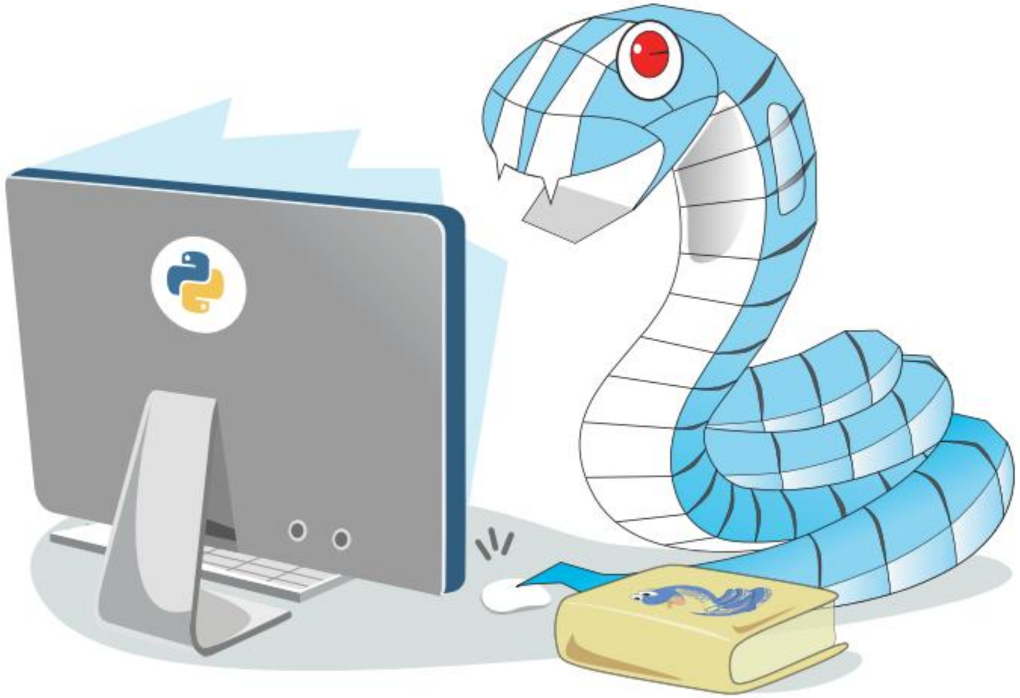


2장 머신러닝을 위한 기초지식

이장에서 배울 것들

- 이 책의 내용을 이해하기 위해서 필요한 수학적 개념을 알아보자.
- 모델, 파라미터, 학습은 어떻게 동작하는가.
- 오차란 무엇이고 이것으로 어떻게 학습을 할 수 있는가.
- 학습과정은 무엇을 최적화하며 하이퍼파라미터란 무엇인가.
- 파라미터와 하이퍼파라미터는 각각 무엇에 영향을 미치는가.

2.1 수학 표기

- 효과적이고 효율적인 설명이 수학적 도구에 의해서만 가능한 경우가 많다. 선형 대수학linear algebra과 미적분, 확률과 통계 등의 지식이 머신러닝에 빈번히 등장한다.

수, 벡터, 행렬

표기법	의미	특징 및 주의
a	스칼라 변수	보통 굵기의 이탤릭체
A	스칼라 상수	대문자
$\mathbf{a}$	벡터	굵은 정자체 소문자
$\mathbf{A}$	행렬	굵은 정자체 대문자
$\mathbf{I}_n$	$n \times n$ 차원 항등행렬	아래 첨자로 차원 표기

예시

수, 벡터, 행렬

표기법	의미
a	스칼라 변수
A	스칼라 상수
$\mathbf{a}$	벡터
$\mathbf{A}$	행렬
$\mathbf{I}_n$	$n \times n$ 차원 항등행렬

$a = 5$

$A = 5.323412$

$\mathbf{a} = (5, 2, 1)^T$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

집합

표기법	의미	특징 및 주의
$\mathbb{R}$	실수 집합	대문자 외곽선
$\mathbb{R}^n$	n 차원 실수 벡터 집합	윗 첨자로 벡터의 차원 표기
$\mathbb{R}^{n \times m}$	n 행 m 열의 행렬 집합	윗 첨자로 행과 열의 수 표기
$\in$	원소	왼쪽이 오른쪽 집합의 원소
$\mathbf{A} \in \mathbb{R}^{n \times m}$	$\mathbf{A}$ 는 n 행 m 열의 행렬	$\mathbf{A}$ 가 $\mathbb{R}^{n \times m}$ 의 원소

집합

표기법
$\mathbb{R}$
$\mathbb{R}^n$
$\mathbb{R}^{n \times m}$
$\in$
$\mathbf{A} \in \mathbb{R}^{n \times m}$

$$x \in \mathbb{R}$$

$$\mathbf{v} = \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} \in \mathbb{R}^3$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \in \mathbb{R}^{3 \times 5}$$

선형 대수

표기법	의미	특징 및 주의
$\mathbf{A}^{\mathrm{T}}$	행렬 $\mathbf{A}$의 전치	첨자 T로 전치 표현
$\mathbf{A}^{-1}$	행렬 $\mathbf{A}$의 역행렬	첨자 –1로 곱셈에 대한 역원 표현
$\mathrm{tr}(\mathbf{A})$	행렬 $\mathbf{A}$의 대각성분 합trace	함수 표현
$\mathbf{A} \otimes \mathbf{B}$	아다마르Hadamard 곱	연산자에 원을 씌어 원소별 연산 표현

선형 대수

표기법
$\mathbf{A}^{\mathrm{T}}$
$\mathbf{A}^{-1}$
$\mathrm{tr}(\mathbf{A})$
$\mathbf{A} \otimes \mathbf{B}$

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad \mathrm{A}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \mathrm{A}^{-1} = \frac{1}{-2} \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$$

$$A \times A^{-1} = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad \mathrm{trace}(A) = 1 + 5 + 9 = 15$$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \qquad \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} 7 & 16 & 27 \\ 40 & 55 & 72 \end{bmatrix}$$

미적분

표기법	의미	특징 및 주의
$f : \mathbb{R}^n \to \mathbb{R}^m$	함수	$\mathbb{R}^n$ 에 속한 값을 $\mathbb{R}^m$ 으로 옮기는 함수
f'	함수 f의 미분	어깨점으로 미분 표현
$\frac{df}{dx}$	함수 f의 x에 대한 미분	미분의 변수를 명시적으로 표현
$f'(a)$	a 위치에서 함수 f의 미분	특정 지점에서의 미분, $f : \mathbb{R} \to \mathbb{R}^*$
$\frac{\partial f}{\partial x_i}$	다차원 입력 함수 f의 x_i 에 대한 편미분	x_i 제외한 모든 입력 변수를 상수로 취급, $f : \mathbb{R}^n \to \mathbb{R}^*$
$\frac{\partial f}{\partial x_i}(\mathbf{a})$	입력 벡터 $\mathbf{a}$ 위치에서 함수 f의 x_i 에 대한 편미분	입력 벡터 $\mathbf{a}$ 위치에서 x_i를 제외한 모든 변수를 상수로 취급하여 미분
∇f	함수 f의 기울기	$f : \mathbb{R}^n \to \mathbb{R}$ 의 기울기
$\nabla_a f$	함수 f의 기울기	벡터 a에 대한 함수 f의 기울기

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad M = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \qquad \boxed{f(\mathbf{v}) = M\mathbf{v}}$$

$$\mathbf{w} = f(\mathbf{v}) = \begin{bmatrix} a \cdot x + b \cdot y + c \cdot z \\ d \cdot x + e \cdot y + f \cdot z \end{bmatrix}$$

$$\mathbf{v} \in \mathbb{R}^3, \quad \mathbf{w} = f(\mathbf{v}) = M\mathbf{v}, \quad \mathbf{w} \in \mathbb{R}^2$$

$$\boxed{f : \mathbb{R}^3 \to \mathbb{R}^2}$$

$$f'$$

$$f(x) = x^2 + 3x + 2$$

$$f'(x) = \frac{d}{dx}(x^2) + \frac{d}{dx}(3x) + \frac{d}{dx}(2) \quad = 2x + 3$$

$$\frac{df}{dx}$$

$$\frac{df}{dx} = 2x + 3$$

$$f'(a)$$

$$f'(1) = 2(1) + 3 = 2 + 3 = 5$$

$$\frac{\partial f}{\partial x_i}$$

$f(x, y, z) = x^2y + 3xy^2z + z^3$

$\frac{\partial f}{\partial x} = 2xy + 3y^2z$

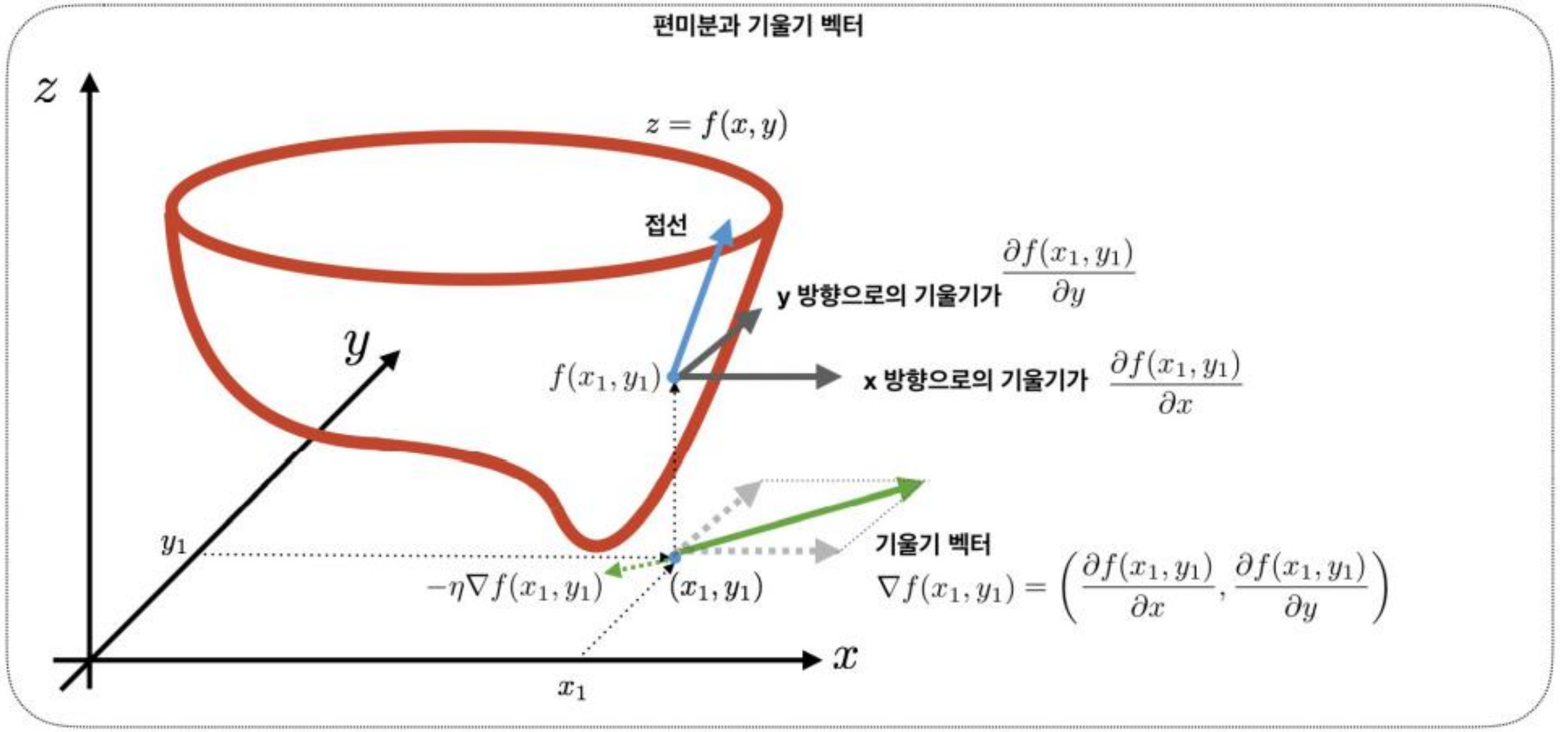
$f(x, y, z) = \quad Ax^2 + Bx + C$

$\frac{\partial f}{\partial x} = 2Ax + B \quad = 2xy + 3y^2z$

- $A = y$
- $B = 3y^2z$
- $C = z^3$

$$\nabla f$$

$$\nabla f(x, y, z) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$$



확률과 통계

표기법	의미	특징 및 주의
$p(X)$	조건 X가 일어날 확률	X는 $a = b$ 와 같이 참 또는 거짓인 값
$p(X, Y)$	결합 확률	X와 Y가 함께 일어날 확률
$p(Y \mid X)$	조건부 확률	X가 참일 때, Y가 일어날 확률
μ	평균	데이터를 대표하는 값
σ, σ^2	표준편차, 분산	$\sigma(\ )$처럼 표현하면 시그모이드 함수

2.2 벡터와 행렬

- 머신러닝은 다수의 특징값을 원소로 하는 데이터를 입력으로 제공하고, 컴퓨터에게 분류나 판단, 그리고 행위를 하게 한다.
- 다수의 원소를 가진 데이터를 벡터 데이터.
  - 데이터는 머신러닝 모델에 제공된 뒤 다양한 변화를 겪게 되는데, 이러한 변화 과정은 벡터에 행렬을 곱하는 일로 표현되는 경우가 많다.
  - 벡터와 행렬을 다루는 대표적인 수학 분야가 선형대수.

- 스칼라scalar란 1, 2, 3.14,...와 같이 크기 값을 가지는 양이며, 벡터vector는 순서가 있는 스칼라 값의 배열을 의미하며, 일반적으로 하나의 열로 내려가며 표기하는 열벡터 표기법을 사용
- 열벡터를 전치시킬 경우 행벡터로 변환시킬 수 있다.

$$\mathbf{x} = \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{vmatrix} \qquad \mathbf{x}^{\mathrm{T}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

- 위와 같이 표현되는 n차원 벡터 x는 $\mathbb{R}^n$의 원소로 다음과 같이 표현한다.

$$\mathbf{x} \in \mathbb{R}^n$$

- 스칼라 값이 크기만을 가진다면, 벡터는 크기와 방향을 갖는 값
- 벡터의 크기를 벡터의 노름[norm]이라고 한다. 벡터 x의 p차 노름은 다음과 같이 정의

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$

- $p = 1$이라면 각 차원별 거리의 합인 맨해튼 거리[Manhatten distance]가 되고, $l1$ 노름이라 한다.
- $p = 2$ 이면 벡터의 시작점에서 끝점을 화살표로 표현할 때 화살표의 길이와 같은 유클리드 노름[Euclidian norm]이나, $l2$ 노름이라 부른다.
- p를 생략하고 노름을 사용할 경우 일반적으로 $p = 2$ 인 $l2$ 노름이다.

- 벡터를 유클리드 노름의 값으로 나누면 길이가 1인 벡터
- 길이가 1인 벡터를 **단위 벡터**unit vector
- 벡터를 길이 1인 벡터로 만드는 일은 벡터의 **정규화**normalization

$\tilde{\mathbf{x}}$ 는 크기가 언제나 1이므로 방향을 표현하는 값이라고 생각할 수 있다.

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$$

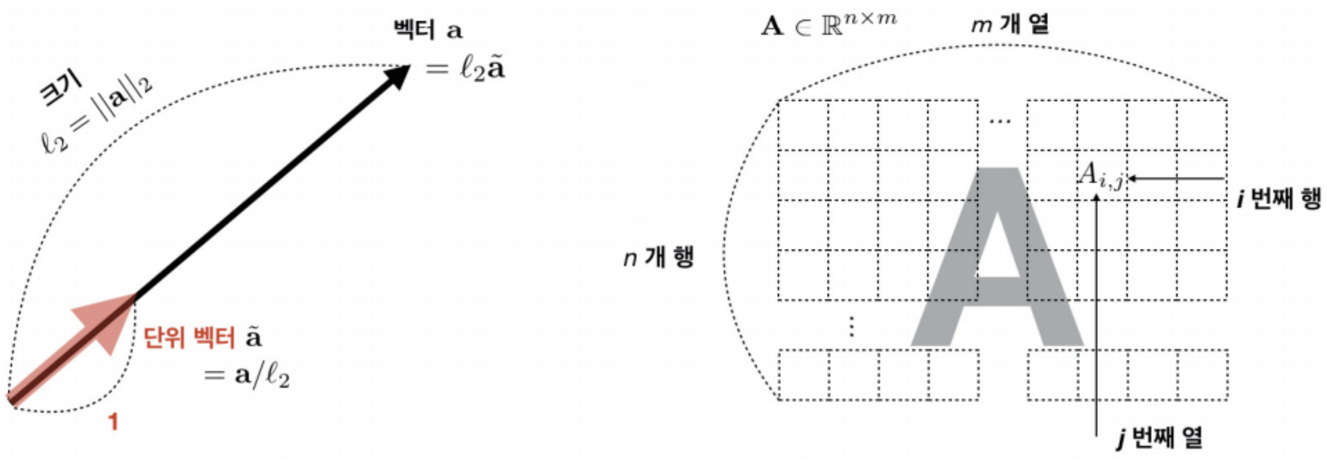
우리는 어떤 벡터 $\mathbf{x}$를 크기 $\|\mathbf{x}\|_2$와 방향 $\tilde{\mathbf{x}}$의 곱으로 표현할 수 있다.

$$\mathbf{x} = \|\mathbf{x}\|_2 \cdot \tilde{\mathbf{x}} = l_2 \times \text{방향}$$

- 행렬matrix은 스칼라 값이나 자료를 행row과 열column로 이루어진 배열의 형태로 나타내는 것
- 행의 수가 n이고 열의 수가 m개인 행렬을 $n \times m$ 행렬이라고 부른다.
- 어떤 행렬 A가 이러한 행렬이라면 다음과 같이 표현

$$\mathbf{A} \in \mathbb{R}^{n \times m}$$

- 행렬의 i행 j열 원소는 $A_{i,j}$로 표현, 하나의 스칼라 값이 된다.
- 벡터는 아래 그림의 왼쪽과 같은 화살표로, 행렬은 오른쪽과 같은 2차원 배열 형태



2.3 벡터와 행렬의 기본 연산

- 두 벡터를 더하거나 빼기 위해서는 두 벡터의 차원이 동일
- 두 벡터 a와 b가 모두 동일 차원의 벡터로 $\mathbb{R}^n$에 속한다고 할 때, 두 벡터를 더하거나 빼서 얻는 새로운 벡터 c는 다음과 같이 구할 수 있다.

$$\mathbf{c} = \mathbf{a} \pm \mathbf{b} \Leftrightarrow c_i = a_i \pm b_i$$

- 가장 간단하면서 벡터 연산 여러 영역에서 유용한 곱은 아다마르Hadamard 곱
- 벡터 덧셈과 뺄셈과 동일한 방식으로 원소별 곱을 구하여 새로운 벡터를 만든다.

$$\mathbf{c} = \mathbf{a} \otimes \mathbf{b} \Leftrightarrow c_i = a_i b_i$$

- 벡터의 곱으로 흔히 사용되는 점곱dot product과 가위곱cross product는 각각 스칼라곱, 벡터곱
- 점곱은 스칼라값, 가위곱은 벡터
- 점곱은 다음과 같이 계산할 수 있으며, 아다마르곱으로 얻어지는 벡터의 모든 원소를 합한 것

$$\mathbf{a}\cdot\mathbf{b}=\sum_{i=1}^{n}a_1b_1=a_1b_1+a_2b_2+\cdots+a_nb_n=\sum_{i=1}^{n}(\mathbf{a}\otimes\mathbf{b})_i$$

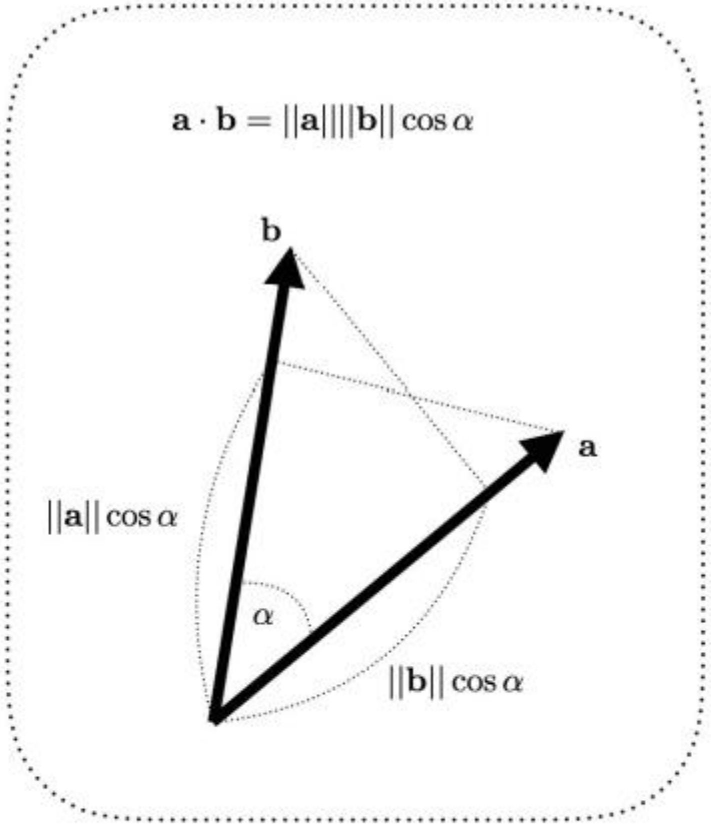
- 점곱이 갖는 기하적 특성
  - 값이 두 벡터가 이루는 사잇각 α의 코사인cosine 값에 비례
  - 두 벡터의 크기에 각각 비례한다는 것

$$\mathbf{a}\cdot\mathbf{b}=\|\mathbf{a}\|\|\mathbf{b}\|\cos\alpha$$

- 가위곱은 벡터곱이므로 크기와 방향을 갖는 새로운 벡터
- 크기는 두 벡터의 크기와 사잇각의 사인sine 값에 비례
- 방향은 두 벡터에 동시에 수직인 방향의 단위 벡터 d

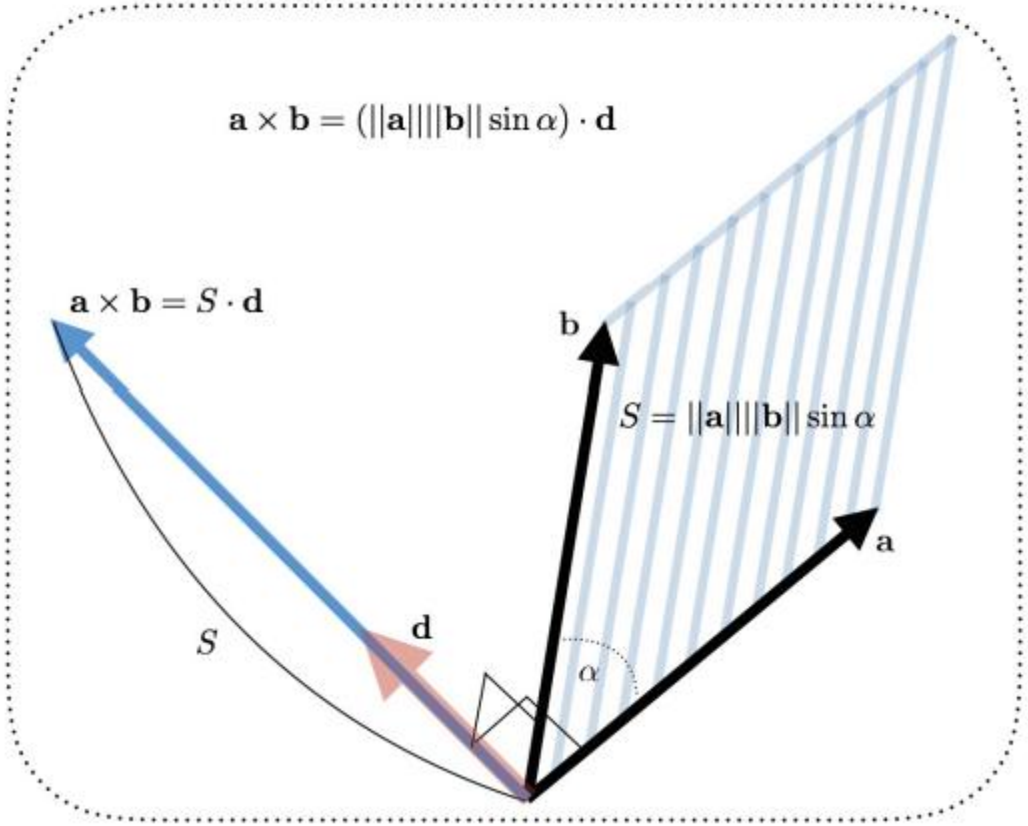
$$\mathbf{a} \times \mathbf{b} = (\|\mathbf{a}\|\|\mathbf{b}\| \sin \alpha) \cdot \mathbf{d}$$

점곱



$\mathbf{a}\cdot\mathbf{b} = ||\mathbf{a}||||\mathbf{b}||\cos\alpha$
$\mathbf{b}$
$\mathbf{a}$
$||\mathbf{a}||\cos\alpha$
α
$||\mathbf{b}||\cos\alpha$

가위곱



$\mathbf{a}\times\mathbf{b} = (||\mathbf{a}||||\mathbf{b}||\sin\alpha)\cdot\mathbf{d}$
$\mathbf{a}\times\mathbf{b} = S\cdot\mathbf{d}$
$\mathbf{b}$
$S = ||\mathbf{a}||||\mathbf{b}||\sin\alpha$
$\mathbf{a}$
$\mathbf{d}$
S
α

- 두 행렬 A와 B를 곱할 때는 A의 열의 개수와 B의 행의 개수가 일치

$$\mathbf{A} \in \mathbb{R}^{n \times m}$$
$$\mathbf{B} \in \mathbb{R}^{m \times l}$$

- 이때 두 행렬의 곱 C는 $\mathbb{R}^{n \times l}$ 집합에 속하며 각 원소는 다음과 같이 계산

$$\mathbf{C} = \mathbf{AB}$$
$$C_{i,j} = \sum_{k=1}^{m} A_{i,k}B_{k,j} = A_{i,1}B_{1,j} + A_{i,2}B_{2,j} + A_{i,3}B_{3,j} + \cdots + A_{i,m}B_{m,j}$$

- 행렬 A의 i 행 벡터를 $\mathrm{A}_{i,*}$ 라고 하고, B의 j열 벡터를 $\mathrm{B}_{*,j}$라고하면, 행렬의 곱 C의 i 행 j 열 원소는 두 벡터의 점곱

$$C_{i,j} = \mathbf{A}_{i,*}\mathbf{B}_{*,j} = \mathbf{A}_{i,*}^{T} \cdot \mathbf{B}_{*,j}$$
$$C_{i,j} = \begin{bmatrix} A_{i,1} & A_{i,2} & \cdots & A_{i,m} \end{bmatrix} \begin{bmatrix} B_{1,j} \\ B_{2,j} \\ \vdots \\ B_{m,j} \end{bmatrix} = \begin{bmatrix} A_{i,1} \\ A_{i,2} \\ \vdots \\ A_{i,m} \end{bmatrix} \cdot \begin{bmatrix} B_{1,j} \\ B_{2,j} \\ \vdots \\ B_{m,j} \end{bmatrix}$$

- 두 벡터 a와 b의 점곱을 표현할 때 벡터를 1개 열을 가진 행렬로 보면 $\mathrm{a} \cdot \mathrm{b}$를 다음과 같은 행렬곱 표현
- 벡터 점곱을 이렇게 전치를 이용해 표현하는 경우가 매우 빈번

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b}$$

2.4 미분과 기울기, 그리고 경사 하강법의 개념

- 머신러닝에 사용되는 여러 기법들을 이해하기 위해 필요한 수학적인 개념 중에서 가장 중요한 개념은 바로 미분 derivative
    - 미분이란 순간 변화량을 구하는 것
    - 독립 변수값의 변화량에 대한 함수값 변화량 비의 극한

$$f'(a) = \lim_{\Delta x \to 0} \frac{f(a + \Delta x) - f(a)}{\Delta x}$$

- 연쇄법칙chain rule은 다음과 같이 어떤 함수 y를 x에 대해 미분할 때, 매개 변수 t를 두어 다음과 같이 미분

$$\frac{dy}{dx} = \frac{dy}{dt}\frac{dt}{dx}$$

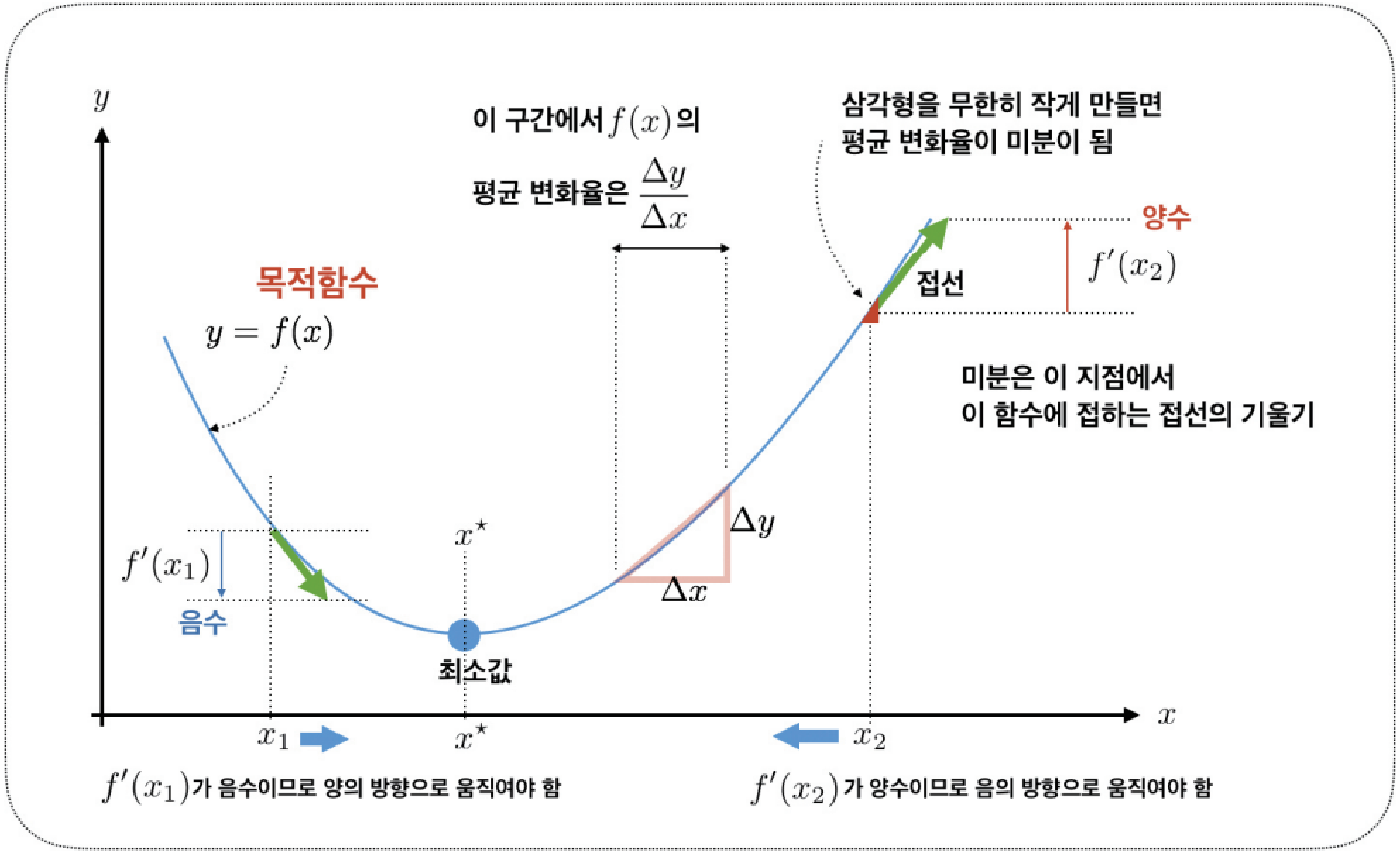
- 이것은 아래 테이블의 오른쪽 하단에 있는 $y = f(g(x))$와 같은 합성함수에 대한 미분을 할 때 유용
  - $t = g(x)$, $y = f(t)$ 로 두고 다음과 같이 미분을 수행

$$\frac{d}{dx}[f(g(x))] = \frac{d}{dg(x)}f(g(x))\frac{dg(x)}{dx} = f'(g(x))g'(x)$$

$\frac{d}{dx}(C) = 0$	$\frac{d}{dx}[Cf(x)] = Cf'(x)$
$\frac{d}{dx}x^n = nx^{n-1}$	$\frac{d}{dx}e^x = e^x$
$\frac{d}{dx}[f(x) \pm g(x)] = f'(x) \pm g'(x)$	$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$

미분의 기본 공식 (함수: *f(x), g(x),* 상수: *C*)

- 함수 $f(x)$의 1차 미분 $f'(x)$는 x가 매우 조금 변화한 정도에 대해 함수값이 어떤 비로 변화하는지 변화율
  - 머신러닝에서 최적화optimization 작업을 할 수 있다.
- 평균 변화율: 변수 x가 Δx 만큼 변할 때, 목적함수는 Δy 만큼 변한다. 이것의 비가 이 구간의 평균 변화율
- 미분과 접선의 기울기: $x = x_2$인 지점에서 이 삼각형을 매우 작게 만들었다. 이 삼각형을 무한히 작게 만들면 이 것이 바로 x_2지점에서의 목적함수 미분이다.
  - 목적함수 곡선에 접하는 선, 즉 접선의 순간변화율
- 경사하강: 접선의 기울기를 따라 반대로 내려오면 해당 지점에서 목적함수의 값이 더 작은 쪽으로 이동할 수 있다. x_2의 위치에서 x가 증가하면 y 도 증가하므로 접선의 기울기는 양수이다.
  - x_2 에서 음의 방향으로 움직이면 목적함수를 줄일 수 있다. 이것을 경사 하강법gradient descent이라고 한다.



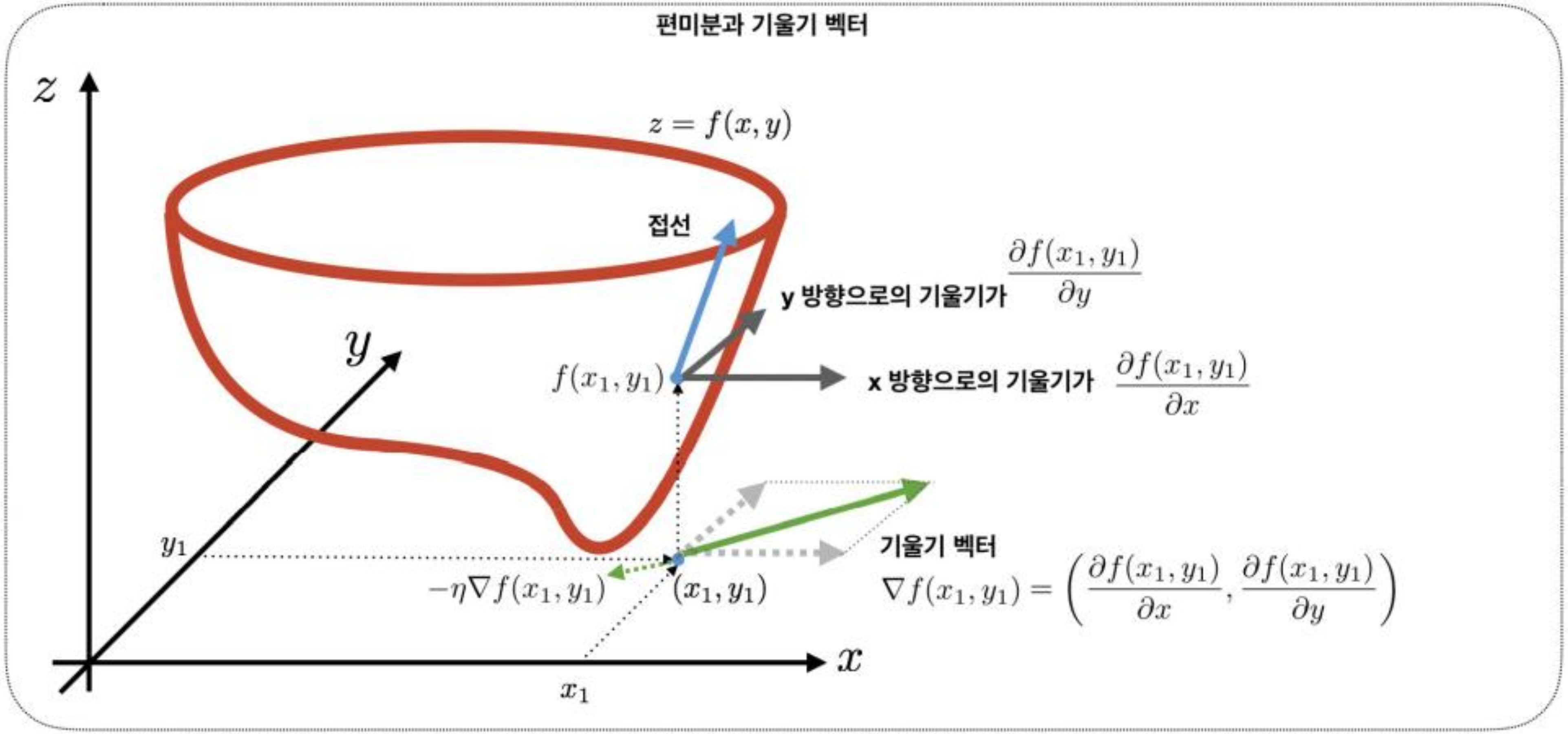
y
이 구간에서 $f(x)$의
평균 변화율은 $\frac{\Delta y}{\Delta x}$
삼각형을 무한히 작게 만들면
평균 변화율이 미분이 됨
양수
$f'(x_2)$
접선
목적함수
$y = f(x)$
미분은 이 지점에서
이 함수에 접하는 접선의 기울기
Δy
Δx
$x^\star$
$f'(x_1)$
음수
최소값
x
x_1
$x^\star$
x_2
$f'(x_1)$가 음수이므로 양의 방향으로 움직여야 함
$f'(x_2)$가 양수이므로 음의 방향으로 움직여야 함

2.5 편미분과 기울기

- 편미분partial derivative이란 둘 이상의 변수들을 가지는 함수 f가 있을 경우, 이 함수를 각각의 변수에 대해서 독립적으로 미분을 하는 방식
    - 다음과 같은 x, y두 변수로 이루어진 함수 $f(x, y) = x^2 + xy + y^2$가 있을 경우 x, y에 대한 편미분은 다음과 같이 각각 구할 수 있다.

$$\frac{\partial f}{\partial x}(x, y) = 2x + y, \quad \frac{\partial f}{\partial y}(x, y) = x + 2y$$

편미분을 사용하는 이유는 다차원 공간에서 정의되는 목적함수의 최적해를 찾기 위해서이다.


편미분과 기울기 벡터
z
$z = f(x, y)$
접선
$\frac{\partial f(x_1, y_1)}{\partial y}$
y 방향으로의 기울기가
y
$f(x_1, y_1)$
x 방향으로의 기울기가 $\frac{\partial f(x_1, y_1)}{\partial x}$
기울기 벡터
y_1
$-\eta \nabla f(x_1, y_1)$
(x_1, y_1)
$\nabla f(x_1, y_1) = \left(\frac{\partial f(x_1, y_1)}{\partial x}, \frac{\partial f(x_1, y_1)}{\partial y} \right)$
x
x_1

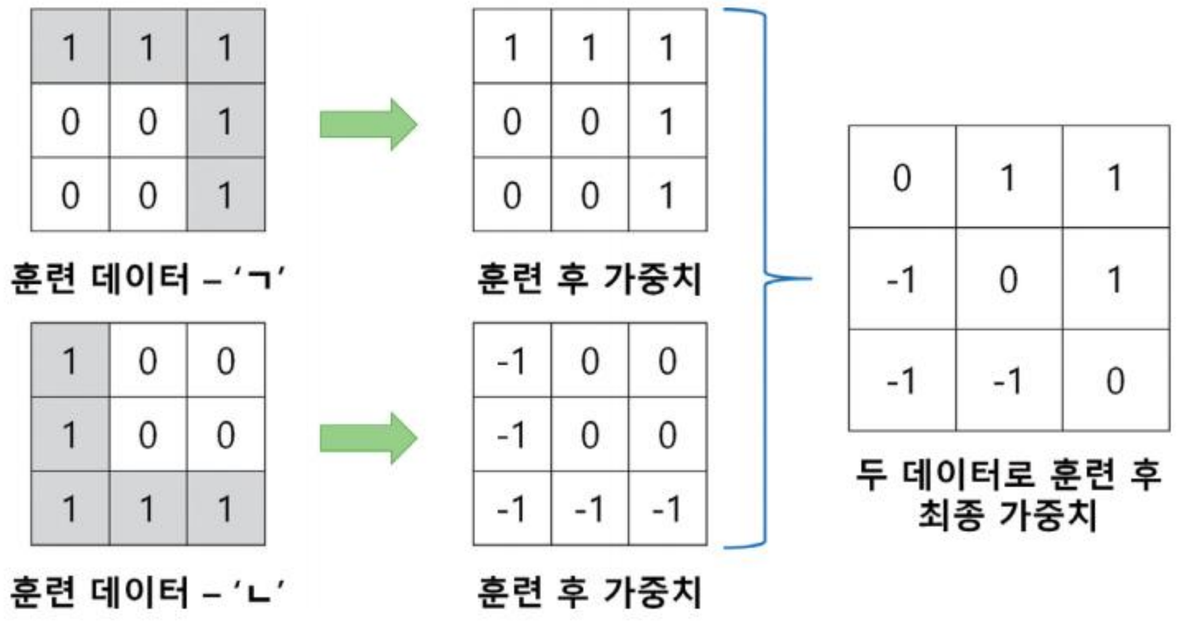
- 그림에서 $\partial f\,(x_1, y_1)/\partial\, x$는 목적함수의 $f(x_1, y_1)$에 닿는 접선이 x축 방향으로 갖는 기울기를 의미함. 비슷하게 $\partial f\,(x_1, y_1)/\partial\, y$는 y 축 방향 기울기
- $x \in \mathbb{R}^{n}$ 의 벡터를 입력으로 하는 함수 $f(x)$ 의 기울기 벡터 $\nabla f(x)$는 모든 차원의 기울기를 원소로 하는 벡터로 다음과 같이 정의

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \cdots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)$$

- 기울기 벡터는 위의 그림에서 녹색으로 표시된 화살표
- η(eta)는 기울기 벡터의 크기를 얼마나 고려하여 이동할지를 결정하는 것으로 이 값을 학습률learning rate이라고 한다.

2.6 모델, 파라미터, 그리고 학습

- 머신러닝은 문제를 해결하는 모델model
- 모델의 동작을 결정하는 파라미터parameter
- 파라미터를 더 좋은 상태로 변경하는 학습learning 동작으로 이루어진다.

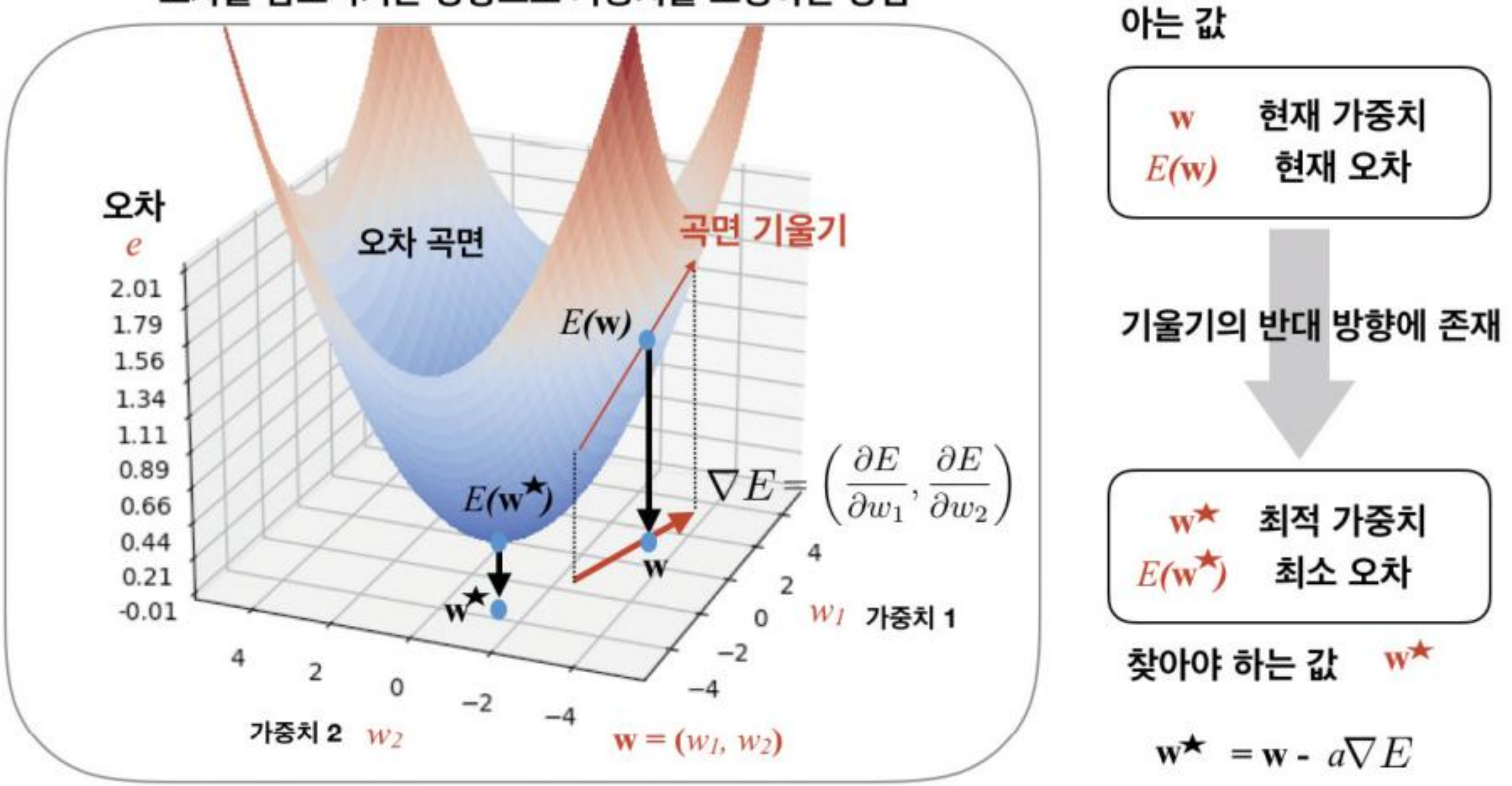


2.7 오차의 기울기를 이용한 학습의 기본 원리와 최적해

- 모델은 현재의 파라미터를 바탕으로 어떤 행위를 할 것이다. 그 결과는 보통은 차이가 난다.
    - 이것이 모델이 오차[error]
    - 오차가 없다 = 학습이 완벽하게 잘 되었다 = 모델이 데이터를 잘 설명한다고 볼 수 있다

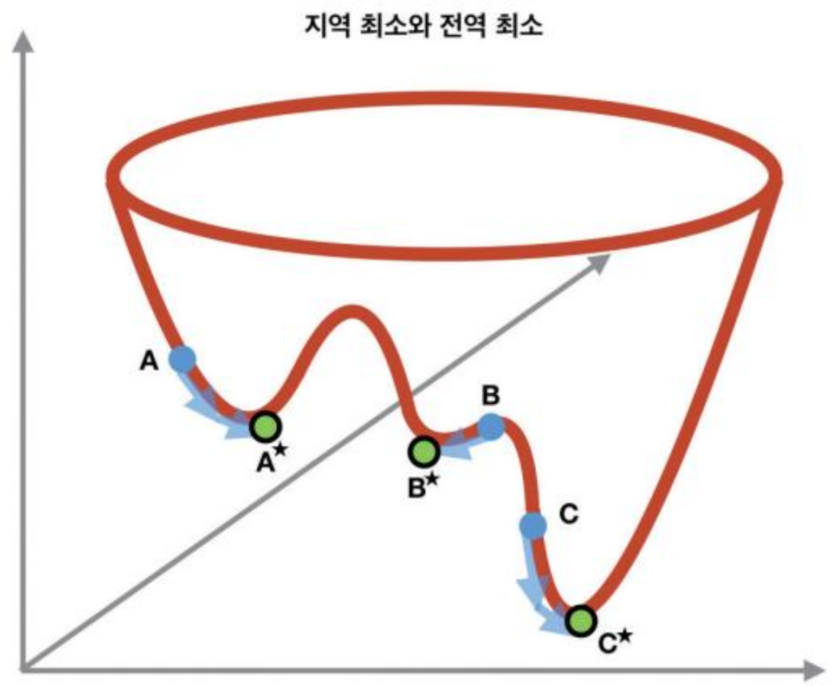
- 학습이란 이 오차가 줄어드는 방향(모델이 데이터를 잘 설명하는 방향)으로 파라미터를 변경하는 일

오차를 감소시키는 방향으로 가중치를 조정하는 방법



- 현재 가중치 w 위치에서 오차 곡면의 기울기를 안다면 기울기를 따라 내려가면 곡면을 낮은 곳으로 향할 수 있다.
- 최적해에 도착했다면 기울기 벡터가 0 벡터가 될 것이며 경사 하강법[gradient descent]을 통한 최적화[optimization]이다.

- 경사 하강법을 통해 얻은 답은 일정한 영역내에서 가장 좋은 답.
- 이렇게 일정한 영역 내에서 가장 좋은 지점을 지역 최소값local minimum
- 전체 공간에서 가장 좋은 해는 전역 최소값global minimum
- 좀더 일반적인 이름으로는 지역 최적값local optimum, 전역 최적값global optimum



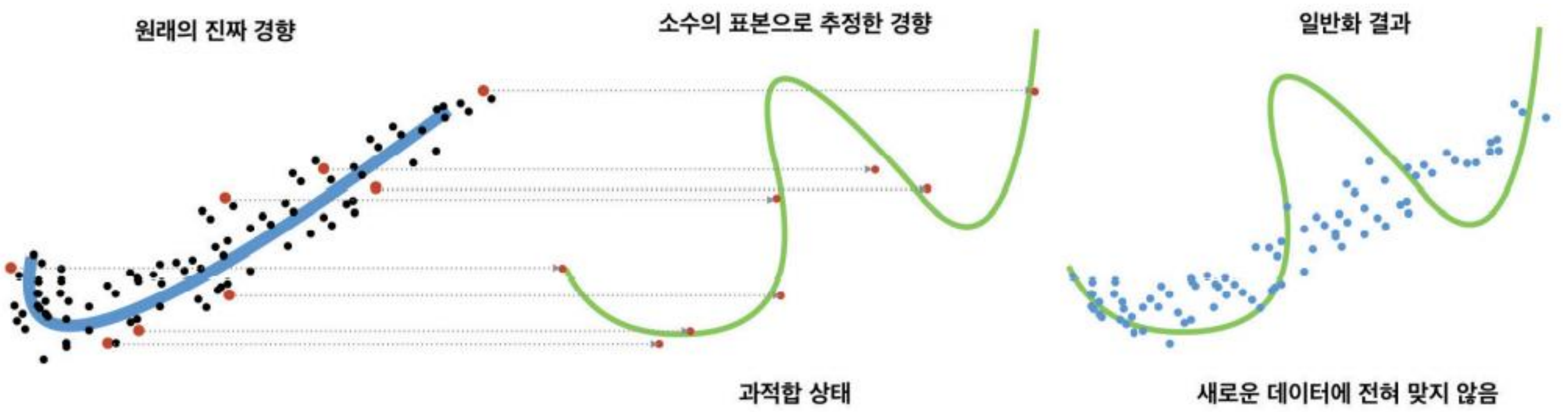
- 경사 하강법을 통한 학습이라는 것은 현재의 가중치 w 를 더 좋은 가중치 w'로 바꾸는 일

$$\mathbf{w}' = \mathbf{w} - \eta \nabla E$$

- 지역 최적값에 붙잡혀 전역 최적값을 찾지 못할 위험이 존재
- η(eta)는 기울기의 반대 방향으로 얼마나 이동할 것인지를 결정하는 학습률
    - 학습 과정에 영향을 미치는 값들을 하이퍼파라미터hyperparameter라고 한다.

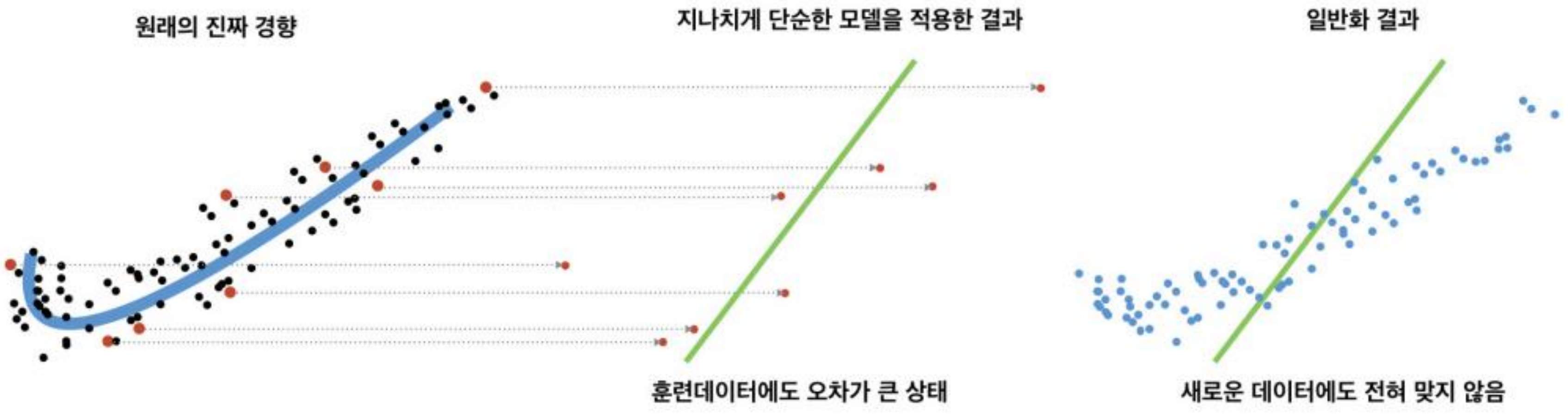
2.8 과적합, 과소적합, 그리고 일반화

- 머신러닝에서 흔히 만나는 문제가 과적합overfitting
- 과적합은 데이터를 분석한 결과가 특정한 데이터 집합에만 매우 정확하게 일치하고, 다른 데이터에는 잘 들어맞지 않는 상태를 의미
- 학습을 통해 모델의 파라미터를 결정한 뒤에 학습하지 않은 데이터에 모델을 적용하여 값을 예측하거나 분류를 수행하는 등의 데이터 분석을 실시하는 것이 일반화generalization
- 과적합이라는 것은 일반화가 제대로 되지 않는다는 것을 의미



- 과적합의 원인
  - 학습에 사용된 데이터의 수가 너무 적다는 것
  - 데이터를 설명하기 위해 사용된 녹색 곡선이 너무 복잡하다는 것

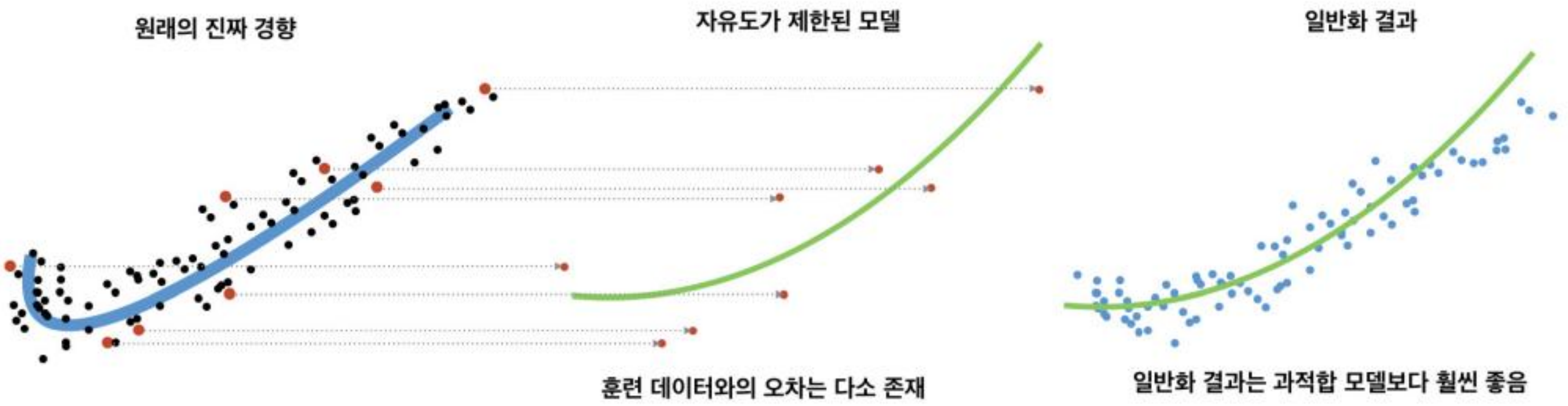
- 과소적합underfitting은 학습이 지나치게 덜 이루어져 새로운 데이터뿐만 아니라 학습 데이터조차 제대로 설명하지 못 하는 모델을 말한다



  - 모델이 지나치게 단순한 경우 발생한다
  - 예측을 제대로 할 수 없는 특징들만 제공되는 경우에 발생한다.
- 입력 데이터의 특징을 바꾸거나 학습 모델의 복잡도를 높이는 방법으로 문제를 해결

2.9 과적합을 피하기 위한 방법들

- 과적합의 근본적인 문제는 모델에 지나친 자유를 부여
- 모델이 복잡하면 불리하게 만드는 복잡성에 대한 규제regulation 혹은 정칙화regularization 기법을 이용



- 최적화의 대상인 오차 함수를 다음과 같이 정칙화가 적용된 새로운 함수로 바꾸어 보자.
    - 이때 추가되는 항 p를 벌칙항이라고 한다.

$$E^r(\mathbf{w}) = E(\mathbf{w}) + \boxed{p}$$

- 이제 학습은 오차를 줄이는 것뿐만 아니라 벌칙항도 같이 줄이는 일을 하게되고 이 벌칙항으로 어떤 것을 사용하는가에 따라 정칙화의 특성도 달라진다.

- 머신러닝에서 정칙화는 파라미터들이 지나치게 큰 값을 갖지 못하게 만드는 것
  - 정칙화를 파라미터 수축shrinkage이라고 부르기도 한다.
  - 대표적인 방법이 라소lasso 기법
  - 라소라는 명칭은 "least absolute shrinkage and selection operator"의 머릿글자를 딴 것
- l_1 노름을 사용하면 L1 정칙화
  - 이때 α는 정칙화를 조절하는 변수

$$E^r(\mathbf{w}) = E(\mathbf{w}) + \alpha\|\mathbf{w}\|_1$$

- l_2 노름을 사용하면 L2 정칙화
  - 능형 회귀ridge regression라고 부르는 방법과 같은 개념이다.

$$E^r(\mathbf{w}) = E(\mathbf{w}) + \alpha\|\mathbf{w}\|_2$$

- 머신러닝은 제한된 데이터를 통해 데이터가 대표하고 있는, 그러나 우리가 아직 본 적이 없는 미지의 데이터를 잘 설명할 수 있는 능력을 갖는 것

	데이터	모델	모델 규제
과적합	데이터의 규모 확대	단순화	강화
과소적합	새로운 특징 확보	복잡화	완화

핵심 정리

- 머신러닝을 위해서는 벡터와 행렬, 선형대수, 미적분 등의 지식이 필요하다.
- 두벡터를 곱할 때는 아다마르 곱, 점곱, 벡터곱 등을 사용할 수 있다.
- 행렬은 하나 이상의 행과 하나 이상의 열로 된 격자 구조에 수가 담겨 있다.
- 다차원 함수를 각각의 변수로 편미분하는 것은 각 차원의 기울기를 구하는 일이다.
- 모든 차원으로의 기울기를 구해, 벡터로 표현하면 기울기 벡터가 된다.
- 기울기 벡터의 반대 방향으로 변수를 이동하면 함수가 최소값이 되는 변수 쪽으로 접근할 수 있다.
- 기울기를 이용하여 최적의 변수를 찾아가는 방법을 경사 하강법이라고 한다.
- 함수가 울퉁불퉁한 모양일 경우 경사 하강법은 함수 전체에서 가장 작은 값을 갖는 전역 최소값이 아니라 일부 영역 내에서 가장 작은 값인 지역 최소값에 머물 수 있다.
- 머신러닝 알고리즘은 모델, 파라미터, 학습의 요소로 이루어진다.

핵심 정리

- 톰 미첼의 머신러닝 정의에 따라 모델, 파라미터, 학습을 설명하면, 모델은 작업을 수행하기 위해 설계된 것으로 파라미터에 의해 동작하도록 구현된다. 모델의 작업은 성능 척도에 의해 평가되며, 다양한 데이터 를 경험하면서 평가 점수를 개선하는 방향으로 파라미터를 변경하는 과정이 학습이다.
- 오차를 측정할 수 있으면 오차를 줄이는 파라미터 변경 방향을 계산할 수 있다.
- 기본적인 오차 최소화 방법은 파라미터로 결정되는 오차 곡면 함수의 기울기를 계산하여 이 기울기를 따 라 내려가면서 기울기가 0이 되는 곳을 찾는 것이다.
- 이렇게 오차가 최소가 되도록 가장 좋은 상태의 파라미를 찾아가는 동작을 최적화라고 하며, 학습은 곧 최적화이다. 파라미터를 오차 감소 방향으로 얼마나 옮겨 놓을 것인지, 이러한 학습 동작을 몇 번이나 해 야 하는지 등을 제어하는 값들을 하이퍼파라미터라고 한다.
- 파라미터는 모델의 동작을 결정하고, 하이퍼파라미터는 최적화 과정에 영향을 미친다.
- 모델이 과적합하지 않도록 하는 방법은 모델이 데이터에 지나치게 맞춰질 수 없도록 억제하는 것이다.