AI 기초

2020년

강봉주

벡터와 행렬

벡터와 행렬

[벡터와 행렬에 대한 이해가 필요한 이유]

- 행렬: 정형화된 데이터 형식

학생번호	키(cm)	몸무게(kg)
0001	173	65
0002	180	80
…	…	…
0050	177	75

- 벡터: 데이터의 행 또는 열(관측값 벡터, 변수 벡터)

[벡터와 행렬에 대한 이해가 필요한 이유]

- 머신러닝, 딥러닝의 모델 식:

회귀분석: $y = \mathrm{X}\beta + \epsilon$,

지지벡터 머신: $h_{w,b} = w^{\mathrm{T}}x + b$,

인공신경망: $z^{(l+1)} = \mathrm{W}^{(l+1)}a^{(l)} + b^{(l+1)}$,

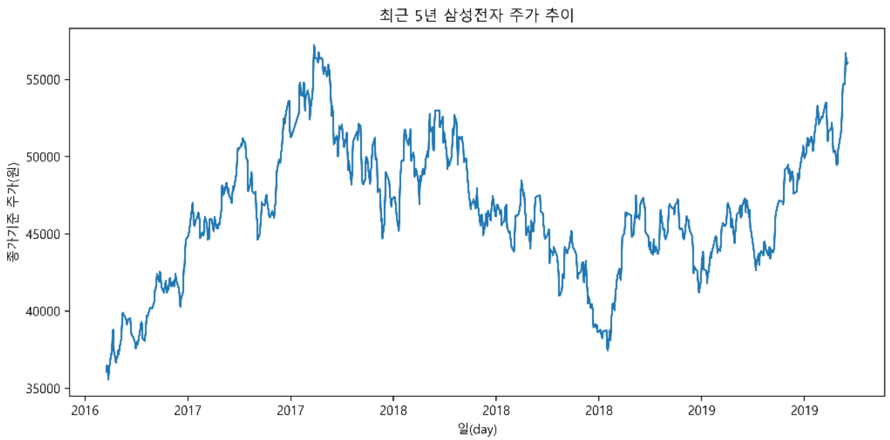
순환신경망: $h_t = \sigma_h(\mathrm{W}_h h_{t-1} + \mathrm{W}_x x_t + b_h)$

[벡터와 행렬에 대한 이해가 필요한 이유]

퀴즈	
	우리 주변에서 흔히 볼 수 있는 벡터로 표현할 수 있는 예를 찾아보자.

[벡터와 행렬에 대한 이해가 필요한 이유]

퀴즈	
	우리 주변에서 흔히 볼 수 있는 벡터로 표현할 수 있는 예를 찾아보자.



그림과 같이

1) Y축을 구성하고 있는 삼성전자 주가
2) X축을 구성하고 있는 일자

벡터

[벡터의 표현]

- 벡터(vector)는 순서가 있는 숫자들의 목록이다. 즉 일종의 배열이다. 표현은 대괄호나 괄호를 통하여 표현한다.

$$\begin{bmatrix} 0.61 \\ 0.93 \\ 0.24 \\ 0.27 \end{bmatrix}, \quad \begin{pmatrix} 0.61 \\ 0.93 \\ 0.24 \\ 0.27 \end{pmatrix} \qquad \begin{bmatrix} 0.61 \\ 0.93 \\ 0.24 \\ 0.27 \end{bmatrix} \neq \begin{bmatrix} 0.93 \\ 0.61 \\ 0.24 \\ 0.27 \end{bmatrix}$$

$$[0.74 \quad 0.75 \quad 0.93 \quad 0.46]$$
$$(0.74 \quad 0.75 \quad 0.93 \quad 0.46)$$

벡터와 행렬

[벡터의 표현]

- 벡터(vector)는 순서가 있는 숫자들의 목록이다. 즉 일종의 배열이다. 표현은 대괄호나 괄호를 통하여 표현한다.

In	# 벡터의 생성
	v = np.array([0.61, 0.93, 0.24, 0.27])
	v

Out	array([0.61, 0.93, 0.24, 0.27])

[**벡터의 원소**]

- 벡터의 원소(element, entry, coefficient, component)는 배열의 값들을 의미한다.

In	`# 벡터의 원소`
	`v = np.array([0.61, 0.93, 0.24, 0.27])`
	`v[0]`

Out	`0.61`

[벡터의 크기]

- 벡터의 크기(size, dimension, length)는 벡터의 원소의 개수를 의미한다.

In	

	# 벡터의 크기
	v = np.array([0.61, 0.93, 0.24, 0.27])
	v.shape

Out	(4,)

[부분 벡터]

- 벡터의 부분벡터(sub vector)는 해당 벡터의 부분(part)를 의미한다.
- 부분 벡터는 다양하게 정의할 수 있다.
- 대표적으로는 색인번호와 쌍점(colon) 기호를 사용하여 표현한다.

$$a = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} \qquad a_{r:s} = \begin{pmatrix} a_r \\ a_{r+1} \\ \dots \\ a_s \end{pmatrix}$$

[부분 벡터]

- 벡터의 부분벡터(sub vector)는 쌍점(colon) 기호를 사용하여 표현한다.
- 파이썬에서는 $r: s = \{r, r+1, \dots, s-1\}$ 이다.

In	`# 부분 벡터의 생성`
	`v = np.array([0.61, 0.93, 0.24, 0.27])`
	`v_sub = v[0:2]`
	`v_sub`

Out	`array([0.61, 0.93])`

[특별한 벡터]

- 모든 원소가 0인 벡터를 0벡터
- $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

In	`# 영 벡터`
	`size = 3`
	`zeros = np.zeros(shape=(size,))`
	`zeros`

Out	`array([0., 0., 0.])`

[특별한 벡터]

- 모든 원소가 1인 벡터를 1벡터

- $1_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

In	`# 1 벡터`
	`size = 3`
	`one = np.ones(shape=(size, ))`
	`one`

Out	`array([1., 1., 1.])`

[특별한 벡터]

- 하나의 원소만 1이고 나머지는 모두 0인 벡터를 단위(unit)벡터
- e_i: i 번째 원소만 1이고 나머지는 0

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

[특별한 벡터]

- 하나의 원소만 1이고 나머지는 모두 0인 벡터를 단위(unit)벡터

In	

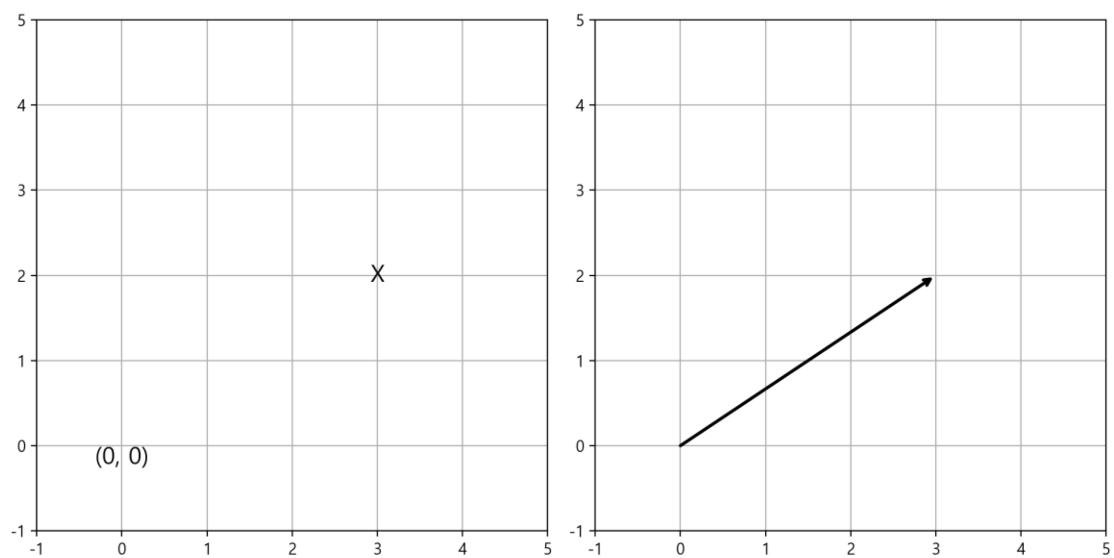
```
# 단위 벡터
size = 3
e = np.zeros(shape=(3,))

# 단위 벡터의 위치
i = 1

# 단위 벡터 생성
e[i]=1
e
```

Out

```
array([0., 1., 0.])
```

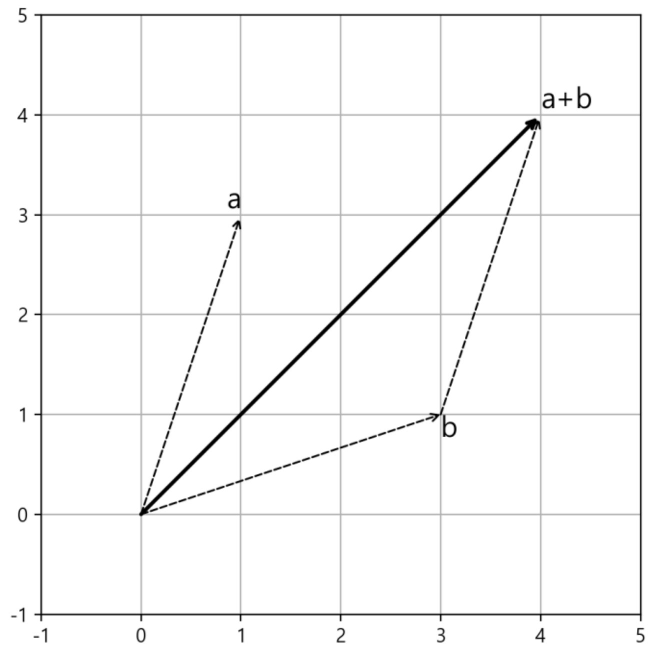
[벡터의 기하학적인 의미]

- $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ 벡터는 좌표 상의 1개의 점으로 표현되거나, 오른쪽 그림과 같이 $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 점을 $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$로 옮기는 위치이동(displacement)을 표현하기도 한다.



[벡터 덧셈]

- 대응되는 각 원소의 합으로 정의
- 2개의 변으로 구성된 평행사변형의 대각선 벡터



$$\begin{pmatrix}1\\3\end{pmatrix}+\begin{pmatrix}3\\1\end{pmatrix}=\begin{pmatrix}4\\4\end{pmatrix}$$

[벡터 덧셈]

- 대응되는 각 원소의 합으로 정의
- 2개의 변으로 구성된 평행사변형의 대각선 벡터

In	# 벡터의 덧셈
	a = np.array([1, 3])
	b = np.array([3, 1])
	a + b

Out	array([4, 4])

[벡터 덧셈]

- 교환 법칙: $a + b = b + a$
- 결합 법칙: $(a + b) + d = a + (b + d)$

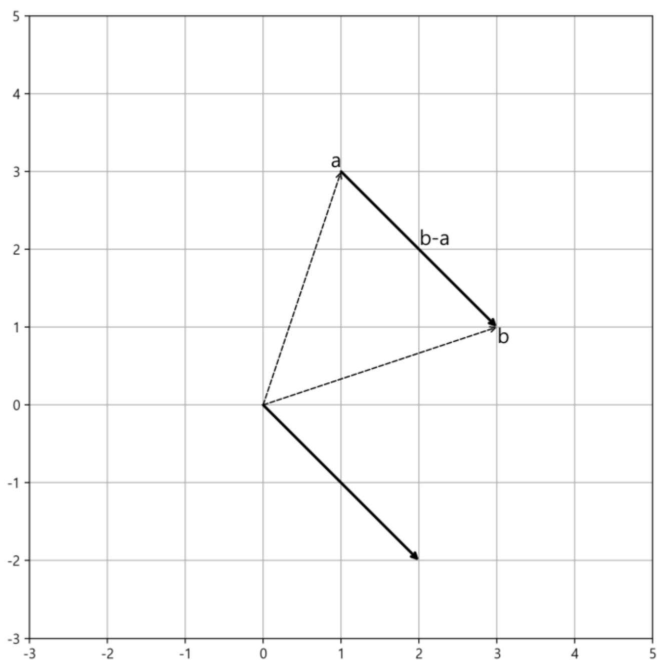
In	# 교환 법칙 a + b == b + a
Out	array([True, True])

In	# 결합 법칙
	d = np.array([1, 2])
	(a + b) + d == a + (b + d)

Out	array([True, True])

[벡터 뺄셈]

- 대응되는 각 원소의 차(difference)로 정의
- 빼는 벡터를 시작점으로 연결한 벡터



$$\begin{pmatrix}3\\1\end{pmatrix} - \begin{pmatrix}1\\3\end{pmatrix} = \begin{pmatrix}2\\-2\end{pmatrix}$$

[벡터 뺄셈]

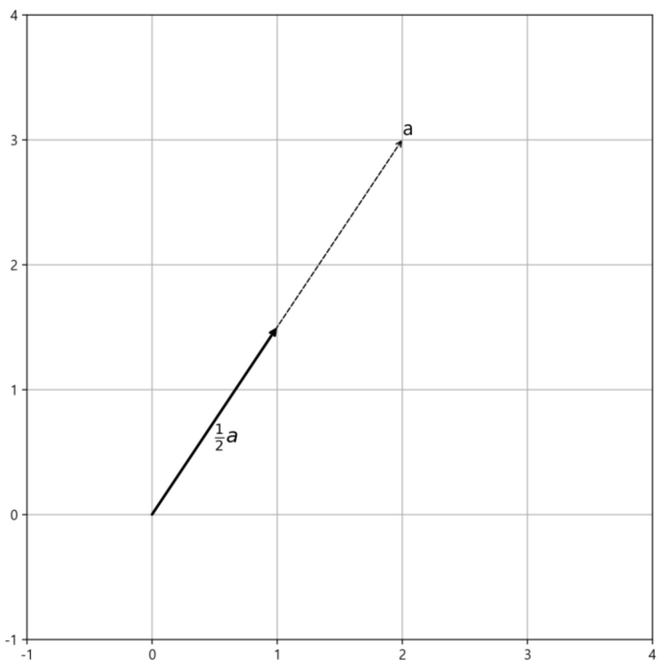
- 대응되는 각 원소의 차(difference)로 정의
- 빼는 벡터를 시작점으로 연결한 벡터

In	`# 벡터의 뺄셈`
	`a = np.array([3, 1])`
	`b = np.array([1, 3])`
	`b-a`

Out	`array([-2,  2])`

[스칼라 곱]

- 벡터에 대한 실수 곱(스칼라 곱: scalar multiplication)은 그 벡터의 모든 원소를 실수 만큼 곱한 것



$$\frac{1}{2}\begin{pmatrix}2\\3\end{pmatrix}=\begin{pmatrix}1\\1.5\end{pmatrix}$$

[스칼라 곱]

- 벡터에 대한 실수 곱(스칼라 곱: scalar multiplication)은 그 벡터의 모든 원소를 실수 만큼 곱한 것

In	`# 스칼라-벡터 곱`
	`alpha = 1/2`
	`a = np.array([2, 3])`
	`alpha * a`

Out	`array([1. , 1.5])`

[스칼라 곱]

- 스칼라 곱 연산의 성질
- 교환 법칙: $\beta a = a\beta$
- 분배 법칙: $(\alpha + \beta)a = \alpha a + \beta a$

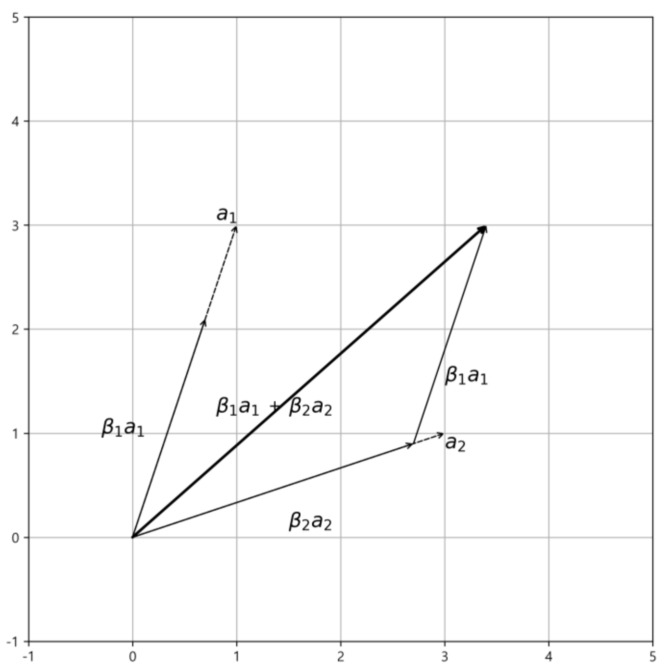
In	`# 교환 법칙` `alpha * a == a * alpha`
Out	`array([ True,  True])`

In	`# 분배 법칙`
	`beta = 0.7`
	`(alpha + beta) * a == alpha * a +beta * a`

Out	`array([ True,  True])`

[선형 결합]

- 벡터 $a_1, \dots, a_k$에 대한 선형결합(linear combination)
- $\beta_1 a_1 + \cdots + \beta_k a_k$



$$0.7 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 0.9 \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 3.4 \\ 3 \end{bmatrix}$$

과제	01
	1) 파이썬으로 1부터 10까지의 수로 구성된 벡터를 생성해보자.
	2) 파이썬으로 7번째 값만 1이고 나머지는 0인 e_7 벡터를 생성해보자.

벡터와 행렬

과제	02
	$\begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}3\\1\end{bmatrix}$ 벡터의 선형 결합으로 $\begin{bmatrix}2.5\\3.5\end{bmatrix}$ 벡터를 만들어보자. 즉, 적절한 β_1, β_2 값을 찾아보자

벡터와 벡터의 기초 연산

-끝-

벡터의 내적과 노름

[내적]

- 내적(inner product) 또는 점적(dot product)
- $a \cdot b = \langle a, b \rangle = a^T b = \sum_i a_i b_i$

$$\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}^T = (-1\ 2\ 3)$$

$$\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}^T \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} = sum \begin{pmatrix} -1 \\ -4 \\ 12 \end{pmatrix} = 7$$

[내적]

- 내적(inner product) 또는 점적(dot product)
- $a^T b = \sum_i a_i b_i$

In
```
# 내적 계산
a = np.array([-1, 2, 3])
b = np.array([1, -2, 4])

# 원소별 곱 벡터
a * b
```
Out
```
array([-1, -4, 12])
```

In
```
# 원소별 곱 벡터 합
np.sum(a*b)
```
Out
```
7
```

[내적]

- 내적(inner product) 또는 점적(dot product)
- $a^T b = \sum_i a_i b_i$

In	`# 함수 이용` `np.inner(a, b)`
Out	7

In	`np.dot(a, b)`
Out	7

[내적]

- 내적의 성질

1) $a^T b = b^T a$

2) $(\alpha a)^T b = \alpha(a^T b)$

3) $(a + b)^T c = a^T c + b^T c$

In
```
# 내적의 성질
a = np.array([-1, 2, 3])
b = np.array([1, -2, 4])

# 교환 법칙
np.dot(a, b) == np.dot(b, a)
```
Out
```
True
```

[내적]

- 내적의 성질

1) $a^T b = b^T a$

2) $(\alpha a)^T b = \alpha(a^T b)$

3) $(a + b)^T c = a^T c + b^T c$

In	`# 결합 법칙`
	`alpha= 0.5`
	`np.dot(alpha * a, b) == alpha * np.dot(a, b)`

Out	`True`

[내적]

- 내적의 성질

1) $a^T b = b^T a$

2) $(\alpha a)^T b = \alpha(a^T b)$

3) $(a + b)^T c = a^T c + b^T c$

In	`# 분배 법칙`
	`c = np.array([1, 2, 3])`
	`np.dot(a+b, c) == np.dot(a,c) + np.dot(b, c)`

Out	`True`

[내적]

- 주의 사항

벡터의 내적이 성립하기 위해서는 2개의 벡터의 크기가 같아야 한다.

$$\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}^T \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} \qquad \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}^T \begin{pmatrix} 1 \\ -2 \\ 4 \\ 7 \end{pmatrix}$$

[내적]

- 전치 연산자

T(전치: transpose) 연산자를 이용하여 내적을 정의

$$\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}^T \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} = (-1 \;\; 2 \;\; 3) \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} = -1 \times 1 + 2 \times (-2) + 3 \times 4 = 7$$

파이썬 프로그램: *

$$\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} * \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 \times 1 \\ 2 \times -2 \\ 3 \times 4 \end{pmatrix} = \begin{pmatrix} -1 \\ -4 \\ 12 \end{pmatrix}$$

[내적]

- 내적의 활용

1) 벡터의 원소 합: $1^T a = a_1 + \cdots + a_n = \sum_{i=1}^{n} a_i$

2) 벡터의 원소 평균: $\frac{1}{n}\ 1^T\ a = \frac{1}{n}(a_1 + \cdots + a_n) = \bar{a}$

3) 벡터의 원소 제곱합: $a^T a = a_1^2 + \cdots + a_n^2 = \sum_{i=1}^{n} a_i^2$

$$1 = \begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}, a = \begin{bmatrix} a_1 \\ \dots \\ a_n \end{bmatrix}$$

$$1^T a = [1 \quad \dots \quad 1] \begin{bmatrix} a_1 \\ \dots \\ a_n \end{bmatrix} = a_1 + \cdots + a_n$$

$$a^T a = [a_1 \quad \dots \quad a_n] \begin{bmatrix} a_1 \\ \dots \\ a_n \end{bmatrix} = a_1^2 + \cdots + a_n^2$$

[내적]

- 내적의 활용

1) 벡터의 원소 합: $1^T a = a_1 + \cdots + a_n = \sum_{i=1}^{n} a_i$

2) 벡터의 원소 평균: $\frac{1}{n} 1^T a = \frac{1}{n}(a_1 + \cdots + a_n) = \bar{a}$

3) 벡터의 원소 제곱합: $a^T a = a_1^2 + \cdots + a_n^2 = \sum_{i=1}^{n} a_i^2$

In
```
# 내적의 활용
a = np.array([1, 2, 3])
size = len(a)
ones = np.ones(shape=(size,))

# 합의 표현
np.dot(ones, a)
```
Out
```
6.0
```

[내적]

- 내적의 활용

1) 벡터의 원소 합: $1^T a = a_1 + \cdots + a_n = \sum_{i=1}^{n} a_i$

2) 벡터의 원소 평균: $\frac{1}{n} 1^T a = \frac{1}{n}(a_1 + \cdots + a_n) = \bar{a}$

3) 벡터의 원소 제곱합: $a^T a = a_1^2 + \cdots + a_n^2 = \sum_{i=1}^{n} a_i^2$

In	# 평균의 표현
	np.dot(ones, a) / len(a)

Out	2.0

[내적]

- 내적의 활용

1) 벡터의 원소 합: $1^T a = a_1 + \cdots + a_n = \sum_{i=1}^{n} a_i$

2) 벡터의 원소 평균: $\frac{1}{n}\, 1^T\, a = \frac{1}{n}(a_1 + \cdots + a_n) = \bar{a}$

3) 벡터의 원소 제곱합: $a^T a = a_1^2 + \cdots + a_n^2 = \sum_{i=1}^{n} a_i^2$

In	# 제곱합의 표현
	np.dot(a, a)

Out	14

[벡터 노름]

- 유클리디안 노름(Euclidean norm), L2 노름
- 기호: ‖ ‖
- $\|a\| = \sqrt{a^T a} = \sqrt{a_1^2 + \cdots + a_n^2}$

$$\left\| \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right\| = \sqrt{\begin{bmatrix} 2 \\ 3 \end{bmatrix}^T \begin{bmatrix} 2 \\ 3 \end{bmatrix}} = \sqrt{[2\ 3] \begin{bmatrix} 2 \\ 3 \end{bmatrix}} = \sqrt{13} = 3.606$$

[벡터 노름]

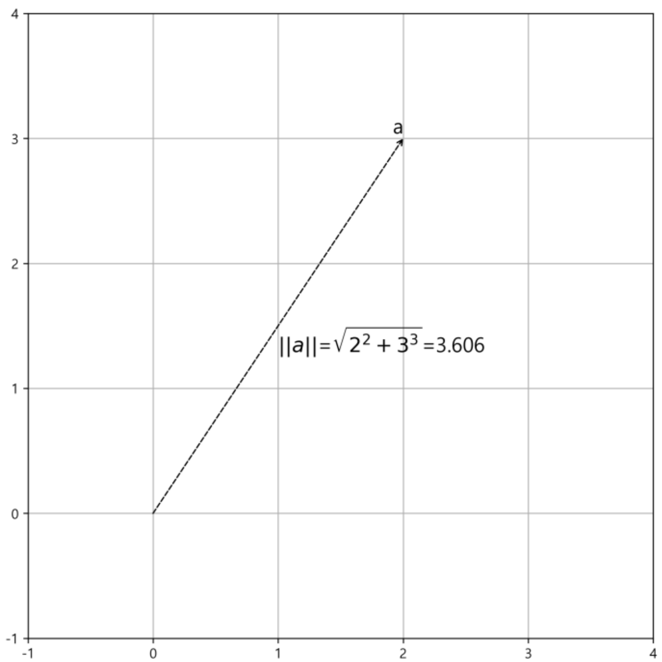
- 유클리디안 노름(Euclidean norm), L2 노름
- 기호: ‖ ‖
- $\|a\| = \sqrt{a^T a} = \sqrt{a_1^2 + \cdots + a_n^2}$

In	`# 필요한 패키지`
	`from scipy import linalg as la`
	
	`# 벡터 노름의 계산`
	`a = np.array([2, 3])`
	`la.norm(a)`

Out	3.606

[벡터 노름]

- 유클리디안 노름(Euclidean norm)의 의미
- $\|a\| = \sqrt{a^T a} = \sqrt{a_1^2 + \cdots + a_n^2}$



벡터와 행렬

과제	03
	1) 파이썬으로 벡터 $a^T = [1\ 2\ 3\ 4]$에 대하여 $\frac{a}{\|a\|}$ 값을 구해보자.
	2) 파이썬으로 $\frac{a}{\|a\|}$ 에 대하여 $\left\|\frac{a}{\|a\|}\right\|$ 의 값을 구해보자.

[벡터 노름]

- 노름의 성질

1) 스칼라 곱: $\|\beta a\| = |\beta|\|a\|$
2) 삼각형 법칙: $\|a + b\| \leq \|a\| + \|b\|$
3) 길이: $\|a\| \geq 0$
4) 길이 $= 0$: $\|a\| = 0 \Rightarrow a = 0$

In
```
# 벡터 노름의 성질
a = np.array([-1, 2, 3])
b = np.array([1, -2, 4])

# 스칼라 곱
beta = 0.5
la.norm(beta * a) == np.abs(beta) * la.norm(a)
```
Out
```
True
```

[벡터 노름]

- 노름의 성질

1) 스칼라 곱: $\|\beta a\| = |\beta|\|a\|$
2) 삼각형 법칙: $\|a + b\| \le \|a\| + \|b\|$
3) 길이: $\|a\| \ge 0$
4) 길이 $= 0$: $\|a\| = 0 \Rightarrow a = 0$

In	`# 삼각형의 한 변의 길이는 다른 두변의 길이의 합보다 작다` `la.norm(a + b) <= la.norm(a) + la.norm(b)`
Out	`True`

[벡터 노름]

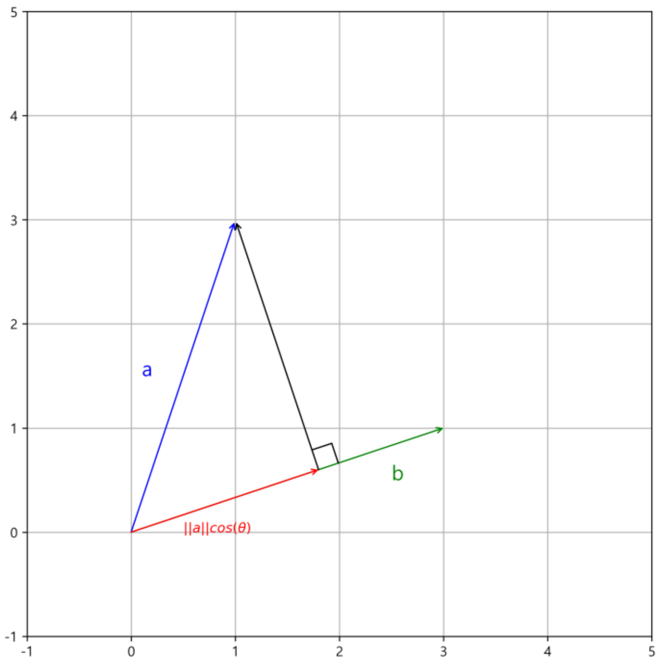
- 노름의 성질

1) 스칼라 곱: $\|\beta a\| = |\beta|\|a\|$
2) 삼각형 법칙: $\|a + b\| \leq \|a\| + \|b\|$
3) 길이: $\|a\| \geq 0$
4) 길이 $= 0$: $\|a\| = 0 \Rightarrow a = 0$

In
```
# 길이는 0보다 크거나 작다
la.norm(a) >= 0
```
Out
```
True
```

[내적과 노름]

- $a^T b = \|a\| \|b\| \cos(\theta)$
- $\cos(\theta)$=1 의 의미: 2개의 벡터가 일치 할 때



$$\cos(\theta) = \frac{a^T b}{\|a\| \|b\|}$$

[내적과 노름]

- $a^T b = \|a\| \|b\| \cos(\theta)$
- $\cos(\theta)$=1 의 의미: 2개의 벡터가 일치 할 때

In
```
# 코사인 theta 구하기
# 데이터 구성
a = np.array([1, 3])
b = np.array([3, 1])

cos_theta_rad = np.dot(a, b) / (la.norm(a) * la.norm(b))

# theta_rad(radian)값 구하기
theta_rad = np.arccos(cos_theta_rad)

# 각도로 전환: pi radian = 180 degree
theta = theta_rad * (180/np.pi)
[cos_theta_rad, theta]
```
Out
```
[0.5999999999999999, 53.13010235415599]
```

과제	04
	1) $[0\ 1]^T,\ [1\ 0]^T$으로 정의된 벡터 $\cos(\theta)$ 값과 두 벡터가 이루는 각도를 구해보자.
	2) $[-1\ \ -1]^T,\ [1\ 0]^T$으로 정의된 벡터 $\cos(\theta)$ 값과 두 벡터가 이루는 각도를 구해보자.

벡터의 내적과 노름

-끝-

[벡터 선형 종속]

- n 차원 벡터 $a_1, \dots, a_p$가 선형 종속(linear dependence)
- $\beta_1 a_1 + \cdots + \beta_p a_p = 0$ 가 0 벡터가 아닌 어떤 β들에 의하여 충족

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, a_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

$$\beta_1 = 1, \beta_2 = 1, \beta_3 = -1$$

$$\beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 = 0$$

$$a_3 = \frac{-\beta_1}{\beta_3} a_1 + \frac{-\beta_2}{\beta_3} a_2 = a_1 + a_2$$

다른 벡터들의 선형 결합으로 표현 가능

[벡터 선형 독립]

- n 차원 벡터 $a_1, \dots, a_p$가 선형 독립(linear independence)
- $\beta_1 a_1 + \cdots + \beta_p a_p = 0$ 이면 $\beta_i = 0, \forall i$

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, a_3 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

$$\beta_1 = 0, \beta_2 = 0, \beta_3 = 0$$

$$\beta_1 a_1 + \cdots + \beta_3 a_3 = 0$$

$$\begin{aligned} &\beta_1 - \beta_3 = 0 \\ &-\beta_2 = 0 \\ &\beta_3 = 0 \end{aligned}$$

[기저]

- 기저(base): n 차원에서 n개의 선형 독립인 벡터
- 2차원 벡터에서는 선형 독립인 벡터의 수는 최대 2개
- n차원 벡터 $a_1, \dots, a_n$가 기저이면, 임의의 n차원 벡터 x는

$$x = \beta_1 a_1 + \cdots + \beta_n a_n$$

3차원 기저의 예:

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, a_3 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, a_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

[기저]

- 기저(base): n 차원에서 n개의 선형 독립인 벡터
- 2차원 벡터에서는 선형 독립인 벡터의 수는 최대 2개
- n차원 벡터 $a_1, \dots, a_n$가 기저이면, 임의의 n차원 벡터 x는

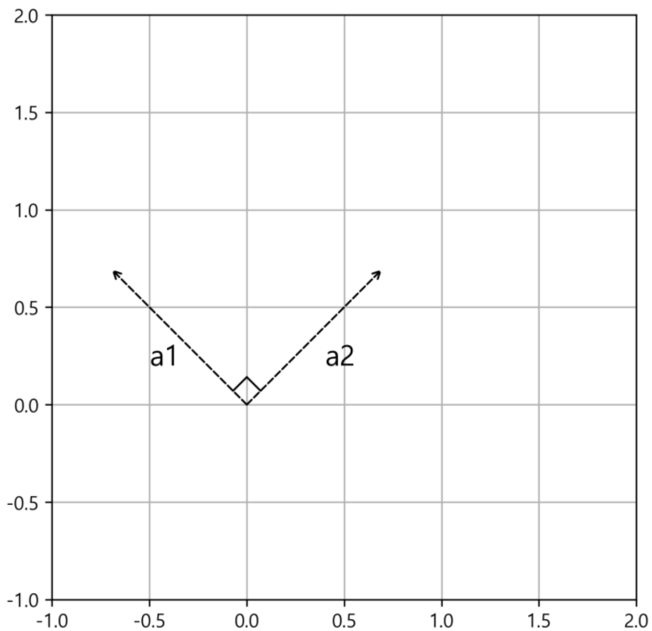
$$x = \beta_1 a_1 + \cdots + \beta_n a_n$$

2차원 기저의 예:

$$a_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, a_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, x = \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} \qquad \rightarrow x = \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} = 0.7a_1 + 0.3a_2$$

[직교정규(orthonormal) 벡터]

- $a_i \perp a_j$
- $a_i^T a_j = 0, \|a_i\| = 1, \forall i$



$$a_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}, a_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

$$\rightarrow a_1^T a_2 = \frac{1}{\sqrt{2}}[-1\ 1]\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} = 0$$

과제	05
	1) $[1\ \ 2\ \ 3]^T, [1\ 1\ 1]^T, [-1\ \ -1\ \ -1]^T$ 로 구성된 벡터는 서로 독립인지를 확인해보자.
	2) 2차원 즉, 평면에서의 기저를 찾아보자.
	3) $a_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, a_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, a_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$가 서로 직교정규 벡터임을 확인해보자.

벡터

-끝-