Linear Regression: Part I

Problem, Estimation, Algorithm

송 준

고려대학교
통계학과 / 융합데이터과학 대학원

Recap: Function Estimation

Optimal predictor:

$$f^* = argmin_f \mathbb{E}[(f(X) - Y\ )^2]$$

Empirical Risk Minimizer:

$$\widehat{f_n} = argmin_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} ((f(X_i) - Y_i\ ))^2$$

Class of predictors   Empirical mean

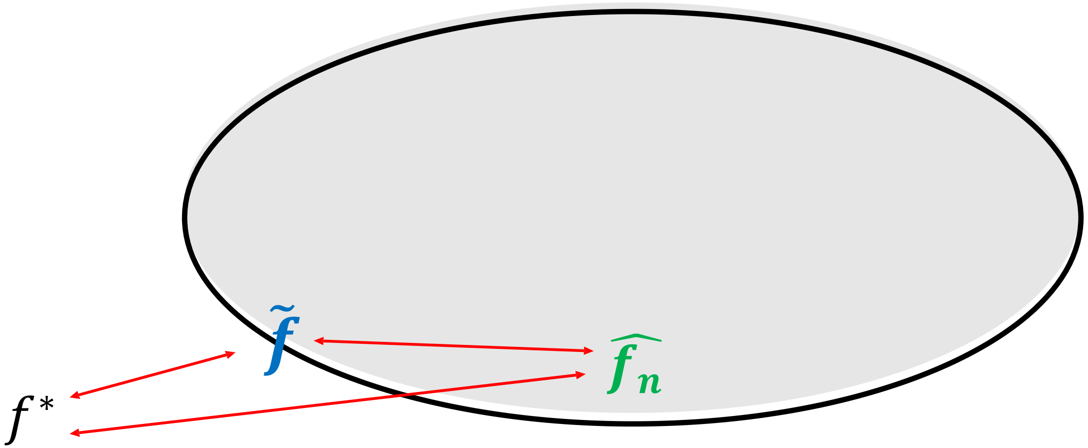
Recap: Function Estimation

Ideal goal: Construct prediction rule $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

$$\tilde{f} = \arg\min_{f \in F} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

$$\widehat{f_n} = \arg\min_{f \in F} \sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$



Recap: Function Estimation – one more

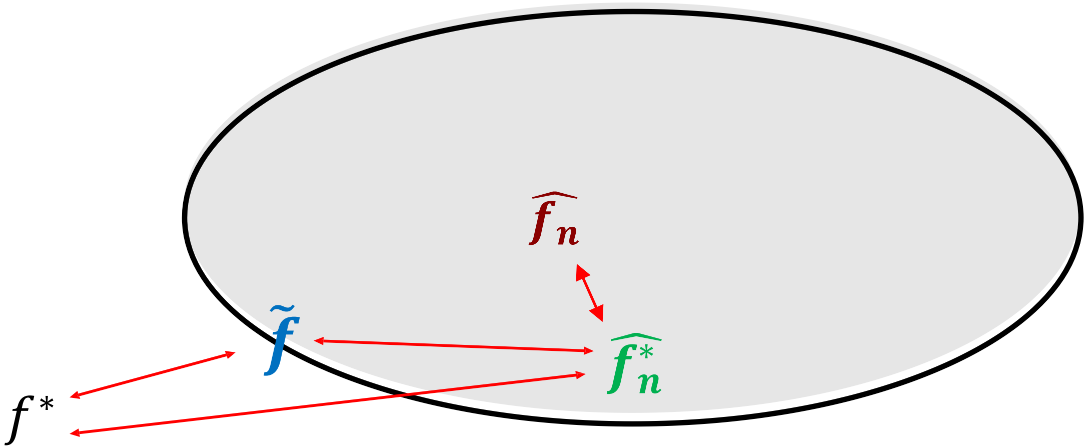
Ideal goal: Construct prediction rule $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

$$\tilde{\boldsymbol{f}} = \arg\min_{f \in F} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

$$\widehat{\boldsymbol{f}^*_{\boldsymbol{n}}} = \arg\min_{f \in F} \sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$

$$\widehat{\boldsymbol{f}_{\boldsymbol{n}}} = \widehat{\arg\min_{f \in F}} \sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$



함수공간 F가 복잡할 경우 minimizer를 찾지 못할 수 있음

Recap: Introduction to Linear Regression

- **Data Assumption:**
  - $(x_1, y_1), \dots, (x_n, y_n), x_i \in R^p,\ \ y_i \in R$
  - (x_i, y_i): a realization of $(X_i, Y_i) \sim i.i.d. (X, Y)$

- **Model Assumption:** X와 Y는 선형관계를 가짐.

$$Y = f(X) + \epsilon$$

- **f 의 형태제약:**

$$\mathrm{F} = \{f: f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p, for\ some\ \beta_0, \beta_1, \dots, \beta_p\}$$

1차 목표: $\beta_0, \beta = (\beta_1, \dots, \beta_p)^T$ 찾기,

Introduction

Introduction

- Part I: Regression Model의 설명과 추정(Estimation)
- Part II: 예측과 추론 (Prediction and Inference)

Introduction

- **회귀 모형 (Regression Model)**:

$$Y = f(X) + \epsilon$$

Introduction

- **회귀 모형 (Regression Model)**:

$$\boxed{Y} = f\boxed{(X)} + \boxed{\epsilon}$$

오차 (error)
*주의: 뒤에 나오는 잔차(residual)와 다른 개념
- 불확정성(uncertainty), noise, etc

종속변수 (Dependent Variable)	독립변수 (Independent Variable)
반응변수 (Response Variable)	설명변수 (Explanatory Variable)
반응변수 (Response Variable)	예측변수 (Predictor Variable)
Output	Input

독립변수, 반응변수는 각각 확률변수로 여러 개의 확률변수가 있을 수 있음

Goal of Regression Models

- **회귀 모형 (Regression Model)**:

$$Y = f(X) + \epsilon$$

- **Goal of Regression Models:**
  - **추정 (Estimation):** 관계를 나타내는 함수 f에 대한 추정
  - **예측 (Prediction)**: X 값이 주어졌을 때 대응되는 Y 값의 예측
  - **추론 (Inference)**: Further investigation
    - 예측이 “얼마나” 정확한가?
    - 함수 f() 가 얼마나 정확한가?
    - 예측변수가 여러 개 있을 때 모든 변수가 Y의 값에 영향을 주나?
    - 모형이 충분히 적합 됐나?

Goal of Regression Models

- **회귀 모형 (Regression Model)**:

$$Y = f(X) + \epsilon$$

- **Goal of Regression Models:**
  - **추정 (Estimation):** 관계를 나타내는 함수 f에 대한 추정
  - **예측 (Prediction)**: X 값이 x로 주어졌을 때 Y 값의 예측
  - **추론 (Inference)**: Further investigation of the data
    - 예측이 “얼마나” 정확한가?
    - 함수 f() 가 얼마나 정확한가?
    - 예측변수가 여러 개 있을 때 모든 변수가 Y의 값에 영향을 주나?
    - 모형이 충분히 적합 됐나?
  - **예측**만 목표로 할 시: 다양한 방법론 적용 가능
  - **추론**을 목표로 할 시: 관계를 나타내는 f()에 제약이 필요함
  - 단순한 모형부터 시작! **f 는 선형함수.**

Linear Regression Models

Linear Regression Models

- **회귀 모형 (Regression Model):**

$$Y = f(X) + \epsilon$$

  - $X = (X_1, \cdots, X_p)$: p차원 확률변수
  - Y: 1차원 확률변수

- **선형 회귀 모형 (Linear Regression Model):**

$$f\colon \mathbb{R}^p \to \mathbb{R}$$

  - f : X와 Y가 선형관계를 가진다

$$\Updownarrow$$

  - $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$ for some $\beta_j, j = 0, \dots, p$

Linear Regression Models

- **선형 회귀 모형 (Linear Regression Model):**

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

  - $X = (X_1, \cdots, X_p)$: p차원 확률변수
  - Y: 1차원 확률변수
  - 오차 항: $\mathrm{E}(\epsilon) = 0, var(\epsilon) = \sigma^2, \epsilon \sim N(0, \sigma^2)$,
  - $\beta = (\beta_0, \beta_1, \ldots, \beta_p) \in \mathbb{R}^{p+1}$: 회귀 모수(**parameter**) 혹은 회귀계수 (**coefficients**), unknown, non-random parameters (to be estimated)

- **표본 Sample:** n개의 data $(X_1, Y_1), \ldots, (X_n, Y_n)$ 는 위 모델을 따르는 random copy

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \epsilon_i, \qquad i = 1, \ldots, n$$

Estimation

Estimation

- **Sample (random):** n개의 data $(X_1, Y_1), \dots, (X_n, Y_n)$ 는 위 모델을 따르는 random copy

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \epsilon_i, \qquad i = 1, \dots, n$$

- **Data (realized values, actual observation):**
  - $(x_1, y_1), \dots, (x_n, y_n), x_i \in R^p,\ y_i \in R$
  - (x_i, y_i): a realization of $(X_i, Y_i) \sim i.i.d.\,(X, Y)$

- **Goal:** Using the data (observations)
  - Estimate $f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  - Estimate $\beta = \left(\beta_0, \beta_1, \dots, \beta_p\right)^T$

Estimation: Simplification

- **Goal:** Using the data (observations)
  - Estimate $f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  - Estimate $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$
- Sometimes, ignore β_0: X 값에 영향을 받지 않는 Y만의 평균값.
  - 편의상 $\beta_0 = 0$ 이라 가정하기도 함 (Y_i 대신 $Y_i - \beta_0$가 output이라고 생각), 혹은
  - input x 에 1이 고정적으로 있다고 가정. $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1p} & \cdots & x_{np} \end{pmatrix}$$

$$y_i \approx \beta_0 \cdot 1 + \beta_1 \cdot x_{i1} + \cdots + \beta_p \cdot x_{ip}$$

Linear Functions

Linear Functions

- Consider the space of linear functions $f_\beta(x)$ defined by

$$f_\beta(x) = \beta^T x = \begin{bmatrix} \beta_1 & \cdots & \beta_p \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix} = \beta_1 x_1 + \cdots + \beta_p x_p$$

- $x \in \mathbb{R}^p$ is called an **input** (a.k.a. **features** or **covariates**)
- $\beta \in \mathbb{R}^p$ is called the **parameters** (a.k.a. **parameter vector**)
- $y = f_\beta(x)$ is called the **label** (a.k.a. **output** or **response**)

Linear Regression Problem

Linear Regression Problem

x가 주어졌을 때 y 값 근사 (estimate)

- **Input** : Data $Z = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$
- **Output** : A linear function $f_\beta(x) = \beta^T x$ such that $y_i \approx \beta^T x_i$

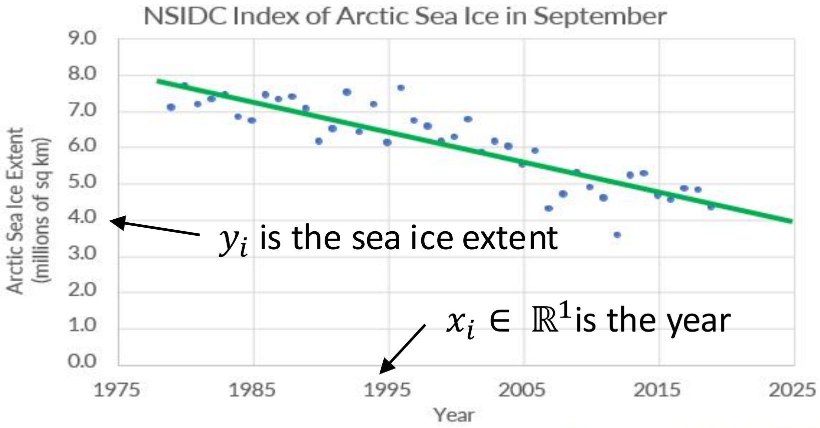


Image: https://www.flickr.com/photos/gsfc/5937599688/
Data from https://nsidc.org/arcticseaicenews/sea-ice-tools/

Choice of Loss Function

f_β 가 주어졌을 때 i번째 관측치에 대한 loss (squared error loss)

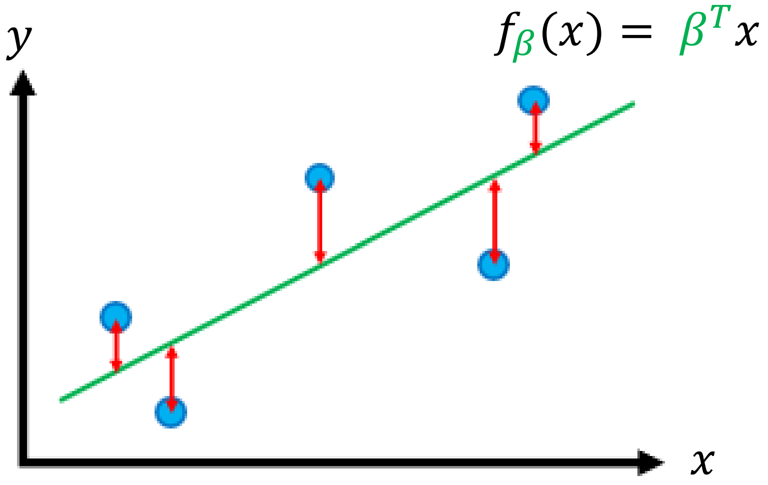
Choice of Loss Function

- $y_i \approx \beta^T x_i$ if $\boxed{(y_i - \beta^T x_i)^2}$ small

- **Mean squared error(MSE) :**

$$\hat{R}(\beta; Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2$$

- Computationally convenient and works well in practice



$$\hat{R}(\beta; Z) = \frac{\updownarrow^2 + \updownarrow^2 + \updownarrow^2 + \updownarrow^2 + \updownarrow^2}{n}$$

Choice of Loss Function

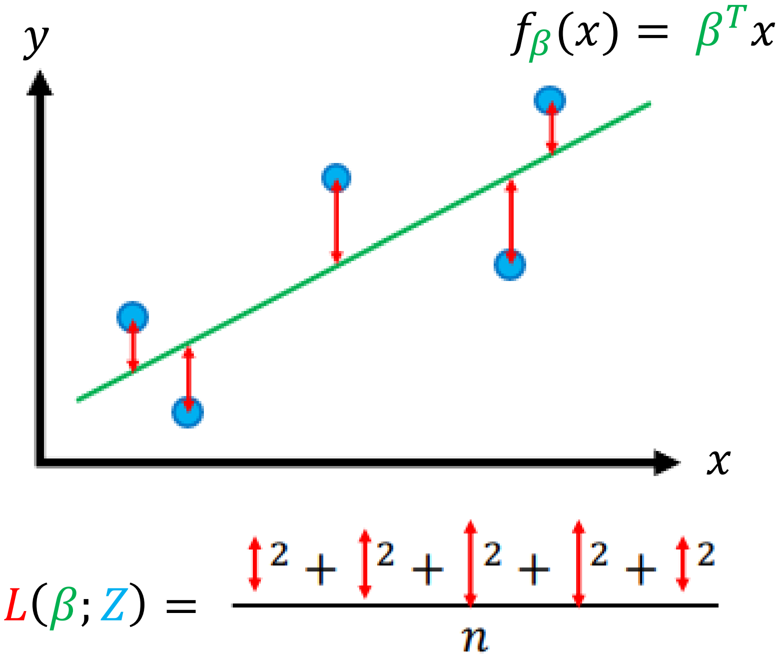
f_β 가 주어졌을 때 i번째 관측치에 대한 loss (squared error loss)

Choice of Loss Function

- $y_i \approx \beta^T x_i$ if $\boxed{(y_i - \beta^T x_i)^2}$ small
- **Mean squared error(MSE) :**

$$L(\beta; Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2$$

- 편의상 위 값을 loss 라고 표현하기도 한다.



Linear Regression Algorithm

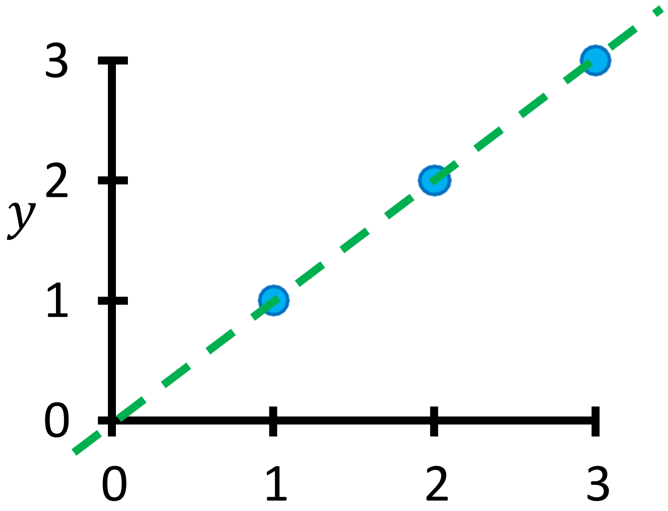
- **Input** : Dataset $Z = \{(x_1, y_1), \cdots, (x_n, y_n)\}$
- Compute

$$\hat{\beta}(Z) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} L(\beta; Z)$$
$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2$$

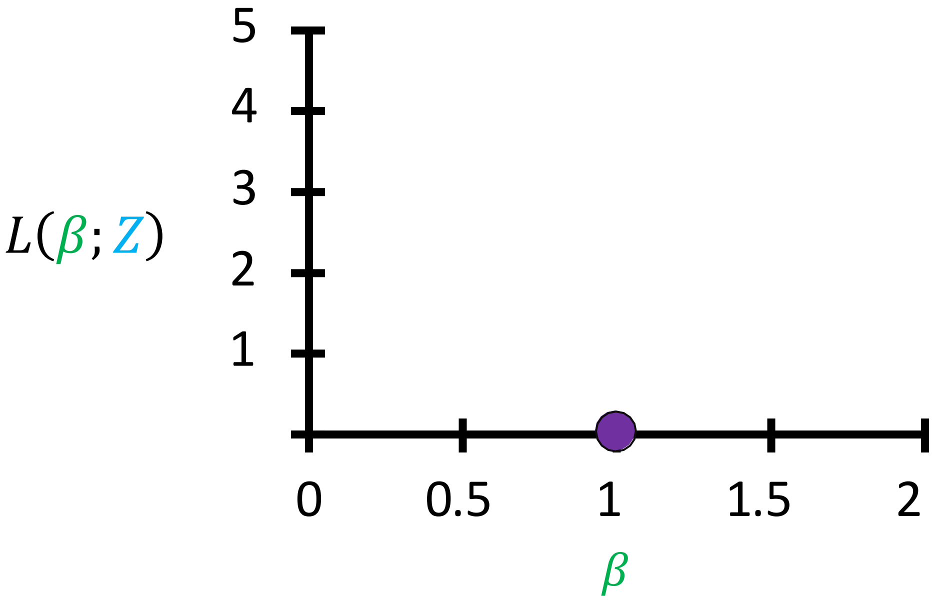
- **Output** : $f_{\hat{\beta}(Z)}(x) = \hat{\beta}(Z)^T x$

Intuition on Minimizing MSE Loss

- Consider $x \in \mathbb{R}$ and $\beta \in \mathbb{R}$

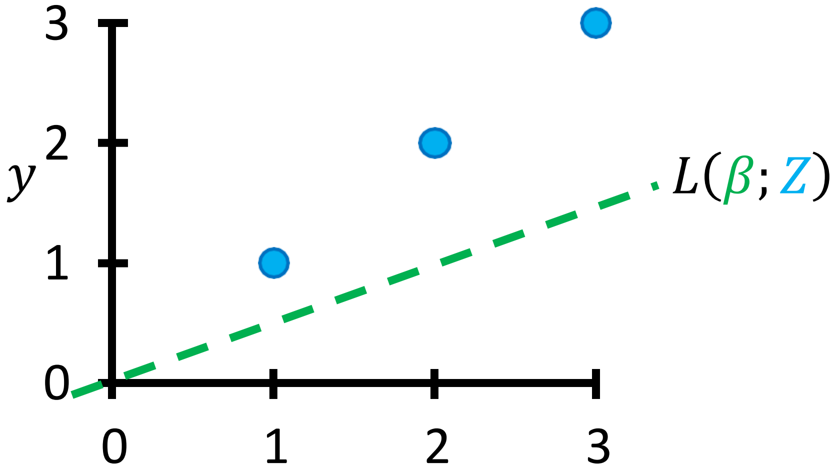


$\beta = 1$

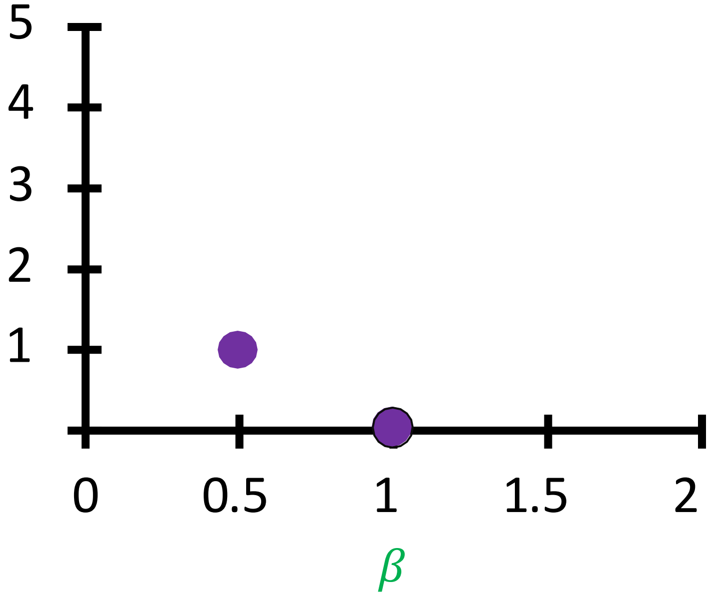


Intuition on Minimizing MSE Loss

- Consider $x \in \mathbb{R}$ and $\beta \in \mathbb{R}$

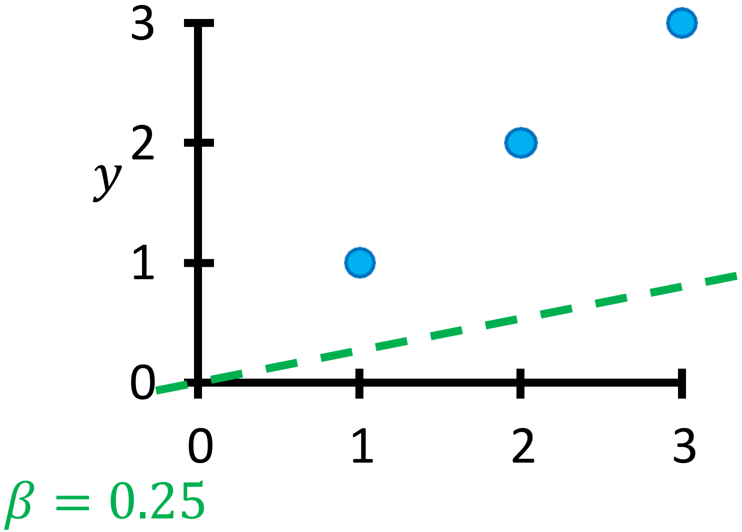


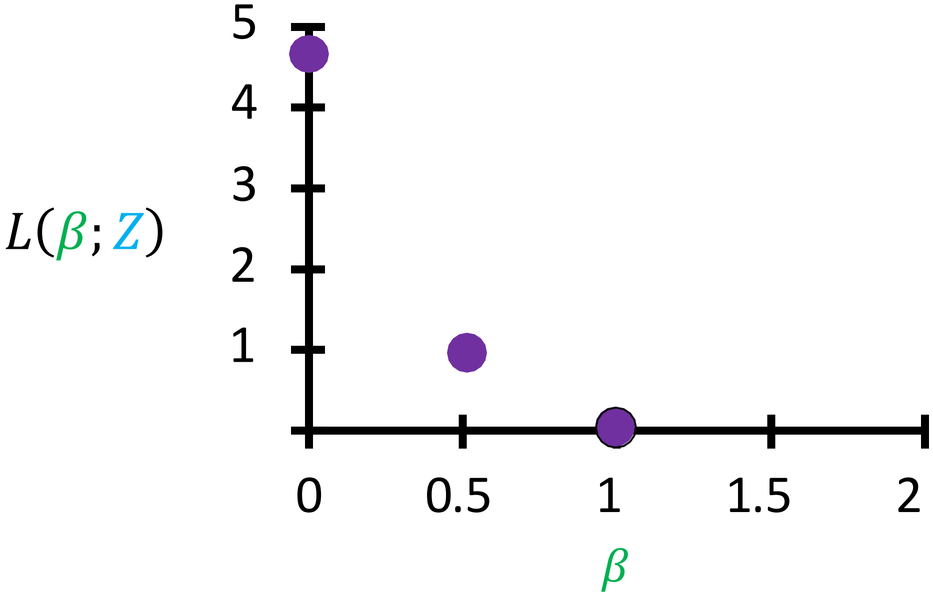
$\beta = 0.5$



Intuition on Minimizing MSE Loss

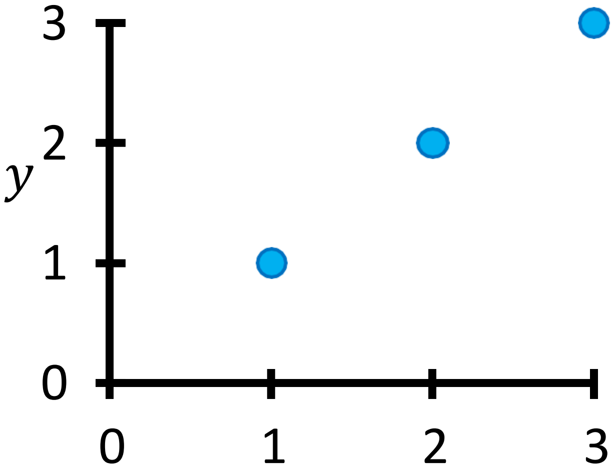
- Consider $x \in \mathbb{R}$ and $\beta \in \mathbb{R}$

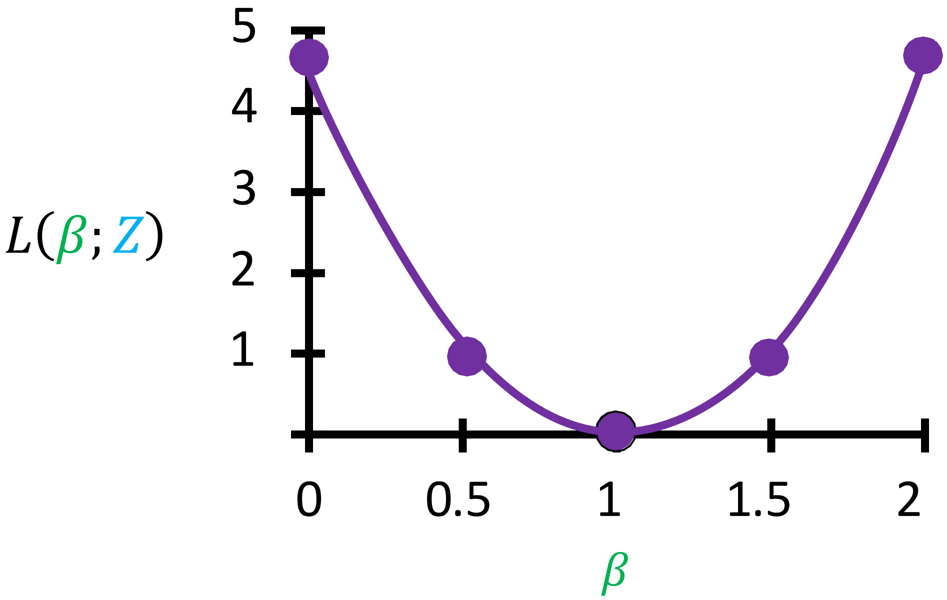




Intuition on Minimizing MSE Loss

- Consider $x \in \mathbb{R}$ and $\beta \in \mathbb{R}$

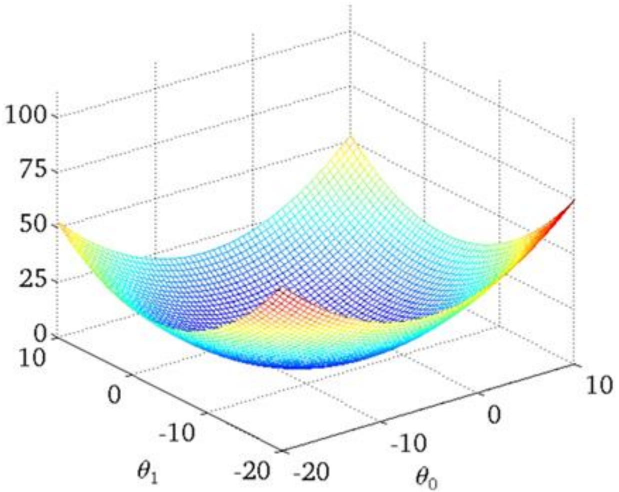


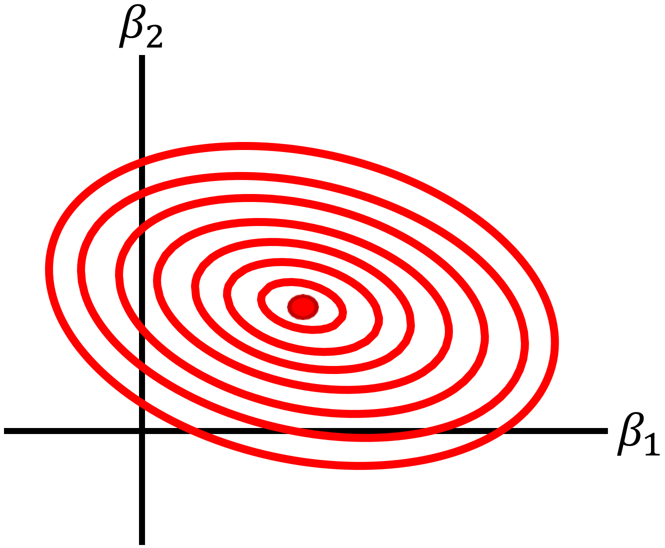


Intuition on Minimizing MSE Loss

- **Convex** ("bowl shaped") in general

$L(\beta; Z)$





- ✓ L 은 β 에 관한 함수
- ✓ Convex 함수는 Unique Minimizer 존재
- ✓ Analytics Solution 이 없더라도 Numerical Solution 을 잘 찾음

Linear Regression: Estimation Summary

General strategy

- Model family $\mathrm{F} = \{f_\beta\}_\beta$
- Loss function $L(\beta; Z)$

Linear regression strategy

- Linear functions $F = \{f_\beta(x) = \beta^T x\}$
- MSE $L(\beta; Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2$

Linear regression algorithm

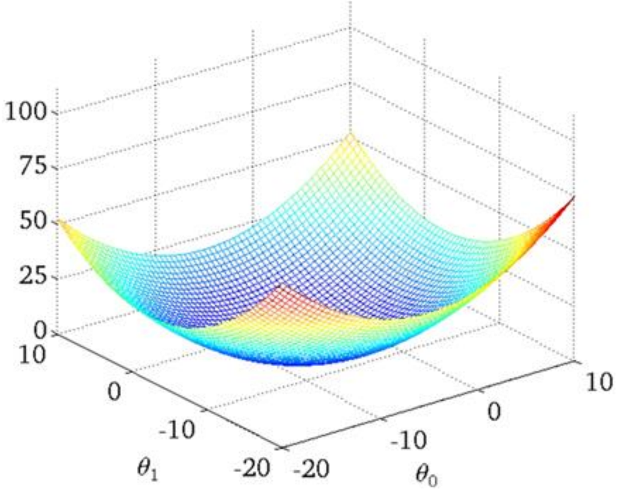
$$\hat{\beta}(Z) = \underset{\beta}{\operatorname{argmin}}\, L(\beta; Z)$$

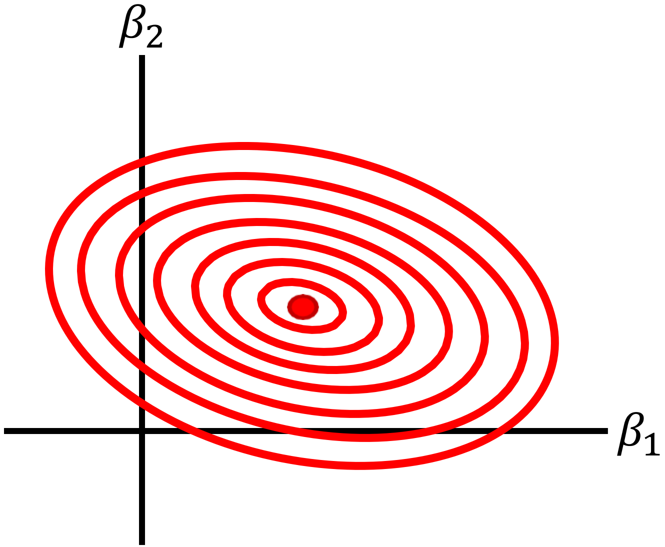
Computing: optimization

Intuition on Minimizing MSE Loss

- **Convex** (“bowl shaped”) in general

$L(\beta; Z)$





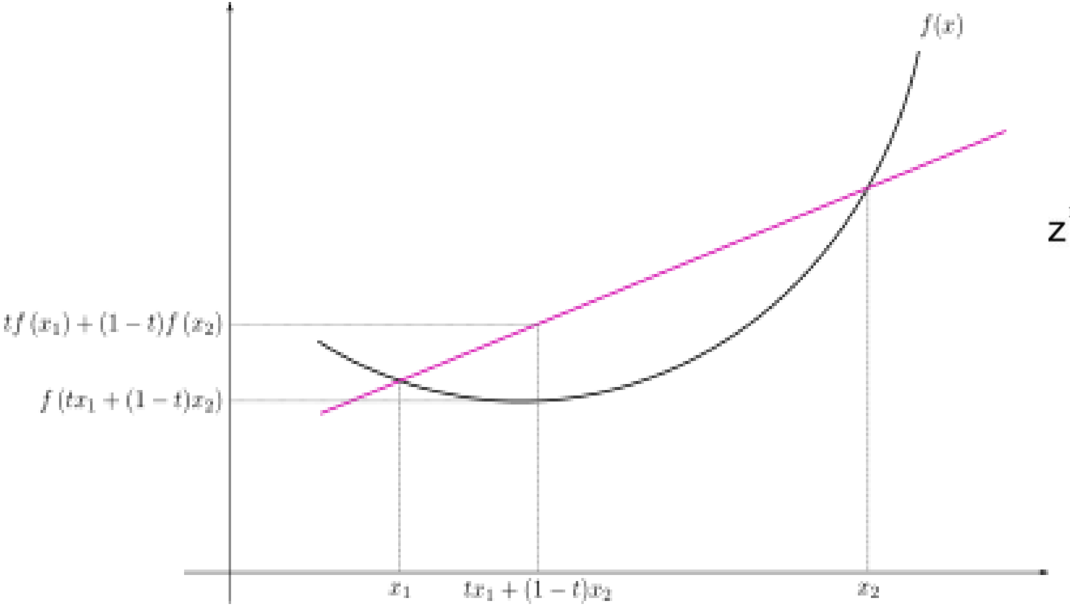
- L 은 β 에 관한 함수
- Convex 함수는 Unique Minimizer 존재
- Analytics Solution 이 없더라도 Numerical Solution 을 잘 찾는 편

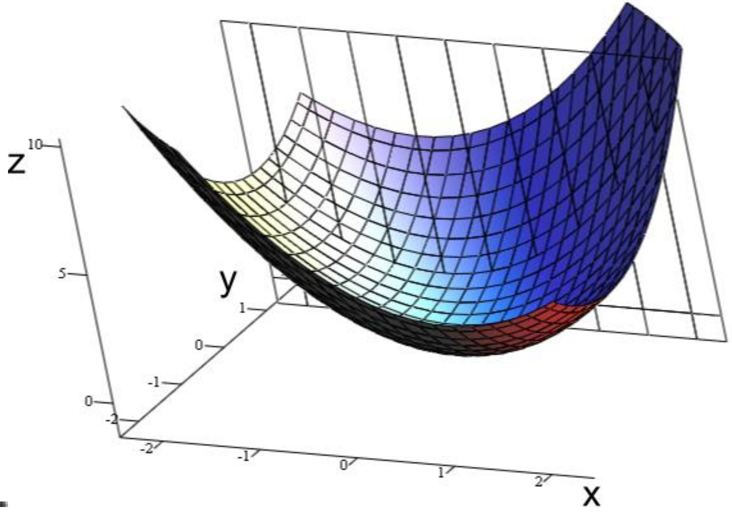
Is the Objective Function Convex?

- Convex function:

For all $0 \leq t \leq 1$ and all $x_1, x_2 \in X$:

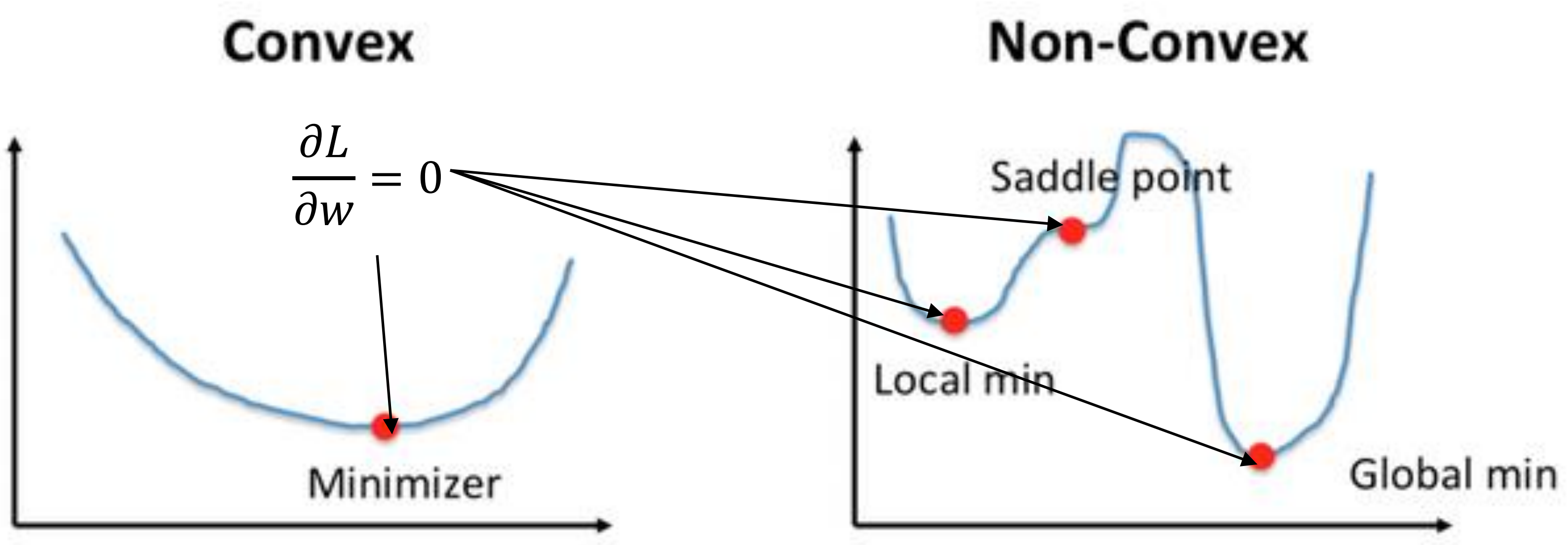
$$f(t\,x_1 + (1-t)x_2) \ \leq tf(x_1)$$





Is the Objective Function Convex?

- If the function is convex & differentiable, we can easily find the minimizer
- not convex, it's difficult.



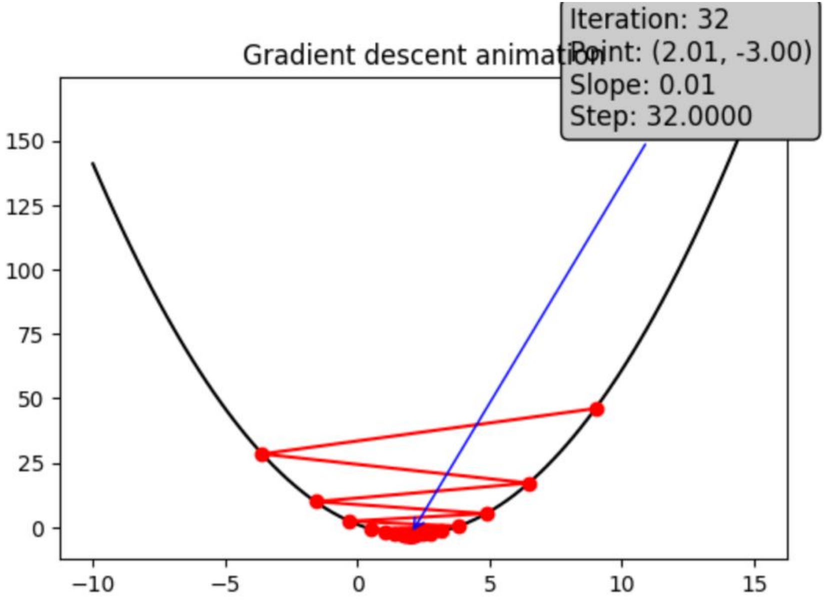
Solution to the Optimization Problem

- **Analytic Solution (Explicit form, closed form – solution): 미분=0**

$$L(\beta; x) = \beta^2 - 2x\beta + 10 = (\beta - x)^2 + 9$$

$$\arg\min_{\beta} L(\beta) = x$$

- **Numerical Solution (Optimization Algorithm)**



Data

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1p} & \cdots & x_{np} \end{pmatrix}, \mathbf{X}\beta = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1p} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

$$\mathbf{X}\beta = \begin{pmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_p x_{1p} \\ \vdots \\ \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_p x_{1p} \end{pmatrix}$$

Analytic Solution

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \mathbf{X}\beta = \begin{pmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_p x_{1p} \\ \vdots \\ \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_p x_{1p} \end{pmatrix}$$

$$\begin{aligned} \hat{\beta} &= \text{argmin}_{\beta \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left(y_i - \beta_0 \cdot 1 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip}\right)^2 \\ &= \text{argmin}_{\beta \in \mathbb{R}^{p+1}} (Y - \mathbf{X}\beta)^{\mathrm{T}} (Y - \mathbf{X}\beta) \end{aligned}$$

- **The Analytic Solution**
- Estimating Equation: $(\boldsymbol{X}^T \boldsymbol{X})\hat{\beta} = \boldsymbol{X}^T \boldsymbol{Y}$

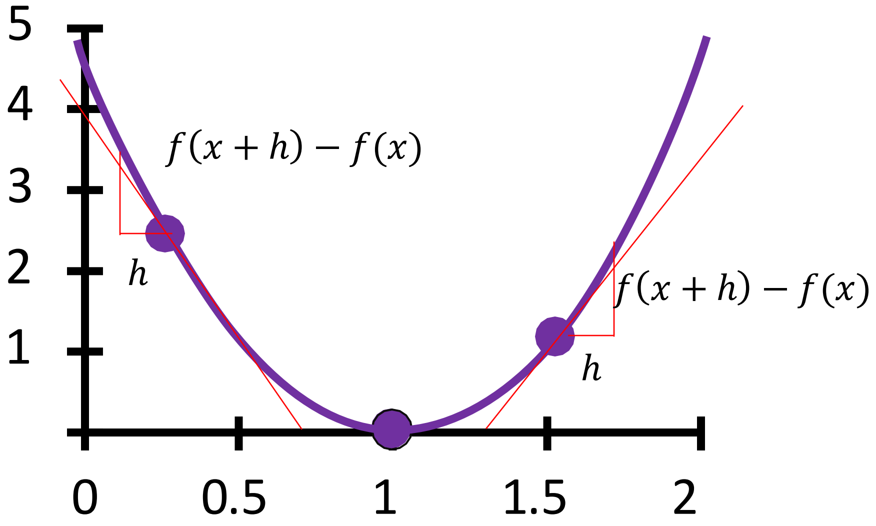
$$\hat{\beta} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}$$

$$\hat{\mathbf{Y}} = \boldsymbol{X}\hat{\beta} = \boldsymbol{X}(\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}$$

Numerical Solution: Gradient Descent

Idea #2: **Follow the slope**

In 1-dimension, the **derivative** of a function gives the slope:



$$\frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- $f'(x_0)$: 함수 **f가 x_0 에서 증가하는 방향**
- $|f'(x_0)|$:
  - ✓ 기울기 크기
  - ✓ 얼마나 빠르게?

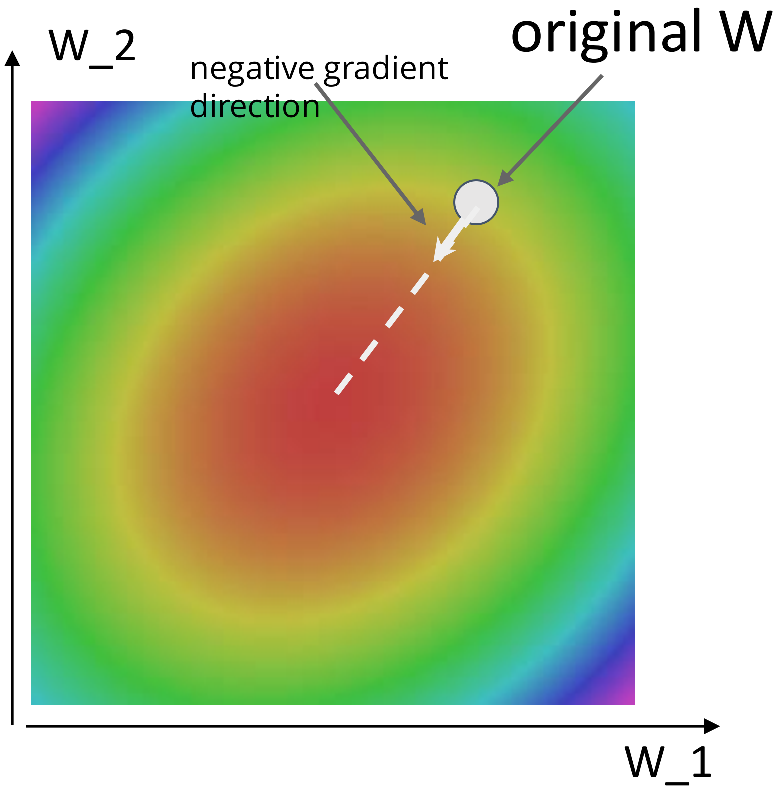
Gradient Descent

Iteratively step in the direction of the negative gradient (direction of local steepest descent)

```
# Vanilla gradient descent
w = initialize_weights()
for t in range(num_steps):
  dw = compute_gradient(loss_fn, data, w)
  w -= learning_rate * dw
```

Hyperparameters:

- Weight initialization method
- Number of steps
- Learning rate



Gradient Descent

See the Jupyter Notebook