

## python随笔1 numpy array, tile, newaxis(09.11.2020)



三思的韦小宝

努力成为新时代的工程师

### python 随笔(1): numpy array, tile, newaxis

#### numpy.array()

举例

```
import numpy as np

c = np.array([[[1,2,3,4],[5,6,7,8]],
              [[9,10,11,12],[13,14,15,16]],
              [[17,18,19,20],[21,22,23,24]]])
```

创建了一个三维数组（有个小窍门，看array(后边的[有几个就是几维数组。此处c = np.array([[[有三个[所以是三维数组)。c的维度是3, shape是(3,2,4)。注意最先定义的维度[1,2,3,4], 其实变成了c中最后的一个维度,

比如:

```
c.ndim
c.shape
```

输出:

```
3
(3, 2, 4)
c[0,0,:]
```

输出:

```
array([1, 2, 3, 4])
```

而

```
c[1,:,1]
```

输出:

```
array([10, 14])
```

#### numpy.tile(c, reps)

tile 将数组c 重复reps 次, 这里reps 是个一维数组

例如:

```
import numpy as np
c = np.array([1,2,3])
a = np.tile(c,2)
```

输出:

```
c
array([1, 2, 3])
a
array([1, 2, 3, 1, 2, 3])
```

上例中, c是一维数组 (c.ndim=1) , 而reps是一个数字2, 当reps是一维数组时:

```
import numpy as np
c = np.array([1,2,3])
a = np.tile(c,(2,3))
# c: array([1, 2, 3])
# a: array([[1, 2, 3, 1, 2, 3, 1, 2, 3],
#          [1, 2, 3, 1, 2, 3, 1, 2, 3]])
```

注意a首先在第一个维度上重复2次, 然后在第二个维度上重复3次。

当c也是一个数组时, 结果类似:

```
import numpy as np
c = np.array([[1,2,3],
              [4,5,6]])
a = np.tile(c,(2,3))
# c:array([[1, 2, 3],
#          [4, 5, 6]])
# a:array([[1, 2, 3, 1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6, 4, 5, 6],
#          [1, 2, 3, 1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6, 4, 5, 6]])
```

tile的英文是“铺瓷砖”, 从上面例子确实可以看出这个意思

如果reps的长度大于c的维度时候 (len(reps)>c.ndim) ,

```
import numpy as np
c = np.array([[1,2,3],
              [4,5,6]])
a = np.tile(c,(2,3,2))

#a:array([[[1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6],
#          [1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6],
#          [1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6]]],
#         [[1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6],
#          [1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6],
#          [1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6]]]])
```

则在a的第三个维度上再将c重复两次。

当 (len(reps)<c.ndim) ,程序会自动在reps前边补上1, 一直到 len(reps) = c.dim时, 再进行tile, 例如:

```
import numpy as np
c = np.array([[1,2,3],
              [4,5,6]])
a = np.tile(c,(3))
# a: array([[1, 2, 3, 1, 2, 3, 1, 2, 3],
#          [4, 5, 6, 4, 5, 6, 4, 5, 6]])
```

## numpy.newaxis

np.newaxis 是None的一个别名，一般用来扩展数组的维度

```
import numpy as np
c = np.array([1,2,3])
c.shape
# (3,)
d = c[:,np.newaxis] #利用np.newaxis 将数组c扩展为2维
d.shape
# (3, 1)
d
# array([[1],
#        [2],
#        [3]])
d = c[:,np.newaxis,np.newaxis] #利用np.newaxis 将数组c扩展为3维
d.shape
# (3, 1, 1)
d
# array([[[1]],
#        [[2]],
#        [[3]])
```

类似的None也可以达到同样的效果

```
import numpy as np
c = np.array([1,2,3])
c.shape
# (3,)
d = c[:,None]
d.shape
# (3, 1)
d
# array([[1],
#        [2],
#        [3]])
d = c[:,None,None]
d.shape
# (3, 1, 1)
d
# array([[[1]],
#        [[2]],
#        [[3]])
```

## 综合举例

利用np.tile和np.newaxis将数组c扩展为矩阵：

```
import numpy as np
c = np.array([1,2,3])
D = np.tile(c[:,np.newaxis],(1,3))
D
# array([[1, 1, 1],
#        [2, 2, 2],
#        [3, 3, 3]])
```

以上代码创立了一个 $3 \times 3$ 的 矩阵D, D的每一列都是c。

编辑于 2020-09-12

[Python 入门](#)

▲ 赞同 ▼    ● 添加评论    ↗ 分享    ♥ 喜欢    ★ 收藏    ...