Pre

2023 年 5 月 7 日

[ ]:
```
import numpy as np
a = np.array([[0,1,2,3],[4,5,6,7],[8,9,10,11]])
a
```

[ ]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

[ ]:
```
a.dtype
```

[ ]:
```
dtype('int64')
```

[ ]:
```
a[1,2]
```

[ ]:
```
6
```

[ ]:
```
a[:,1:3]
```

[ ]:
```
array([[ 1,  2],
       [ 5,  6],
       [ 9, 10]])
```

[ ]:
```
a.ndim
```

[ ]:
```
2
```

[ ]:
```
a.shape
```

[ ]:
```
(3, 4)
```

[ ]:
```
a.strides
```

[ ]:
```
(32, 8)
```

[ ]:
```
b = a.reshape(4, 3)
b
```

[ ]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])
```

[ ]:
```
# reshape 操作产生的是 view 视图，只是对数据的解释方式发生变化，数据物理地址相同
a.ctypes.data
```

[ ]:
```
140533973156400
```

[ ]:
```
b.ctypes.data
```

[ ]:
```
140533973156400
```

[ ]:
```
id(a) == id(b)
```

[ ]:
```
False
```

[ ]:
```
# 数据在内存中连续存储
from ctypes import string_at
string_at(b.ctypes.data, b.nbytes).hex()
```

[ ]:
```
'000000000000000001000000000000000200000000000000030000000000000004000000000000000500000000000000060000000000000007000000000000000800000000000000090000000000000000a000000000000000b00000000000000'
```

[ ]:
```
# b 的转置 c，c 仍共享相同的数据 block，只改变了数据的解释方式，“以列优先的方式解释行优先的存储”
c = b.T
c
```

[ ]:
```
array([[ 0,  3,  6,  9],
       [ 1,  4,  7, 10],
       [ 2,  5,  8, 11]])
```

[ ]:
```
c.ctypes.data
```

[ ]:
```
140533973156400
```

[ ]:
```
string_at(c.ctypes.data, c.nbytes).hex()
```

[ ]:
```
'00000000000000000100000000000000020000000000000003000000000000000400000000000000
00500000000000000060000000000000070000000000000008000000000000000900000000000000
00a000000000000000b00000000000000'
```

[ ]:
```
a
```

[ ]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

[ ]:
```
# copy 会复制一份新的数据，其物理地址位于不同的区域
c = b.copy()
c
```

[ ]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])
```

[ ]:
```
c.ctypes.data
```

[ ]:
```
140533159951984
```

[ ]:
```
string_at(c.ctypes.data, c.nbytes).hex()
```

[ ]:
```
'00000000000000000100000000000000020000000000000003000000000000000400000000000000
00500000000000000060000000000000070000000000000008000000000000000900000000000000
00a000000000000000b00000000000000'
```

[ ]:
```
# slice 操作产生的也是 view 视图，仍指向原来数据 block 中的物理地址
d = b[1:3, :]
d
```

[ ]:
```
array([[3, 4, 5],
       [6, 7, 8]])
```

[ ]:
```
d.ctypes.data
```

[ ]:
```
140533973156424
```

[ ]:
```
print('data buff address from {0} to {1}'.format(b.ctypes.data, b.ctypes.data +␣
↪b.nbytes))
```

```
data buff address from 140533973156400 to 140533973156496
```