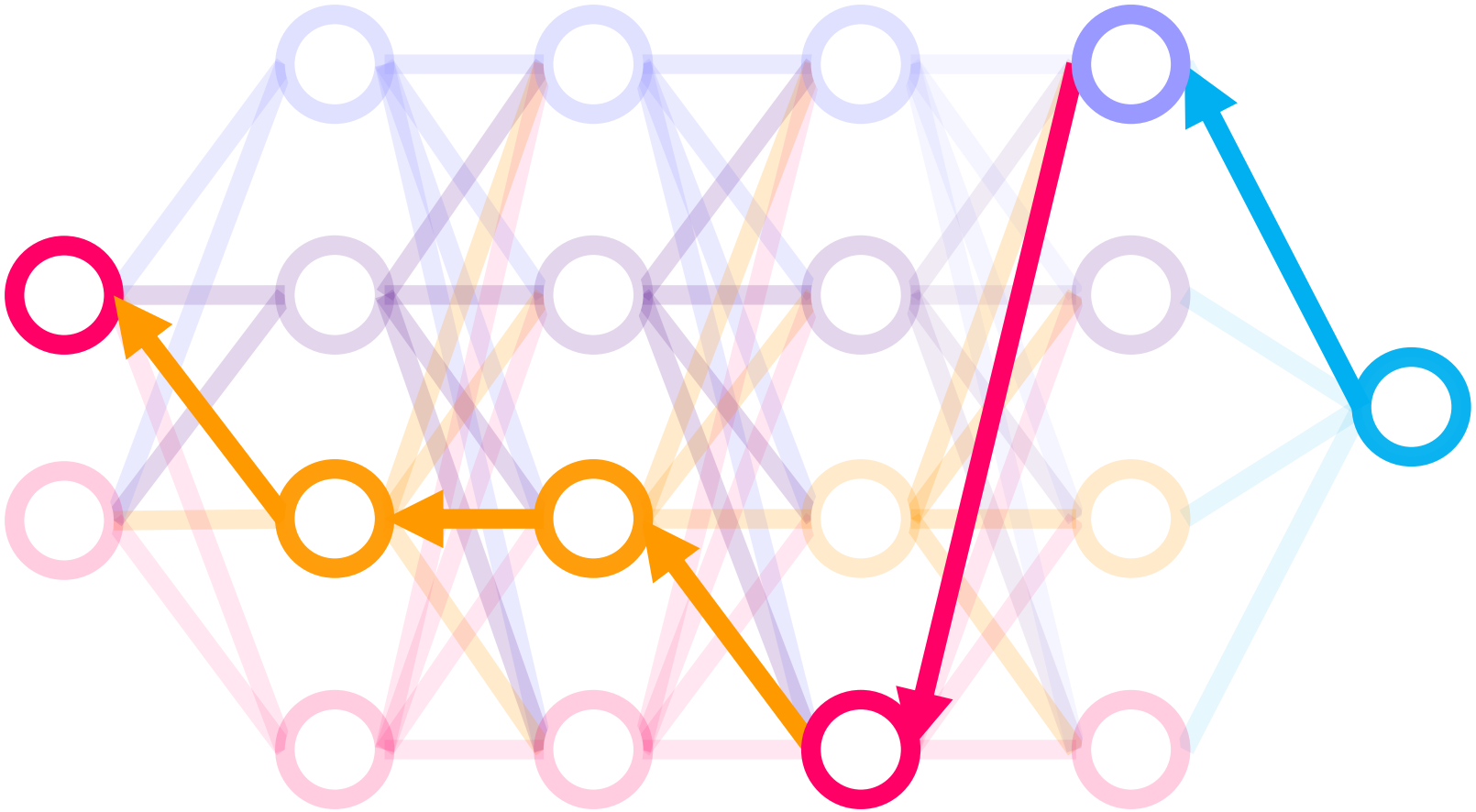
신박AI

다층 퍼셉트론의 학습에는



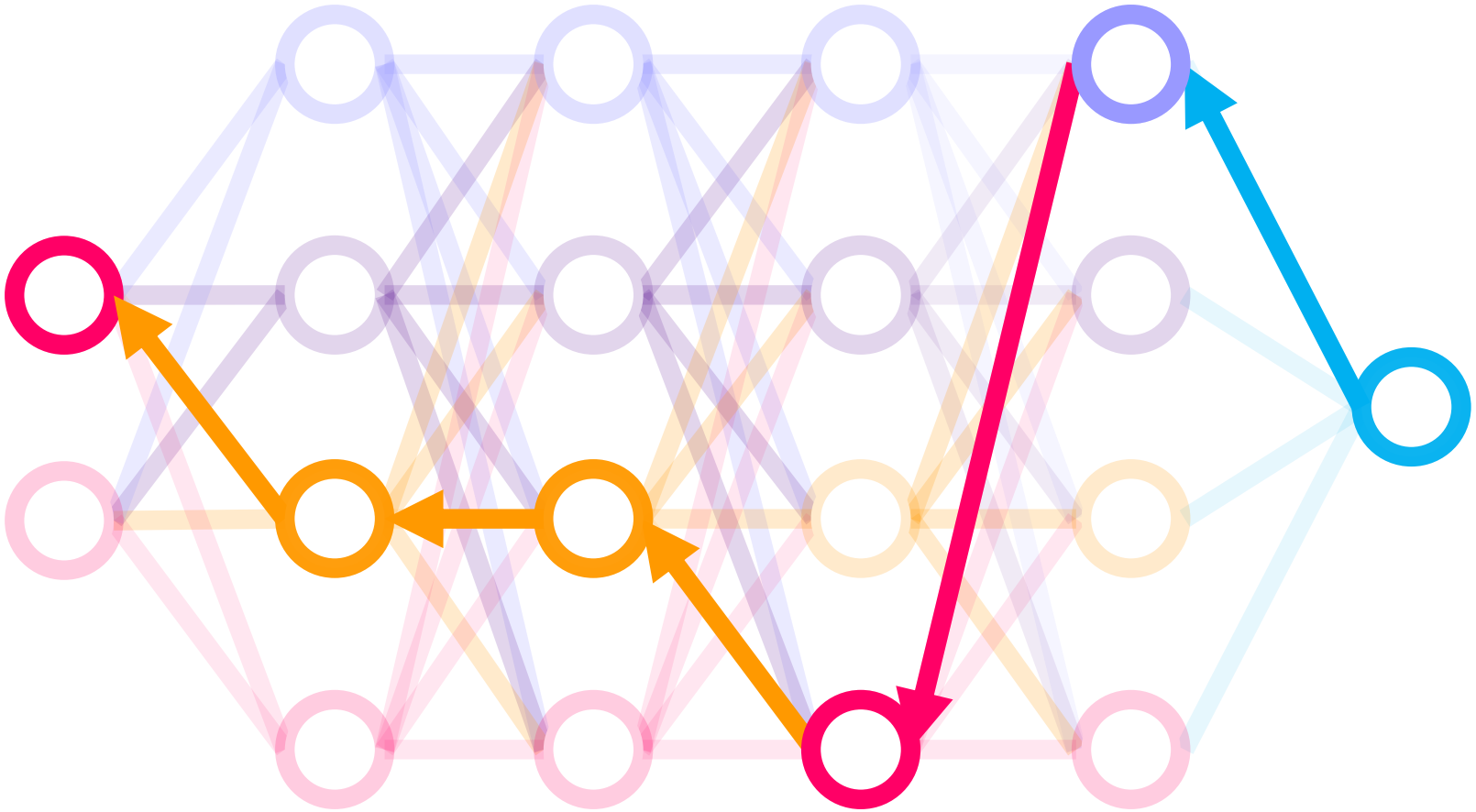
신박AI

오류역전파 backpropagation 라는 알고리즘을 사용합니다



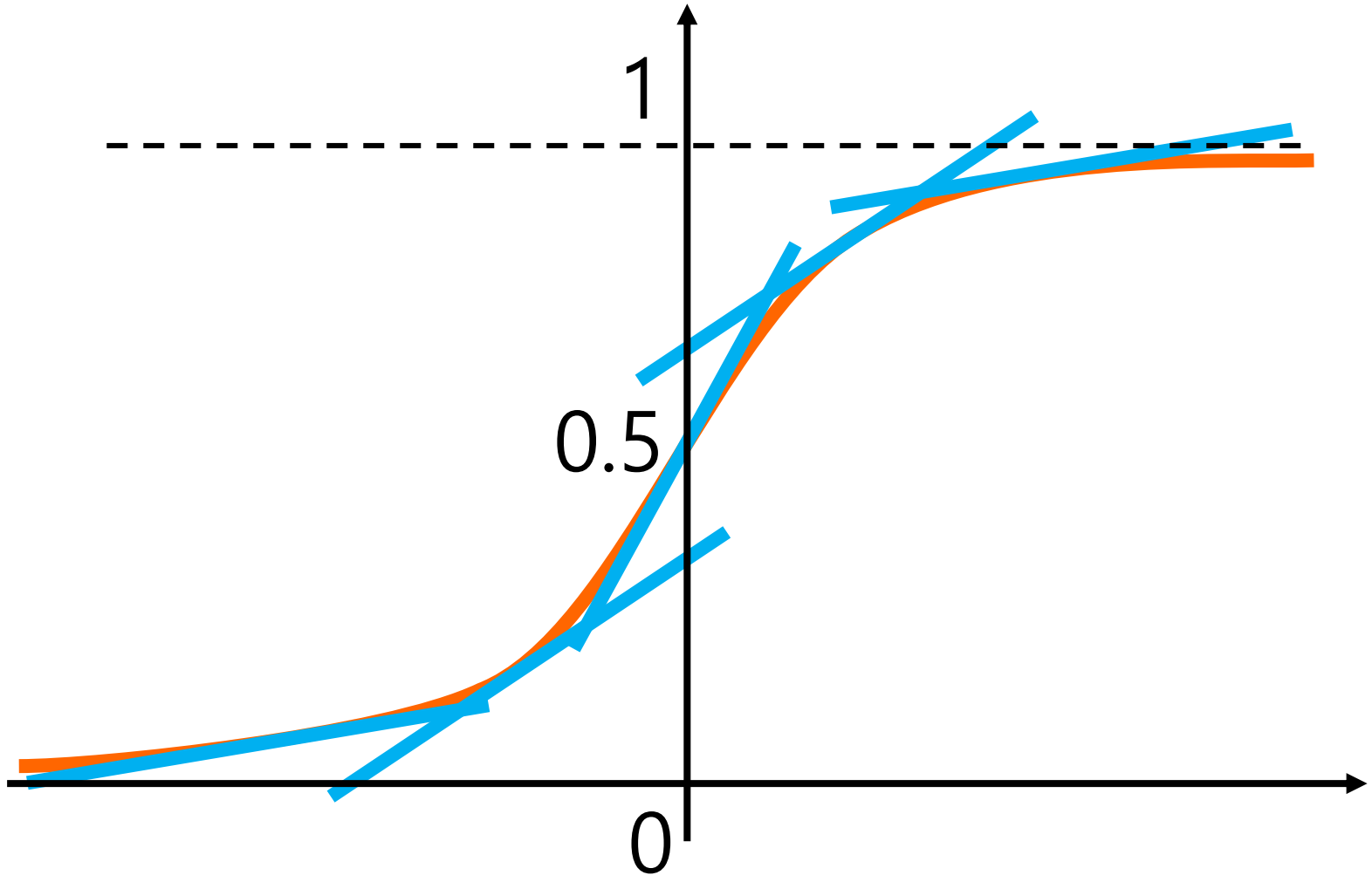
신박AI

오류역전파는 경사하강법 gradient descendent을 사용하는데

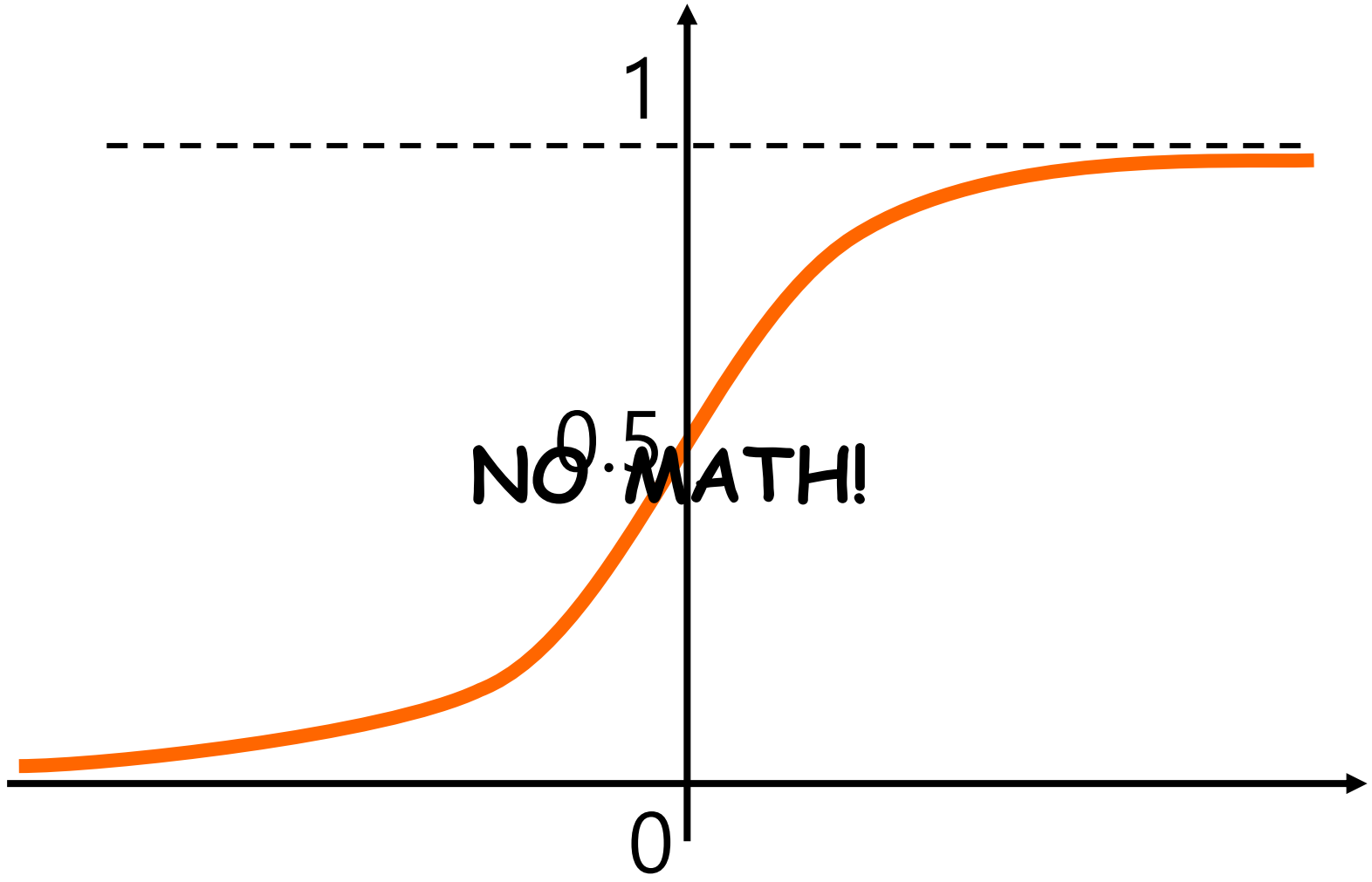


신박AI

경사하강법에는 활성화 함수가 미분가능 differentiable 해야합니다

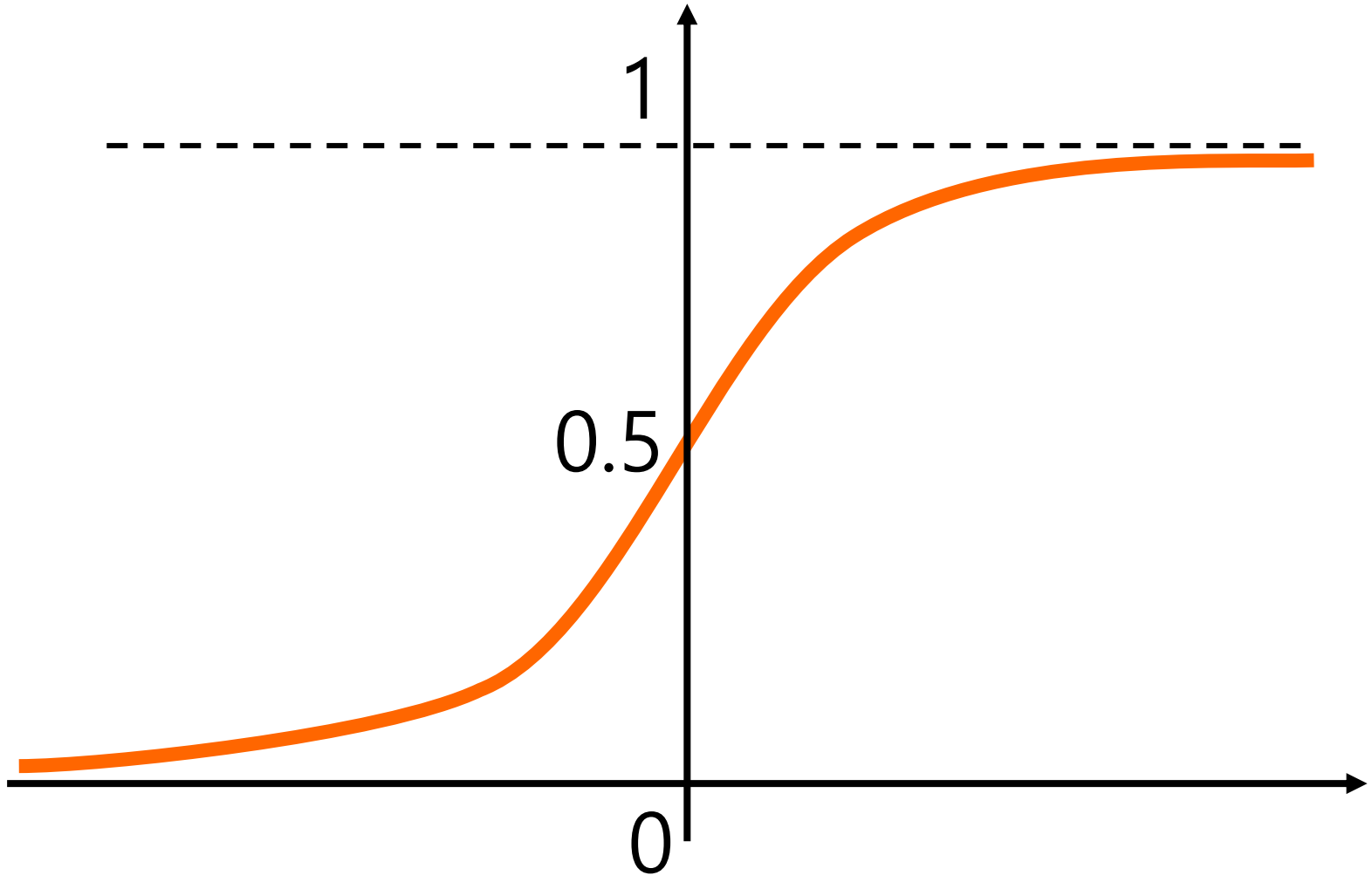


미분가능에 관한 부분은 수학의 영역이므로 여기서 다루지는 않겠습니다



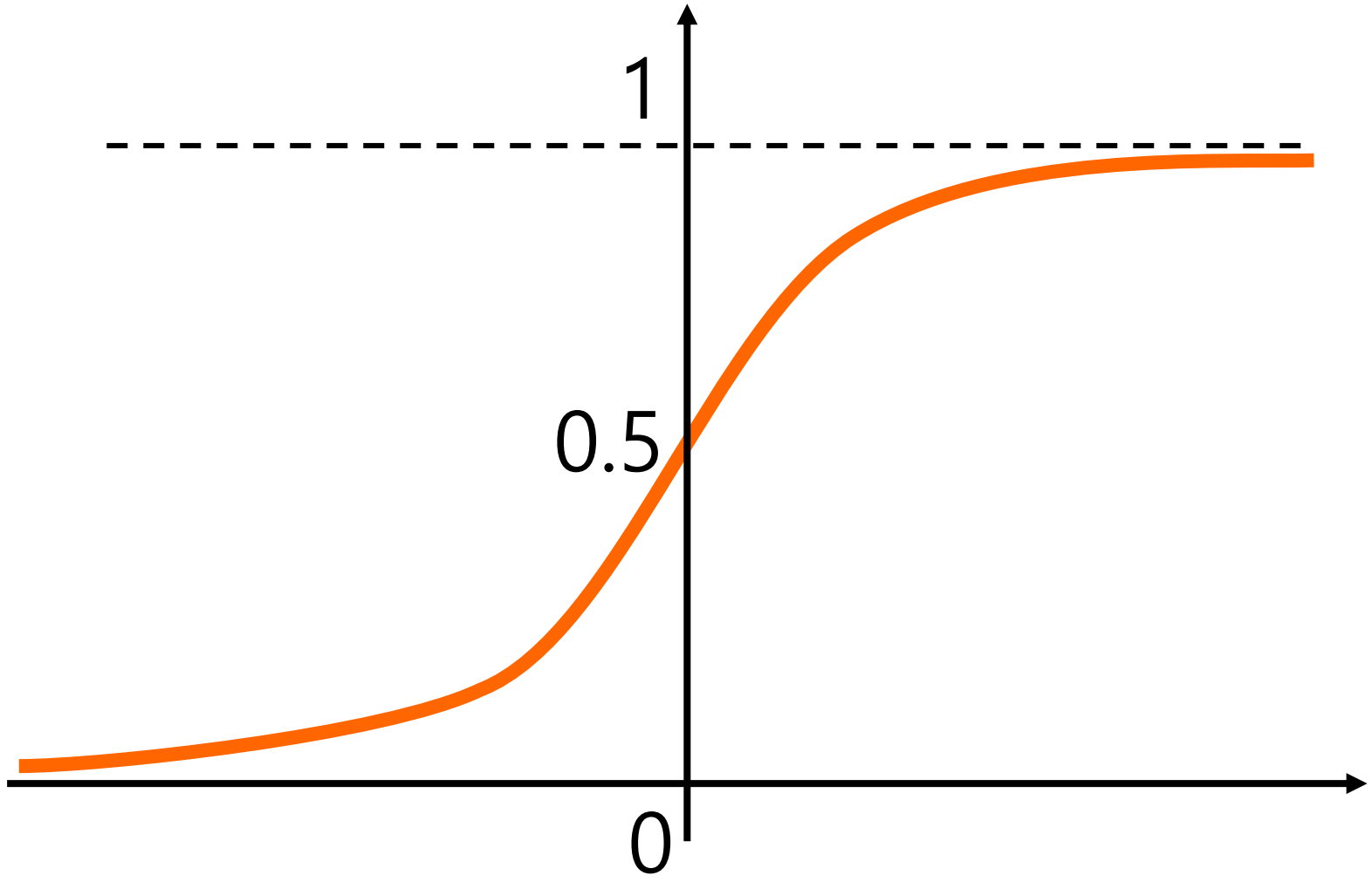
신박AI

아무튼 미분가능한 활성화 함수인 시그모이드 함수가



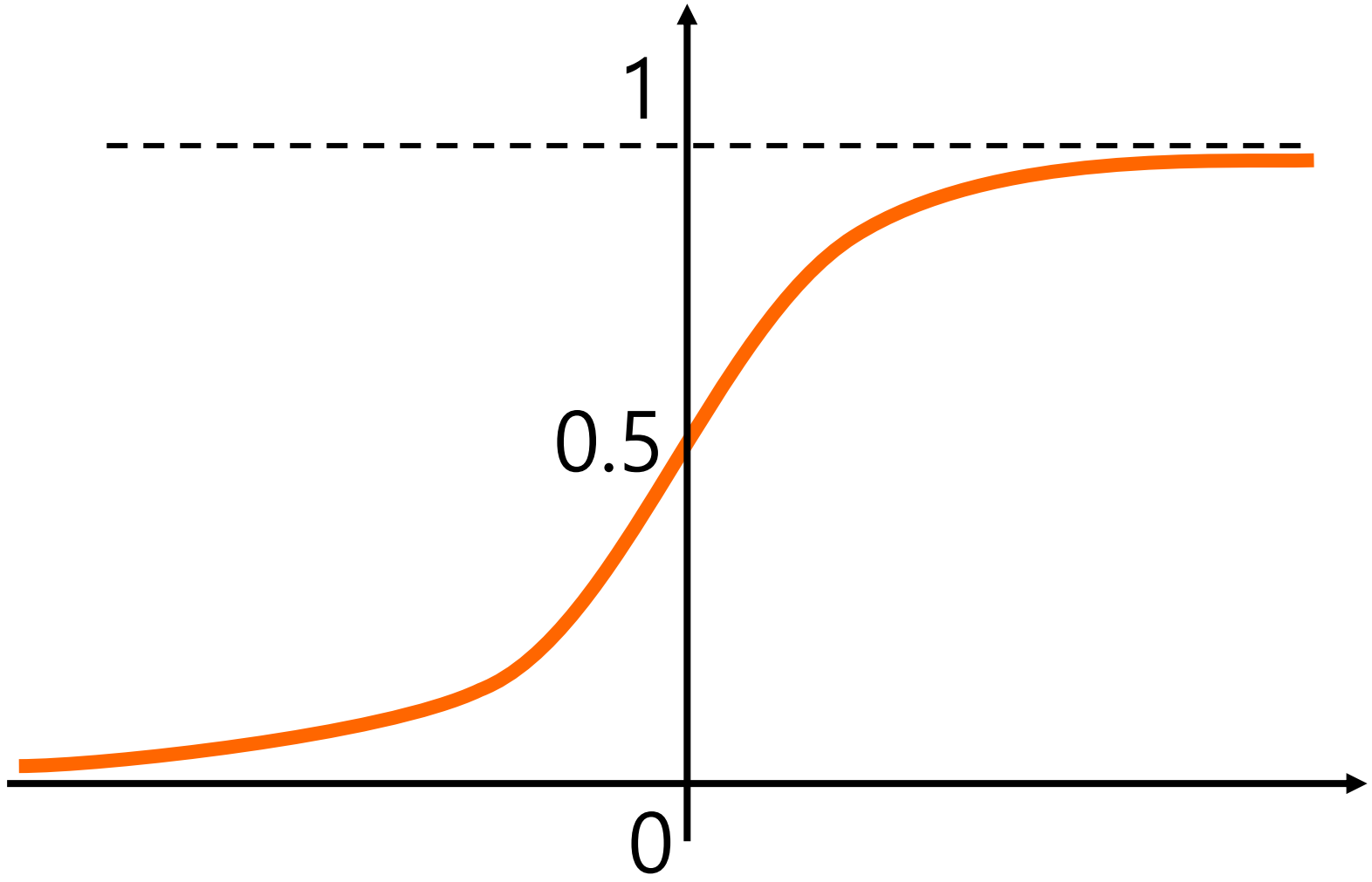
신박AI

다층 퍼셉트론의 새로운 장을 연 것만은 기억해 주시기를 부탁드립니다



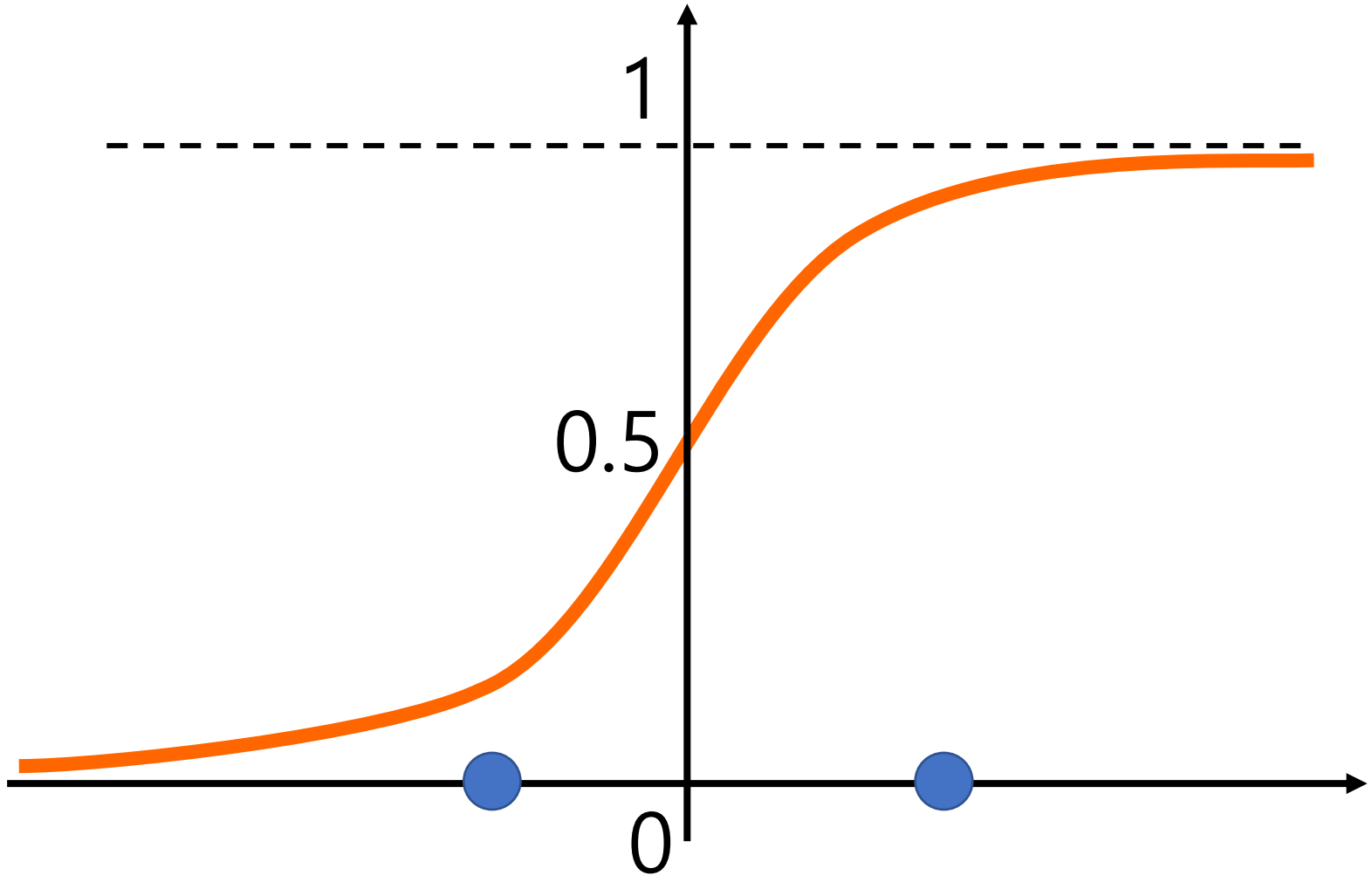
신박AI

시그모이드 함수가 갖는 장점으로는, 무엇보다도

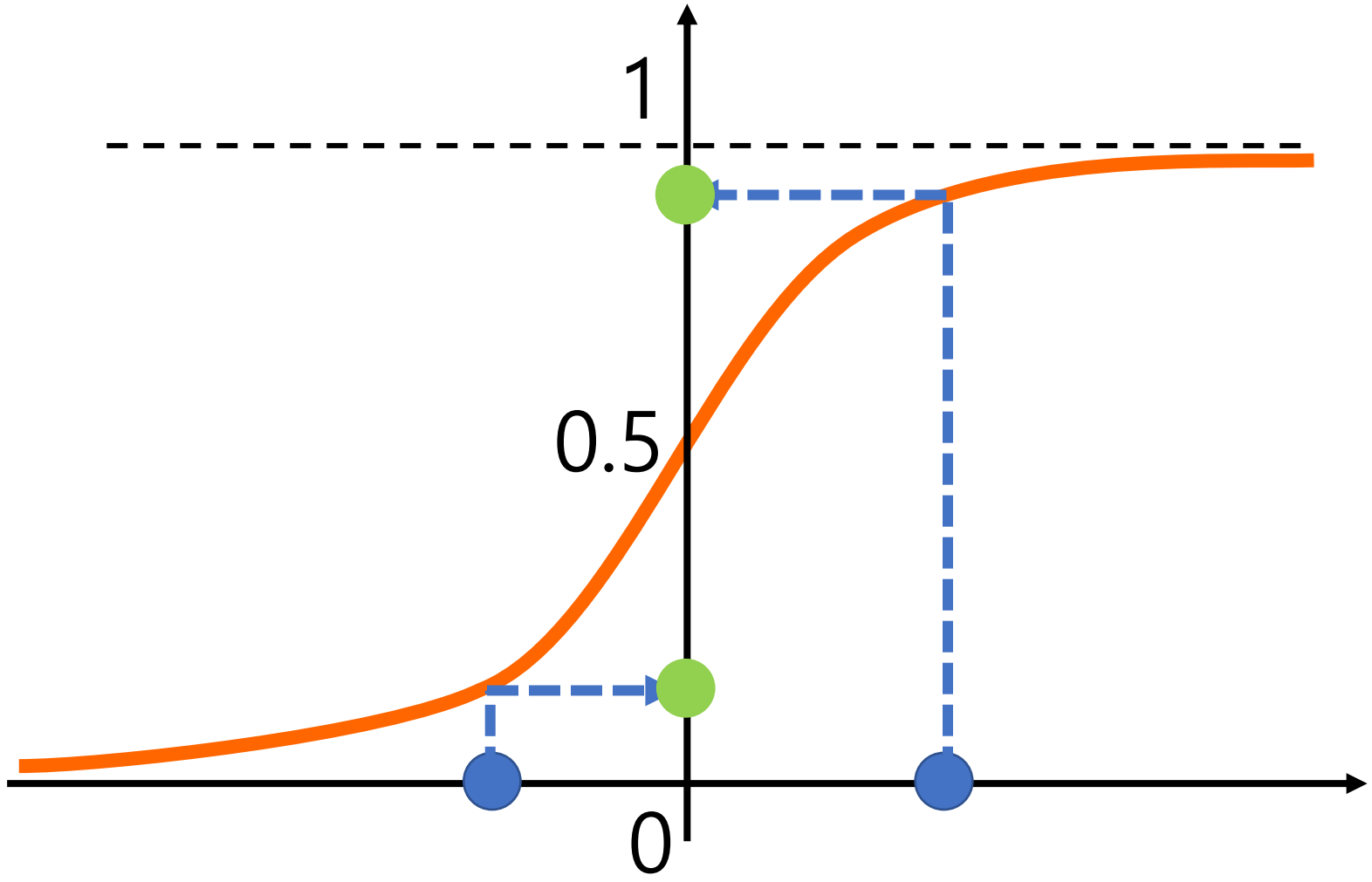


신박AI

이전 계단함수로는 표현할 수 없었던,

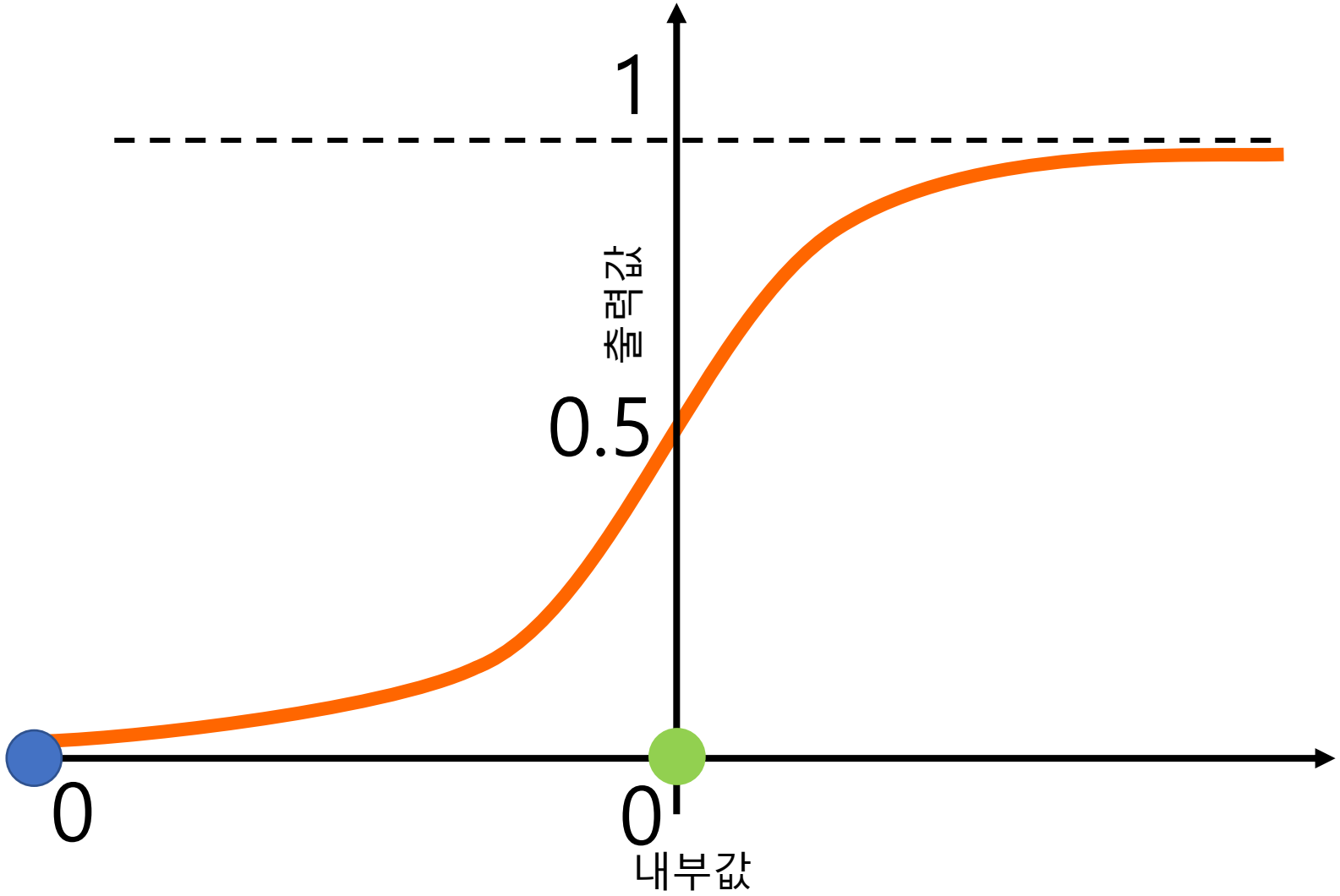


내부값의 차이를 표현할 수가 있었습니다



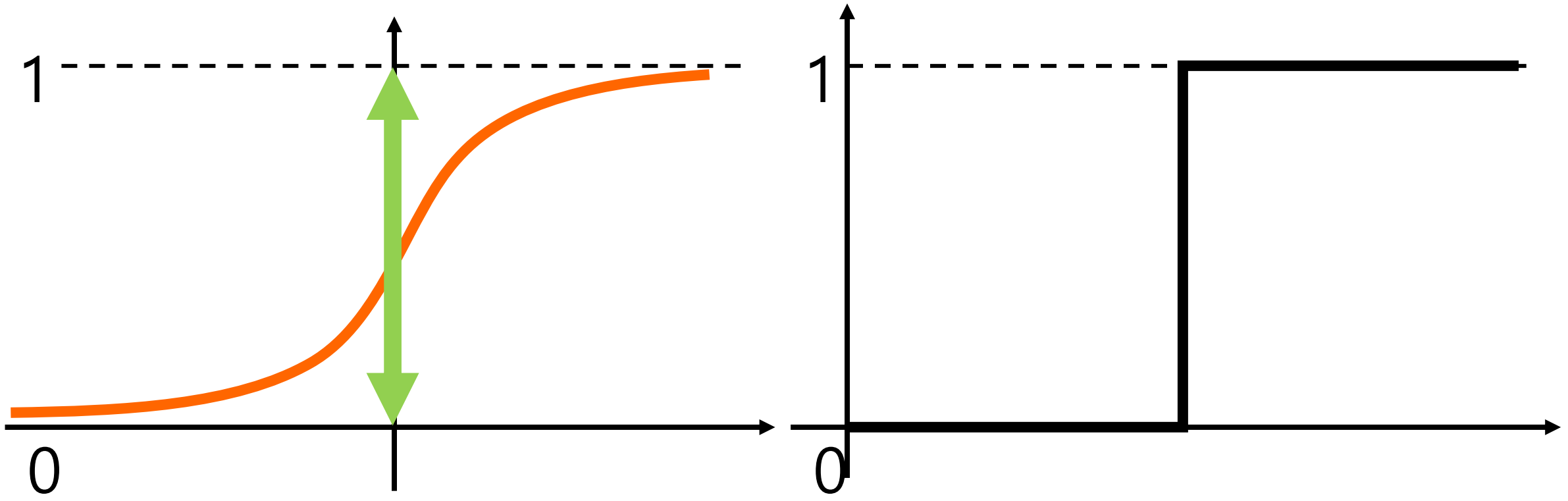
신박AI

즉 시그모이드는 신경망의 내부값에 따라 0과 1사이의 값을 출력하는 함수입니다



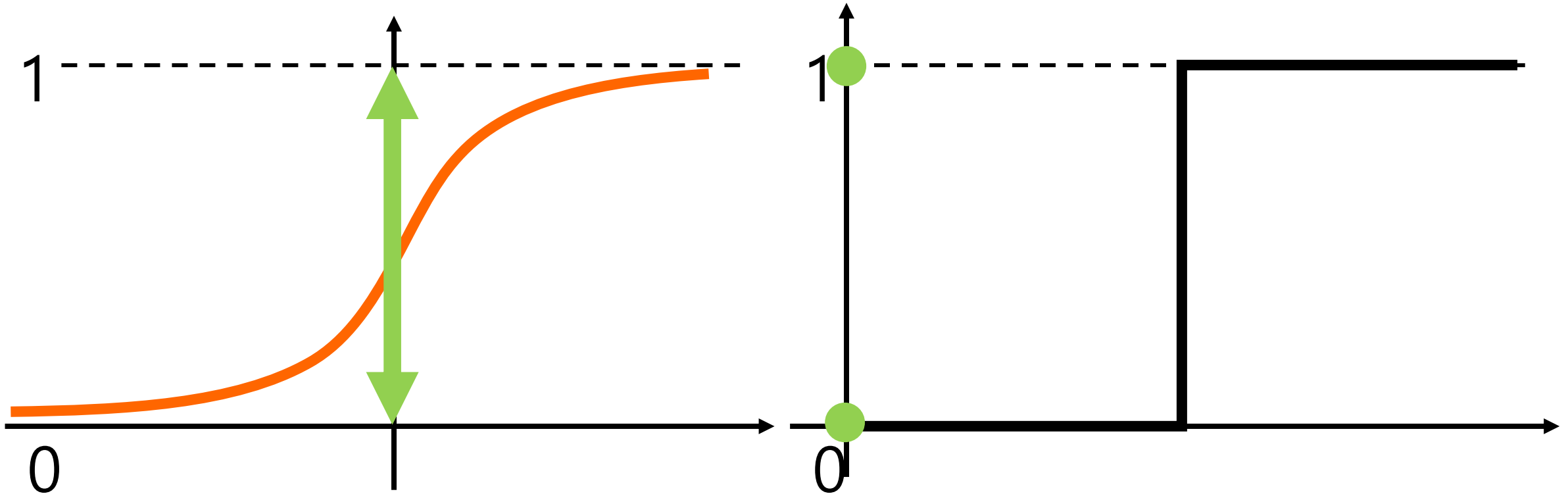
신박AI

시그모이드의 출력값이 0과 1사이이기 때문에,



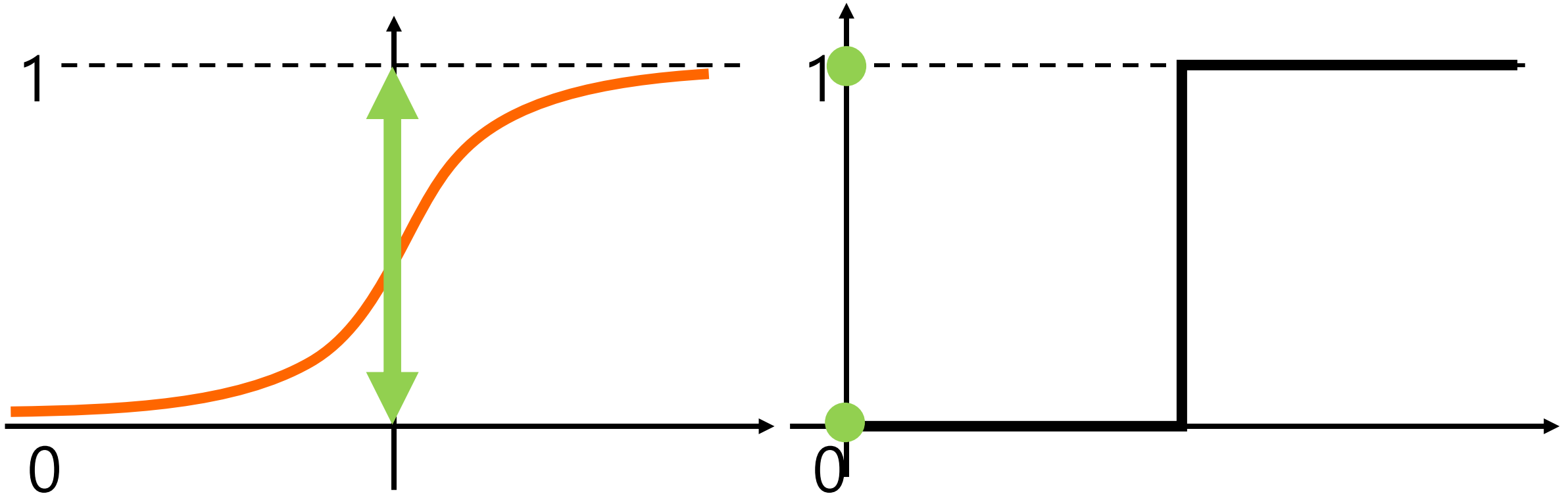
신박AI

0과 1로만 출력값을 갖는 계단 함수에 비하면 훨씬 다양한 오차를 계산할 수가 있습니다.



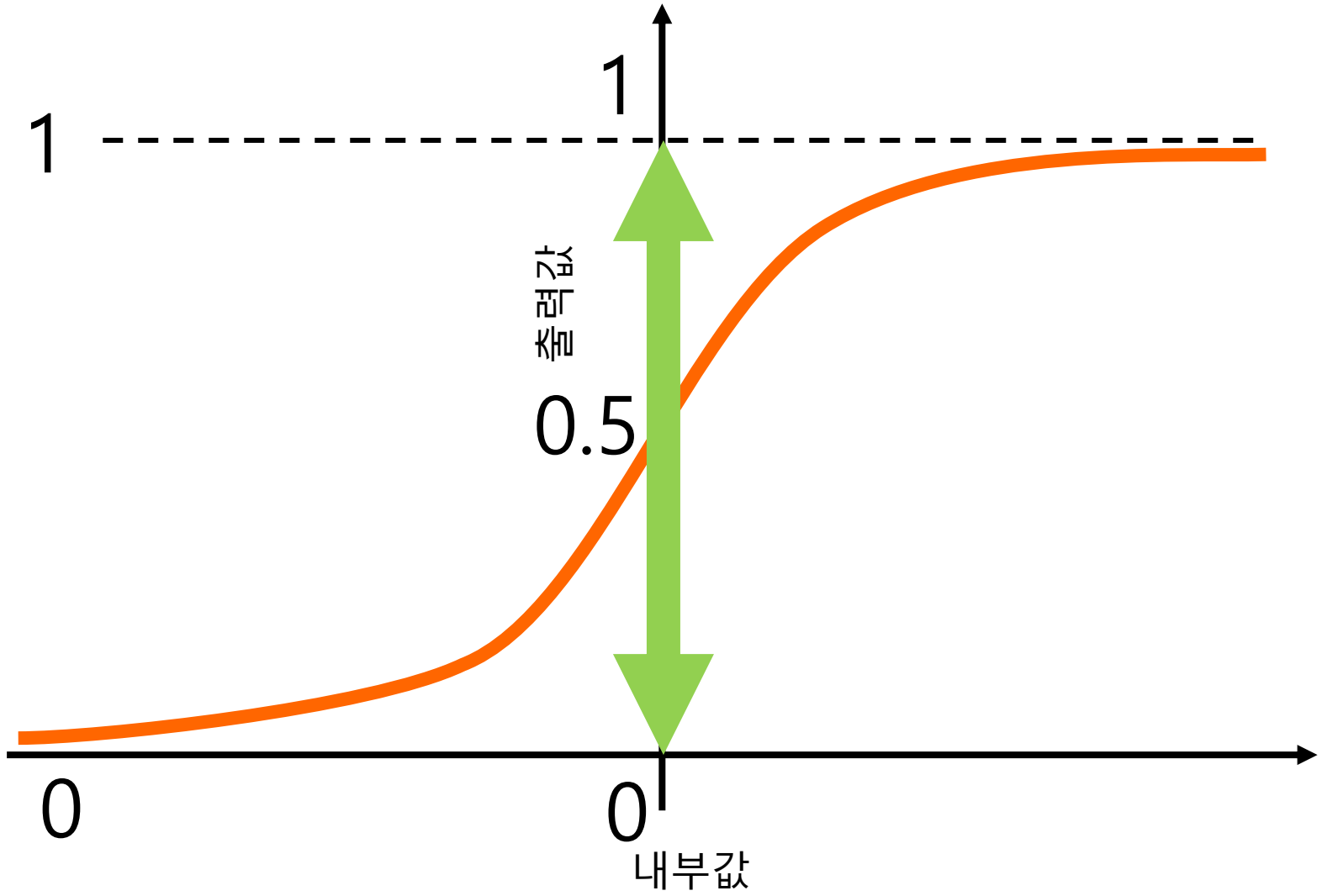
신박AI

0과 1로만 출력값을 갖는 계단 함수에 비하면 훨씬 다양한 오차를 계산할 수가 있습니다.



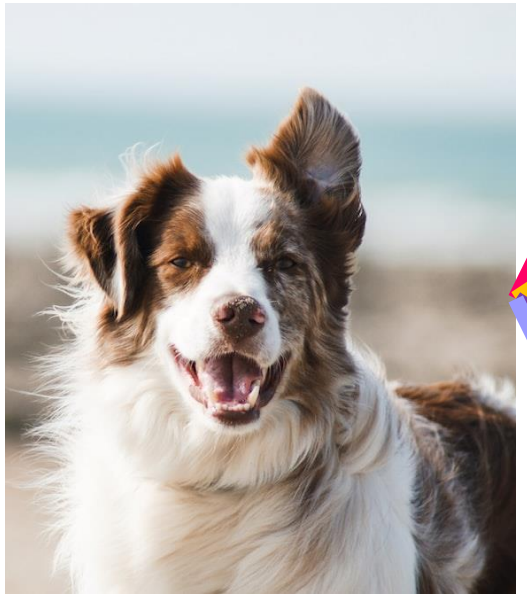
신박AI

또 0과 1사이라는 말은 내부값을 확률로 변환해주는 의미가 있습니다



신박AI

예를들자면, 개와 고양이를 분류하는 classifier의 경우



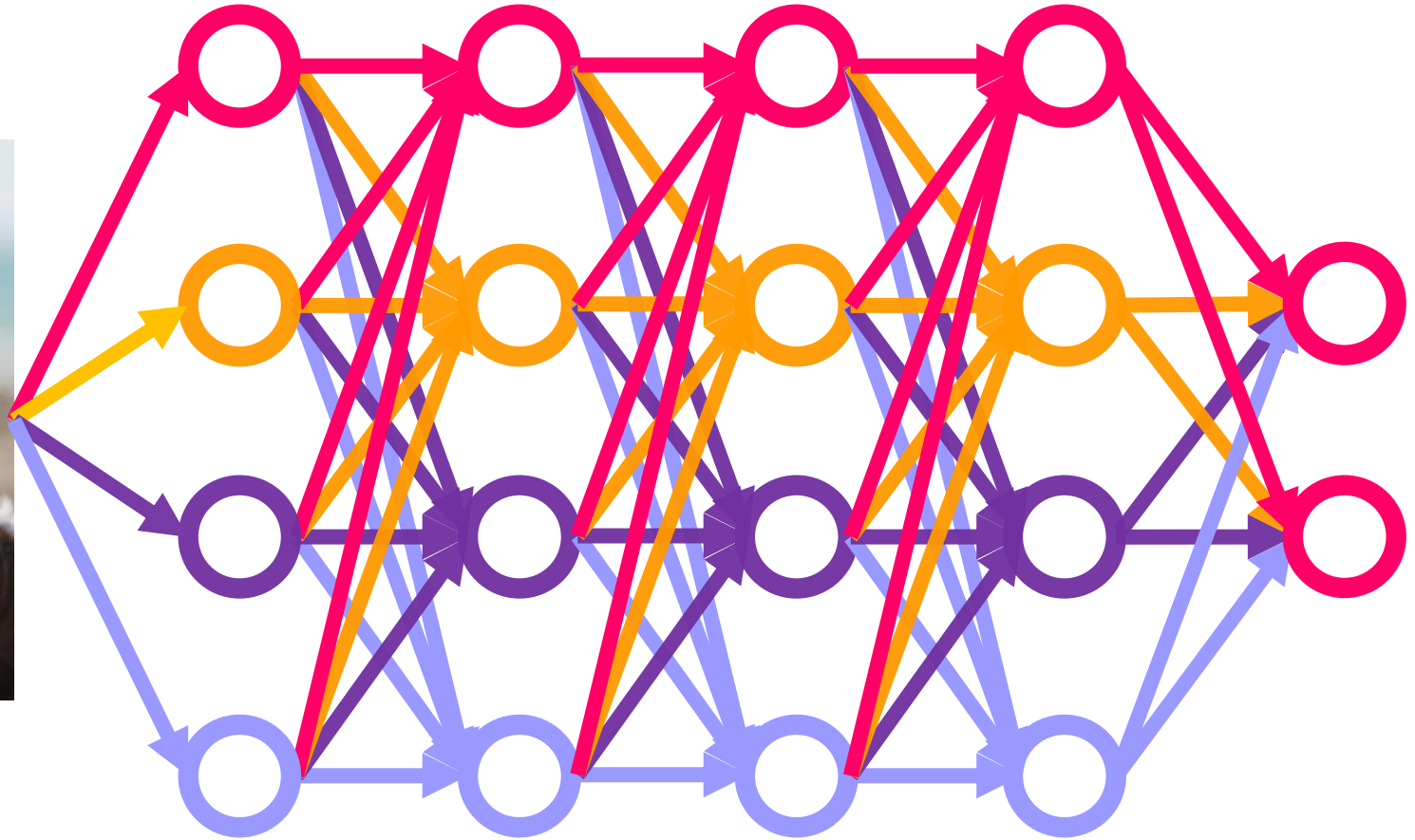
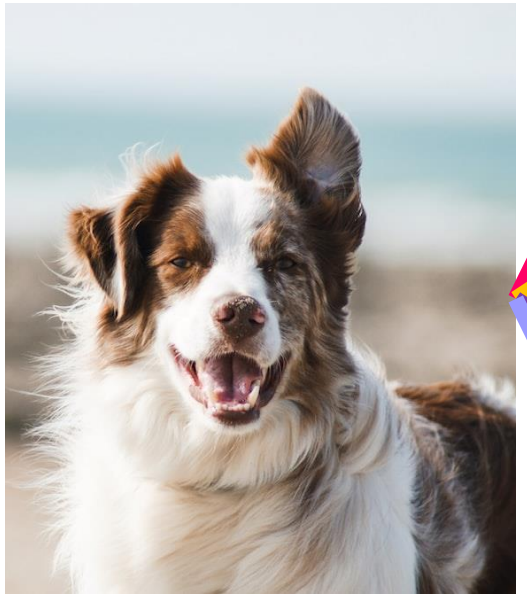


Photo by Pauline Loroy on Unsplash

신박AI

시그모이드 함수에 의해서 최종 출력값은



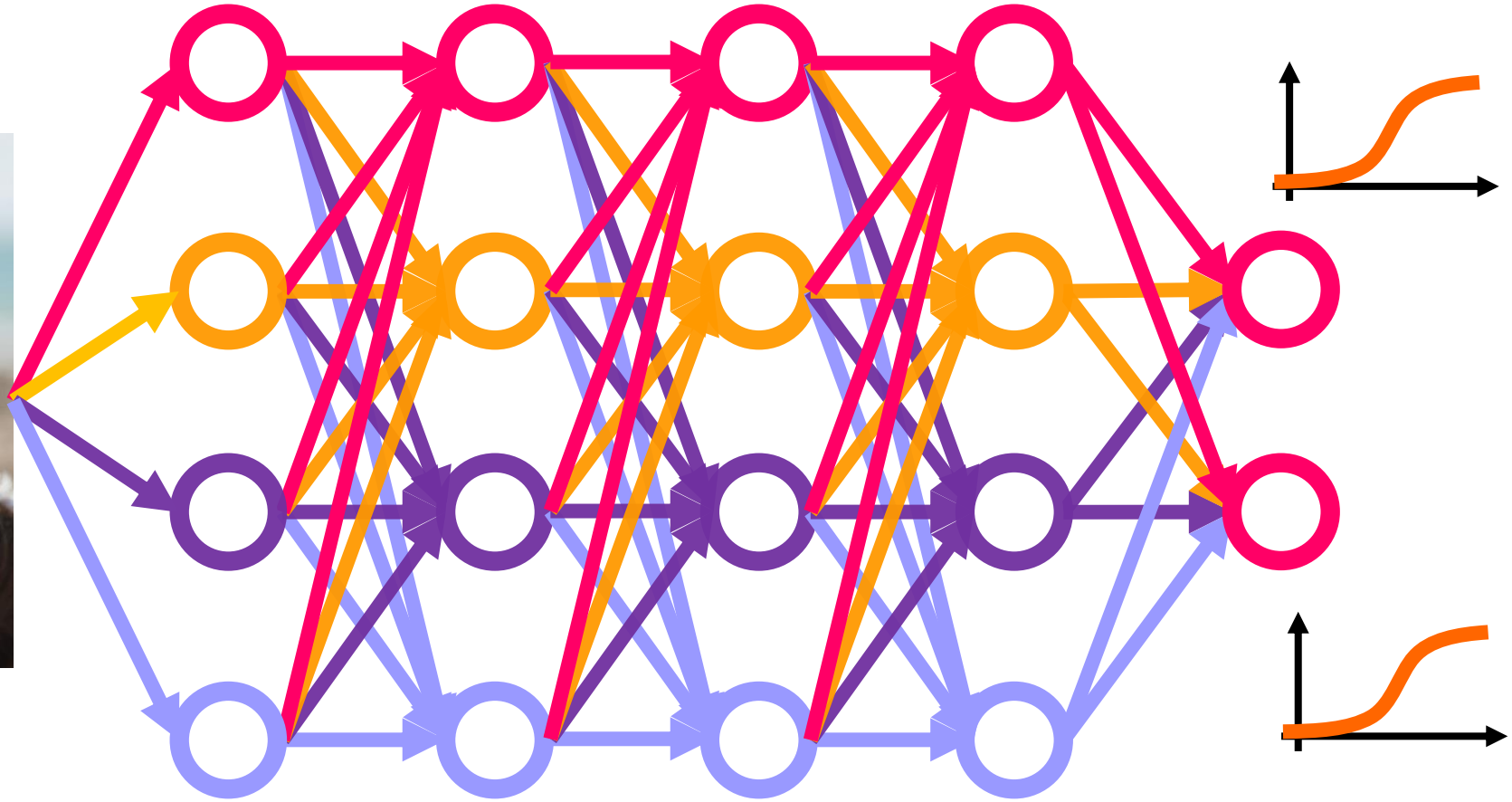
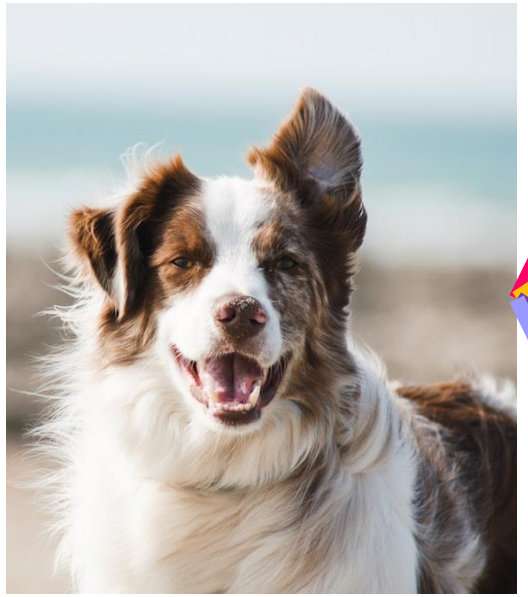


Photo by Pauline Loroy on Unsplash

신박AI

0과 1사이의 확률로 출력할 수 있습니다



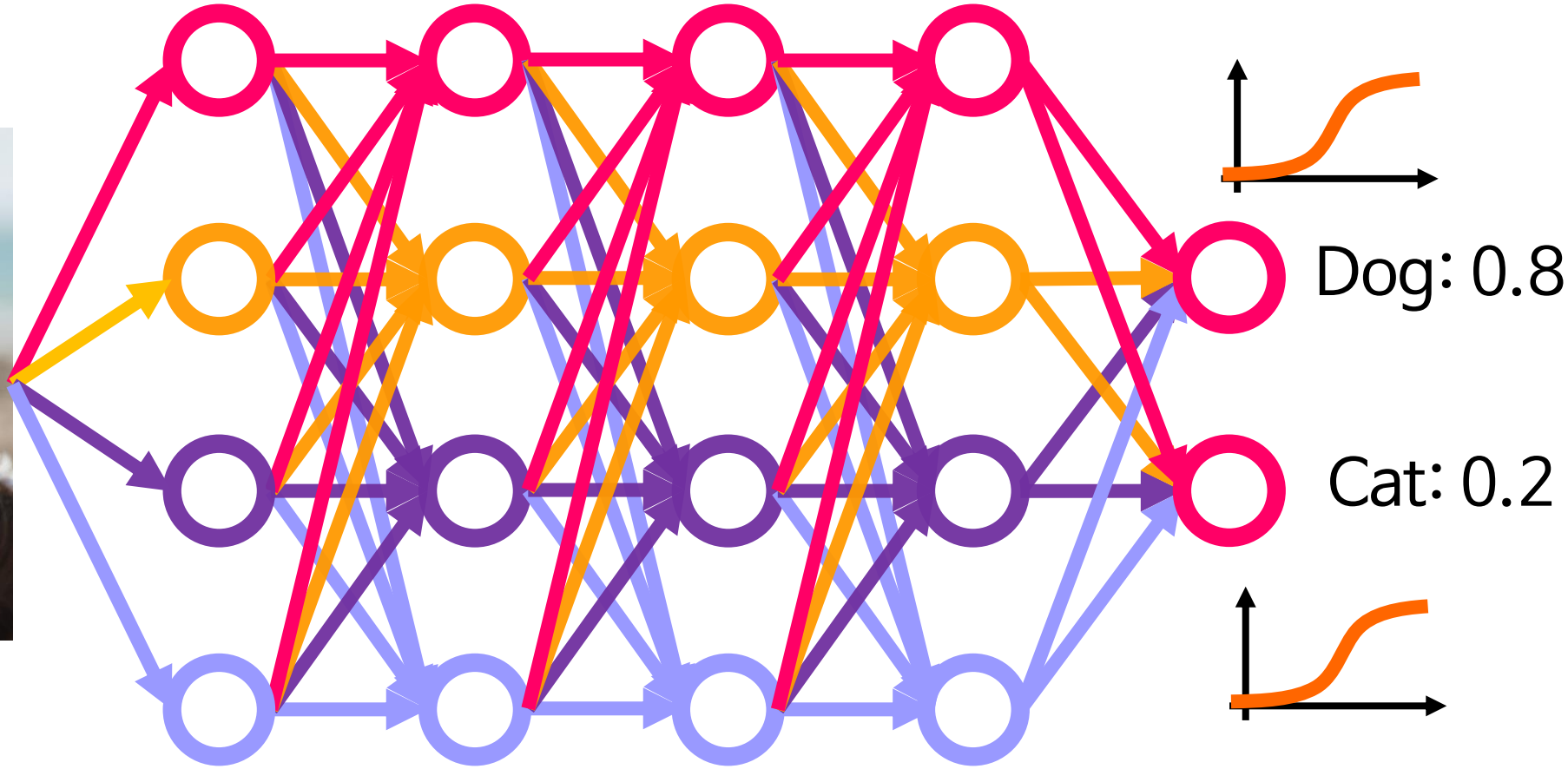
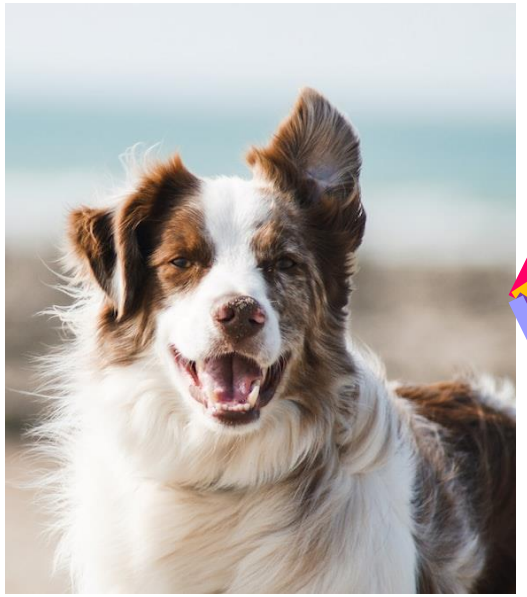


Photo by Pauline Loroy on Unsplash

신박AI

확률로 변환한다는 것은 여러 다양한 의미가 있을 수 있겠지만,



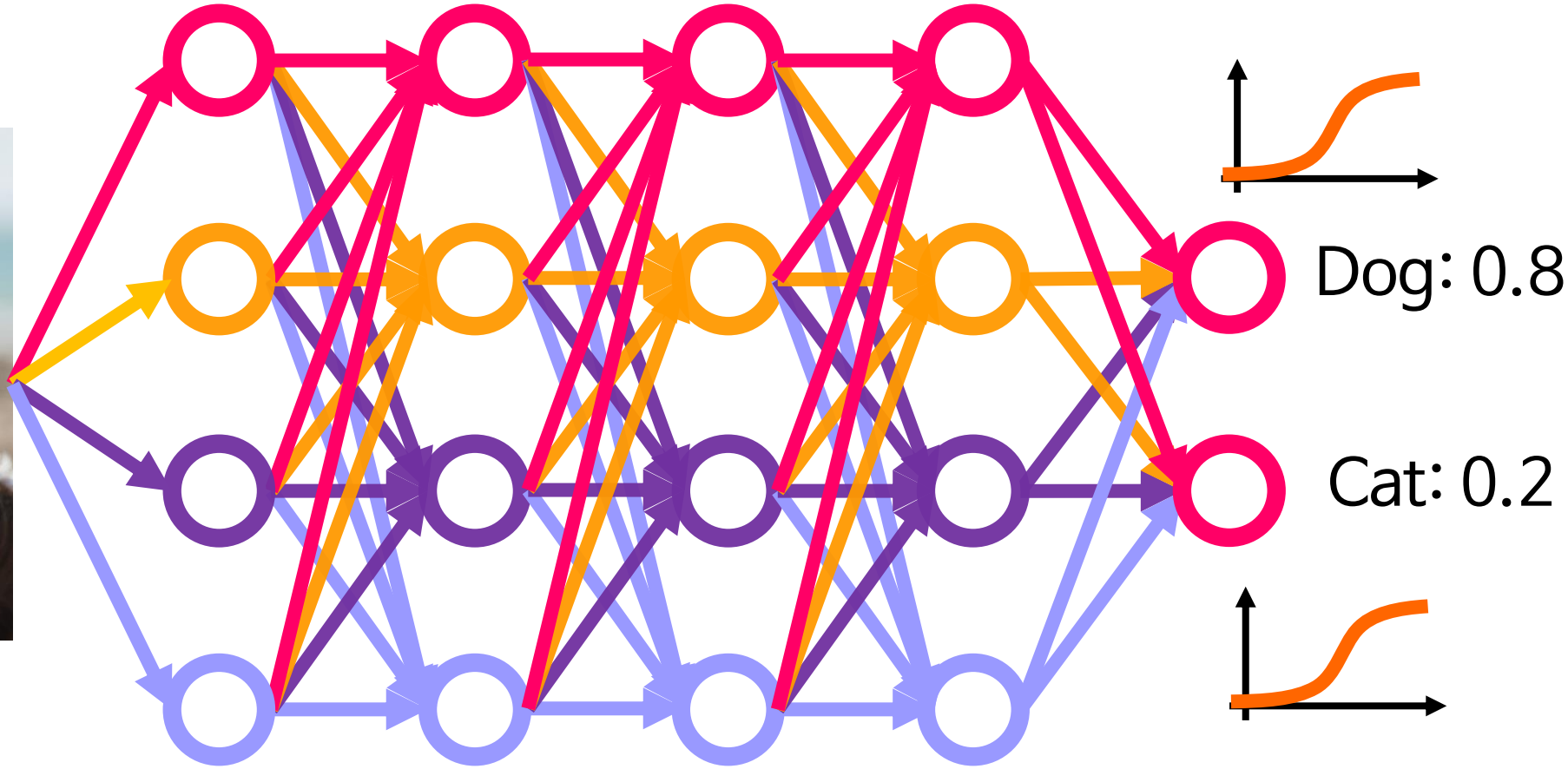
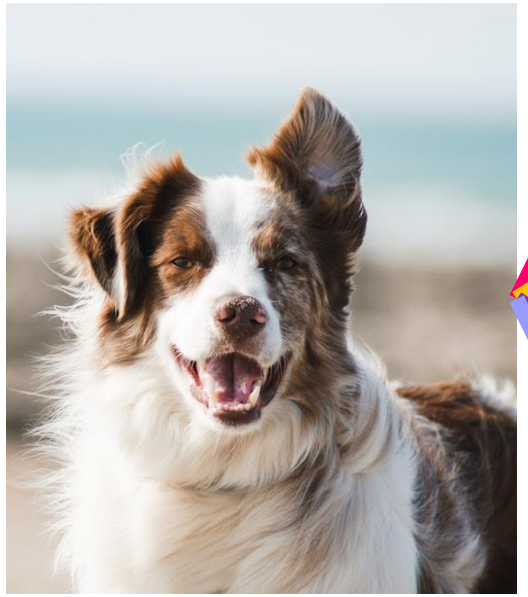


Photo by Pauline Loroy on Unsplash

신박AI

가장 중요한 부분은 신경망 출력값이 단순한 숫자가 아닌,



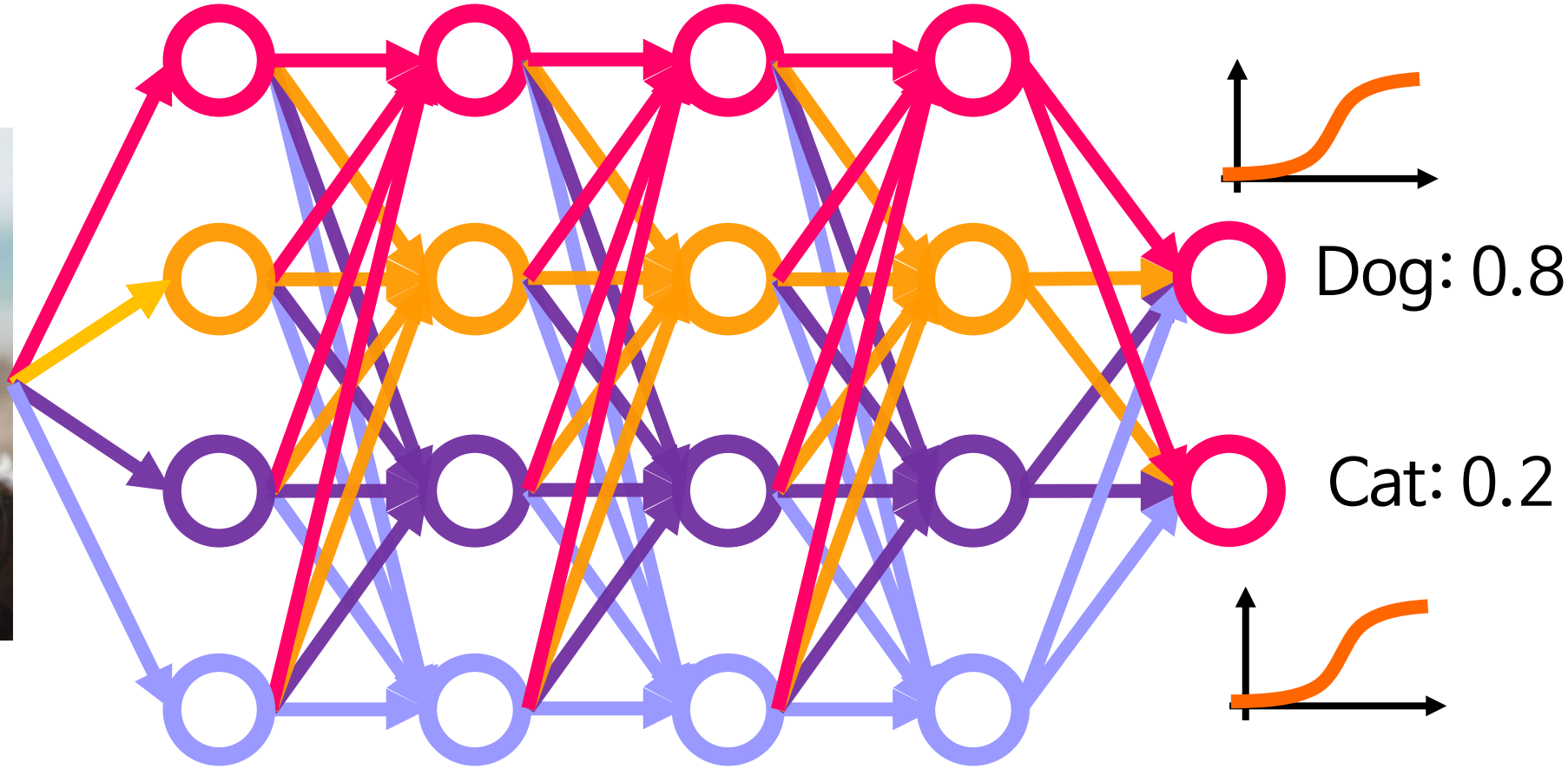
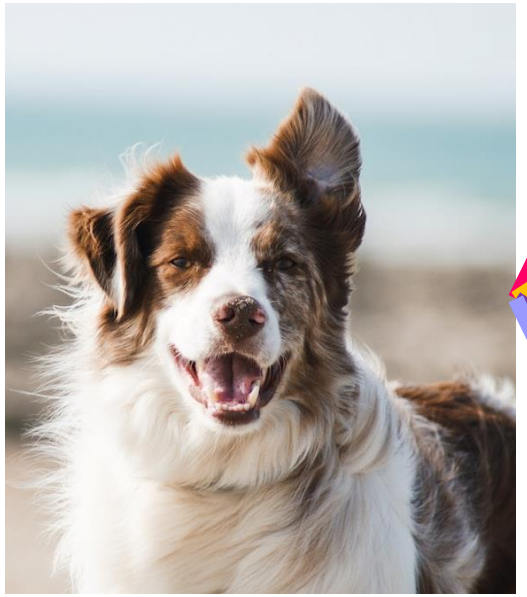


Photo by Pauline Loroy on Unsplash

신박AI

의미를 가진 숫자로 변환되기 때문입니다



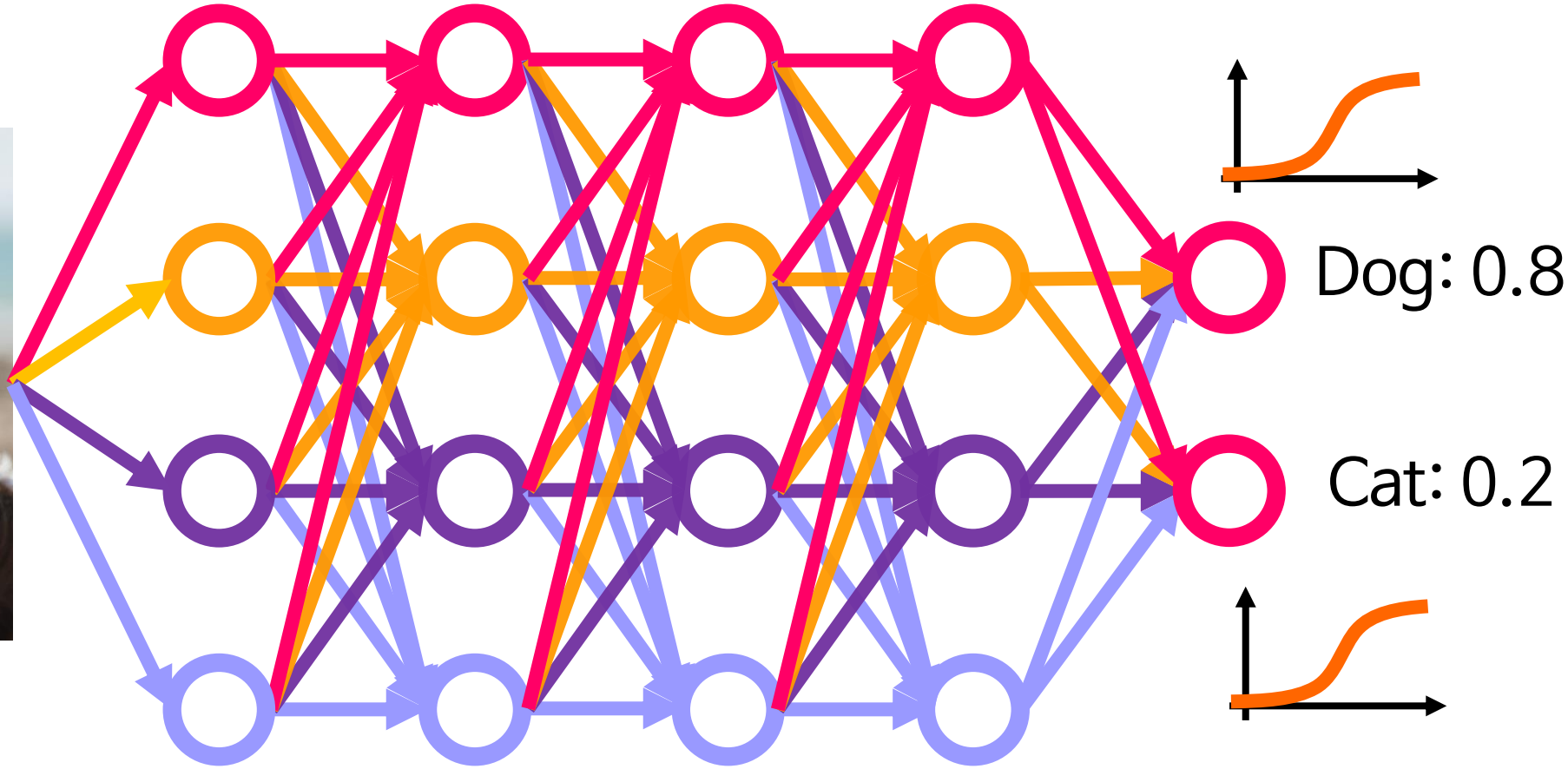


Photo by Pauline Loroy on Unsplash

신박AI

예를들어, 만약 누군가가 자신의 발 크기가 11이라고 한다면

Photo by Rachael Ren on Unsplash

신박AI

도대체 발이 큰건지 작은건지 알수가 없습니다

Photo by Rachael Ren on Unsplash

신박AI

그러나 발크기 11은 상위 95%라는 사실을 알게되면 (북미 사이즈 기준),

Photo by Rachael Ren on Unsplash

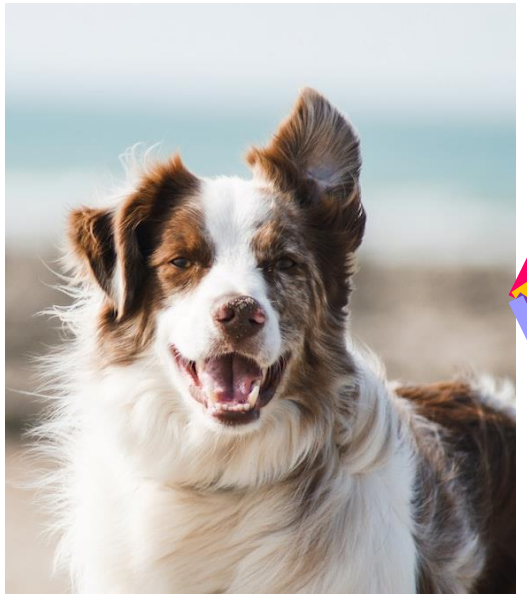
신박AI

비로소 그 사람의 발이 큰 것을 알게 되는것 처럼,

Photo by Rachael Ren on Unsplash

신박AI

신경망 출력값을 0과 1사이의 확률로 표현하는 것이



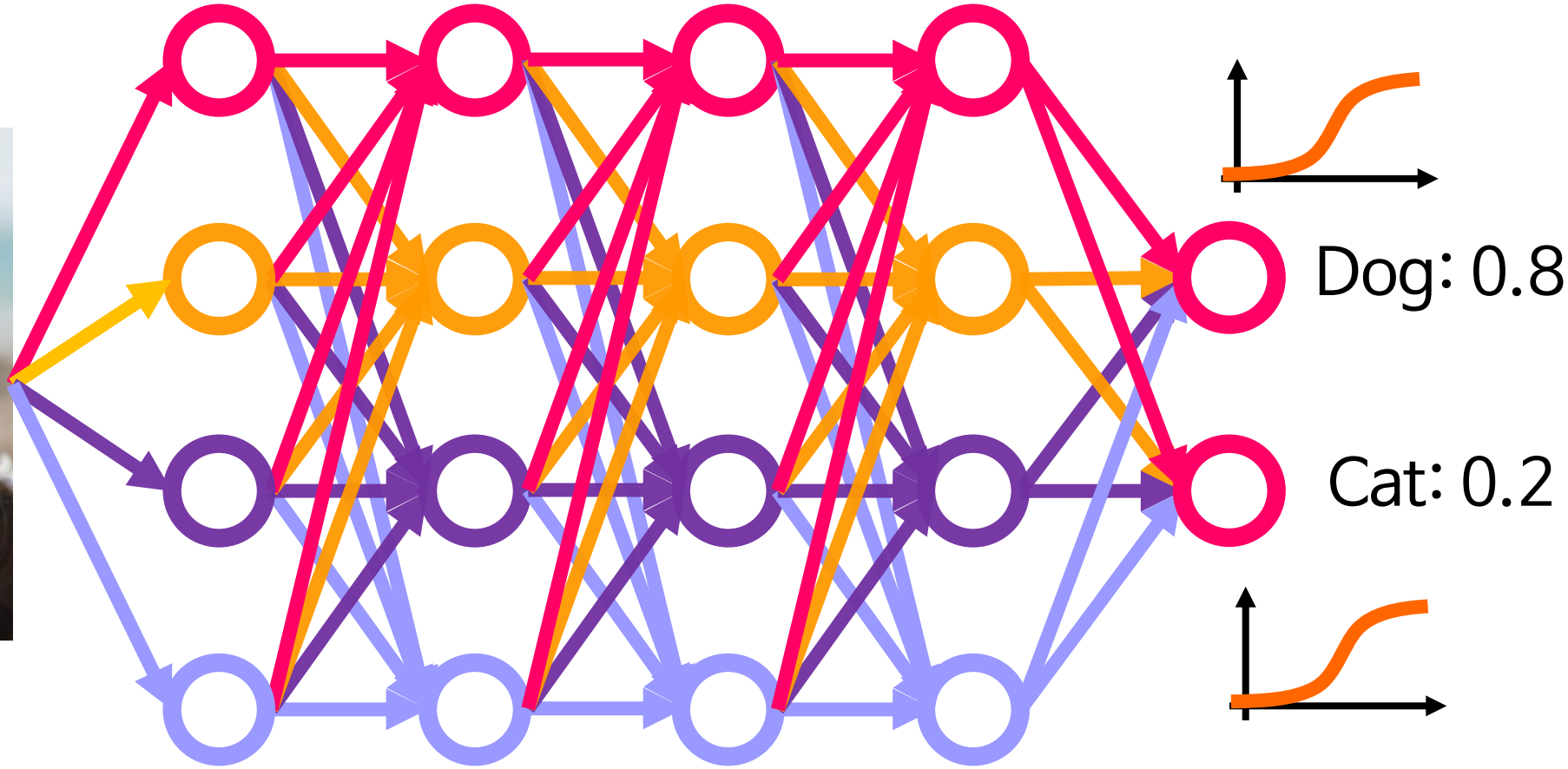
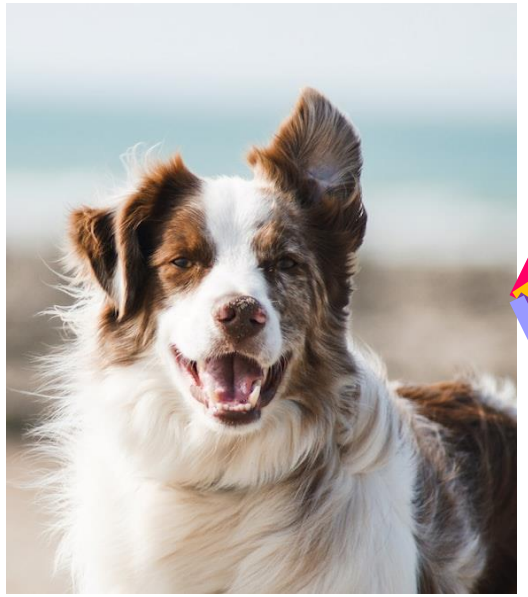


Photo by Pauline Loroy on Unsplash

신박AI

데이터(입력값)의 참의미를 더 잘 표현할 수 있게 됩니다



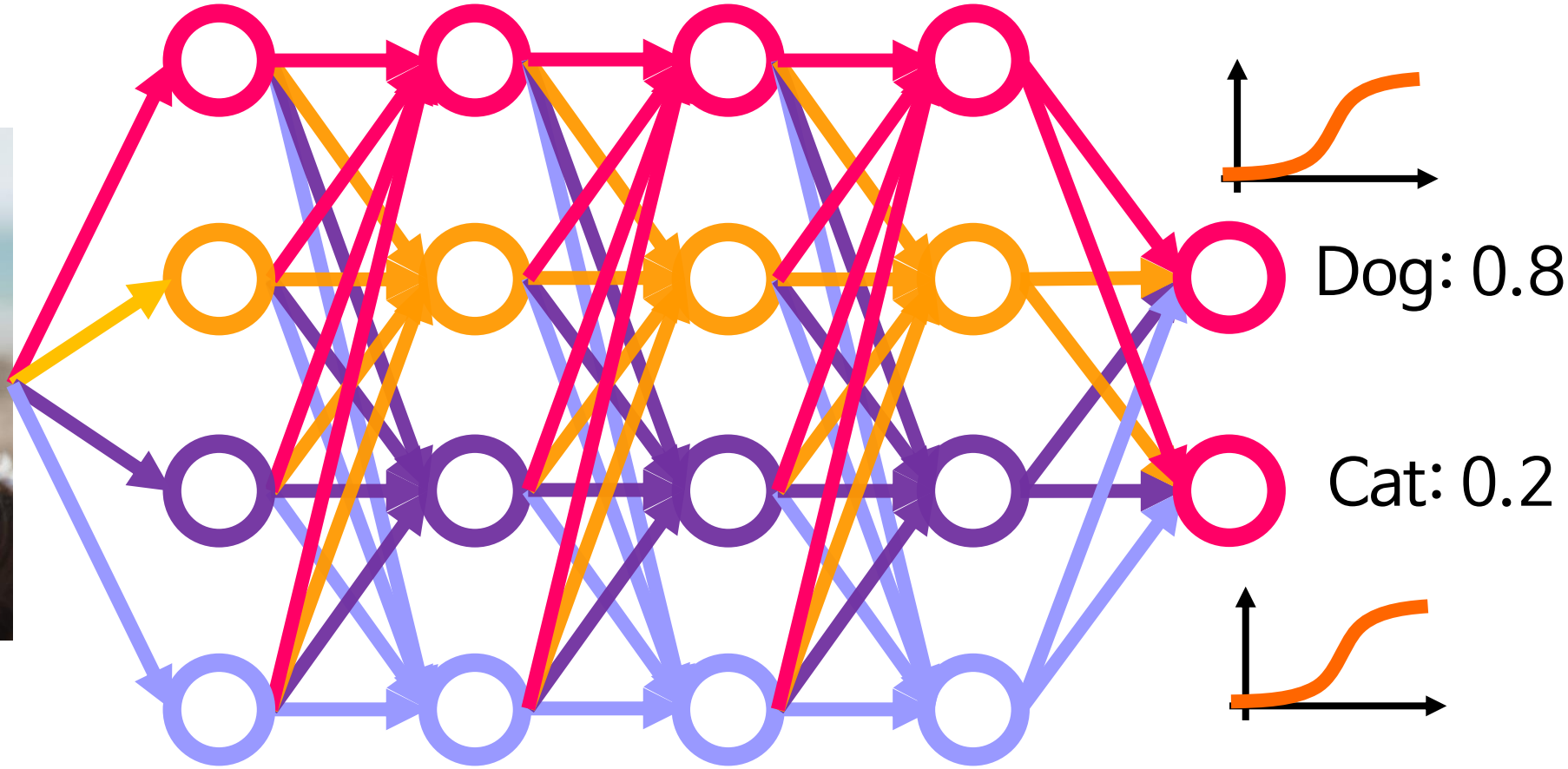
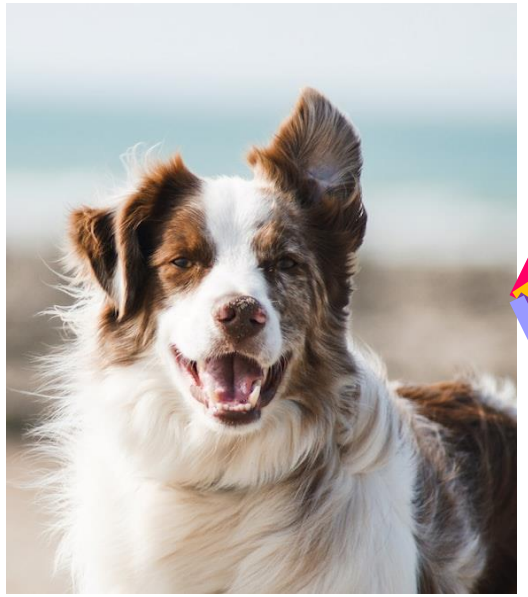


Photo by Pauline Loroy on Unsplash

신박AI

데이터(입력값)의 참의미를 더 잘 표현할 수 있게 됩니다



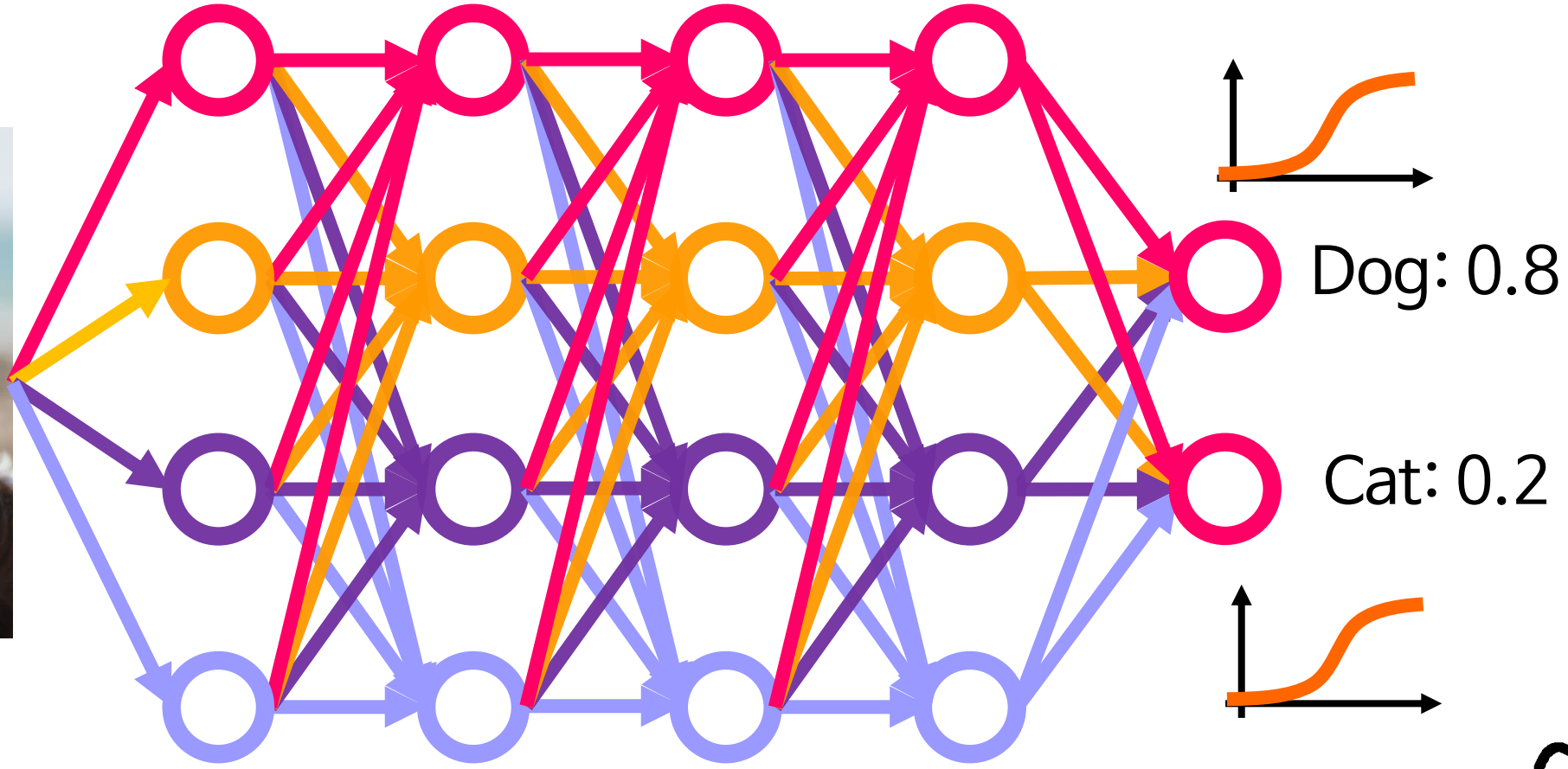
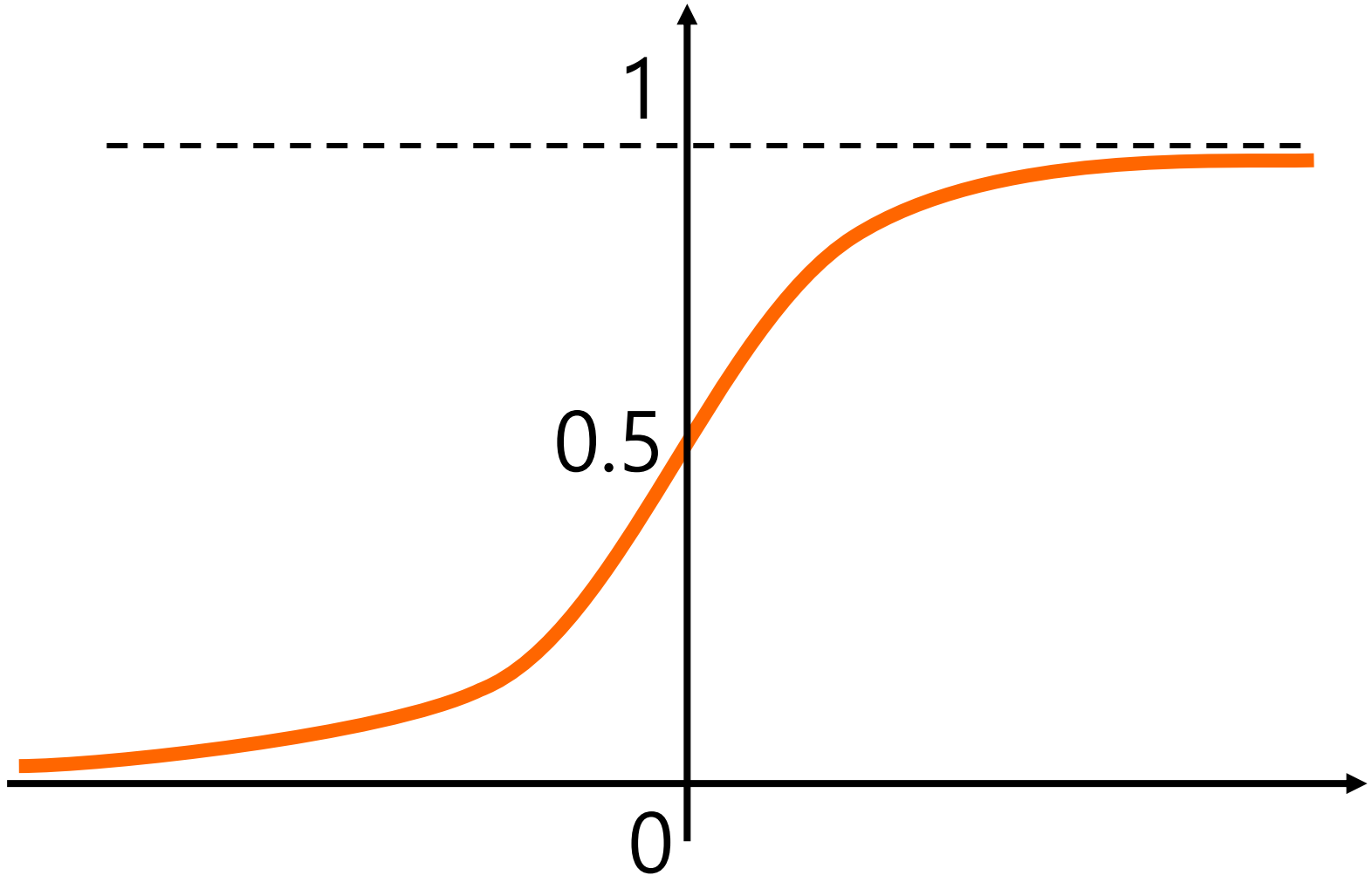
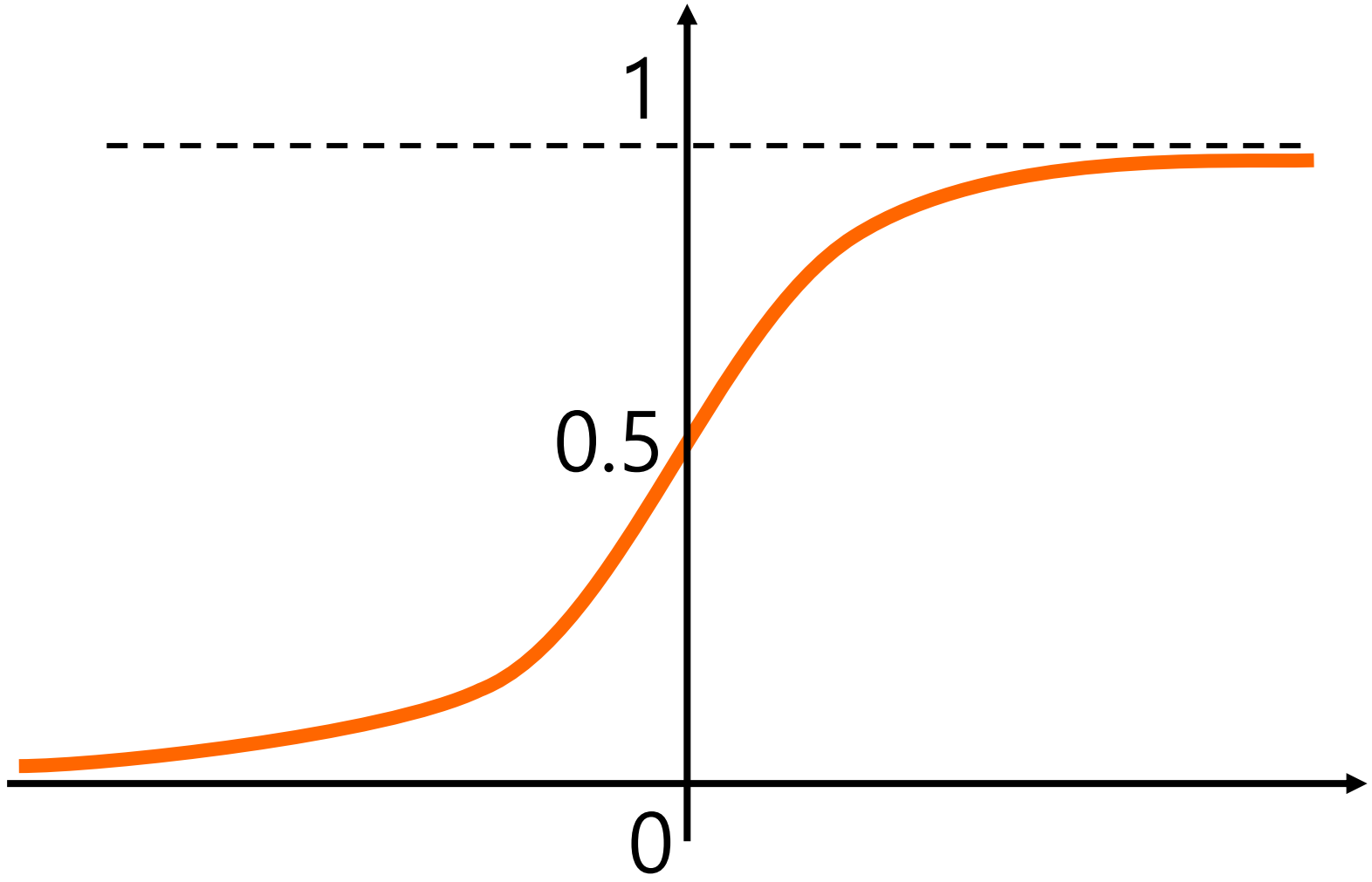


Photo by Pauline Loroy on Unsplash

하지만 이런 시그모이드 함수라도 완벽할 수는 없는데요,

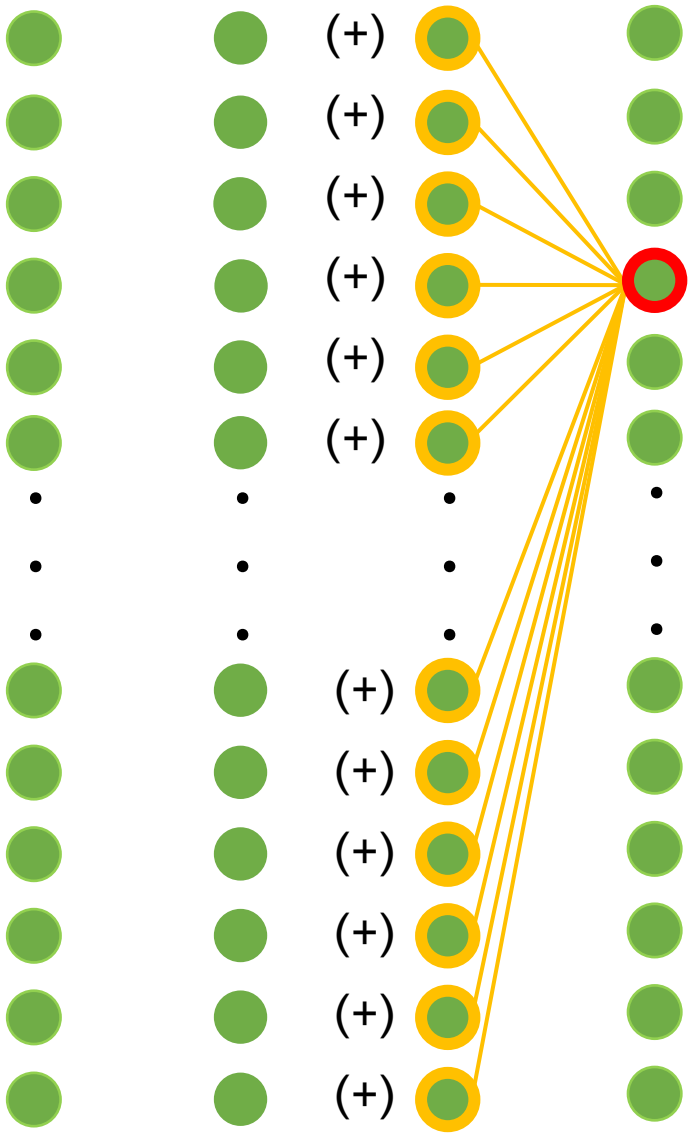


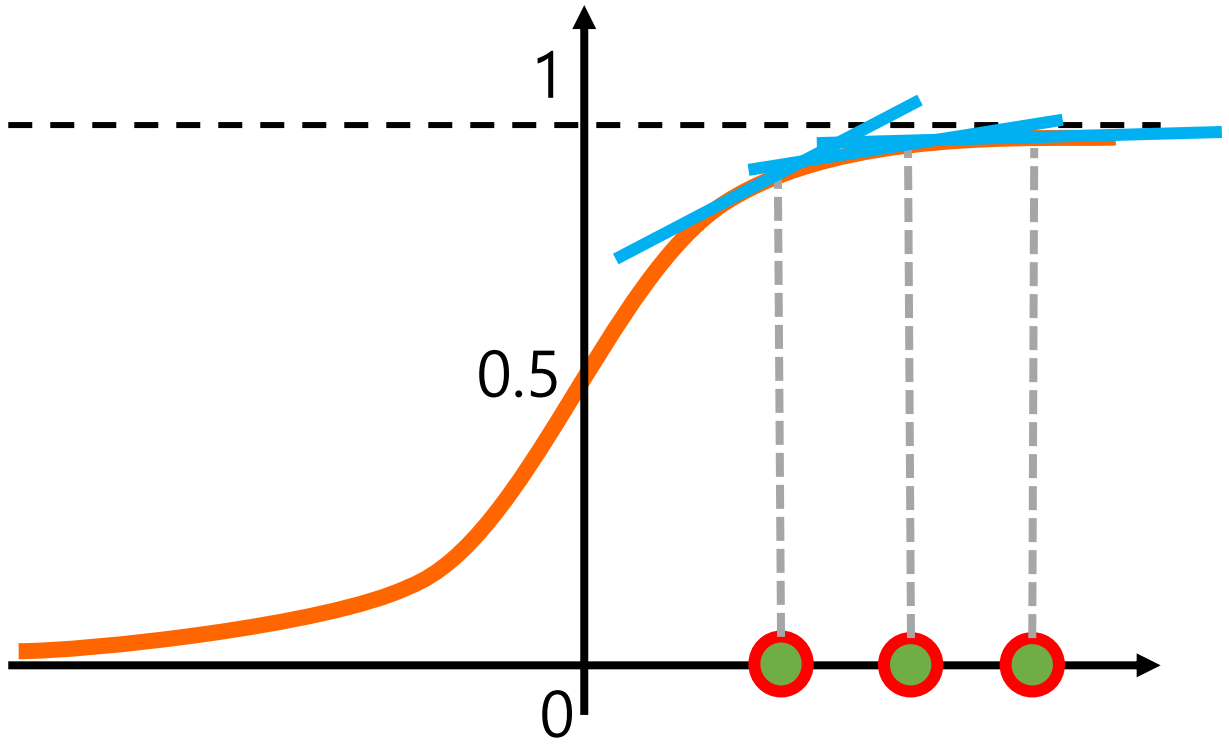
분명한 약점도 있습니다



신박AI

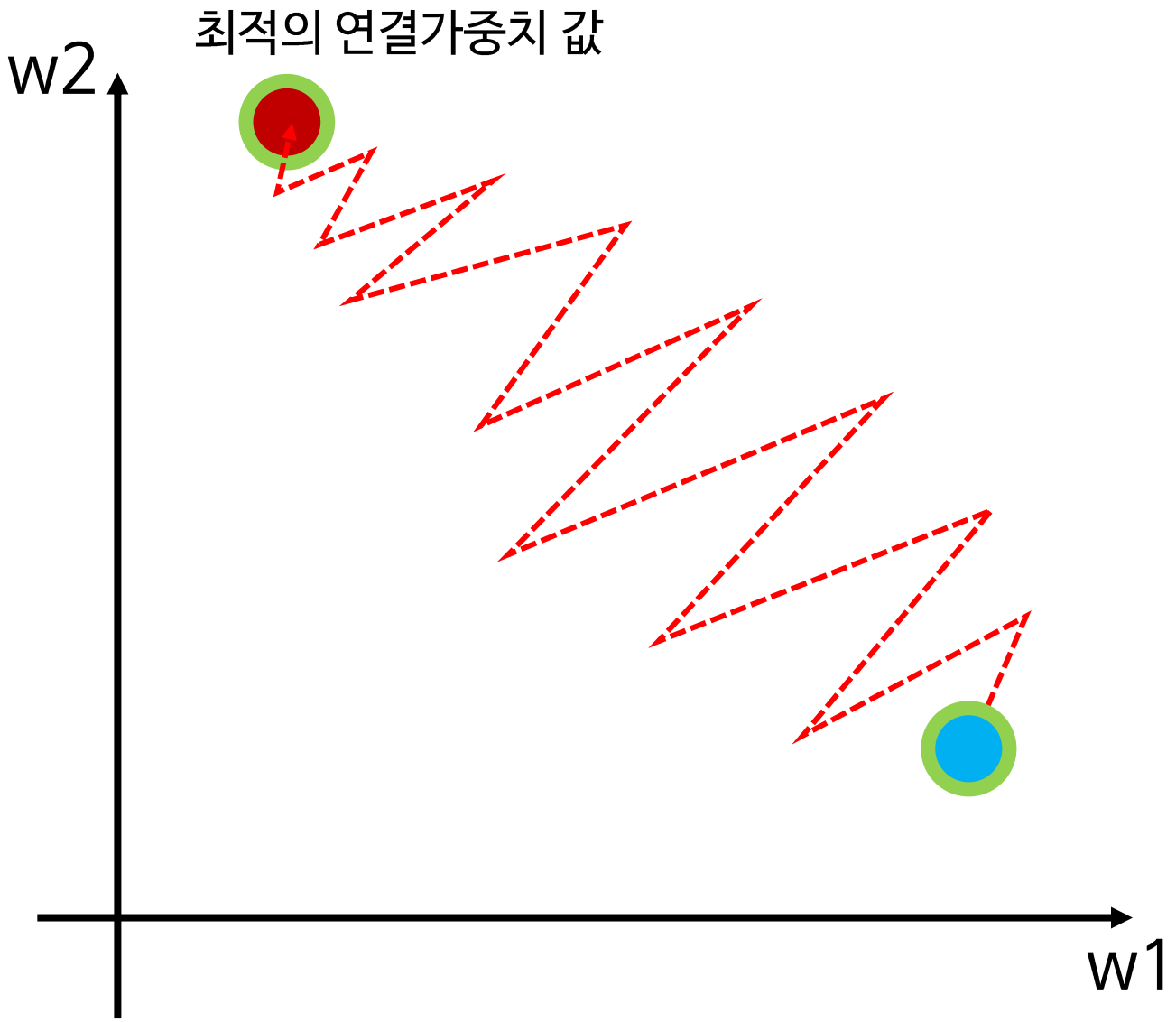
시그모이드 함수의 대표적인 약점으로는 기울기가 사라지는 현상과





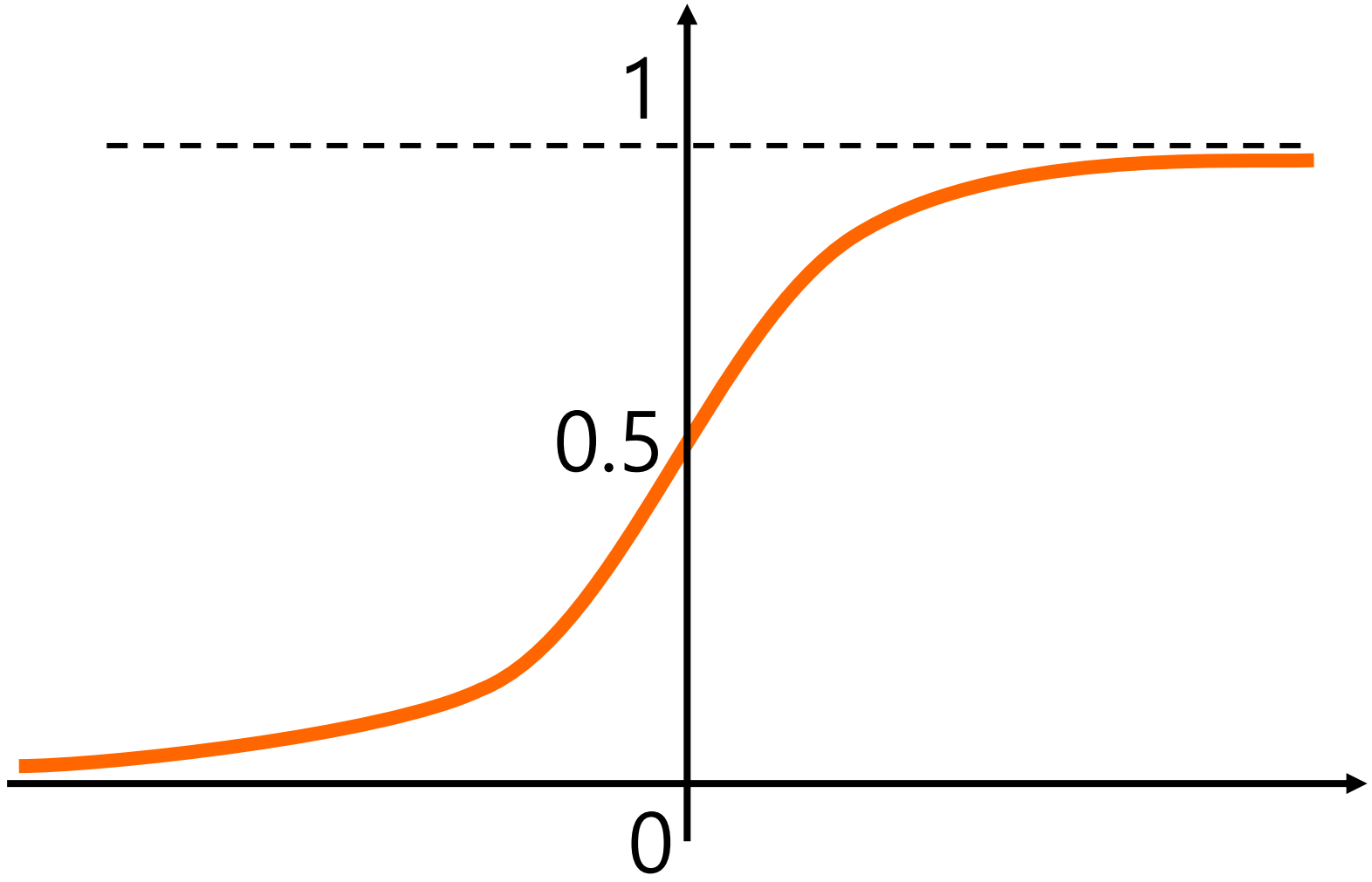
신박AI

학습 가중치 변화 중에 나타나는 지그재그 현상이 대표적인 약점입니다



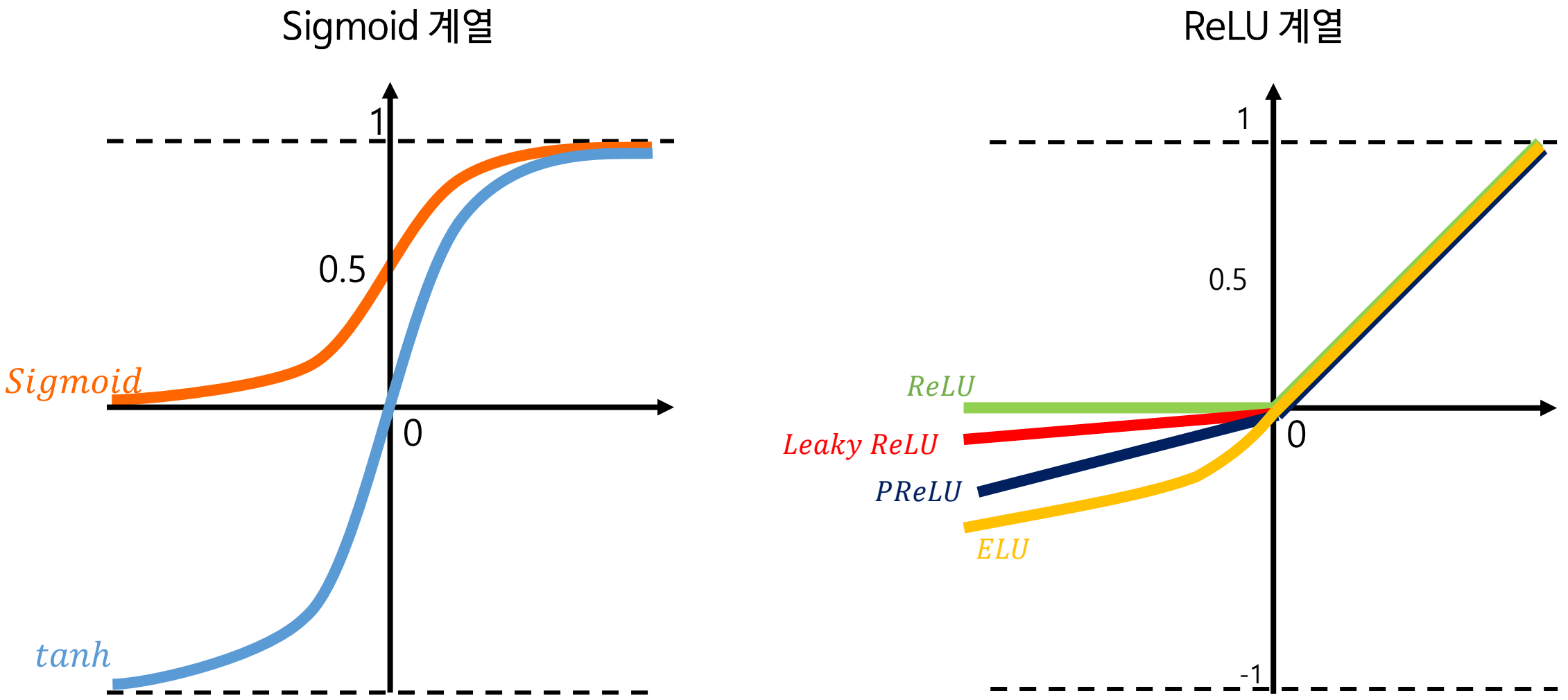
신박AI

다음 영상에서는 시그모이드 함수의 이러한 한계들을 자세히 살펴보고



신박AI

그 약점을 극복하는 여러 다른 활성화 함수를 소개해 드리겠습니다!



신박AI

감사합니다!

신박AI

이 영상은 여러분의 관심과 사랑으로 제작됩니다
사용하실때는 출처 '신박AI'를 밝혀주세요

신박AI

신박AI