

노트북: 첫 번째 노트북

만든 날짜: 2020-10-29 오전 9:13

URL: <https://continuous-development.tistory.com/119?category=736681>

Python

[Python] Numpy의 reshape 통한 차원 변경 (재배열)

2020. 10. 12. 19:16 수정 삭제 공개

차원 바꾸기

reshape

reshape를 통해 차원을 지정해줄 수 있다.

```
In [4]: import numpy as np
```

기본적으로 Numpy를 import를 해주자.

```
In [5]: def aryInfo(ary):  
        print('type : {}'.format(type(ary)))  
        print('shape : {}'.format(ary.shape))  
        print('dimension : {}'.format(ary.ndim))  
        print('dtype : {}'.format(ary.dtype))  
        print('Array Data : \n',ary)
```

이 함수는 변수의 형태를 보기 위해서 만든 함수이다.

```
In [7]: arr = np.arange(0,12,1)
        aryInfo(arr)

type : <class 'numpy.ndarray'>
shape : (12,)
dimension : 1
dtype : int32
Array Data :
[ 0  1  2  3  4  5  6  7  8  9 10 11]
```

이렇게 1차원인 np가 있다. 이 값을 형태를 변경해보자.

```
In [9]: arr1 = arr.reshape(4,3)
        aryInfo(arr1)

type : <class 'numpy.ndarray'>
shape : (4, 3)
dimension : 2
dtype : int32
Array Data :
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

위와 같이 **변수명.reshape**를 작성하고 내가 원하는 형태의 행열의 값을 입력한다. 여기서는 4,3이라고 작성하면 4행 3열의 2차원 np가 생긴다.

```
In [10]: arr2 = arr.reshape(2,2,-1)
         aryInfo(arr2)

type : <class 'numpy.ndarray'>
shape : (2, 2, 3)
dimension : 3
dtype : int32
Array Data :
[[[ 0  1  2]
  [ 3  4  5]]
 [[ 6  7  8]
  [ 9 10 11]]]
```

또한 위와같이 2,2,-1라고 작성하면 2가지의 차원 / 2행 / -1이라는 것은 나머지를 맞춰서 넣는다는 뜻으로 12개의 값들이 앞에서 선행되었던 차원과 행에 맞춰서 들어간다. 이렇게 3차원의 np가 생긴다.

flatten() , ravel()

차원에 대해서 무조건 1차원을 하기 위해서는 flatten() 과 ravel()을 사용한다.

```
In [14]: arr1 = arr.reshape(5,1)
          aryInfo(arr1)

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

위 같은 형태의 배열이 있을 때

```
In [15]: arr1.flatten()
Out [15]: array([0, 1, 2, 3, 4])
```

```
In [17]: arr1.ravel()
Out [17]: array([0, 1, 2, 3, 4])
```

flatten 과 ravel을 이렇게 사용한다.

```
In [18]: arr2 = arr.reshape(1,5).copy()
          aryInfo(arr2)

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

만약 원본과 다르게 새 값을 만들기 위해서는 copy() 명령어를 쓴다.

'Python' 카테고리의 다른 글

[Python] Numpy의 배열 행 열 삭제

[Python] Numpy 배열 합치기(concatenate)

[Python] Numpy의 reshape 통한 차원 변경(재배열)

[Python] Numpy에 있는 다양한 함수 사용법 - 2(전치행렬,zeors,ones, iterator,...

[Python] Numpy를 통한 배열 indexing(Boolen indexing, fancy indexing)

[Python] Numpy에 대한 기초 정리와 사용법 정리

numpy reshape

Numpy 차원 배열

Numpy 차원 변경



나무늘보스

혼자 끄적끄적하는 블로그 입니다.