

通用函数

一元函数：

函数	描述
np.abs	绝对值
np.sqrt	开根
np.square	平方
np.exp	计算指数(e^x)
np.log, np.log10, np.log2, np.log1p	求以e为底，以10为底，以2为底，以(1+x)为底的对数
np.sign	将数组中的值标签化，大于0的变成1，等于0的变成0，小于0的变成-1
np.ceil	朝着无穷大的方向取整，比如5.1会变成6，-6.3会变成-6
np.floor	朝着负无穷大方向取整，比如5.1会变成5，-6.3会变成-7
np rint, np.round	返回四舍五入后的值
np.modf	将整数和小数分隔开来形成两个数组
np.isnan	判断是否是nan
np.isinf	判断是否是inf
np.cos, np.cosh, np.sin, np.sinh, np.tan, np.tanh	三角函数
np.arccos, np.arcsin, np.arctan	反三角函数

二元函数：

函数	描述
np.add	加法运算（即 $1+1=2$ ），相当于+
np.subtract	减法运算（即 $3-2=1$ ），相当于-
np.negative	负数运算（即-2），相当于加个负号
np.multiply	乘法运算（即 $2*3=6$ ），相当于*
np.divide	除法运算（即 $3/2=1.5$ ），相当于/
np.floor_divide	取整运算，相当于//
np.mod	取余运算，相当于%
greater,greater_equal,less,less_equal,equal,not_equal	>,>=,<,<=,!=的函数表达式
logical_and	&的函数表达式
logical_or	的函数表达式

聚合函数：

函数名称	NAN安全版本	描述
np.sum	np.nansum	计算元素的和
np.prod	np.nanprod	计算元素的积
np.mean	np.nanmean	计算元素的平均值
np.std	np.nanstd	计算元素的标准差
np.var	np.nanvar	计算元素的方差
np.min	np.nanmin	计算元素的最小值
np.max	np.nanmax	计算元素的最大值
np.argmin	np.nanargmin	找出最小值的索引
np.argmax	np.nanargmax	找出最大值的索引
np.median	np.nanmedian	计算元素的中位数

使用 `np.sum` 或者是 `a.sum` 即可实现。并且在使用的時候，可以指定具体哪个轴。同样 `Python` 中也内置了 `sum` 函数，但是 `Python` 内置的 `sum` 函数执行效率没有 `np.sum` 那么高，可以通过以下代码测试了解到：

```
a = np.random.random(1000000)
%timeit sum(a) #使用Python内置的sum函数求总和，看下所花费的时间
%timeit np.sum(a) #使用Numpy的sum函数求和，看下所花费的时间
```

布尔数组的函数：

函数名称	描述
np.any	验证任何一个元素是否为真
np.all	验证所有元素是否为真

比如想看下数组中是不是所有元素都为0，那么可以通过以下代码来实现：

```
np.all(a==0)
# 或者是
(a==0).all()
```

比如我们想要看数组中是否有等于0的数，那么可以通过以下代码来实现：

```
np.any(a==0)
# 或者是
(a==0).any()
```

排序：

1. `np.sort`：指定轴进行排序。默认是使用数组的最后一个轴进行排序。

```
a = np.random.randint(0,10,size=(3,5))
b = np.sort(a) #按照行进行排序，因为最后一个轴是1，那么就是将最里
面的元素进行排序。
c = np.sort(a,axis=0) #按照列进行排序，因为指定了axis=0
```

还有 `ndarray.sort()`，这个方法会直接影响到原来的数组，而不是返回一个新的排序后的数组。

2. `np.argsort`：返回排序后的下标值。示例代码如下：

```
np.argsort(a) #默认也是使用最后的一个轴来进行排序。
```

3. 降序排序：`np.sort` 默认会采用升序排序。如果我们想采用降序排序。那么可以采用以下方案来实现：

```
# 使用负号  
-np.sort(-a)
```

更多：

<https://docs.scipy.org/doc/numpy/reference/index.html>