기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 02. 머신러닝 기초와 배경

정 정 민

programmers

기본 개념과 용어 정리

선형대수??

- 수 들이 모여있는 개념과 관련된 식을 연구하는 수학의 한 분야로
  - 수가 모여 있다는 것은 벡터 혹은 행렬이라고 함 등
  - 뒷 장표에서 자세히 설명
- 수를 다루는 많은 학문(데이터, 공학, 과학적 분석 등)에서 수의 연산을 빠르고 효과적으로 하기 위해 사용하는 도구

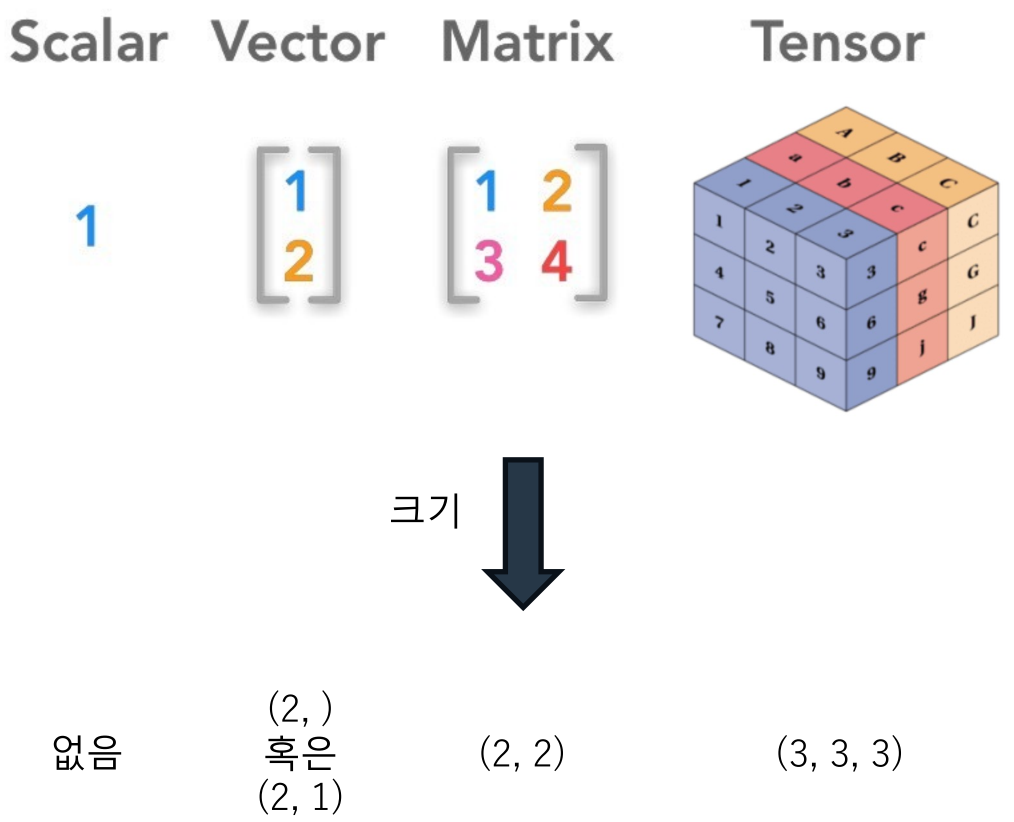
- 수의 집합을 기하학적인 형상으로 적용하여 표현
  - 시각적이고 직관적으로 수의 값을 이해할 수 있음
  - 기하학적으로 의미를 갖는 다양한 변환을 수학적으로 정의할 수 있음 (회전, 스케일링 등)

- 컴퓨터 그래픽스, 엔지니어링, 물리학, 컴퓨터 과학, 머신러닝 등 다양한 분야에서 응용됨
- 특히 머신러닝에서는 데이터를 표현하고 변환하는데 필수적인 도구로 사용!

수의 집합 : 스칼라, 벡터, 행렬, 텐서

- 숫자는 특정한 방향으로 줄을 서듯 모일 수 있음
  - 방향을 보통 차원이라고 부르며
  - 숫자들이 얼마나 모이는지에따라 크기가 정해짐

- **스칼라** : 다른 숫자와 함께하지 않고 홀로 존재하는 수
- **벡터** : 한쪽 방향(차원)으로만 숫자가 모인 형태 → 1차원
- **행렬** : 두 방향으로 숫자가 줄을 선 형태 → 2차원
- **텐서** : 벡터와 행렬을 일반화한 개념.
  - 0차원 텐서를 스칼라라고
  - 1차원 텐서를 벡터라고
  - 2차원 텐서를 행렬이라고 부르며
  - 3차원 이상의 수 집합을 나타내는 용어이기도 함



https://www.linkedin.com/pulse/machine-learning-basics-scalars-vectors-matrices-tensors-prasad/
https://dev.to/mmithrakumar/scalars-vectors-matrices-and-tensors-with-tensorflow-2-0-1f66

데이터를 벡터와 행렬로

- 한 친구 A의 정보가 다음과 같음
    - 출석 번호 : 3 / 혈액형 : O / MBTI : ENTP / 키 : 163 / 중간고사 평균 : 92 / 기말고사 평균 : 96
- 이 친구의 정보는 아래와 같이 벡터로 나타낼 수 있음
    - 숫자와 글자를 모두 받는다고 가정
    - [3, O, ENTP, 163, 92, 96]
    - 총 6의 크기를 갖는 vector로 변환 가능

- 여기서 이 친구 학급에 있는 모든 친구들의 정보를 다 이렇게 변환하면
    - [[3, O, ENTP, 163, 92, 96],
      [4, A, INTP, 168, 89, 88],
      …
      [10, B, ESTJ, 155, 88, 82],
      [11, AB, ENFP, 170, 90, 91]]

- 이처럼 데이터를 벡터 혹은 행렬의 형태로 변환 가능

행렬 연산과 성질

행렬의 덧셈과 뺄셈

- 행렬도 수의 집합이므로 연산이 가능
  - 행렬 뿐 아니라, 벡터 그리고 나아가 모든 종류의 텐서에 적용 가능

- 행렬의 덧셈과 뺄셈은 같은 크기의 행렬끼리만 가능
- 각 행렬의 같은 자리에 있는 원소끼리의 덧셈과 뺄셈
  - 이렇게 같은 자리에 있는 원소끼리의 연산을 element-wise operation이라고 함

- $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} + \begin{bmatrix}5\\6\end{bmatrix} =$ 연산 안됨

- $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} + \begin{bmatrix}5 & 6\\7 & 8\end{bmatrix} = \begin{bmatrix}6 & 8\\10 & 12\end{bmatrix}$
  - Element-wise sum!

행렬의 곱셈

- 행렬의 곱셈은 일반적인 수의 곱과는 다름

- 행렬의 곱은 하나의 행렬의 각 행과 다른 행렬의 각 열 간의 내적을 의미함
  - 내적이란, 벡터 간의 연산을 의미함
  - 두 벡터의 동일한 위치에 있는 원소를 곱한 후, 그 결과를 모두 더하는 연산 (결과는 스칼라)
    - $a = [1\ 2\ 3]$ / $\mathrm{b} = [4\ 5\ 6]$
    - $a \cdot b = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 32$

- Element-wise 연산이 아니므로 행렬의 크기가 달라도 연산이 가능
- 대신, 앞선 행렬의 열과 뒷 행렬의 행의 크기가 같아야 함

$$\begin{array}{cc} & n \\ m & \begin{bmatrix} \quad A \quad \end{bmatrix} \end{array} \times \begin{array}{cc} & k \\ n & \begin{bmatrix} \quad B \quad \end{bmatrix} \end{array} = \begin{array}{cc} & k \\ m & \begin{bmatrix} \quad AB \quad \end{bmatrix} \end{array}$$

우리가 아는 수의 곱셈은 뭘까?

- 같은 크기의 행렬에서 각 요소 별로 곱하기하는 연산이 존재함
  - Element-wise multiplication

- 이를 Hadamard Product 라고 함
- 예를 들어
  - $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \circ \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}$

전치 행렬 (Transpose)

- 하나의 행렬이 주어질 때, 행과 열을 바꾼 행렬을 전치 행렬이라고 함

- A 행렬의 크기가 $m \times n$ 이라면
- A 행렬의 전치 행렬은 A^{T} 으로 표기하고, 그것의 크기는 $n \times m$ 이 됨

- 예를 들어
- $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 의 전치 행렬은 $A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$

- 대각선 원소는 전치 과정에서 그대로 유지됨
- 만약 $A = A^T$ 인 경우, A 를 대칭 행렬이라고 함

- 또한, 곱셈의 경우 순서가 바뀜
- $(AB)^T = B^T A^T$

역행렬 (Inverse Matrix)

- 특정 행렬 A에 어떤 행렬 B를 곱해보니 결과가 항등 행렬 $I = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & 1 & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$ 이라면
- 여기서 B를 A의 역행렬이라고 함

- 모든 행렬이 역행렬을 갖는것은 아님
- A 행렬이 역행렬을 가지려면, A는 반드시 가역(또는 비특이, non-singular) 해야 함
    - 이를 수학적으로 표현하면 $\det(A) \neq 0$ 을 만족해야 함
    - 방정식을 비유적으로 생각하면 $ax = 1$ 을 만족하는 x를 찾고자 한다면 $a \neq 0$ 이 아니어야 함!
    - '가역적'이라는 표현은 원래 상태(I)로 돌릴 수 있음을 의미함

- 아래와 같은 성질이 있음
    - $(A^{-1})^{-1} = A$
    - $(AB)^{-1} = B^{-1}A^{-1}$
    - $(A^T)^{-1} = (A^{-1})^T$

고유벡터와 고유값

선형 변환 (Linear Transformation)

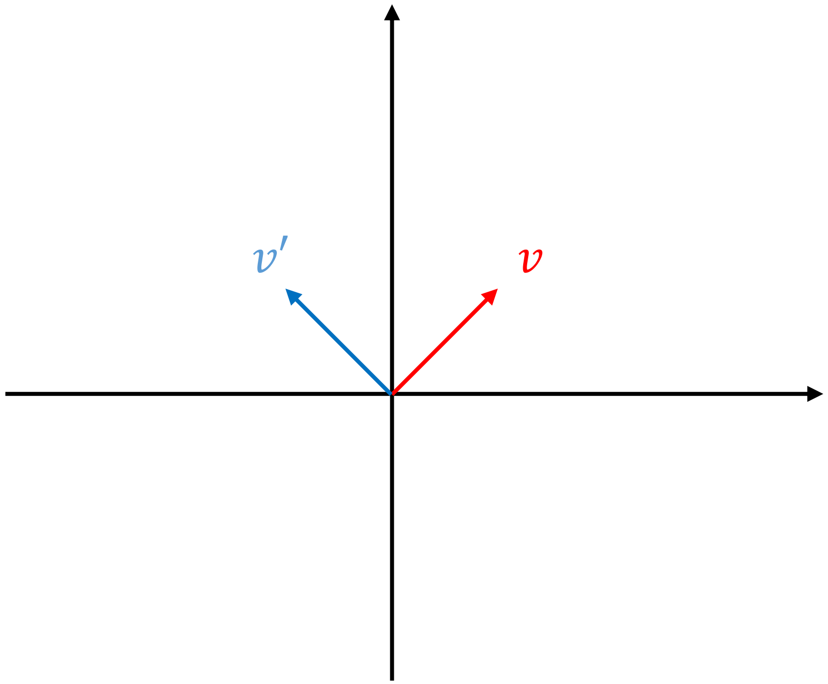
- 어떤 벡터(v)가 있다고 해볼까요?
- 이 벡터는 벡터 크기 만큼의 차원 공간에 존재

- 여기에 특정 행렬(A)를 곱해서 새로운 벡터 (v')을 만들었다고 가정

- v와 v'는 A 행렬에 의해 방향이 바뀜

- 예를 들어..
  - $v = [1 \quad 1]$, $A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ 이라면 $Av = \begin{bmatrix} -1 \\ 1 \end{bmatrix} = v'$

- 즉,
  - 특정 벡터에 어떠한 행렬을 곱하면
  - 벡터의 방향 혹은 크기가 변경
  - 이렇듯 벡터의 방향과 크기의 변경을 선형 변환이라고 함

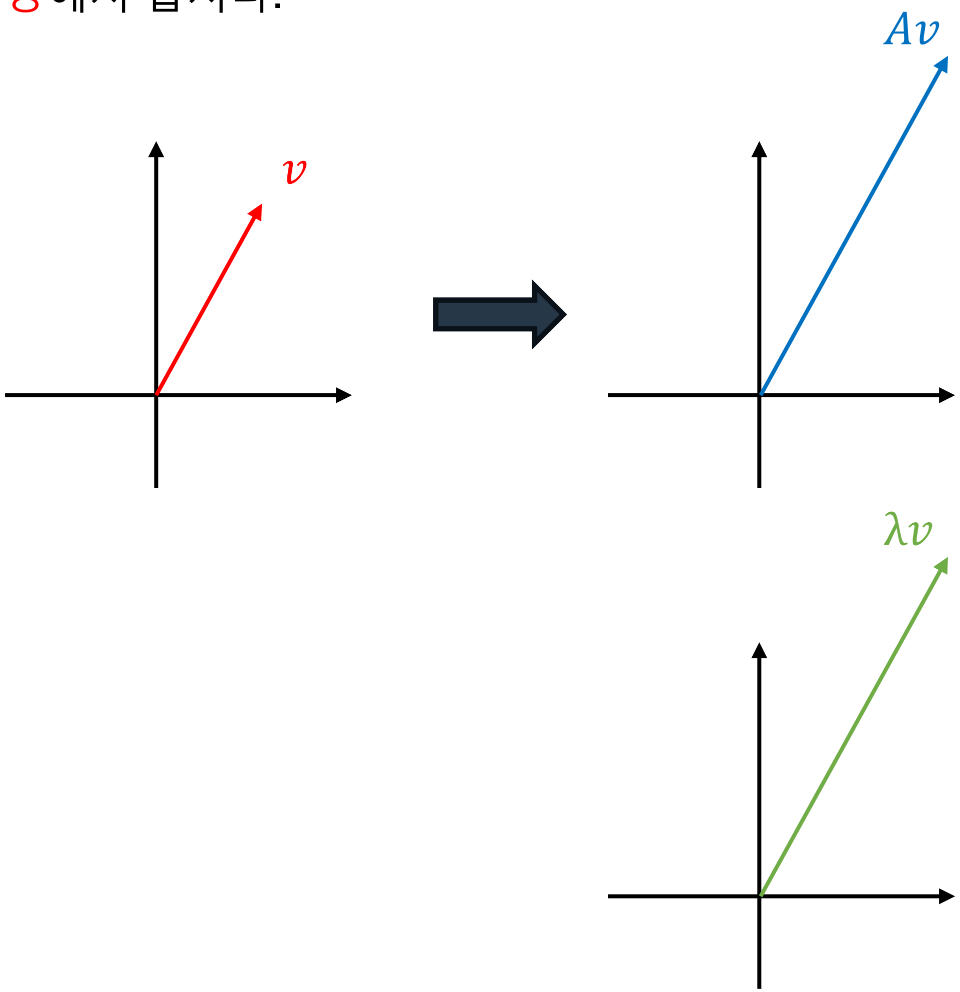


고유벡터(Eigenvector)와 고유값 (Eigenvalue)

- 이제는 관점을 좀 바꿔볼까요?
- 앞서서는 벡터(v)를 기준으로 살펴봤는데, 이번에는 특정 행렬(A)의 입장에서 봅시다.

- 행렬 A에는 다양한 벡터를 곱할 수 있음
- 그러다 보면 아래와 같은 특징을 갖는 벡터 존재할 수 있음

$$Av = \lambda v$$



- 여기서 λ : 임의의 상수
- 이를 해석하면
  - A 행렬에 임의의 벡터 v를 곱하니, 그 결과가
  - 벡터 v의 크기를 상수 λ배 한 벡터와 같다!

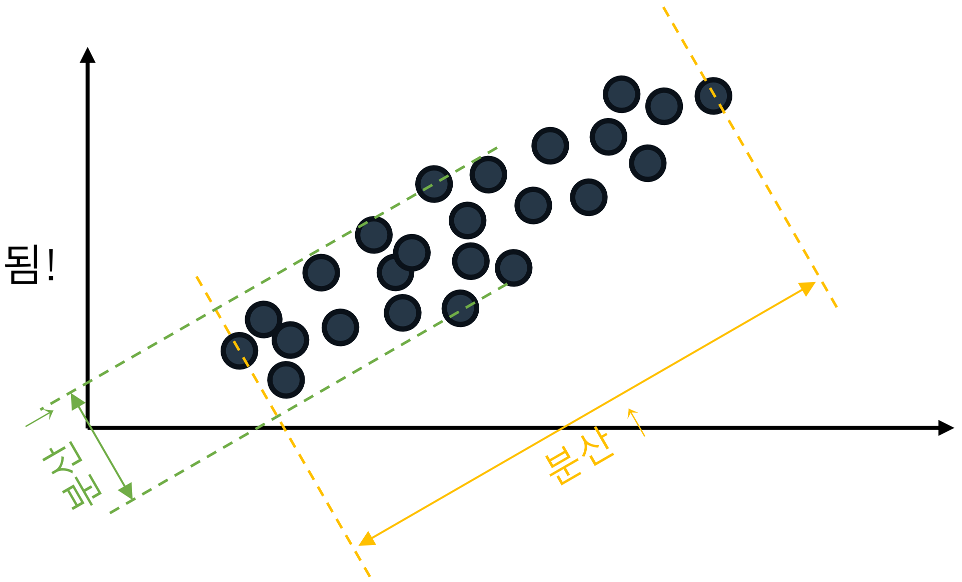
- 그런 벡터 v를 고유벡터(Eigenvector)라고 하고
- 그런 상수 λ를 고유값(Eigenvalue)라고 함

뭔진 알았는데... 그걸 왜 & 어디에 쓰나요??

- 행렬 A의 고유벡터는 행렬 A의 값이 가장 많이 분산되는 방향을 나타냄
- 분산이 많이 된다는 것은 많은 정보력을 갖고 있다고 볼 수 있음

- 일반적으로 데이터를 불러오면 행렬의 형태를 갖게 됨
- 이 데이터를 담아온 행렬을 A라고 보면,
- 데이터가 담고 있는 여러 정보 중 가장 의미가 큰 방향이 고유 벡터가 됨!

- 해당 방향으로 얼만큼 분산이 이루어졌는지
- 분산의 크기를 나타내는 정도가 고유값



- 이러한 고유벡터와 고유값은 복수개가 가능하며
- 고유값을 기준으로 나열된 고유벡터는 해석력이 큰 방향의 순서를 의미함

- 이 둘은 데이터를 이해하고 계산하는 과정에서 사용됨
  - 의미를 유지한 상태로 데이터를 전처리 하거나
  - 행렬 계산을 간소화 하는 과정에서 사용됨

특이값 분해

특이값 분해 (Singular Value Decomposition, SVD)

- 중학교 수학에서 소인수 분해 기억 나시나요?
  - $30 = 2 \times 3 \times 5$
  - 즉, 복잡한 수(30)를 그 수를 구성하는 기본적인 블록(소수 2,3,5)으로 분해

- 행렬에도 이와 비슷한 과정이 있음
  - 복잡한 행렬 A $(m \times n)$을 더 간단한 세 가지 행렬(U, Σ, V^T)로 분해
  - $A = U\Sigma V^T$
    - U의 열 벡터들은 A의 왼쪽 특이 벡터로 AA^T의 고유 벡터
    - V의 열 벡터들은 A의 오른쪽 특이 벡터로 A^TA의 고유 벡터
    - Σ의 대각선 위의 값들로 A의 특이값
      - 원래 행렬 A의 데이터가 가진 중요도를 의미

- 이런 과정을 특이값 분해(SVD)라고 함

"고유벡터와 고유값" 그리고 SVD의 관계

- 둘 다 특정 행렬 A에서 정보를 뽑아내는 과정

- 하지만 고유벡터와 고유값 분석은 행렬 A가 정사각 행렬일 경우에 사용 가능
- 반면, SVD는 직사각 행렬 A에 대해 사용 가능

- 따라서 SVD가 조금 더 일반적인 경우를 나타내며
- 고유벡터와 고유값은 SVD의 스페셜 케이스로 보면 됨

	SVD	고유벡터와 고유값
$A\ (m \times n)$ 직사각 행렬	사용 가능	사용 불가
$A\ (m \times m)$ 정사각 행렬	사용 가능	사용 가능

AA^T는 무슨 의미인가요??

- A 행렬의 크기가 $(m \times n)$ 이라고 할 때,
- 정사각 행렬이 아니므로 바로 고유벡터와 고유값 분석을 할 수 없음

- 따라서 행렬 A에서 행과 열 방향으로 나눠 따로
- 행 사이의 관계를 따로 보고 열 사이의 관계를 따로 보고자 함

- AA^T는 $(m \times m)$ 의 크기를 갖고 있어 고유벡터와 고유값 분석이 가능
- AA^T는 원래 행렬 A의 행 사이의 관계도가 데이터의 형태로 존재하며
- 그들의 정보력 중 분산이 크고 중요한 의미를 갖는 방향 벡터가 U 행렬 안에 정리될 것

- A^TA도 마찬가지
- 단 원본 행렬 A의 열 사이의 관계를 바탕으로 고유값 및 고유벡터 분석 진행

마지막 Σ

- AA^T 혹은 A^TA 의 고유값의 제곱근 값을 A의 특이값이라고 함
  - AA^T 의 고유값과 A^TA 의 고유값은 서로 같음

- Σ 행렬은 A 의 특이값을 대각선 위치에 갖고 있고
- 대각선을 제외한 나머지 모든 값은 0

- 고유값 분석과 마찬가지로
- 행렬 A의 선형 변환에서 중요한 스케일링 정보를 포함하고 있음

정리하면.. $\boldsymbol{m \times n}$ 크기를 갖는 행렬 $\boldsymbol{A}$는

$$A = U\Sigma V^T$$

- **$\boldsymbol{U}$ (왼쪽 특이 벡터들)**
  - 크기 : $m \times m$
  - 원본 행렬 A의 행 정보를 바탕으로 중요도를 파악
- **$\boldsymbol{V}$ (오른쪽 특이 벡터들)**
  - 크기 : $n \times n$
  - 원본 행렬 A의 열 정보를 바탕으로 중요도를 파악
- **$\boldsymbol{\Sigma}$ (특이값들)**
  - 크기 : $m \times n$
  - 행렬 A 선형 변환 과정에서 영향을 미치는 스케일링 정보(특이값)를 포함

E.O.D