1장 기초 수학과 미적분

오일러의 수와 자연로그

1) 오일러의 수

- 오일러의 수 e는 약 2.71828
- 이자를 복리로 계산할 때, 월 -> 일 -> 분 -> 시 -> 무한히 작은 순간을 n에 적용
- $(1+\frac{1}{n})^n$ 에서 n이 무한대로 커지면 약 2.781828에 수렴
- 오일러의 수가 밑인 지수 함수의 도함수(미분값)가 자기 자신이기 때문에 계산 상의 효율성으로 인해 머신 러닝에서 유용

2) 자연로그

- 오일러의 수 e를 밑으로 사용하는 로그: $ln()$
- $ln(10)$은 e를 거듭제곱해서 10이 되는 수

미분(Derivative)

- x의 아주 작은 변화량에 대한 y의 변화량, 기울기
- 편도함수란?
  - $f(x, y) = 2x^3 + 3y^3$ 일 때, x와 y변수는 각각 고유한 도함수(기울기=gradient) $\frac{df}{dx}$와 $\frac{df}{dy}$를 갖음
- 연쇄법칙이란?
  - 신경망 층을 훈련할 때 각 노드의 도함수를 곱해서 계산함
  - 다음 두 개의 함수에서 $\frac{dz}{dx}$를 구하려면?
    - $y = x^2 + 1, z = y^3 - 2$
    - $\frac{dz}{dx} = \frac{dz}{dy} \times \frac{dy}{dx}$

적분

2장 확률

확률 이해

- 확률(probability) vs 가능도(likelihood)
  - 확률: 일어나지 않은 사건(미래)에 대한 예측
    - Sum(모든 상호 배타적인 결과의 확률) = 1
  - 가능도: 이미 발생한 사건의 빈도를 측정
    - Sum(모든 상호 배타적인 결과의 확률) 이 1 아닐 수 있음
    - 머신러닝에서는 확률(미래)을 예측하기 위해 데이터 형태에 대한 가능도(과거)를 사용
  - 오즈(Odds)
    - $P(X) = \frac{O(X)}{1+O(X)} \qquad O(X) = \frac{P(X)}{1-P(X)}$

    - 오즈가 2.0 이면? 일어날 확률이 그렇지 않을 확률보다 두배 더 높다
    - 백분율보다 더 직관적으로 설명 가능
  - 상호 배타성(독립적)
    - A 또는 B 중 하나의 결과만 가능하고 둘 다 허용 불가
    - $P(A\,and\,B) = 0$
  - 결합법칙
    - 두 사건이 동시에 일어날 확률
    - 상호 배타적이지 않을 때
      - $P(A\,and\,B) = P(A|B) \times P(B)$
    - 상호 배타적일 때(즉, 독립적일 때)
      - $P(A\,and\,B) = P(A) \times P(B)$
  - 덧셈 법칙
    - $P(A\,or\,B) = P(A) + P(B) - P(A|B) \times P(B)$

조건부 확률(Conditional Probability)

- P(A Given B) 또는 P(A|B): B가 발생했을 때, A가 발생할 확률
- 커피가 암과과 연관이 있는지 연구하려면, 커피를 마시는 사람이 암에 걸릴 확률(조건부 확률)을 구해야 함
  - P(암|커피)

베이즈 정리(Bayesian Rule)

- 사후 확률을 구하기 위한 규칙으로 사후 확률을 라이클리후드와 사전 확률의 곱으로 표현한 것
- MAP, MLE??
- 새로운 정보가 들어왔을 때, 우리의 믿음(확률)을 어떻게 바꿔야 하는지를 알려주는 규칙
- 새로운 증거에 따라 기존 확률(Prior)을 "업데이트(Posteior)"
- $P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} = \frac{P(A\ AND\ B)}{P(B)}$
- $사후확률(Posteior) = \frac{우도(Likelihood) \times 사전확률(Prior)}{증거거}$

이항 분포

베타 분포

3장 기술 통계와 추론 통계

- 기술 통계
  - 데이터를 요약
  - 평균 mean, 중앙값 median, 모드 mode, 차트, 종 곡선을 계산
    - 중앙값: 정렬된 값 집합 중 가장 가운데 값, 이상치로 평균을 신뢰할 수 없을 때 유용한 대안
    - 모드: 가장 자주 발생하는 값
- 추론 통계

    - 표본을 기반으로 모집단에 대한 속성을 발견
    - 일종의 유추

3-1 기술 통계

모집단

- 연구하고자 하는 특정 관심 그룹
- 분산
    - 편차의 제곱의 평균
    - $\sigma^2 = \frac{\sum(x_i - \mu)^2}{N}$
- 표준편차
    - 분산에서 제곱을 제거거
    - $\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$

표본

- 모집단의 하위 집합
- 무작위, 편향되지 않음
- 모집단에 대한 속성을 추론하기 위한 목적
- 분산
    - $s^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$
- 표준편차
    - $s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$
- n이 아닌 n-1로 나누는 이유는?
    - 표본의 편향을 줄이고 표본에 기반한 모집단의 분산을 과소평가하지 않기 위해
    - 분모에서 하나 작은 값을 사용함으로써 분산을 증가시키고 표본의 불확실성을 더 많이 포착

정규 분포(Normal Distribution), 가우스 분포(Gaussian distribution)

- $N(m, \sigma^2)$
- 평균 근처가 가장 질량이 크고, 대칭 형태를 띤 종 모양 분포
- 자연과 일상생활에서 일어나는 많은 현상이 이 분포를 따르고 있으며
- 어떤 분포라도 표본이 충분이 크면 표본의 평균은 정규 분포를 따름
- 이 분포의 퍼짐 정도 = 표준 편자
- 꼬리는 가능성이 가장 낮은 부분이며 0에 수렴하지만 0은 아님
- 자연과 일상생활에서 일어나는 많은 현상과 유사
- 중심 극한 정리 덕분에 정규 분포가 아닌 문제에도 일반화 가능
- 머신 러닝에서 어떻게 활용 되나?
    - 인공신경망에서 초기 기중치의 값이 정규 분포를 따른다고 가정

- 각 가중치를 **무작위(random)**로 설정하면 뉴런마다 출력이 다르고, 각기 다른 역할을 하게 되어 학습이 가능(그렇지 않으면, 똑같은 출력을 내고 똑같이 업데이트되기 때문에 학습이 되지 않음)
    - 표준 정규분포를 샘플링 함

표준 정규 분포(Standard Normal Distribution)

- 평균이 0이고 표준 편차가 1이 되도록 정규 분포의 크기를 재조정, 즉 정규화
- 평균과 분산이 다른 정규 분포 간의 퍼짐 정도를 쉽게 비교
- 모든 x 값을 표준 편차, 즉 z 점수로 표현
    - $z = \frac{x-\mu}{\sigma}$
- 변동 계수(Coefficient Variation)
    - 두 분포를 비교해 각 분포가 얼마나 퍼져 있는지 정량화
    - $cv = \frac{\sigma}{\mu}$

3-2. 추론 통계

중심 극한 정리(Central Limit Theorem, CLT)

- 모집단에서 충분히 많은 표본을 추출하면 해당 모집단이 정규 분포를 따르지 않더라도 표본의 평균이 정규 분포를 따른다.
    - 조건 IID: Independently Identically Distributed 된 Random Sample
    - 어떤 정규 분포? $N(\mu, \frac{\sigma^2}{n})$
    - 특히, 모집단이 어떤 분포를 따르더라도 표본 평균의 분포는 무조건 정규 분포를 따르게 할 수 있다는 것이 가장 핵심이라고 생각합니다.
- 충분히 많은 표본 채취 -> 각각의 평균을 계산 -> 이를 하나의 분포로 표현 -> 정규 분포를 따름
- 특성1: 표본 평균의 평균 = 모집단의 평균
- 특성2: 모집단이 정규 분포이면 표본 평균도 정규 분포
- 특성3: 모집단이 정규 분포가 아니라도 표본 크기가 30보다 큰 경우 표본 평균은 대략적으로 정규 분포를 따름
- 특성4: 표본 평균의 표준 편차는 모집단 표준 편차를 n의 제곱근으로 나눈 값과 동일
    - $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$
    - n은 표본 평균의 개수, 즉 표본 수
    - 표본이 무한히 많아질수록록 표본 평균의 표준편차는 모집단의 표준편차와 유사해짐
- So What? 정규 분포가 아닌 경우에도 표본을 기반으로 모집단에 대한 유용한 정보를 추론할 수 있음

신뢰구간

- 표본 평균(또는 파라미터)이 모집단 평균의 특정 범위 속한다고 얼마나 믿는지
- 예시
    - 표본 평균이 64.408이고 표준 편차가 2.05인 골든 리트리버 31마리의 표본을 기준으로 모집단 평균이 63.686에서 65.1296 사이에 있다고 95% 확신한다.
- ± 1.95996 은 표준 정규 분포의 중심에서 확률의 95%에 해당하는 임계 z값

p 값이란?

가설 검정

t 분포: 소규모 표본 처리

4장 선형대수학

벡터란

- 벡터는 데이터를 시각적으로 표현
    - 공간상에서 방향과 길이를 가진 화살표
    - 데이터의 한 조각을 나타냄
        - 따라서, 데이터를 조작하는 것은 곧 벡터를 조작하는 것
    - 꼬리가 항상 자표 원점(0,0)에서 시작
    - 수평으로 3스텝, 수직으로 2스텝이동하는 벡터는?
        - $\vec{v} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$
    - 예시
        - 집의 편적이 18평방미터이고 가치가 30만달러인 데이터
        - $\vec{v} = \begin{bmatrix} 18 \\ 30만 \end{bmatrix}$
- 벡터의 덧셈에서 교환 법칙(Commutative Property)이 성립
    - $\vec{v}$이동 후 $\vec{w}$ 이동 하든지, 그 반대로 하든지 상관 없음
- 스케일링
    - 스칼라(Scalar)라는 하나의 값을 곱해 벡터를 늘이거나 줄임
        - 스칼라는 방향은 없고 크기만 있는 수(일반적인 실수나 복소수)
    - 스케일을 조정해도 여전히 같은 선상에 존재함 = 선형종속

Vector Space란

- 8개의 공리를 만족하는 벡터의 집합
    - 덧셈과 스칼라 곱 연산에 대해 닫혀 있는(항상 같은 공간 안에 머무는) 벡터들의 집합
- 8개의 공리
    - 덧셈 관련
      (1) 교환 법칙: u+v=v+u
      (2) 결합 법칙: (u+v)+w=u+(v+w)
      (3) 항등원 존재: 0 벡터가 존재하여 v+0=v
      (4) 역원 존재: v+(−v)=0

    - 스칼라곱 관련
      (5) 분배법칙 1: a(v+w)=av+aw
      (6) 분배법칙 2: (a+b)v=av+bv
      (7) 결합법칙: a(bv)=(ab)v
      (8) 항등원: 1·v=v
- 대표적으로 linear sub space
    - 벡터들의 linear combination으로 표현할 수 있는 space

내적(Dot Product)이란

- 두 벡터가 얼마나 닮았는지를 표현, 하나의 스칼라 값을 반환
    - $\begin{bmatrix}1\\3\end{bmatrix} \cdot \begin{bmatrix}4\\2\end{bmatrix}$ = 1 x 4 + 3 x 2 = 10
- 대수적 정의
    - 각 성분끼리 곱하고, 그 결과를 모두 더함
- 기하학적 정의
    - 하나의 벡터를 다른 벡터에 projection 후 길이를 서로 곱함
        - 방향이 유사할 수록 projection 후 길이가 유지됨
    - a·b=∥a∥ ∥b∥cos(θ)
        - ∥a∥, ∥b∥: 벡터의 크기(길이)
    - 내적 = 0: 두 벡터가 직각 (수직)
- 예를 들어, Word2Vec에서 임베딩 모델에서 단어를 벡터로 표현한 뒤, 두 단어 벡터의 내적을 통해 의미적 유사도를 계산

스팬과 선형 종속

- 다른 두 방향을 향하는 두개의 벡터를 스케일링하고 더하면 세로운 벡터를 얼마든지 만들 수 있음
- 스팬(Span)
    - 가능한 벡터의 전체 공간
- 선형 독립(Linearly Independent)
    - 서로 방향이 다른 두 벡터는 선형 독립이며 스팬이 무한함
    - Independent한 벡터의 수만큼 차원이 결정됨
- 선형 종속(Linearly Dependent)
    - 같은 방향이나 선상에 존재하는 두 벡터는 선형 종속
    - 하나의 벡터를 스케일링하여 다른 벡터를 만들 수 있을 때 서로 종속
- 선형 독립일 때 벡터들이 유연하게 움직이고, 해를 쉽게 구함
    - 연립 방정식에서 선형 종속이 있는 경우 변수가 사라지기 때문에 문제를 풀 수 없음

선형 변환

- 선형 종속의 경우를 제외하고, 벡터의 결합은 원하는 방향과 길이를 가질 수 있음
- 기저 벡터의 움직임을 이용해 벡터를 늘이고, 줄이고, 비틀고, 회전
- 선형 변환은 수학적 데이터 조작의 핵심

- 기저 벡터(Basis Vector)
  - 길이가 1이고 서로 수직이거나 독립적이며 양의 방향을 가리킴
  - 모든 벡터를 만들거나 변화하기 위한 구성 요소
  - $\hat{i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \hat{j} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{기저벡터} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- 4가지 선형 변환
  - 스케일: 늘어나거나 줄어듬
  - 회전: 공간을 돌림
  - 반전: 공간을 뒤집어 $\hat{i}$과 $\hat{j}$의 위치를 바꿈(Transpose)
  - 전단: 특정 방향의 직선과의 거리에 비례하여 각 포인트를 이동

행렬 벡터의 곱셈

- 선형 변환 후에 기저벡터 $\hat{i}$, $\hat{j}$이 도착하는 곳을 추적하는 개념
- 벡터를 만드는 것과 벡터를 변환하는 것이 실제로 같음을 깨달아야 함
- 기저 벡터가 선형 변환 전후의 출발점이라 생각하면 상대적일 것임
- 벡터의 곱셈을 통해 기저벡터 $\hat{i}$, $\hat{j}$이 변할 때 $\overrightarrow{v}$도 같이 변함
- 즉, 행렬의 곱셈은 벡터 공간에 여러 개의 변환을 적용하는 것
  - 순서: 가장 안쪽부터 바깥쪽으로 변환을 적용
  - 행렬A * 행렬B * 행렬C = (행렬A * 행렬B) * 행렬C = 행렬A * (행렬B * 행렬C)

Matrix의 4가지 Fundamental Vector Spaces

- 어떤 m×n 행렬 A에 대해, 다음의 네 가지 벡터 공간을 정의
  - 스팬하는 영역
- $A \in \mathbb{R}^{m \times n}$이고 $y = Ax$ 일 때
  - n: 입력 벡터의 차원 (열의 수)
  - m: 출력 벡터의 차원 (행의 수)
  - A: 입력 벡터 x를 받아 출력 벡터 y로 보내는 선형 변환
  - x: 입력공간 $\mathbb{R}^n$의 벡터
  - y: 출력공간 $\mathbb{R}^m$의 벡터

1. 열공간 (Column Space) = $C(A)$ or $range(A)$
   - 행렬 A의 열벡터들로 생성된 공간
     - 모델이 만들어낼 수 있는 결과값의 세트(출력 세계)
   - A가 어떤 벡터 x에 작용해서 도달할 수 있는 영역
     - 출력 벡터 y는 A의 열벡터들을 선형결합(linear combination) 하여 구성

        - 입력 x의 각 성분은 행렬 A의 열벡터에 각각 곱해져 더해지는 구조

$$A = \begin{bmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{bmatrix} \in \mathbb{R}^{m\times 3}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$Ax = x_1a_1 + x_2a_2 + x_3a_3$$

    - 예시:클래스 확률 벡터, 회귀 값, 문장 임베딩 등
    - 어떤 벡터 b가 열공간에 속한다는 것은?
        - 행렬 A의 열벡터들을 적절히 조합해서 b를 만들 수 있다.
2. 행공간 (Row Space) = $C(A^T)$
    - 행공간은 A의 **행벡터들(row vectors)**이 span하는 공간
        - 각 행벡터는 입력 벡터 x와의 내적을 통해 출력값의 한 요소를 만듦
        - 입력 벡터 x는 행벡터들이 정의된 공간, 즉 $\mathbb{R}^n$ 상에서 작동
        - 입력 벡터 x에서 어느 방향이 실제로 y에 영향을 미치는지를 결정
        - 행공간이 가지는 입력 벡터 x에 대한 영향력
    - 행공간은 모델이 이해하는 입력 데이터의 구조(입력 세계)
        - 행공간은 입력 벡터 중 출력에 영향을 주는 "정보 방향"만을 모은 것
        - 입력공간 전체($\mathbb{R}^n$) 중에서 행공간 방향으로 투영된 정보만 출력에 반영
        - 반대로 영공간(null space) 방향은 출력에 아무 영향 없음
    - A의 행들을 열처럼 생각해서 만든 A^T의 열공간
    - 예시: 이미지 벡터 (784차원), 텍스트 임베딩 벡터 등
3. 영공간 (Null Space) = $N(A)$
    - Ax=0(출력이 0이 되는)을 만족하는 모든 벡터 x
        - 즉, 행렬 A가 모두 0으로 보내버리는 입력 공간
        - x가 스팬하는 영역이 아닌, x들이 모인 공간(집합)
        - 행공간(Row Space)과 직교
            - A의 행공간과 입력 벡터가 만날 수 없어, 출력이 없음음
    - 어떤 입력 x가 영공간에 속한다는 것은?
        - A를 통해 아무 정보도 전달하지 못한다는 의미(출력공간에 아무 영향 없음)
        - 즉, A는 그 방향의 벡터를 완전히 '없애버린다'(행렬 A를 거치면 모든 성분 소멸)
    - 머신러닝적 해석
        - Null Space 방향은 모델이 감지하지 못하는 정보의 방향
        - "Null Space 방향의 벡터는 모델에게는 투명 인간과 같다. 아무리 존재해도, 모델은 그것을 '보지 못하고', 반응하지도 않는다. 그래서 출력값에 아무런 영향도 미치지 않는다."
4. 좌영공간 (Left Null Space) = $N(A^T)$
    - $A^Ty = 0$ 을 만족하는 모든 출력 벡터 y의 집합
        - 전치 행렬 A^T의 null space

        - 열공간(Column Space)과 직교
            - $Ax = b$ 에서 열공간에 b가 존재하지 않음
            - Ax의 선형변환으로 b를 표현할 수 없음
            - x의 해를 구할 수 없음
    - 즉, A^T가 0으로 보내는 벡터들

Let $A \in R^{m \times n}$. Prove that if the rank of A is r, then the dimension of its null space is $n−r$.

- Rank-Nullity 정리에 따르면, 행렬 A의 열 개수 n은 랭크 r과 널 스페이스의 차원의 합입니다. 따라서 널 스페이스의 차원은 $n−r$입니다.
- Rank-Nullity 란?
    - rank(A) + nullity(A) = 열의 수
    - A가 어떤 선형 변환을 나타낼 때, 출력 벡터 공간에 영향을 미치는 방향들과 입력은 있지만 결과가 0이 되는 방향들의 합은 항상 전체 입력 공간(열의 수) n을 완전히 설명해야 함

직교성(Orthogonality)

- 두 벡터 a와 b가 서로 직교한다
    - $a \cdot b = 0$
- 기하학적 의미
    - 두 벡터가 이루는 각이 90도
    - Projection 했을 때, 그 길이가 0
    - 서로 전혀 닮지 않음 $\rightarrow$ 전혀 관련이 없음 $\rightarrow$ 아무 정보도 공유하지 않음
- Row Space ⊥ Null Space
- Col Space ⊥ Left Null Space

랭크(Rank)

- 행렬에서 컬럼 스페이스의 차원의 수 입니다.
- 여기서, 컬럼 스페이스란 행렬의 컬럼들이 span하는 space 입니다.
- 그리고 span이란 선형 결합으로 표현할 수 있는 영역입니다.
- 또한 랭크는 로우 스페이스의 차원과도 동일합니다.
- rank(A): 행렬에서 선형적으로 독립적인 행벡터 또는 열벡터의 최대 개수
    - 이 행렬이 담고 있는 진짜 정보의 차원 수
    - 이 행렬에서 중복되지 않은 방향의 개수
    - 예시1: 선형 종속
        - $A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{bmatrix}$
        - 세 개의 행벡터 모두가 사실 [1,2]의 배수 → 선형 종속
        - 즉, 하나의 방향만 표현할 수 있음

        - rank(A) = 1
    - 예시2: 독립적인 벡터가 많을 때
        - $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
        - 3개의 독립적인 축 방향을 표현 → 완전한 3차원 정보 보유
        - rank(A) = 3
- 열랭크(column rank)와 행랭크(row rank)가 항상 같음
    - **independent 한 column의 수와 independent 한 row의 수는 항상 같다**
    - $rank(A) = rank(A^T)$
    - 행과 열은 서로 전치(transpose) 관계
    - 선형 변환으로 출력이 span하는 공간의 차원은 하나의 수로 고정
    - 결국, "이 행렬이 표현할 수 있는 독립적 방향성의 수"는 하나
- 랭크(rank)는 머신러닝에서 데이터 또는 모델이 표현할 수 있는 독립적인 정보의 차원 수를 의미
    - 데이터 행렬의 랭크
        - $X \in \mathbb{R}^{n \times d}$
        - n: 샘플 수(행), d: 특성 수(열)
        - rank(A)=d: 모든 특성이 독립적 → 완전한 차원 정보
        - rank(A)<d: 특성 간 중복 존재 → 차원 축소 가능 (PCA 적용 가능)
        - rank(A)<<d: 특성이 과도하게 중복됨 → 고차원 과적합 가능성↑
    - 가중치 행렬의 랭크: 표현력의 한계 또는 구조적 제약
    - 즉, 랭크가 높을수록 → 데이터가 더 "다양한 방향성"을 가짐

행렬식(Determinant)

- 역행렬의 여부를 나타내는 값
    - $det(A) = 0$ 이면
      → 선형 종속 발생
      → 차원 축소
      → Ax=b를 만족하는 x가 없거나(열공간에 없음), 무수히 많음(예를 들어 직선 위의 모든 점)
      → 유일한 해가 존재하지 않음
      → 역행렬(완벽한 undo 연산자) 연산을 할 수 없음
      → 역행렬이 없음
- 넓이 또는 부피
- 선형 변환을 할 때, 공간을 '확장' 또는 '축소'하게 되는데, 두 벡터로 형성된 영역이 선형 변환에 따라 얼마나 변하는가를 의미미
    - 2차원 면적, 3차원 이상일 경우 부피
    - 단순 전단이나 회전은 면적이 변하지 않으므로 행렬식에 영향이 없음
- 예시
    - $2\hat{i}$과 $3\hat{j}$

    - 이 변환은 면적을 6배 증가 시킴, $\det(A)$
- 행렬식은 변환이 선형 종속인지 알려줌
    - 행렬식이 0이면, 공간의 면적이 없어지고 더 작은 차원으로 축소되었음을 의미
    - 2차원 → 1차원, 3차원 → 2차원
- 공간을 얼마나 쥐어짜거나 늘리는지
    - det=1: 원래 부피 유지
    - det=0: 정보 손실 있음 (데이터가 눌림)
    - ｜det｜>1: 데이터가 확장됨

항등 행렬(Identity Matrix)

- 곱해서 자기자신을 결과로 출력하게 하는 행렬
- 대각선의 값이 1이고 다른 값은 0인 정사각 행렬(가로 세로 길이 동일일)
- $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- 항등 행렬을 만들 수 있다면 변환을 취소하고 원래 기저 벡터를 찾았다는 의미

역행렬(Inverse Matrix)

- 행렬의 변환을 취소, A^{-1}
- A^{-1}과 A를 곱셈하면 항등 행렬이 됨

역행렬을 구하는 원리는?

- 여러가지 방법이 있지만 가우스 조던 소거법으로 구할 수 있다.
    - 정사각행렬 A의 오른쪽에 항등 행렬 I를 붙여 확장 행렬 $[A \mid I]$을 만듦
      $\rightarrow$ 가우스-조던 소거법으로 A를 항등 행렬 I로 변경
      $\rightarrow$ 원래 I였던 부분은 A^{-1}로 변함
      $\rightarrow$ $[I \mid A^{-1}]$ 가 만들어짐

역행렬을 구하는 시간복잡도는?

- 만약 가우스 조던 소거법을 이용한다면, 빅O 노테이션으로 n^3이 되는 것으로 알고 있습니다.

고유값(eigenvalue), 고유 벡터(eigenvector)

- 행렬 A와 열벡터 x를 곱하는 행위를 선형 변환이라고 할 수 있는데, 선형 변환 후 방향이 변하지 않는 벡터를 아이겐벡터라고 한다.
    - 이 때, 크기가 변한 정도를 고유값이라고 합니다.
- $Ax = \lambda x$
    - λ: 고유값
    - 벡터 x: 고유벡터
- 고유값을 구하는 방법?

- $det(A - \lambda I) = 0$ 을 만족하는 λ의 값 찾기
- 고유벡터를 구하는 방법?
  - $(A - \lambda I)x = 0$ 을 만족하는 벡터 x 찾기
  - 즉, $(A - \lambda I)x$의 Null Space(영공간) 찾기

고유값 분해(Eigendecomposition)

PCA 란?

- Principal Component Analysis의 준말로, 우리 말로는 주성분 분석입니다.
- 고차원 데이터를 더 낮은 차원으로 줄이면서 핵심 정보(분산이 큰 방향)를 최대한 유지하는 차원 축소(dimensionality reduction) 기법
- 용량이 큰 데이터를 처리할 때, 차원을 축소하면 불필요한 노이즈를 제거하고 메모리 사이즈를 줄여 처리 속도를 향상하는 데 유용합니다.
- 방법은 데이터의 분포를 가장 잘 설명하는 축을 찾고, 그 축으로 데이터를 정사영 내리는 것
- 데이터의 분포를 가장 잘 설명하는 축은 공분산 행렬(코베리언스 메트릭스)의 고유벡터(아이겐벡터)
  - 고유값이 큰 방향이 가장 정보가 많은 주성분

SVD 란?

- Singular Value Decomposition, 특이값 분해
- 모든 실수 행렬 $A \in \mathbb{R}^{m \times n}$ 에 대해 다음과 같은 분해가 존재
  - $A = U \sum V^T$
  - U: 좌측 직교 행렬(열: A의 열공간 기저), m x m
  - $\sum$: 특이값을 포함하는 대각 행렬, m x n
  - V^T: 우측 직교 행렬(열: $A^T A$의 고유벡터), n x n
- $Ax = U \sum V^T x$
  - **기준 변경** $\rightarrow$ **크기 조정** $\rightarrow$ **다시 회전**
  - x를 V^T로 회전하여 기준 축 변경
  - $\sum$로 각 축을 스케일
  - U로 다시 회전하여 출력 공간으로 이동
- 왜 사용하는가?
  - 고유값 분해(Eigendecomposition)는 정방행렬에서만 가능하고, 반드시 대각화 가능한 경우에만 동작
  - SVD는 정방이든 아니든, 대칭이든 아니든 관계없이 모든 실수 행렬에 대해 항상 존재 → 가장 보편적인 행렬 분해 방법
  - 분해하는 이유는? 큰 특이값(top-k singular values)만 남기고 나머지를 제거함으로써 데이터의 핵심적인 구조만 유지한 채 차원을 줄일 수 있습니다. 이는 PCA와 유사한 방식으로 노이즈 제거와 연산 효율 개선에 사용

5장 선형 회귀(Linear Regression)

개념

- **하나 이상의 독립 변수(x)**를 이용해 **종속 변수(y)**를 예측하는 통계적 기법
- 예측값과 실제값 사이의 오차를 최소화하는 직선 또는 평면(고차원일 경우)을 찾는 것이 핵심
- 데이터를 가장 잘 대변하는 최적의 선을 찾은 과정
- 주로 예측 알고리즘에서서 활용

수식

- 단순 선형회귀(독립변수 1개)
  - $y = wx + b$
  - y: 예측값, x: 입력값(독립변수), w: 기울기 (slope, 계수), b: 절편 (bias, y절편)
- 다중 선형회귀(독립변수 여러개)
  - $y = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + \cdots + w_nx_n$
  - 계산 시 행렬을 이용

과정

- 주어진 데이터에 대해 오차(잔차)의 제곱합이 최소가 되도록 w와 b를 찾는 것
- $Loss(MSE) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$

6장 로지스틱 회귀

개념

- 입력값에 대해 0 또는 1 (혹은 두 클래스 중 하나)에 속할 확률을 예측
- 분류(Classification) 알고리즘으로, 특히 이진 분류(Binary Classification) 문제에서 활용

수식

- $\sigma(z) = \frac{1}{1+e^{-z}}$
- $z = w^Tx + b$
- $\sigma(z) \in (0, 1)$: 항상 0과 1사이의 실수값을 출력

특징

- 입력이 클수록 ($\rightarrow +\infty$) 출력은 1에 가까워짐
- 입력이 작을수록 ($\rightarrow -\infty$) 출력은 0에 가까워짐
- z = 0일 때, $\sigma(0) = 0.5$

과정

- 분류로 해석

    - $\hat{y} = \sigma(w^T x + b)$ 가 1로 분류될 확률은?
    - $\hat{y} = 0.8$ 이면, "1로 분류될 확률이 80%"
    - $\hat{y} = 0.3$ 이면, "1로 분류될 확률이 30%, 0으로 분류될 확률이 더 높아"
- 최종 예측 결정
    - 보통은 임계값(보통 0.5)을 기준으로 분류
    - $\hat{y} >= 0.5$이면, 클래스 1(Positive)
    - $\hat{y} < 0.5$이면, 클래스 0(Negativ)

7장 신경망