

# 경사하강법

## • 개요

손실함수

경사하강법 (1)

경사하강법 (2)

Hyperparameter

경사 하강법의 한계

# 경사하강법(Gradient Descent, GD)

경사 하강법은 함수값이 낮아지는 방향으로 독립 변수 값을 이동해가면서 최종적으로는 최소 값을 갖도록 하는 독립 변수 값을 찾는 방법

≈ 안개가 가득 낀 산 속에서 내려갈 때 산의 높이가 가장 낮아지는 방향으로 한 발자국씩 이동하는 과정

## • 개요

손실함수

경사하강법 (1)

경사하강법 (2)

Hyperparameter

경사 하강법의 한계

# 경사하강법의 목적과 이유

; 함수의 최솟값을 찾는 문제에서 사용

➔ 미분 계수가 0인 지점을 찾지 않고 경사하강법을 사용하는 이유는?

- 함수가 너무 복잡해서 미분 계수를 구하기 어려운 경우
- 경사하강법을 이용하는 것이 미분 계수를 구하는 것보다 더 쉬운 경우
- 방대한 데이터 양을 효율적으로 계산하기 위해



# 손실함수

개요

- 손실함수

경사하강법 (1)

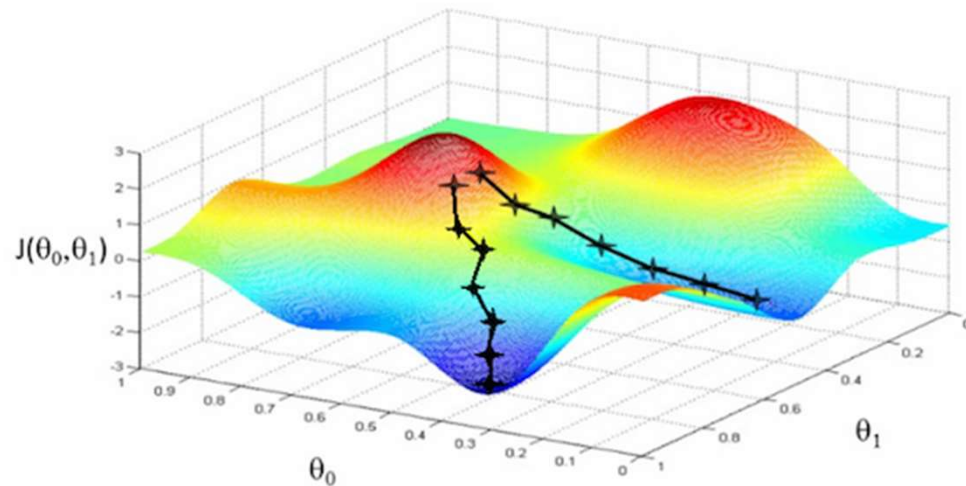
경사하강법 (2)

Hyperparameter

경사 하강법의 한계

## 최적화(Optimization)

딥러닝 분야에서 최적화란 손실 함수(Loss Function) 값을 최소화하는 모델 파라미터를 구하는 것.



## 개요

### 손실함수

#### 경사하강법 (1)

#### 경사하강법 (2)

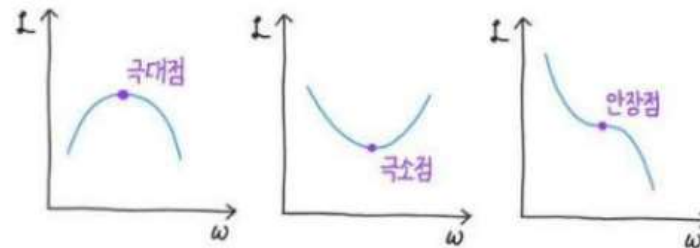
#### Hyperparameter

#### 경사 하강법의 한계

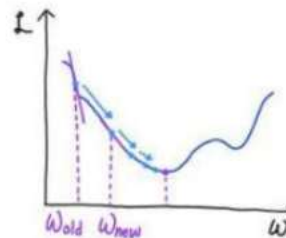
## 손실함수와 경사하강법

<손실함수가  $w$ 에 대한 일변수함수임을 가정>

- 다양한 손실함수의 개형



- 경사하강법 : 손실함수를 따라 내려가면서 극소를 만족시키는  $w$ 를 찾는 과정



개요

- 손실함수

경사하강법 (1)

경사하강법 (2)

Hyperparameter

경사 하강법의 한계

## 일반적인 손실 함수의 표기

→ 얼마나 잘못 되었는지를 나타내는 함수

- 모델 파라미터 :  $w_1, w_2, w_3, \dots, w_M$
- Data points :  $(x_n, y_n)_{n=0}^{N-1}$

$$\mathcal{L}(w_1, \dots, w_M; (x_n, y_n)_{n=0}^{N-1})$$



# 일변수함수에서의 경사하강법



개요

손실함수

## • 경사하강법 (1)

경사하강법 (2)

Hyperparameter

경사 하강법의 한계

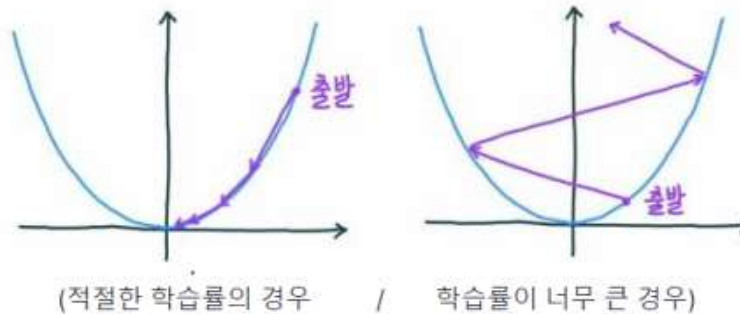
# 일변수 함수에서의 경사하강법

- $\mathcal{L}(w)$  의 최소를 찾아야 함.

- $\frac{d\mathcal{L}}{dw} < 0 \rightarrow \text{increase } w, \quad \frac{d\mathcal{L}}{dw} > 0 \rightarrow \text{decrease } w$

$$\therefore w_{new} = w_{old} - \delta \cdot \left. \frac{d\mathcal{L}}{dw} \right|_{w=w_{old}}$$

- $\delta$  : 학습률(Learning rate) : 한 번의 Step에서 가중치 값을 변화시키는 양에 대한 척도



개요

손실함수

## • 경사하강법 (1)

경사하강법 (2)

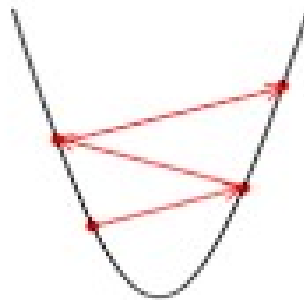
Hyperparameter

경사 하강법의 한계

# 경사하강법

- $w_{new} = w_{old} - \delta \cdot \frac{dL}{dw} |_{w=w_{old}}$
- $\delta$  : 학습률(Learning rate) : 한 번의 step에서 가중치 값을 변화시키는 양에 대한 척도

Big Learning Rate



Just right



Too small



개요

손실함수

## • 경사하강법 (1)

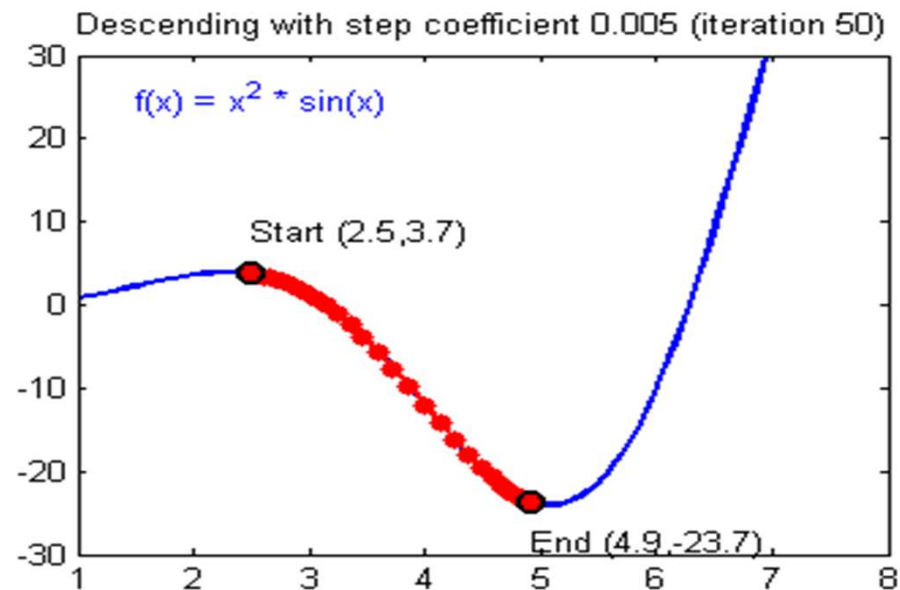
경사하강법 (2)

Hyperparameter

경사 하강법의 한계

# 경사하강법

- 경사하강법의 종료 조건
- $\frac{dL}{dw} \big|_{w=w_{old}} \cong 0$  (Thresholding) 등 다양한 방법이 있음.



개요

손실함수

## • 경사하강법 (1)

경사하강법 (2)

Hyperparameter

경사 하강법의 한계

# 경사하강법

- 경사하강법의 종료 조건
- Python 코드

```
import numpy as np
def descent_down_parabola(w_start, learning_rate, num_steps):
    w_values = [w_start]
    for _ in range(num_steps):
        w_old = w_values[-1]
        w_new = w_old - learning_rate * (2 * w_old)
        w_values.append(w_new)
    return np.array(w_values)
```



# 다변수함수에서의 경사하강법

개요

손실함수

경사하강법 (1)

## • 경사하강법 (2)

Hyperparameter

경사 하강법의 한계

## 다변수 함수에서의 경사하강법

- 모델 파라미터 :  $w_1, w_2, w_3, \dots, w_M$
- $w_{new} = w_{old} - \delta \cdot \nabla L(w)|_{w=w_{old}}$

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_M \end{pmatrix}, \quad \nabla L(w)|_{w=w_{old}} = \begin{pmatrix} \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_M} \end{pmatrix},$$



개요

손실함수

경사하강법 (1)

## • 경사하강법 (2)

Hyperparameter

경사 하강법의 한계

## 다변수 함수에서의 경사하강법

- $L(w_1, w_2) = 3w_1^2 + 4w_2^2$  에 대한 경사하강법

$$\begin{pmatrix} w_{1,new} \\ w_{2,new} \end{pmatrix} = \begin{pmatrix} w_{1,old} \\ w_{2,old} \end{pmatrix} - \delta \cdot \begin{pmatrix} 6w_{1,old} \\ 8w_{2,old} \end{pmatrix}$$

- Python Code

```
import numpy as np
def descent_down_parabola(w_start, learning_rate, new_steps):
    xy_values = [w_start]
    for _ in range(num_steps):
        xy_old = xy_values[-1]
        xy_new = xy_old - learning_rate * (np.array([6., 8.]) * xy_old)
        xy_values.append(xy_new)
    return np.array(xy_values)
```



### 3.Hyperparameter



개요

손실함수

경사하강법 (1)

경사하강법 (2)

• Hyperparameter

경사 하강법의 한계

## 경사하강법의 Hyperparameter

- 경사하강법의 Hyperparameter
  - : 경사하강법 함수 내에서 사용되는 변수와 방법
  - $w_{old,0}$  ,  $\delta$  , *Criteria for stopping step*, 반복횟수



개요

손실함수

경사하강법 (1)

경사하강법 (2)

• Hyperparameter

경사 하강법의 한계

## 경사하강법의 Hyperparameter

1.  $\mathbf{w}_{old,0}$  : Gaussian Random Distribution으로 Randomly choose
2.  $\delta$ (학습률, Learning Rate)
  - 10의 거듭제곱들을 각각 학습률로 사용하여 경사하강법 진행
3. Criteria for stopping step; “큰 변화가 없으면 종료한다”
  - ① Gradient 값을 그래프에 나타낸 뒤 사용자의 판단하에 종료
  - ② Auto-stopping option : n번 반복 후 값의 변화가 없을 시 종료
  - ③ Thresholding : 특정 값 이하로 gradient가 감소하면 종료
4. 반복횟수



개요

손실함수

경사하강법 (1)

경사하강법 (2)

• **Hyperparameter**

경사 하강법의 한계

## 경사하강법의 Hyperparameter

적절한 Hyperparameter 의 설정방법

1. 대략적으로 하이퍼파라미터 값의 범위를 설정한다.
2. 설정된 범위에서 하이퍼파라미터의 값을 무작위로 추출한다.
3. 1단계에서 샘플링한 값을 사용하여 학습하고, 정확도를 평가한다.
4. 1,2 단계를 반복하며, 그 정확도를 보고 범위를 점점 좁힌다.



#### 4. 경사하강법의 한계

개요

손실함수

경사하강법 (1)

경사하강법 (2)

Hyperparameter

• **경사 하강법의 한계**

## 경사하강법의 한계

1. 데이터가 많아질수록 계산량 증가
2. Local minimum 문제
3. Plateau 문제



개요

손실함수

경사하강법 (1)

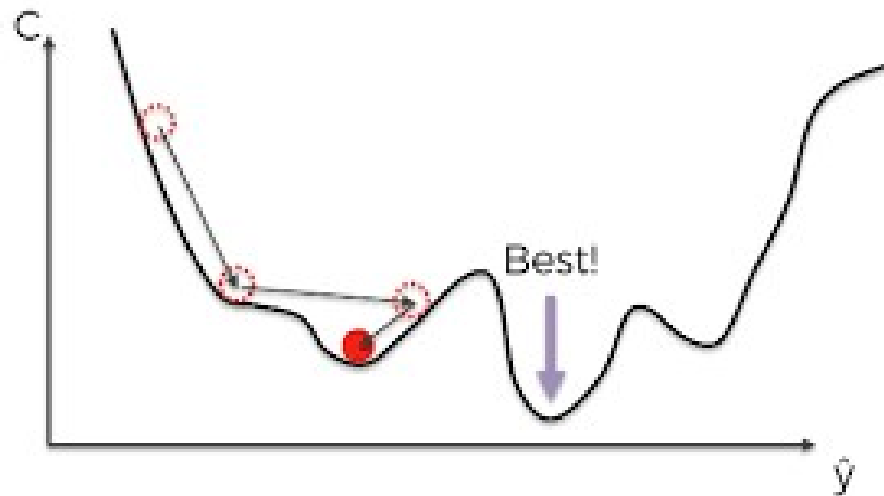
경사하강법 (2)

Hyperparameter

## • 경사 하강법의 한계

# 경사하강법의 한계

- Local minimum
  - Global minimum 이 아닌 Local minimum에 빠지게 된다!



개요

손실함수

경사하강법 (1)

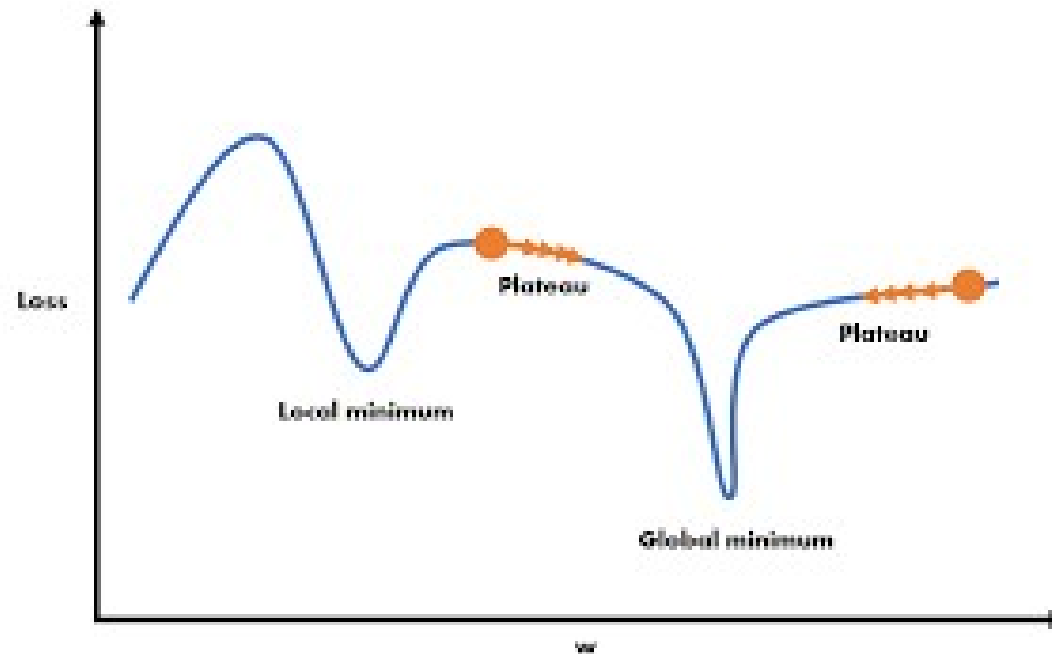
경사하강법 (2)

Hyperparameter

## • 경사 하강법의 한계

# 경사하강법의 한계

- Plateau
  - 평탄한 지역에서는 학습 속도가 매우 느려지며, 느려지다 못해 정지해버릴 수도 있다.



감사합니다