19기 정규세션

ToBig's 19기 강의자
고나경

Optimization
최적화

Contents

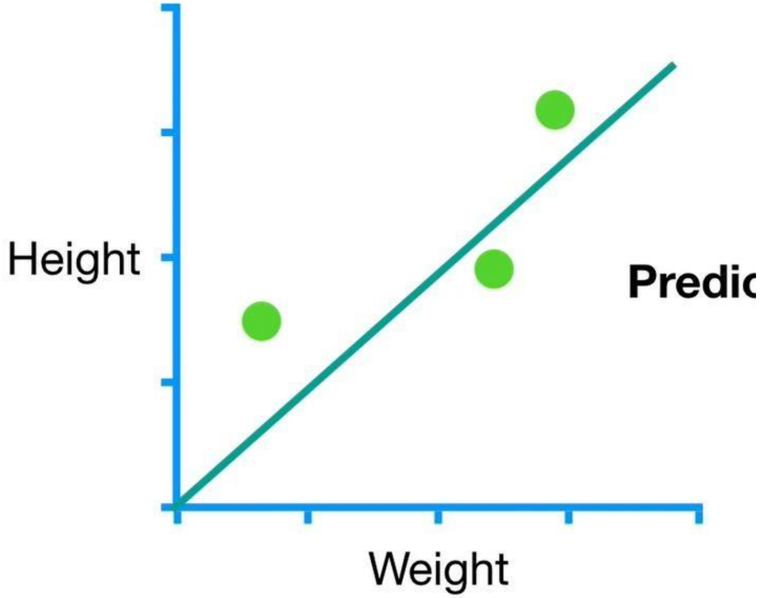
Introduction

최적화란?

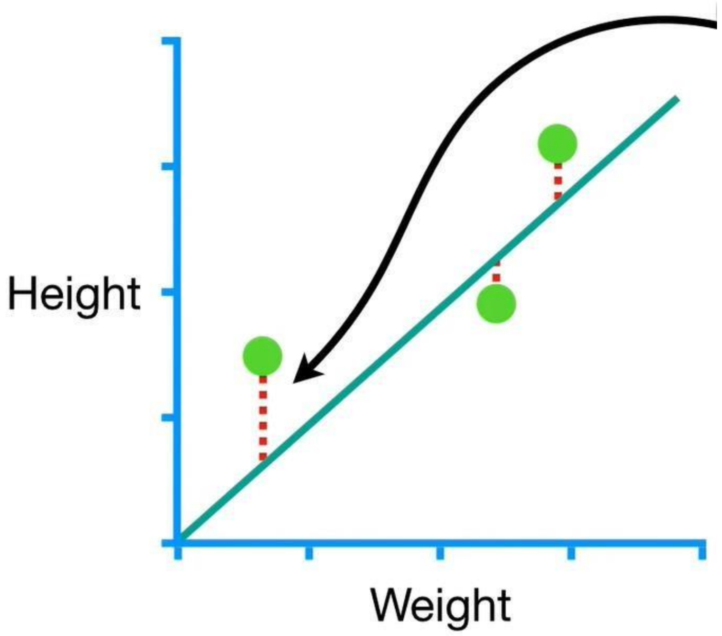
몸무게를 통해 키를 예측해본다면?

키(Y)= a*몸무게(X) + b

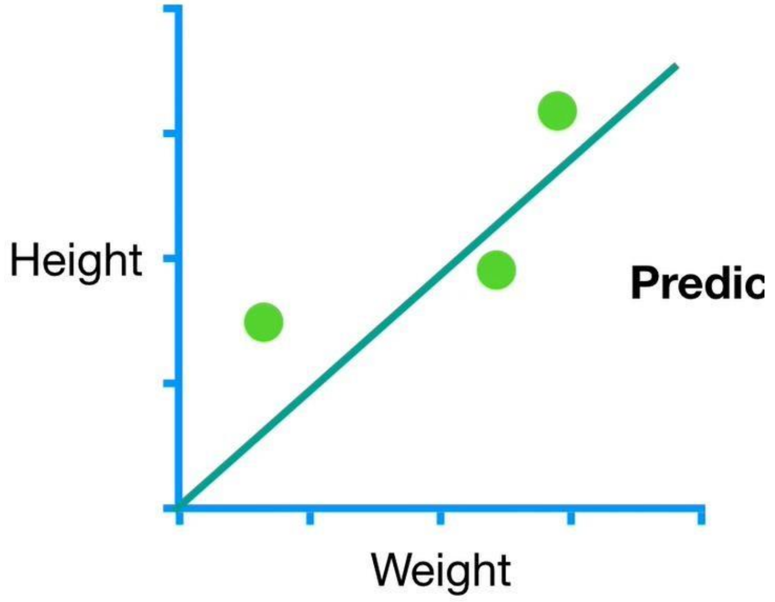
모수: a,b



a=1, b=0으로 시작!



잔차합(Loss) 계산



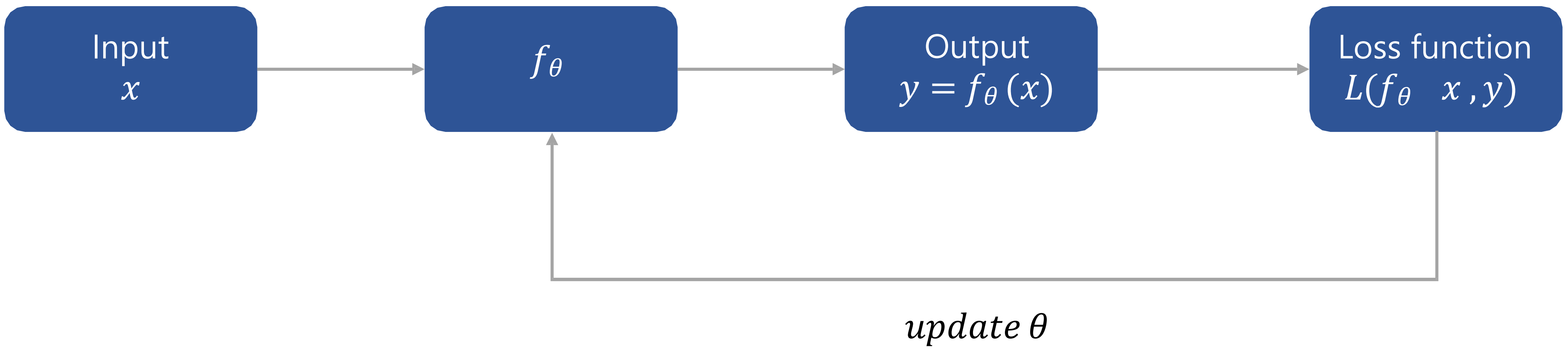
a=0.8, b=0.5로 업데이트!
다시 잔차합 계산 후 업데이트 반복

Unit 01 | Introduction

우리가 추정하는 데이터의 종류는 두가지

연속형 변수 Continuous Variable	이산형 변수 Discrete Variable
선형 회귀 Linear Regression	로지스틱 회귀 Logistic Regression
-> 값을 추정	-> 강아지 사진 맞나 Yes or No -> 분류 문제

✓ Optimization



- $\boldsymbol{argmin_\theta L(f_\theta\ x, y)}$

최적의 모수, 진짜 실제값 θ을 찾아가는 과정!

Unit 02 | MLE

그럼 계수들을 어떤 원리를 베이스로 바꿀까?
-> MLE

Unit 02 | MLE

A 주머니

Ⓡ Ⓡ

Ⓑ Ⓑ

B 주머니

Ⓡ Ⓡ Ⓡ

Ⓑ

1) 하나의 주머니 선택
2) 선택된 주머니에서 5번 복원추출: Ⓡ 나온 횟수 관측 ⇨ **A, B 중 어떤 주머니인가?**

random sample X_1, X_2, X_3, X_4, X_5 ⇒ 통계량(statistic) $Y = X_1 + X_2 + X_3 + X_4 + X_5$ 관측

$Y \sim \mathbf{bin}(5, \theta)$ $\Omega =$ {1/2,3/4}

$$f(y;\theta) = \binom{5}{y} \theta^y (1-\theta)^{5-y}$$ ➡ $\theta \in \Omega$, 중 $f(y;\theta)$를 **최대로 하는** θ **값** 선택

likelihood function (가능도/우도 함수)

A 주머니 ☞ $Y \sim$ bin(5, 1/2) & **B 주머니** ☞ $Y \sim$ bin(5, 3/4)

y	0	1	2	3	4	5
$f(y;\theta=1/2)$	1/32	5/32	10/32	10/32	5/32	1/32
$f(y;\theta=3/4)$	1/1024	15/1024	90/1024	270/1024	405/1024	243/1024

$$f(y;\theta) = \binom{5}{y}\theta^{y}(1-\theta)^{5-y}$$

관측결과		θ에 대한 추론	
$y =$ **0, 1, 2, 3**	⇨	$\hat{\theta} = 1/2$	**(A 주머니)**
$y =$ **4, 5**	⇨	$\hat{\theta} = 3/4$	**(B 주머니)**

Unit 02 | MLE

각 표본의 표본분포 : $f_{\mu,\sigma^2}(x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$

우도(likelihood) : $P(x|\theta) = \prod_{i=1}^{n} f_{\mu,\sigma^2}(x_i) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$

로그-우도 : $\sum_{i=1}^{n} \log \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) = \sum_{i=1}^{n} \left\{-\frac{(x_i-\mu)^2}{2\sigma^2} - \log(\sigma) - \log(\sqrt{2\pi})\right\}$

✓ Likelihood

Linear regression

$\hat{y}_i = \theta^T X_i$

$Y \mid X = x \sim N(\theta^T X_i, \sigma^2)$

$X \sim$ (θ과는 무관)

$$L(Y_i|X_i;\theta) = \prod_{i=1}^{m} pdf(x_i, y_i) = \prod_{i=1}^{m} pdf_{y|x}(y_i|x_i) \bullet pdf_x(x_i)$$

$$f(x|y) = \frac{f(x,y)}{f_Y(y)}$$

$$= \prod^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \theta^T X_i)^2}{2\sigma^2}\right) * (\theta\text{과는 무관})$$

$$\log(L(Y_i|X_i;\theta)) = \sum_{i=1}^{m} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(y_i - \theta^T X_i)^2}{2\sigma^2} + \log(\theta\text{과는 무관})\right)$$

$$= -\frac{1}{2\sigma^2} \underbrace{\sum_{i=1}^{m} (y_i - \theta^T X_i)^2}_{\text{RSS}} + n\log\frac{1}{\sqrt{2\pi\sigma^2}} + n\log(\theta\text{과는 무관})$$

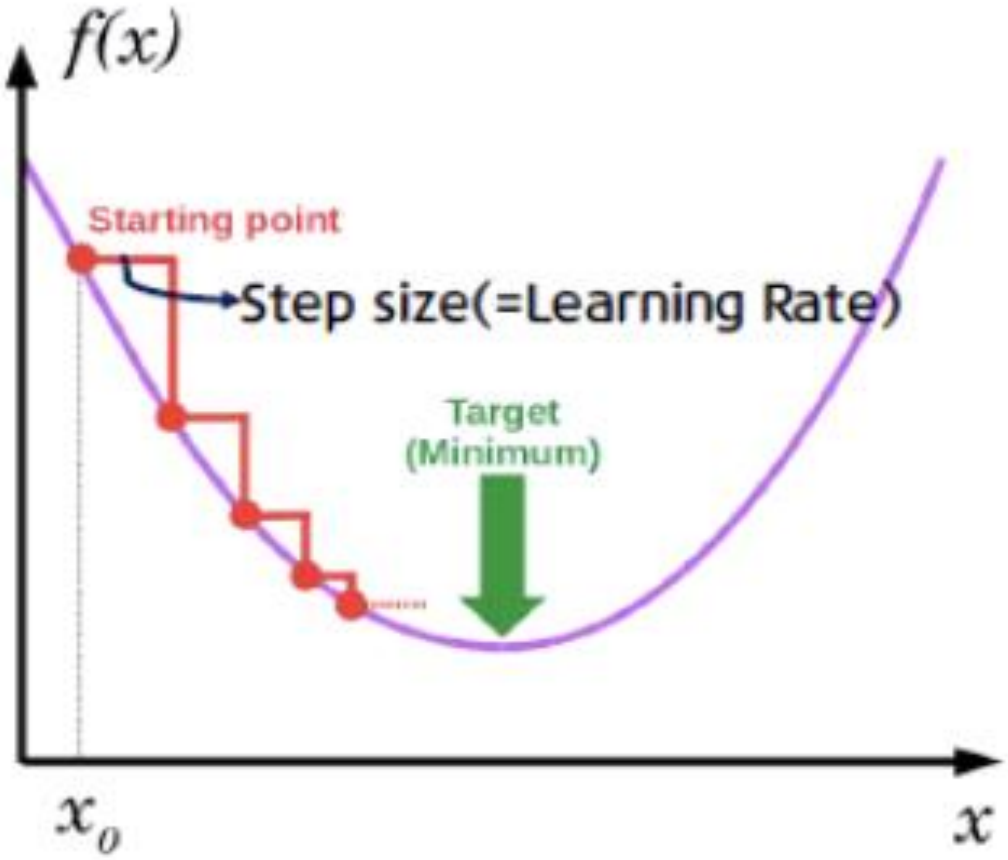
Maximizing loglik ⇔ minimizing RSS

구체적 방법?

-> 경사하강법(Gradient Descent)

Unit 03 | Gradient Descent Algorithm

✓ Gradient Descent

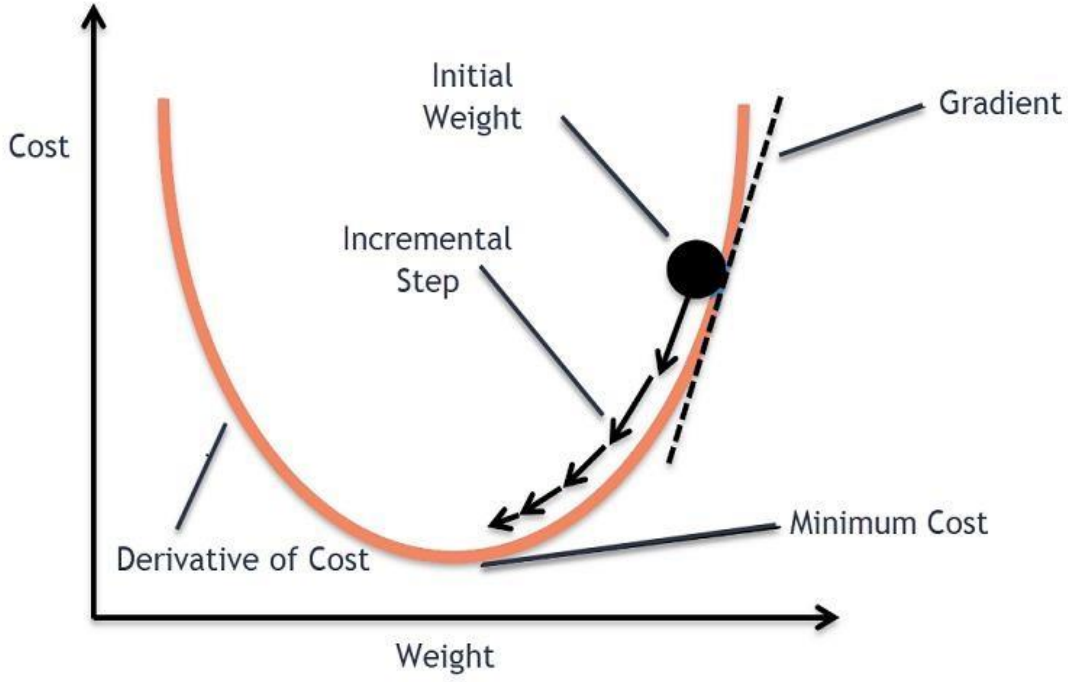


Unit 03 | Gradient Descent Algorithm

선형회귀의 목적함수: $l(\theta) = \frac{1}{2}\Sigma(y_i - \theta^T X_i)^2$

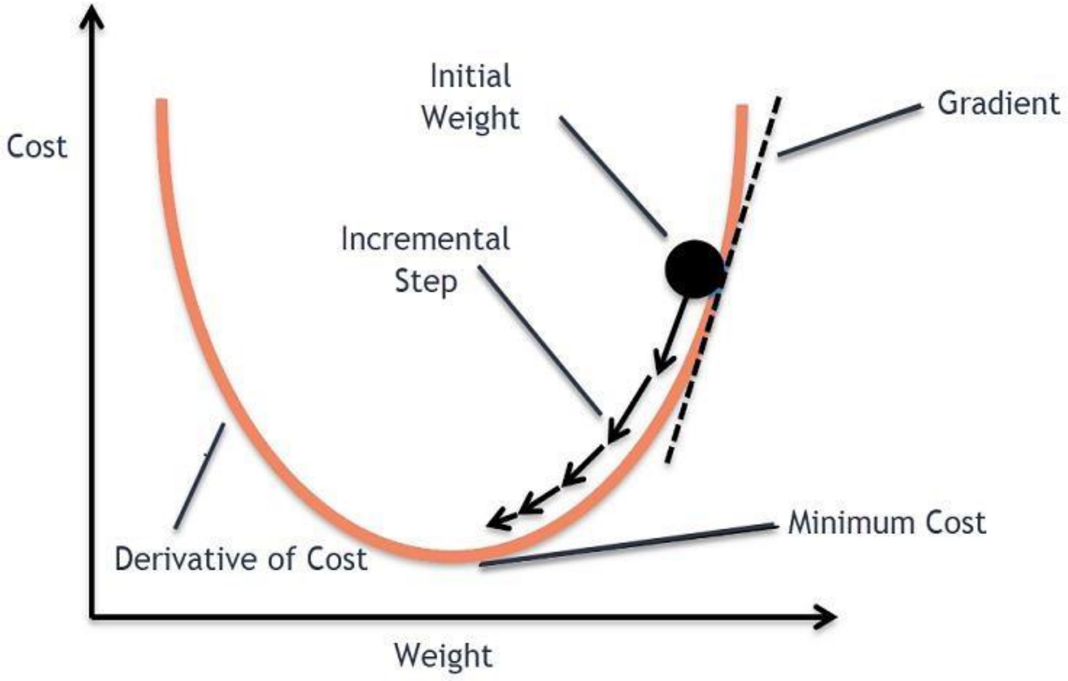
*목적함수란?: 우리가 최적화 하고자 하는 함수

경사하강법에서의 목적함수란?:
우리는 로스값을 최소화 하는 것이 목적이니,
θ로 어떤 값을 넣었느냐에 따라
로스 $l(\theta)$ 가 얼마나 나오는지 계산해주는 함수
(비용함수)



Unit 03 | Gradient Descent Algorithm

✓ Gradient Descent : 로스가 최소점에 도달할 때까지 계속 θ 에서 기울기만큼 빼주는 것



함수의 기울기가 0이 되는 지점까지(최소점까지) 계속 현재의 θ값에서 기울기만큼 빼준다.

$$: \boldsymbol{\theta} := \boldsymbol{\theta} - \boldsymbol{\alpha} \frac{\partial}{\partial \boldsymbol{\theta}_j} \boldsymbol{l}(\boldsymbol{\theta})$$

여기서 $\boldsymbol{\alpha}$ 는 learning rate로 사용자가 임의의 값을 설정.
한 번에 기울기 그대로 빼주면 너무 많이 움직이거나 너무 적게 움직일 수 있어서, 적절한 '스케일링 ' 을 하는 것

선형회귀 목적함수 기울기:

$$\frac{\partial}{\partial \theta_j} l(\theta) = -\Sigma(y_i - \theta^T X_i) X_{ij}$$

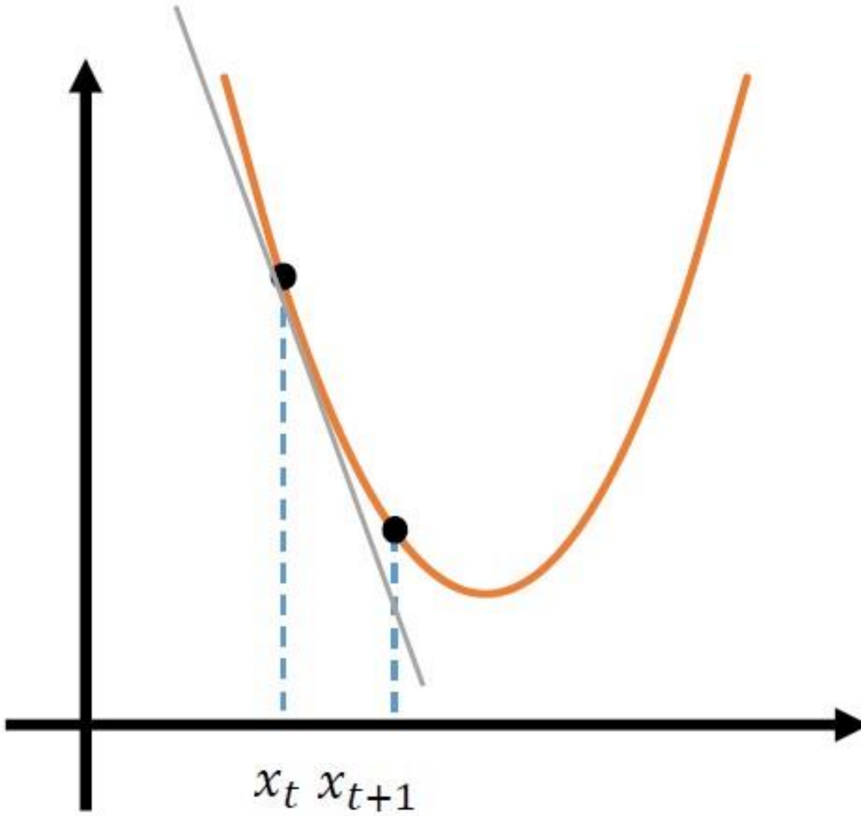
Unit 03 | Gradient Descent Algorithm

✓예시 $f(x) = x^2$

$$x_{t+1} \leftarrow x_t - \alpha \nabla f(x_t)$$

1) $x_t = 2, \alpha = 0.01, f'(x_t) = 4$
$1.96 \leftarrow 2 - 0.01 * 4 = 2 - 0.01 * 4$
$f(1.96) \leq f(2)$

2) $x_t = -3, \alpha = 0.1, f'(x_t) = -6$
$-2.4 \leftarrow -3 - 0.1 * (-6)$
$f(-2.4) \leq f(-3)$



Unit 03 | Gradient Descent Algorithm

✓ 편미분

- $f(x, y) = x^2 y$

$$\frac{\partial f}{\partial x} = \underbrace{\frac{\partial}{\partial x} x^2 y}_{\substack{\text{Treat } y \text{ as constant;} \\ \text{take derivative.}}} = 2xy$$

$$\frac{\partial f}{\partial y} = \underbrace{\frac{\partial}{\partial y} x^2 y}_{\substack{\text{Treat } x \text{ as constant;} \\ \text{take derivative.}}} = x^2 \cdot 1$$

$$f_x = \frac{\partial f}{\partial x} = lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h}$$

And partial derivative of function f with respect y keeping x as constant, we get;

$$f_y = \frac{\partial f}{\partial y} = lim_{h \to 0} \frac{f(x, y+h) - f(x, y)}{h}$$

Unit 03 | Gradient Descent Algorithm

✓ Gradient Descent Algorithm

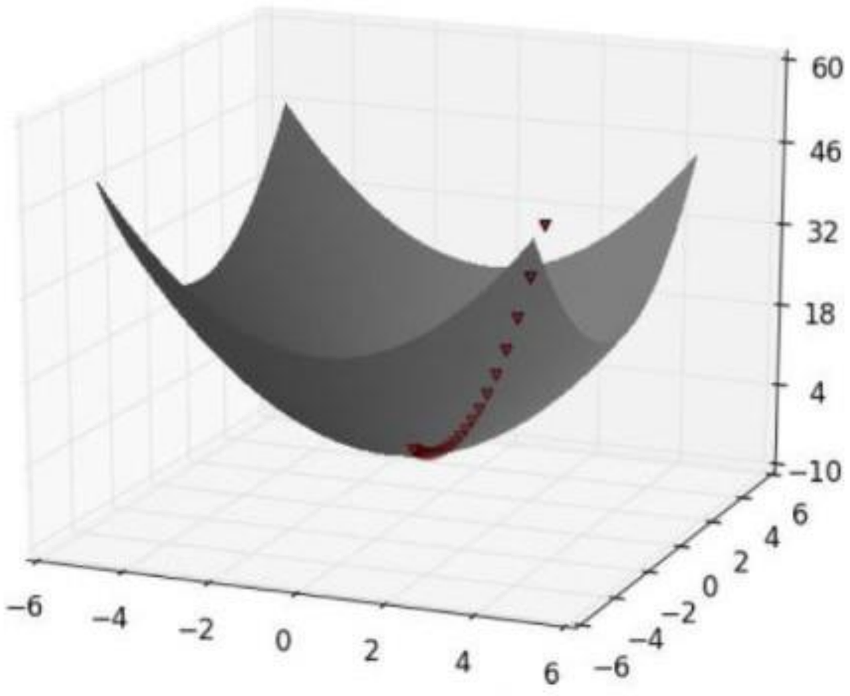


Figure 8-1. Finding a minimum using gradient descent

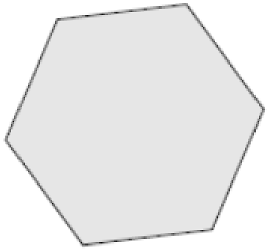
이런식으로 경사하강법을 이용하면 로스값, 목적함수(경사하강법에서 비용함수)를 최소화시킬 수 있다!

여기까지가 선형회귀 문제를 푸는데 경사하강법을 이용하여 최적의 모수를 찾는 것.

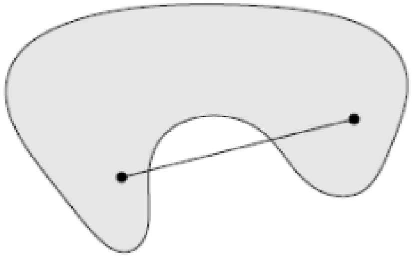
이제 로지스틱 회귀 문제를 푸는데 경사하강법을 똑같이 적용하고 싶지만 한가지 문제점이 있다.
-> 로지스틱 회귀에서의 비용을 MSE(평균제곱오차)방법으로 추정하면 목적함수가 convex하지 않다는 것.

볼록집합(convex set)

$$\text{for all } x_1, x_2 \in C, \lambda x_1 + (1-\lambda)x_2 \in C, \lambda \in (0,1)$$



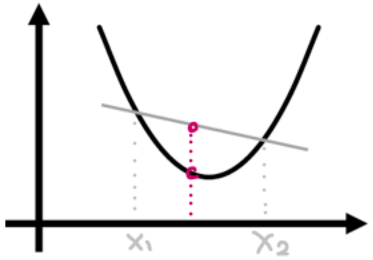
Convex set



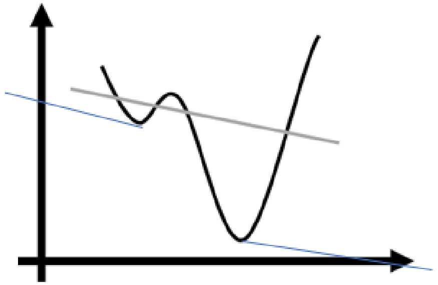
Convex set (x)

볼록함수(convex functions)

함수 $f: I \to R$가 모든 $0<\lambda<1, x \in I, y \in I$에 대해 $f(\lambda x+(1-\lambda)y) \leq \lambda f(x)+(1-\lambda)f(y)$ 를 만족시키면 함수 f를 볼록(convex) 이라고 한다.



Convex



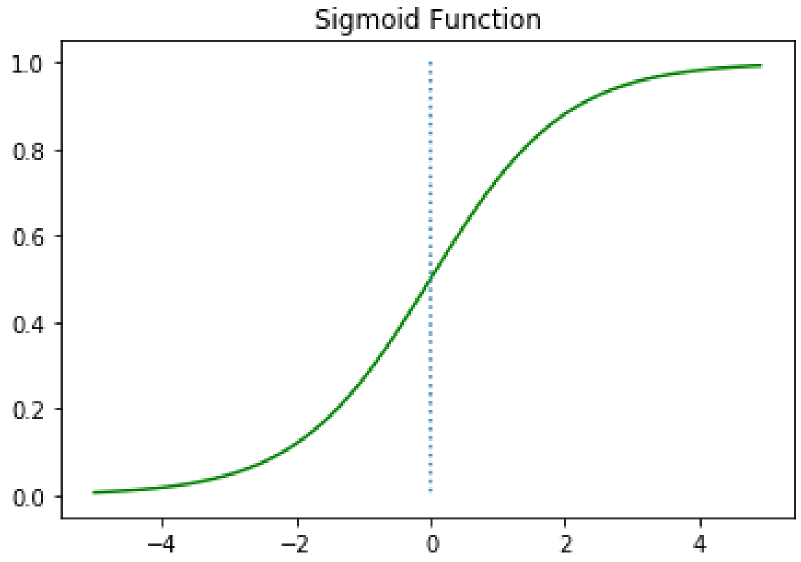
Non-convex

✓ 로지스틱 회귀?

로지스틱 회귀 함수: sigmoid 함수

$$p(X_i) = \frac{1}{1 + e^{-X_i\theta}}$$

$$X_i\theta = (x_{i1}\theta_1 + x_{i2}\theta_2 + \cdots + x_{ij}\theta_j + \cdots)$$

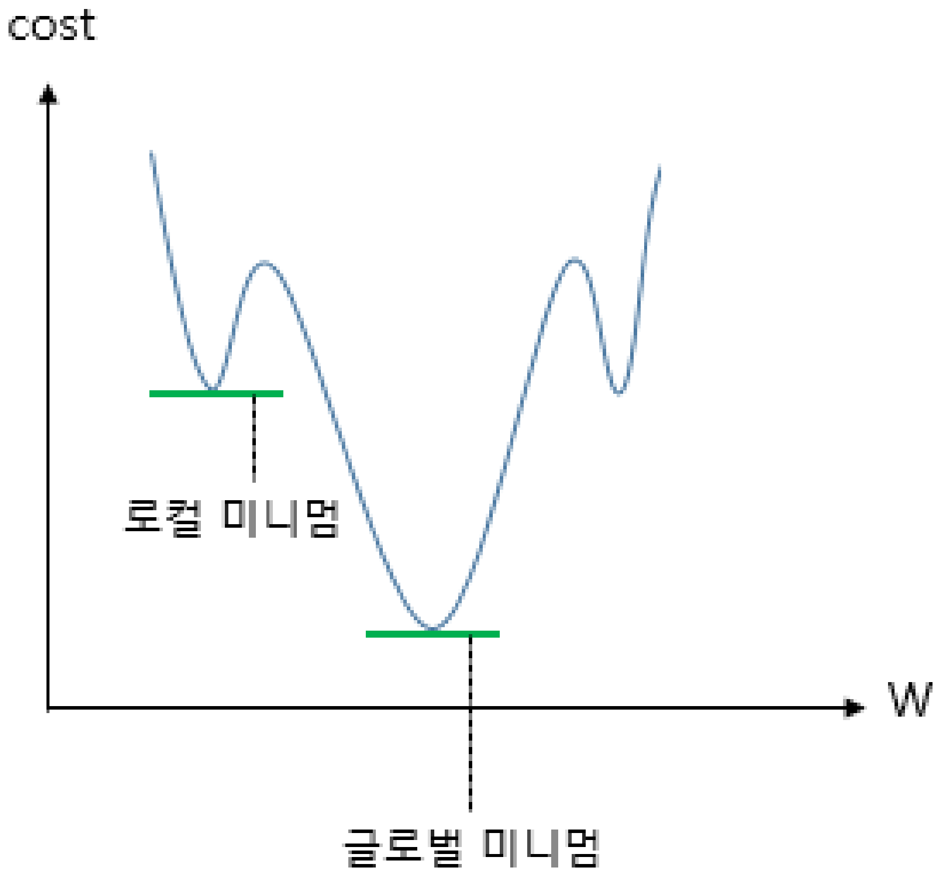


Ex) 강아지 사진인지 아닌지에 대한 분류를 한다!

X에 데이터를 대입하면 $p\,X_i$의 값으로 해당 데이터가 강아지 사진일 확률을 구해준다.

Unit 03 | Gradient Descent Algorithm

✓ 로지스틱 회귀는 MSE로 목적함수를 구해보면 어떻게 생겼을까?



✓ Likelihood

✓ 그럼 로지스틱 회귀에서는 목적함수를 어떻게 설정해야할까?
-> 로지스틱의 likelihood 함수를 응용해서 목적함수를 만들어보자!

Logistic regression

$$Y|X \sim Ber(p(X)) \qquad P(X) = \frac{\exp(X_i^T\theta)}{1+\exp(X_i^T\theta)}$$

X ~ (θ과는 무관)

$$L(Y_i|X_i;\theta) = \prod_{i=1}^{m} pdf_{X,Y}(x_i, y_i) = \prod_{i=1}^{m} pdf_{Y|X}(x_i, y_i)\, pdf_X(x_i)$$

$$f(x|y) = \frac{f(x,y)}{f_Y(y)}$$

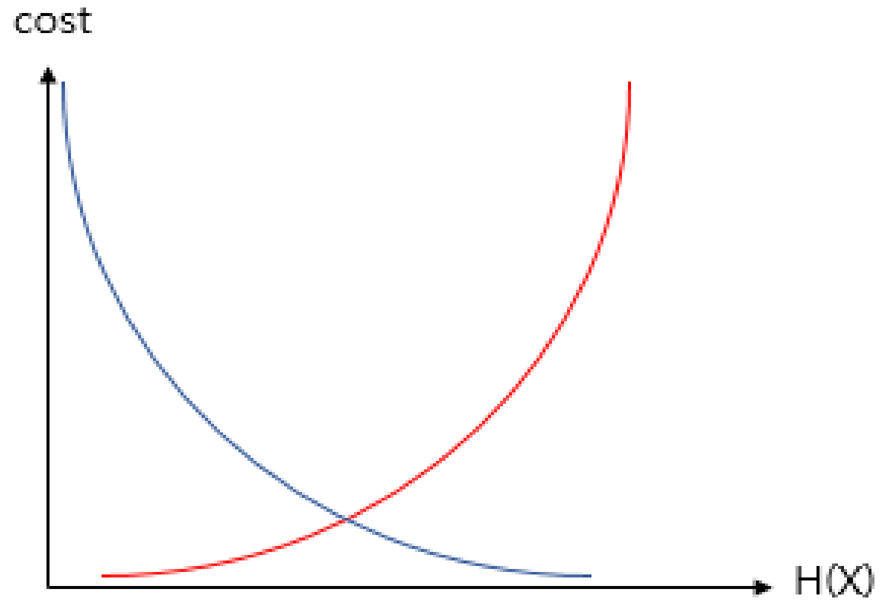
$$= \prod_{i=1}^{m} P(x_i)^{y_i}(1-P(x_i))^{1-y_i} \quad * \; (\theta\text{과는 무관})$$

$$\propto \prod_{i=1}^{n} p(x_i)^{y_i}\{1-p(x_i)\}^{1-y_i}$$

이산확률변수의 확률밀도함수(**pdf**: probability density function)를 보통 확률질량함수(**pmf**: probability mass function)라고 한다

Unit 03 | Gradient Descent Algorithm

로지스틱 likelihood 함수에 –log를 취해주면 얻게되는
$-\{y_i log p\ X_i\ +\ (1-y_i)\log(1-p\,(X_i))\}$를 그려보면



$$\mathrm{Cost}(p(x_i),y) = \begin{cases} -\log(p(x_i)) & \text{if } y=1 \\ -\log(1-p(x_i)) & \text{if } y=0 \end{cases}$$

실제값이 1일 때가 파란색, 실제값이 0일 때가 빨간색 그래프이다.
Y=1일때는 파란색 함수를, y=0일때는 빨간색 함수를 채택하니,
convex한 목적 함수(비용함수)를 얻게 된 것.

✓Logistic Regression & Gradient Descent

Logistic Regression의 목적함수 Negative Log Likelihood는 볼록하다.

$$L(X) = \prod p(X_i)^{y_i}\left(1 - p(X_i)\right)^{(1-y_i)}$$

$$l(X) = -\log L\,(X) = -\sum \{y_i \log p(X_i) + (1-y_i)\log\left(1 - p(X_i)\right)\}$$

$$p(X_i) = \frac{1}{1 + e^{-X_i\theta}} \qquad \left(= \phi(z) = \frac{1}{1+e^{-z}}\right)$$

$$\hat{\theta} = argmin_\theta l(X)$$

✓Logistic Regression & Gradient Descent

$$p(X_i) = \phi(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-X_i^T\theta}}$$

$$l(y_i|X_i;\theta) \quad = -\sum\{y_i \log p(X_i) + (1-y_i)\log(1-p(X_i))\}$$

$y_i \log p(X_i) - y_i \log(1-p(X_i)) + \log(1-p(X_i))$

$y_i \cdot \log\left(\frac{p(X_i)}{1-p(X_i)}\right) + \log(1-p(X_i))$

$$= -\sum\left\{y_i \log\frac{p(X_i)}{1-p(X_i)} + \log(1-p(X_i))\right\}$$

$$= -\sum\left\{y_i \log\frac{1}{e^{-X_i^T\theta}} - \log\left(\frac{1+e^{-X_i^T\theta}}{e^{-X_i^T\theta}}\right)\right\}$$

$$= -\sum\{y_i X_i^T\theta - \log(1+e^{X_i^T\theta})\}$$

Unit 03 | Gradient Descent Algorithm

✓Logistic Regression & Gradient Descent

$$X_i\theta = (x_{i1}\theta_1 + x_{i2}\theta_2 + \cdots + x_{ij}\theta_j + \cdots)$$

$$\frac{\partial}{\partial\theta_j} l(y_i|X_i;\theta) = -\frac{\partial}{\partial\theta_j}\sum\{y_i X_i\theta - \log(1 + e^{X_i\theta})\}$$

$$= -\sum(y_i x_{ij} - \frac{e^{X_i\theta}}{1+e^{X_i\theta}} x_{ij})$$

$$= -\sum(y_i x_{ij} - \frac{1}{1+e^{-X_i\theta}} x_{ij})$$

$$= -\sum(y_i - p_i)x_{ij}$$ = 미분값(기울기)

Unit 03 | Gradient Descent Algorithm

✓Logistic Regression & Gradient Descent

미분을 통해 방향을 결정했으니 모수들을 각각 업데이트를 해주자

입력 데이터(batch)의 개수를
고려

$$x_{t+1} \leftarrow x_t - \alpha \nabla f(x_t)$$

$$\theta_j^{t+1} \leftarrow \theta_j^t - \alpha \boxed{\frac{1}{N}} \frac{\partial}{\partial \theta_j^t} l(\theta_j^t) = \theta_j^t + \alpha \frac{1}{N} \sum (y_i - p_i) X_{ij}$$

✓ 배치(Batch)란?

한 번 기울기를 계산하여 계수를 업데이트할 때 사용하는 데이터 셋.
이때 데이터 셋의 크기(데이터 수)를 배치 크기(batch size)라고 한다.

Unit 03 | Gradient Descent Algorithm

✓ 배치(Batch)란?

- Batch Gradient Descent(BGD)

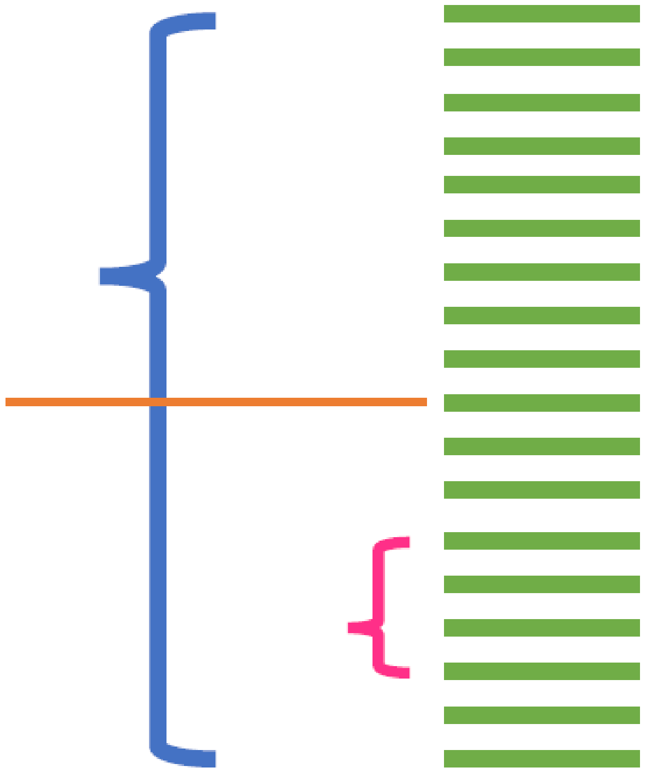
  학습 한 번에 모든 데이터셋을 사용하여 기울기를 업데이트함

- Stochastic Gradient Descent(SGD)

  학습 한 번에 임의의 1개의 데이터만 사용하여 기울기를 업데이트함

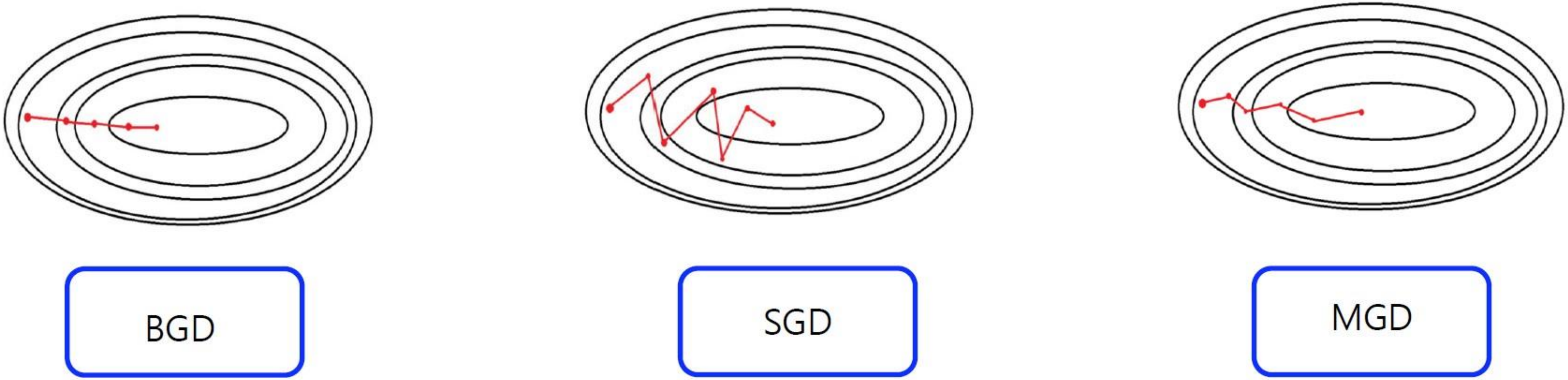
- Mini batch Gradient Descent(MGD)

  학습 한 번에 데이터셋의 일부만 사용하여 기울기를 업데이트함



✓ 배치(Batch)란?

수렴과정 시각화

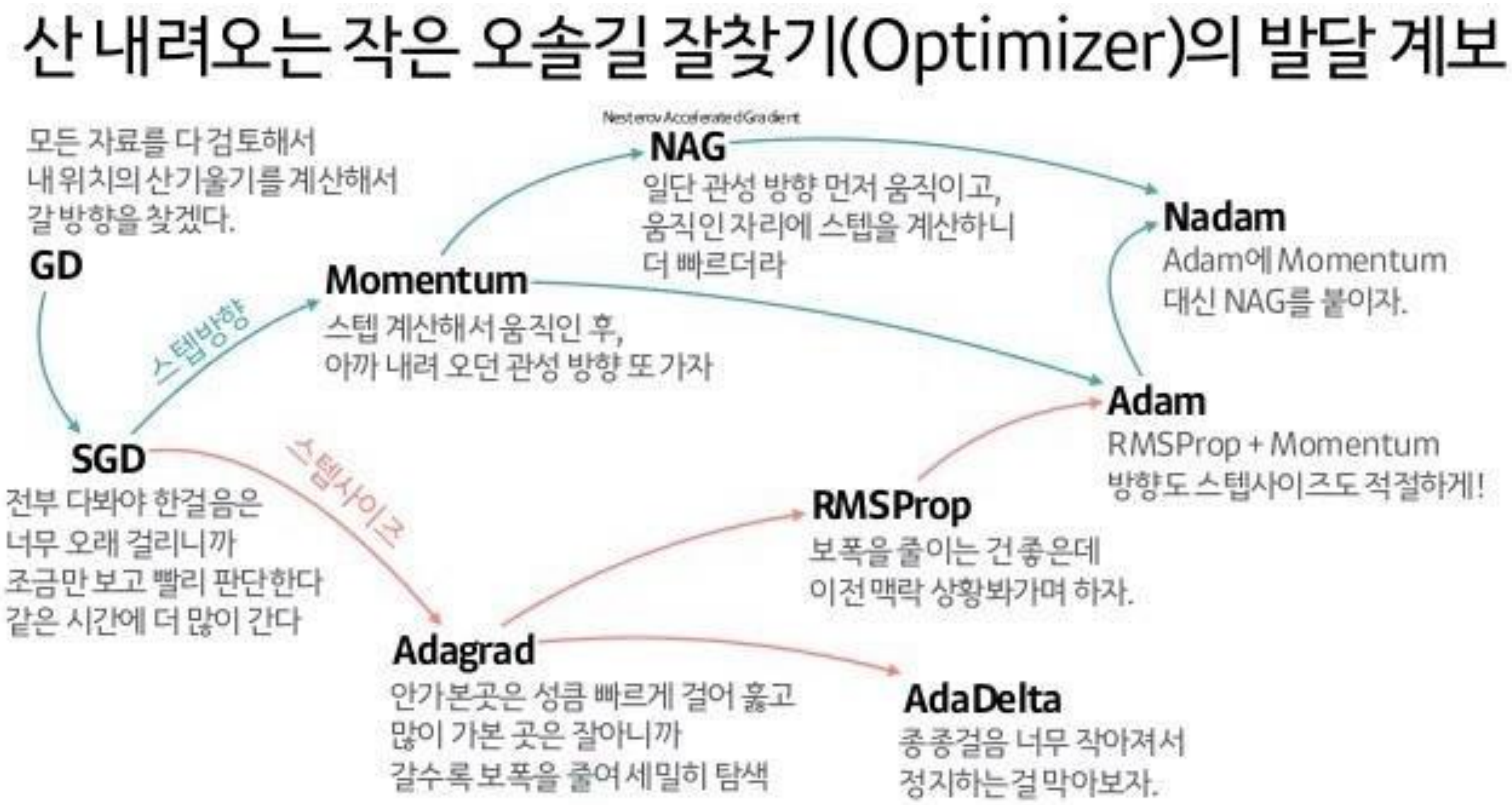


Unit 04 | Optimizer

경사하강법 이후로 어떻게 더 발전했는가

-> Optimizer들의 발전

Unit 04 | Optimizer



Unit 04 | Optimizer

α : learning rate
dw : loss의 그래디언트
ρ : 마찰계수

✓SGD

$$W^{(t+1)} = W^{(t)} - \alpha \cdot dw$$

W의 갱신값을 이전의 값에서 조금씩 내려가는 방식
적절한 Learning Rate를 구하는 것이 어렵다는 문제점이 존재한다.

✓Momentum

$$V^{(t+1)} = \rho \cdot V^{(t)} + dw$$

$$W^{(t+1)} = W^{(t)} - \alpha \cdot V^{(t+1)}$$

속도의 개념을 도입해 이전에 갔던 방향을 기억
W을 갱신할 때는 이전의 얼마나 내려가있는지를 기억하는 속도를 통해
많이 내려갔다면 그내려간 방향으로 조금 더 가려는 관성

✓Adagrad

α : learning rate
dw : loss의 그래디언트

$$g^{(t+1)} = g^{(t)} + dw \cdot dw \qquad W^{(t+1)} = W^{(t)} - \alpha \cdot \frac{dw}{\sqrt{g^{(t+1)} + \epsilon}}$$

Parameter 중 값의 변화가 큰 것에 대해서는 Step size를 작게 하여 근사치로 빠르게 수렴하게 만들고,
값의 변화가 적은 것에 대해서는 Step Size를 크게하여 세밀하게 값의 변화를 확인하는 방식이다

✓RMSProp

$$g^{(t+1)} = \beta \cdot g^{(t)} - (1-\beta) \cdot dw \cdot dw \qquad W^{(t+1)} = W^{(t)} - \frac{\alpha \cdot dw}{\sqrt{g^{(t+1)} + \epsilon}}$$

기울기를 단순 누적하지 않고 지수 가중 이동 평균를
사용하여 최신 기울기들이 더 크게 반영되도록 하였다.

✓Adam

α : learning rate
dw : loss의 그래디언트
ρ : 마찰계수

$$V^{(t+1)} = \beta_1 \cdot V^{(t)} + (1-\beta_1) \cdot dw$$ (1차 모멘트)

$$g^{(t+1)} = \beta_2 \cdot V^{(t)} + (1-\beta_2) \cdot dw \cdot dw$$ (2차 모멘트)

$$W^{(t+1)} = W^{(t)} - \alpha \cdot \frac{V^{(t+1)}}{\sqrt{g^{(t+1)}} + \epsilon} \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t}$$

진행하던 속도에 관성을 주고, 최근 경로의 곡면의 변화량에 따른 적응적 학습률을 갖는 알고리즘이다.
매우 넓은 범위의 아키텍처를 가진 서로 다른 신경망에서 잘 작동한다는 것이 증명되어, 일반적 알고리즘에 현재 가장 많이 사용되고 있다.
.

Unit 04 | Optimizer

- ✓ Complete ‘wk2_optimization_assignment.ipynb’

1. 빈칸을 채워주세요! (마크다운,코드)
2. 완성된 함수로 주어진 데이터에 대해 gradient descent를 진행해주세요!
3. 완성된 코드에 대해 상세하게 주석을 달아주세요!

*과제에 대해 이해가 안되거나 잘 안된다면 연락 부탁드립니다.

Tobig's 13기 이지용님 자료

Tobig's 14기 오주영님 자료

Tobig's 16기 김건우님 자료

Tobig's 17기 유현우님 자료

https://www.youtube.com/watch?v=vMh0zPT0tLI

https://velog.io/@regista/%EB%B9%84%EC%9A%A9%ED%95%A8%EC%88%98Cost-Function-%EC%86%90%EC%8B%A4%ED%95%A8%EC%88%98Loss-function-%EB%AA%A9%EC%A0%81%ED%95%A8%EC%88%98Objective-Function-Ai-tech

https://angeloyeo.github.io/2020/07/17/MLE.html

https://daebaq27.tistory.com/35

https://gosamy.tistory.com/240

https://mazdah.tistory.com/783

https://www.youtube.com/watch?v=CzeOFc9ngwo&t=6s

https://www.youtube.com/watch?v=XepXtl9YKwc&t=267s

https://kite-mo.github.io/2020/03/09/logistic/

https://www.youtube.com/watch?v=sDv4f4s2SB8&t=1090s

https://ratsgo.github.io/convex%20optimization/2017/12/25/convexset/

https://velog.io/@idj7183/Optimizer-%EB%B0%9C%EC%A0%84-%EC%97%AD%EC%82%AC

https://www.youtube.com/watch?v=9DrEYpGuxfo&list=PLgIRZO0FZ91ysyIVniyqMTxvlisY-aIPP&index=9