



# 인공신경망과딥러닝심화

## Lecture 09. 오차 역전파에서 딥러닝으로

동덕여자대학교  
데이터사이언스 전공  
권 범

# 목차

- ❖ 01. 딥러닝의 태동, 오차 역전파
- ❖ 02. 활성화 함수와 고급 경사 하강법
- ❖ 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

# 시작하기 전에

## ❖ 오차 역전파와 딥러닝의 탄생 (1/2)

- 해결되지 않던 XOR 문제를 다층 퍼셉트론으로 해결
- 한 가지 문제를 만남

은닉층에 포함된 가중치를 업데이트할 방법이 없었던 것

- 
- 이는 기나긴 인공지능의 겨울을 지나 **오차 역전파**라는 방법을 만나고 나서야 해결

- 
- 오차 역전파는 후에 지금 우리가 아는 **딥러닝의 탄생**으로 이어짐

오차 역전파는 어떻게 해서  
은닉층의 오차를 업데이트할 수 있었을까?

# 시작하기 전에

## ❖ 오차 역전파와 딥러닝의 탄생 (2/2)

- 여기서 두 가지 길이 있음

### 첫 번째 길

- ✓ 오차 역전파의 개념을 이해하고 넘어가는 것
- ✓ 오차 역전파의 개념만 알아도 앞으로 배울 과정을 마치는 데는 아무런 문제가 없음
- ✓ 개념 숙지를 목표로 학습할 분은 이번 강의에 소개하는 내용만 익히면 됨

### 두 번째 길

- ✓ 개념과 함께 계산 방법까지 익히고 다음으로 넘어가는 것
- ✓ 딥러닝 알고리즘을 더 깊이 이해하고 싶거나,  
텐서플로 같은 라이브러리에 맡기지 않고 직접 코딩을 해야 한다면  
오차 역전파의 계산법을 숙지하기 바람

# 01. 딥러닝의 태동, 오차 역전파

02. 활성화 함수와 고급 경사 하강법

03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

# 01. 딥러닝의 태동, 오차 역전파

## ❖ 오차 역전파의 개념에 대해 알아보기 (1/7)

- 앞서 XOR 문제를 해결했지만, 입력 값과 출력 값을 알고 있는 상태에서 가중치  $w$ 와 바이어스  $b$ 를 미리 알아본 후 이를 집어넣었음

이것은 '모델링'이라고 할 수 없음



- 우리가 원하는 것은 데이터를 통해 스스로 가중치를 조절하는 **학습**의 실현  
오차 역전파 방법은 어떻게 해서 숨겨진 은닉층의 가중치를 업데이트할 수 있었을까?

이를 설명하기 위해  
다시 경사 하강법 이야기로 돌아가 보자

# 01. 딥러닝의 태동, 오차 역전파

## ❖ 오차 역전파의 개념에 대해 알아보기 (2/7)

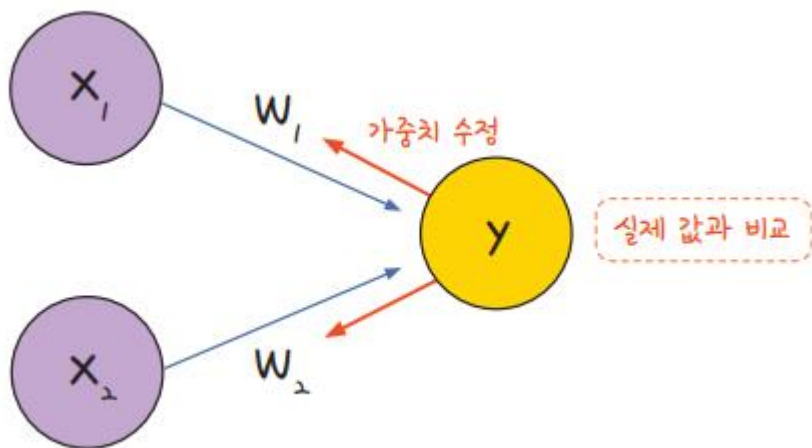
- 경사 하강법은 임의의 가중치를 선언하고  
결괏값을 이용해 오차를 구한 후  
이 오차가 최소인 지점으로 계속해서 조금씩 이동시키는 것  
이 오차가 최소인 지점은 미분했을 때 기울기가 0이 되는 지점이라고 했음
- 지금 이야기하고 있는 경사 하강법은 '단일 퍼셉트론',  
즉 입력층과 출력층만 존재할 때 가능

# 01. 딥러닝의 태동, 오차 역전파

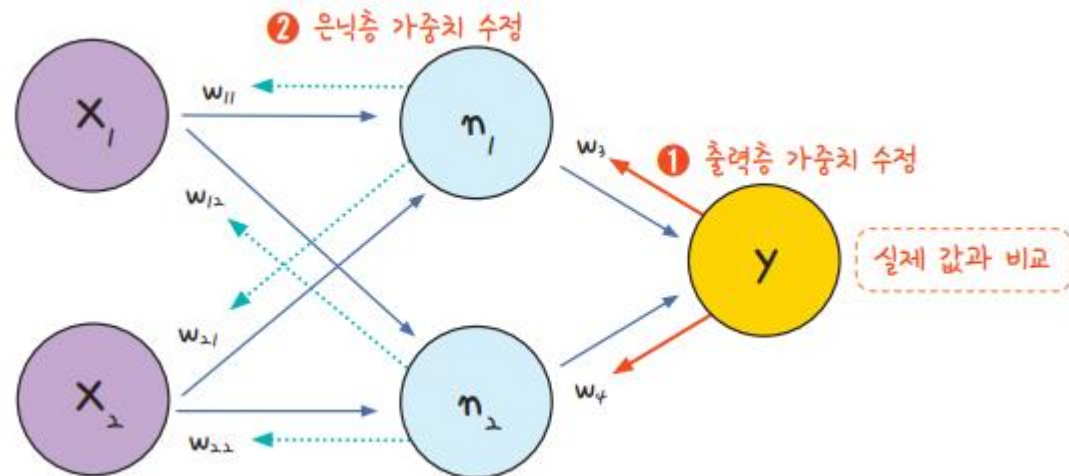
## ❖ 오차 역전파의 개념에 대해 알아보기 (3/7)

- 은닉층이 생기면서 우리는 두 번의 경사 하강법을 실행해야 함
- 즉, 그림 (a)와 같이 가중치를 한 번 수정하면 되던 작업이, 오른쪽 그림 (b)와 같이 가중치를 두 번 수정해야 하는 것

(a) 단일 퍼셉트론에서 오차 수정



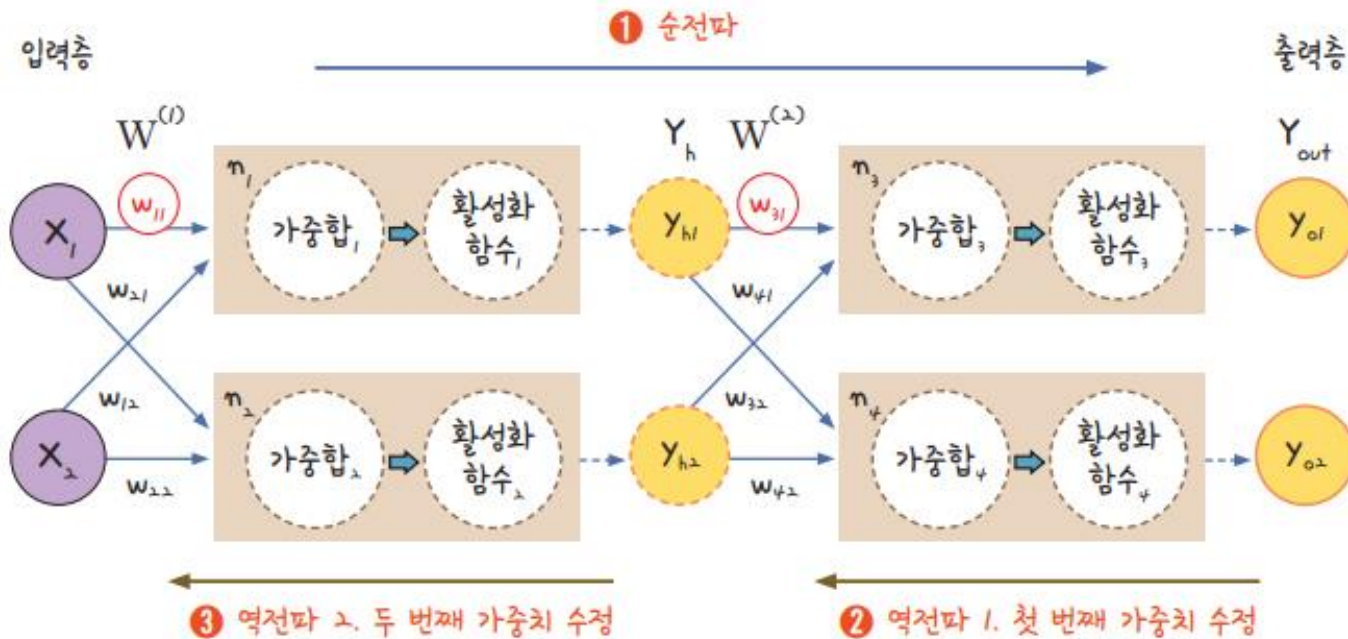
(b) 다층 퍼셉트론에서 오차 수정





# 01. 딥러닝의 태동, 오차 역전파

## ❖ 오차 역전파의 개념에 대해 알아보기 (4/7)



오차 역전파의 개념

- ✓ 먼저 ①처럼 한 번의 순전파가 일어남
- ✓ 이 과정을 통해 각 가중치의 초깃값이 정해짐
- ✓ 이 초깃값의 가중치로 만들어진 값과 실제 값을 비교해 출력층의 오차를 계산
- ✓ 목표는 이때 계산된 출력층의 오차를 최소화시키는 것
- ✓ 이를 위해 ② 첫 번째 가중치를 수정하는 과정과 ③ 두 번째 가중치를 수정하는 과정이 이어짐

# 01. 딥러닝의 태동, 오차 역전파

## ❖ 오차 역전파의 개념에 대해 알아보기 (5/7)

- 예를 들어, 첫 번째 가중치 중 하나인  $w_{31}$ 을 업데이트한다고 하자
- 경사 하강법에서 배운 대로  $w_{31}$ 을 업데이트하기 위해서는 오차 공식을 구하고  $w_{31}$ 에 대해 편미분해야 함
- 합성 함수의 미분이므로 체인 룰을 적용해 편미분을 구하고, 이를 이용해  $w_{31}$ 을 업데이트



- 이제 두 번째 가중치를 업데이트할 차례
- 예를 들어,  $w_{11}$ 을 업데이트한다고 하면 마찬가지로 오차 공식을 구하고  $w_{11}$ 에 대해 편미분하면 됨 그런데 여기서 문제가 생김
- 앞에서는 출력층의 결과와 실제 값을 비교해 오차를 얻었음
  - ✓ 은닉층은 겉으로 드러나지 않으므로 그 값을 알 수 없음
  - ✓ 그리고 오차를 구할 만한 적절한 출력 값도 없음

# 01. 딥러닝의 태동, 오차 역전파

## ❖ 오차 역전파의 개념에 대해 알아보기 (6/7)

- 이 문제는 다시 출력층의 오차를 이용하는 것으로 해결
- $w_{31}$ 의 경우  $y_{o1}$ 의 오차만 필요했지만,  $w_{11}$ 은  $y_{o1}$ 과  $y_{o2}$ 가 모두 관여되어 있음
- 오차 두 개를 모두 계산해 이를 편미분하게 됨
- 물론 계산식은 조금 더 복잡해짐
- 이 과정을 마치면 첫 번째 가중치 업데이트 공식과 두 번째 가중치 업데이트 공식이 다음과 같이 정리

**첫 번째 가중치 업데이트 공식:**  $(y_{o1} - y_{real}) \times y_{o1}(1 - y_{o1}) \times y_{h1}$

**두 번째 가중치 업데이트 공식:**  $(\delta y_{o1}w_{31} + \delta y_{o2}w_{41}) \times y_{h1}(1 - y_{h1}) \times x_1$

# 01. 딥러닝의 태동, 오차 역전파

## ❖ 오차 역전파의 개념에 대해 알아보기 (7/7)

첫 번째 가중치 업데이트 공식:  $(y_{o1} - y_{real}) \times y_{o1}(1 - y_{o1}) \times y_{h1}$

두 번째 가중치 업데이트 공식:  $(\delta y_{o1} w_{31} + \delta y_{o2} w_{41}) \times y_{h1}(1 - y_{h1}) \times x_1$

- 위 공식에서 밑줄 친 부분을 잘 보면 두 식 모두 'out(1-out)' 형태를 취하고 있음 이를 델타식이라고 함
- 은닉층의 숫자가 늘어도 이러한 형태가 계속해서 나타나게 되므로, 이를 이용해 깊은 층의 계산도 할 수 있게 됨
- 이렇게 깊은 층을 통해 학습할 수 있는 계기가 마련되면서 드디어 **딥러닝**이 태동하게 된 것
- 우리는 텐서플로를 이용한 프로그래밍을 진행하므로 더 이상 자세한 설명은 생략함

## 02. 활성화 함수와 고급 경사 하강법

01. 딥러닝의 태동, 오차 역전파

03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

## 02. 활성화 함수와 고급 경사 하강법

### ❖ 기울기 소실 문제 (1/3)

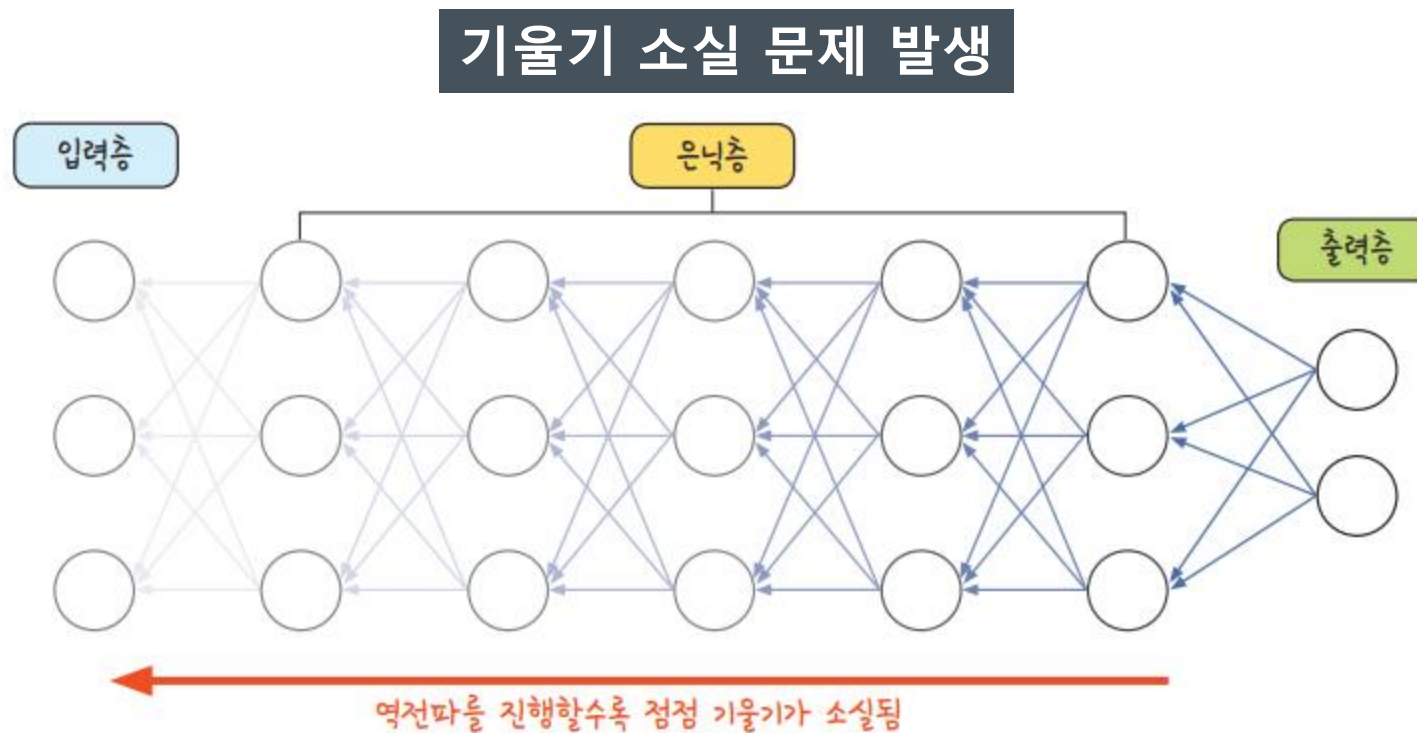
- 앞서 델타식을 이용해 깊은 신경망의 계산이 가능해졌음을 이야기했음
- 이제 수많은 층을 연결해 학습하면, 여러 난제를 해결하는 인공지능이 완성될 것 같아 보임
- 아직 한 가지 문제가 더 남아 있음

“  
기울기 소실 문제  
(Vanishing Gradient Problem)”

## 02. 활성화 함수와 고급 경사 하강법

### ❖ 기울기 소실 문제 (2/3)

- 아래 그림과 같이 깊은 층을 만들어 보니 출력층에서 시작된 가중치 업데이트가 처음 층까지 전달되지 않는 현상이 생기는 문제가 발견
- 이는 활성화 함수로 사용된 시그모이드 함수의 특성 때문임

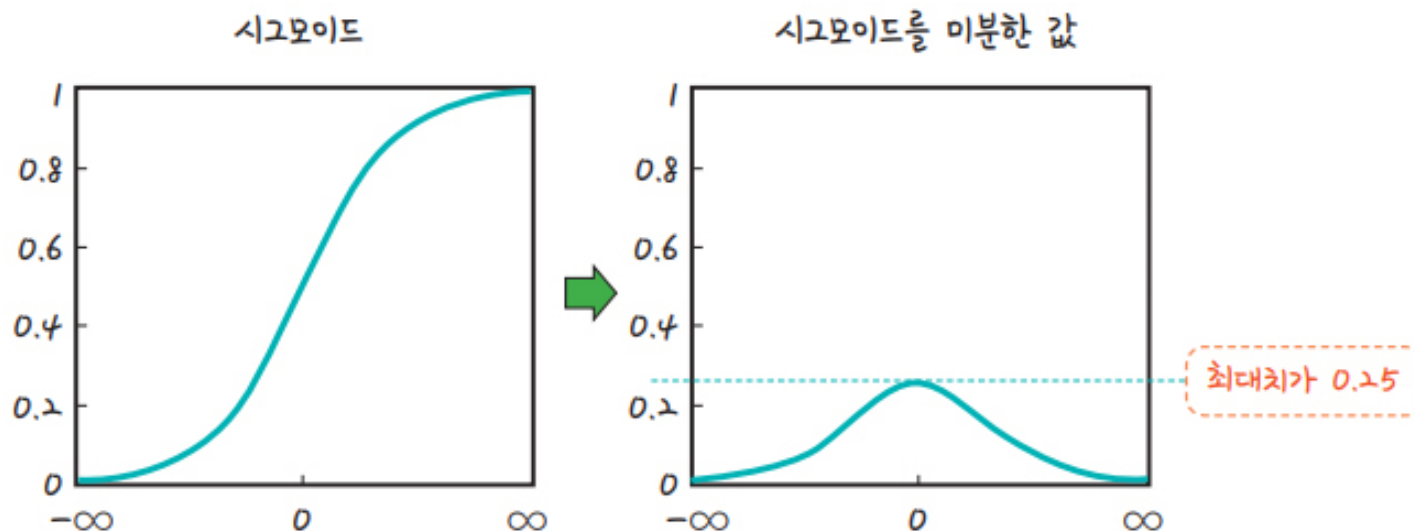


## 02. 활성화 함수와 고급 경사 하강법

### ❖ 기울기 소실 문제 (3/3)

- 아래 그림과 같이 시그모이드 함수를 미분하면 최대치는 0.25
- 1보다 작으므로 계속 곱하다 보면 0에 가까워짐
- 여러 층을 거칠수록 기울기가 사라져 가중치를 수정하기 어려워지는 것

### 시그모이드의 미분



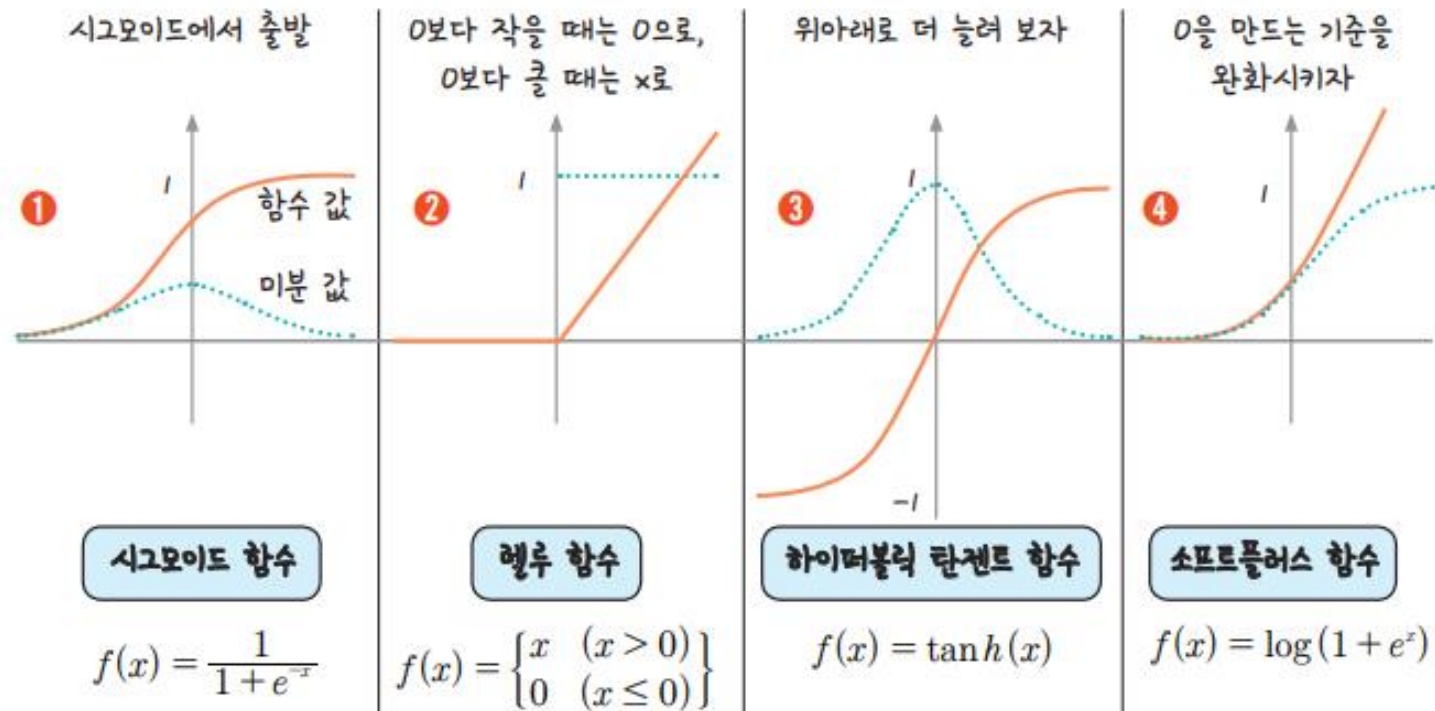


## 02. 활성화 함수와 고급 경사 하강법

### ❖ 렐루 함수 (1/3)

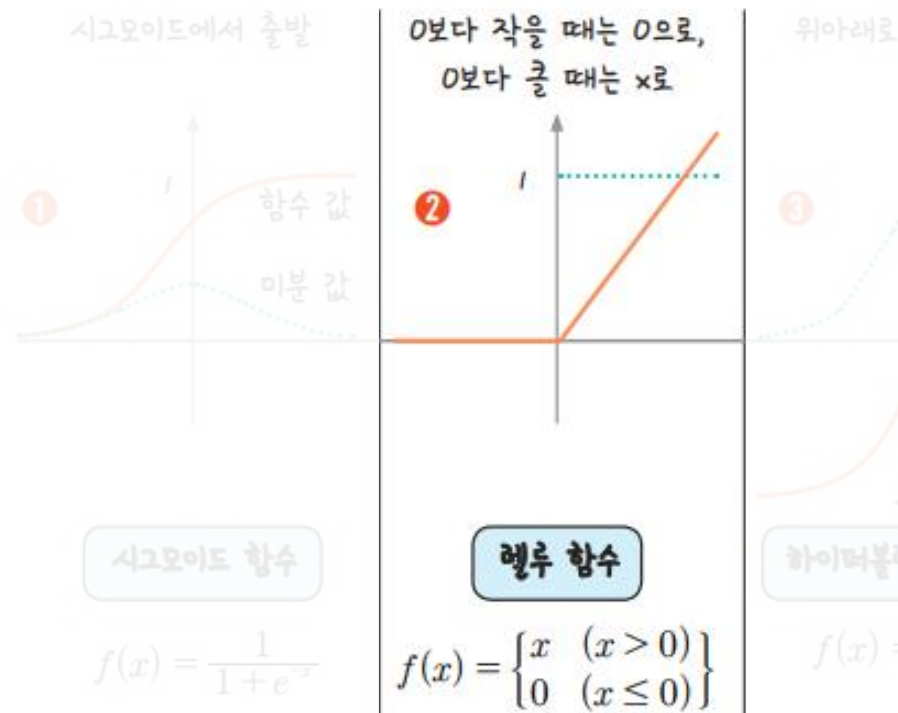
- 이를 해결하고자 제프리 힌튼 교수는 **렐루(ReLU)**라는 새로운 활성화 함수를 제안

#### 여러 활성화 함수의 도입



## 02. 활성화 함수와 고급 경사 하강법

### ❖ 렐루 함수 (2/3)



- ✓ ②의 렐루는  $x$ 가 0보다 작을 때 출력 값을 0으로 처리하고, 0보다 큰 값은  $x$ 를 그대로 사용하는 방법
- ✓ 파란색 점선이 미분을 한 결과인데, 그림에서 보이듯  $x$ 가 0보다 크기만 하면 미분 값은 1이 됨
- ✓ 활성화 함수로 렐루를 쓰면 여러 번 오차 역전파가 진행되어도 맨 처음 층까지 값이 남아 있게 됨

- ✓ 학습은 결국 오차를 최소화하는 가중치를 찾는 과정
- ✓ 출력층에서 알아낸 오차가 역전파를 통해 입력층까지 거슬러 올라가면서 잘못된 가중치들을 수정할 수 있게 되자, 더 깊은 층을 쌓아 올리는 것이 가능해졌음

## 02. 활성화 함수와 고급 경사 하강법

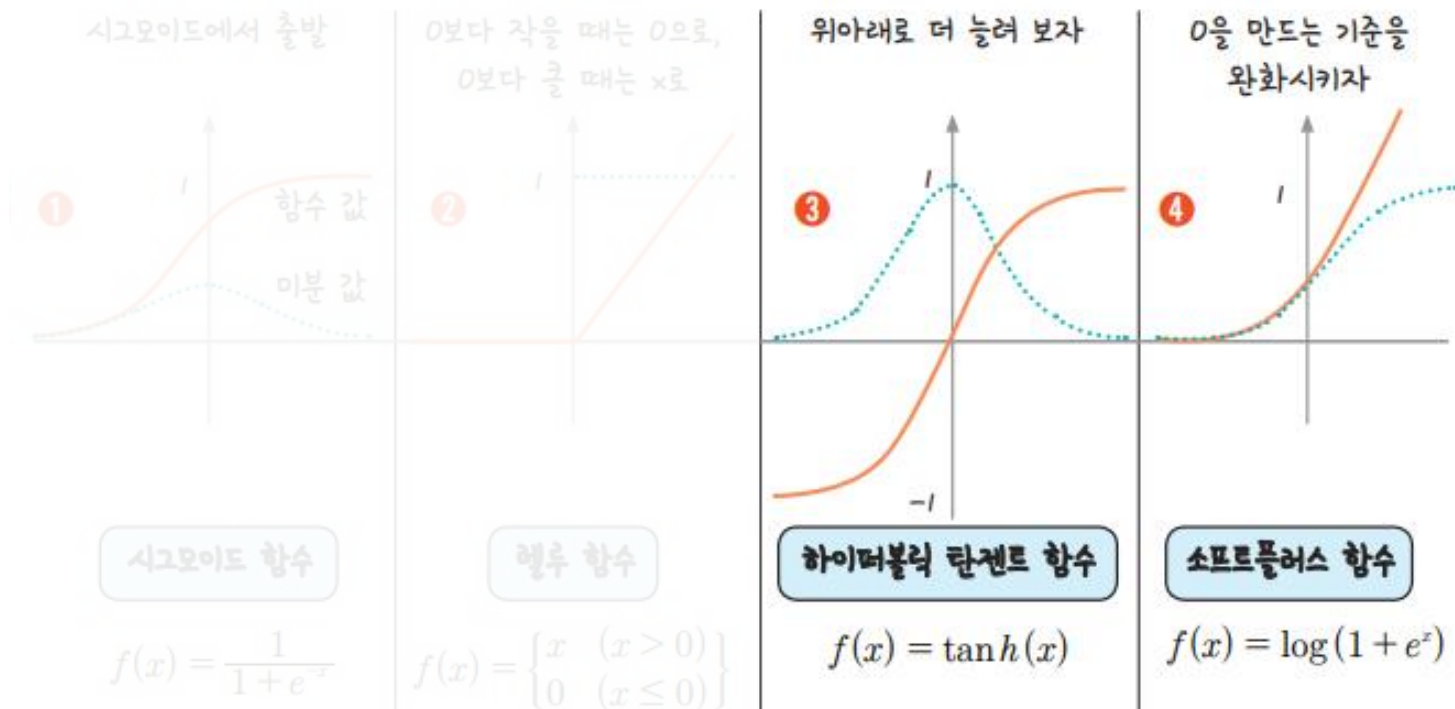
### ❖ 렐루 함수 (3/3)

- 활성화 함수는 그 이후로도 여러 데이터 과학자에 의해 연구되어

③ 하이퍼볼릭 탄젠트(Hyperbolic Tangent) 함수나

④ 소프트플러스(Softplus) 함수 등

좀 더 나은 활성화 함수를 만들기 위한 노력이 이어지고 있음



## 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

- 01. 딥러닝의 태동, 오차 역전파
- 02. 활성화 함수와 고급 경사 하강법
- 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

## 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

### ❖ 고급 경사 하강법의 등장

- 우리는 가중치를 업데이트하는 방법으로 경사 하강법을 배웠음
- 경사 하강법은 정확하게 가중치를 찾아가지만, 계산량이 매우 많다는 단점이 있음
- 이러한 점을 보완한 **고급 경사 하강법이 등장**하면서  
딥러닝의 발전 속도는 더 빨라졌음

## 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

### ❖ 확률적 경사 하강법(Stochastic Gradient Descent, SGD) (1/2)

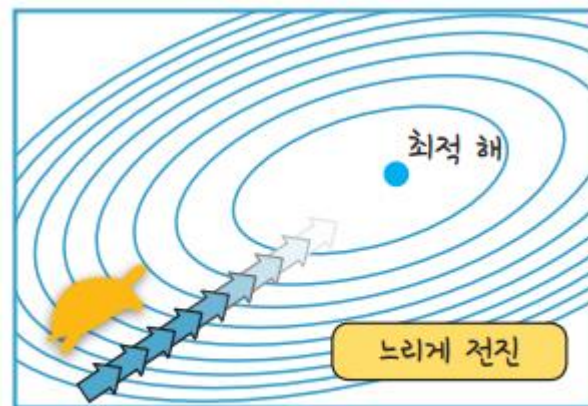
- 경사 하강법은 한 번 업데이트할 때마다  
전체 데이터를 미분하므로 속도가 느릴 뿐 아니라,  
최적 해를 찾기 전에 최적화 과정이 멈출 수도 있음  
**확률적 경사 하강법**은 경사 하강법의 이러한 단점을 보완한 방법
- 전체 데이터를 사용하는 것이 아니라,  
**랜덤하게 추출한 일부 데이터만 사용**하기 때문에  
빠르고 더 자주 업데이트할 수 있다는 장점이 있음

# 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

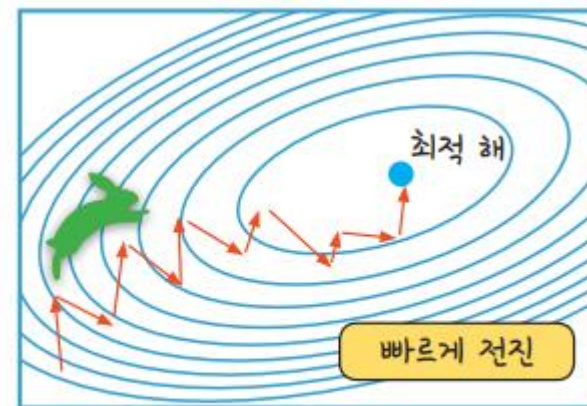
## ❖ 확률적 경사 하강법(Stochastic Gradient Descent, SGD) (2/2)

- 아래 그림은 경사 하강법과 확률적 경사 하강법의 차이를 보여 줌
- 랜덤한 일부 데이터를 사용하는 만큼 확률적 경사 하강법은  
중간 결과의 진폭이 크고 불안정해 보일 수도 있음
- 속도가 확연히 빠르면서도 최적 해에 근사한 값을 찾아낸다는 장점 덕분에  
경사 하강법의 대안으로 사용되고 있음

### 경사 하강법과 확률적 경사 하강법의 비교



경사 하강법



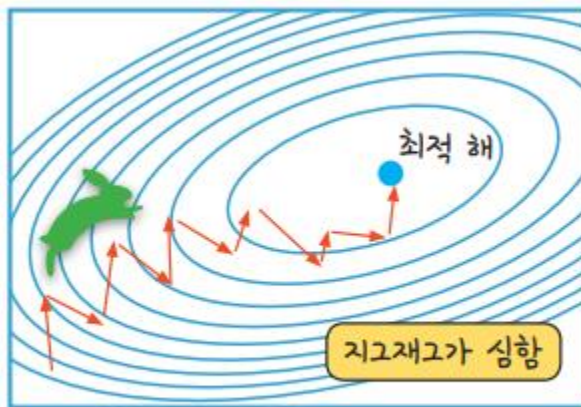
확률적 경사 하강법

# 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

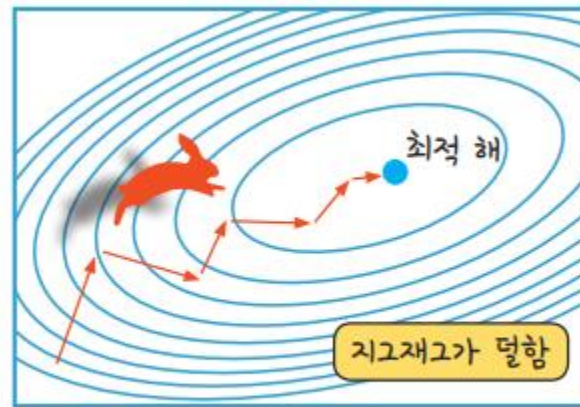
## ❖ 모멘텀 확률적 경사 하강법(Momentum SGD)

- 모멘텀이란 단어는 '관성, 탄력, 가속도'라는 뜻
- 모멘텀 확률적 경사 하강법(모멘텀 SGD)이란 말 그대로 경사 하강법에 탄력을 더해 주는 것
- 다시 말해 경사 하강법과 마찬가지로 매번 기울기를 구하지만, 이를 통해 오차를 수정하기 전 바로 앞 수정 값과 방향(+, -)을 참고해 같은 방향으로 일정한 비율만 수정되게 하는 방법

### 모멘텀을 적용했을 때



확률적 경사 하강법



모멘텀을 적용한 확률적 경사 하강법

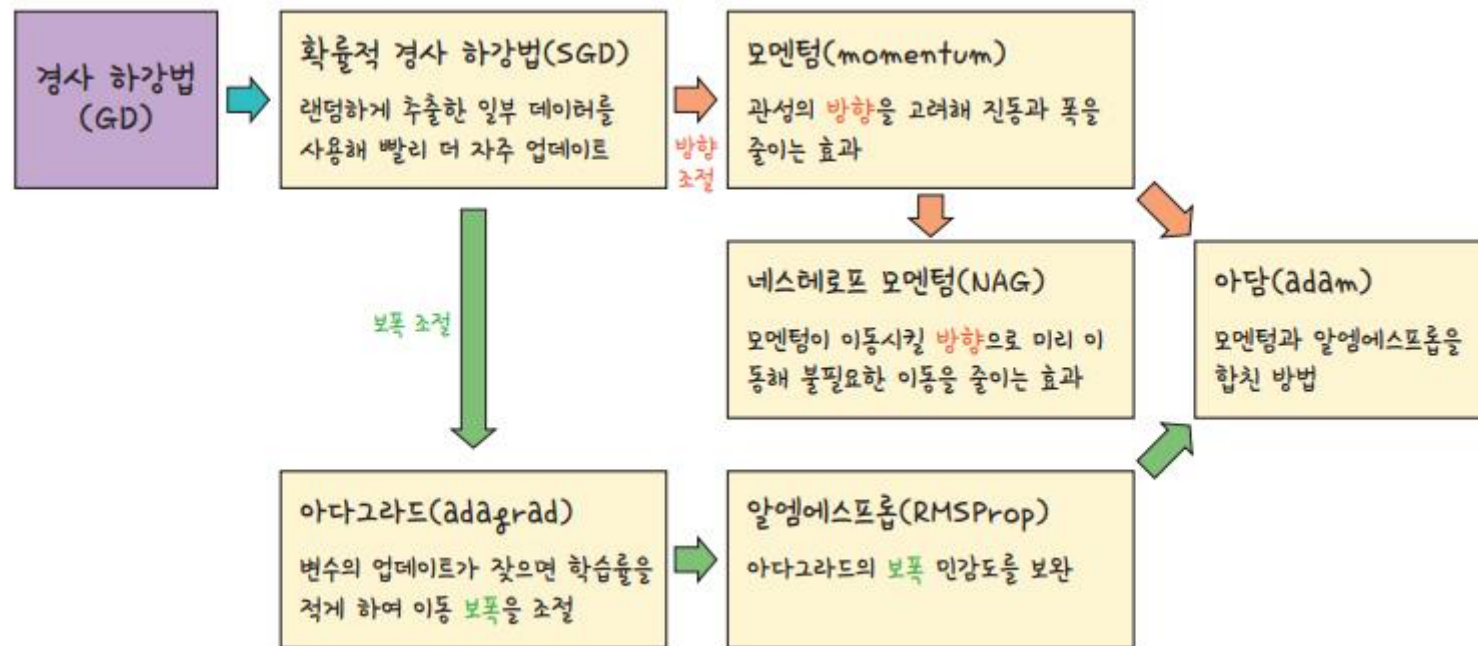
수정 방향이 양수(+) 방향으로 한 번,  
음수(-) 방향으로 한 번 지그재그로  
일어나는 현상이 줄어들고,  
이전 이동 값을 고려해 일정 비율만큼  
다음 값을 결정하므로 관성 효과를 낼 수 있음



# 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

## ❖ 딥러닝에서 사용되는 고급 경사 하강법의 변천

- 이 밖에도 딥러닝의 학습을 더 빠르고 정확하게 만들기 위한 노력이 계속되었음
- 지금은 정확도와 속도를 모두 향상시킨  
아담(Adam)이라는 고급 경사 하강법이 가장 많이 쓰이고 있음
- 아래 그림은 경사 하강법이 어떻게 해서 아담에 이르게 되었는지 보여 줌



## 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

### ❖ 경사 하강법과 옵티마이저

- 이렇게 오차를 최소화하는 경사 하강법들을  
딥러닝에서는 **옵티마이저**라고 한다고 했음
  - 앞에 소개한 고급 경사 하강법들은 텐서플로에 포함되어 있는  
**optimizer**라는 객체에 이름을 적어 주는 것만으로 손쉽게 실행할 수 있음
  - 또 시그모이드, 렐루 등 **활성화 함수**도  
**activation**이라는 객체에 이름을 적어 주는 것으로 손쉽게 실행할 수 있음
- 이어지는 수업에서 이러한 부분을 실습할 예정

텐서플로에서 사용되는 대부분의 개념과 용어를 상세히 배웠음  
드디어 텐서플로를 이용한 모델링을 할 준비가 되었음

- ❖ 01. 딥러닝의 태동, 오차 역전파
- ❖ 02. 활성화 함수와 고급 경사 하강법
- ❖ 03. 속도와 정확도 문제를 해결하는 고급 경사 하강법

# THANK YOU!

## Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: [bkwon@dongduk.ac.kr](mailto:bkwon@dongduk.ac.kr)