

Numpy.array()详解、np.array与np.asarray辨析、np.array和np.ndarray的区别

原创 THEAQING 于 2019-04-06 15:25:45 发布 60524 收藏 267 版权
分类专栏: Python



Python 专栏收录该内容

39 订阅 15 篇文章

订阅专栏

记录一下 `numpy .array()` 的详细用法，以及与 `np.asarray()` 和 `np.ndarray()` 的区别。

目录

1. Numpy.array()详解

1.1 函数形式

1.2 参数详解

1.3 具体用法

2. Asarray和Array辨析

2.1 object对象是普通迭代序列时

2.2 object对象是ndarray对象时

3.Numpy.ndarray()

1. Numpy.array()详解

该函数的作用一言蔽之就是用来产生数组。

1.1 函数形式

```
1 numpy.array(object,  
2 dtype=None,  
3 copy=True,  
4 order='K',  
5 subok=False,  
6 ndmin=0)
```

1.2 参数详解

object: 必选参数，类型为 `array_like`，可以有四种类型：数组，公开数组接口的任何对象，`__array__` 方法返回数组的对象，或任何（嵌套）序列。`np.array()` 的作用就是按照一定要求将 `object` 转换为数组。

dtype: 可选参数，用来表示数组元素的类型。如果没有给出，那么类型将被确定为保持序列中的对象所需的最小类型。注：This argument can only be used to 'upcast' the array. For downcasting, use the `.astype(t)` method.

copy: 可选参数，类型为 `bool` 值。如果为 `true`（默认值），则复制对象。否则的话只有在以下三种情况下才会返回副本：（1）`__array__` returns a copy；（2）`obj` is a nested sequence；（3）`if a copy is needed to satisfy any of the other requirements (dtype, order, etc.)`

order: {'K', 'A', 'C', 'F'}, optional。指定阵列的内存布局。该参数我至今还没有遇到过具体用法，这句话的意思就



THEAQING

关注

65

subok: 可选参数，类型为bool值。如果为True，则子类将被传递，否则返回的数组将被强制为基类数组（默认）。或者说，True：使用object的内部数据类型，False：使用object数组的数据类型。

ndmin: 可选参数，类型为int型。指定结果数组应具有的最小维数。

返回对象

out: 输出ndarray，满足指定要求的数组对象。

1.3 具体用法

简单示例

```
1 import numpy as np
2
3 arr01 = np.array([1,2,3])
4 print(arr01) #[1 2 3]
5 print(type(arr01)) #<class 'numpy.ndarray'>
6 print(arr01.dtype) #int32
7
8 #Upcasting
9 arr02 = np.array([1.,2.,3.])
10 print(arr02) #[1. 2. 3.]
11 print(arr02.dtype) #float64
12
13 #More than one dimension:
14 arr03 = np.array([[1,2],[3,4]])
15 print(arr03)
16 """
17 [[1 2]
18  [3 4]]
19 """
20
```

dtype参数使用示例

```
1 import numpy as np
2
3 #指定数组元素类型为复数类型
4 DYX= np.array([1,2,3],dtype = complex)
5 print(DYX) #[1.+0.j 2.+0.j 3.+0.j]
6 print(DYX.dtype) #complex128
7
8 #由多个元素组成的数据类型:
9 HXH = np.array([(1,2),(3,4)],dtype=[('a','<i4'),('b','<i8')])
10 print(HXH) #[(1, 2) (3, 4)]
11 #下面的输出有点神奇，我也只能记住规律了。
12 print(HXH["a"]) #[1 3]
13 print(HXH["b"]) #[2 4]
14 print(HXH.dtype) #[('a', '<i4'), ('b', '<i8')]
15 print(HXH["a"].dtype) #int32
16 print(HXH["b"].dtype) #int64
17
18
19 TSL = np.array([(1,2,3),(4,5,6)],dtype=[("a","i"),("b","i"),("c","i")])
20 print(TSL["a"]) #[1 4]20 | print(TSL["a"].dtype) #int32
```

上述代码中，numpy的数据类型，可以参看：<http://www.runoob.com/numpy/numpy-dtype.html>

subok参数使用示例

```
1 import numpy as np
2
3 DYX = np.array(np.mat('1 2; 3
4 #没有显示的写出subok的值，但是默
```



THEAQING

关注

65



```

5 | print(DYX) 6 | #数组类型
7 | print(type(DYX)) #<class 'numpy.ndarray'>
8 | """
9 | [[1 2]
10 |  [3 4]]
11 | """
12 |
13 | HXH = np.array(np.mat('1 2; 3 4'), subok=True)
14 | print(HXH)
15 | #矩阵类型
16 | print(type(HXH)) #<class 'numpy.matrixlib.defmatrix.matrix'>
17 | """
18 | [[1 2]
19 |  [3 4]]
20 | """

```

前文对subok的描述是这样的：“如果为True，则子类将被传递，否则返回的数组将被强制为基类数组（默认）”。

在上文的代码中“np.mat('1 2; 3 4')”，就是子类，是矩阵类型。DYX = np.array(np.mat('1 2; 3 4'))中subok为False，返回的数组类型被强制为基类数组，所以DYX的类型是<class 'numpy.ndarray'>，是数组；HXH = np.array(np.mat('1 2; 3 4'), subok=True)中subok为True，子类被传递，所以HXH的类型是矩阵<class 'numpy.matrixlib.defmatrix.matrix'>。

这就是区别所在。

ndmin参数使用示例

```

1 | import numpy as np
2 |
3 | DYX = np.array([1,2,3],ndmin=0)
4 | print(DYX,DYX.shape) #[1 2 3] (3,)
5 |
6 | HXH = np.array([1,2,3],ndmin=1)
7 | print(HXH,HXH.shape) #[1 2 3] (3,)
8 |
9 | TSL = np.array([1,2,3],ndmin=2)
10 | print(TSL,TSL.shape) #[[1 2 3]] (1, 3)

```

其他两个参数copy和order，我至今还没有遇到过，所以暂且不表。谁有介绍这两个参数用法的博客吗？

2. Asarray和Array辨析

Numpy.asarray的用法不再赘述，主要介绍一下二者的区别。

2.1 object对象是普通迭代序列时

```

1 | import numpy as np
2 |
3 | data = [1,1,1]
4 | print(type(data)) #<class 'list'> 列表类型
5 | arr_ar = np.array(data)
6 | arr_as = np.asarray(data)
7 |
8 | #输出上没有区别
9 | print(arr_ar) #[1 1 1]
10 | print(arr_as) #[1 1 1]
11 |
12 | data[1]=2
13 | #改变原序列对arr_ar和arr_as没影响
14 | print(arr_ar) #[1 1 1]
15 | print(arr_as) #[1 1 1]
16 |
17 | #此时data是[1, 2, 1]

```



THEAQING

关注

65

```

18 #改变arr_ar和arr_as对原序列没有影响19 arr_ar[1]=3
20 print(data) #[1, 2, 1]
21 arr_as[1]=3
22 print(data) #[1, 2, 1]

```

可见在参数对象是普通迭代序列时，asarray和array没有区别（在我的理解范围内）。

2.2 object对象是ndarray对象时

```

1 import numpy as np
2
3 data = np.ones((3,))
4 #print(type(data)) #<class 'numpy.ndarray'> 数组类型
5 arr_ar = np.array(data)
6 arr_as = np.asarray(data)
7
8 print(arr_ar) #[1. 1. 1.]
9 print(arr_as) #[1. 1. 1.]
10
11 """
12 这边区别就出来了。修改原始序列后，
13 np.array()产生的数组不变，
14 但是np.asarray()产生的数组发生了变化
15 """
16 data[1]=2
17 print(arr_ar) #[1. 1. 1.]
18 print(arr_as) #[1. 2. 1.] !!!
19
20
21 """
22 这边也有区别，修改array产生的数组，不影响原始序列
23 修改asarray产生的数组，会影响原始序列
24 """
25 #此时data=[1. 2. 1.]
26 arr_ar[2]=3
27 print(data) #[1. 2. 1.]
28 arr_as[2]=3
29 print(data) #[1. 2. 3.]

```

我们总结一下：**相同点**：array和asarray都可以将数组转化为ndarray对象。

区别：当参数为一般数组时，两个函数结果相同；当参数本身就是ndarray类型时，array会新建一个ndarray对象，作为参数的副本，但是asarray不会新建，而是与参数共享同一个内存。重点就是这个共享内存。

3.Numpy.ndarray()

这是最近在一个项目里看到的用法，搜索了一下用法，只在stackoverflow看到了一个问题：“**What is the difference between ndarray and array in numpy?**”。

地址如下：<https://stackoverflow.com/questions/15879315/what-is-the-difference-between-ndarray-and-array-in-numpy>

numpy.array只是一个创建ndarray的便利函数；它本身不是一个类。他讲到也可以使用numpy.ndarray创建一个数组，但这不是推荐的方法。numpy.ndarray()是一个类，而numpy.array()是一个创建ndarray的方法/函数。

在numpy docs中，如果你想从ndarray类创建一个数组，你可以用引用的2种方式来做：

(1) .using array(), zeros() or empty() methods: Arrays should be constructed using array, zeros or empty. parameters given here refer to a low-level



THEAQING

关注

65



array。【1-使用array(), zeros()或empty()方法：数组应该使用array, zeros()或empty()构造。这里给出的参数引用用于实例化数组的低级方法(ndarray(...))。】


(2) .from ndarray class directly: There are two modes of creating an array using new: If buffer is None, then only shape, dtype, and order are used. If buffer is an object exposing the buffer interface, then all keywords are interpreted. 【2-来自ndarray类：使用new创建数组有两种模式：如果buffer是None，则只使用shape，dtype和order。如果buffer是公开buffer接口的对象，则解释所有关键字。】

所以说老老实实用numpy.array()吧。

本篇完结，行笔匆忙，如有错误，还望指出。

文章知识点与官方知识档案匹配，可进一步学习相关知识

Python入门技能树 科学计算基础软件包NumPy 安装配置 32313 人正在系统学习中



与50位技术专家面对面

20年技术见证，附赠技术全景图

>

相关推荐

- 

np.array的几点特殊用法

哈巴泰拉的技术博客 3029

1、[...]和[:,:]的区别：L=np.array([[1,2],[2,3],[3,4]],[[4,5],[6,7],[7,8]]) # print(L[0]) # print(L[0][...])
- 

numpy中np.array()功能

weixin_38410551的博客 3万+

功能：将数据转化为矩阵 a=[[1,2,3],[4,5,6],[7,8,9]] b=np.array(a) c=np.asarray(a) a[2]=1 prin...
- 

python中np array_Python数据分析之numpy数组全解析_we...

4-8

上面我们将range()函数结果传递给np.array(),np.arange()实现的就是这个功能,所以说,np.ara...
- 

关于np.array(np里的数组)_CHEN_BR的博客_np数组

3-21

c=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])c.ndim##2,维度信息c.shape##(4, 3)c.size##12c...
- 

np.array参数

qq_35608277的博客 2883

numpy.ones(shape, dtype = None, order = 'C') 1. Shape 空数组的形状，整数或整数元组 2. ...
- 

np.array()

Candyerer的博客 618

1.numpy构造函数 np.array([1,2,3,4,5]) ///一维数组 np.array([1,2,3],[2,3,4])///二维数组 1.1 n...
- 

python np.array()函数原理详细分析——列表和数组

帅kakaci的博客 3095

一、数组和列表 列表不存在维度问题，数组是有维度的 np.array()把列表转化为数组 问题：...
- 

python中数组（numpy.array）的基本操作 热门推荐

fu6543210的博客 51万+

为什么要用numpy Python中提供了list容器，可以当作数组使用。但列表中的元素可以是...
- 

Numpy np.array 相关常用操作学习笔记

weixin_34185364的博客 5507

1.numpy构造函数 用法： np.array([1,2,3,4,5]) 1.1 numpy array 和 python list 有什么区别？ ...
- 

np.array 数组操作

Jq的博客 403

NumPy 数组有助于对大量数据进行高级数学和其他类型的操作。通常，这些操作的执行效率...
- 

np.array()用法

chunlinqin的博客 1500
- 

Numpy中数组的创建：np.array()方法详细介绍

Henryw1107的博客 2134

之前，我们有提到如何下载Numpy并且添加到Pycharm的解释器中，这次，我们来看看numpy...
- 

Python numpy np.array与np.asarray之间的区别

winner3的专栏 1391

np.array与np.asarray 都可以将list数据转化...
- 

np.array()与np.asarray()



THEAQING

关注

65