22기 정규세션

ToBig's 21기 김창현
22기 우단비

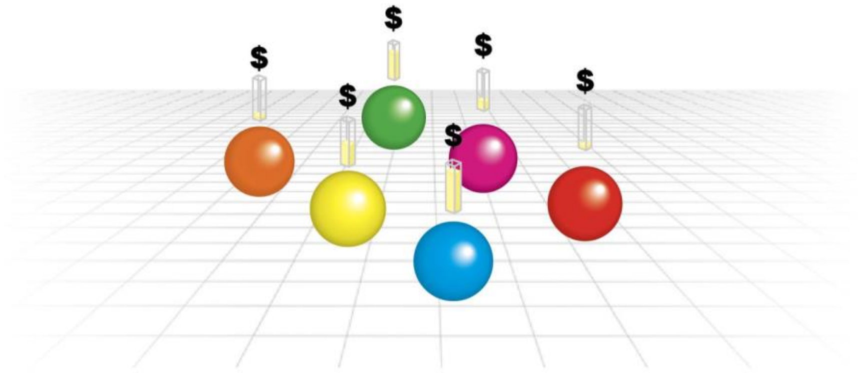
최적화
Optimization

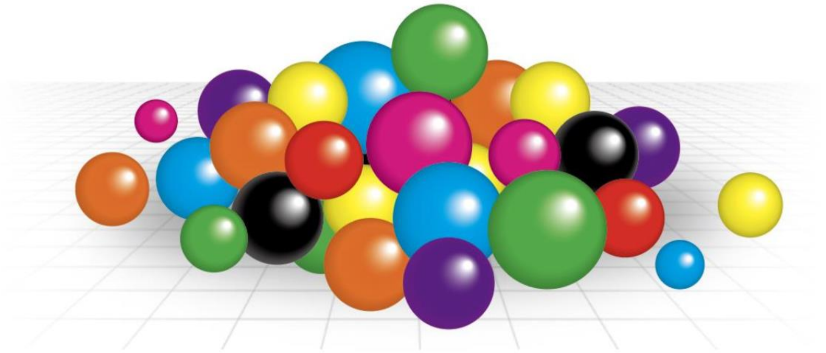
Contents

01 | What is Optimization?

Unit 01 | What is Optimization?

최적화(Optimization)이란?





Unit 01 | What is Optimization?

최적화(Optimization)이란?

- 주어진 문제에서 특정한 목표를 가장 잘 달성할 수 있도록 하는 과정!
- Machine Learning에서의 최적화 : 모델의 성능을 극대화하거나 손실을 최소화하기 위해 모델의 매개변수를 조정하는 과정

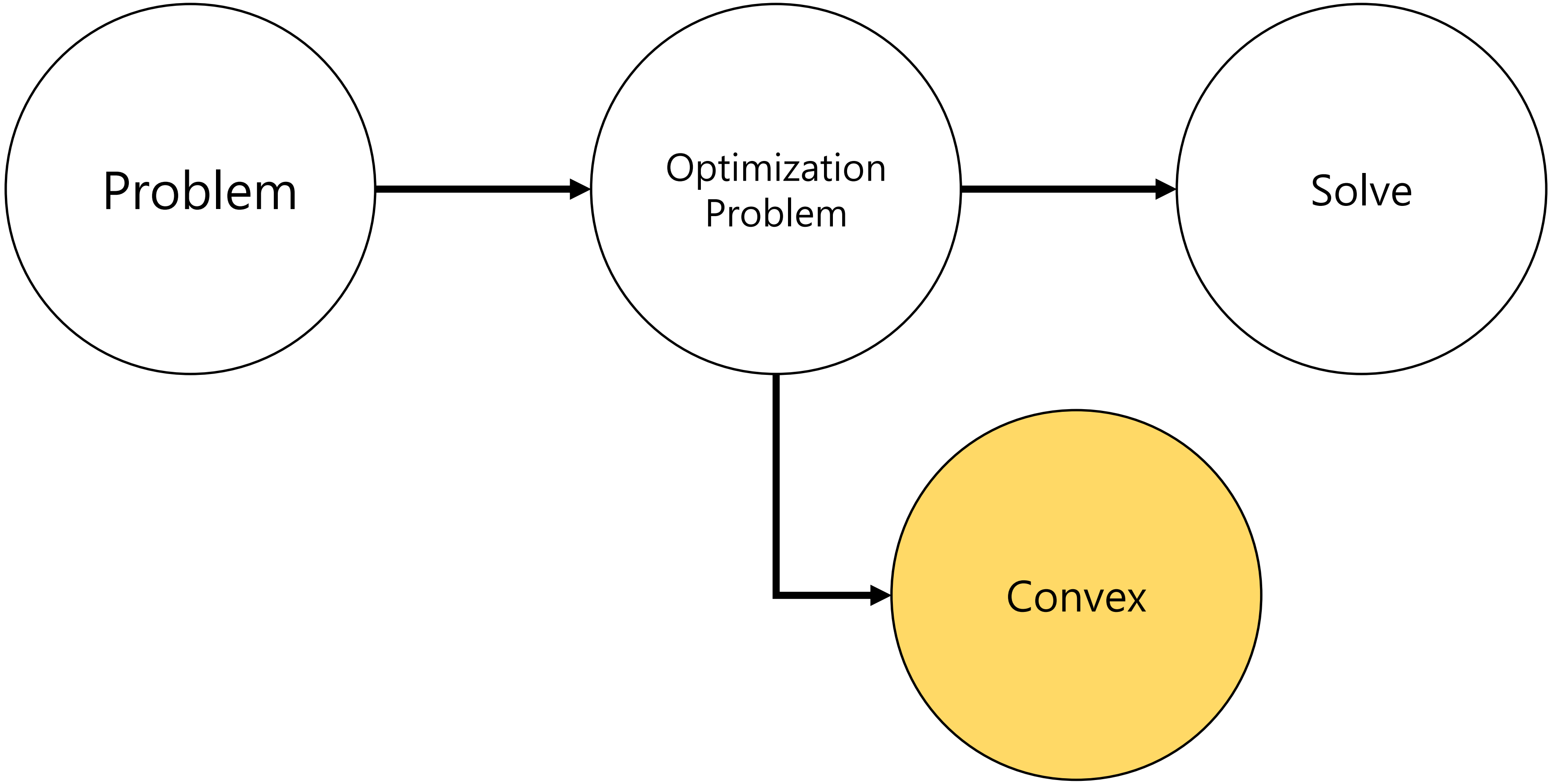
$$\min_{x \in R^n} f(x)$$

$$h_i(X) = 0 \qquad i = 1, \ldots$$

$$g_j(X) \leq 0 \qquad j = 1, \ldots$$

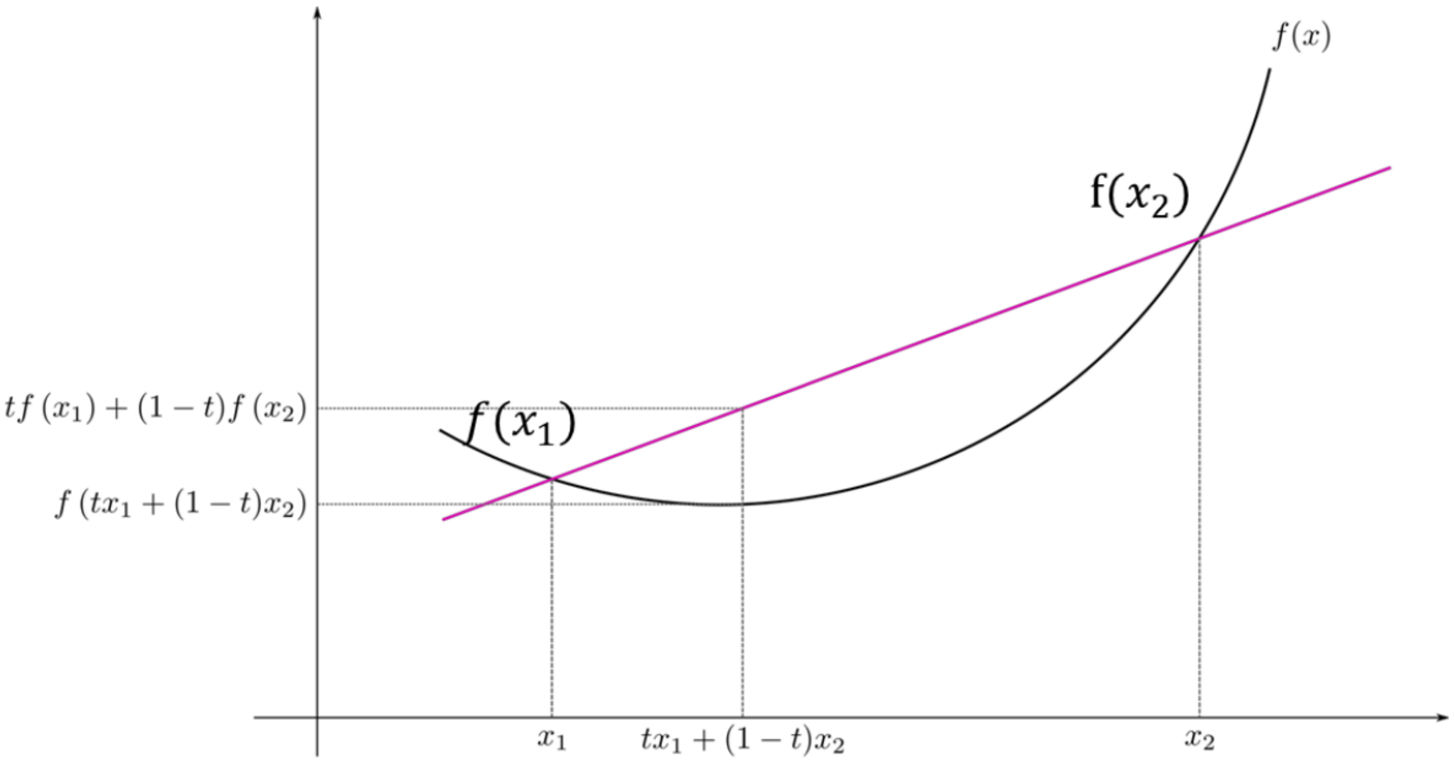
1. Cost Function
2. Decision Variable
3. Equality Constraints
4. Inequality Constraints

Unit 01 | What is Optimization?



02 | Convex function

☑ 왜 Convex Function을 찾는가? : 내가 풀려는 문제가 정말 정답이 있는 함수인지 알아야한다!



Unit 02 | Convex Function

1. $Convex\ Function$: $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 가 주어졌을 때, f 의 정의역을 $\mathrm{dom}(f)$ 라고 하자!

함수 f 가 Convex 라고 하려면 다음 조건을 만족해야한다.

1. $\mathrm{dom}(f)$ 가 Convex 다.
2. $dom(f)$ 의 모든 $\mathrm{x, y}$ 그리고 $\theta \in [0,1]$에 대하여 다음을 만족해야한다.

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y)$$

- 만약 $0 < \theta < 1$ 일 때 위의 부등식이 항상 성립한다면 $f(x)$ 를 Strictly Convex 함수라고 부른다.

- Convex Function $f(x)$는 정의역의 내부에서 항상 연속이다.

Convex Function : 그냥 오목함수!

Unit 02 | Convex Function

1. Convex Function의 특성

(1) (First Order Condition) : 만약 f 가 미분 가능하다면 모든 f 의 정의역에서 다음이 성립한다.

$$f(y) \geq f(x) + \nabla f(x)^{\top}(y - x)$$

(2) (Second order condition) : 만약 f가 두번 미분 가능하다면 $\nabla^2 f(x)$는 항상 양정치이다. (Positive definite)

1. Convex Function의 특성 – (1)

$f((1-h)x+hy) = f(x+h(y-x))$ for $h \in (0,1)$ and

(미분 계수) ------ $$\lim_{h \to 0} \frac{f(x+h(y-x))-f(x)}{h} = \nabla f(x)^\top (y-x).$$

In addition,

(convex의 정의) ------ $$\frac{f((1-h)x+hy)}{h} \le \frac{(1-h)f(x)+hf(y)}{h}.$$

Thus,

$$f(y) - f(x) \ge \nabla f(x)^\top (y-x).$$

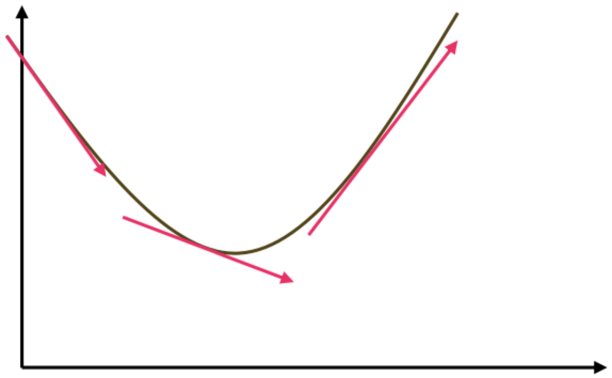
1. Convex Function의 특성 – (2)

- 2차 미분이 모든 정의역에 대해 항상 양의 값을 가진다면 임의의 함수는 Convex Function이라고 할 수 있다

- Mathematical formulation

$$f''(x) \geq 0 \quad \text{for all} \quad x$$

함수의 도함수는 단조 증가함!



Unit 02 | Convex Function

Q. Convex Function을 loss로 정의했을 때 어떻게 minimize 할 것인가?

1. Gradient Descent Method (경사하강법)
2. Newton-Raphson Method (뉴턴-랩슨 방법)
3. Prima-Dual Method
4. Interior-Point Method
5. Conjugate Gradient Method
6. ADMM Method
7. .... etc

Unit 02 | Convex Function

1. Gradient Descent Method

Q. Gradient Descent Algorithm이란?

- 주어진 함수의 최솟값을 찾기 위한 반복적인 최적화 알고리즘.
- 함수의 기울기를 이용해 현재 위치에서의 경사를 따라 내려가면서 최솟값에 접근.

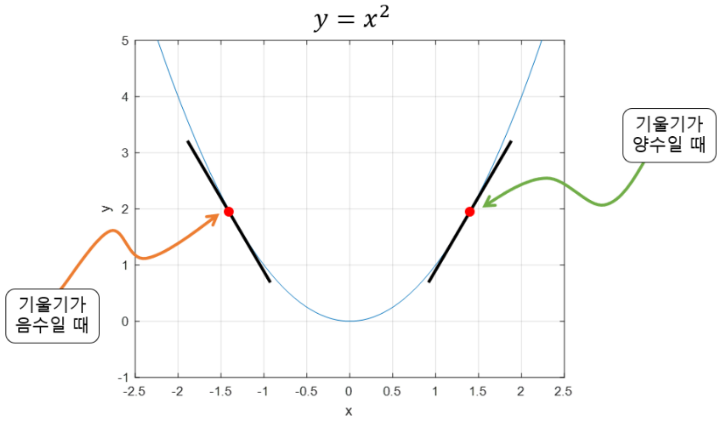
1. **초기화**: 매개변수 θ를 임의의 값으로 초기화

2. **기울기 계산**: 현재 매개변수에서 목적 함수의 기울기 $\nabla_\theta J(\theta)$를 계산

3. **매개변수 업데이트**: 매개변수를 기울기의 반대 방향으로 업데이트

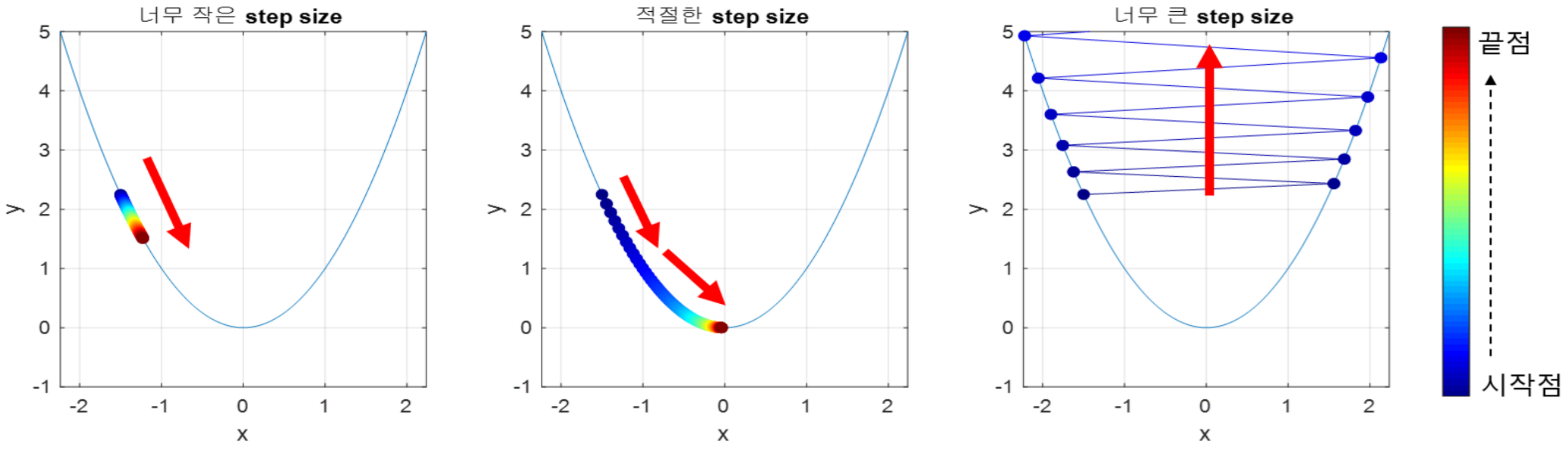
$$\theta := \theta - \alpha \nabla_\theta J(\theta) \qquad (\alpha : \text{step size})$$

4. **수렴 검사**: 목적 함수의 값이 더 이상 크게 변하지 않거나, 기울기의 크기가 매우 작아지면 알고리즘을 종료



1. Gradient Descent Method

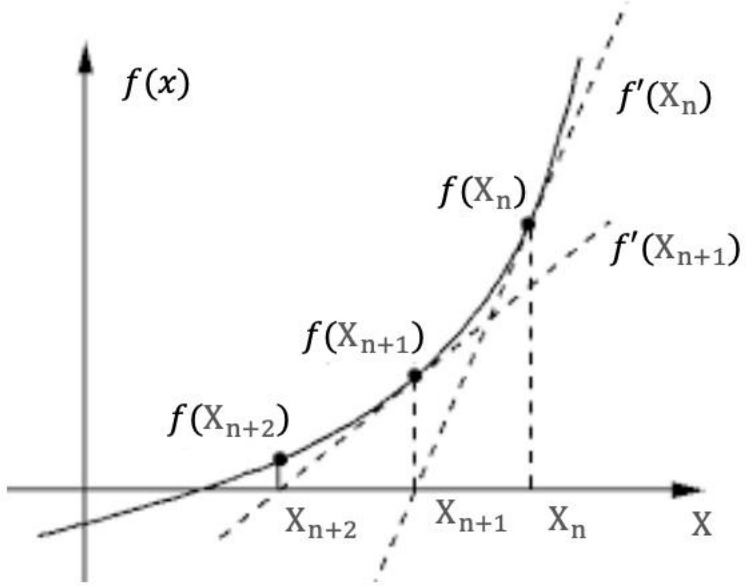
☑ Importance of step size



2. Newton-Raphson Method

Q. Newton-Raphson Method란?

- 정의 : 두 번 미분가능한 함수에 대하여 second-order Taylor expansion으로 함수를 근사한 뒤,
  근사 함수의 최솟값을 찾으며 해에 접근하는 방법
- 장점1 : Gradient descent에 비하여 무척 빠른 수렴속도를 보임
- 장점2 : Learning rate의 값을 정확히 제시한 방법
- 단점 : 2차 미분을 사용하기 때문에 연산량이 큼



2. Newton-Raphson Method

Q. Newton-Raphson Method란?

1.초기 추정값 설정 (Initial Guess): 함수 $f(x)$의 근을 찾기 위해 초기 추정값 x_0을 설정

2.반복 계산 (Iterative Calculation): 각 반복 단계에서 다음 식을 사용하여 새로운 추정값을 계산

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

3. 수렴 조건 확인 (Convergence Check):

1. 수렴 조건을 만족할 때까지(예: $|x_{n+1} - x_n|$이 매우 작을 때) 반복을 계속하기
2. 보통 함수 값 $|f(x_{n+1})|$이 충분히 작은 경우에도 반복을 멈춤

2. Newton-Raphson Method

Q. Newton-Raphson Method란?

- 2차 미분항을 실제로 계산하여 quadratic approximation을 수행하고, update step을 진행
- Learning rate의 설정이 필요 없음

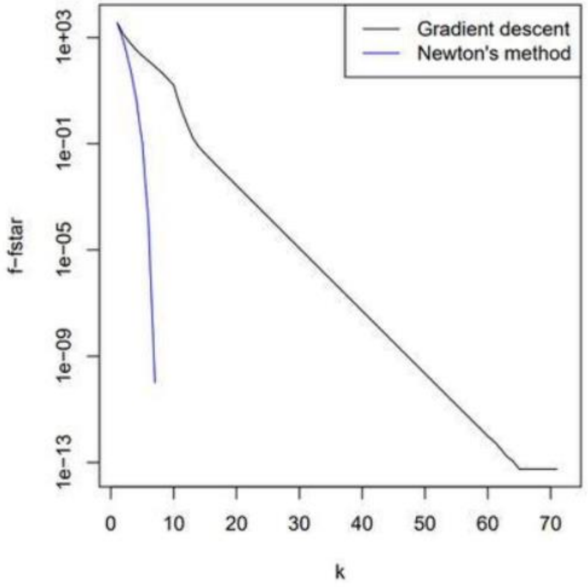
$$\min_x f(x) \Rightarrow$$

$$choose\ initial\ x^{(0)} \in \mathbb{R}^n,$$

$$x^{(k)} = x^{(k-1)} - \boxed{\left(\nabla^2 f\left(x^{(k-1)}\right)\right)^{-1}} \nabla f\left(x^{(k-1)}\right), \qquad k = 1,2,3 \cdots$$

☑ Gradient descen와 Newton's method 사이의 비교

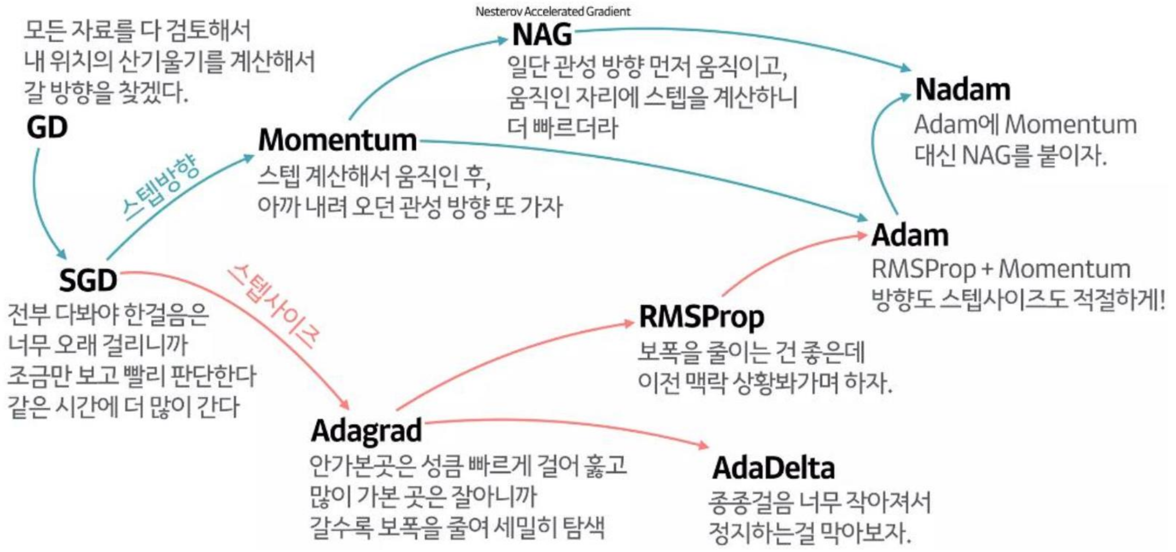
- Gradient descent는 미분 항을 정방행렬에 상수가 곱해진 값으로 가정하고 Gradient를 계산하기 때문에, 등고선(counter)의 접선 방향에 수직하게(perpendicular) 수렴함을 확인할 수 있고, Newton's method에 비해 느린 수렴 속도를 보임.

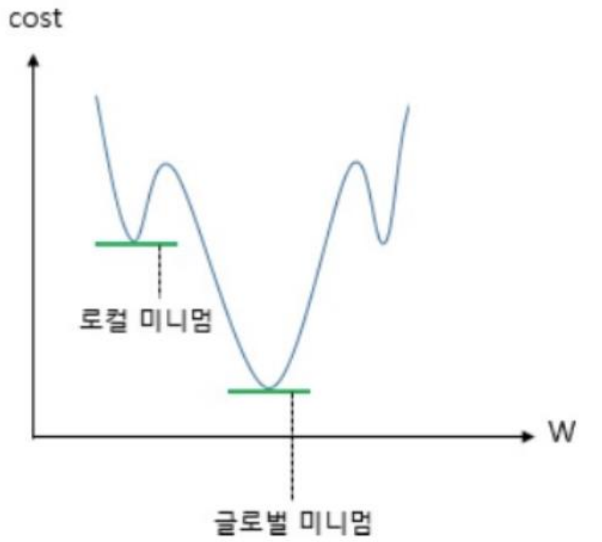


03 | Optimizer

- 딥러닝에 사용되는 다양한 Optimizer

- 우리가 풀려는 문제 Loss Function을 최적화하고자 함
- ex. MSE : 최대한 답과 같도록 / Cross Entropy : 최대한 분류를 잘하도록
- 그러나 딥러닝의 Loss Function은 상상 이상으로 복잡
- Local Minimum을 피해 Global Minimum에 도달하기 위해 Optimizer가 발전





1. Minimum에 도달하기 위한 과정

1. **초기화**: 매개변수 θ를 임의의 값으로 초기화

2. **기울기 계산**: 현재 매개변수에서 목적 함수의 기울기 $\nabla_\theta J(\theta)$를 계산

3. **매개변수 업데이트**: 매개변수를 기울기의 반대 방향으로 업데이트

$$W_{t+1} = W_t - \alpha \nabla f(W_t)$$ (α : step size)

4. **수렴 검사**: 목적 함수의 값이 더 이상 크게 변하지 않거나, 기울기의 크기가 작아지면 알고리즘 종료

1. SGD (Stochastic Gradient Descent)

$$W^{(t+1)} = W^{(t)} - \eta\, dw$$ (η : learning rate(학습률), dw: 기울기)

- **계산 효율성** : 전체 데이터셋을 사용하는 대신 미니배치를 사용하므로
  메모리 사용량이 적고 계산 속도가 빠름
- **빠른 수렴** : 데이터의 샘플링에 의해 더 다양한 경로로 최적화 진행
- **단점** : 자주 진동하거나 국소 최적해에 머물러 있을 수 있음

2. Momentum

$$v_t = \beta v_{t-1} + dw$$ --------- Momentum 파트

$$w_t = w_{t-1} - \alpha v_t$$ ---------------- Mementum 고려하여 가중치 갱신

- Momentum : 관성을 의미, 가던 방향을 유지하는 개념
- ν : 이전 결과가 누적될수록 β가 제곱되어서 값이 커짐
- 이에 Local minimum이나, Minimum이 아닌데 미분계수가 0인 지점(안정점)에 멈추지 않고 기존에 가던 방향을 유지하며 업데이트를 계속할 수 있음
- 그러나 global minimum에서도 멈추지 못하고 지나칠 수 있음

3. RMSProp

$$h_t = \rho h_{t-1} + (1-\rho)dw \cdot dw$$ ---------- 지수 이동평균으로 이전 기울기 반영

$$w_t = w_{t-1} - \gamma \frac{1}{\sqrt{h_t}} dw$$ ----------------- 가중치 업데이트

- ρ를 크게 할 경우 : 이전까지의 학습률 감쇠(decay)를 중시
- ρ를 작게 할 경우 : 현재 gradient를 중시
- 기울기를 단순 누적하지 않고 지수 가중이동 평균을 사용해 최신 기울기를 더 크게 반영

4. Adam(Adaptive Moment Estimation)

$$v_t = \beta v_{t-1} + dw$$ ------------------ Momentum

$$h_t = \rho h_{t-1} + (1-\rho) dw \cdot dw$$ ------ 최신 기울기 반영

$$w_t = w_{t-1} - \frac{\alpha v_t}{\sqrt{h_t}}$$ ----------------- 가중치 업데이트

- Momentum처럼 진행하던 방향을 유지하려는 관성을 따름
- Momentum과 RMSProp의 장점을 결합하여 딥러닝에서 일반적으로 가장 많이 사용됨
- 이후 Nadam, Radam, AdamW 등 더욱 우수한 optimizer 등장

04 | Lagraingian & KKT condition

Unit 04 | Lagraingian & KKT condition

0. Notation

☑ 딥러닝의 복잡한 Loss function말고 실생활에서 나올 수 있는 문제를 최적화해보자!

1. p^* : 우리가 찾으려는 최적의 값

2. $f(x)$: 우리가 최적의 값을 찾고자 하는 함수

3. $\mathrm{h}(x)$: 등제약조건 (Equality Constraint) (ex. $x + y = 0$)

4. $\mathrm{g}(x)$: 부등제약조건 (Inequality Constraint) (ex. $x + y < 0$)

5. λ, μ : 라그랑주 승수 , 제약조건을 다루는데 이용하는 역할

1. Normal Convex function

☑ 제약조건이 없는 일반적인 최적화

- 대표적인 형태 : $p^\star = \min_{\mathbf{x}} f(\mathbf{x})$

- Local 최소점이 되기 위한 필요 조건 : $\nabla_{\mathbf{x}} f(\mathbf{x}^\star) = 0$

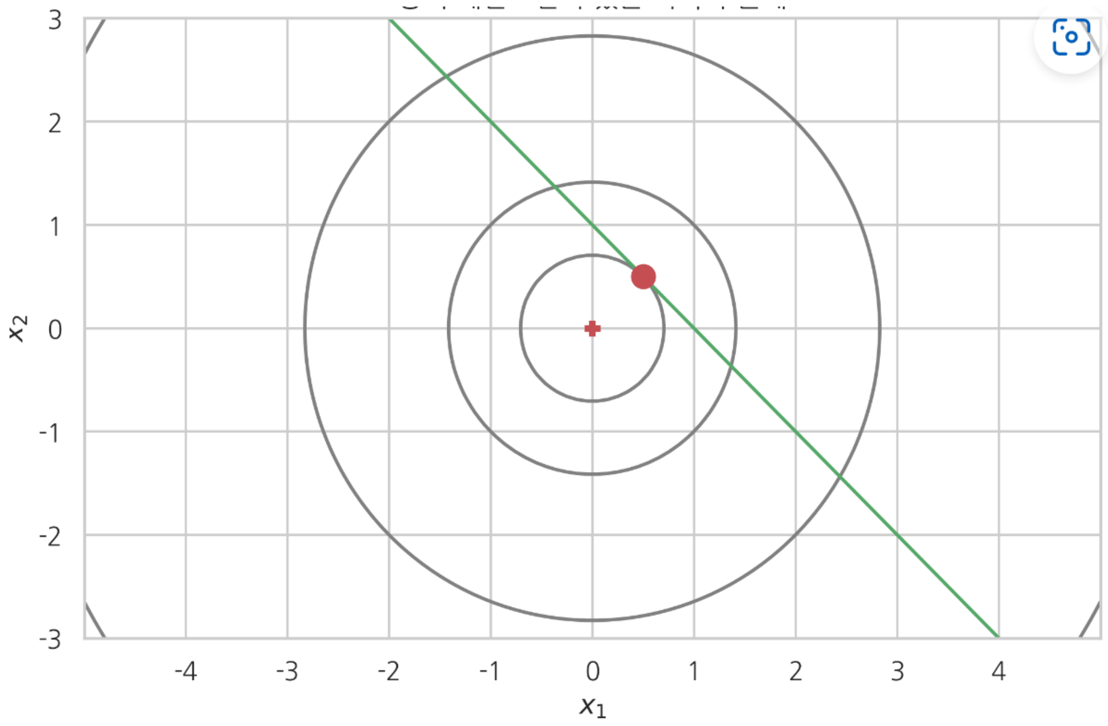
- **예시 : $f(x) = x^2 - 2x + 1$, $f(x)$의 최소값은 $x^* = 1$일 때 $p^* = 0$**

- **세상의 문제가 이렇게 단순할까?**

2. Equality Constrain function

$$f(x_1, x_2) = x_1^2 + x_2^2$$

$$h(x_1, x_2) = x_1 + x_2 - 1 = 0$$



☑ 함수 $f(x_1, x_2)$는 원점에 가까울 수록 $Minimum$ 이다.

예시)
$(0, x_1)$, $(x_2, 0)$ 위치에 도시를 건설하려고 한다.
원점으로부터 두 도시의 거리의 합은 1이어야하고(등제약조건)
두 도시의 거리를 최소화 하고자 한다(목적함수)
거리는 피타고라스 정리의 의해 $f(x_1, x_2) = x_1^2 + x_2^2$ 로 정의된다!

2. Equality Constrain function

- 등제약조건 (Equality Constrain)이 있는 최적화 문제

$$p^\star = \min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to} \quad h_j(\mathbf{x}) = 0, \;\; j = 1, \ldots, k$$

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_k(\mathbf{x}) \end{bmatrix} = 0$$

☑ 라그랑주 승수법을 이용해 제약조건이 없는 문제로 바꿔서 풀자!

3. Lagrange multiplier

$$p^\star = \min_{\mathbf{x},\lambda_1,\ldots,\lambda_k} L(\mathbf{x},\lambda_1,\lambda_2,\ldots,\lambda_k)$$

$$= \min_{\mathbf{x},\lambda} L(\mathbf{x},\lambda)$$

where

$$L(\mathbf{x},\lambda) = f(\mathbf{x}) + \sum_{j=1}^{k} \lambda_j h_j(\mathbf{x})$$

$$= f(\mathbf{x}) + \lambda^T \mathbf{h}(\mathbf{x})$$

☑ 원래 해와 바꾼 해가 같을 조건

$$\nabla_{\mathbf{x}} L(\mathbf{x}^\star,\lambda^\star) = 0$$

$$\nabla_{\lambda} L(\mathbf{x}^\star,\lambda^\star) = 0 \ (\text{or } h_j(\mathbf{x}^\star) = 0,\ j = 1,\ldots,k)$$

☑ 예시

$$\min_{x,y} f(x,y) = x + y$$

$$subject\ to\ g(x,y) = x^2 + y^2 - 1 = 0$$

$$L(x,y,\lambda) = f(x,y) + \lambda \cdot g(x,y)$$

4. Inequality Constraint function

- 부등제약조건이 존재하는 경우도 존재한다!

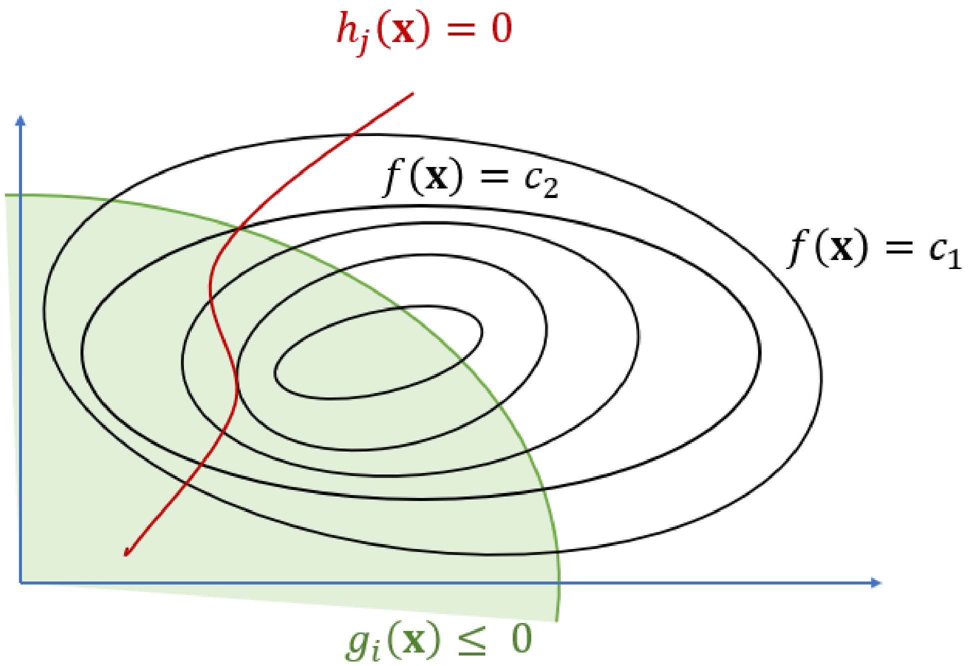
$$p^\star = \min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to} \quad g_i(\mathbf{x}) \le 0, \quad i = 1, \ldots, m$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, k$$

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ \vdots \\ g_m(\mathbf{x}) \end{bmatrix} \le 0$$

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_k(\mathbf{x}) \end{bmatrix} = 0$$



예시)
자원 1을 두 파트로 나눠서 A기계 / B기계에 넣자
각 기계에서 발생하는 오염물질은 각 파트에 들어간 양의 제곱만큼 나온다
자원은 무조건 양수여야한다.
오염물질의 양을 최소화 하자!

4. Inequality Constraint function

- Inequality Constraint 또한 라그랑지안으로 표현하여 해결

$$L(\mathbf{x}, \mu, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i g_i(\mathbf{x}) + \sum_{j=1}^{k} \lambda_j h_j(\mathbf{x})$$

- Equality Constraint와 마찬가지로 Gradient가 0이 되는 점이 최적해가 될 후보이다.

$$\min_{x,y} x^2 + y^2$$

$$\text{subject to :} \quad x^2 + y^2 - 5 \le 0$$

$$-x \le 0$$

$$-y \le 0$$

$$x + 2y - 4 = 0$$

$$L(x, y, \mu_1, \mu_2, \mu_3, \lambda) =$$

$$x^2 + y^2 + \mu_1(x^2 + y^2 - 5) - \mu_2 x - \mu_3 y + \lambda(x + 2y - 4)$$

☑ 특정 조건을 만족해야 최적의 해이다.

Unit 04 | Lagraingian & KKT condition

5. KKT(Karush-Kuhn-Tucker) condition

특정 조건을 만족해야 최적의 해라고 할 수 있다!

1. Stationarity (정지 조건): 라그랑지안 함수 $L(x, \lambda, \mu)$ 를 최적화 변수 x에 대해 편미분한 값이 0이어야 함.

$$\nabla_x L(x, \lambda, \mu) = \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) + \sum_{j=1}^{k} \mu_j \nabla h_j(x) = 0$$

2. Primal feasibility (원 문제의 실현 가능성): $g_i(x) = 0 \quad \text{for } i = 1, \ldots, m \qquad h_j(x) \leq 0 \quad \text{for } j = 1, \ldots, k$

-원래의 제약 조건을 만족해야 함.

3. Dual feasibility (쌍대 문제의 실현 가능성): $\mu_j \geq 0 \quad \text{for } j = 1, \ldots, k$

(Primal & Dual Problem)

-KKT 승수 μ_j가 0 이상이어야 함.

4. Complementary slackness (상호 배타 조건): $\mu_j h_j(x) = 0 \quad \text{for } j = 1, \ldots, k$

- 부등식 제약 조건과 KKT 승수의 곱이 0 이상이어야 함.

6. Pirmal & Dual Problem

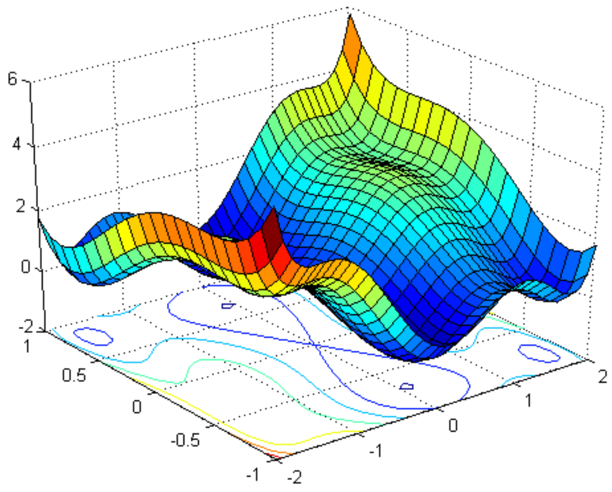
원본 문제보다 반드시 작은 함수를 찾아 그 함수를 최대화 하자!

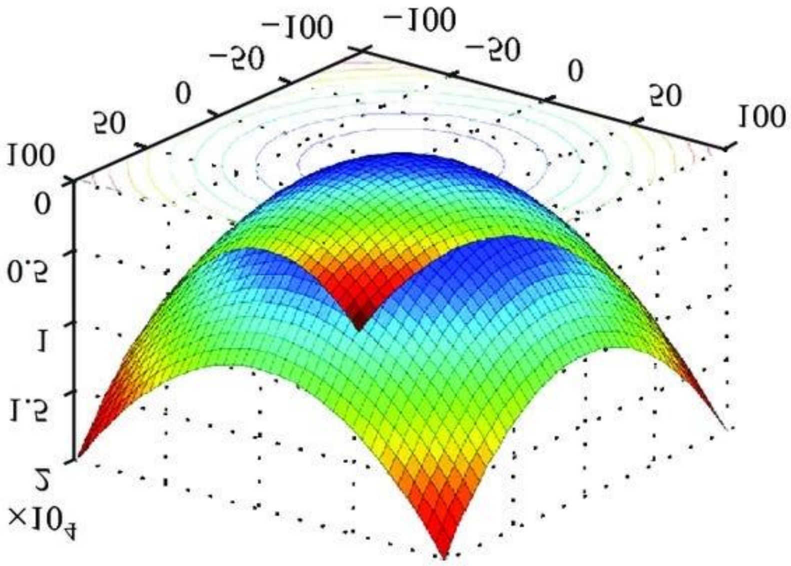
- $L(\mathbf{x}, \mu, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i g_i(\mathbf{x}) + \sum_{j=1}^{k} \lambda_j h_j(\mathbf{x})$ -------------- Original Lagrangian Problem
- $d(\mu, \lambda) = \min_{\mathbf{x}} L(\mathbf{x}, \mu, \lambda) = \min_{\mathbf{x}} \left(f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i g_i(\mathbf{x}) + \sum_{j=1}^{k} \lambda_j h_j(\mathbf{x}) \right)$ --- Dual Function
- $f(\mathbf{x}) \geq L(\mathbf{x}, \mu, \lambda) \geq \min_{\mathbf{x}} L(\mathbf{x}, \mu, \lambda) = d(\mu, \lambda)$ (정의상 Infimum, 편의상 min으로 진행)

☑ 왜 굳이 $d(\mu, \lambda)$ 로 바꿔서 하는가 ? -> 모든 Dual Function은 Concave이다.

6. Pirmal & Dual Problem

☑ 왜 굳이 $d(\mu, \lambda)$ 로 바꿔서 하는가 ? -> 모든 Dual Function은 Concave이다.





6. Pirmal & Dual Problem

☑ $f(x)$ 보다 반드시 작은 함수의 최댓값이 $f(x)$의 최저점일까?

[KKT Condition for Dual Problem]

1. Stationarity (정지 조건): 라그랑지안 함수 $L(x, \lambda, \mu)$ 를 최적화 변수 x에 대해 편미분한 값이 0이어야 함.

2. Primal feasibility (원 문제의 실현 가능성): 원래의 제약 조건을 만족해야 함.

3. Dual feasibility (쌍대 문제의 실현 가능성): KKT 승수 μ_j가 0 이상이어야 함.

4. Complementary slackness (상호 배타 조건): 부등식 제약 조건과 KKT 승수의 곱이 0 이상이어야 함.

Unit 04 | Lagraingian & KKT condition

6. Pirmal & Dual Problem

☑ 만약 조건을 만족하지 않는다면!

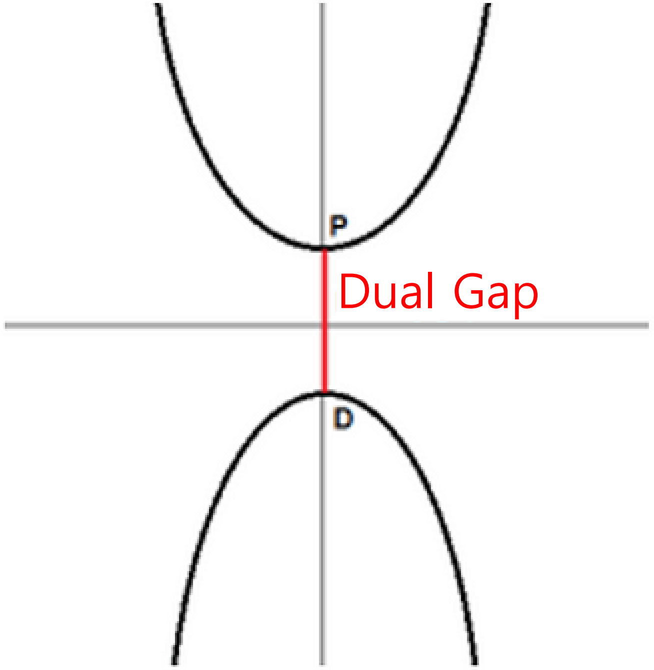


Figure 1: Weak Duality due to the Duality Gap.

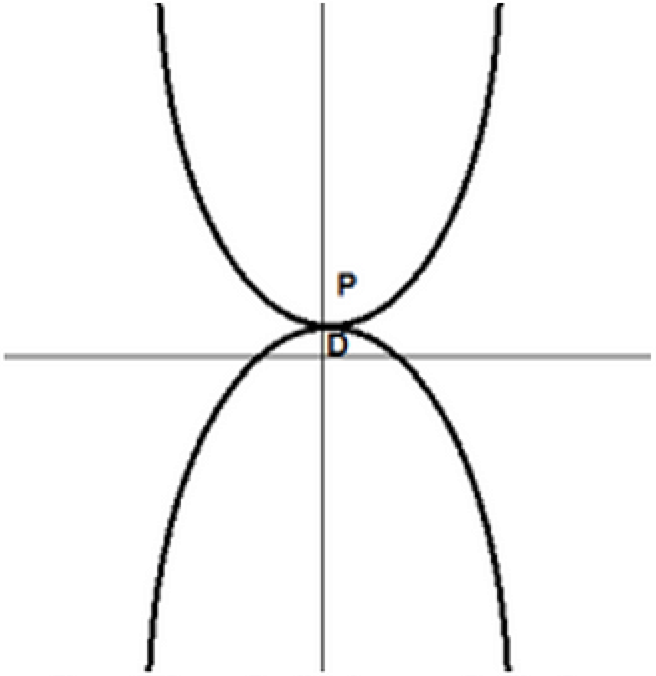


Figure 2: Strong Duality due to no Duality Gap
(Exhibits Complementary Slackness)

- **KKT condition을 적용한 최적화 example**

문제

$$\min_{x,y} x^2 + y^2$$

$$\text{subject to : } x^2 + y^2 - 5 \leq 0$$

$$-x \leq 0$$

$$-y \leq 0$$

$$x + 2y - 4 = 0$$

제약조건 (g, h)

$$x + 2y - 4 = 0$$

$$x^2 + y^2 - 5 \leq 0$$

$$-x \leq 0$$

$$-y \leq 0$$

라그랑주

$$L(\mathbf{x}, \mu, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i g_i(\mathbf{x}) + \sum_{j=1}^{k} \lambda_j h_j(\mathbf{x})$$

$$L(x, y, \mu_1, \mu_2, \mu_3, \lambda) = x^2 + y^2 + \mu_1(x^2 + y^2 - 5) - \mu_2 x - \mu_3 y + \lambda(x + 2y - 4)$$

• **KKT condition을 적용한 최적화 example**

KKT Condition

1. stationarity:

$$\frac{\partial L}{\partial x} = 2x + 2x\mu_1 - \mu_2 + \lambda = 0$$

$$\frac{\partial L}{\partial y} = 2y + 2y\mu_1 - \mu_3 + 2\lambda = 0$$

2. primal constraints:

$$x^2 + y^2 - 5 \leq 0$$

$$-x \leq 0$$

$$-y \leq 0$$

$$x + 2y - 4 = 0$$

3. dual constraints: $\mu_1, \mu_2, \mu_3 \geq 0$

4. complementary slackness:

$$\mu_1(x^2 + y^2 - 5) = 0$$

$$\mu_2 x = 0$$

$$\mu_3 y = 0$$

조건을 만족해야
최적의 해

↓

다양한 경우의 수를
따져보자!

- **KKT condition을 적용한 최적화 example**

(1) $\mu_1 = \mu_2 = \mu_3 = 0 \rightarrow x = \frac{4}{5}, y = \frac{8}{5}, \lambda = -\frac{8}{5}$

(2) $\mu_1 = \mu_2 = 0 \rightarrow y = 0, x = 4 \rightarrow \text{infeasible}$

(3) $\mu_1 = \mu_3 = 0 \rightarrow x = 0, y = 2, \lambda = -2, \mu_2 = -2$

(4) $\mu_2 = \mu_3 = 0 \rightarrow x^2 + y^2 - 5 = 0 \rightarrow x = 2.48, y = 0.76$

(5) $\mu_1 = 0 \rightarrow x = y = 0 \rightarrow \text{infeasible}$

(6) $\mu_2 = 0 \rightarrow x^2 + y^2 - 5 = 0, y = 0 \rightarrow x = \sqrt{5} \rightarrow \text{infeasible}$

(7) $\mu_3 = 0 \rightarrow x^2 + y^2 - 5 = 0, x = 0 \rightarrow y = \sqrt{5} \rightarrow \text{infeasible}$

(8) $x^2 + y^2 - 5 = 0, x = 0, y = 0 \rightarrow \text{infeasible}$

(1), (3), (4)번이 실행 가능한(feasible) 해

그 중에서 가장 작은 값을 내는 최적해는

(1)번으로서 $x^* = \frac{4}{5}, y^* = \frac{8}{5}$ 이고

그때의 값은 $f(x^*, y^*)$ = 3.2이다.

Unit 04 | Lagraingian & KKT condition

☑ 라그랑지안과 KKT condition이 어디에 쓰였는가?

1. LASSO 회귀

$$\hat{\beta}_{lasso} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2$$

$$\text{subject to} \quad \|\beta\|_1 \leq t$$

2. RIDGE 회귀 회귀

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2$$

$$\text{subject to} \quad \|\beta\|_2^2 \leq t$$

Unit 04 | Lagraingian & KKT condition

☑ 라그랑지안과 KKT condition이 어디에 쓰였는가?

3. Suppor Vector Machine

$$\text{minimize} \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 \quad \text{for all } i$$

Q & A

들어주셔서 감사합니다.