

Numpy: 提供了一个在Python中做科学计算的基础库, 重在数值计算, 主要用于多维数组(矩阵)处理的库。用来存储和处理大型矩阵, 比Python自身的嵌套列表结构要高效的多。本身是由C语言开发, 是个很基础的扩展, Python其余的科学计算扩展大部分都是以它为基础。

ndarray 多维数组

NumPy数组是一个多维的数组对象(矩阵), 称为 `ndarray`, 具有矢量算术运算能力和复杂的广播能力, 并具有执行速度快和节省空间的特点。

注意: `ndarray`的下标从0开始, 且数组里的所有元素必须是相同类型

性能对比

```
import numpy as np
def pySum():
    a = list(range(10000))
    b = list(range(10000))
    c = []
    for i in range(len(a)):
        c.append(a[i]**2 + b[i]**2)
    return c
%timeit pySum()
#-----
import numpy as np
def npSum():
    a = np.arange(10000)
    b = np.arange(10000)
    c = a**2 + b**2
    return c
%timeit npSum()
```

*****RESULT***

numpy的向量化运算的效率要远远高于python的循环遍历运算(效率相差好几百倍)

*****END***

创建数组

Method-1: 基于list和tuple

```

# 一维数组
# 基于list
arr1 = np.array([1,2,3,4])
print(arr1)
# 基于tuple
arr_tuple = np.array((1,2,3,4))
print(arr_tuple)
# 二维数组 (2*3)
arr2 = np.array([[1,2,4], [3,4,5]])
print(arr2)

```

```

[1 2 3 4]
[1 2 3 4]
[[1 2 4]
 [3 4 5]]

```

Method-2:基于np.arange

```

# 一维数组
arr1 = np.arange(5)
print(arr1)
# 二维数组
arr2 = np.array([np.arange(3), np.arange(3)])
print(arr2,arr2.ndim)

```

```

[0 1 2 3 4]
[[0 1 2]
 [0 1 2]] 2

```

Method 3: 基于arange以及reshape创建多维数组

```
# 创建三维数组
arr = np.arange(24).reshape(2,3,4)
print(arr)
print('数组维度: {}'.format(arr.ndim))
print('大小: {}'.format(arr.size))
print('尺寸: {}'.format(arr.shape))
```

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

```
 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
```

数组维度: 3

大小: 24

尺寸: (2, 3, 4)

Numpy数组各维度轴的方向示意图



Numpy的数值类型

数据类型	说明
bool	布尔类型，True或False，占用1比特
inti	其长度取决于平台的整数，一般是int32或int64
int8	字节长度的整数，取值：[-128, 127]
int16	16位长度的整数，取值：[-32768, 32767]
int32	32位长度的整数，取值： $[-2^{31}, 2^{31} - 1]$
int64	64位长度的整数，取值： $[-2^{63}, 2^{63} - 1]$
uint8	8位无符号整数，取值：[0, 255]
uint16	16位无符号整数，取值：[0, 65535]
uint32	32位无符号整数，取值： $[0, 2^{32} - 1]$
uint64	64位无符号整数，取值： $[0, 2^{64} - 1]$
float16	16位半精度浮点数：1位符号位，5位指数，10位尾数
float32	32位半精度浮点数：1位符号位，8位指数，23位尾数
float64或float	双精度浮点数：1位符号位，11位指数，52位尾数
complex64	复数类型，实部和虚部都是32位浮点数
complex128或complex	复数类型，实部和虚部都是64位浮点数

数组形状

reshape() 和 resize()

函数resize()的作用跟reshape()类似，但是resize()会改变所作用的数组，相当于有inplace=True的效果

```
print(arr.reshape(2,12))
print(arr)
arr.resize(2,12)
print(arr)
```

数组的类型转换

```
#ndarray-->list
aList = arr.tolist()
#list--> ndarray
arr = np.asarray(aList)
print(arr,type(arr))
```

numpy常用统计函数

请注意函数在使用时需要指定axis轴的方向，若不指定，默认统计整个数组。

- np.sum(), 返回求和
- np.mean(), 返回均值
- np.max(), 返回最大值
- np.min(), 返回最小值
- np.std(), 返回标准偏差 (standard deviation) : (平均值u)
- 在描述一个波动范围时标准差比方差更方便

- $$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

- np.var(), 返回方差 (variance) : 用来度量随机变量(x1...)和其数学期望 (即均值x) 之间的偏离程度
- 标注差和方差都是，反映的是一维数组的离散程度
- $$s^2 = \frac{1}{n} [(x_1 - x)^2 + (x_2 - x)^2 + \dots + (x_n - x)^2]$$
- np.cumsum(), 返回累加值
- np.cumprod(), 返回累乘积值