선형대수학 클린업 1주차

❤ 선형대수학 클린업에 오신 걸 환영합니다 ❤

[WELCOME]

선형대수학 클린업에서는 선형대수를 공부합니다. (당연)
행대수를 이미 수강하신 분들이시니까, 그렇게 막~~~ 새롭거나 막~~~ 낯설지는 않을 거예요!!

선형대수 개념을 수식적, 공간적으로 이해하는 것이 클린업의 목표입니다. 수식보다는 언어/공간적으로 이야기를 풀어갈 예정이며, 매주 응용 개념과 함께 선형대수의 쓸모를 경험하는 시간도 가지려 합니다. 기대해주세요!

부족한게 많은 팀장이지만..! 많이 소통하고, 서로 부족한 점은 채워주면서 즐거운 시간 만들어요!!!

[스터디 FLOW]

Week1 "선형대수학! 자기소개 시작!"

선형대수학 기본 개념 + 딥러닝 기본원리에 적용하기

벡터, 행렬, 선형방정식, 선형변환

Week2 "선형대수학 개념 살찌우기!"

벡터관련 공간개념+ 회귀에 적용하기

역행렬, 행렬식, 부분공간, 기저, span, 투영벡터

Week3 "통계이론 적용타임!"

PCA, SVD, LSA 등의 통계 개념을 선대관점에서 이해하기

[TODAY's GOAL]

** 기본 개념부터 차근차근 다져봅시다 **

1. **선형대수 왜 공부해야 하지?!**
2. **기본개념**
   1) 선형의 개념
   2) 벡터
   3) 행렬
3. **선형방정식과 선형결합**
   1) 선형방정식 Ax=b
   2) Ax=b 판별 및 해 구하기
   3) 선형결합
4. **선형변환**
5. **딥러닝에 적용해보자! : Affine Transformation**

지금부터 시작합니다!

[Contents]

1. 선형대수(linear algebra) 왜 공부하지?

선형대수학은 선형성을 연구하는 대수학의 한 분야이다. *이 문장만으로 감이 온다면, 그대는 이미 선형대수 master~*

선형대수학은 선형성을 바탕으로 선형변환과 그때의 공간에 대해 연구하는데, 우선은 간단하게 '연립선형방정식의 해 구하기'를 핵심연구로 하는 대수학의 한 분야 정도로만 이해해도 좋다. 기본단위는 '행렬'과 '벡터'이며, 이를 바탕으로 문제를 공간적으로 이해 및 처리한다. 이것이 우리 사회의 여러 문제들을 수학적으로 이해하고 해결할 때 적용되면서, 선형대수는 현재 가장 연구가 활발한 21세기 수학분야 중 하나가 되었다.

통계학에서도 선형대수는 문제의 이해와 처리, 그 시작점에 위치한다.
데이터분석에서 수십만 차원의 고차원 데이터를 다루는 것은 매우 흔한 일이다. 하지만 이것을 아무런 처리 없이 그 자체만으로 이해하고 처리하는 것은 사실상 불가능하다. 그렇게 하지 않는다는 표현이 더 적절할지도 모르겠다. 효율적으로 데이터를 다루기 위해서 데이터프레임화나, 행렬화, 차원축소 등의 과정이 수반되어야 하는데, 이때 사용되는 것이 바로 선형대수의 원리 아니겠는가? 머신러닝부터 딥러닝까지, 전처리부터 처리 모델 알고리즘까지 선형대수가 포함되지 않는 곳이 없으며, 선형대수를 기본으로 한다. 영어 공부가 알파벳부터 시작하듯, 통계를 이해하는데 선형대수가 선행되어야 하는 것은 어쩌면 당연한 것이다.

의미부여를 좀 강하게 하긴 했지만! 어쨌든 선대가 정말정말 중요한건 사실이고, 3주동안의 클린업을 통해서 어떤 식으로 선대가 쓰이는지 살펴보아요~~

2. 기본개념

A. 선형성

함수f 에 대해 다음의 두 식이 항상 성립할 때, 함수f 를 선형이라고 한다.

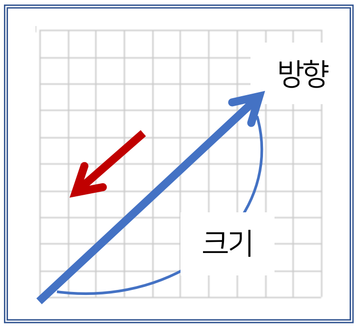
가산성 : 임의의 수 x와 y에 대해 $f(x+y) = f(x) + f(y)$
동차성 : 임의의 수 x와 α에 대해 $f(\alpha x) = \alpha f(x)$

B. 벡터(vector)

1) 개념

선형대수학의 기본단위이자, 데이터분석시에도 중요한 개념이므로 제대로 살펴보자.

✓ 벡터의 물리적 의미

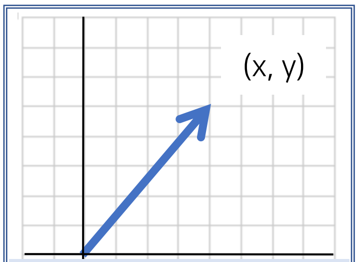


물리적으로 벡터는 '방향'과 '크기'로 구성된다. '어느 방향으로 얼마만큼의 힘 또는 속도'를 갖는지를 모두 포함하는 개념으로, 공간상에서는 화살표로 벡터를 표현한다. 특히 '방향'을 포함한다는 점에서, 하나의 수치로 길이나 넓이를 표현하는 스칼라(scalar)와 차이가 있다.

✓ 데이터분석의 기본단위

컴퓨터과학에서 벡터는 순서가 있는 숫자의 리스트이며, 데이터를 벡터의 성분으로 매칭하여 표현한다. 즉, 데이터분석에서 대부분의 변수를 벡터로 표현한다는 것이다. 사실, 통프입을 들어본 그대라면, 벡터가 데이터분석의 기본단위라는 것을 쉽게 받아들일 수 있을 것이다. R을 비롯한 통계 분석프로그램에서는 행렬, 배열, 데이터프레임 등 다양한 데이터형태를 정의하지만, 그 뿌리에는 다 벡터가 있지 않았던가. 결국 벡터는 다른 데이터를 구성하는 기본임과 동시에, 데이터를 선형적인 관점에서 이해할 수 있는 핵심 매개가 된다.

✓ 선형대수학의 공간적 이해



2차원 공간, $\mathbb{R}^2$

선형대수학에서는 위의 두 개념을 종합하여, 공간적으로 벡터를 이해한다. 벡터는 원점을 중심으로 뻗어 나가는 화살표이며, 이들을 구성하는 값을 벡터의 원소(component)라고 한다. 벡터로 이뤄지는 공간을 벡터공간(vector space, $\mathbb{R}$)이라고 하며, 이때의 차원은 성분의 수로 결정된다. 즉, n개의 원소로 이뤄진 벡터는 n차원 공간에 있다.

2) 벡터의 연산

✓ 수식적으로 계산하기

벡터의 연산은 원소끼리 이뤄진다. 벡터 v, w와 c는 상수에 관한 예시를 보자!

$$v = \begin{pmatrix} a \\ b \end{pmatrix} \quad w = \begin{pmatrix} c \\ d \end{pmatrix}$$

상수배	덧셈	뺄셈
$cv = \begin{pmatrix} cv \\ cd \end{pmatrix}$	$v+w = \begin{pmatrix} a+c \\ b+d \end{pmatrix}$	$v-w = \begin{pmatrix} a-c \\ b-d \end{pmatrix}$

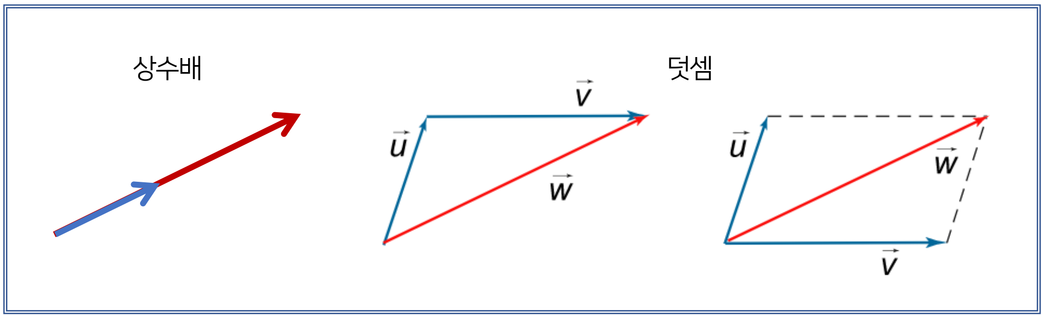
✓ 기하학적으로 이해하기

- 상수배

벡터의 스칼라c를 곱하는 것은 기하학적으로 벡터의 길이를 c배 한다는 의미이다. c의 절댓값배로 벡터의 길이가 변화하며, c의 부호에 따라 벡터의 방향도 바뀐다.

- 덧셈 뺄셈

벡터 u에서 벡터 v만큼 이동하면서 생기는 평행사변형의 대각선이 u+v 즉, 벡터 w이다. 이때 벡터v에 대해 -1배 한 뒤(방향을 바꾼 뒤), 벡터 u와 더하면 벡터 u-v가된다.



🌞 기본 연산을 왜 이렇게 자세하게 다루냐고요? 🌞

범인은 바로 '선형성' !! 선형성을 만족하기 위해서는 기본적으로 상수배와 합이 이용되기 때문에, 분석에 기초가 되는 벡터의 상수배와 합을 신경써서 다뤄봤습니다^^

3) 벡터의 연산법칙

$\mathbb{R}^n$의 벡터 x, y, z와 스칼라 h, k에 대하여 다음이 성립한다.

(1) $\mathrm{x}+\mathrm{y}=\mathrm{y}+\mathrm{x}$
(2) $(\mathrm{x}+\mathrm{y})+\mathrm{z}=\mathrm{x}+(\mathrm{y}+\mathrm{z})$
(3) $\mathrm{x}+0=\mathrm{x}=0+\mathrm{x}$
(4) $\mathrm{x}+(-\mathrm{x})=0=(-\mathrm{x})+\mathrm{x}$
(5) $k(\mathrm{x}+\mathrm{y})=k\mathrm{x}+k\mathrm{y}$
(6) $(h+k)\mathrm{x}=h\mathrm{x}+k\mathrm{x}$
(7) $(hk)\mathrm{x}=h(k\mathrm{x})$
(8) $1\mathrm{x}=\mathrm{x}$

C. 행렬(matrix)

1) 개념

실수를 직사각형 모양의 행과 열로 배열한 것을 행렬이라고 한다. 배열 안의 수 각각을 행렬의 원소(element) 또는 성분(entry)라고 하며, 행(m)과 열(n)의 개수를 이용해 크기(m x n)를 표현한다. m과 n의 크기가 같아 n x n의 행렬이 되는 경우, 이때의 행렬을 정사각행렬(square matrix)이라 하고 성분 $a_{11}, a_{22}, \dots, a_{nn}$을 주대각성분이라고 한다.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ \vdots & \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} \end{bmatrix}$$

2) 행렬의 연산

- 상수배 : 모든 원소에 같은 상수를 곱함
- 행렬의 합 : 크기가 같은 두 행렬에서 같은 위치에 있는 원소끼리 합함
- 행렬의 곱 : 앞 행렬의 열과 뒤 행렬의 행이 크기가 같을 때, 원소의 곱의 합, AB와 BA는 다름 주의!

$$C = AB \qquad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ a_{m1} & \cdots & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ b_{n1} & \cdots & \cdots & b_{nr} \end{bmatrix}$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$$

$$\underset{(m \times n)}{A} \quad \underset{(n \times r)}{B} \quad = \quad \underset{(m \times r)}{C}$$

(A의 열의 개수 n = B의 행의 개수 n)

- 예시

$$A = \begin{bmatrix} 1 & 3 & -2 \\ 3 & 1 & 2 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & 4 & -5 \\ -1 & 2 & 3 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & -1 \\ 3 & 2 \\ -1 & 3 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 1+2 & 3+4 & -2-5 \\ 3-1 & 1+2 & 2+3 \end{bmatrix} = \begin{bmatrix} 3 & 7 & -7 \\ 2 & 3 & 5 \end{bmatrix} \qquad 3A = \begin{bmatrix} 1x3 & 3x3 & -2x3 \\ 3x3 & 1x3 & 2x3 \end{bmatrix} = \begin{bmatrix} 3 & 9 & -6 \\ 9 & 3 & 6 \end{bmatrix}$$

$$AC = \begin{bmatrix} 1x2+3x3+2x\text{-}1 & 1x-1+3x2+2x3 \\ -2x2+1x3+2x-1 & -2x-1+1x2+2x3 \end{bmatrix} = \begin{bmatrix} 9 & 11 \\ -3 & 10 \end{bmatrix}$$

3) 행렬의 종류

- 영행렬 (zero matrix) : 모든 원소가 0인 행렬, 0으로 표기
- 정방행렬 (square matrix) : 행과 열의 개수가 같은 n x n 크기의 정사각형 모양행렬
- 대각행렬 (diagonal matrix) : 정방행렬 중 주대각선(main diagonal, aii) 이외의 값이 0인 행렬
- 단위행렬 (identity matrix) : 대각행렬 중 주대각선이 1인 행렬(I)
- 전치행렬 (Transpose) : 행과 열을 교환하여 얻는 행렬(A^t)

  m x n 크기의 행렬 A의 전치행렬은 n x m

  정방행렬은 주대각선을 기준으로 대칭

대각행렬

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$

단위행렬

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

전치행렬

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$B = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \quad B^T = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

4) 역행렬

n x n의 정방행렬A에 대하여, 행렬A에 다른 행렬 C를 곱한 값이 n x n의 단위행렬일 때, C를 A의 역행렬이라고 한다.

특정한 행렬A는 오직 하나의 역행렬만 갖기 때문에 해가 유일하다(unique)하고 하며, 역행렬을 갖는 행렬A에 대해서는 가역하다(invertible)고 한다.

$$\overset{\text{Matrix A}}{A} \cdot \overset{\text{Inverse of Matrix A}}{A^{-1}} = \underset{\text{Identity Matrix}}{I}$$

$$Ax = b$$

$$A^{-1}Ax = A^{-1}b$$

$$x = A^{-1}b$$

• Properties

$$(A^{-1})^{-1} = A \qquad (AB)^{-1} = B^{-1}A^{-1} \qquad (A^T)^{-1} = (A^{-1})^T$$

• 2×2 matrix의 경우

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow A^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

If ad-$bc = 0$, then A is not invertible.

3. 선형방정식과 선형결합

A. 선형방정식(Linear equation)

양의 정수 n에 대하여, $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$ 형태로 표현되는 식을 선형방정식이라고 하고, 하나 이상의 선형방정식 집합을 연립선형방정식이라 한다. 아래의 연립선형방정식을 행렬 차원으로 정리하면 오른쪽과 같으며, Ax=b (행렬 x 벡터 = 벡터) 꼴이 된다

[연립선형방정식]

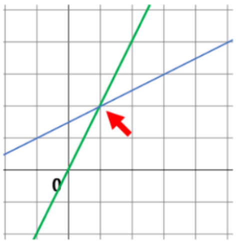
$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

이걸 그래서 어디다 쓰는거냐고요??

$\begin{cases} 2x - y = 0 \\ -x + 2y = 3 \end{cases}$ 의 해를 구하여라.

이 문제를 푸는 것은 결코 어렵지 않다. 문자와 식이 모두 2개인 연립방정식의 해구하기?? 그쯤이야 뭐~ 완전 가능이지!



하지만!! 문자와 식의 개수가 맞지 않다면? 일반화된 해를 찾고 싶다면? 이럴 때 사용하는 것이 바로 선형방정식이다!!

위의 연립방정식은 $x\begin{bmatrix} 2 \\ -1 \end{bmatrix} + y\begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$ 와 같이 표현할 수 있으며,

이를 Ax=b의 꼴로 표현하면 $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$,

일반화해서 표현하면 $x\begin{bmatrix} 2 \\ -1 \end{bmatrix} + y\begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ 이다.

B. Ax=b 판별 및 해 구하기

기본 연산을 이용해 문자를 소거하는 Gauss-Jordan 소거법으로 Ax=b의 해를 구하고, 판별해보자.

1) 해 구하기

✓ Elementary Row Operations (ERO, 기본행 연산)

(1) 두 식을 교환한다 $\mathrm{R}_i \leftrightarrow \mathrm{R}_j$

(2) 한 식에 0이 아닌 실수를 곱한다. $k\mathrm{R}$

(3) 한 식에 0이 아닌 실수배를 하여 다른 식에 더한다. $k\mathrm{R}_i + \mathrm{R}_j$

✓ Row echelon form (REF)

(1) 성분이 모두 0인 행이 존재하면 그 행은 행렬의 맨 아래에 위치한다.

(2) 각 행에서 처음으로 나타나는 0이 아닌 성분은 1이다. 이때 이 1을 'leading 1(선행성분, pivot)'이라고 한다.

(3) i 행과 (i+1)행에 모두 leading1이 존재하면, (i+1)행의 선행성분은 i 행의 선행성분보다 오른쪽에 위치한다.

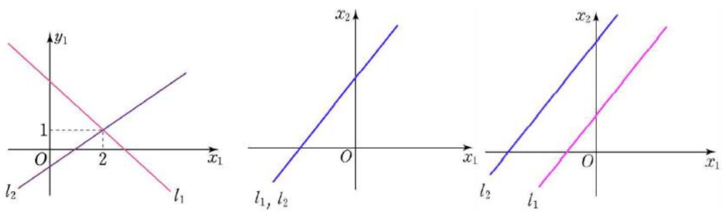
$$\begin{bmatrix} 1 & 5 & -3 & -4 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 4 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} \boxed{1} & 1 & -2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

이때 마지막 4번째 성질을 추가로 만족하면 Reduced row echelon form (RREF)라고 한다.

(4) leading entry를 포함하는 열의 선행선분 외의 성분은 모두 0이다.

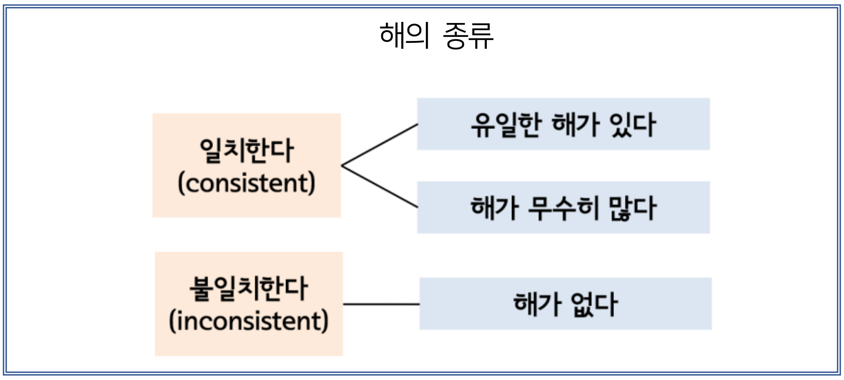
$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & -2 & 0 & -1 \\ 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2) 해의 종류



일반적으로 연립선형방정식은 (1) 유일한 해를 갖는다. (2) 해가 무수히 많다. (3) 해가 없다. 중 하나만 만족한다.

해가 1개 이상 존재하는 경우는 consistent, 해가 존재하지 않는 경우는 inconsistent하다고 한다.



3) 해의 판별

Ax = b를 [A | b]의 형태로 만들고, G-J elimination method를 이용해 RREF인 [H | c]로 만든다.

(1) H의 모든 열에 피벗이 존재할 때 : 유일한 해가 있다.

(2) H의 일부 열에 피벗이 없을 때 : 해가 무수히 많다.

이때 피벗이 없는 열의 변수를 자유변수(free variable)이라 한다.

(3) H의 피벗이 없는 행의 오른쪽 숫자가 0이 아닐 때 : 해가 없다.

$$\begin{aligned} x + y + 3z &= 0\\ 2x + 3y - z &= 9\\ 3x - 2y + 3z &= -4 \end{aligned} \xrightarrow{[A\mid b]\text{형태}} \left(\begin{array}{ccc|c} 1 & 1 & 3 & 0\\ 2 & 3 & -1 & 9\\ 3 & -2 & 3 & -4 \end{array}\right) \xrightarrow[\text{r3-3*r1}]{\text{r2- 2*r1}} \left(\begin{array}{ccc|c} 1 & 1 & 3 & 0\\ 0 & 1 & -7 & 9\\ 0 & -5 & -6 & -4 \end{array}\right)$$

$$\xrightarrow{\text{r3+5*r2}} \left(\begin{array}{ccc|c} 1 & 1 & 3 & 0\\ 0 & 1 & -7 & 9\\ 0 & 0 & -41 & 41 \end{array}\right) \xrightarrow{\text{r3 /-41}} \left(\begin{array}{ccc|c} 1 & 1 & 3 & 0\\ 0 & 1 & -7 & 9\\ 0 & 0 & 1 & -1 \end{array}\right)$$

$$\longrightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1\\ 0 & 1 & 0 & 2\\ 0 & 0 & 1 & -1 \end{array}\right) \longrightarrow \begin{array}{c} \mathbf{x = 1}\\ \mathbf{y = 2}\\ \mathbf{z = -1} \end{array}$$

해가 하나이다.

$$\left(\begin{array}{ccc|c} 1 & 2 & 0 & 3\\ 0 & 1 & 0 & 1\\ 0 & 0 & 0 & 1 \end{array}\right)$$

해가 없다

$$\left(\begin{array}{ccc|c} 1 & 0 & -2 & 1\\ 0 & 1 & 1 & 1\\ 0 & 0 & 0 & 0 \end{array}\right) \longrightarrow \begin{array}{l} x = 2z + 1\\ y = -z + 1 \end{array} \longrightarrow \begin{array}{l} z = t\ (\text{자유변수})\\ x = -2t + 3\\ y = -t + 1 \end{array}$$

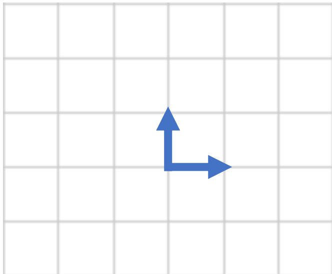
해가 무수히 많다

D. 선형결합(Linear combination)

선형결합이란 벡터들을 스칼라배와 벡터 덧셈을 통해 조합하여 새로운 벡터를 얻는 연산을 말한다.

$$Ax = [a_1\, a_2 \cdots\; a_n]\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = a_1x_1 + a_2x_2 + \ldots + a_nx_n$$

위의 그림과 같이 m x n 행렬 A와 n개의 원소를 가진 벡터 x를 곱할 때, a의 조합을 span이라고 한다. 벡터를 구성하는 기본벡터(격자)를 '기저벡터'라 할 때, 벡터 공간 내 모든 벡터는 기저벡터의 span(조합)으로 구성할 수 있다.



위와 같은 2차원 벡터공간의 모든 벡터는 기저벡터인 (0,1)과 (1,0)의 조합을 통해 표현 가능하다. 기저와 span에 대해서 할 말은 많지만, 오늘은 이정도만 하고 자세한 내용은 2주차에서 살펴보자.

본 장에서는 'Ax=b의 해가 존재한다'가 'a의 조합으로 b를 표현할 수 있다', '벡터 b가 span(a1,a2,⋯,an)에 있다'를 의미한다 정도로만 이해해주세요. *패-스*

4. 선형변환(Linear transformation)

A. 선형'변환'의 의미

변환(transformation)을 input에 대해 output을 반환하는 함수처럼 이해할 수 있다.

선형방정식 Ax=b를 변환의 관점에서 이해하면, input으로 받은 벡터x에 A라는 처치를 한 뒤 output으로 새로운 벡터 b를 반환하는 것과 같다.

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

이 방정식을 $\mathbb{R}^2$의 벡터 x에 2x2 크기의 행렬 A를 곱함으로써 벡터 x를 새로운 $\mathbb{R}^2$의 벡터로 변환하는 것으로 볼 수 있으며, 결국 Ax=b의 해를 찾는 것은 A를 곱했을 때 b로 변환되는 벡터 x를 찾는 문제가 된다.

B. '선형'변환의 의미

1) 수식적 의미

$\mathbb{R}^n$에서 $\mathbb{R}^m$으로 변환하는 T가 선형의 조건을 만족하면 선형변환이라고 한다.

(1) T(u+v) = T(u) + Y(v)

(2) T(ku) = k T(u) (k는 상수)

정의역에서 원소 두개를 뽑아 선형결합해서 나오는 함수가, 선형결함을 나중에 해서 나오는 함수 값과 일치해야 한다고 해석할 수 있으며, 때문에 공간위의 직선은 변환 후에도 직선이어야 하고 원점은 변환 후에도 원점이어야 한다.

2) 공간적 의미

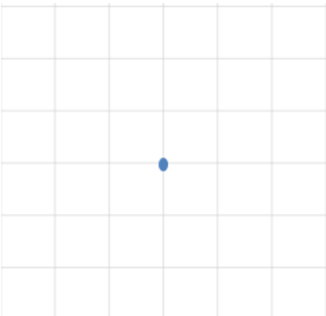
벡터공간 내 모든 벡터는 기저의 span(상수배의 합)으로 구성할 수 있으므로, 선형변환을 새로운 기저, 새로운 공간의 형성으로도 이해할 수 있다. 구체적으로 예시를 살펴보자.

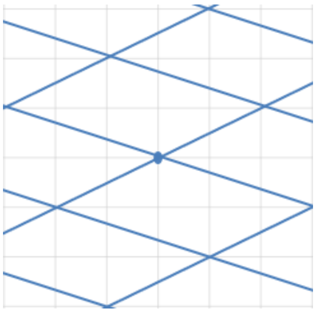
$$\begin{bmatrix} 2 & -3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

벡터 (1,1)에 행렬을 곱해 (-1,2)로 변환한 경우이다. T(x)를 선형변환이라고 할 때,

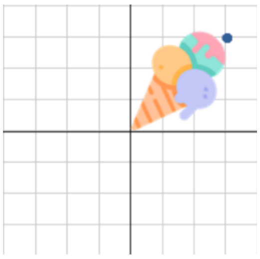
$$\begin{aligned} \mathrm{T}\begin{bmatrix} 1 \\ 1 \end{bmatrix} &= 1\mathrm{T}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1\mathrm{T}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= 1\begin{bmatrix} 2 & -3 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1\begin{bmatrix} 2 & -3 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= 1\begin{bmatrix} 2 \\ 1 \end{bmatrix} + 1\begin{bmatrix} -3 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \end{bmatrix} \end{aligned}$$

로 쪼갤 수 있는데, 이는 기저벡터 (1,0), (0,1)의 조합에서 기저벡터(2,1), (-3,1)의 조합으로 변형, 새로운 공간이 형성되었음을 의미한다.



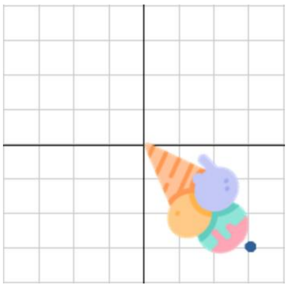


3) 선형변환의 종류



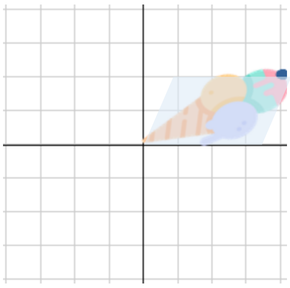
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 이라 하자

- Reflection (특정 축 중심 반전)



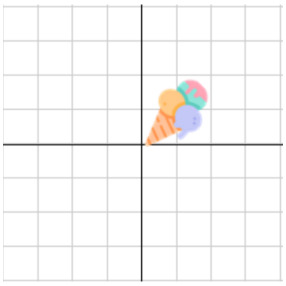
$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

- Shear (평행사변형으로 변환)



$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$

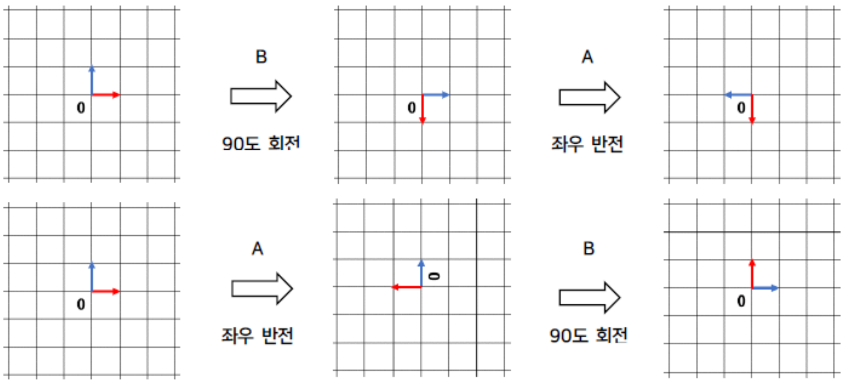
- Contractions and Expansions (특정 k배로 수축 또는 확장)



$\begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}$

행렬을 선형변환의 차원에서 이해할 수 있다면 AB 와 BA가 다른 이유도 시각적으로 이해할 수 있다!

A는 공간을 오른쪽으로 90도 회전시키는 선형변환, B는 공간을 좌우로 뒤집는 선형변환이라고 하면
AB = 공간을 좌우로 뒤집은 다음 오른쪽으로 90도 회전시키는 선형변환
BA = 공간으로 오른쪽으로 90도 회전시킨 다음 좌우로 뒤집는 선형변환



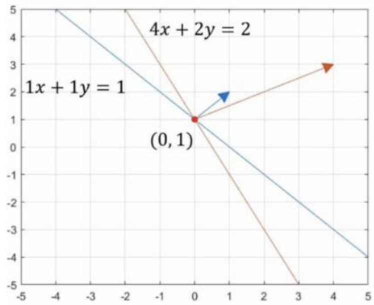
4) 선형변환으로 Gauss-Jordan 소거법으로 이해하기

(1) Ax=b를 [A|b] 형태로 변경

$$\left[\begin{array}{cc|c} 2 & 2 & 2 \\ 4 & 2 & 2 \end{array}\right]$$

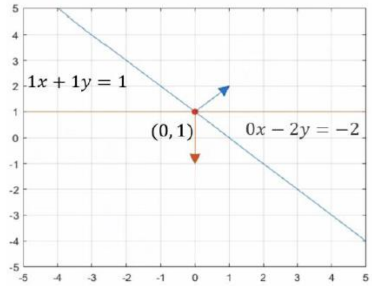
(2) Scaling (r1 * 0.5)

$$\left[\begin{array}{cc|c} 1 & 1 & 1 \\ 4 & 2 & 2 \end{array}\right]$$



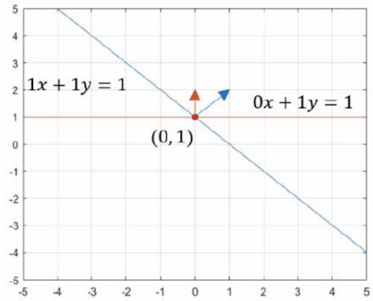
(3) Subtraction (r2 – 4*r1)

$$\left[\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & -2 & -2 \end{array}\right]$$



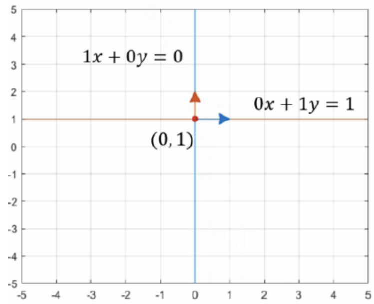
(4) Scaling (r2 * -0.5)

$$\left[\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & 1 & 1 \end{array}\right]$$



(5) Subtraction (r1 – r2)

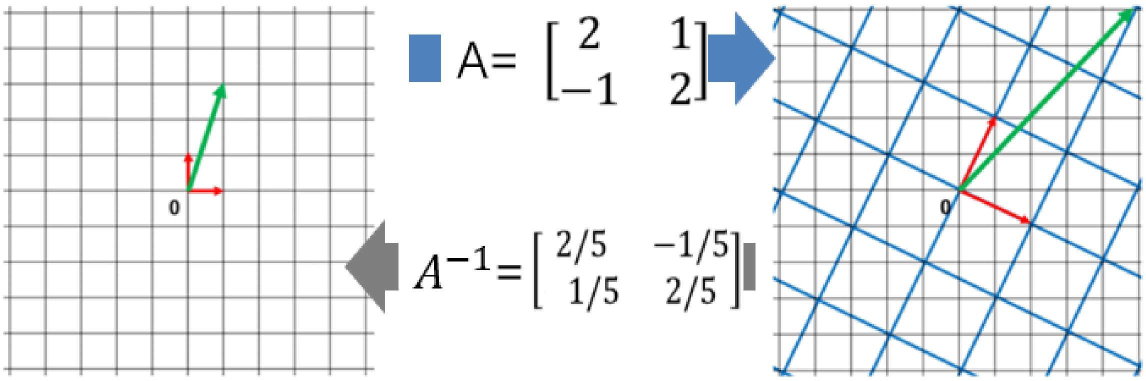
$$\left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right]$$



5) 선형변환으로 역행렬 이해하기

어떤 벡터x가 A라는 선형변환을 통해 b라는 벡터로 변환되었을 때, 벡터b를 다시 벡터x로 되돌리는 선형변환!으로 역행렬을 다시 이해해보자.

- 역행렬이 있는 경우 (unique, invertible)

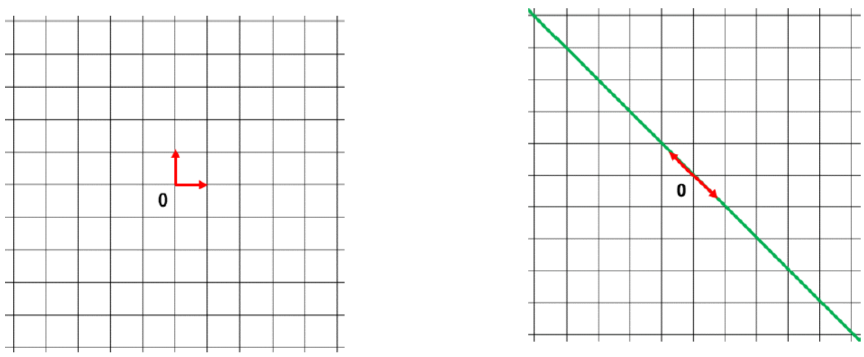


검은 좌표계에서 행렬A를 곱하면 파란 좌표계로 선형변환이 이뤄지는데, 이 때의 파란 좌표계에 A의 역행렬을 곱하면 검은 좌표계로 되돌리는 것이 가능하다.

정방행렬A(nxn)의 역행렬이 존재한다는 것은 'Ax=b가 유일한 해를 갖는다(unique)'와 같은 의미라고 했었다. Ax의 해가 유일하다는 것은 특정x를 선형변환한 Ax가 유일하다는 것을 의미하며, 결국, x와 Ax가 서로 일대일 대응임을 뜻한다.

- 역행렬이 없는 경우

위의 예시는 선형변환 전과 후의 차원이 2차원으로 유지되는 경우였다. 그렇다면, 공간을 압축시키는 선형변환에 의해 선형변환 전 후의 차원이 달라진다면 어떻게 될까? 결론부터 말하자면 이 경우에는 역행렬이 존재하지 않는다.



선형변환 결과로 2차원의 벡터가 다음과 같이 1차원으로 축소되었다고 하자. 이 경우 1차원 벡터를 어떠한 특정한 2차원 벡터로 되돌려야 하는지 알 수 없다. 이는 x와 Ax가 서로 일대일 대응이 아니었음을 뜻하며, 결국 'Ax=b의 해가 유일하지 않다.', '역행렬이 존재하지 않는다.'를 의미한다.

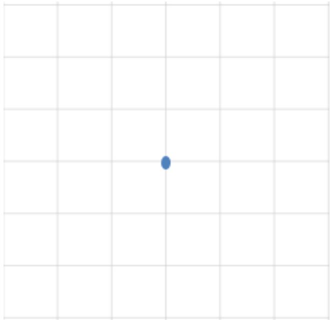
차원 축소의 기하학적 해석과 관련해 행렬식을 살펴봐야 맞지만! 이미 복잡하니까 2주차로 미룹시다~

5. 딥러닝에 적용해보자! : Affine Transformation

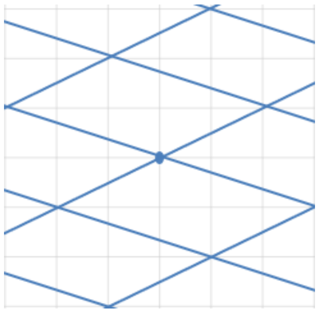
A. Affine transformation

선형변환은 '선형성'을 바탕으로 원점의 변함없이 벡터의 변환을 의미했다. 좌표는 바뀌지만 이동은 아니었다.

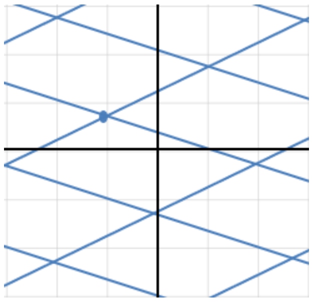
이제부터 살펴볼 아핀변환은 선형변환에 '평행이동'까지 추가된 개념으로, y= Ax+b로 표현하는 것이 일반적이다. 아핀변환은 평행이동으로 인해 원점이 변하기 때문에 선형성을 만족하지 않는다.



변환 전



선형변환



아핀변환

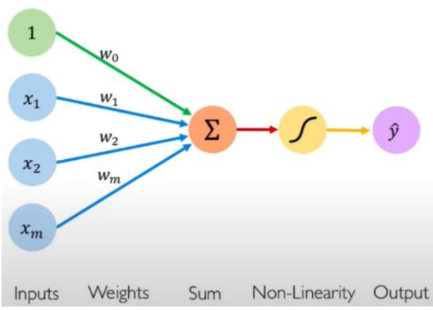
아핀변환을 bias(평행이동, b)까지 포함해서 나타내면 아래와 같이 표현가능하다.

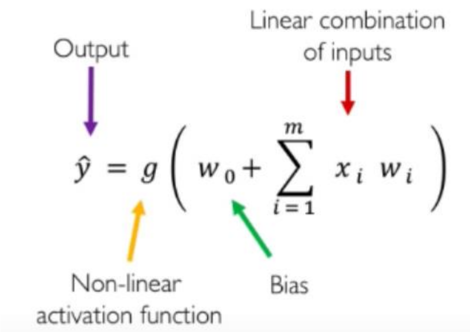
Ax+b (2차원)	선형결합	A'x' (3차원)
$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix}$	$a\begin{bmatrix} 2 \\ 1 \end{bmatrix} + b\begin{bmatrix} 1 \\ 2 \end{bmatrix} + 1\begin{bmatrix} c \\ d \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & c \\ 1 & 2 & d \end{bmatrix} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}$

B. 딥러닝

* 딥러닝에 대한 자세한 내용은 클린업을 참고하세요!!

딥러닝은 인공신경망을 이용하는 러신머닝으로, 주로 비정형데이터를 분석하는데 사용한다. 딥러닝의 학습은 입력된 input layer를 여러 개의 hidden layer을 거치면서 이뤄진다. 그것은 '활성화 함수로 입력값 예측' -> '손실함수로 예측과 실제의 오차 측정' -> '손실함수를 바탕으로 가중치 업데이트'의 반복이며, 그 결과로 선에서 모양, 모양에서 물건으로 점차 추상적인 표상이 가능해진다.



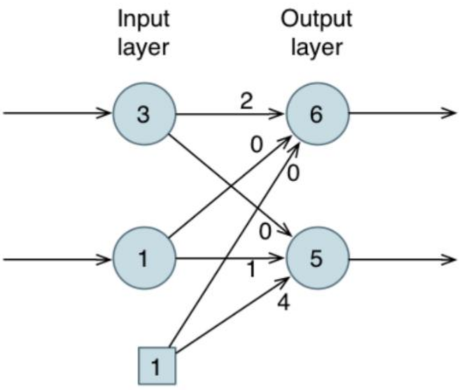


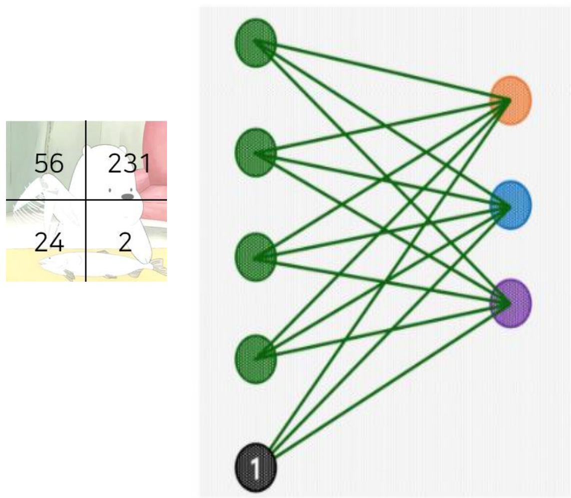
C. Affine과 딥러닝

아핀변환을 바탕으로 딥러닝의 원리를 이해해보자.

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

Ax=b의 선형변환에 bias (0,4)를 더한 아핀변환 식이 다음과 같을 때, x를 input layer의 입력 값으로, A를 이들의 가중치로 보면, 아래와 같이 딥러닝 모델에 적용 가능하다.





가중치행렬 input bias output

$$\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2 \\ 1.5 & 1.3 & 2.1 & 1 \\ -.2 & 0.3 & 0.7 & -1.3 \end{bmatrix} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix} = \begin{bmatrix} -96.8 \\ 439.9 \\ 71.1 \end{bmatrix}$$

$$= 56 \begin{bmatrix} 0.2 \\ 1.5 \\ -.2 \end{bmatrix} + 231 \begin{bmatrix} -0.5 \\ 1.3 \\ 0.3 \end{bmatrix} + 24 \begin{bmatrix} 0.1 \\ 2.1 \\ 0.7 \end{bmatrix} + 2 \begin{bmatrix} 2 \\ 1 \\ -1.3 \end{bmatrix} + 1 \begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}$$

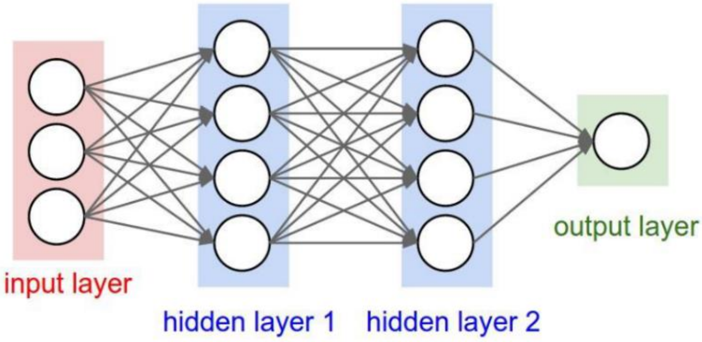
$$= \begin{bmatrix} 0.2 & -0.5 & 0.1 & 2 & 1.1 \\ 1.5 & 1.3 & 2.1 & 1 & 3.2 \\ -.2 & 0.3 & 0.7 & -1.3 & -1.2 \end{bmatrix} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \\ 1 \end{bmatrix}$$

🌟 '딥러닝이 아핀변환이다. 이 말인가요?

'딥러닝의 학습에서 bias항이 들어가는 것이 아핀변환과 맥락을 같이한다.'가 더 올바른 표현일 것 같아요.

딥러닝의 인풋 벡터는 가중치 행렬을 통한 선형변환에 의해 예측값, 즉 아웃풋 벡터로 변환되는데, 이때 bias 항을 고려하는 것이 아핀변환에서 평행이동을 고려하는 것과 같다. 는 뜻입니다!

🌟 그리고!! 혹시 딥러닝에서 가중치를 업데이트할 때 '활성화 함수'를 이용한다고 했던 거 기억하시나요?



딥러닝의 학습 과정에 '비선형' 활성화 함수를 바탕으로 네트워크를 구성하는 이유도 '선형'의 맥락으로 이해할 수 있어요!

f(x)=kx를 선형함수, g(x)=ax+b를 비선형함수라고 합시다. 딥러닝의 학습과정에서 선형함수 f(x)로 n번 층을 쌓으면 k^nx로, k^n를 한번 적용하는 것과 차이가 없어요. 즉, 여러 hidden layer을 쌓고 가중치를 업데이트하는 이점이 없다는 것이죠! 때문에 딥러닝에서는 sigmoid나 tanh 같은 '비선형' 활성화 함수를 이용해, 겹겹이 층을 쌓는 과정에 의미를 부여한답니다~

[Code]

벡터 뿌시기!

1. 원소를 1,2,3으로 하는 벡터를 만들어 봅시다.
2. 원소를 1~9로 하는 벡터를 만들어 봅시다.
3. 원소의 데이터 타입으로 a정수, b실수, c문자열을 부여해봅시다.
4. 벡터의 type, 차원, 원소개수를 확인해봅시다.
5. 논리 연산자를 이용해 필터링해 봅시다.
6. x2의 짝수 원소를 0으로 바꿔봅시다.
7. 벡터 연산을 해봅시다 (상수배, 합)

행렬 뿌시기!

1. 원소를 1~12로 하는 3x4 행렬을 만들어 봅시다.
2. 행렬의 type, 차원차원, 원소개수를 확인해봅시다.
3. 논리연산자로 행렬을 필터링해봅시다. (T,F,T)
4. y1에서 3의 배수를 -10으로 바꿔봅시다.
5. 행렬 연산을 해봅시다.(상수배, 합, 원소끼리곱, 행렬곱, 역행렬)

x+2y=3, 3x+4y=5 연립방정식의 해를 풀어봅시다.

끝! 고생하셨습니다!