[Python] Numpy를 통한 배열 연산 — 나무늘보의 개발 블로그

노트북:	첫 번째 노트북
만든 날짜:	2020-10-30 오후 9:46
URL:	https://continuous-development.tistory.com/122?category=736681

Python

[Python] Numpy를 통한 배열 연산

2020. 10. 12. 22:57 수정 삭제 공개

배열의 연산

- vector operation(명시적으로 반복문을 사용하지 않더라도 모든 원소에 대해서 연산 가능)

In [56]:
```
x = np.arange(1,10001)
y = np.arange(10001, 20001)
```

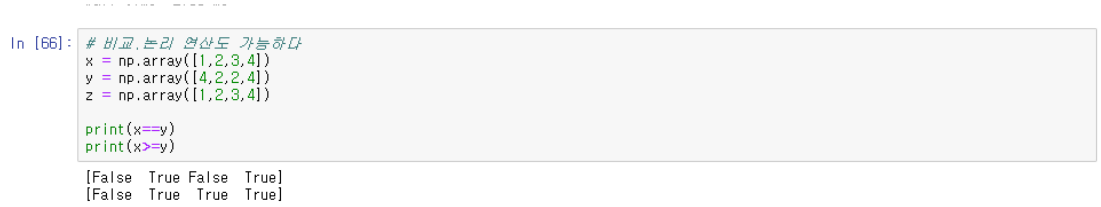
In [63]:
```
# 일반적인 계산
%%time
z = np.zeros_like(x)
print(z)
for i in range(10000):
    z[i] = x[i] + y[i]

print(z[:10])
```

```
[0 0 0 ... 0 0 0]
[10002 10004 10006 10008 10010 10012 10014 10016 10018 10020]
Wall time: 12 ms
```

In [64]:
```
# numpy 를 통한 계산
%%time
z = x + y
print(z[:10])
```

```
[10002 10004 10006 10008 10010 10012 10014 10016 10018 10020]
Wall time: 2.99 ms
```

시간이 훨씬 빠른 걸 볼 수 있다.

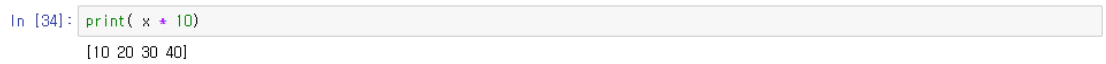


```
In [66]: # 비교,논리 연산도 가능하다
         x = np.array([1,2,3,4])
         y = np.array([4,2,2,4])
         z = np.array([1,2,3,4])

         print(x==y)
         print(x>=y)

         [False  True False  True]
         [False  True  True  True]
```

비교 논리 연산도 가능하다.

```
In [68]: # 배열의 모든 원소가 같은지 다른지를 판단하고 싶다면?
         print(np.all(x==y))
         print(np.all(x==z))

         False
         True
```

두 원소가 같은지에 대해 비교 연산도 가능하다.



```
In [34]: print( x * 10)

         [10 20 30 40]
```

기본적인 덧셈 연산도 가능하다. 원래 행열 계산에서는 길이가 맞아야 되는데 이렇게 길이가 안 맞아도 계산이 가능한 이유는 brodcasting 때문이다.

brodcasting

행렬 또는 백터에 덧셈 또는 뺄셈을 하려면 원래는 크기가 같아야 한다.
하지만 numpy에서는 brodcasting 을 통해
크기를 자동으로 맞춰준다.

벡터(또는 행렬)끼리 덧셈 혹은 뺄셈을 하려면 두 벡터(또는 행렬)의 크기가 같아야 한다. 넘파이에서는 서로 다른 크기를 가진 두 배열의 사칙 연산도 지원한다. 이 기능을 브로드캐스팅(broadcasting)이라고 하는데 크기가 작은 배열을 자동으로 반복 확장하여 크기가 큰 배열에 맞추는 방법이다.

예를 들어 다음과 같이 벡터와 스칼라를 더하는 경우를 생각하자.

$$x = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad x + 1 = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} + 1 = ?$$

브로드캐스팅은 다음과 같이 스칼라를 벡터와 같은 크기로 확장시켜서 덧셈 계산을 하는 것이다.

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

예를 들어 선형 대수에서 두 벡터의 합은 다음과 같이 구한다.

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ 10000 \end{bmatrix}, \quad y = \begin{bmatrix} 10001 \\ 10002 \\ 10003 \\ \vdots \\ 20000 \end{bmatrix},$$

일 때, 두 벡터의 합

$$z = x + y$$

은 다음과 같이 구한다.

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ 10000 \end{bmatrix} + \begin{bmatrix} 10001 \\ 10002 \\ 10003 \\ \vdots \\ 20000 \end{bmatrix} = \begin{bmatrix} 1 + 10001 \\ 2 + 10002 \\ 3 + 10003 \\ \vdots \\ 10000 + 20000 \end{bmatrix} = \begin{bmatrix} 10002 \\ 10004 \\ 10006 \\ \vdots \\ 30000 \end{bmatrix}$$

브로드캐스팅은 다음처럼 더 차원이 높은 경우에도 적용된다.

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

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

```
In [32]: x = np.vstack([range(7)[i:i+3]for i in range(5)])
         aryInfo(x)

type : <class 'numpy.ndarray'>
shape : (5, 3)
dimension : 2
dtype : int32
Array Data :
 [[0 1 2]
 [1 2 3]
 [2 3 4]
 [3 4 5]
 [4 5 6]]
```

2차원의 5행 3열의 행렬이 있고

```
In [33]: y = np.arange(5)[:,np.newaxis]
         aryInfo(y)

type : <class 'numpy.ndarray'>
shape : (5, 1)
dimension : 2
dtype : int32
Array Data :
 [[0]
 [1]
 [2]
 [3]
 [4]]
```

2차원의 5행 1열의 행렬이 있었을 때

```
In [34]: x+y
Out[34]: array([[ 0,  1,  2],
                [ 2,  3,  4],
                [ 4,  5,  6],
                [ 6,  7,  8],
                [ 8,  9, 10]])
```

이렇게 계산할 수 있다.

'Python' 카테고리의 다른 글□

[Python] Numpy를 통한 정렬하기□

[Python] Numpy 를 통한 최대값, 최소값 , 통계함수 사용하기□

[Python] Numpy를 통한 배열 연산□

[Python] Numpy의 배열 행 열 삭제□

[Python] Numpy 배열 합치기(concatenate)□

[Python] Numpy의 reshape 통한 차원 변경(재배열)□

Numpy 배열 덧셈 Numpy 배열 연산 Numpy 행렬계산

나무늘보스

혼자 끄적끄적하는 블로그 입니다.