

NumPy 高级索引

NumPy 比一般的 Python 序列提供更多的索引方式。除了之前看到的用整数和切片的索引外，数组可以由整数数组索引、布尔索引及花式索引。

整数数组索引

以下实例获取数组中(0,0)，(1,1)和(2,0)位置处的元素。

实例

```
import numpy as np
x = np.array([[1, 2], [3, 4], [5, 6]])
y = x[[0,1,2], [0,1,0]]
print (y)
```

输出结果为：

```
[1  4  5]
```

以下实例获取了 4X3 数组中的四个角的元素。行索引是 [0,0] 和 [3,3]，而列索引是 [0,2] 和 [0,2]。

实例

```
import numpy as np
x = np.array([[ 0, 1, 2],[ 3, 4, 5],[ 6, 7, 8],[ 9, 10, 11]])
print ('我们的数组是：')
print (x)
print ('\n')
rows = np.array([[0,0],[3,3]])
cols = np.array([[0,2],[0,2]])
y = x[rows,cols]
print ('这个数组的四个角元素是：')
print (y)
```

输出结果为：

我们的数组是：

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

这个数组的四个角元素是：

```
[[ 0  2]
 [ 9 11]]
```

返回的结果是包含每个角元素的 ndarray 对象。

可以借助切片：或 ... 与索引数组组合。如下面例子：

实例

```
import numpy as np
a = np.array([[1,2,3], [4,5,6],[7,8,9]])
b = a[1:3, 1:3]
c = a[1:3,[1,2]]
d = a[...,[1:]]
print(b)
print(c)
print(d)
```

输出结果为：

```
[[5 6]
 [8 9]]
[[5 6]
 [8 9]]
[[2 3]
 [5 6]
 [8 9]]
```

布尔索引

我们可以通过一个布尔数组来索引目标数组。

布尔索引通过布尔运算（如：比较运算符）来获取符合指定条件的元素的数组。

以下实例获取大于 5 的元素：

实例

```
import numpy as np
x = np.array([[ 0, 1, 2],[ 3, 4, 5],[ 6, 7, 8],[ 9, 10, 11]])
print ('我们的数组是:')
print (x)
print ('\n')
# 现在我们会打印出大于 5 的元素
print ('大于 5 的元素是:')
print (x[x > 5])
```

输出结果为：

```
我们的数组是:
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

大于 5 的元素是：

```
[ 6  7  8  9 10 11]
```

以下实例使用了 `~`（取补运算符）来过滤 NaN。

实例

```
import numpy as np
a = np.array([np.nan, 1,2,np.nan,3,4,5])
print (a[~np.isnan(a)])
```

输出结果为：

```
[ 1.  2.  3.  4.  5.]
```

以下实例演示如何从数组中过滤掉非复数元素。

实例

```
import numpy as np
a = np.array([1, 2+6j, 5, 3.5+5j])
print (a[np.iscomplex(a)])
```

输出如下：

```
[2.0+6.j  3.5+5.j]
```

花式索引

花式索引指的是利用整数数组进行索引。

花式索引根据索引数组的值作为目标数组的某个轴的下标来取值。对于使用一维整型数组作为索引，如果目标是一维数组，那么索引的结果就是对应位置的元素；如果目标是二维数组，那么就是对应下标的行。

花式索引跟切片不一样，它总是将数据复制到新数组中。

1、传入顺序索引数组

实例

```
import numpy as np
x=np.arange(32).reshape((8,4))
print (x[[4,2,1,7]])
```

输出结果为：

```
[[16 17 18 19]
 [ 8  9 10 11]
 [ 4  5  6  7]
 [28 29 30 31]]
```

2、传入倒序索引数组

实例

```
import numpy as np
x=np.arange(32).reshape((8,4))
print (x[[-4,-2,-1,-7]])
```

输出结果为：

```
[[16 17 18 19]
 [24 25 26 27]
 [28 29 30 31]
 [ 4  5  6  7]]
```

3、传入多个索引数组（要使用np.ix_）

实例

```
import numpy as np
x=np.arange(32).reshape((8,4))
print (x[np.ix_([1,5,7,2],[0,3,1,2])])
```

输出结果为：

```
[[ 4  7  5  6]
 [20 23 21 22]
 [28 31 29 30]
 [ 8 11  9 10]]
```

← NumPy 切片和索引

NumPy 广播(Broadcast) →

 点我分享笔记

