

기계학습과 수학

선형대수

벡터와 행렬

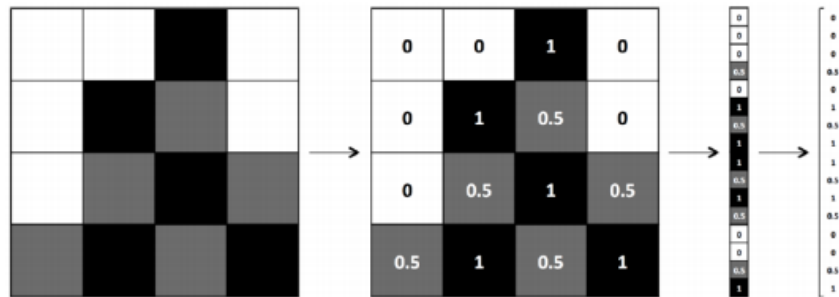
- 벡터

- 샘플을 특징 벡터로 표현

예) Iris 데이터에서 꽃받침의 길이, 꽃받침의 너비, 꽃잎의 길이, 꽃잎의 너비라는 4개의 특징이 각각 5.1, 3.5, 1.4, 0.2인 샘플

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}$$

예) 사진(image)을 벡터화하여 표현하는 과정



- 여러 개의 특징 벡터를 첨자로 구분 (\mathbf{x}_i)

$$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 4.7 \\ 3.2 \\ 1.3 \\ 0.2 \end{pmatrix}, \dots, \mathbf{x}_{150} = \begin{pmatrix} 5.9 \\ 3.0 \\ 5.1 \\ 1.8 \end{pmatrix}$$

- 종류와 크기 표현의 예) $\mathbf{x} \in \mathbb{R}^n$

- 행렬

- 여러 개의 벡터를 담음

요소: $x_{i,j}$, i 번째 행: $x_{i,:}$, j 번째 열: $x_{:,j}$

- 훈련집합을 담은 행렬 : 설계행렬

$$\mathbf{X} = \begin{pmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ \vdots & \vdots & \vdots & \vdots \\ 6.2 & 3.4 & 5.4 & 2.3 \\ 5.9 & 3.0 & 5.1 & 1.8 \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \\ \vdots & \vdots & \vdots & \vdots \\ x_{149,1} & x_{149,2} & x_{149,3} & x_{149,4} \\ x_{150,1} & x_{150,2} & x_{150,3} & x_{150,4} \end{pmatrix}$$

행(row) ←
↑ 열(column)

- 전치 행렬 : $A(i,j)$ 를 $A'(j,i)$ 로 바꾸는 것
- 행렬을 이용하면 방정식을 간결하게 표현 가능하다

예) 다항식의 행렬 표현

$$f(\mathbf{x}) = f(x_1, x_2, x_3)$$

$$= 2x_1x_1 - 4x_1x_2 + 3x_1x_3 + x_2x_1 + 2x_2x_2 + 6x_2x_3 - 2x_3x_1 + 3x_3x_2 + 2x_3x_3 + 2x_1 + 3x_2 - 4x_3 + 5$$

$$= (x_1 \ x_2 \ x_3) \begin{pmatrix} 2 & -4 & 3 \\ 1 & 2 & 6 \\ -2 & 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + (2 \ 3 \ -4) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + 5$$

$$= \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

- 특수한 행렬들

- 정사각행렬(정방행렬)
- 대각행렬
- 단위행렬
- 대칭행렬

$$\text{정사각행렬} \begin{pmatrix} 2 & 0 & 1 \\ 1 & 21 & 5 \\ 4 & 5 & 12 \end{pmatrix}, \quad \text{대각행렬} \begin{pmatrix} 50 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 8 \end{pmatrix},$$

$$\text{단위행렬} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{대칭행렬} \begin{pmatrix} 1 & 2 & 11 \\ 2 & 21 & 5 \\ 11 & 5 & 1 \end{pmatrix}$$

- 행렬 연산

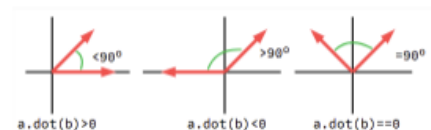
- 행렬 곱셈

$$\mathbf{C} = \mathbf{AB}, \text{ 이때 } c_{ij} = \sum_{k=1,s} a_{ik} b_{kj}$$

- 교환법칙 성립 X : $AB \neq BA$
- 분배법칙 / 결합법칙 성립 : $A(B+C) = AB + AC$, $A(BC) = (AB)C$
- 벡터의 내적(Inner Product)

$$\text{벡터의 내적 } \mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \sum_{k=1,d} a_k b_k \quad (2,2)$$

$$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix} \text{와 } \mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix} \text{의 내적 } \mathbf{x}_1 \cdot \mathbf{x}_2 \text{는 } 37.49$$



- $C = AB$ 예시

$$\begin{matrix} m & & & & \\ & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & p & & & \end{matrix} = \begin{matrix} & \blacksquare & \blacksquare & & \\ m & \blacksquare & \blacksquare & & \\ & \blacksquare & \blacksquare & & \\ & \blacksquare & \blacksquare & & \\ & n & & & \end{matrix} \bullet \begin{matrix} & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & p & & & \end{matrix}$$

Must match

$$= c_{ij} = \sum_k A_{i,k} B_{k,j}$$

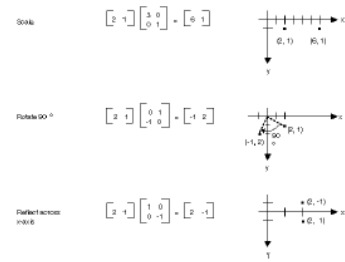
$$= A: m \times n, B: n \times p, C: m \times p$$

- 행렬 곱셈을 통한 벡터의 변환(Function / Mapping) 예시

$$\begin{bmatrix} 4 & -3 & 1 & 3 \\ 2 & 0 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \end{bmatrix} \text{ and } \begin{bmatrix} 4 & -3 & 1 & 3 \\ 2 & 0 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Diagram illustrating matrix multiplication by A from \mathbb{R}^d to \mathbb{R}^2 . The left side shows vectors x and u in \mathbb{R}^d being multiplied by A to produce vectors b and 0 in \mathbb{R}^2 .

대표적 변환 예)



○ 텐서

■ 3차원 이상의 구조를 가진 숫자 배열

- 0차 : 수(Scalar)
- 1차 : 벡터
- 2차 : 행렬
- 고차원 ...

예) 3차원 구조의 RGB 컬러 영상

$$\mathbf{A} = \begin{pmatrix} \begin{pmatrix} 3 & 0 & 1 & 2 & 6 & 7 \\ 3 & 1 & 2 & 3 & 5 & 6 \\ 1 & 2 & 2 & 2 & 2 & 3 \\ 3 & 0 & 0 & 1 & 1 & 0 \\ 5 & 4 & 1 & 3 & 3 & 3 \\ 2 & 2 & 1 & 2 & 2 & 1 \end{pmatrix} & \begin{pmatrix} 6 \\ 3 \\ 0 \\ 3 \\ 1 \end{pmatrix} \end{pmatrix}$$

놈과 유사도

- 벡터와 행렬의 크기를 놈(Norm)으로 측정

- 벡터의 p차 놈

$$p\text{-차 놈: } \|\mathbf{x}\|_p = \left(\sum_{i=1,d} |x_i|^p \right)^{\frac{1}{p}} \quad (2.3)$$

1차 ($p = 1$) 놈, 2차 ($p = 2$) 놈 Euclidean norm, 최대 ($p = \infty$) 놈 max norm

$$\text{최대 놈: } \|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \dots, |x_d|) \quad (2.4)$$

예) $\mathbf{x} = (3 \ -4 \ 1)$ 일 때, 2차 놈은 $\|\mathbf{x}\|_2 = (3^2 + (-4)^2 + 1^2)^{1/2} = 5.099$

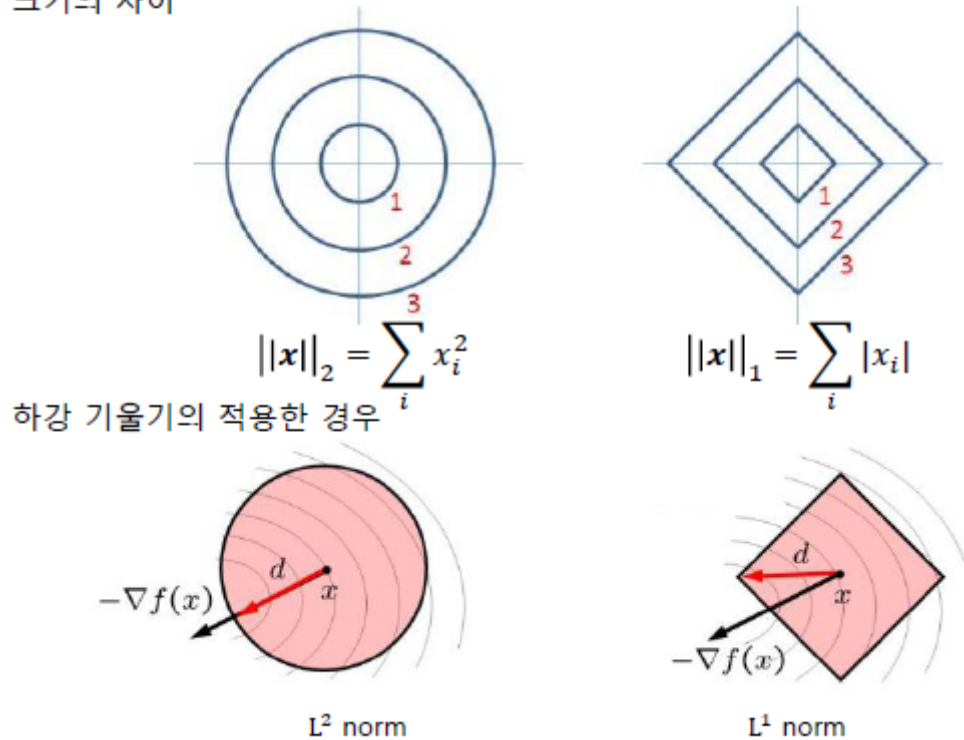
- 행렬의 프robe니우스(Frobenius Norm) : 행렬의 크기를 측정

$$\text{프로베니우스 놈: } \|A\|_F = \left(\sum_{i=1,n} \sum_{j=1,m} a_{ij}^2 \right)^{\frac{1}{2}} \quad (2.6)$$

$$\text{예를 들어, } \left\| \begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix} \right\|_F = \sqrt{2^2 + 1^2 + 6^2 + 4^2} = 7.550$$

- 1차 놈(Manhattan Distance, L1)과 2차 놈(Euclidean Distance, L2) 비교

크기의 차이



- 유사도(Similarity)와 거리(Distance)
 - 벡터를 기하학적으로 해석

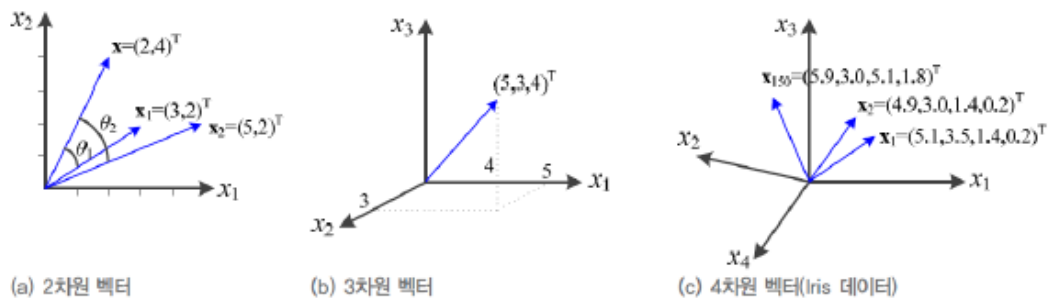


그림 2-2 벡터를 기하학적으로 해석

- 코사인 유사도(Cosine Similarity)

$$\text{cosine_similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} = \cos(\theta) \quad (2.7)$$

퍼셉트론의 해석

- 퍼셉트론 : 1958년 로젠블라트가 고안한 분류기(Classifier) 모델

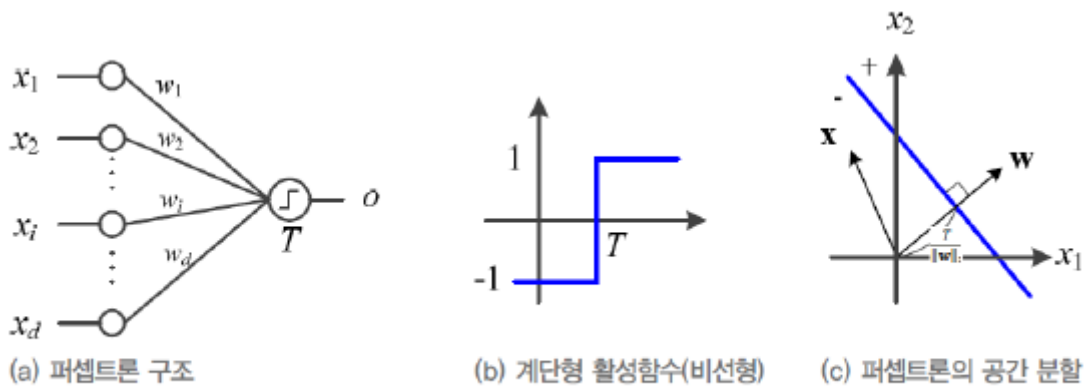


그림 2-3 퍼셉트론의 구조와 동작

- 그림 2-3의 c의 파란 직선은 두 개의 부분공간을 나누는 결정직선(Decision Line)

w에 수직이고 원점으로부터 $\frac{T}{\|w\|_2}$ 만큼 떨어져 있음

- 동작을 수식으로 표현하면

$$o = \tau(\mathbf{w} \cdot \mathbf{x}), \quad \text{이때} \quad \tau(a) = \begin{cases} 1, & a \geq T \\ -1, & a < T \end{cases} \quad (2.8)$$

- 활성화 함수(Activation Function)으로는 계단함수(Step Function) 사용
- 3차원 특징공간은 결정평면(Decision Plane), 4차원 이상은 결정 초평면(Decision Hyperplane)
- 3차원 특징공간을 위한 퍼셉트론

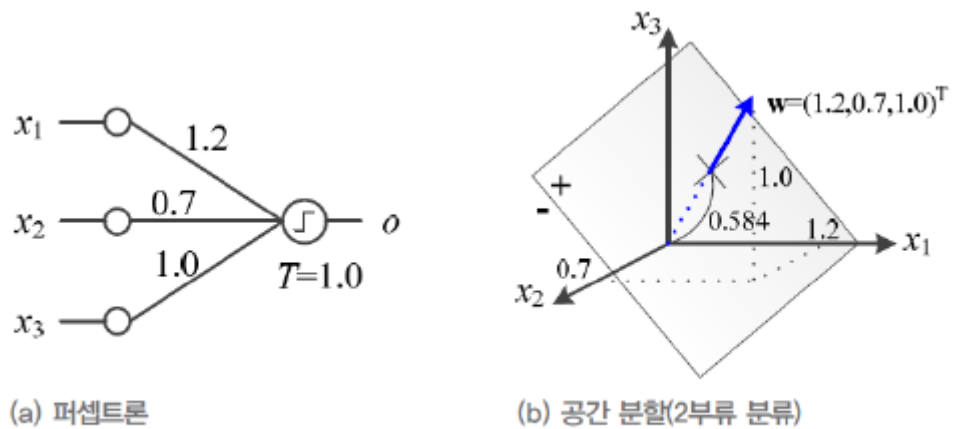
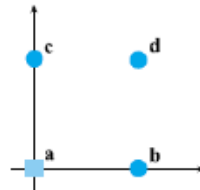


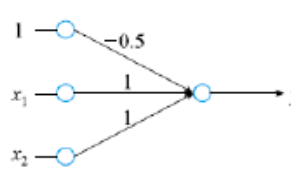
그림 2-4 퍼셉트론의 예(3차원)

- 예제

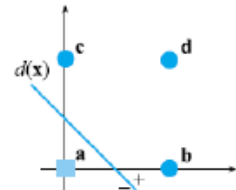
$$\begin{aligned} \mathbf{a} &= (0,0)^T, \quad t_a = -1 \\ \mathbf{b} &= (1,0)^T, \quad t_b = 1 \\ \mathbf{c} &= (0,1)^T, \quad t_c = 1 \\ \mathbf{d} &= (1,1)^T, \quad t_d = 1 \end{aligned}$$



(a) OR 분류 문제



(b) OR 분류기로서 퍼셉트론

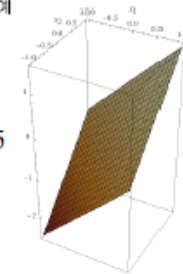


(c) 퍼셉트론은 선형 분류기

그림 4.3 퍼셉트론의 예

이 퍼셉트론은 $\mathbf{w} = (1,1)^T$, $b = -0.5$

따라서 결정 직선은 $d(\mathbf{x}) = x_1 + x_2 - 0.5$

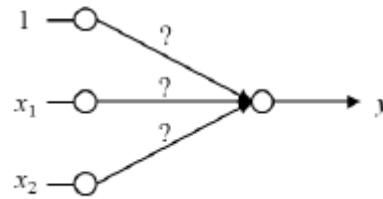
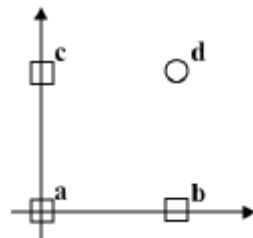


▪ 샘플 a를 맞추나 보자? $y = \tau(\mathbf{w}^T \mathbf{c} + b) = \tau((1,1) \begin{pmatrix} 0 \\ 1 \end{pmatrix} - 0.5) = \tau(0.5) = 1$

▪ 나머지 샘플 b, c, d도 맞추는가?

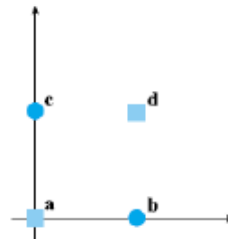
○ AND 분류 문제는?

$$\begin{array}{cccc} \mathbf{a}=(0,0)^T & \mathbf{b}=(1,0)^T & \mathbf{c}=(0,1)^T & \mathbf{d}=(1,1)^T \\ t_a=-1 & t_b=-1 & t_c=1 & t_d=1 \end{array}$$

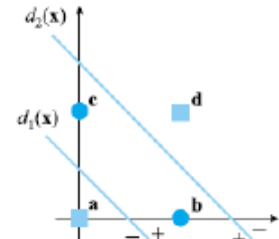


○ XOR 분류 문제는?

- 퍼셉트론은 75% 분류 한계
 - 이 한계를 어떻게 극복?
- 두 개의 퍼셉트론 (결정 직선) 사용



(a) XOR 분류 문제



(b) 두 개의 직선으로 해결

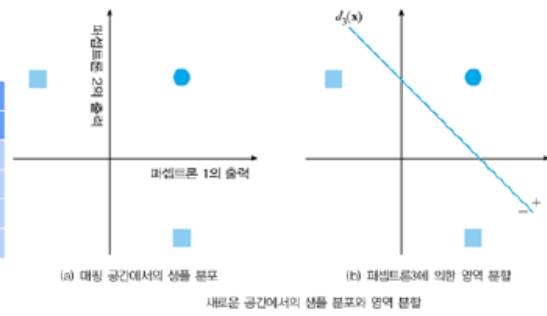
○ 다층 퍼셉트론(Multi-Layer Perceptron)

- 두 단계에 걸쳐 문제해결
 - 단계 1 : 원래 특징 공간을 새로운 공간으로 매핑
 - 단계 2 : 새로운 공간에서 분류
- XOR의 경우, 다음과 같은 조건을 활용

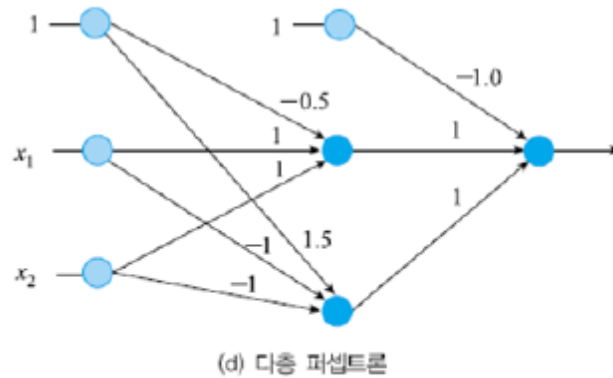
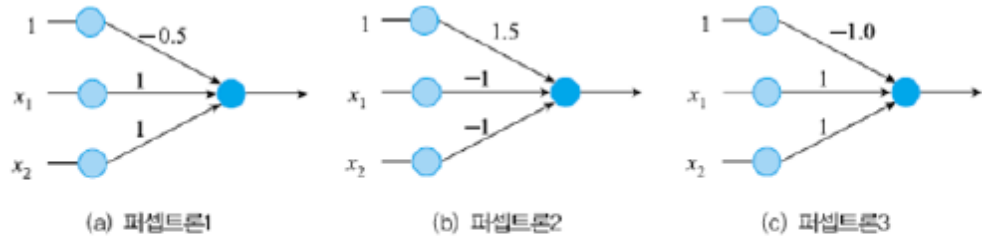
$$\left. \begin{array}{l} \mathbf{w}_1^T \mathbf{x} + b_1 > 0 \text{ 이고 } \mathbf{w}_2^T \mathbf{x} + b_2 > 0 \text{ 이면, } \mathbf{x} \in \omega_1 \\ \mathbf{w}_1^T \mathbf{x} + b_1 < 0 \text{ 이거나 } \mathbf{w}_2^T \mathbf{x} + b_2 < 0 \text{ 이면, } \mathbf{x} \in \omega_2 \end{array} \right\}$$

두 단계로 XOR 문제 해결

샘플	특징 벡터 (x)		첫 번째 단계		두 번째 단계
	x_1	x_2	퍼셉트론1	퍼셉트론2	퍼셉트론3
a	0	0	-1	+1	-1
b	1	0	+1	+1	+1
c	0	1	+1	+1	+1
d	1	1	-1	-1	-1

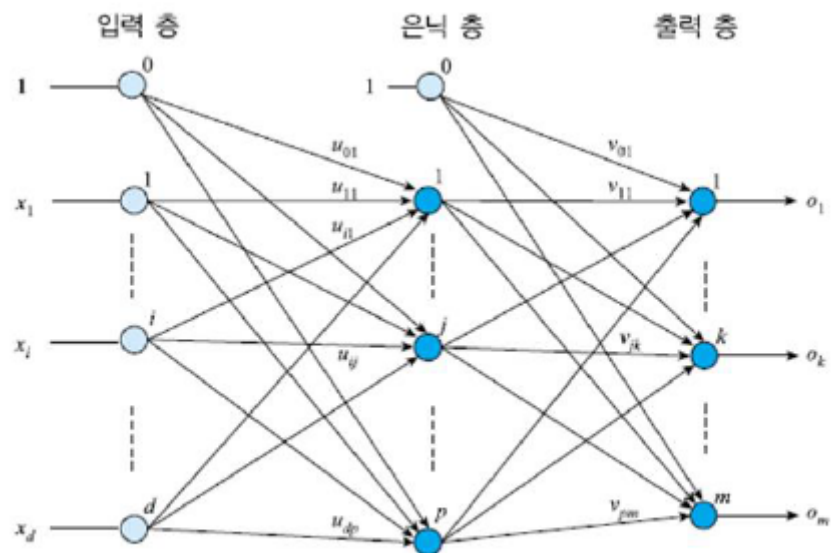


■ 다층 퍼셉트론을 활용한 XOR 분류 문제



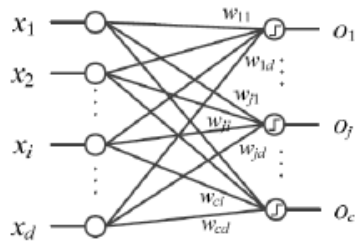
세 개의 퍼셉트론과 이들을 연결하여 만든 다층 퍼셉트론

■ 다층 퍼셉트론의 구조(입력층, 은닉층, 출력층)



다층 퍼셉트론의 구조와 표기

- 출력이 여러 개인 퍼셉트론의 표현



출력은 벡터 $\mathbf{o} = (o_1, o_2, \dots, o_c)^T$ 로 표기

j 번째 퍼셉트론의 가중치 벡터를

$\mathbf{w}_j = (w_{j1}, w_{j2}, \dots, w_{jd})^T$ 와 같이 표기

그림 2-5 출력이 여러 개인 퍼셉트론

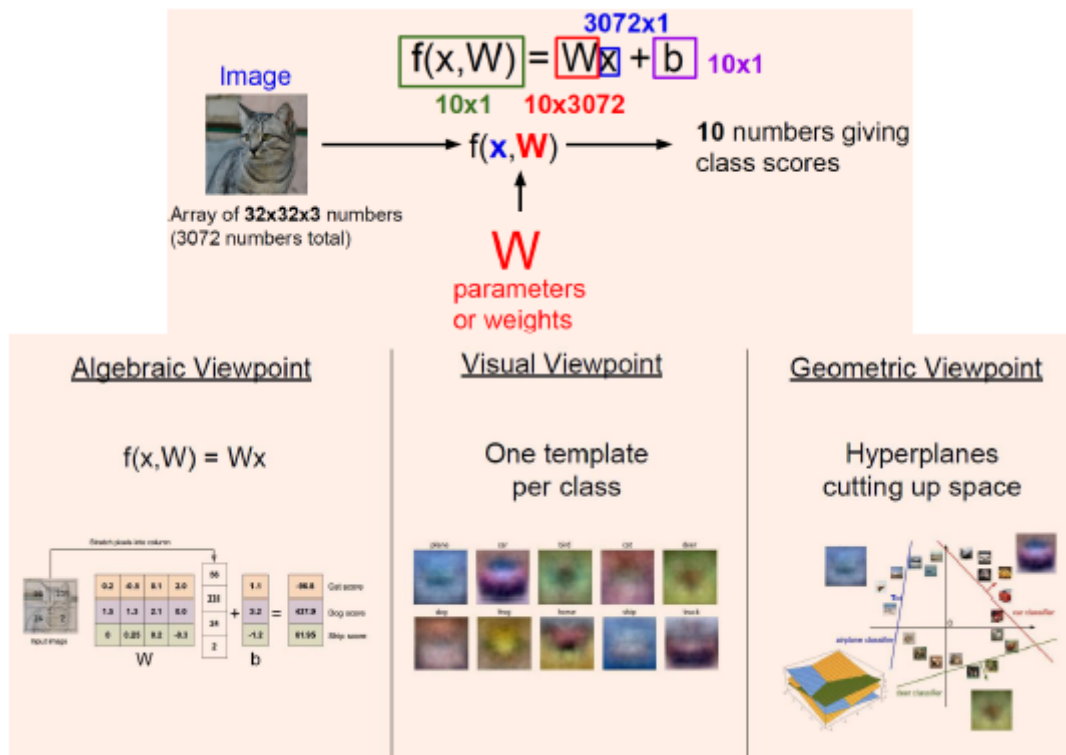
동작을 수식으로 표현하면,

$$\mathbf{o} = \tau \begin{pmatrix} \mathbf{w}_1 \cdot \mathbf{x} \\ \mathbf{w}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{w}_c \cdot \mathbf{x} \end{pmatrix} \rightarrow \text{행렬로 간결하게 쓰면 } \mathbf{o} = \tau(\mathbf{W}\mathbf{x})$$

이때 $\mathbf{W} = \begin{pmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_c^T \end{pmatrix}$

가중치 벡터를 각 부류의 기준 벡터로 간주하면, c 개 부류의 유사도를 계산하는 셈

- 선형 분류기 이해



- 학습의 정의

- 추론(Inferring):

식(2.10)은 학습을 마친 알고리즘을 현장의 새로운 데이터에 적용했을 때 일어나는 과정

분류라는 과업: $\hat{\mathbf{o}} = \tau(\tilde{\mathbf{W}} \tilde{\mathbf{x}})$ (2.10)

- 학습(Learning): 훈련집합의 샘플에 대해 식(2.11)을 가장 잘 만족하는 \mathbf{W} 를 찾아내는 과정

학습이라는 과업: $\tilde{\mathbf{o}} = \tau(\tilde{\mathbf{W}} \tilde{\mathbf{x}})$ (2.11)

- 현대 기계 학습에서 퍼셉트론의 중요성

- 딥러닝은 퍼셉트론을 여러 층으로 확장하여 만들

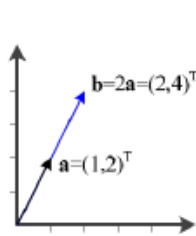
선형결합과 벡터공간

- 벡터 : 공간상의 한 점으로 화살표 끝이 벡터의 좌표에 해당
- 선형결합이 만드는 벡터공간
 - 기저(basis)벡터 a와 b의 선형 결합(Linear Combination)

$$\mathbf{c} = \alpha_1 \mathbf{a} + \alpha_2 \mathbf{b}$$

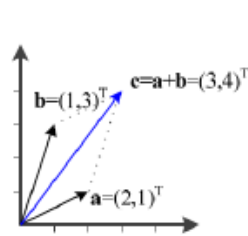
(2,12)

- 선형결합으로 만들어지는 공간을 벡터공간(Vector Space)이라 부름

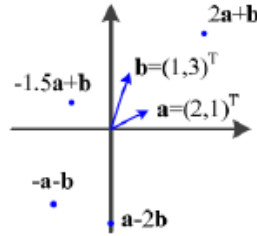


(a) 벡터에 스칼라 곱

그림 2-6 벡터의 연산

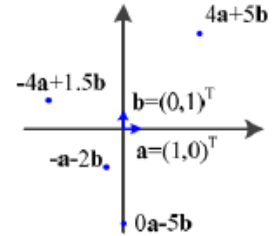


(b) 두 벡터의 덧셈



(a) 기저 벡터와 벡터공간

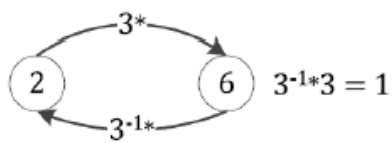
그림 2-7 벡터공간



(b) 정규직교 기저 벡터

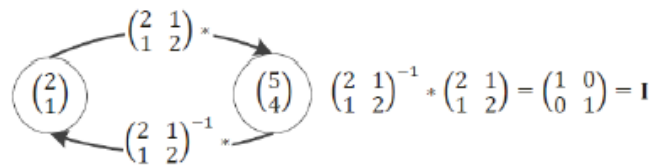
역행렬

- 원리



(a) 역수의 원리

그림 2-9 역행렬



(b) 역행렬의 원리

- 정사각행렬 A의 역행렬 A^{-1}

$$A^{-1}A = AA^{-1} = I$$

- 예를 들어, $\begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix}$ 의 역행렬은 $\begin{pmatrix} 2 & -0.5 \\ -3 & 1 \end{pmatrix}$

- 역행렬을 활용한 방정식 표현과 해

- 방정식: $\mathbf{Ax} = \mathbf{b}$ 의 확장
 - $\mathbf{A} \in \mathbb{R}^{m \times n}$: 알고 있는 행렬
 - $\mathbf{b} \in \mathbb{R}^m$: 알고 있는 벡터
 - $\mathbf{x} \in \mathbb{R}^n$: 알고 싶은 알지 못한 벡터

$$A_{1,:}\mathbf{x} = A_{1,1}x_1 + A_{1,2}x_2 + \cdots + A_{1,n}x_n = b_1$$

$$A_{2,:}\mathbf{x} = A_{2,1}x_1 + A_{2,2}x_2 + \cdots + A_{2,n}x_n = b_2$$

...

$$A_{m,:}\mathbf{x} = A_{m,1}x_1 + A_{m,2}x_2 + \cdots + A_{m,n}x_n = b_m$$

- 선형 방정식의 경우
 - 불능: 해 없음
 - 부정: 다수의 해 존재
 - 유일해 존재 \rightarrow 역행렬을 이용하여 해를 구함 \rightarrow

$$\begin{aligned}\mathbf{Ax} &= \mathbf{b} \\ \mathbf{A}^{-1}\mathbf{Ax} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{I}_n\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b}\end{aligned}$$

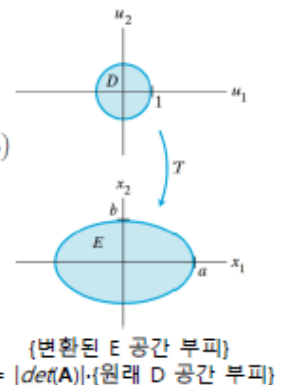
- 정리

정리 2-1 다음 성질은 서로 필요충분조건이다.

- \mathbf{A} 는 역행렬을 가진다. 즉, 특이행렬이 아니다.
- \mathbf{A} 는 최대계수를 가진다.
- \mathbf{A} 의 모든 행이 선형독립이다.
- \mathbf{A} 의 모든 열이 선형독립이다.
- \mathbf{A} 의 행렬식은 0이 아니다.
- $\mathbf{A}^T\mathbf{A}$ 는 양의 정부호(positive definite) 대칭 행렬이다.
- \mathbf{A} 의 고윳값은 모두 0이 아니다.

- 행렬식(Determinant)

$$\left. \begin{aligned} \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} &= ad - bc \\ \det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} &= aei + bfg + cdh - ceg - bdi - afh \\ \text{예를 들어 } \begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix} \text{의 행렬식은 } 2 \cdot 4 - 1 \cdot 6 &= 2 \end{aligned} \right\} (2.15)$$



- 기하학적 의미

- 행렬식의 절대값은 주어진 행렬의 곱에 의한 공간의 확장 또는 축소로 볼 수 있음
 - If $\det(\mathbf{A}) = 0$, 하나의 차원을 따라 축소되어 공간의 부피를 잃게 됨

- If $\det(A) = 1$, 공간의 부피를 유지한 변환
- 차원에서의 기하학적인 예시
 - 2차원에서는 2개의 행 벡터가 이루는 평행사변형 넓이
 - 3차원에서는 3개의 행 벡터가 이루는 평행사각기둥의 부피

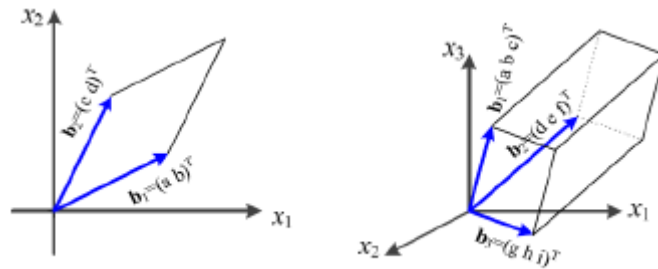


그림 2-10 행렬식의 기하학적 해석

• 정부호(Definiteness) 행렬

양의 정부호 행렬: $\mathbf{0}$ 이 아닌 모든 벡터 \mathbf{x} 에 대해, $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$

■ 예를 들어, $\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ 는 $(x_1 \ x_2) \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1^2 + 2x_2^2$ 이므로

$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ 는 양의 정부호 행렬

○ 종류

양의 준정부호 positive semi-definite 행렬: $\mathbf{0}$ 이 아닌 모든 벡터 \mathbf{x} 에 대해, $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$

음의 정부호 negative definite 행렬: $\mathbf{0}$ 이 아닌 모든 벡터 \mathbf{x} 에 대해, $\mathbf{x}^T \mathbf{A} \mathbf{x} < 0$

음의 준정부호 negative semi-definite 행렬: $\mathbf{0}$ 이 아닌 모든 벡터 \mathbf{x} 에 대해, $\mathbf{x}^T \mathbf{A} \mathbf{x} \leq 0$

행렬 분해

• 분해(Decomposition)란?

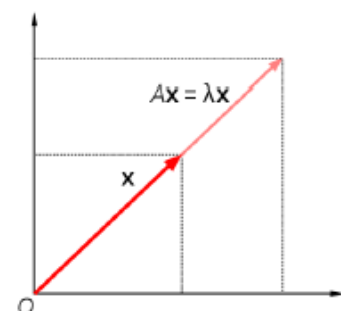
- 정수 3717은 특성이 보이지 않지만, $3 \times 3 \times 7 \times 59$ 로 소인수분해하면 특성이 보이듯 분해하는 것

• 고유값(Eigenvalue)과 고유 벡터(Eigenvector)

고유 벡터 \mathbf{v} 와 고유값 λ

$$\mathbf{A} \mathbf{v} = \lambda \mathbf{v} \quad (2, 20)$$

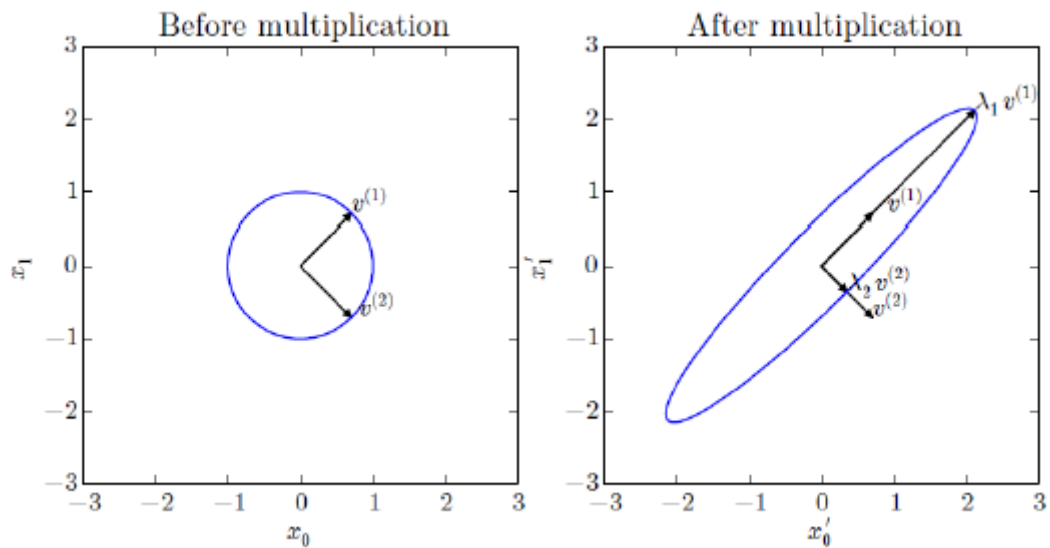
- 2차원 공간에서의 고유값과 고유 벡터의 기하학적 해석



예를 들어, $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ 이고 $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = 1 \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ 이므로,

$$\lambda_1 = 3, \lambda_2 = 1 \text{ 이고 } \mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

• 고유값의 효과



- 왼쪽: 원으로 표현된 단위 벡터 $\mathbf{u} \in \mathbb{R}^2$ 의 모든 점
- 오른쪽: 행렬 \mathbf{A} 의 곱에 의한 $\mathbf{A}\mathbf{u}$ 모든 점, \mathbf{A} 는 원을 고유 벡터 방향으로 고윳값만큼 크기 변환만 시킴
- 고윳값과 고유 벡터의 기하학적 해석

예제 2-5

[그림 2-12]의 반지름이 1인 원 위에 있는 4개의 벡터 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ 가 $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ 에 의해 어떻게 변환되는지 살펴보자. 변환 후의 벡터를 각각 $\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3, \mathbf{x}'_4$ 로 표기한다.

$$\mathbf{x}'_1 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \end{pmatrix}$$

$$\mathbf{x}'_2 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\mathbf{x}'_3 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$

$$\mathbf{x}'_4 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$$

눈 여겨 볼 점은 \mathbf{A} 의 고유 벡터 $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ 과 방향이 같은 \mathbf{x}_1 과 \mathbf{x}_3 이다. 이들은 변환 때문에 길이가 달라지더라도 방향은 그대로 유지한다. 식 (2.20)을 충실히 따르고 있다. 이때 길이의 변화는 고윳값 λ 에 따른다. 즉, \mathbf{x}_1 은 3배 만큼, \mathbf{x}_3 은 1배만큼 길이가 변한다. 나머지 \mathbf{x}_2 와 \mathbf{x}_4 는 길이와 방향이 모두 변한다. 파란 원 위에 있는 모든 점을 변환하면 빨간색의 타원이 된다. 파란 원 위에 존재하는 무수히 많은 점(벡터) 중에 방향이 바뀌지 않는 것은 고유 벡터에 해당하는 \mathbf{x}_1 과 \mathbf{x}_3 뿐이다.

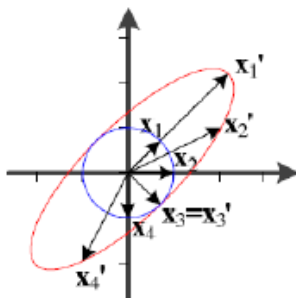


그림 2-12 고유 벡터의 공간 변환

- 고유 분해(Eigen-decomposition)

$$A = Q\Lambda Q^{-1} \quad (2.21)$$

- Q 는 A 의 고유 벡터를 열에 배치한 행렬이고 Λ 는 고유값을 대각선에 배치한 대각행렬

$$\begin{aligned} A &= Q\Lambda Q^T \\ &= \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1 v_1 & \lambda_2 v_2 & \cdots & \lambda_n v_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \\ &= \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_n v_n v_n^T \end{aligned}$$

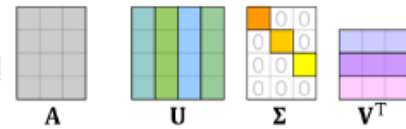
- 예를 들어, $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}$
- 고유 분해는 고유값과 해당 고유 벡터가 존재하는 정사각행렬에만 적용 가능
- 하지만, 기계 학습에서는 정사각행렬이 아닌 경우의 분해도 필요하므로

고유 분해는 한계를 가짐

- $n \times m$ 행렬 A 의 특잇값 분해(Singular Value Decomposition)

$$A = U\Sigma V^T \quad (2.22)$$

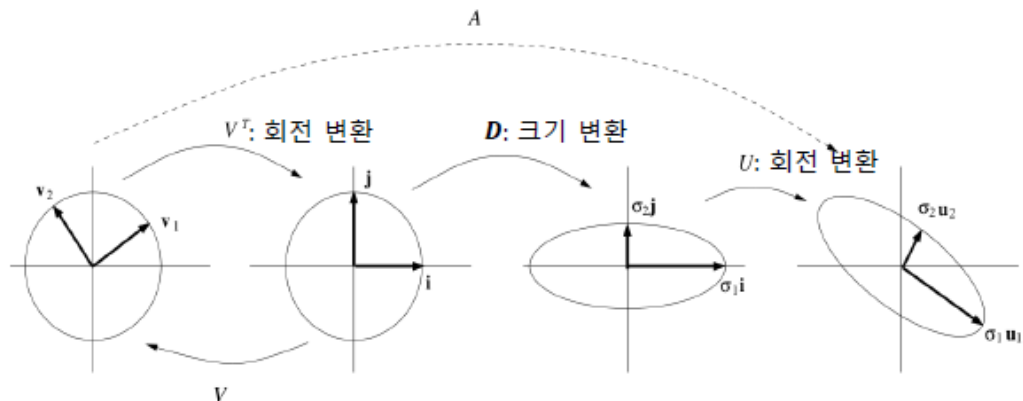
- 왼쪽 특이행렬 U 는 AA^T 의 고유 벡터를 열에 배치한 $n \times n$ 행렬
- 오른쪽 특이행렬 V 는 $A^T A$ 의 고유 벡터를 열에 배치한 $m \times m$ 행렬
- Σ 는 AA^T 의 고유값의 제곱근을 대각선에 배치한 $n \times m$ 대각행렬



예를 들어, A 를 4×3 행렬이라고 했을 때 다음과 같이 특잇값 분해가 된다.

$$\begin{aligned} A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 3 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix} &= \begin{pmatrix} -0.1914 & -0.2412 & 0.1195 & -0.9439 \\ -0.5144 & 0.6990 & -0.4781 & -0.1348 \\ -0.6946 & -0.6226 & -0.2390 & 0.2697 \\ -0.4651 & 0.2560 & 0.8367 & 0.1348 \end{pmatrix} \\ &\begin{pmatrix} 3.7837 & 0 & 0 \\ 0 & 2.7719 & 0 \\ 0 & 0 & 1.4142 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -0.7242 & -0.4555 & -0.5177 \\ -0.6685 & 0.2797 & 0.6891 \\ 0.1690 & -0.8452 & 0.5071 \end{pmatrix} \end{aligned}$$

- 기하학적 해석



- 정사각행렬이 아닌 행렬의 역행렬을 구하는데 사용됨

$$A = U \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_s \\ & & & 0 \end{pmatrix} V^T \xrightarrow{\text{Pseudoinverse}} A^+ = V \begin{pmatrix} 1/\sigma_1 & & \\ & \ddots & \\ & & 1/\sigma_s & 0 \end{pmatrix} U^T$$

확률과 통계

확률 기초

- 확률변수 (Random value)

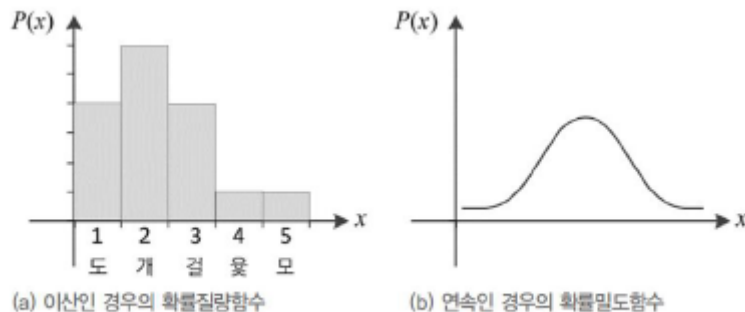


그림 2-13 윗을 던졌을 때 나올 수 있는 다섯 가지 경우(왼쪽부터 도, 개, 걸, 윗, 모)

- 다섯 가지 경우 중 한 값을 갖는 확률변수 x
- x 의 정의역: { 도, 개, 걸, 윗, 모 }
- 확률분포 (probability distribution)
 - 확률질량함수 (probability mass function): 이산 확률 변수

$$P(x = \text{도}) = \frac{4}{16}, P(x = \text{개}) = \frac{6}{16}, P(x = \text{걸}) = \frac{4}{16}, P(x = \text{윗}) = \frac{1}{16}, P(x = \text{모}) = \frac{1}{16}$$

- 확률밀도함수 (probability density function): 연속 확률 변수



(a) 이산인 경우의 확률질량함수

(b) 연속인 경우의 확률밀도함수

그림 2-14 확률분포

- 확률벡터 (random vector)
 - Iris에서 x 는 4차원 확률 벡터

$$\mathbf{x} = (x_1, x_2, x_3, x_4)^T = (\text{꽃받침 길이}, \text{꽃받침 너비}, \text{꽃잎 길이}, \text{꽃잎 너비})^T$$

- 간단한 확률실험 장치
 - 주머니에서 번호표를 뽑은 다음, 번호에 따라 해당 병에서 공을 뽑고 색을 관찰함
 - 번호를 y , 공의 색을 x 라는 확률변수로 표현하면

정의역은 $y \in \{①, ②, ③\}$, $x \in \{\text{파랑, 하양}\}$

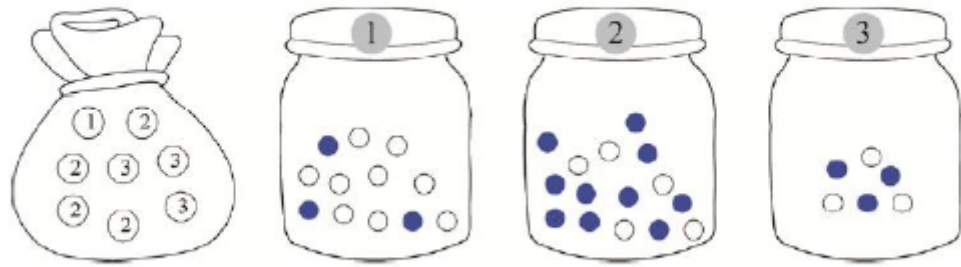



그림 2-15 확률 실험

- 곱 규칙 (product rule) 과 합 규칙 (sum rule)

 1568701322664

- 조건부 확률 (conditional probability)에 의한 결합확률 계산

$$\text{곱 규칙: } P(y, x) = P(x|y)P(y) \quad (2.23)$$

- 하얀 공이 뽑힐 확률

$$\begin{aligned} P(\text{하양}) &= P(\text{하양}|①)P(①) + P(\text{하양}|②)P(②) + P(\text{하양}|③)P(③) \\ &= \frac{9}{12} \frac{1}{8} + \frac{5}{15} \frac{4}{8} + \frac{3}{6} \frac{3}{8} = \frac{43}{96} \end{aligned}$$

- 합 규칙과 곱 규칙에 의한 주변확률 (marginal probability) 계산

$$\text{합 규칙: } P(x) = \sum_y P(y, x) = \sum_y P(x|y)P(y) \quad (2.24)$$

- 조건부 확률

$$P(y = y \mid x = x) = \frac{P(y = y, x = x)}{P(x = x)}$$

- 확률의 연쇄 법칙 (chain rule)

$$P(x^{(1)}, \dots, x^{(n)}) = P(x^{(1)}) \prod_{i=2}^n P(x^{(i)} \mid x^{(1)}, \dots, x^{(i-1)})$$

- 독립 (independence)

$$\forall x \in \mathbf{x}, y \in \mathbf{y}, p(x = x, y = y) = p(x = x)p(y = y)$$

- 조건부 독립 (conditional independence)

$$\forall x \in \mathbf{x}, y \in \mathbf{y}, z \in \mathbf{z}, p(x = x, y = y \mid z = z) = p(x = x \mid z = z)p(y = y \mid z = z)$$

- 기대값 (expectation)

$$\mathbb{E}_{x \sim P}[f(x)] = \sum_x P(x)f(x) \quad \rightarrow \quad \text{linearity of expectations:}$$

$$\mathbb{E}_x[\alpha f(x) + \beta g(x)] = \alpha \mathbb{E}_x[f(x)] + \beta \mathbb{E}_x[g(x)]$$

베이즈 정리와 기계 학습

- 베이즈 정리 (Bayes's rule)

$$P(y, x) = P(x|y)P(y) = P(x, y) = P(y|x)P(x)$$

$$\longrightarrow P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (2.26)$$

- 다음 질문을 식으로 쓰면

“하얀 공이 나왔다는 사실만 알고 어느 병에서 나왔는지 모르는데, 어느 병인지 추정하라.”

$$\hat{y} = \operatorname{argmax}_y P(y|x) \quad (2.27)$$

베이즈 정리를 적용하면, $\hat{y} = \operatorname{argmax}_y P(y|x = \text{하양}) = \operatorname{argmax}_y \frac{P(x = \text{하양}|y)P(y)}{P(x = \text{하양})}$

세 가지 경우에 대해 확률을 계산하면,

$$P(\text{①}|\text{하양}) = \frac{P(\text{하양}|\text{①})P(\text{①})}{P(\text{하양})} = \frac{\frac{9}{128} \cdot \frac{1}{43}}{\frac{43}{96}} = \frac{9}{43}$$

$$P(\text{②}|\text{하양}) = \frac{P(\text{하양}|\text{②})P(\text{②})}{P(\text{하양})} = \frac{\frac{5}{158} \cdot \frac{4}{43}}{\frac{43}{96}} = \frac{16}{43} \longrightarrow \text{③번 병일 확률이 가장 높음}$$

$$P(\text{③}|\text{하양}) = \frac{P(\text{하양}|\text{③})P(\text{③})}{P(\text{하양})} = \frac{\frac{3}{68} \cdot \frac{3}{43}}{\frac{43}{96}} = \frac{18}{43}$$

- 해석 : 사후 (posteriori) 확률 = 우도 (likelihood) 확률 * 사전 (prior) 확률

$$\overbrace{P(y|x)}^{\text{사후확률}} = \frac{\overbrace{P(x|y)}^{\text{우도}} \overbrace{P(y)}^{\text{사전확률}}}{P(x)}$$

- 기계학습에 적용

- 예시) Iris 데이터 분류 문제

- 특징 벡터 \mathbf{x} , 부류 $y \in \{\text{setosa}, \text{versicolor}, \text{virginica}\}$
- 분류 문제를 argmax 로 표현하면 식 (2.29)

$$\hat{y} = \operatorname{argmax}_y P(y|\mathbf{x}) \quad (2.29)$$



그림 2-16 붓꽃의 부류 예측 과정

$$\text{특징추출} \rightarrow \mathbf{x} = (7.0, 3.2, 4.7, 1.4)^T \xrightarrow{\text{사후확률 추정}} \begin{matrix} P(\text{setosa}|\mathbf{x}) = 0.18 \\ P(\text{versicolor}|\mathbf{x}) = 0.72 \\ P(\text{virginica}|\mathbf{x}) = 0.10 \end{matrix} \xrightarrow{\text{argmax}} \text{versicolor}$$

- 사후확률 $P(y|x)$ 를 직접 추정하는 일은 아주 단순한 경우를 빼고 불가능
- 따라서 베이즈 정리를 이용하여 추정
 - 사전확률

$$\text{사전확률: } P(y = c_i) = \frac{n_i}{n} \quad (2.30)$$

- 우도확률 : 밀도추정기법으로 추정

최대 우도

- 매개변수 (모수) θ 를 모르는 상황에서 매개변수를 추정하는 문제

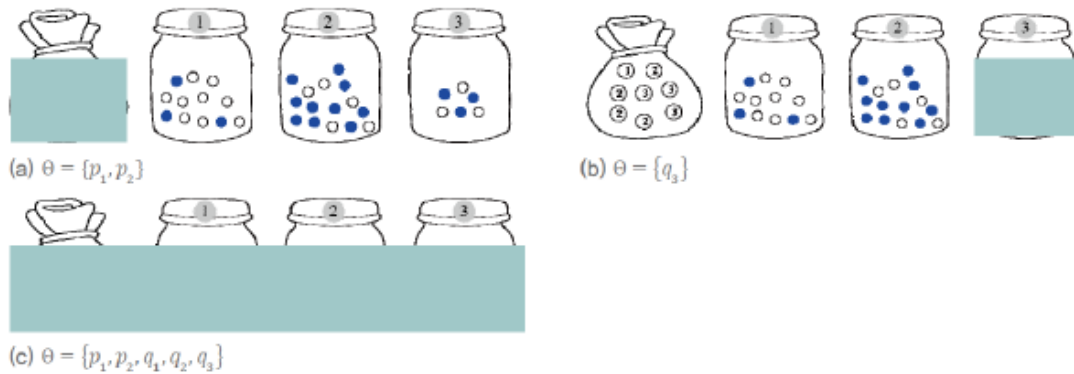


그림 2-17 매개변수가 감추어진 여러 가지 상황

관측된 데이터집합 $\mathbb{X} = \{\bullet \circ \circ \bullet \circ \bullet \circ \circ \bullet \circ \circ\}$ 라 할 때,

“데이터 \mathbb{X} 가 주어졌을 때, \mathbb{X} 를 발생시켰을 가능성을 최대로 하는 매개변수 $\theta = \{q_3\}$ 의 값을 찾아라.”

- 최대 우도법 (maximum likelihood)
 - 어떤 확률변수의 관찰된 값들을 토대로 그 확률변수의 매개변수를 구하는 방법
 - [그림 2-17(b)] 문제를 수식으로 쓰면,

$$\hat{q}_3 = \operatorname{argmax}_{q_3} P(\mathbb{X}|q_3) \quad (2.31)$$

- 일반화 하면,

$$\text{최대 우도 추정: } \hat{\theta} = \operatorname{argmax}_{\theta} P(\mathbb{X}|\theta) \quad (2.32)$$

- 수치 문제를 피하기 위해 로그 표현으로 바꾸면,

$$\text{최대 로그우도 추정: } \hat{\theta} = \operatorname{argmax}_{\theta} \log P(\mathbb{X}|\theta) = \operatorname{argmax}_{\theta} \sum_{i=1}^n \log P(x_i|\theta) \quad (2.34)$$

- 단조 증가하는 로그 함수를 이용하여 계산 단순화

평균과 분산

- 데이터의 요약 정보로서 평균과 분산

$$\left. \begin{aligned} \text{평균 } \mu &= \frac{1}{n} \sum_{i=1}^n x_i \\ \text{분산 } \sigma^2 &= \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \end{aligned} \right\} \quad (2.36)$$

$$\operatorname{Var}(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$$

- 평균 벡터(치우침 정도)와 공분산 행렬 (covariance matrix) (확률변수의 상관정도)

$$\mu = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad (2.37)$$

$$\Sigma = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \quad (2.39)$$

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \dots & \sigma_{dd} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \dots & \sigma_d^2 \end{pmatrix}$$

Covariance matrix:

$$\operatorname{Cov}(\mathbf{x})_{ij} = \operatorname{Cov}(x_i, x_j)$$

- 평균 벡터와 공분산 행렬 예제

예제 2-7

Iris 데이터베이스의 샘플 중 8개만 가지고 공분산 행렬을 계산하자.

$$\mathbb{X} = \{\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 4.7 \\ 3.2 \\ 1.3 \\ 0.2 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 4.6 \\ 3.1 \\ 1.5 \\ 0.2 \end{pmatrix}, \mathbf{x}_5 = \begin{pmatrix} 5.0 \\ 3.6 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_6 = \begin{pmatrix} 5.4 \\ 3.9 \\ 1.7 \\ 0.4 \end{pmatrix}, \mathbf{x}_7 = \begin{pmatrix} 4.6 \\ 3.4 \\ 1.4 \\ 0.3 \end{pmatrix}, \mathbf{x}_8 = \begin{pmatrix} 5.0 \\ 3.4 \\ 1.5 \\ 0.2 \end{pmatrix}\}$$

먼저 평균벡터를 구하면 $\boldsymbol{\mu} = (4.9125, 3.3875, 1.45, 0.2375)^T$ 이다. 첫 번째 샘플 \mathbf{x}_1 을 식 (2.39)에 적용하면 다음과 같다.

$$\begin{aligned} (\mathbf{x}_1 - \boldsymbol{\mu})(\mathbf{x}_1 - \boldsymbol{\mu})^T &= \begin{pmatrix} 0.1875 \\ 0.1125 \\ -0.05 \\ -0.0375 \end{pmatrix} \begin{pmatrix} 0.1875 & 0.1125 & -0.05 & -0.0375 \end{pmatrix} \\ &= \begin{pmatrix} 0.0325 & 0.0211 & -0.0094 & -0.0070 \\ 0.0211 & 0.0127 & -0.0056 & -0.0042 \\ -0.0094 & -0.0056 & 0.0025 & 0.0019 \\ -0.0070 & -0.0042 & 0.0019 & 0.0014 \end{pmatrix} \end{aligned}$$

나머지 7개 샘플도 같은 계산을 한 다음, 결과를 모두 더하고 8로 나누면 다음과 같은 공분산 행렬을 얻는다.

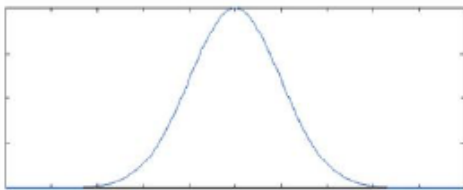
$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.0661 & 0.0527 & 0.0181 & 0.0083 \\ 0.0527 & 0.0736 & 0.0181 & 0.0130 \\ 0.0181 & 0.0181 & 0.0125 & 0.0056 \\ 0.0083 & 0.0130 & 0.0056 & 0.0048 \end{pmatrix}$$

유용한 확률분포

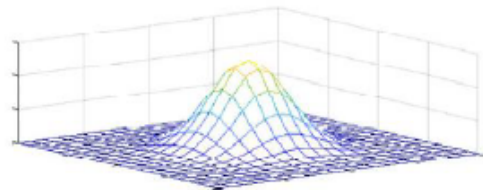
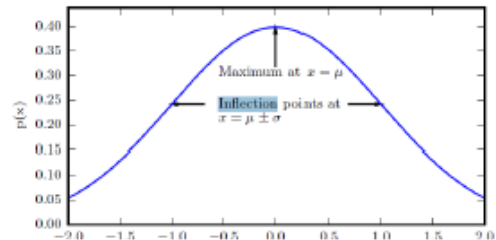
- 가우시안 분포 (Gaussian distribution)

▪ 평균 μ 와 분산 σ^2 으로 정의

$$N(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right)$$



(a) 1차원



(b) 2차원

그림 2-19 가우시안 분포

- 다차원 가우시안 분포

평균벡터 $\boldsymbol{\mu}$ 와 공분산행렬 $\boldsymbol{\Sigma}$ 로 정의

$$N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|} \sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

- 베르누이 분포 (Bernoulli distribution)

성공($x=1$) 확률 p 이고 실패($x=0$) 확률이 $1-p$ 인 분포

$$Ber(x; p) = p^x(1-p)^{1-x} = \begin{cases} p, & x = 1 \text{ 일 때} \\ 1-p, & x = 0 \text{ 일 때} \end{cases}$$

- 이항 분포 (Binomial distribution)

성공 확률이 p 인 베르누이 실험을 m 번 수행할 때 성공할 횟수의 확률분포

$$B(x; m, p) = C_m^x p^x (1-p)^{m-x} = \frac{m!}{x! (m-x)!} p^x (1-p)^{m-x}$$

확률질량함수

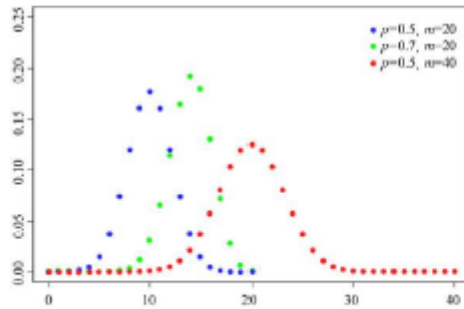
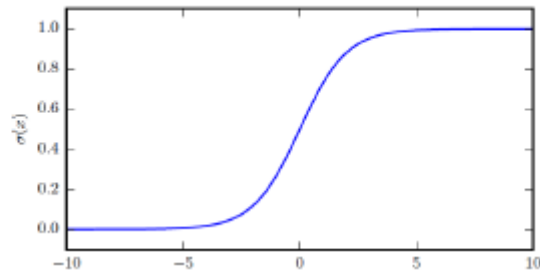


그림 2-20 이항 분포

- 확률 분포와 연관된 유용한 함수들

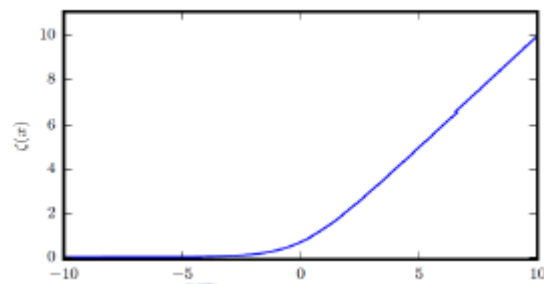
- 로지스틱 시그모이드 함수 (logistic sigmoid function)

일반적으로 베르누이 분포의 매개변수를 조정을 통해 얻어짐



- 소프트플러스 함수 (softplus function)

정규 분포의 매개변수의 조정을 통해 얻어짐



- 지수 분포 (exponential distribution)

$$p(x; \lambda) = \lambda \mathbf{1}_{x \geq 0} \exp(-\lambda x)$$

- 라플라스 분포 (laplace distribution)

$$\text{Laplace}(x; \mu, \gamma) = \frac{1}{2\gamma} \exp\left(-\frac{|x - \mu|}{\gamma}\right)$$

- 디랙 분포 (dirac distribution)

$$p(x) = \delta(x - \mu)$$

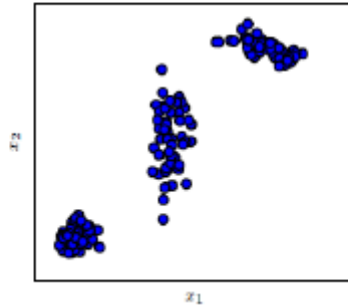
- 경험적 분포 (empirical distribution)

$$\hat{p}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

- 혼합 분포들 (mixture distributions)

$$P(\mathbf{x}) = \sum_i P(c = i)P(\mathbf{x} | c = i)$$

3개의 요소를 가진 가우시안 혼합 분포 예시 (가우시안 혼합 모델 추정 가능)



- 변수 변환 (change of variables)
 - 기존 확률변수를 새로운 확률변수로 바꾸는 것
 - 변환 $y = g(x)$ 와 가역성을 가진 g 에 의해 정의되는 x, y 두 확률변수를 가정할 때 두 확률 변수는 다음과 같이 상호 정의될 수 있음

$$p_x(\mathbf{x}) = p_y(g(\mathbf{x})) \left| \det \left(\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

예) 확률변수 x 의 확률질량함수가 다음과 같을 때,

$$\left(\frac{4}{5}\right)\left(\frac{1}{5}\right)^{x-1}, x=1, 2, \dots$$

새로운 확률변수 $y=x^2$ 의 확률질량함수는 다음과 같이 정의됨

$$y=x^2 \Rightarrow x=\sqrt{y}$$

$$f(x)=f(\sqrt{y})=\left(\frac{4}{5}\right)\left(\frac{1}{5}\right)^{\sqrt{y}-1}=g(y) \longrightarrow g(y)=\begin{cases} \left(\frac{4}{5}\right)\left(\frac{1}{5}\right)^{\sqrt{y}-1} & , y=1, 4, 9, \dots \\ 0 & , \text{elsewhere} \end{cases}$$

정보이론

- 정보이론과 확률통계는 많은 교차점을 가짐
- 확률통계는 기계학습의 기초적인 근간을 제공
 - 정보이론 관점에서 기계학습을 접근이 가능
 - 해당 확률 분포를 추정하거나 확률 분포 간의 유사성을 정량화 등의 기계 학습에 정보이론을 활용한 예로서 엔트로피, 교차 엔트로피, KL 다이버전스
- 정보이론 : 사건이 지닌 정보를 수량화할 수 있나?
 - "아침에 해가 뜬다" 와 오늘 "아침에 일식이 있었다"라는 두 사건중 어느 것이 더 많은 정보를 가지는지
 - 정보이론의 기본 원리 ► 확률이 작을수록 많은 정보
 - 자주 발생하는 사건보다 잘 일어나지 않는 사건 (unlikely event)의 정보량이 많음

자기 정보 (self information)

사건(메시지) e_i 의 정보량

(단위: 로드의 밑이 2인 경우, 비트^{bit} 또는 로그의 밑이 자연상수 경우, 나츠^{nat})

$$h(e_i) = -\log_2 P(e_i) \quad \text{또는} \quad h(e_i) = -\log_e P(e_i) \quad (2.44)$$

예를 들면, 동전에서 앞면이 나오는 사건의 정보량은 $-\log_2 (1/2)=1$ 이고,

주사위에서 1이 나오는 사건의 정보량은 $-\log_2 (1/6) \approx 2.58$ 임

따라서, 후자의 사건이 전자의 사건보다 높은 정보량을 가짐

엔트로피 (entropy)

- 확률 변수 x 의 불확실성을 나타내는 엔트로피
- 모든 사건 정보량의 기대값으로 표현

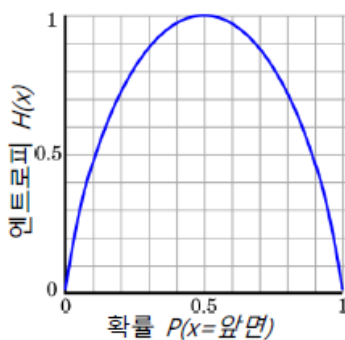
$$\text{이산 확률분포} \quad H(x) = - \sum_{i=1,k} P(e_i) \log_2 P(e_i) \quad \text{또는} \quad H(x) = - \sum_{i=1,k} P(e_i) \log_e P(e_i) \quad (2.45)$$

$$\text{연속 확률분포} \quad H(x) = - \int_{\mathbb{R}} P(x) \log_2 P(x) \quad \text{또는} \quad H(x) = - \int_{\mathbb{R}} P(x) \log_e P(x) \quad (2.46)$$

- 예를 들면, 동전의 앞뒤의 발생 확률이 동일한 경우의 엔트로피는

$$\begin{aligned} H(x) &= - \sum_x P(x) \log P(x) \\ &= - (0.5 \times \log_2 0.5 + 0.5 \times \log_2 0.5) \\ &= -\log_2 0.5 \\ &= -(-1) \end{aligned}$$

- 동전의 발생 확률에 따른 엔트로피 변화



→ 공평한 동전을 사용할 때에 가장 큰 엔트로피를 구할 수 있으며, 동전 던지기 결과 전송에는 최대 1비트가 필요함을 의미

- 모든 사건이 동일한 확률을 가질 때
즉, 불확실성이 가장 높은 경우, 엔트로피가 최고임

- 자기 정보와 엔트로피 예제

웁을 나타내는 확률변수를 x 라 할 때 x 의 엔트로피는 다음과 같다.

$$H(x) = -\left(\frac{4}{16}\log_2\frac{4}{16} + \frac{6}{16}\log_2\frac{6}{16} + \frac{4}{16}\log_2\frac{4}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{1}{16}\log_2\frac{1}{16}\right) = 2.0306\text{비트}$$

주사위는 눈이 6개인데 모두 1/6이라는 균일한 확률을 가진다. 이 경우 엔트로피를 계산하면 다음과 같다.

$$H(x) = -\left(\frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6}\right) = 2.585\text{ 비트}$$

▪ 주사위가 웁보다 엔트로피가 높은 이유는?

- 주사위는 모든 사건이 동일한 확률 \rightarrow 어떤 사건이 일어날지 웁보다 예측이 어려움
- 주사위가 웁보다 더 **무질서하고 불확실성이 큼** \rightarrow **엔트로피가 높음**
- 정의역의 크기가 크면 엔트로피도 커짐
 - 주사위 6개, 웁 5개 하지만 해당 주사위와 웁의 정의역을 동일하게 해도 주사위의 엔트로피가 큼

• **교차 엔트로피 (cross entropy)** : 두 확률분포 P 와 Q 사이의 교차 엔트로피

$$H(P, Q) = -\sum_x P(x)\log_2 Q(x) = -\sum_{i=1,k} P(e_i)\log_2 Q(e_i) \quad (2.47)$$

- 딥러닝의 손실함수로 자주 사용됨
- 식을 전개하면

$$\begin{aligned} H(P, Q) &= -\sum_x P(x)\log_2 Q(x) \\ &= -\sum_x P(x)\log_2 P(x) + \sum_x P(x)\log_2 P(x) - \sum_x P(x)\log_2 Q(x) \\ &= H(P) + \underbrace{\sum_x P(x)\log_2 \frac{P(x)}{Q(x)}}_{\text{KL 다이버전스 divergence}} \end{aligned}$$

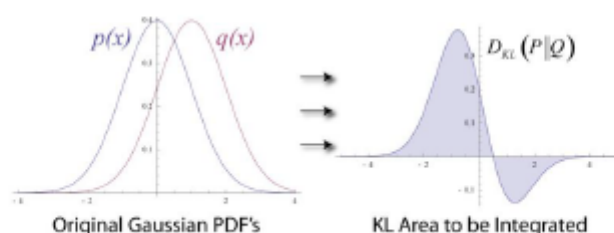
▪ 여기서 P 를 데이터의 분포라 하면, 이는 학습 과정에서 **변화하지 않음**

▪ 교차 엔트로피를 손실함수로 사용하는 경우, **KL 다이버전스의 최소화함과 동일**

• KL 다이버전스

- 식 (2.48)은 P 와 Q 사이의 KL 다이버전스
- 두 확률분포 사이의 거리를 계산할 때 주로 사용

$$KL(P \parallel Q) = \sum_x P(x)\log_2 \frac{P(x)}{Q(x)} \quad (2.48)$$



- 교차 엔트로피와 KL 다이버전스의 관계

$$P \text{와 } Q \text{의 교차 엔트로피 } H(P, Q) = H(P) + \sum_x P(x) \log_2 \frac{P(x)}{Q(x)} \quad (2.49)$$

$$= P \text{의 엔트로피} + P \text{와 } Q \text{ 간의 KL 다이버전스}$$

- 가지고 있는 데이터 분포 $P(x)$ 와 추정된 데이터 분포 $Q(x)$ 간의 차이 최소화하는데 교차 엔트로피 사용

예제 2-9

[그림 2-21]과 같이 정상적인 주사위와 찌그러진 주사위가 있는데, 정상적인 주사위의 확률분포는 P , 찌그러진 주사위의 확률분포는 Q 를 따르며, P 와 Q 가 다음과 같이 분포한다고 가정하자.

$$P(1) = \frac{1}{6}, P(2) = \frac{1}{6}, P(3) = \frac{1}{6}, P(4) = \frac{1}{6}, P(5) = \frac{1}{6}, P(6) = \frac{1}{6}$$

$$Q(1) = \frac{3}{12}, Q(2) = \frac{1}{12}, Q(3) = \frac{1}{12}, Q(4) = \frac{1}{12}, Q(5) = \frac{3}{12}, Q(6) = \frac{3}{12}$$



그림 2-21 확률분포가 다른 두 주사위

확률분포 P 와 Q 사이의 교차 엔트로피와 KL 다이버전스는 다음과 같다.

$$H(P, Q) = -\left(\frac{1}{6} \log_2 \frac{3}{12} + \frac{1}{6} \log_2 \frac{1}{12} + \frac{1}{6} \log_2 \frac{1}{12} + \frac{1}{6} \log_2 \frac{1}{12} + \frac{1}{6} \log_2 \frac{3}{12} + \frac{1}{6} \log_2 \frac{3}{12}\right) = 2.7925$$

$$KL(P \parallel Q) = \frac{1}{6} \log_2 \frac{2}{3} + \frac{1}{6} \log_2 2 + \frac{1}{6} \log_2 2 + \frac{1}{6} \log_2 2 + \frac{1}{6} \log_2 \frac{2}{3} + \frac{1}{6} \log_2 \frac{2}{3} = 0.2075$$

[예제 2-8]에서 P 의 엔트로피 $H(P)$ 는 2.585이었다. 따라서 식 (2.49)가 성립함을 알 수 있다.

최적화

- 순수 수학 최적화와 기계 학습 최적화의 차이
 - 기계학습의 최적화는 단지 **훈련집합**이 주어지고, 훈련집합에 따라 정해지는 **목적함수의 최저점**을 찾아야 함
 - 주로 SGD(스토캐스틱 경사 하강법) 사용
 - 데이터로 미분하는 과정 필요 ▶ 오류 역전파 알고리즘

매개변수 공간의 탐색

- 학습 모델의 매개변수 공간
 - 높은 차원에 비해 **훈련집합의 크기가 작아 참인 확률분포를 구하는 일은 불가능**
 - 기계학습은 적절한 모델을 선택하고, 목적함수를 정의하고, 모델의 매개변수 공간을 탐색하여 목적함수가 최저가 되는 최적점을 찾는 전략 사용
 - 특징공간에서 해야 하는 일을 모델의 매개변수 공간에서 하는 일로 대체

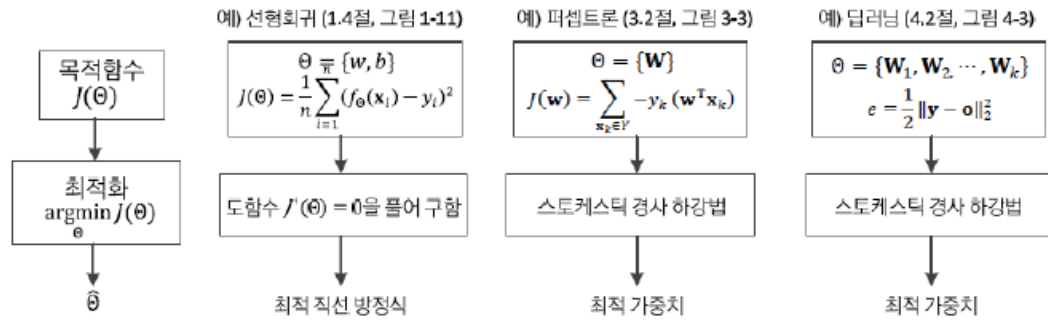


그림 2-22 최적화를 이용한 기계 학습의 문제풀이 과정

- 목적함수 : 세타 = 가설에 의해 생성
- 손실함수, W는 미분이 가능한 형태여야 함
- 특징공간보다 수 배 ~ 수만배 넓은
 - 선형회귀에서의 특징공간은 1차원, 매개변수 공간은 2차원
 - MNIST 인식하는 딥러닝 모델 : 784차원 특징공간, 수십~수백만 차원 매개변수 공간

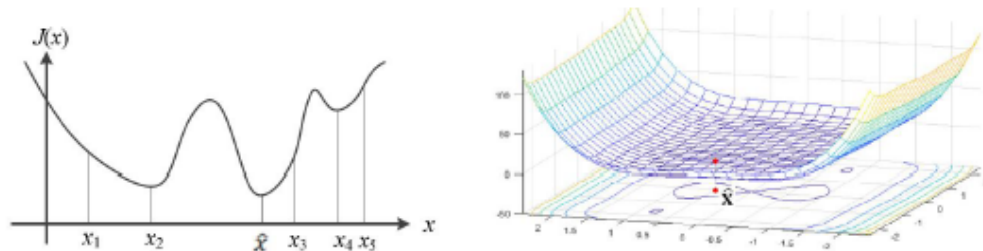


그림 2-23 최적해 탐색

[그림 2-23] 개념도의 매개변수 공간: \hat{x} 은 전역(global) 최적해, x_2 와 x_4 는 지역(local) 최적해
 x_2 와 같이 전역 최적해에 가까운 지역 최적해를 찾고 만족하는 경우 많음

- 최적화 문제 해결
 - 낱날탐색(exhaustive search) 알고리즘
 - 차원이 조금만 높아져도 적용 불가능

알고리즘 2-1 낱날탐색 알고리즘

입력: 훈련집합 X 와 Y

출력: 최적해 $\hat{\theta}$

- 1 가능한 해를 모두 생성하여 집합 S 에 저장한다.
- 2 min 을 충분히 큰 값으로 초기화한다.
- 3 for (S 에 속하는 각 점 $\theta_{current}$ 에 대해)
- 4 if ($J(\theta_{current}) < min$) $min = J(\theta_{current})$, $\theta_{best} = \theta_{current}$
- 5 $\hat{\theta} = \theta_{best}$

- 무작위탐색(random search) 알고리즘
 - 아무 전략이 없는 순진한 알고리즘

알고리즘 2-2 무작위 탐색 알고리즘

입력: 훈련집합 \mathcal{X} 와 \mathcal{Y}

출력: 최적해 $\hat{\theta}$

```

1   $min$ 을 충분히 큰 값으로 초기화한다.
2  repeat
3      무작위로 해를 하나 생성하고  $\theta_{current}$  라 한다.
4      if( $J(\theta_{current}) < min$ )  $min = J(\theta_{current})$ ,  $\theta_{best} = \theta_{current}$ 
5  until(멈춤 조건)
6   $\hat{\theta} = \theta_{best}$ 
    
```

- 기계학습이 사용하는 전형적인 알고리즘
 - 목적함수(J)가 작아지는 방향을 주로 미분으로 찾아냄(라인 3)

알고리즘 2-3 기계 학습이 사용하는 전형적인 탐색 알고리즘(1장의 [알고리즘 1-1]과 같음)

입력: 훈련집합 \mathcal{X} 와 \mathcal{Y}

출력: 최적해 $\hat{\theta}$

```

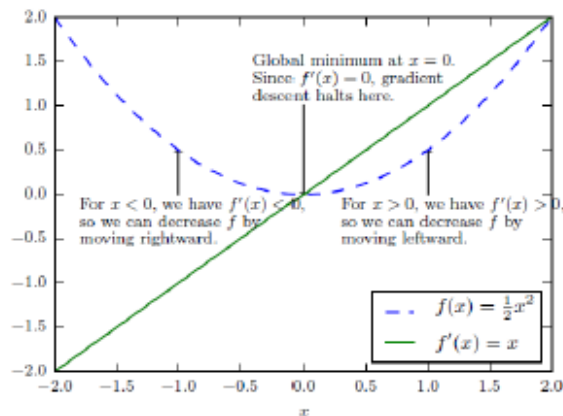
1  난수를 생성하여 초기해  $\theta$ 을 설정한다.
2  repeat
3       $J(\theta)$ 가 작아지는 방향  $d\theta$ 를 구한다.
4       $\theta = \theta + d\theta$ 
5  until(멈춤 조건)
6   $\hat{\theta} = \theta$ 
    
```

미분

- 미분에 의한 최적화
 - 미분의 정의

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}, \quad f''(x) = \lim_{\Delta x \rightarrow 0} \frac{f'(x + \Delta x) - f'(x)}{\Delta x} \quad (2.51)$$

- 1차 도함수 $f'(x)$ 는 함수의 기울기, 즉 값이 커지는 방향(증가하는 방향)을 지시
 - 기울기가 양수면 오른쪽이 상승방향, 음수면 왼쪽이 상승방향
 - $-f'(x)$ 방향에 목적함수의 최저점이 존재



[알고리즘 2-3]에서 $d\theta$ 로 $-f'(x)$ 를 사용함 ← 경사 하강 알고리즘의 핵심 원리

- 편미분(Partial Derivative)
 - 변수가 여러 개인 함수의 미분
 - 미분값이 이루는 벡터를 그래디언트라 부름

여러 가지 표기: $\nabla f, \frac{\partial f}{\partial \mathbf{x}}, \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right)^T$

예)

$$\left. \begin{aligned} f(\mathbf{x}) = f(x_1, x_2) &= \left(4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2) x_2^2 \\ \nabla f = f'(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} &= \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right)^T = (2x_1^5 - 8.4x_1^3 + 8x_1 + x_2, 16x_2^3 - 8x_2 + x_1)^T \end{aligned} \right\} \quad (2.52)$$

- 기계 학습에서 편미분
 - 매개변수 집합 Θ 에 많은 변수가 있으므로 편미분 사용
- 편미분으로 얻은 **그레디언트에 따라 최저점을 찾아가는** 예제

예제 2-10

초기점 $\mathbf{x}_0 = (0.5, 0.5)^T$ 라고 하자. \mathbf{x}_0 에서의 그레디언트는 $f'(\mathbf{x}_0) = (-2.5125, -2.5)^T$ 즉, $\nabla f|_{\mathbf{x}_0} = (-2.5125, -2.5)^T$ 이다. [그림 2-25]는 \mathbf{x}_0 에서 그레디언트를 화살표로 표시하고 있어, $-f'(\mathbf{x}_0)$ 은 최저점의 방향을 제대로 가리키는 것을 확인할 수 있다. 하지만 얼마만큼 이동하여 다음 점 \mathbf{x}_1 로 옮겨갈지에 대한 방안은 아직 없다. 2.3.3절에서 공부하는 경사 하강법은 이에 대한 답을 제공한다.

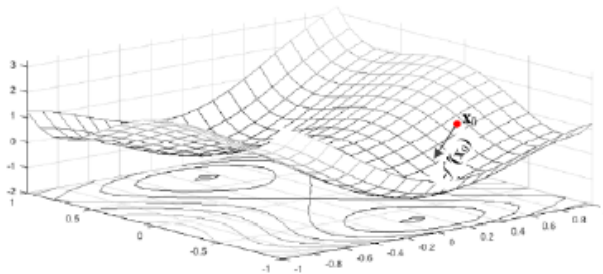


그림 2-25 그레디언트는 최저점으로 가는 방향을 알려 줌

- 독립변수와 종속변수의 구분
 - $[y = wx + b]$ 에서 x 는 독립변수, y 는 종속변수
 - 기계학습에서 이런 해석은 무의미(예측을 위한 해석에 불과)
 - 최적화는 예측이 아니라 학습 단계에 필요

식 (1.8)에서 Θ 가 독립변수이고 $e = J(\Theta)$ 라 하면 e 가 종속변수임

$$J(\Theta) = \frac{1}{n} \sum_{i=1}^n (f_{\Theta}(\mathbf{x}_i) - y_i)^2 \quad (1.8)$$

- 연쇄법칙(Chain Rule)

합성함수 $f(x) = g(h(x))$ 와 $f(x) = g(h(i(x)))$ 의 미분

$$\left. \begin{aligned} f'(x) &= g'(h(x))h'(x) \\ f'(x) &= g'(h(i(x)))h'(i(x))i'(x) \end{aligned} \right\} \quad (2.53)$$

예) $f(x) = 3(2x^2 - 1)^2 - 2(2x^2 - 1) + 5$ 일 때 $h(x) = 2x^2 - 1$ 로 두면,

$$f'(x) = \underbrace{(3 * 2(2x^2 - 1) - 2)}_{g'(h(x))} \underbrace{(2 * 2x)}_{h'(x)} = 48x^3 - 32x$$

- 다층 퍼셉트론은 합성함수

$\frac{\partial o_i}{\partial u_{23}^1}$ 를 계산할 때 연쇄법칙 적용

3.4절 (오류 역전파)에서 설명

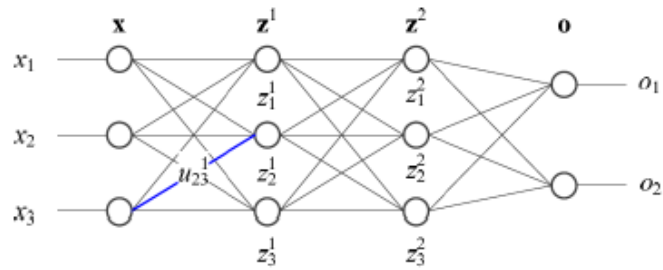


그림 2-26 다층 퍼셉트론은 합성함수

- 야코비언 행렬(Jacobian Matrix)
 - 편미분을 행렬의 형태로 확장한 것, 행렬 연산은 GPU 연산에 효율적
 - 자주 사용됨

함수 $f: \mathbb{R}^d \mapsto \mathbb{R}^m$ 을 미분하여 얻은 행렬

예)

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \nabla f_1 \\ \vdots \\ \nabla f_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \dots & \frac{\partial f}{\partial x_n} \end{pmatrix}$$

$$f: \mathbb{R}^2 \mapsto \mathbb{R}^3 \ni f(\mathbf{x}) = (2x_1 + x_2^2, -x_1^2 + 3x_2, 4x_1x_2)^T$$

$$J = \begin{pmatrix} 2 & 2x_2 \\ -2x_1 & 3 \\ 4x_2 & 4x_1 \end{pmatrix} \quad J|_{(2,1)^T} = \begin{pmatrix} 2 & 2 \\ -4 & 3 \\ 4 & 8 \end{pmatrix}$$

- 헤시안 행렬(Hessian Matrix)
 - 2차 미분

2차 편도함수

예)

$$f(\mathbf{x}) = f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

$$\text{헤시안 행렬 } H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

$$H = \begin{pmatrix} 10x_1^4 - 25.2x_1^2 + 8 & 1 \\ 1 & 48x_2^2 - 8 \end{pmatrix}$$

$$H|_{(0,1)^T} = \begin{pmatrix} 8 & 1 \\ 1 & 40 \end{pmatrix}$$

경사 하강 알고리즘

- 경사하강법(gradient descent)이 낮은 곳을 찾아가는 원리

$\mathbf{g} = d\boldsymbol{\theta} = \frac{\partial J}{\partial \boldsymbol{\theta}}$ (기울기)이고, ρ 는 학습률 (이동할 거리 조절)

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \rho \mathbf{g} \quad (2.58)$$

함수의 기울기 (경사)를 구하여 기울기가 낮은 쪽으로 반복적으로 이동하여 극값에 도달

- 얼마나 이동할건지를 결정해야 함
- 배치(batch) 경사 하강 알고리즘
 - 샘플의 그레이디언트를 평균한 후 한꺼번에 갱신
 - 훈련집합 전체를 다 봐야 갱신이 일어나므로 학습 과정이 오래 걸리는 단점

알고리즘 2-4 배치 경사 하강 알고리즘(BGD)

입력: 훈련집합 \mathbb{X} 와 \mathbb{Y} , 학습률 ρ

출력: 최적해 $\hat{\theta}$

```

1  난수를 생성하여 초기해  $\theta$ 를 설정한다.
2  repeat
3     $\mathbb{X}$ 에 있는 샘플의 그레이디언트  $\nabla_1, \nabla_2, \dots, \nabla_n$ 을 계산한다.
4     $\nabla_{total} = \frac{1}{n} \sum_{i=1, n} \nabla_i$  // 그레이디언트 평균을 계산
5     $\theta = \theta - \rho \nabla_{total}$ 
6  until(멈춤 조건)
7   $\hat{\theta} = \theta$ 
    
```

훈련집합

$$\mathbb{X} = \{x_1, x_2, \dots, x_n\}$$

$$\mathbb{Y} = \{y_1, y_2, \dots, y_n\}$$

• 스토캐스틱 경사 하강(SGD) 알고리즘

- 한 샘플 혹은 미니배치의 그레이디언트를 계산한 후 즉시 갱신
 - 대표성을 취하는 샘플을 보고 그레이디언트를 계산
- 라인 3~6을 한번 반복 : 한 세대(epoch)

알고리즘 2-5 스토캐스틱 경사 하강 알고리즘(SGD)

입력: 훈련집합 \mathbb{X} 와 \mathbb{Y} , 학습률 ρ

출력: 최적해 $\hat{\theta}$

```

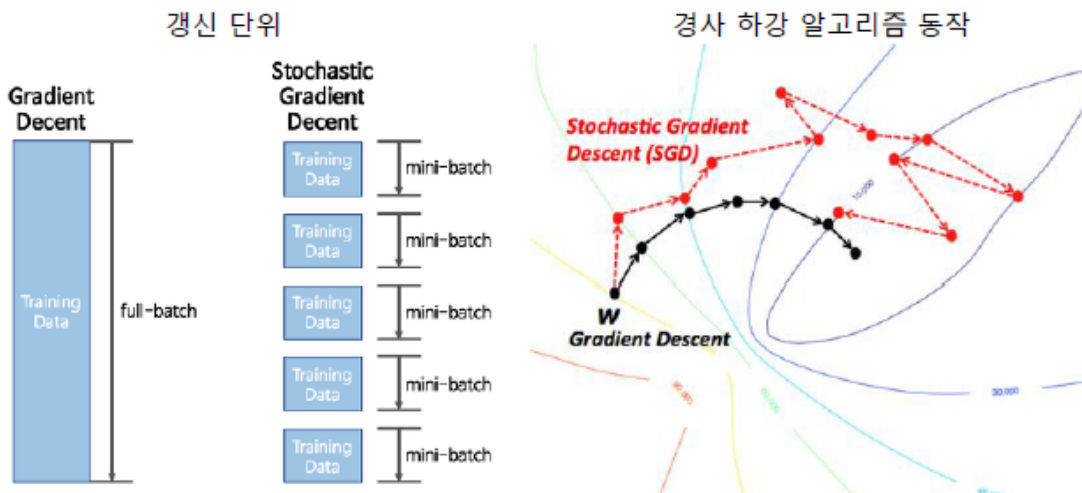
1  난수를 생성하여 초기해  $\theta$ 를 설정한다.
2  repeat
3     $\mathbb{X}$ 의 샘플의 순서를 섞는다.
4    for ( $i=1$  to  $n$ )
5       $i$ 번째 샘플에 대한 그레이디언트  $\nabla_i$ 를 계산한다.
6       $\theta = \theta - \rho \nabla_i$ 
7  until(멈춤 조건)
8   $\hat{\theta} = \theta$ 
    
```

- 다른 방식의 구현 (라인 3~6 대치)

```

3   $\mathbb{X}$ 에서 임의로 샘플 하나를 뽑는다.
4  뽑힌 샘플의 그레이디언트  $\nabla$ 를 계산한다.
5   $\theta = \theta - \rho \nabla$ 
    
```

• 경사 하강 알고리즘 비교



- 배치 경사 하강 알고리즘 : 정확한 방향으로 수렴, 느림
- 스토캐스틱 경사 하강 알고리즘 : 수렴이 다소 헤맬 수 있음, 빠름
 - 결국 최소값에 도달하긴 한다

• 추가 경사 하강 알고리즘 비교

