변환(Transformation)이란?

변환(Transformation)은 사실 **함수(Function)** 의 다른 이름입니다:

- 입력을 받아서 출력을 내보냅니다
- 선형대수에서는: 벡터를 입력받아 다른 벡터를 출력

왜 "함수" 대신 "변환"이라고 부를까?

"변환"이라는 단어는 **움직임**을 암시합니다:

- 입력 벡터가 출력 벡터로 "이동"한다고 상상
- 공간 전체가 변형되는 것으로 시각화
- 모든 점이 동시에 움직이는 모습

시각화: 공간이 변형되는 것으로 생각하기

변환을 이해하는 가장 좋은 방법:

1. **격자(Grid) 그리기**
   - 2D 평면에 균등한 간격의 수평선과 수직선
   - 좌표를 나타내는 격자
2. **변환 관찰**
   - 격자의 모든 점이 어떻게 움직이는지 관찰
   - 격자선이 어떻게 휘고, 회전하고, 늘어나는지 관찰
   - 공간 전체가 "찌그러지고 변형"되는 느낌
3. **특정 벡터 추적**
   - 입력 벡터의 끝점이 어디로 가는지 추적
   - 그것이 출력 벡터

선형변환 (Linear Transformation)

임의의 변환은 매우 복잡할 수 있습니다. 다행히 **선형대수는 특별한 종류의 변환만 다룹니다.**

선형변환의 두 가지 핵심 성질:

1. **모든 직선은 직선으로 유지**
   - 직선이 휘어지거나 곡선이 되면 안 됨
   - 격자선이 평행하고 균등하게 유지
2. **원점은 고정**
   - 원점 (0, 0)이 움직이면 안 됨

비선형 변환 예시 (선형이 아님):

- 직선을 곡선으로 만드는 변환
- 원점을 움직이는 변환

선형변환의 결과:

- 격자선이 평행하고 균등하게 유지됨
- 수직선이 수직이 아닐 수 있음 (각도는 변할 수 있음)
- 하지만 평행성과 균등한 간격은 유지

선형변환을 수치적으로 표현하기

핵심 통찰: 선형변환을 완전히 설명하려면 **기저벡터가 어디로 가는지만** 알면 됩니다!

왜 기저벡터만으로 충분한가?

2차원의 경우, 모든 벡터는 기저벡터의 선형결합입니다:

```
[x] = x·î + y·ĵ
[y]
```

선형변환의 성질:

- 선형결합을 보존합니다
- 즉, 변환 전후에 선형결합 관계가 유지됨

수학적으로:

```
변환 전: v = x·î + y·ĵ

변환 후: 변환된 v = x·(변환된 î) + y·(변환된 ĵ)
```

의미:

- î가 어디로 가는지 알고
- ĵ가 어디로 가는지 알면
- 모든 벡터 [x, y]가 어디로 가는지 계산 가능!

행렬의 등장

변환된 기저벡터의 좌표를 정리하는 방법:

2×2 행렬:

```
[a  b]
[c  d]
```

여기서:

- **첫 번째 열 [a, c]**: 변환 후 î의 좌표
- **두 번째 열 [b, d]**: 변환 후 ĵ의 좌표

행렬은 선형변환의 완전한 설명입니다!

행렬-벡터 곱셈

벡터 [x, y]에 변환을 적용하는 방법:

```
[a  b] [x]   [ax + by]
[c  d] [y] = [cx + dy]
```

기하학적 의미:

```
x·[a] + y·[b] = [ax + by]
  [c]     [d]   [cx + dy]
```

- x배 스케일된 (변환된 î)
-
    - y배 스케일된 (변환된 ĵ)

핵심:

- 좌표 [x, y]는 기저벡터의 스케일 명령
- 변환 후에도 같은 스케일 명령을 사용
- 하지만 기저벡터가 바뀌었으므로 결과가 다름

구체적 예제들

예제 1: 90도 반시계 회전

변환 효과:

- î = [1, 0] → [0, 1]
- ĵ = [0, 1] → [-1, 0]

행렬:

```
[0  -1]
[1   0]
```

벡터 [3, 2] 적용:

```
[0  -1] [3]   [0·3 + (-1)·2]   [-2]
[1   0] [2] = [1·3 + 0·2   ] = [3 ]
```

결과: [3, 2] → [-2, 3]

예제 2: Shear (전단) 변환

변환 효과:

- î = [1, 0] → [1, 0] (그대로)
- ĵ = [0, 1] → [1, 1] (오른쪽으로 밀림)

행렬:

```
[1  1]
```

```
[0  1]
```

의미:

- y좌표만큼 오른쪽으로 밀어냄
- 평행사변형으로 찌그러뜨림

예제 3: x축으로 투영

변환 효과:

- î = [1, 0] → [1, 0] (그대로)
- ĵ = [0, 1] → [0, 0] (원점으로)

행렬:

```
[1  0]
[0  0]
```

의미:

- 2D 평면을 1D 직선(x축)으로 압축

선형독립과 차원 축소

기저벡터가 선형종속이 되면?

변환 후 î와 ĵ가 같은 직선 위에 있으면:

- 2D 전체 공간이 1D 직선으로 압축됨
- 정보 손실 발생

예:

```
î → [1, 2]
ĵ → [2, 4]  (= 2 × î의 변환)

행렬: [1  2]
      [2  4]
```

모든 벡터가 한 직선 위로만 변환됩니다.

선형변환의 종류들

1. 회전 (Rotation)

```
[cos(θ)  -sin(θ)]
[sin(θ)   cos(θ)]
```

- 원점을 중심으로 θ만큼 회전
- 거리와 각도 보존

2. 스케일링 (Scaling)

```
[s  0]
[0  s]
```

- 모든 방향으로 s배 확대/축소
- s > 1: 확대
- 0 < s < 1: 축소

3. 반사 (Reflection)

```
[-1  0]  (y축 대칭)
[0   1]
```

4. 전단 (Shear)

```
[1  k]
[0  1]
```

- 한 방향으로 밀어냄
- 평행사변형으로 변형

암기 vs 이해

나쁜 접근:

```
[a  b] [x]   [ax + by]
[c  d] [y] = [cx + dy]
```

이 공식을 단순 암기

좋은 접근:

1. 행렬의 열 = 변환된 기저벡터
2. [x, y] = x·î + y·ĵ (변환 전)
3. 변환 후 = x·(변환된 î) + y·(변환된 ĵ)
4. 자연스럽게 행렬-벡터 곱셈 공식 도출

이해의 힘:

- 공식을 외울 필요 없음
- 왜 그렇게 되는지 설명 가능
- 다른 상황에 응용 가능

선형변환의 핵심 직관

1. **행렬 = 변환**
    - 행렬은 단순한 숫자 배열이 아님
    - 공간을 변형시키는 방법의 설명

2. **열 = 변환된 기저벡터**
   - 첫 번째 열: î가 어디로 가는가
   - 두 번째 열: ĵ가 어디로 가는가
3. **곱셈 = 적용**
   - 행렬-벡터 곱셈 = 변환 적용
   - 기하학적 의미: 공간 변형 후 벡터의 새 위치
4. **선형성 = 제약**
   - 직선 유지, 원점 고정
   - 이 제약 덕분에 기저벡터만으로 모든 것 결정

3차원으로의 확장

3×3 행렬:

```
[a  b  c]
[d  e  f]
[g  h  i]
```

- 첫 번째 열: 변환된 î
- 두 번째 열: 변환된 ĵ
- 세 번째 열: 변환된 k̂

행렬-벡터 곱셈 (3D):

```
[a  b  c] [x]   [ax + by + cz]
[d  e  f] [y] = [dx + ey + fz]
[g  h  i] [z]   [gx + hy + iz]
```

같은 논리:

- x·(변환된 î) + y·(변환된 ĵ) + z·(변환된 k̂)

왜 이것이 중요한가?

이 개념이 선형대수의 나머지를 이해하는 열쇠입니다:

- 행렬 곱셈 (다음 장): 변환의 합성
- 행렬식 (6장): 변환이 공간을 얼마나 확대/축소하는가
- 역행렬 (7장): 변환을 되돌리기
- 고유벡터 (14장): 변환 후에도 방향이 유지되는 벡터
- 등등...

모든 것이 "공간의 변형"이라는 시각적 직관으로 연결됩니다.

---

2. 퀴즈

문제 1

"변환(transformation)"과 "함수(function)"는 본질적으로 같은 의미이지만, 왜 선형대수에서는 "변환"이라는 용어를 선호하는지 설명하시오. 그리고 변환을 시각화할 때 격자(grid)를 사용하는 이유를 서술하시오.

문제 2

선형변환의 두 가지 핵심 성질을 설명하고, 다음 변환이 선형인지 비선형인지 판단하시오: (a) 모든 벡터를 2배로 늘린 후 오른쪽으로 3만큼 평행이동 (b) 원점을 중심으로 45도 회전 (c) $y = x^2$ 곡선을 따라 변형

문제 3

다음 행렬이 주어졌을 때:

```
[2  1]
[0  3]
```

(a) 이 행렬이 나타내는 선형변환에서 기저벡터 $\hat{\imath}$와 $\hat{\jmath}$가 각각 어디로 변환되는지 쓰시오. (b) 벡터 [4, 5]를 이 변환에 적용한 결과를 계산하시오. (c) 계산 과정을 "변환된 기저벡터의 선형결합"으로 해석하여 설명하시오.

문제 4

선형변환에서 "기저벡터가 어디로 가는지만 알면 모든 벡터의 변환을 알 수 있다"는 원리를 설명하시오. 이것이 왜 선형변환에서만 성립하는지, 선형변환의 어떤 성질 때문인지 논리적으로 서술하시오.

문제 5

다음 행렬을 고려하시오:

```
[3  6]
[1  2]
```

이 행렬이 나타내는 변환의 기하학적 특성을 설명하고, 2차원 평면이 어떻게 변형되는지 서술하시오. 특히 두 번째 열이 첫 번째 열의 스칼라 배수라는 사실이 변환에 어떤 영향을 미치는지 논하시오.

---

3. 퀴즈 답

문제 1 답안

"변환"이라는 용어를 선호하는 이유:

"함수"와 "변환"은 수학적으로 같은 개념(입력→출력 매핑)이지만, "변환"이라는 단어는 **움직임(movement)** 을 암시합니다:

1. **시각적 직관 제공:**
   - "함수"는 정적인 느낌 (단순히 값을 계산)
   - "변환"은 동적인 느낌 (공간이 변형되고 움직임)
   - 입력 벡터가 출력 벡터로 "이동"한다고 상상하게 함
2. **전체 공간의 관점:**
   - 개별 벡터가 아니라 **공간 전체**가 변형된다고 생각
   - 모든 점이 동시에 움직이는 것으로 시각화

- 찌그러지고, 회전하고, 늘어나는 공간의 모습

격자를 사용하는 이유:

1. **좌표 시스템 시각화:**
   - 수평선과 수직선이 좌표축을 나타냄
   - 격자점이 정수 좌표를 표시
2. **변형 관찰:**
   - 변환 전: 균등하고 평행한 격자
   - 변환 후: 격자가 어떻게 휘고, 회전하고, 늘어나는지 명확히 보임
   - 직선이 유지되는지, 평행성이 보존되는지 쉽게 확인
3. **선형성 확인:**
   - 선형변환: 격자선이 직선이고 평행하게 유지
   - 비선형변환: 격자선이 곡선이 되거나 평행성 상실
4. **전체 효과 파악:**
   - 개별 벡터 몇 개만으로는 전체 변환을 이해하기 어려움
   - 격자 전체를 보면 공간이 어떻게 변형되는지 한눈에 파악

결론: "변환"이라는 용어와 격자 시각화는 선형대수를 **기하학적으로 이해**하는 데 핵심적인 도구입니다.

문제 2 답안

선형변환의 두 가지 핵심 성질:

1. **모든 직선은 직선으로 유지 (Lines remain lines):**
   - 변환 전에 직선이었던 것은 변환 후에도 직선
   - 직선이 휘어지거나 곡선이 되면 안 됨
   - 결과적으로: 격자선이 평행하고 균등한 간격 유지
2. **원점은 고정 (Origin remains fixed):**
   - 점 (0, 0)은 항상 (0, 0)으로 변환
   - 원점이 다른 곳으로 이동하면 선형변환이 아님

각 변환의 판단:

(a) 모든 벡터를 2배로 늘린 후 오른쪽으로 3만큼 평행이동

비선형입니다.

이유:

- "오른쪽으로 3만큼 평행이동"이 문제
- 원점 [0, 0]이 [3, 0]으로 이동
- **성질 2 위반** (원점이 고정되지 않음)

수식으로:

```
[x] → 2[x] + [3] = [2x + 3]
[y]     [y]   [0]   [2y    ]
```

[0, 0] → [3, 0] ≠ [0, 0] ✗

(b) 원점을 중심으로 45도 회전

선형입니다.

이유:

- 모든 직선이 직선으로 유지 (회전은 직선을 휘지 않음)
- 원점은 회전 중심이므로 고정
- **성질 1, 2 모두 만족** ✓

행렬 표현:

```
[cos(45°)  -sin(45°)]   [√2/2  -√2/2]
[sin(45°)   cos(45°)] = [√2/2   √2/2]
```

(c) $y = x^2$ 곡선을 따라 변형

비선형입니다.

이유:

- 직선이 포물선으로 휘어짐
- 예: x축 ($y = 0$인 직선) → $y = x^2$ 곡선
- **성질 1 위반** (직선이 유지되지 않음)

정리: 선형변환은 매우 제한적입니다:

- 평행이동(translation) ✗
- 포물선 변형 ✗
- 회전 ✓
- 스케일링 ✓
- 반사 ✓
- 전단(shear) ✓

문제 3 답안

주어진 행렬:

```
[2  1]
[0  3]
```

(a) 변환된 기저벡터

행렬의 열이 변환된 기저벡터입니다:

- **$\hat{i}$ = [1, 0]의 변환:**
  - 첫 번째 열 = [2, 0]
  - $\hat{i}$ → [2, 0]
- **$\hat{j}$ = [0, 1]의 변환:**
  - 두 번째 열 = [1, 3]
  - $\hat{j}$ → [1, 3]

(b) 벡터 [4, 5]의 변환

행렬-벡터 곱셈:

```
[2  1] [4]   [2×4 + 1×5]   [8 + 5]   [13]
[0  3] [5] = [0×4 + 3×5] = [0 + 15] = [15]
```

답: [4, 5] → [13, 15]

(c) 선형결합으로 해석

변환 전:

```
[4, 5] = 4·î + 5·ĵ
       = 4[1, 0] + 5[0, 1]
```

의미: "î를 4배, ĵ를 5배 스케일해서 더함"

변환 후:

선형변환은 선형결합을 보존하므로:

```
변환된 [4, 5] = 4·(변환된 î) + 5·(변환된 ĵ)
               = 4[2, 0] + 5[1, 3]
               = [8, 0] + [5, 15]
               = [13, 15]
```

해석:

1. 원래 벡터는 "î 4개 + ĵ 5개"
2. 변환 후에도 여전히 "새 î 4개 + 새 ĵ 5개"
3. 하지만 새 î = [2, 0], 새 ĵ = [1, 3]
4. 따라서 최종 위치 = [13, 15]

핵심:

- 스케일 명령 (4와 5)은 변하지 않음
- 기저벡터가 바뀌었으므로 결과가 달라짐
- 이것이 행렬-벡터 곱셈의 기하학적 의미

문제 4 답안

원리: "기저벡터의 변환만 알면 충분하다"

설명:

모든 2D 벡터는 기저벡터의 **선형결합**으로 표현됩니다:

```
v = [x, y] = x·î + y·ĵ
```

선형변환의 핵심 성질:

선형변환은 다음 두 가지를 보존합니다:

1. **덧셈 보존:**
   ```
   T(v + w) = T(v) + T(w)
   ```
2. **스칼라 곱셈 보존:**
   ```
   T(c·v) = c·T(v)
   ```

논리적 유도:

변환 T를 벡터 v = x·î + y·ĵ에 적용하면:

```
T(v) = T(x·î + y·ĵ)
     = T(x·î) + T(y·ĵ)          (덧셈 보존)
     = x·T(î) + y·T(ĵ)          (스칼라 곱셈 보존)
```

결론:

- T(î)만 알면 x·T(î) 계산 가능
- T(ĵ)만 알면 y·T(ĵ) 계산 가능
- 둘을 더하면 T(v) 계산 가능

왜 선형변환에서만 성립하는가?

비선형 변환의 반례:

예: T([x, y]) = [x^2, y^2] (제곱 변환)

```
T(î) = T([1, 0]) = [1, 0]
T(ĵ) = T([0, 1]) = [0, 1]
```

두 기저벡터가 그대로인데, 다른 벡터는?

```
v = [2, 3] = 2î + 3ĵ

만약 선형결합 보존이라면:
T(v) = 2T(î) + 3T(ĵ) = 2[1,0] + 3[0,1] = [2, 3]

실제로는:
T([2, 3]) = [4, 9] ≠ [2, 3]  ✗
```

비선형변환은 선형결합을 보존하지 않으므로, 기저벡터만으로는 불충분합니다!

정리:

선형변환만이 이 성질을 가지는 이유:

1. 선형결합 보존 = 덧셈과 스칼라 곱셈 보존
2. 모든 벡터 = 기저벡터의 선형결합
3. 따라서 기저벡터의 변환만 알면 모든 벡터 결정

4. 이것이 선형변환의 정의 그 자체

이 원리 덕분에 무한히 많은 벡터의 변환을 단 몇 개의 숫자(행렬의 원소)로 표현할 수 있습니다!

문제 5 답안

주어진 행렬:

```
[3  6]
[1  2]
```

기저벡터의 변환:

- î → [3, 1] (첫 번째 열)
- ĵ → [6, 2] (두 번째 열)

선형종속 관찰:

```
ĵ의 변환 = [6, 2] = 2 × [3, 1] = 2 × (î의 변환)
```

두 번째 열이 첫 번째 열의 **2배**입니다. 즉, 변환된 기저벡터들이 **선형종속**입니다.

기하학적 의미: 차원 축소

2D → 1D 압축:

1. **변환 전:**
   - 2차원 평면 전체
   - î와 ĵ가 서로 다른 방향
2. **변환 후:**
   - 모든 벡터가 한 직선 위로만 매핑됨
   - 그 직선: [3, 1] 방향의 직선

왜 직선인가?

임의의 벡터 [x, y]를 변환하면:

```
[3  6] [x]   [3x + 6y]   [3(x + 2y)]   [3]
[1  2] [y] = [1x + 2y] = [1(x + 2y)] = [1] · (x + 2y)
```

결과는 항상 벡터 [3, 1]의 스칼라 배수 형태입니다!

시각화:

- 2D 평면의 모든 점들이
- 원점을 지나고 [3, 1] 방향의 직선 위로
- "눌려서" 압축됩니다

정보 손실:

예를 들어:

- [1, 0] → [3, 1]
- [0, 0.5] → [3, 1]
- [2, -1] → [3, 1]

서로 다른 입력 벡터들이 같은 출력으로 변환됩니다. 원래 정보를 복원할 수 없습니다 (역변환 불가능).

결론:

두 번째 열이 첫 번째 열의 스칼라 배수라는 사실은:

1. **변환된 기저벡터가 선형종속** → 한 방향만 span
2. **2D 공간 전체가 1D 직선으로 압축** → 차원 축소
3. **정보 손실 발생** → 역변환 불가능 (행렬식 = 0)
4. **모든 벡터가 같은 직선 위로 매핑** → 제한된 출력 범위

이런 변환을 "특이(singular)" 변환이라고 하며, 나중에 배울 행렬식(determinant) 개념과 직접 연결됩니다. 행렬식이 0이면 차원이 축소됩니다.