Linear Algebra

Matrix Operation
(addition/subtraction, multiplication, linear equations)

소프트웨어 꼰대 강의

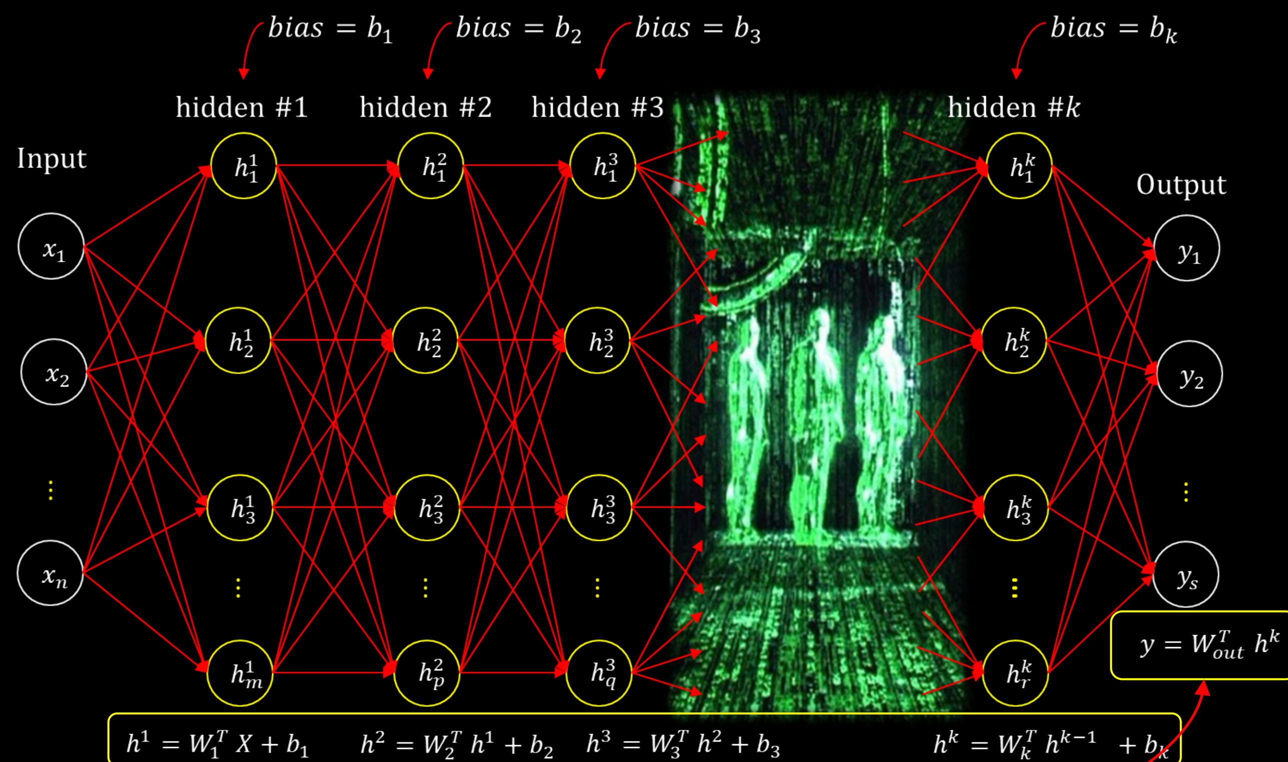
노기섭 교수

(kafa46@cju.ac.kr)

Types of Matrix Operation

■ 행렬 연산의 종류

- 더하기 / 빼기
- 곱하기
  - 상수배
  - 행렬과 행렬의 곱



이번 강의에서 최종적으로 하고 싶은 것

도대체 여기서 무슨일이 벌어지는지 확인하기!

Matrix Operation

■ **더하기 / 빼기**

$$A \pm B = (a_{ij} \pm b_{ij})$$

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad A + B = \begin{pmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{pmatrix} = \begin{pmatrix} 6 & 8 \\ 10 & 12 \end{pmatrix}$$

$$B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \qquad A - B = \begin{pmatrix} 1-5 & 2-6 \\ 3-7 & 4-8 \end{pmatrix} = \begin{pmatrix} -4 & -4 \\ -4 & -4 \end{pmatrix}$$

■ **상수배 ('스칼라배' 이라고도 부릅니다 ^^.)**

$$cA = (ca_{ij}), where \;\; c \;\; is \;\; a \;\; constant\; number \; (NOT \;\; a \;\; matrix)$$

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad 3A = 3\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ 9 & 12 \end{pmatrix}$$

행렬과 행렬의 곱 - 직관적으로 이해하기

행(Row)

열(Row)

c_{11} c_{12} c_{21} c_{22}

$$A_{2\times3} = \begin{pmatrix} \boxed{1 \quad 2 \quad 3} \\ \boxed{4 \quad 5 \quad 6} \end{pmatrix} \qquad B_{3\times2} = \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix}$$

$$AB = C = \begin{pmatrix} 58 & 64 \\ 136 & 154 \end{pmatrix}$$

MUST be same!

$$A_{m\times n} \times B_{n\times p}$$

결과 $C_{m\times p}$

행렬과 행렬의 곱 - 수식으로 이해하기

$$A_{m\times \mathrm{n}} \qquad B_{n\times p}$$

$$AB = (c_{ik}) \quad such\ \ that\ \ c_{ik} = \sum_{j=1}^{n} a_{ij} \times b_{jk}$$

$$A_{2\times 3} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad B_{3\times 2} = \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix}$$

$$AB = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

$$\begin{pmatrix} 58 & 64 \\ 136 & 154 \end{pmatrix}$$

$$\begin{aligned} c_{11} = \sum_{j=1}^{3} a_{1j} \times b_{j1} &= a_{11} \times b_{11} + a_{12} \times b_{21} + a_{13} \times b_{31} \\ &= 1 \times 7 + 2 \times 9 + 3 \times 11 = 58 \end{aligned}$$

$$\begin{aligned} c_{12} = \sum_{j=1}^{3} a_{1j} \times b_{j2} &= a_{11} \times b_{12} + a_{12} \times b_{22} + a_{13} \times b_{32} \\ &= 1 \times 8 + 2 \times 10 + 3 \times 12 = 64 \end{aligned}$$

$$\begin{aligned} c_{21} = \sum_{j=1}^{3} a_{2j} \times b_{j1} &= a_{21} \times b_{11} + a_{22} \times b_{21} + a_{23} \times b_{31} \\ &= 4 \times 7 + 5 \times 9 + 6 \times 11 = 136 \end{aligned}$$

$$\begin{aligned} c_{22} = \sum_{j=1}^{3} a_{2j} \times b_{j2} &= a_{21} \times b_{12} + a_{22} \times b_{22} + a_{23} \times b_{32} \\ &= 4 \times 8 + 5 \times 10 + 6 \times 12 = 154 \end{aligned}$$

행렬과 행렬의 곱

■ 행렬 곱하기의 특징

- 순서가 다르면 결과는 다르다.
  - 수 체계에서는 성립!

    $x \times y = y \times x$

  - 행렬의 세계에서는...

    $AB \neq BA$

$$A_{m\times n} \times B_{n\times p} = C_{m\times p}$$

$$B_{n\times p} \times A_{m\times n} = ???$$

같을수도 있지만, 다를 수도 있다.

- 교환법칙이 성립하지 않는다.
  - 헐~~ 왜 그런가요?
  - 현재까지로는 이해하기 어렵습니다. ㅠ
    - 행렬의 곱은 선형 변환의 합성이기 때문입니다.
    - 수학적으로 합성함수와 같기 때문입니다.
    - 나중에 벡터를 공부하면서 점점 알아게게 됩니다 ^^.

System of Linear Equations

■ 행렬이 발명된 이유

- 연립일차방정식을 풀기 위해 발명

■ 연립일차방정식을 행렬로 표현하는 방법

'첨가행렬' 이라고 부릅니다.

$$\left.\begin{aligned} x + 2y &= 20 \\ x + y &= 12 \end{aligned}\right]$$

표현 방법		풀이 방법
$\begin{pmatrix} 1 & 2 & 20 \\ 1 & 1 & 12 \end{pmatrix}$	⇨	가우스-조던 소거법
$\begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 20 \\ 12 \end{pmatrix}$	⇨	역행렬 이용 (Inverse Matrix)

'상수행렬' 이라고 부릅니다.

'계수행렬' 이라고 부릅니다.

Overview on Gauss-Jordan Elimination

■ Gauss-Jordan Elimination (가우스 조던 소거법)

- 머신러닝(딥러닝)에서는 사용하지 않습니다.
  - 그래도 행렬 연산의 기본 취지를 이해하는데 도움이 됩니다.
  - 우리가 연립일차방정식을 푸는 방식과 똑같습니다.

- 복잡해 보이지만 알고 있어야 하는 이유
  - 모든 연립일차방정식은 가우스-조던 소거법을 적용하여 쉽게 프로그래밍 가능합니다. ^^
  - 참고 블로그
    - https://www.codesansar.com/numerical-methods/gauss-jordan-method-python-program-output.htm

- 방법
  - 각 행에 상수를 곱할 수 있다 (단 0을 곱하는 것은 안됨)
  - 각 행의 위치를 서로 바꿀 수 있다.
  - 어떤 행을 상수배 하여 다른 행에 더할 수 있다.

Toy Example on Gauss-Jordan Elimination

$$\begin{aligned} x - 2y &= 1 \\ 3x + 2y &= 12 \end{aligned} \xrightarrow{\text{첨가행렬}} \begin{pmatrix} 1 & -2 & 1 \\ 3 & 2 & 11 \end{pmatrix}$$

1행에 -3을 곱한다.

$$\begin{aligned} -3x + 6y &= -3 \\ 3x + 2y &= 12 \end{aligned} \xrightarrow{\text{첨가행렬}} \begin{pmatrix} -3 & 6 & -3 \\ 3 & 2 & 11 \end{pmatrix}$$

2행에 1행을 더한다.

$$\begin{aligned} -3x + 6y &= -3 \\ 8y &= 8 \end{aligned} \xrightarrow{\text{첨가행렬}} \begin{pmatrix} -3 & 6 & -3 \\ 0 & 8 & 8 \end{pmatrix}$$

1행을 -3으로 나눈다

$$\begin{aligned} x - 2y &= 1 \\ 8y &= 8 \end{aligned} \xrightarrow{\text{첨가행렬}} \begin{pmatrix} 1 & -2 & 1 \\ 0 & 8 & 8 \end{pmatrix}$$

2행을 8로 나눈다

$$\begin{aligned} x - 2y &= 1 \\ y &= 1 \end{aligned} \xrightarrow{\text{첨가행렬}} \begin{pmatrix} 1 & -2 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

2행에 2를 곱하여 1행에 더한다.

$$\begin{aligned} x &= 3 \\ y &= 1 \end{aligned} \xrightarrow{\text{첨가행렬}} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \end{pmatrix}$$

최종 해를 구한다 ⇨ $x = 3, \quad y = 1$

Row echelon form (행사다리꼴)

$$\begin{pmatrix} \blacksquare & * & * & * & * \\ 0 & 0 & \blacksquare & * & * \\ 0 & 0 & 0 & \blacksquare & * \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$\blacksquare$: leading entry of a row

$*$: arbitrary number

가우스 소거법

Reduced Row echelon form
(기약행사다리꼴)

$$\begin{pmatrix} 1 & * & 0 & 0 & * \\ 0 & 0 & 1 & 0 & * \\ 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Leading entry of a row in each non-zero row is 1

Leading entry is only non-zero in the column

가우스-조던 소거법

수고하셨습니다 ..^^..