

numpy——数组支持

1.简介

Numpy提供了一个在Python中做科学计算的基础库，重在数值计算，主要用于处理多维数组（矩阵）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此为基础。

- 高性能科学计算和数据分析的基础包
- ndarray，多维数组（矩阵），具有矢量运算能力，快速、节省空间
- 矩阵运算，无需循环，可完成类似Matlab中的矢量运算
- 线性代数、随机数生成

2.安装与使用

- 已有发行版：anaconda/winpython/Pyzo
- 使用pip安装：pip install numpy

```
import numpy as np
np.arange(n[,dtype=int/float/bool])
np.array([[],[]])
```

3.基本操作

3.1 创建数组

- 由列表创建
`a=np.array([[1,3,2,5],[1,2,2,7]])`
- arrange
`np.arange(5,dtype=int)`

3.2 常见操作

- 查看维度：np.shape(arr)
- 修改维度：arr.reshape((row,col)) 本身不修改，返回修改的数组。如果为一维，需要是(n,)。(n,1)和(n,)是不一样的。亦可np.reshape(c,(row,col))
- 查看类型：
`type(arr)` numpy.ndarray 数组类型
`arr.dtype` int32 元素类型
- 修改数据类型：arr.astype("") 科学计数法转整数为int
- 拉平操作：arr.flatten() 展开成一维
- 广播运算：加/减/乘/除上一个常数，为数组中所有元素加/减/乘/除

广播原则：若两数组的后缘维度的轴**长度相符**或一方**长度为1**，则认为是广播兼容，广播在缺失和长度为1的维度上进行

- 同维运算
- 保留小数位数：np.round(arr,n)
- 转置：arr.transpose()
- 交换轴：swapaxes(1,0)

注：由数组调用的方法，不改变原数组，而是返回修改后的数组

3.3 读取数据

- CSV文件 (Comma-Separated Value)
- np.loadtxt(frame,dtype=np.float/str/int,delimiter=" ",skiprows=0,usecols=None,unpack=False)
- frame文件、字符串或产生器，可以是压缩文件
- dtype加载数据类型
- delimiter分隔符，默认空格
- skiprows跳过前x行
- usecols读取指定列，索引、元组类型
- unpack转置，True时分别写入不同数组变量，False只写入一个变量

3.4 索引与切片

- 取某一行：arr[n]
- 取连续多行：arr[n:]
- 取不连续多行：arr[[a,b,c,d,...]]
- 取行列：arr[a,b] 其中a,b可以用连续和离散方式表示，离散时前后长度一致
- 布尔索引
 - arr<n：返回数组中所有位置是否满足该条件，返回满足条件的索引
 - 三元运算符：np.where(condition,val1,val2) condition是arr相关条件
 - 裁剪：arr.clip(a,b)小于a的替换成a，大于b的替换成b
 - **注：三元运算符和裁剪均不修改原数组**
 - np.nan是float类型

3.5 数据操作

- 数据拼接
 - 竖直拼接
 - np.vstack((,))
 - 水平拼接
 - np.hstack((,))
- 数据生成
 - 全0：np.zeros((row,col))
 - 全1：np.ones((row,col))
 - 单位阵：np.eye(n)
- 获取最值及位置
 - np.max(arr,axis=0)
 - np.min(arr,axis=0)
 - np.argmax(arr,axis=0)

- np.argmax(arr,axis=1)
- np.sum(arr,axis=0) arr.sum(axis=0)
- 随机数生成 np.random.
 - rand(d0,d1,...)多维均匀分布
 - randn(d0,d1,...)标准正态分布
 - randint(low,high,(shape))整数
 - uniform(low,high,(shape))均匀分布
 - normal(mu,sigma,(size))正态分布
 - seed(s)随机数种子 设定之后，使得产生相同随机数
- nan和inf
 - 缺失值，不合理运算
 - np.nan和np.inf均为float类型
 - np.nan!=np.nan
 - 判断nan: np.isnan(a)或a!=a
 - 判断数组中nan的个数
 - np.count_nonzero(a!=a)
 - np.count_nonzero(np.isnan(a))
 - 缺失值处理
 - 删除有缺失值的一行
 - 替换为列均值（中值）

3.6 统计函数

- sum
- mean
- np.median(arr,axis=0)
- max
- min
- np.ptp极值差
- std
- 默认返回多维数组的全部统计结果，若指定指定坐标轴axis，则返回该轴上结果

本文由Regent Wan原创，关注公众号可以勾搭哟！更多文章请关注公众号：学术创客。



微信搜一搜

学术创客