

4회차 퍼셉트론과 역전파

SMARCLE 2022Winter.AI.study



4팀 이용빈, 최원서, 김용현, 유혁재, 유정수

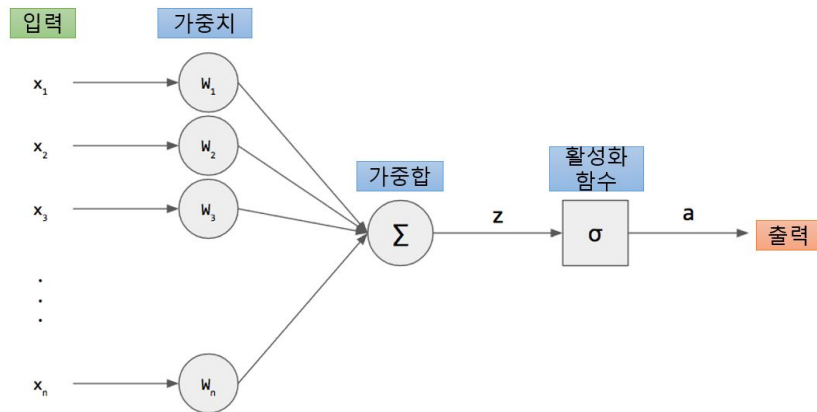
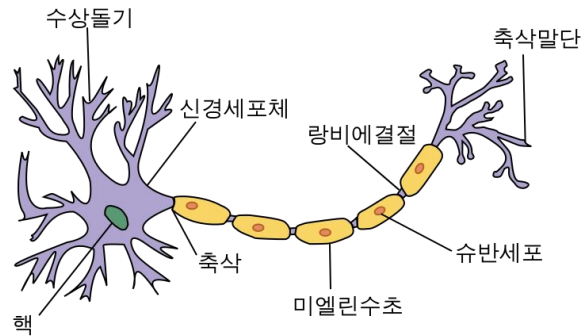
1. 퍼셉트론
2. XOR 예제 코드
3. 오차역전파
4. Activation, Optimizer

----- 쉬는시간 -----

5. pre_class quiz 풀이
6. 오차역전파 심화
7. 확률적 경사 하강법 실습

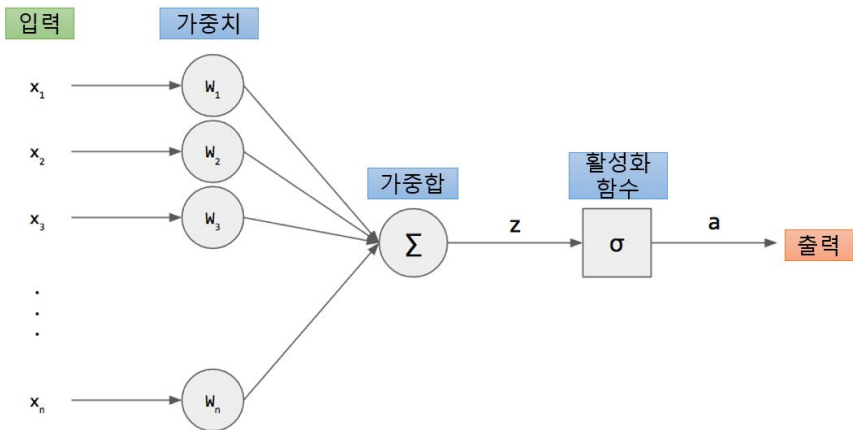
퍼셉트론

SMARCLE 2022_Winter_AI_study



뉴런 : 전위가 임계값을 넘으면 신호 전달, **else 신호 X**
→ **활성화 함수**와 비슷! → 인공 신경망의 연구 시작!

퍼셉트론 : 신경망을 이루는 가장 중요한 기본 단위
(뇌에 비유하면 뉴런에 해당!)



가중치(weight) - 기울기 a

바이어스(bias) - 절편 b

가중합(weighted sum)

입력값과 가중치의 곱을 모두 더한 다음 거기에 바이어스(b)를 더한 값

ex. $w_1 x_1 + w_2 x_2 + b$

활성화 함수(activation function)

입력된 데이터의 가중 합을 출력 신호로 변환하는 함수

ex. 시그모이드 함수

XOR 예제 코드

SMARCLE 2022_Winter_AI_study

X ₁	X ₂	결괏값
0	0	0
0	1	0
1	0	0
1	1	1

AND 진리표

X ₁	X ₂	결괏값
0	0	0
0	1	1
1	0	1
1	1	1

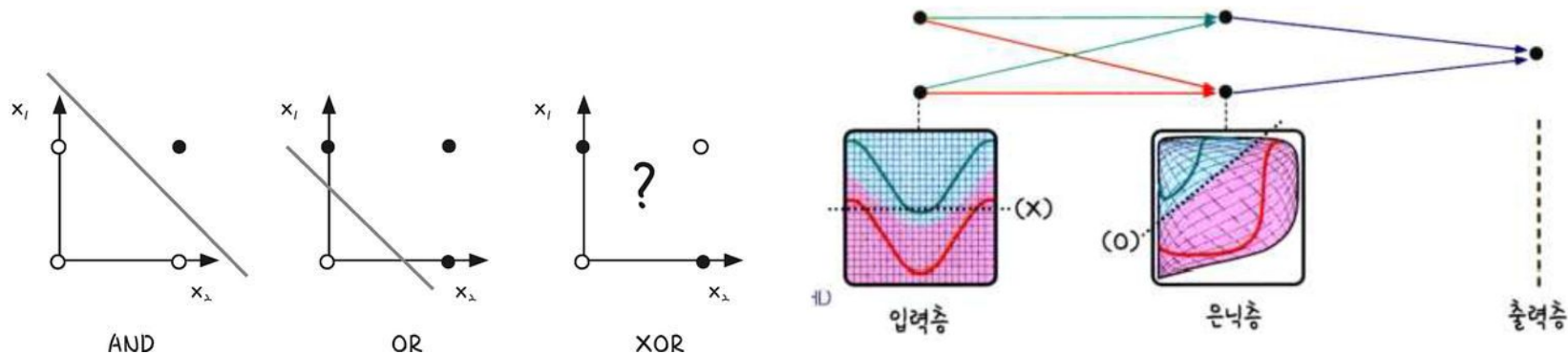
OR 진리표

X ₁	X ₂	결괏값
0	0	0
0	1	1
1	0	1
1	1	0

XOR 진리표

XOR 예제 코드

SMARCLE 2022_Winter_AI_study



흰점과 검은 점을 구분하는 직선을 그을 수 없음

단일 퍼셉트론의 **한계!**

다층 퍼셉트론 (multilayer perceptron)으로 해결가능

XOR 예제 코드

SMARCLE 2022_Winter_AI.study



<https://colab.research.google.com/drive/1I47X627sY-NZY1OnjsmdnjAXP0bml3rW#scrollTo=RP6QcEFb2ORy>

XOR 예제 코드

SMARCLE 2022_Winter_AI_study



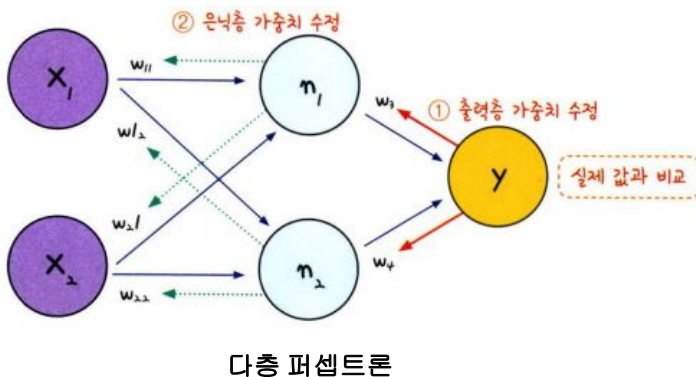
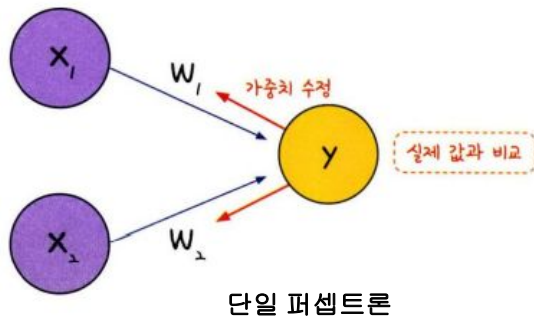
지금까지 친 코드는 시그모이드 함수 값을 실제로 구해서 출력값을 계산하는 코드!
실제로 역전파를 적용한 코드는 **심화과정**에서……

- 신경망 내부의 **가중치**를 수정 할 때 사용
- 경사 하강법의 **확장 개념**

구동 방식

- 임의의 초기 가중치(W) 준 후 결과 계산
- 결과값과 원하는 값의 오차 계산
- 경사하강법 이용
- 오차가 줄어들 지 않을 때까지 반복

(계산 방향이 출력층에서 앞으로 진행)



경사하강법을 이용할 때

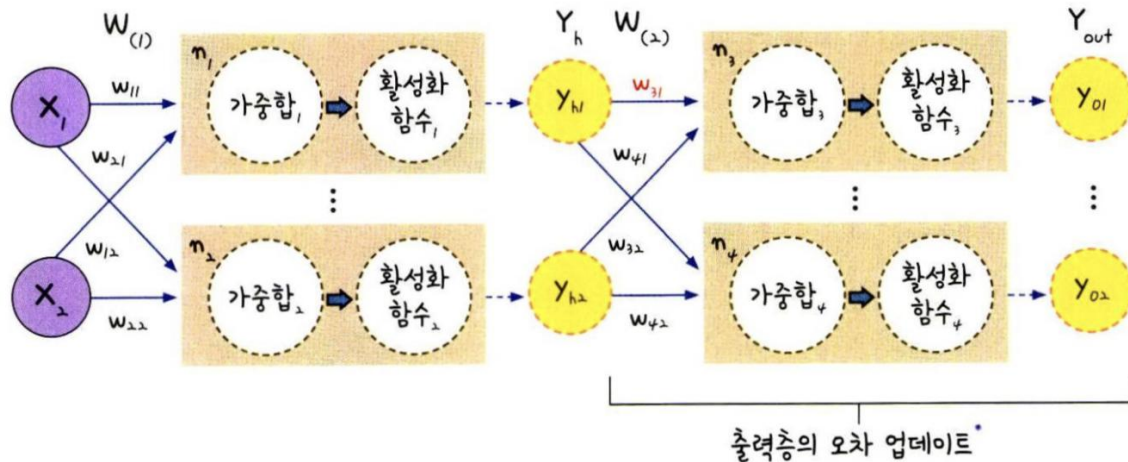
- 바로 앞 가중치를 “오차가 작아지는 **방향**”으로 업데이트
= 미분값(기울기)이 0이 되는 방향
- 즉, 가중치에 기울기를 뺏을 때 가중치의 변화가 전혀 없는 상태
- 가중치에 기울기를 빼도 값의 변화가 없을 때까지 반복

$$W(t+1) = W_t - \frac{\partial \text{오차}}{\partial W}$$

(새 가중치 = 현 가중치 - ‘가중치에 대한 기울기’)

오차 역전파(심화)

SMARCLE 2022_Winter_AI_study



$$w_{31}(t+1) = w_{31}t - \frac{\partial \text{오차} Y_{out}}{\partial w_{31}}$$

w_{31} t는 한 단계 앞에서 이미 계산된 값이므로 구할 필요 없음

$$\frac{\partial \text{오차} Y_{out}}{\partial w_{31}} \quad \leftarrow \text{실제로 구해야 하는 값!}$$

오차 역전파(심화)

SMARCLE 2022_Winter_AI_study

$$\text{오차 } y_{o1} = \frac{1}{2}(y_{t1} - y_{o1})^2 \quad \text{오차 } y_{o2} = \frac{1}{2}(y_{t2} - y_{o2})^2$$

오차 Y_{out} 은 밖으로 도출되는 모든 오차의 합

(y_{t1} 과 y_{t2} 는 **실제 값**, 즉 주어진 데이터를 통해 알 수 있는 상수)

$$\text{오차 } Y_{out} = \frac{1}{2}(y_{t1} - y_{o1})^2 + \frac{1}{2}(y_{t2} - y_{o2})^2$$

이렇게 구한 오차식을 합성미분하면 다음과 같다.

오차 Y_{out} 식에서 y_{o1} 이 시그모이드 함수로 구해지고 시그모이드 함수에는 가중합이 들어가기 때문에 미분식 3개의 곱으로 나타남

(각 미분식을 구하는 것은 쉽지만 합성미분 적용이 어려움.....)

$$\frac{\partial \text{오차 } Y_{out}}{\partial w_{31}} = \frac{\partial \text{오차 } Y_{out}}{\partial y_{o1}} * \frac{\partial y_{o1}}{\partial \text{가중합}_3} * \frac{\partial \text{가중합}_3}{\partial w_{31}}$$

오차 역전파(심화)

SMARCLE 2022_Winter_AI_study

$$\frac{\partial \text{오차 } Y_{out}}{\partial y_{o1}} = \frac{\partial \text{오차 } y_{o1} + \partial \text{오차 } y_{o2}}{\partial y_{o1}} = \frac{\partial \frac{1}{2}(y_{t1} - y_{o1})^2}{\partial y_{o1}} + \frac{\partial \frac{1}{2}(y_{t2} - y_{o2})^2}{\partial y_{o1}} = (y_{t1} - y_{o1}) \times (-1) = (y_{o1} - y_{t1})$$

$$\frac{\partial y_{o1}}{\partial \text{가중합}_3} = \frac{\partial \sigma(\text{가중합}_3)}{\partial \text{가중합}_3} = \text{활성화 함수}_3 \text{의 미분} = \sigma(\text{가중합}_3) \cdot (1 - \sigma(\text{가중합}_3)) = y_{o1} \cdot (1 - y_{o1})$$

Plus. 시그모이드 함수 미분



$$\begin{aligned} \frac{d}{dx} \sigma(x) &= \frac{d}{dx} \left[\frac{1}{1 + e^{-x}} \right] = \frac{d}{dx} (1 + e^{-x})^{-1} = -(1 + e^{-x})^{-2} (-e^{-x}) \\ &= \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} = \frac{1}{1 + e^{-x}} \cdot \frac{(1 + e^{-x}) - 1}{1 + e^{-x}} \\ &= \frac{1}{1 + e^{-x}} \cdot \left(1 - \frac{1}{1 + e^{-x}} \right) = \sigma(x) \cdot (1 - \sigma(x)) \end{aligned}$$

오차 역전파(심화)

SMARCLE 2022_Winter_AI_study

$$\frac{\partial \text{가중합}_3}{\partial w_{31}} = \frac{\partial (w_{31}y_{h1} + w_{41}y_{h2} + 1)}{w_{31}} = y_{h1}$$

$$\frac{\partial \text{오차} Y_{out}}{\partial w_{31}} = (y_{o1} - y_{t1}) * y_{o1} * (1 - y_{o1}) * y_{h1}$$

$$w_{31}(t+1) = w_{31}t - (y_{o1} - y_{t1}) * y_{o1} * (1 - y_{o1}) * y_{h1}$$

잠깐!!

$$\delta y = (y_{o1} - y_{t1}) * y_{o1} * (1 - y_{o1})$$

$$w_{31}(t+1) = w_{31}t - \delta y * y_{h1}$$

바이어스는 1로 설정.

because ...

시그모이드 함수가 가장 안정된 예측을 하게 하는 바이어스이기 때문

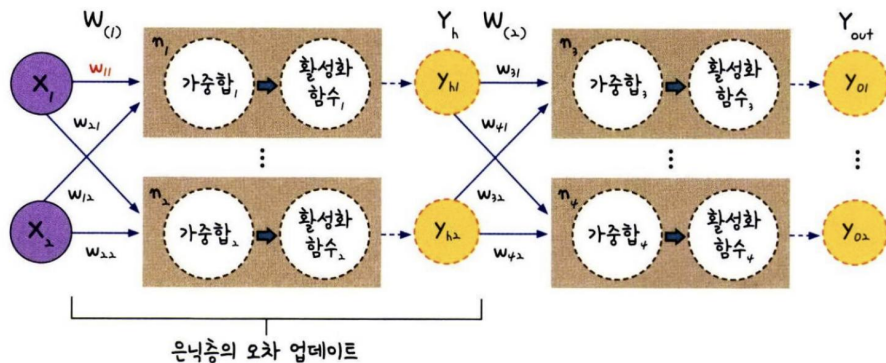
δy 를 델타식이라고 하며 오차 업데이트를 반복할 때마다 그대로 사용해서 오차를 구할 수 있음

반복되는 식의 형태를 델타식으로 지정해놓으면 편함

오차 역전파(심화)

SMARCLE 2022_Winter_AI_study

- 은닉층의 오차 업데이트



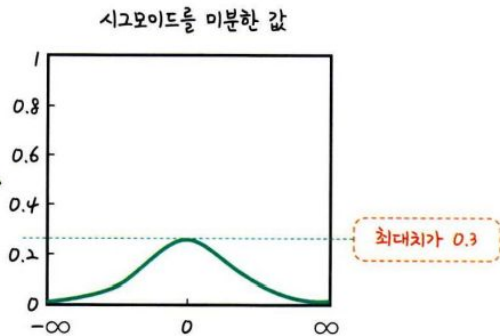
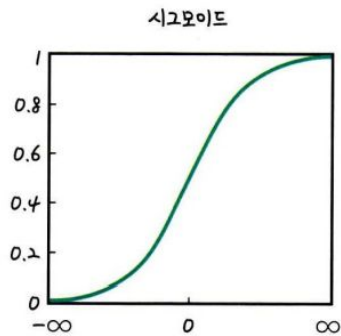
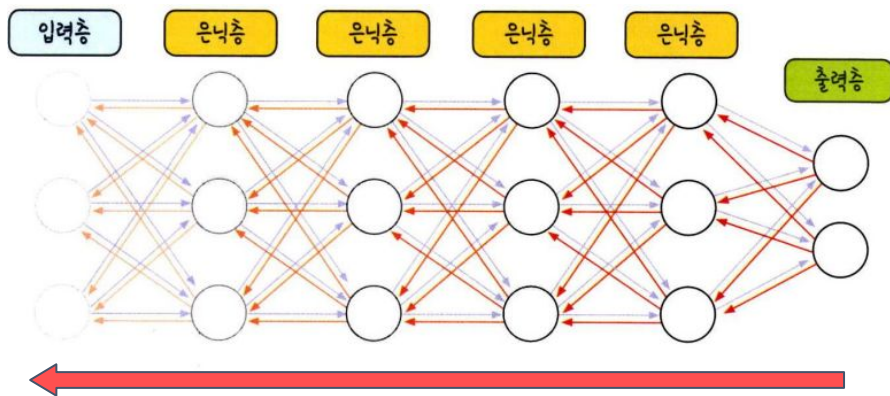
출력층에서 앞에서 한 것처럼 똑같은 방식으로 전개하면 델타식을 구할 수 있으며 이를 통해 오차 업데이트가 가능해진다.

은닉층이 더 있을 경우에는 출력층에서 입력층 방향으로 이동하면서 오차 업데이트를 진행하면 된다.

강력한 심화학습이 기다리고 있습니다..

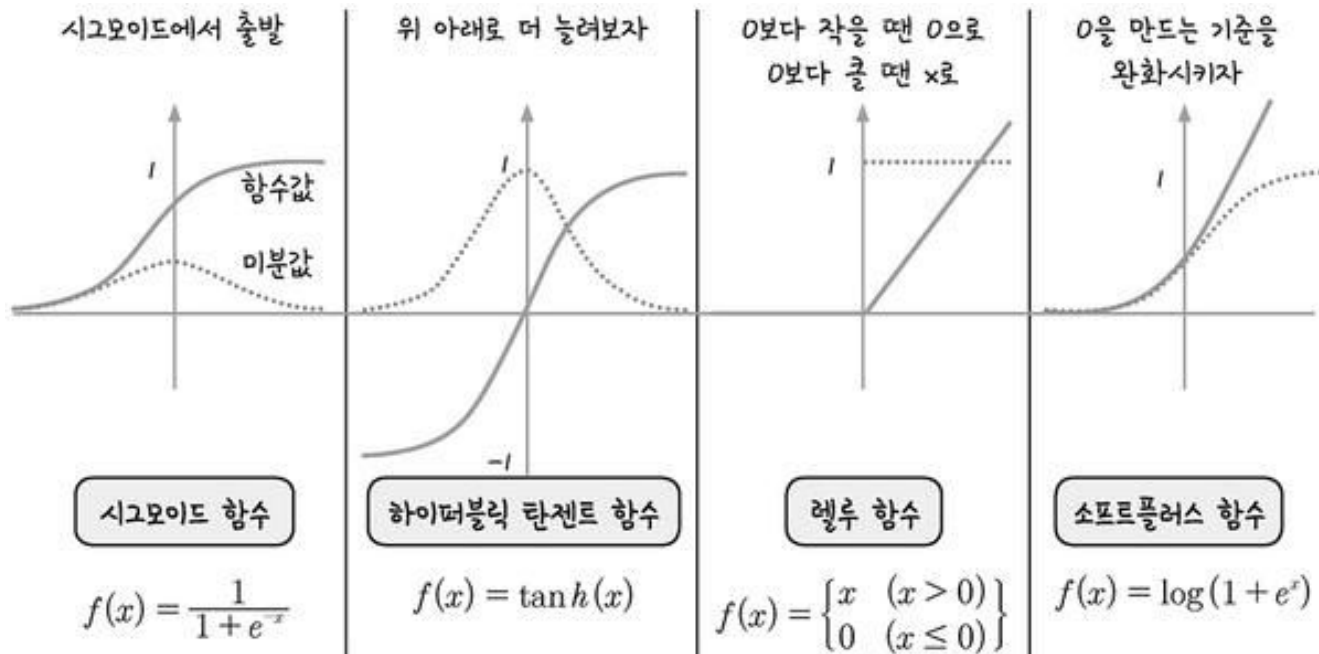
기울기 소실

SMARCLE 2022_Winter_AI_study



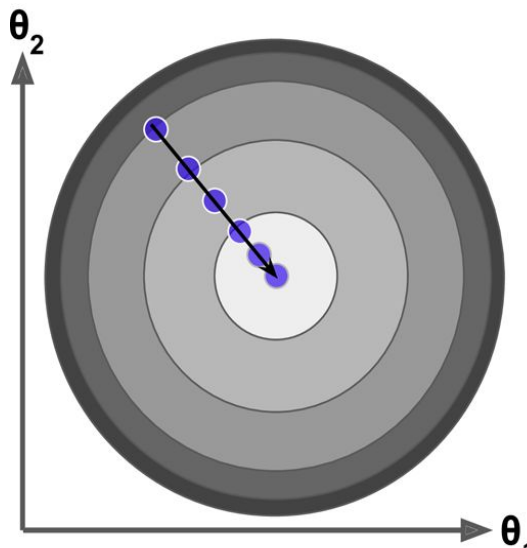
Activation

SMARCLE 2022_Winter_AI_study

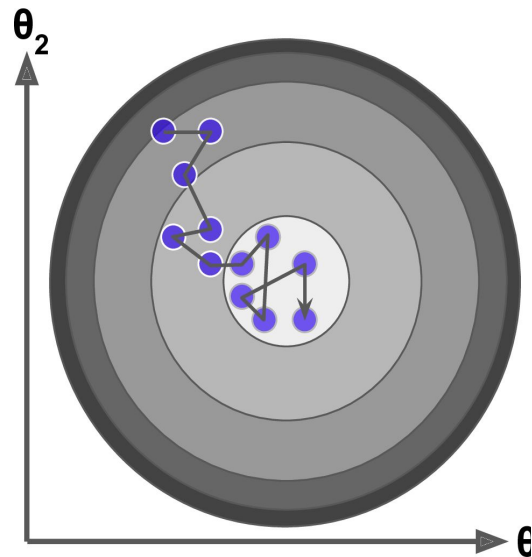


확률적 경사하강법

SMARCLE 2022_Winter_AI_study



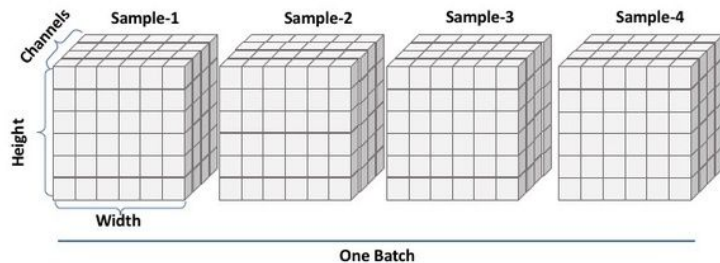
경사 하강법 (GD)



확률적 경사 하강법 (SGD)

batch, mini batch GD

SMARCLE 2022_Winter_AI_study



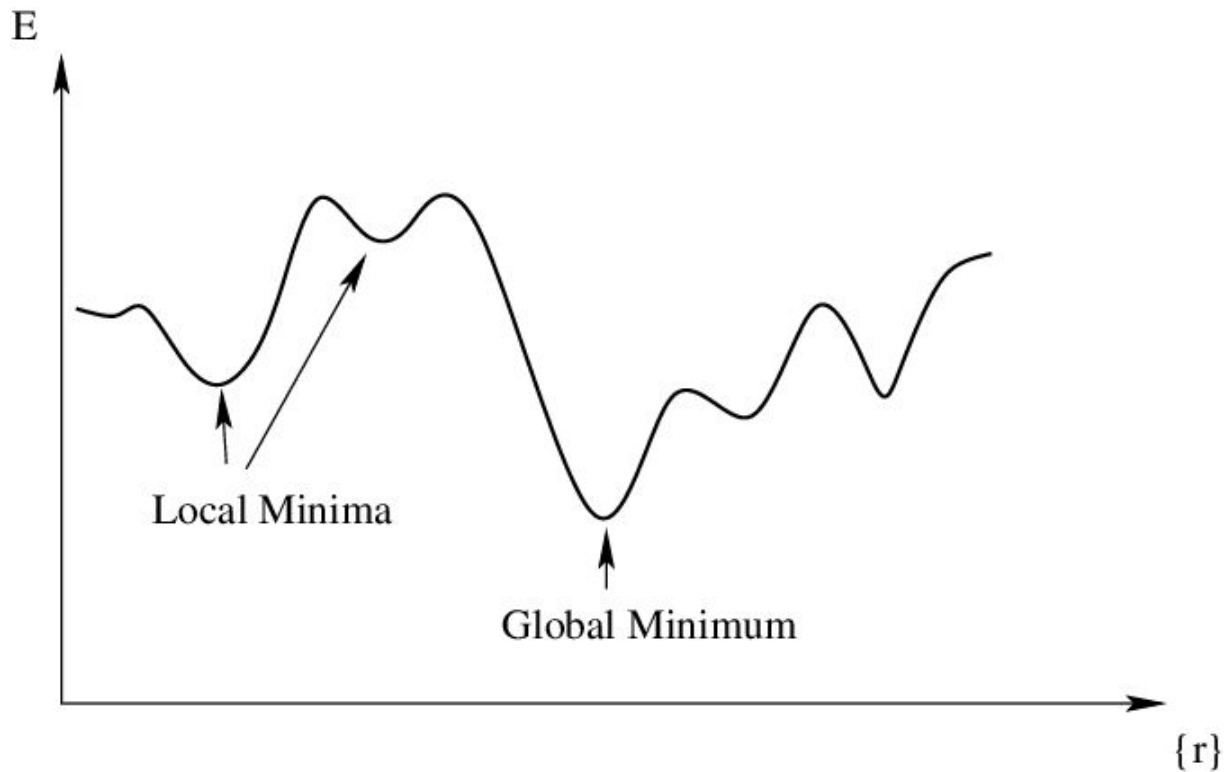
batch



mini batch

limitation

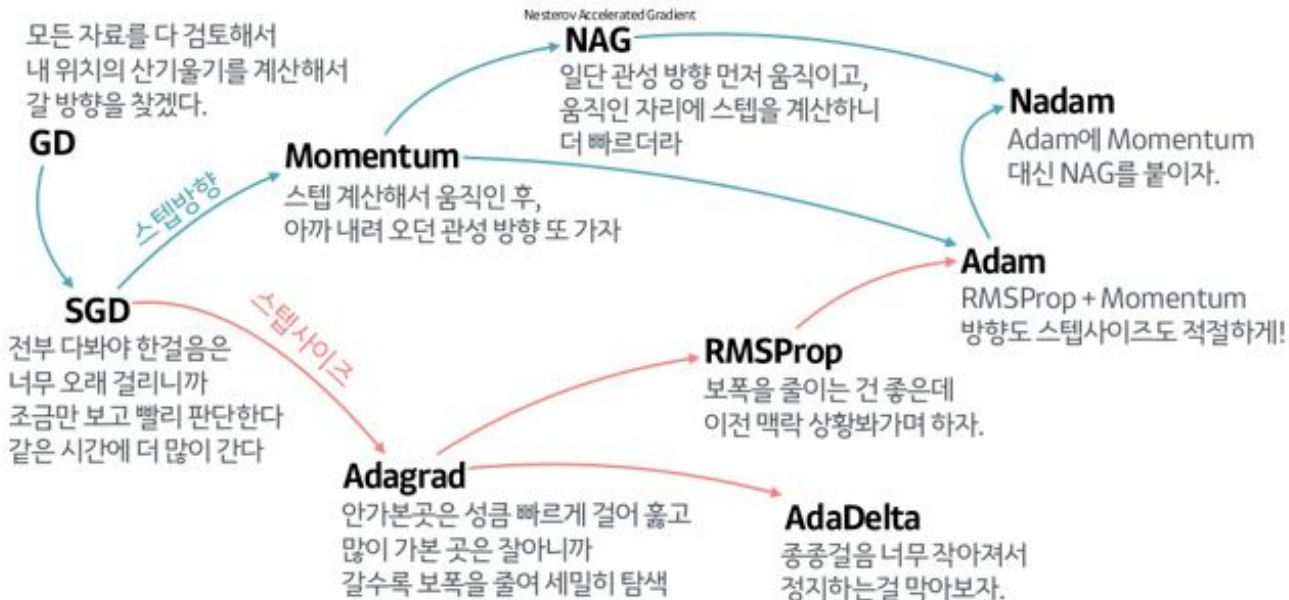
SMARCLE 2022_Winter_AI_study



고급 경사하강법	개요	효과
확률적 경사하강법(SGD)	랜덤하게 추출한 데이터를 사용해 더빨리, 자주 업데이트를 하게 하는 것	속도개선
모멘텀(Momentum)	관성의 방향을 고려해 진동과 폭을 줄이는 효과	정확도 개선
네스테로프 모멘텀(NAG)	모멘텀이 이동시킬 방향으로 미리 이동해서 그레이디언트를 계산, 불필요한 이동을 줄이는 효과	정확도 개선
아다그라드(Adagrad)	변수의 업데이트가 잦으면 학습률을 적게 하여 이동 보폭을 조절하는 방법	보폭크기 개선
알엠에스프롭(RMSProp)	아다그라드의 보폭 민감도를 보완한 방법	보폭 크기 개선
아담(Adam)	모멘텀 + 알엠에스프롭	정확도와 보폭크기 개선

Optimizer

SMARCLE 2022_Winter_AI_study



쉬는시간!!

SMARCLE 2022_Winter_AI_study

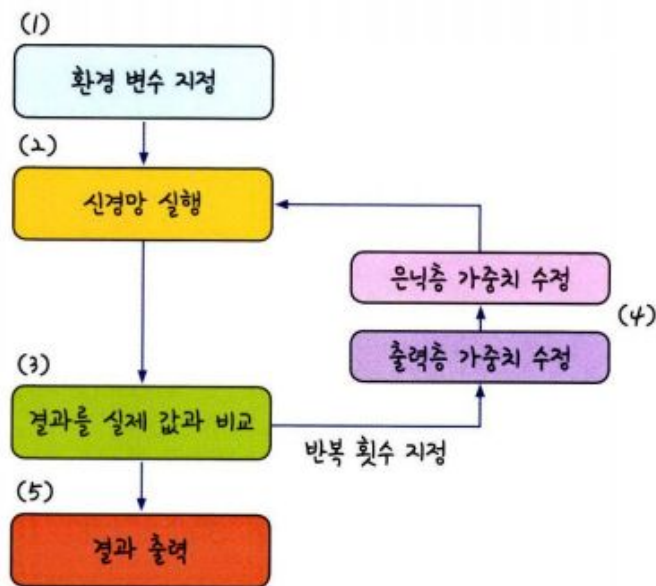


코딩으로 확인하는 오차 역전파

- 신경망의 구현 과정

- 1 | 환경 변수 지정: 데이터셋(입력 값, 결괏값), 학습률, 활성화 함수, 가중치 포함.
- 2 | 신경망 실행:
초기값 입력 → 활성화 함수, 가중치 → 결과값
- 3 | 결과를 실제 값과 비교: 오차 측정
- 4 | 역전파 실행: 출력층, 은닉층의 가중치 수정
- 5 | 결과 출력

자세한 과정은 실습에서 다룰 예정



(심화)파이썬 코딩으로 XOR 문제의 역전파 해결

실습 코드

https://colab.research.google.com/drive/1rWCimKLgImevjl-y8_1ksLF0s4C6rYza?usp=sharing





Colaboratory

<https://colab.research.google.com/drive/1V7cBdy6FmfGxVrkKWmHyCK3i00S00aSo?usp=sharing>

Fin

SMARCLE 2022_Winter_AI.study

Thank you for watching

4팀 이용빈, 최원서, 김용현, 유혁재, 유정수

A decorative graphic on the left side of the page, consisting of a vertical line of dark blue squares, each with a white spiral binding element on its left side.

contents

SMARCLE 2022_Winter_AI.study