Numpy中的arg运算

argmin
argmax
argsort
argpartition

```
In [1]: import numpy as np
```

```
In [2]: np.random.seed(100)
        x = np.random.random(50)
        x
```

```
Out[2]: array([0.54340494, 0.27836939, 0.42451759, 0.84477613, 0.00471886,
               0.12156912, 0.67074908, 0.82585276, 0.13670659, 0.57509333,
               0.89132195, 0.20920212, 0.18532822, 0.10837689, 0.21969749,
               0.97862378, 0.81168315, 0.17194101, 0.81622475, 0.27407375,
               0.43170418, 0.94002982, 0.81764938, 0.33611195, 0.17541045,
               0.37283205, 0.00568851, 0.25242635, 0.79566251, 0.01525497,
               0.59884338, 0.60380454, 0.10514769, 0.38194344, 0.03647606,
               0.89041156, 0.98092086, 0.05994199, 0.89054594, 0.5769015 ,
               0.74247969, 0.63018394, 0.58184219, 0.02043913, 0.21002658,
               0.54468488, 0.76911517, 0.25069523, 0.28589569, 0.85239509]
        )
```

```
In [3]: np.min(x)
```

```
Out[3]: 0.004718856190972565
```

```
In [4]: np.argmin(x)
```

```
Out[4]: 4
```

```
In [5]: x[4]
```

```
Out[5]: 0.004718856190972565
```

```
In [6]: np.max(x)
```

```
Out[6]: 0.9809208570123115
```

```
In [7]: np.argmax(x)
```

```
Out[7]: 36
```

```
In [8]: x[36]
```

```
Out[8]: 0.9809208570123115
```

argwhere可以查询满足要求的数据的下标索引

```
In [9]: ind = np.argwhere(x>0.5)
        ind
```

```
Out[9]: array([[ 0],
               [ 3],
               [ 6],
               [ 7],
               [ 9],
               [10],
               [15],
               [16],
               [18],
               [21],
               [22],
               [28],
               [30],
               [31],
               [35],
               [36],
               [38],
               [39],
               [40],
               [41],
               [42],
               [45],
               [46],
               [49]])
```

```
In [10]: x[ind]
```

```
Out[10]: array([[0.54340494],
                [0.84477613],
                [0.67074908],
                [0.82585276],
                [0.57509333],
                [0.89132195],
                [0.97862378],
                [0.81168315],
                [0.81622475],
                [0.94002982],
                [0.81764938],
                [0.79566251],
                [0.59884338],
                [0.60380454],
                [0.89041156],
                [0.98092086],
                [0.89054594],
                [0.5769015 ],
                [0.74247969],
                [0.63018394],
                [0.58184219],
                [0.54468488],
                [0.76911517],
                [0.85239509]])
```

```
In [11]: x = np.arange(10)
```

```
In [12]: np.random.shuffle(x)        # random.shuffle可以将一个数组打乱顺序
         x
```

```
Out[12]: array([5, 6, 0, 1, 9, 8, 2, 7, 4, 3])
```

```
In [13]: np.sort(x)
```

```
Out[13]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [14]: x
```

```
Out[14]: array([5, 6, 0, 1, 9, 8, 2, 7, 4, 3])
```

numpy.sort()排序后不影响原来的数组

```
In [15]: x.sort()      # 这样进行排序后原来的数组就排序了
```

```
In [16]: x
```

```
Out[16]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [17]: np.random.shuffle(x)
In [18]: x

Out[18]: array([3, 1, 9, 0, 8, 2, 6, 7, 5, 4])

In [19]: ind = np.argsort(x)
         ind

Out[19]: array([3, 1, 5, 0, 9, 8, 6, 7, 4, 2])

In [20]: x[3]

Out[20]: 0

In [21]: ind[:3]

Out[21]: array([3, 1, 5])

In [22]: x[ind[:3]]

Out[22]: array([0, 1, 2])

In [23]: x[ind[-3:]]

Out[23]: array([7, 8, 9])

In [24]: np.partition(x, 4)    # 数组x以4为标准，比之小的放左边，比之大的放右边

Out[24]: array([1, 3, 2, 0, 4, 9, 6, 7, 5, 8])

In [25]: x

Out[25]: array([3, 1, 9, 0, 8, 2, 6, 7, 5, 4])

In [26]: np.argpartition(x, 4)

Out[26]: array([1, 0, 5, 3, 9, 2, 6, 7, 8, 4])

In [27]: X = np.random.randint(20, size=(4, 5))
         X

Out[27]: array([[12,  1,  6, 10,  0],
                [ 2, 19,  4, 18,  4],
                [ 3,  9, 16, 16,  6],
                [ 5,  6,  7, 11, 19]])

In [28]: np.sort(X)           # 直接排序是对每一行排序

Out[28]: array([[ 0,  1,  6, 10, 12],
                [ 2,  4,  4, 18, 19],
                [ 3,  6,  9, 16, 16],
                [ 5,  6,  7, 11, 19]])
```

```
In [29]: np.sort(X, axis=0)        # 对每一列排序

Out[29]: array([[ 2,  1,  4, 10,  0],
                [ 3,  6,  6, 11,  4],
                [ 5,  9,  7, 16,  6],
                [12, 19, 16, 18, 19]])
```

```
In [30]: np.argsort(X, axis=0)

Out[30]: array([[1, 0, 1, 0, 0],
                [2, 3, 0, 3, 1],
                [3, 2, 3, 2, 2],
                [0, 1, 2, 1, 3]])
```