Machine Learning

Linear Regression 1

Dept. SW and Communication Engineering

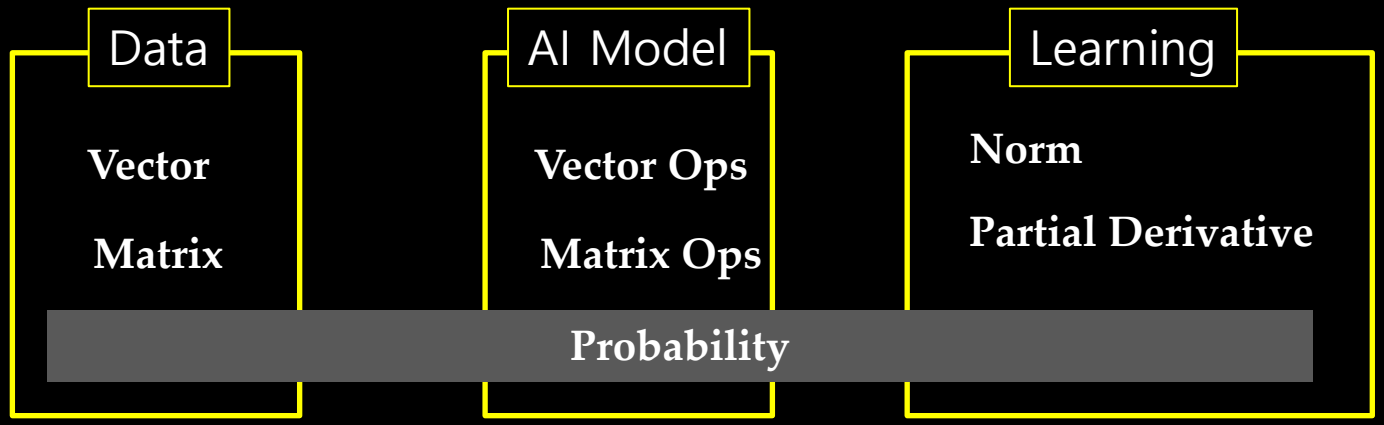
Prof. Giseop Noh (kafa46@hongik.ac.kr)

Contents

Study Goals

- The concept of regression & dataset
- Simple linear regression
- Gradient Desecent
- Multiple linear regression

Mathematics in Machine Learning



Concept of Regression & Dataset

Concept of Regression

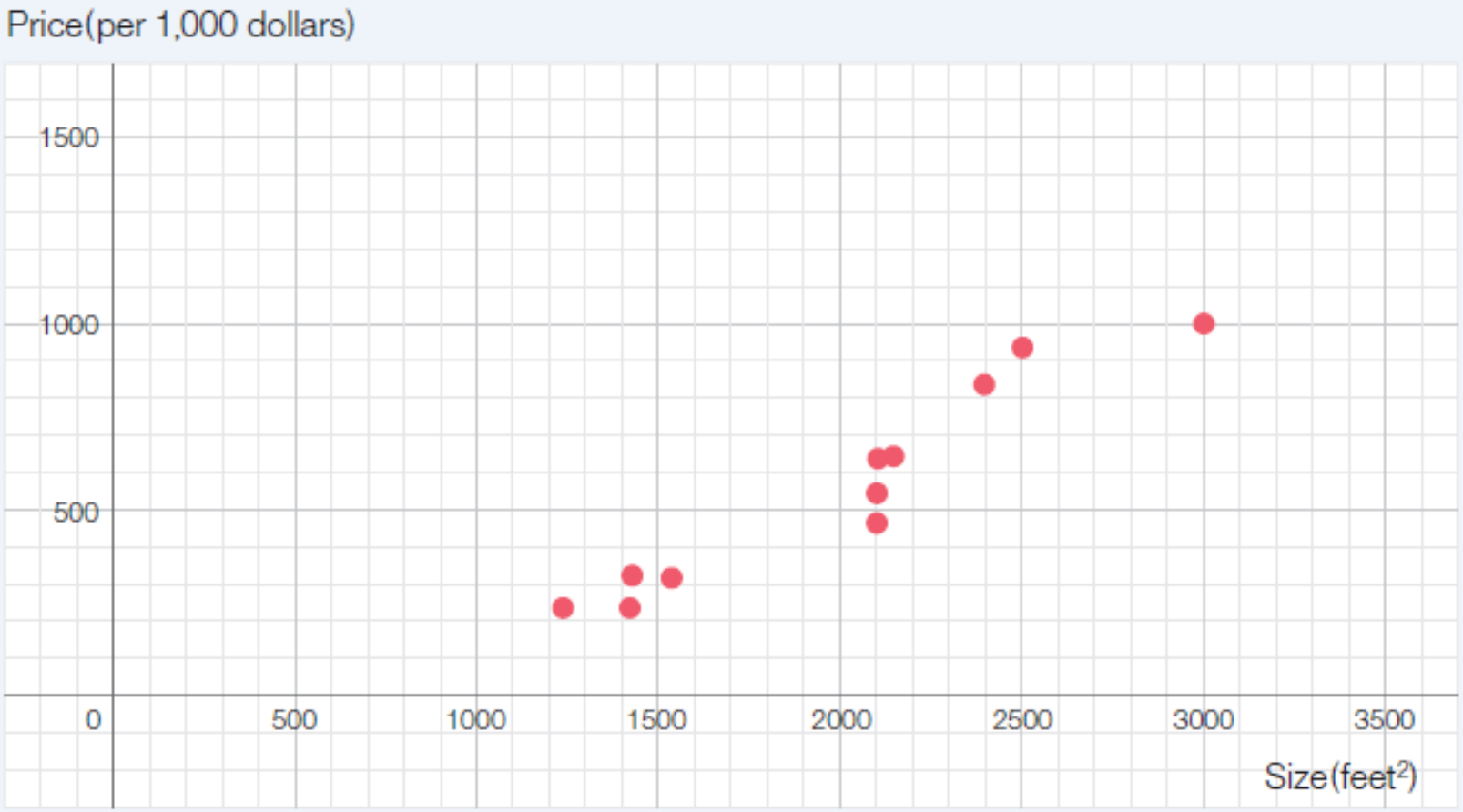
Regression

- Represents the relationship between input and output variables.
- Used to predict the output value for a new input value or to understand the effect of input variables on the output variable.
- Output (predicted value): continuous value

Example

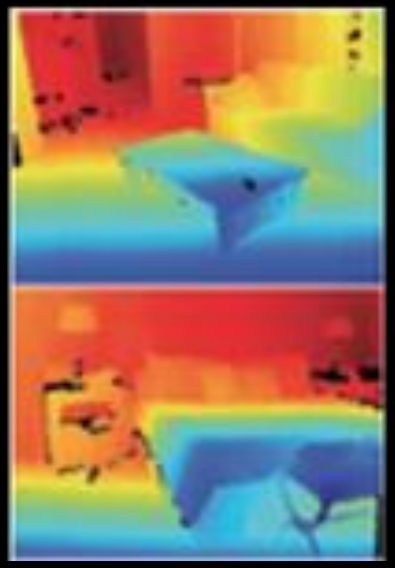
- Predicting house prices based on house size → Relation, Prediction
  - Relation: What is the relationship between house price and house size?
  - Prediction: If the house size is 1,202, what is the expected house price?

Example


Price(per 1,000 dollars)
1500
1000
500
0
500
1000
1500
2000
2500
3000
3500
Size(feet²)

Regression vs. Classification

- **Regression**
  - Expresses the degree of proximity (distance) using colors.
  - Example: Blue represents nearby locations, while red represents distant locations, etc.
- **Classification**
  - Represents different object types with different colors.
  - Example: Each pixel is distinguished, where 0 means a desk, 1 means a chair, etc.



Dataset

Dataset in Regression (Supervised Learning Dataset)

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

- $\boldsymbol{x_i}$: i-th input data of the dataset.
  - It is represented as a $\boldsymbol{d}$-dimensional vector $\boldsymbol{x_i} = [\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, \cdots, \boldsymbol{x_{id}}]^{\boldsymbol{T}}$, where $\boldsymbol{x_{ij}}$ is the $\boldsymbol{j}$-th feature of x_i.
- $\boldsymbol{y_i}$: The real-valued output corresponding to the i-th input in the dataset
  - while y_i can be a multivariate vector, in this context it is limited to a scalar value.
- N: The size of dataset

Purposes of Regression

Two Purposes of Regression

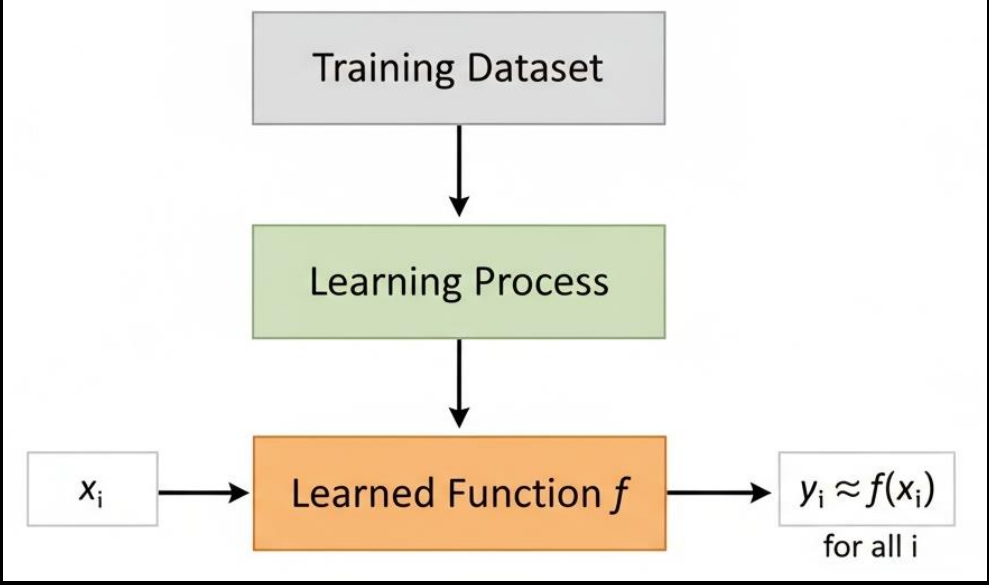
- Prediction Aspect
  - Predicting the output value for a new input value.
  - Example: What is the weight of a person who is 170 cm tall?

- Interpretation or Relationship Aspect
  - Understanding how the output value changes concerning the input value.
  - Example: Do taller people tend to weigh more?

Learning

■ Learning in Regression

- The process of finding a function that produces an appropriate output value when given an input value.
- The process of finding a function that represents the relationship between independent and dependent variables.



- **Correlation (correlation) ≠ Causation**
- **Regression identifies correlation and does not imply causation.**

  Therefore, when interpreting results, they should not be understood as causal relationships.

Commonly Confused Terms in Machine Learning

Category	Terms
Input Variable	Feature, Independent Variable
Output Variable	Dependent Variable
Input Data	Observation, Feature Vector, Sample, Input Value, Input Vector
Actual Value	Ground Truth, Label, Class
Predicted Value	Output Value, Result Value
Regression	Regression Model, Regression Function
Loss Function	Cost Function, Objective Function
Model Parameters	Model Weights, Model Hyperparameters

Simple Linear Regression

Principle of Simple Linear Regression

Simple Linear Regression

- Basic model consisting of one independent variable and one dependent variable.

  $w = [w_0, w_1]^T$

- w is a parameter vector (also called coefficients or weights) and consists of two parameters.

Notation

$$f_w(X_i) = f_w(x_i) = w_0 + w_1 x_i$$

$$X_i = \left[x_{i1,}\ x_{i2,}\ x_{i3,} \cdots,\ x_{ij,} \cdots,\ x_{id}\right]^T \in \mathbb{R}^d$$

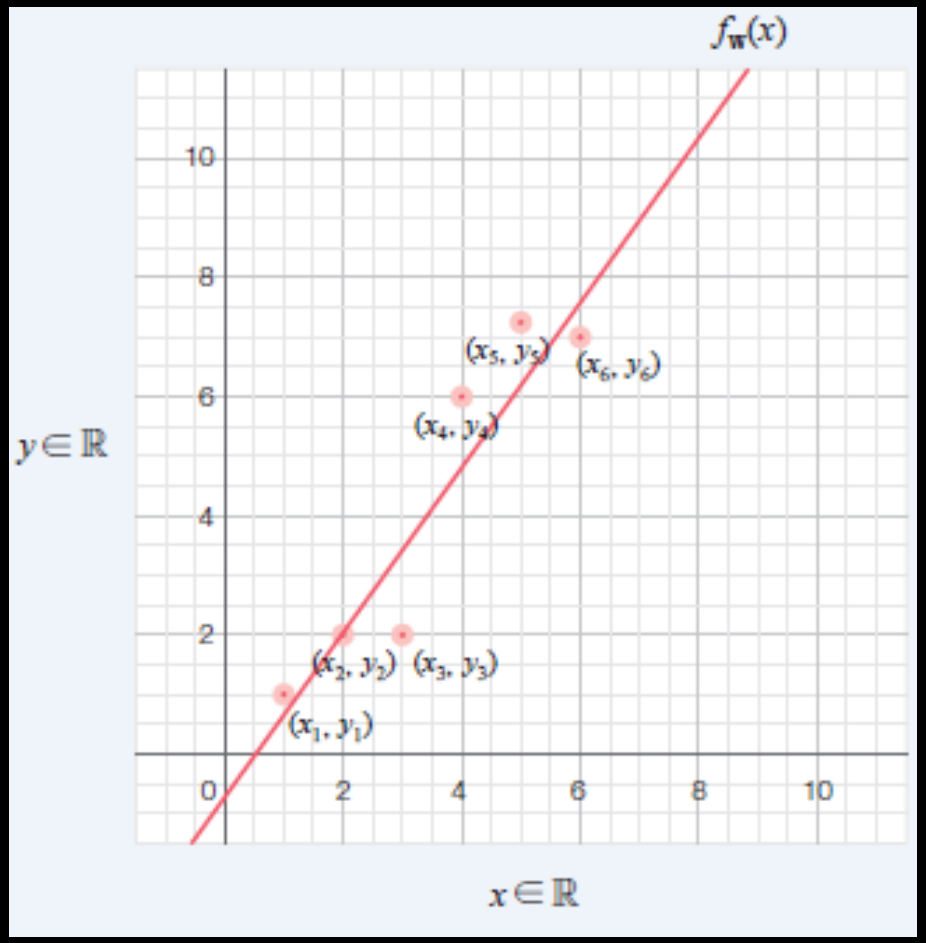
- x_i: The i-th input value of the training/testing data.
- d: The number of dimensions of the input value.
  - In the case of simple linear regression, d=1,

    so instead of representing the vector as $x_i = [x_{i1}]^T$, it is expressed as a scalar x_i.
- w: The parameters of the regression model (also called coefficients or weights).

Simple Linear Regression

- **Simple Linear Regression Model**

$$f_{\mathbf{w}}(x_i) = w_0 + w_1 x_i$$



Loss Function

■ Loss Function

- Calculated by comparing the actual values with the predicted values.
- When training data is given,
  - the predicted values of the linear regression model vary depending on *w*.

$$L(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f_{\mathbf{w}}(x_i))^2$$

- N: Number of input values
- $f_{\mathbf{w}}(x_i)$: The ith predicted value ($= \hat{y}_i$)
- y_i: The ith actual value

Loss Function: Key Idea

- **Using the training dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$**
  - determine $w = [w_0, w_1]^T$ so that $f_w(x_i)$ is as close as possible to y_i.

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w}) = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - f_{\mathbf{w}}(x_i))^2$$

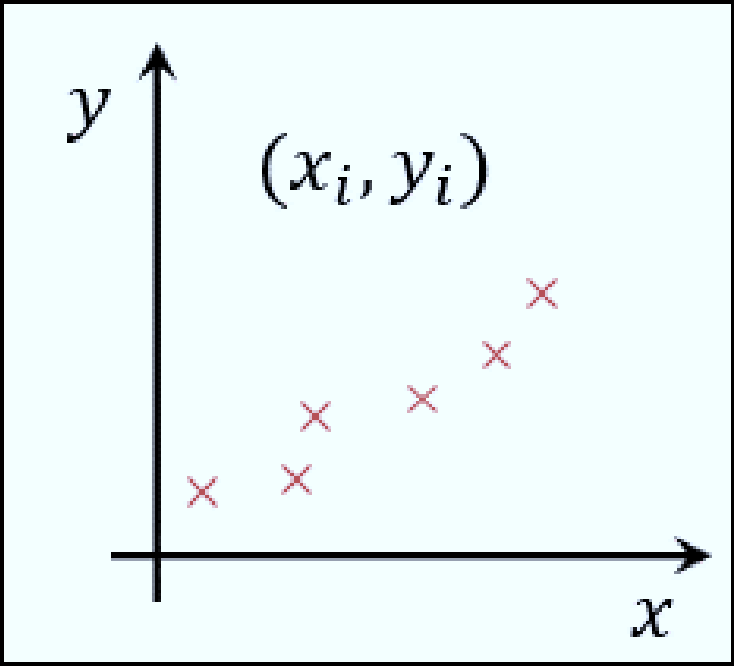
$\mathbf{w}^*$: The value of $\mathbf{w}$ that minimizes the loss function $L(\mathbf{w})$, representing the optimal parameters of the model. It is given by:

$$\mathbf{w}^* = [w_0^* \quad w_1^*]^T$$

Training Objective

- **The loss function (also known as the cost function or training objective function) is used to train the model parameters $w = [w_0, w_1]^T$.**

- **The smaller the loss function $\boldsymbol{L(w)}$, the closer the model's predicted output $f_w(x_i)$ is to the actual value y_i for the training data.**



Example of Loss Function

Assuming $w_0 = 0$ in Simple Linear Regression

Category	Description
Data	$D = \{(x_i, y_i)\}_{i=1}^{N}$
Model	$f_{\mathbf{w}}(x) = w_0 + w_1 x$ $f_{\mathbf{w}}(x) = w_1 x$ (assuming $w_0 = 0$)
Parameters	$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}^T$ w_1
Loss Function	$L(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(x_i)\right)^2$
Objective	$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w})$ $w_1^* = \arg\min_{w_1} L(w_1)$

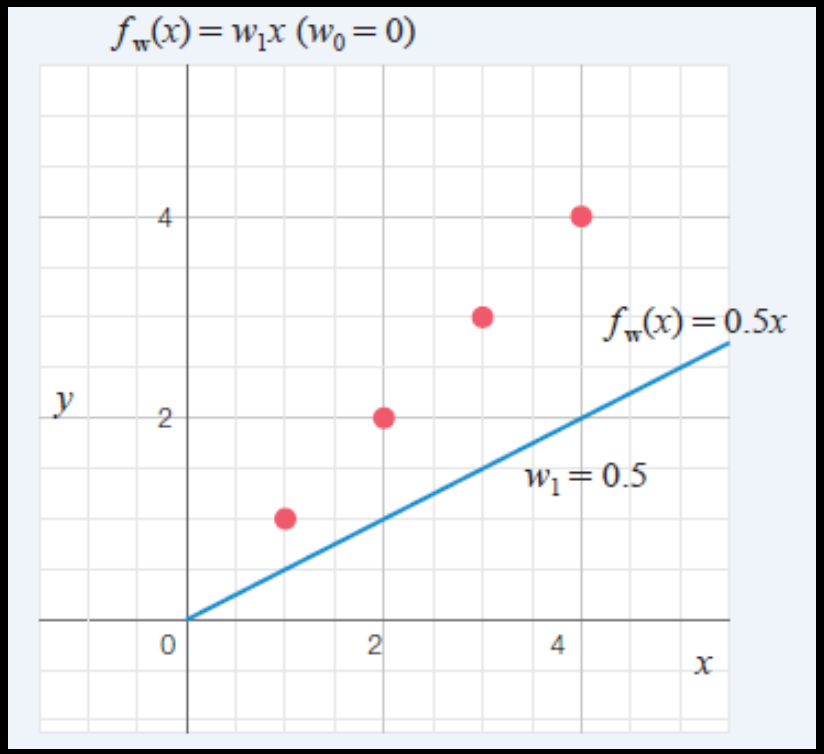
Handy Computing on Loss Function

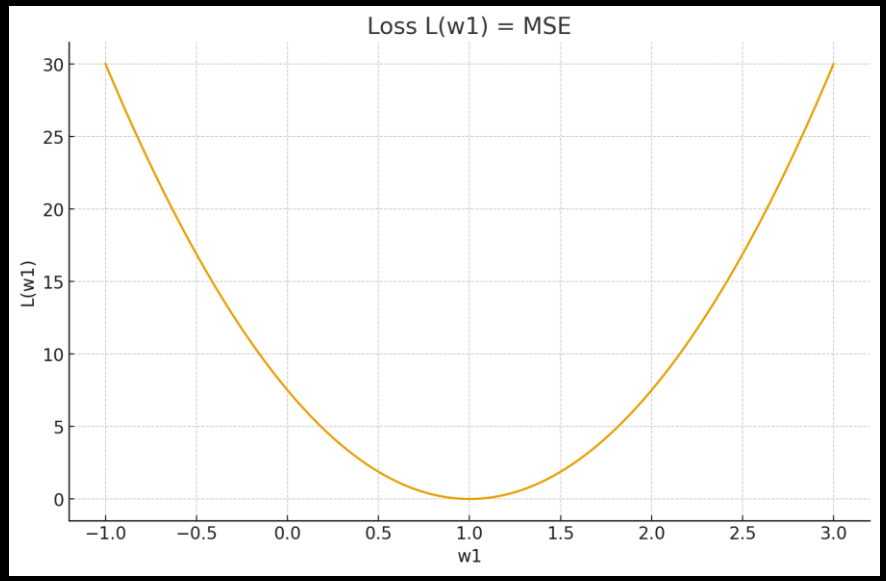
- $\boldsymbol{D} = \{(\mathbf{1},\mathbf{1}),(\mathbf{2},\mathbf{2}),(\mathbf{3},\mathbf{3}),(\mathbf{4},\mathbf{4}),\}$

$$f_{\mathbf{w}}(x) = w_1 x \quad (w_0 = 0)$$

When

$$w_1 = 0.5$$





$$L(w_1) = ?$$

$When\ w_1 = 0.5$

x	Actual y	Predicted $f(x) = 0.5x$	Error $y - f(x)$	Squared Error $\left(y - f(x)\right)^2$
1	1	0.5	0.5	0.25
2	2	1.0	1.0	1.00
3	3	1.5	1.5	2.25
4	4	2.0	2.0	4.00

Sum of squared errors

= 0.25 + 1.00 + 2.25 + 4.00 = 7.5

$$L(w_1 = 0.5) = \frac{7.5}{4} = 1.875$$

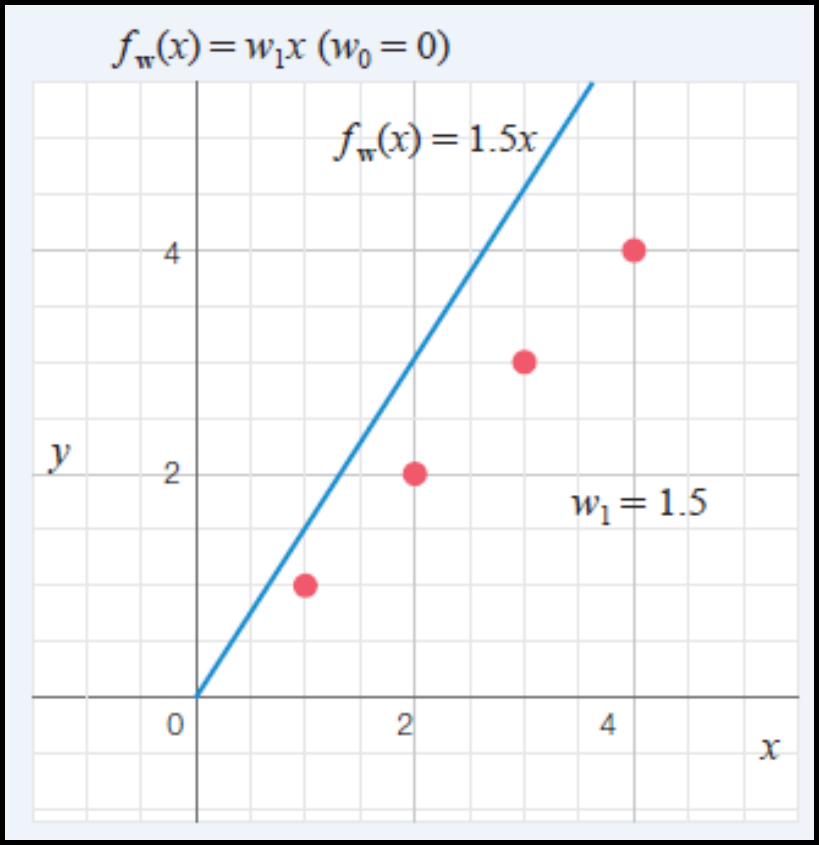
Handy Computing on Loss Function

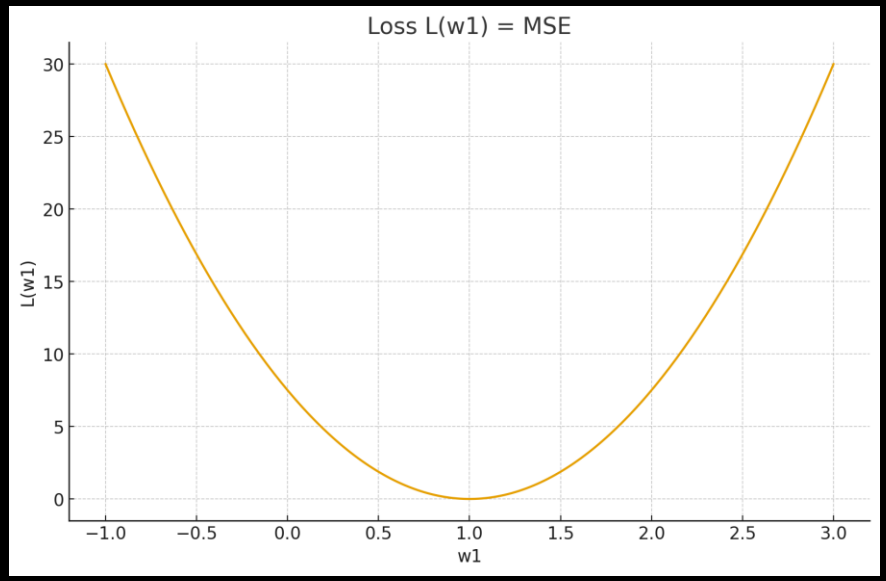
- $\boldsymbol{D} = \{(\mathbf{1},\mathbf{1}),(\mathbf{2},\mathbf{2}),(\mathbf{3},\mathbf{3}),(\mathbf{4},\mathbf{4}),\}$

$$f_{\mathbf{w}}(x) = w_1 x \quad (w_0 = 0)$$

When

$$w_1 = 1.5$$





$$L(w_1) = ?$$

$When\ w_1 = 1.5$

x	Actual y	Predicted $f(x) = 1.5x$	Error $y - f(x)$	Squared Error $(y - f(x))^2$
1	1	1.5	-0.5	0.25
2	2	3.0	-1.0	1.00
3	3	4.5	-1.5	2.25
4	4	6.0	-2.0	4.00

Sum of squared errors

= 0.25 + 1.00 + 2.25 + 4.00 = 7.5

$$L(w_1 = 1.5) = \frac{7.5}{4} = 1.875$$

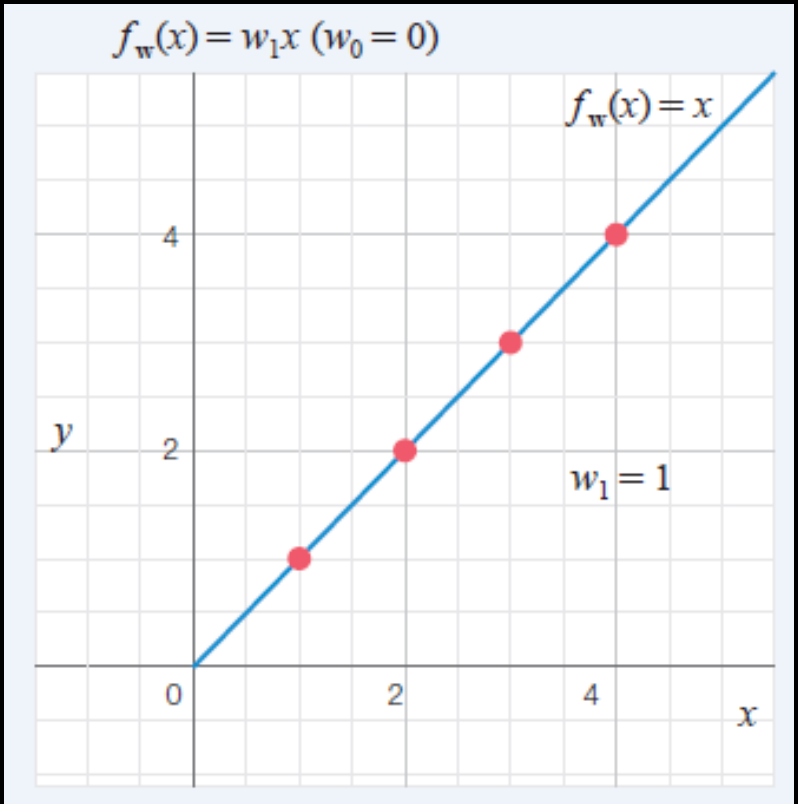
Handy Computing on Loss Function

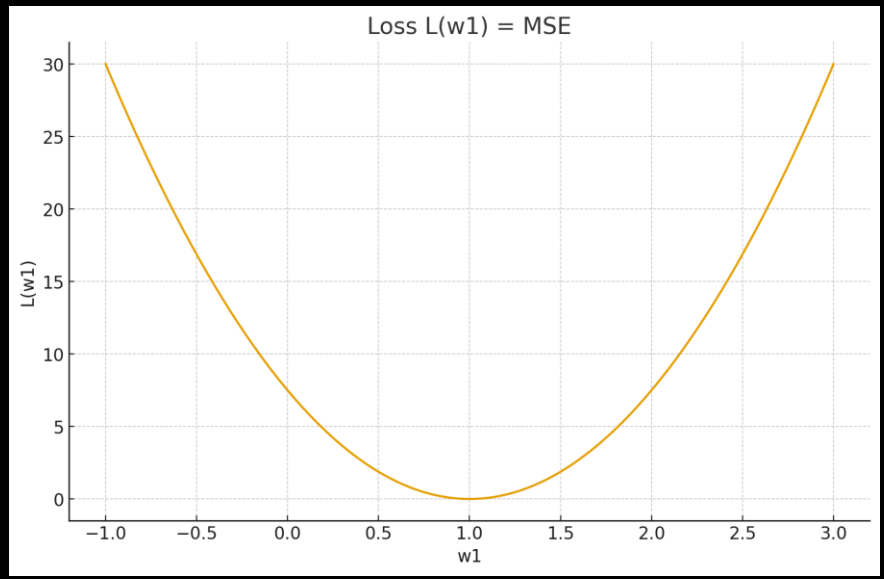
- $\boldsymbol{D} = \{(\mathbf{1},\mathbf{1}),(\mathbf{2},\mathbf{2}),(\mathbf{3},\mathbf{3}),(\mathbf{4},\mathbf{4}),\}$

$$f_{\mathbf{w}}(x) = w_1 x \quad (w_0 = 0)$$

When

$$w_1 = 1.0$$





$$L(w_1) = ?$$

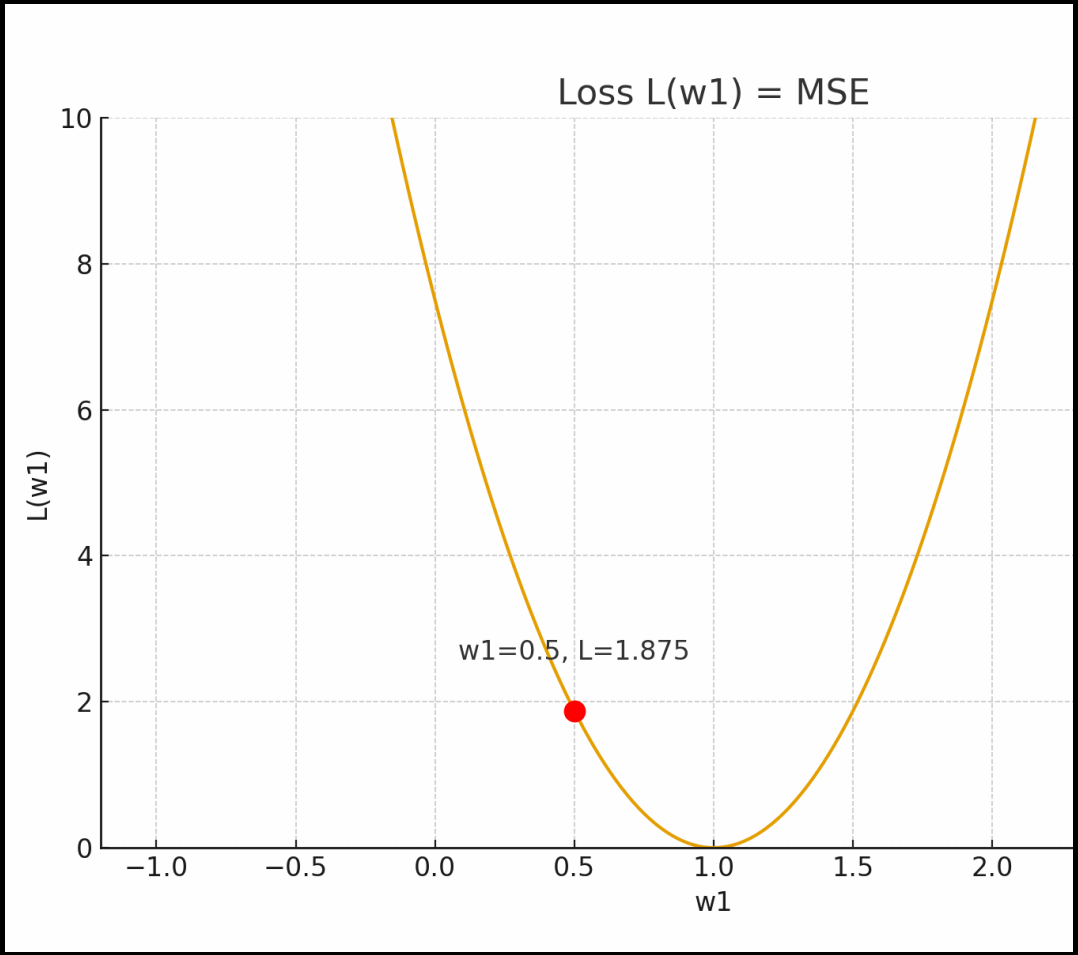
$When\ w_1 = 1.0$

x	Actual y	Predicted $f(x) = 1.0x$	Error $y - f(x)$	Squared Error $\left(y - f(x)\right)^2$
1	1	1.0	0	0
2	2	2.0	0	0
3	3	3.0	0	0
4	4	4.0	0	0

Sum of squared errors

= 0 + 0 + 0 + 0 = 0

$$L(w_1 = 1.0) = \frac{0}{4} = 0$$



This animation shows how the loss changes as we move the weight w_1.

First, at $w_1 = 0.5$, the loss is 1.875 because the slope is too small and the model underfits.

Then we move to $w_1 = 1.5$, the loss is again 1.875, showing that the loss curve is symmetric around the optimal value.

Finally, at $w_1 = 1.0$, The loss becomes zero.

This is the optimal slope since the model perfectly fits all data points.

Types of Loss Functions

- **There are various loss functions, but the most used ones are:**
  - Mean Square Error (MSE)

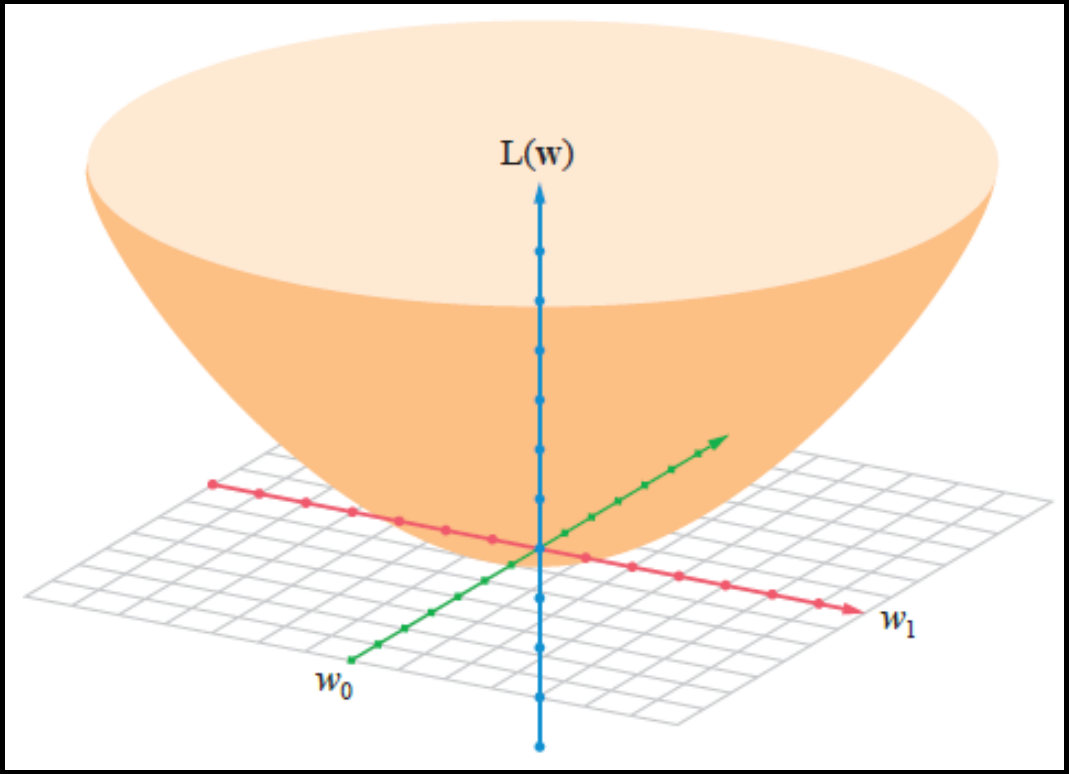
$$\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_{\mathbf{w}}(x_i)\right)^2$$

  - Mean Absolute Error (MAE)

$$\frac{1}{N}\sum_{i=1}^{N}\left|y_i - f_{\mathbf{w}}(x_i)\right|$$

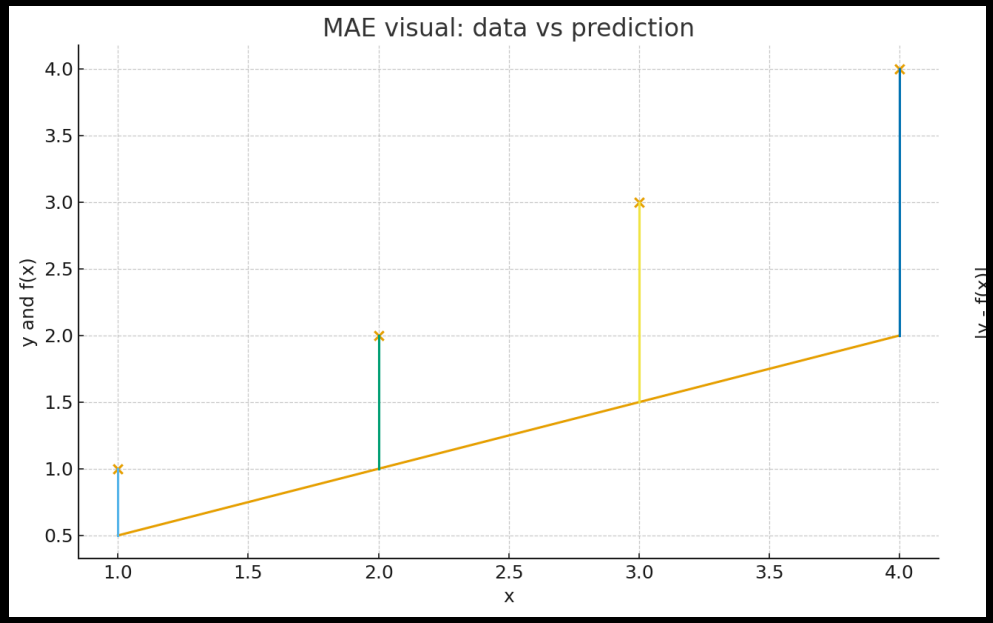
Loss function graph for simple linear regression

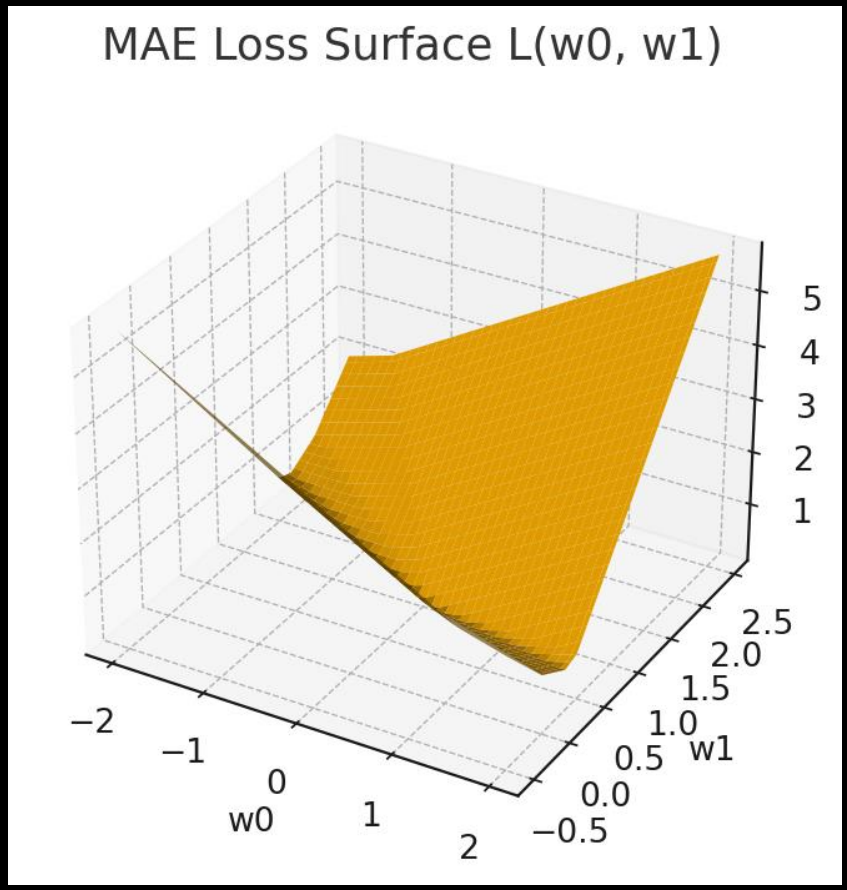
- **Commonly, MSE is typically convex**
  - A single global minimum
  - The shape of the graph depends on the parameter space, with a bowl-shaped surface where the minimum corresponds to the optimal weight w^*.



Loss function graph for simple linear regression

- **MAE (Average of all absolute errors)**
  - All errors contribute equally (no squaring)
  - Easy to interpret: “On average, predictions are off by this much”
  - More robust to outliers than MSE





Gradient Descent in SLR

Training Process of a Simple Linear Regression Model

- **Given the loss function $L(w)$**
  - The training process aims to find the parameters $w^* = [w_0^*, w_1^*]$ that minimize the loss function

$$w^* = \arg\min_w L(w)$$

  - The simplest method is to try several candidate values for the parameters $w = [w_0, w_1]^T$ in the model and select the one that results in the smallest loss function value.

  - However, determining the candidate values is not always straightforward.

Goal of Gradient Descent

- **Finding w^* by Minimizing the Loss Function $L(w)$**
- **Given the loss function $L(w)$,**
  - the goal is to find:

$$w^* = \arg\min_w L(w)$$

- **Gradient Descent**
  - Optimization algorithm that uses the gradient to minimize the given loss function.
    - The gradient at a point represents the slope of the tangent to the function at that point.
    - This slope indicates the direction in which the function increases the most.
    - Therefore, the negative gradient direction of the current point $w = [w_0, w_1]^T$, given by

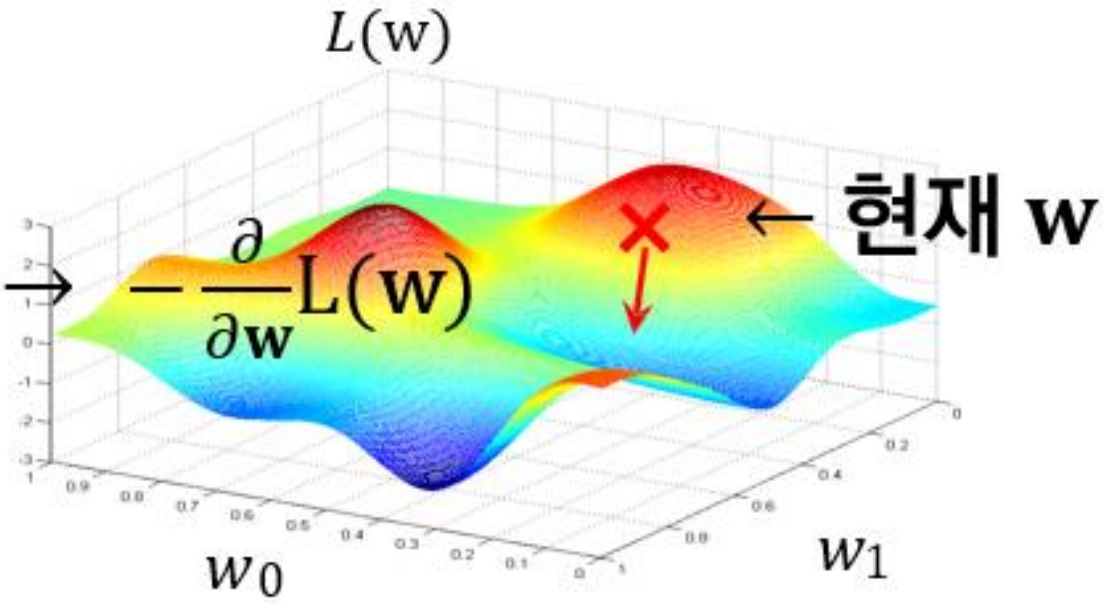
$$-\frac{\partial}{\partial w} L(w)$$

is the direction that maximally decreases the loss function $L(w)$,

and this is called the gradient descent direction.

Goal of Gradient Descent

음의 기울기 방향: $-\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w})$



Gradient Descent Procedure

- **Start from an initial value** $w = [w_0, w_1]^T$

  (e.g., $w_0 = 0, w_1 = 1$).

- **Iteratively update** w **in the direction that reduces** $L(w)$ **until the stopping condition is met.**

1. **Update for** w_0 **(bias term):**

$$w_0 \leftarrow w_0 - \alpha \frac{\partial}{\partial w_0} L(w_0, w_1)$$

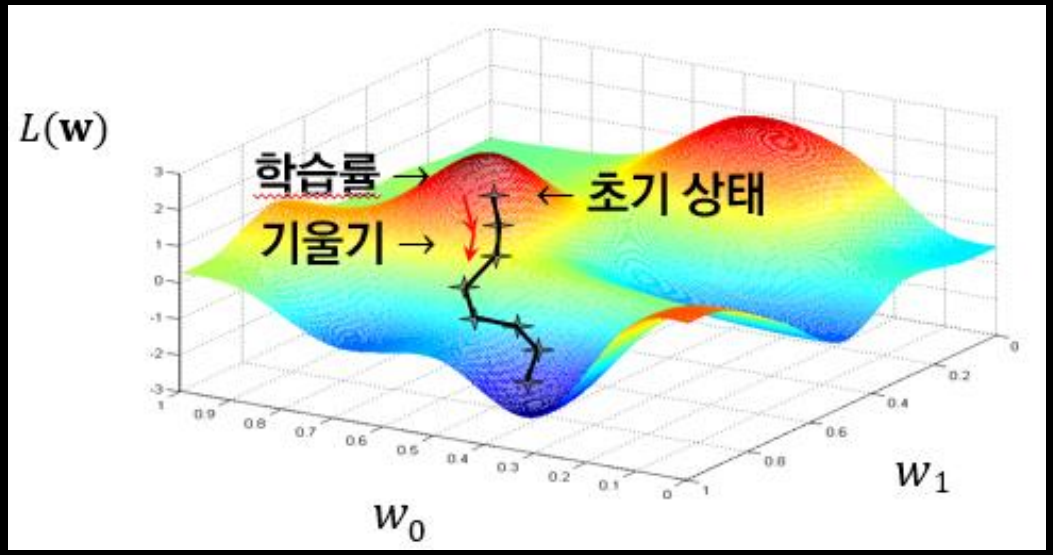
2. **Update for** w_1 **(weight coefficient):**

$$w_1 \leftarrow w_1 - \alpha \frac{\partial}{\partial w_1} L(w_0, w_1)$$

Applying Gradient Descent

- **Following factors must be considered:**
  - Initial Value: Where should the optimization start?
  - Gradient: In which direction should the parameters be updated?
  - Learning Rate: By how much should the parameters be updated each step?

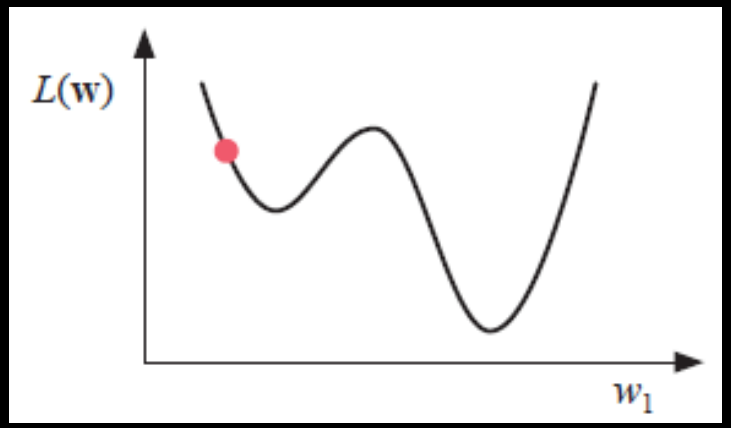
- **These factors influence the convergence speed and accuracy of the optimization process.**

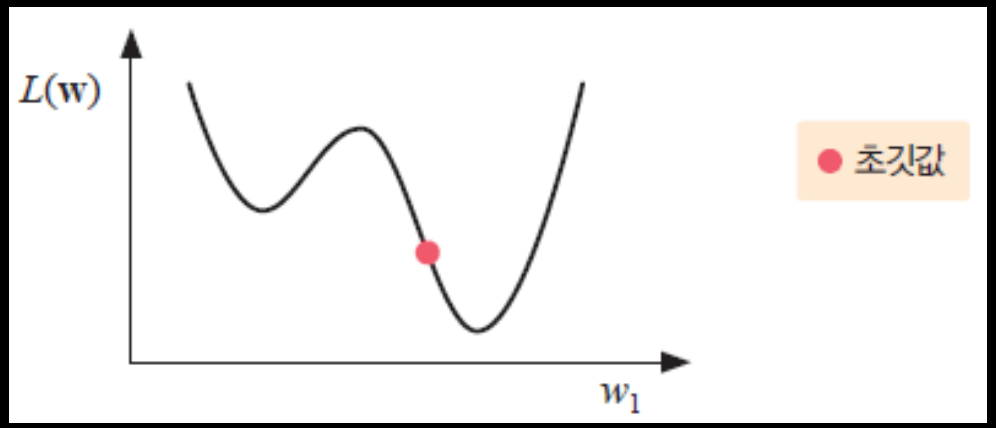


Applying Gradient Descent: Initial Value

- **Initial Value: Where to Start?**
- **Choice of the initial value**
  - can make it difficult to find the global minimum of the loss function or
  - significantly affect the time required to find it.

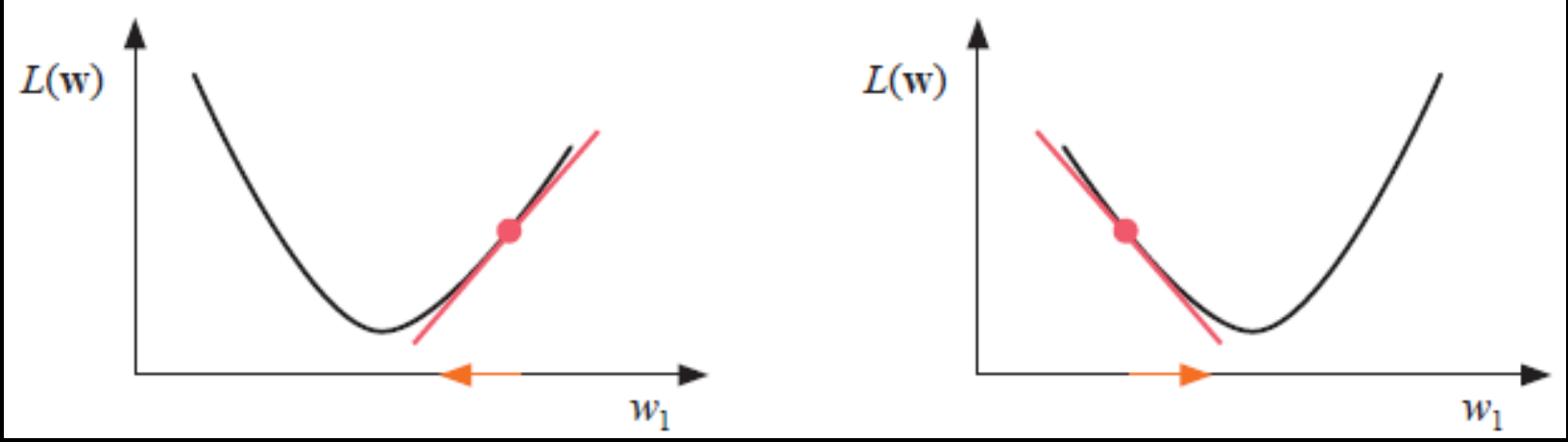
- **Initial state of w is an important factor in optimization.**





Applying Gradient Descent: Gradient

- **Gradient: In Which Direction to Update?**
  - When the gradient is positive → The loss function value is increasing
    - multiply the gradient by negative (-): move in the negative direction.
  - When the gradient is negative → The loss function value is decreasing
    - multiply the gradient by negative (-): move in the positive direction.



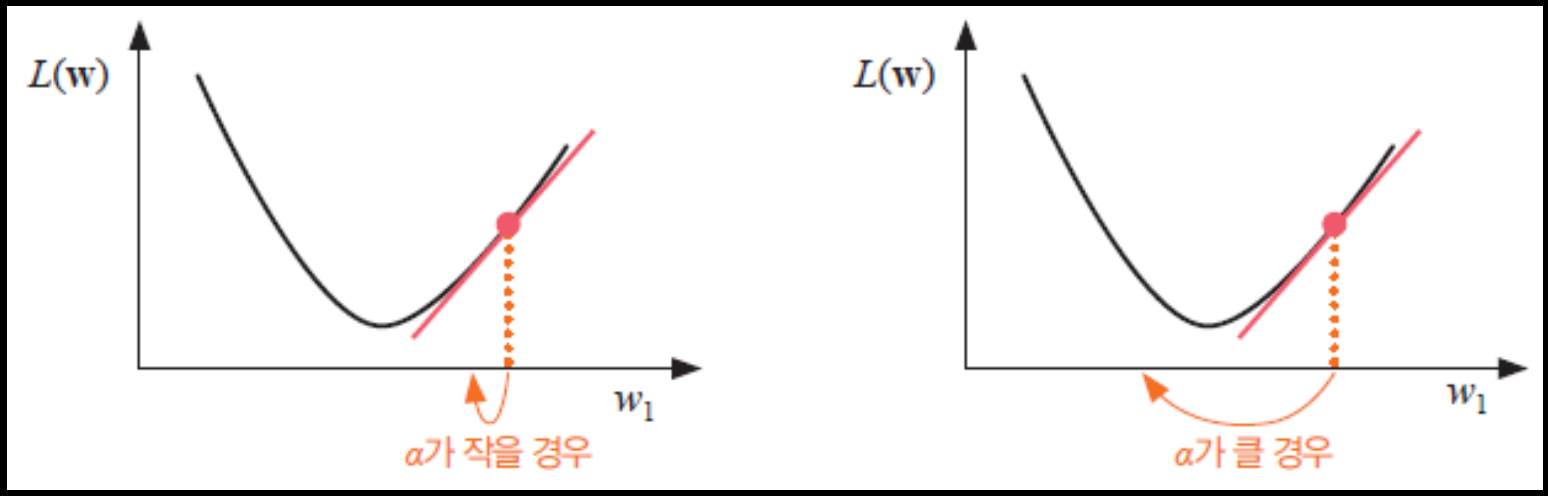
Applying Gradient Descent: Learning Rate

■ Learning Rate: How Much to Update?

- If the learning rate is too high → The loss function may diverge instead of converging.
- If the learning rate is too low

  → The convergence to the minimum can be slow or

  might not reach the minimum at all.
- Thus, selecting an appropriate learning rate based on experience and dynamically adjusting it is crucial.



Multiple Linear Regression

Multiple Linear Regression

- **Simple Linear Regression**
  - Relationship between one independent variable and one dependent variable.
- **Multiple Linear Regression**
  - Relationship between two or more independent variables and one dependent variable.

크기(feet^2)	방 개수	층 개수	집 연식	집 가격(1,000달러 기준)
2,105	5	1	40	469
1,325	3	3	30	230
1,425	3	2	20	300
872	2	1	40	170
⋮	⋮	⋮	⋮	⋮

$\mathbf{x}_i = [x_{i1}\ x_{i2}\ \cdots\ x_{ij}\ \cdots\ x_{id}]^T$ y_i

Representation of Multiple Linear Regression

$$f_{\mathbf{w}}(\mathbf{x}_i) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i$$

$$\mathbf{x}_i = \begin{bmatrix} x_{i0} & x_{i1} & x_{i2} & x_{i3} & \cdots & x_{ij} & \cdots & x_{id} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{d+1}$$

$$\mathbf{w} = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_j & \cdots & w_d \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{d+1}$$

- $\mathbf{x}_i$: 학습/테스트 데이터의 i번째 입력. x_{ij}는 i번째 입력의 j번째 특징값
- $\mathbf{w}$: 회귀 모델의 파라미터(매개변수 또는 가중치)

Training of MLR

- **By applying gradient descent using Mean Squared Error (MSE) as the loss function, the parameters**

$$\boldsymbol{W} = [\boldsymbol{w_0}, \boldsymbol{w_1}, \cdots, \boldsymbol{w_d}]^{\boldsymbol{T}} \in \mathbb{R}^{\boldsymbol{d+1}}$$

are updated iteratively

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - f_{\mathbf{w}}(x_i))^2$$

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} L(\mathbf{w})$$

$$= w_j - \alpha \frac{\partial}{\partial w_j} \frac{1}{N}\sum_{i=1}^{N}\left\{y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right\}^2$$

$$= w_j + \alpha \frac{2}{N}\sum_{i=1}^{N}\left\{y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right\} x_{ij}$$

In the Next Lecture

- **We will explore real world problem**
  - Practice & Exercise!
  - have a fun!

수고하셨습니다 ..^^..
Thank you!