

주요 개념

스칼라

벡터

행렬

텐서

벡터의 내적

벡터 끼리 곱의 한 종류

벡터의 놈

정칙화

행렬의 곱

아다마르 곱(요소별 곱)

전치

역행렬

단위행렬

정방행렬

행렬식

선형변환

표준기저

고윳값과 고유벡터

고유방정식

코사인 유사도

스칼라 / 벡터 / 행렬 / 텐서 기본 개념

텐서는 스칼라를 여러 개의 차원으로 나열한 것 스칼라, 벡터, 행렬을 포함한다.

텐서



사용 함수 : array()

벡터의 내적

내적 : 벡터 끼리 곱의 한 종류
각 요소끼리 곱한 값을 총합
조건: 두 개의 벡터의 요소 수가 같아야 한다.



두 개 데이터의 상관관계를 구할 때 등에 사용

```
1  # 리스트 4.6_벡터의 내적을 계산한다.
2  import numpy as np
3
4  a = np.array([1,2,3])
5  b = np.array([3,2,1])
6
7  print('-----dot함수-----')
8  print(np.dot(a,b))
9  print()
10 print('-----sum함수-----')
11 print(sum(a*b))
```

사용 함수 : dot()

Sum()

벡터의 놈

놈 : 벡터의 '크기'를 나타내는 양

L2놈 `np.linalg.norm()` --> 디폴트가 L2인 함수
벡터의 각 요소를 제곱합하여 제곱근을 구함

사용 함수 : `np.linalg.norm()`

L1놈 `np.linalg.norm(,1)` --> 1을 줘서 L1놈 구함
벡터의 각 요소의 절댓값을 더해서 계산

놈은 인공지능에서 정칙화에 쓰인다.
정칙화란 필요 이상으로 네트워크 학습이 진행되는 것을
파라미터로 조절해서 예방하는 것

*linalg*는 *Linear Algebra*라는 선형 대수학의 준말

행렬의 곱 vs 아다마르 곱(요소별 곱)

```
12 print('-----요소별 곱하기-----')
13 print(a*b)
14 print()
15 print('-----요소별 나누기-----')
16 print(a/b)
17 print()
18 print('-----요소별 더하기-----')
19 print(a+b)
20 print()
21 print('-----요소별 빼기-----')
22 print(a-b)
```

행렬 곱 사용 함수 : dot()

아다마르 곱 사용 함수 : 일반 연산자

전치

하는 법: 배열 이름.T

a=

1	2	3	4
---	---	---	---



a.T=

1
2
3
4

전치를 통해
앞 행렬의 열수와 뒤 행렬의 행수를 일치시켜
곱을 가능하게 할 수도 있다.

```
1 # 리스트 4.14_ 행렬을 전치한다.
2 import numpy as np
3
4 a = np.array([[1,2,3],
5               [4,5,6]]) # 2x3의 행렬
6 print(a)
7 print()
8 print(a.T) # 원본을 바꾸진 않는다.
9 print()
10 print(a)
```

$\begin{bmatrix} [1 & 2 & 3] \\ [4 & 5 & 6] \end{bmatrix}$

$\begin{bmatrix} [1 & 4] \\ [2 & 5] \\ [3 & 6] \end{bmatrix}$

$\begin{bmatrix} [1 & 2 & 3] \\ [4 & 5 & 6] \end{bmatrix}$

단위 행렬(E) 사용 함수 : eye()

정수에서 1처럼 곱했을 때 자기 자신이 나오도록 하는 행렬

같은 크기의 행렬에 곱해도 곱하는 대상의 행렬을 변화시키지 않는 성질이 있다.
행과 열 수가 같은 정방행렬의 경우에만 가능하다.

역행렬과 행렬식

A라는 임의의 정방행렬과 곱했을 때 **E**(단위행렬)이 나오도록 하는 행렬 → 사용 함수 : `linalg.inv()`

조건: **행렬식=0**이 아니어야 한다. → 사용 함수 : `np.linalg.det()`

즉. `np.linalg.det()`의 값이 0이 아니어야 한다.

역행렬은 인공지능에서 변수끼리의 상관관계를 알아보는 회귀분석에 활용된다.

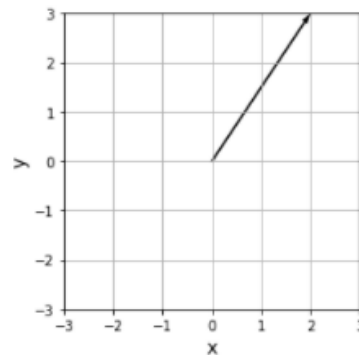
inv는 inverse의 약자

벡터 그리기 사용 함수 : quiver()

quiver(시작점의 x좌표, 끝점의 y좌표, 화살표의 x성분, 화살표의 y성분, angles=화살표의 각도의 결정방법, scale_units=스케일의 단위, scale=스케일, color=화살표의 색)

```
1 import numpy
2 import matplotlib.pyplot as plt
3
4 # 화살표를 그리는 함수 만들기
5 def arrow(start, size, color) :
6     plt.quiver(start[0], start[1], size[0], size[1], |
7               angles="xy", scale_units="xy", scale=1, color=color)
8
9
10 # 화살표의 시작점
11 s = np.array([0,0]) # 원점
12
13 # 벡터
14 a = np.array([2,3])
15
16 arrow(s, a, color='black')
17
18 # 그래프 표시
19 plt.xlim([-3,3])
20 plt.ylim([-3,3])
21 plt.xlabel("x", size=14)
22 plt.ylabel("y", size=14)
23 plt.grid()
24 plt.gca().set_aspect("equal")
25 plt.show
```

<function matplotlib.pyplot.show(*args, **kw)>



선형변환

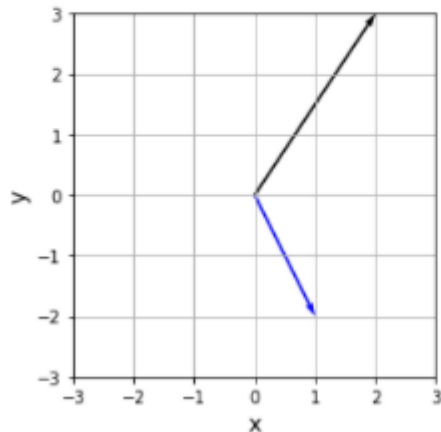
벡터에서 벡터로의 변환을 선형변환이라고 한다.

인공지능에서 뉴럴 네트워크로 정보를 전파시키는데 선형변환을 사용

A가 정방 행렬이 아니면 선형변환에 의해 벡터의 요소가 변하게 된다.(p154)

변환 전 벡터: [2 3]
변환 후 벡터: [1 -2]

Out[6]: <function matplotlib.pyplot.show(*args, **kw)>



```
1 import numpy
2 import matplotlib.pyplot as plt
3
4
5 # 변환 전 벡터
6 a = np.array([2,3])
7
8 A = np.array([[2,-1],[2,-2]])
9
10 # 선형변환
11 b = np.dot(A, a)
12
13 print('변환 전 벡터:', a)
14 print('변환 후 벡터:', b)
15
16 # 화살표를 그리는 함수 만들기
17 def arrow(start, size, color):
18     plt.quiver(start[0], start[1], size[0], size[1],
19               angles="xy", scale_units="xy", scale=1, color=color)
20
21 # 화살표의 시작점
22 s = np.array([0,0]) # 원점
23
24 arrow(s, a, color='black')
25 arrow(s, b, color='blue')
26
27 # 그래프 표시
28 plt.xlim([-3,3])
29 plt.ylim([-3,3])
30 plt.xlabel("x", size=14)
31 plt.ylabel("y", size=14)
32 plt.grid()
33 plt.gca().set_aspect("equal")
34 plt.show
```

표준기저

벡터는 표준기저와 상수의 곱의 합으로 표현할 수 있다.

고윳값과 고유벡터의 계산_ 고유방정식 사용함수: `linalg.eig()`

인공지능에서 데이터를 요약하는 주성분 분석이라는 기법에 이용된다.

고윳값과 고유벡터의 계산_ 고유방정식

사용함수: `linalg.eig()`

인공지능에서 데이터를 요약하는 주성분 분석이라는 기법에 이용된다.

```
1 # 리스트 4.28 linalg.eig() 함수를 사용해 고윳값 고유벡터를 구한다.
2 import numpy as np
3
4 a = np.array([[1,2],
5               [3,4]]) # 2x3의 행렬
6
7 ev = np.linalg.eig(a)
8
9 # 0번 인덱스 요소는 고윳값
10 print(ev[0])
11
12 print()
13
14 # 1번 인덱스 요소는 고유벡터
15 print(ev[1])
```

```
[-0.37228132  5.37228132]
```

```
[[-0.82456484 -0.41597356]
 [ 0.56576746 -0.90937671]]
```

각 열이 고유벡터를 나타낸다.

각 고유벡터는 L2노름이 1이 된다. 고유벡터는 L2노름이 1인 벡터를 단위 벡터라고 한다.

코사인 유사도

사용함수: `np.dot() / np.linalg.norm()`

벡터끼리의 방향의 가까운 정도.

2개 벡터의 방향이 얼마나 일치하고 있는지를 나타내는 지표

$\cos\theta$ 의 값은

$\theta = 0 \rightarrow \cos\theta = 1$ ($\cos\theta$ 의 최댓값) \rightarrow 유사도가 동일

$\theta \uparrow \rightarrow \cos\theta$ 의 값은 작아진다. \rightarrow 유사도가 작아진다.

$\cos\theta = -1$ ($\cos\theta$ 의 최솟값) \rightarrow 반대