

(/apps/redir
utm_source
banner-click

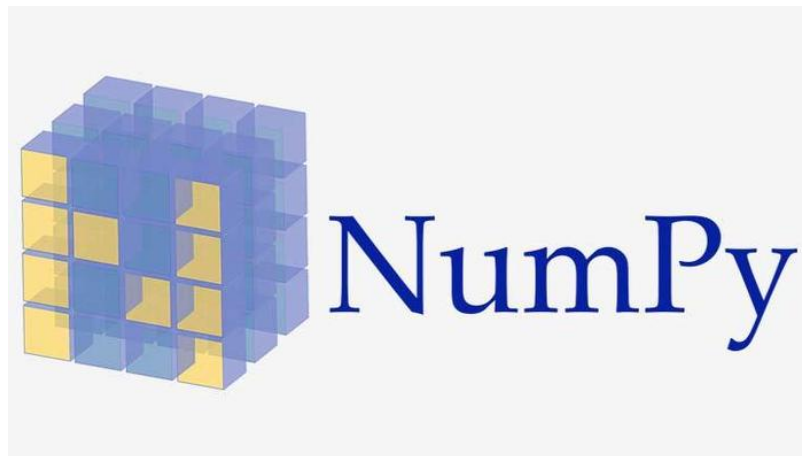
机器学习三剑客之Numpy



木子昭 (/u/c5d047065c42) +关注

2017.12.31 19:48* 字数 558 阅读 30533 评论 2 喜欢 49 赞赏 1

(/u/c5d047065c42)



NumPy

NumPy是Python语言的一个扩充程序库。支持高级大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。Numpy内部解除了Python的GIL(全局解释器锁),运算效率极好,是大量机器学习框架的基础库!

NumPy简单创建数组

```
import numpy as np
# 创建简单的列表
a = [1, 2, 3, 4]
# 将列表转换为数组
b = np.array(a)
```

NumPy查看数组属性

数组元素个数

b.size

数组形状

b.shape



数组维度

```
b.ndim
```

数组元素类型

```
b.dtype
```

(/apps/redir
utm_source
banner-clic

快速创建N维数组的api函数

- 创建10行10列的数值为浮点1的矩阵

```
array_one = np.ones([10, 10])
```

- 创建10行10列的数值为浮点0的矩阵

```
array_zero = np.zeros([10, 10])
```

- 从现有的数据创建数组
 - array(深拷贝)
 - asarray(浅拷贝)

Numpy创建随机数组 `np.random`

• 均匀分布

- `np.random.rand(10, 10)` 创建指定形状(示例为10行10列)的数组(范围在0至1之间)
- `np.random.uniform(0, 100)` 创建指定范围内的一个数
- `np.random.randint(0, 100)` 创建指定范围内的一个整数

• 正态分布

给定均值/标准差/维度的正态分布 `np.random.normal(1.75, 0.1, (2, 3))`

- 数组的索引, 切片

```
# 正态生成4行5列的二维数组  
arr = np.random.normal(1.75, 0.1, (4, 5))  
print(arr)  
# 截取第1至2行的第2至3列(从第0行算起)  
after_arr = arr[1:3, 2:4]  
print(after_arr)
```



```

In [53]: arr = np.random.normal(1.75, 0.1, (4, 5))

In [54]: print(arr)

[[ 1.62273269  1.89722964  1.61242972  1.59515991  1.49664786]
 [ 1.63876594  1.6904798  1.75099384  1.80397133  1.75823123]
 [ 1.76833811  1.67125745  1.75108021  1.83840158  1.7902311 ]
 [ 1.77314175  1.6289854  1.7364316  1.70858196  1.97275401]]

In [55]: after_arr = arr[1:3, 2:4]

In [56]: print(after_arr)

[[ 1.75099384  1.80397133]
 [ 1.75108021  1.83840158]]

```

数组索引

(/apps/redir
utm_source
banner-clc

- 改变数组形状(要求前后元素个数匹配)

```

In [60]: print("reshape函数的使用!")
one_20 = np.ones([20])
print("-->1行20列<--")
print (one_20)

one_4_5 = one_20.reshape([4, 5])
print("-->4行5列<--")
print (one_4_5)

reshape函数的使用!
-->1行20列<--
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]
 1.  1.]
-->4行5列<--
[[ 1.  1.  1.  1.  1.]
 [ 1.  1.  1.  1.  1.]
 [ 1.  1.  1.  1.  1.]
 [ 1.  1.  1.  1.  1.]]

```

改变数组形状

```

print("reshape函数的使用!")
one_20 = np.ones([20])
print("-->1行20列<--")
print (one_20)

one_4_5 = one_20.reshape([4, 5])
print("-->4行5列<--")
print (one_4_5)

```



Numpy计算(重要)

条件运算

(/apps/redir
utm_source
banner-clic

学生姓名	体育平时成绩	体育期末成绩
老王	80	88
大毛	82	81
三毛	84	75
宝儿	86	83
徐四	75	81

原始数据

```
In [63]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])  
stus_score > 80  
Out[63]: array([[False,  True],  
               [ True,  True],  
               [ True, False],  
               [ True,  True],  
               [False,  True]], dtype=bool)
```

条件判断

条件判断

```
import numpy as np  
stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])  
stus_score > 80
```

```
In [64]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])  
np.where(stus_score < 80, 0, 90)  
Out[64]: array([[90, 90],  
               [90, 90],  
               [90,  0],  
               [90, 90],  
               [ 0, 90]])
```

三目运算

如果数值小于80,替换为0;如果大于80,替换为90

三目运算

```
import numpy as np  
stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])  
np.where(stus_score < 80, 0, 90)
```

统计运算

- 指定轴最大值 `amax` (参数1: 数组; 参数2: `axis=0/1`; 0表示列1表示行)



```
In [67]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一列的最大值(0表示列)
print("每一列的最大值为:")
result = np.amax(stus_score, axis=0)
print(result)

# 求每一行的最大值(1表示行)
print("每一行的最大值为:")
result = np.amax(stus_score, axis=1)
print(result)

每一列的最大值为:
[86 88]
每一行的最大值为:
[88 82 84 86 81]
```

求最大值

求最大值

(/apps/redir
utm_source
banner-clic

```
stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一列的最大值(0表示列)
print("每一列的最大值为:")
result = np.amax(stus_score, axis=0)
print(result)

# 求每一行的最大值(1表示行)
print("每一行的最大值为:")
result = np.amax(stus_score, axis=1)
print(result)
```

• 指定轴最小值 amin

```
In [68]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一列的最小值(0表示列)
print("每一列的最小值为:")
result = np.amin(stus_score, axis=0)
print(result)

# 求每一行的最小值(1表示行)
print("每一行的最小值为:")
result = np.amin(stus_score, axis=1)
print(result)

每一列的最小值为:
[75 75]
每一行的最小值为:
[80 81 75 83 75]
```

求最小值

求最小值

```
stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一列的最小值(0表示列)
print("每一列的最小值为:")
result = np.amin(stus_score, axis=0)
print(result)

# 求每一行的最小值(1表示行)
print("每一行的最小值为:")
result = np.amin(stus_score, axis=1)
print(result)
```

• 指定轴平均值 mean

```
In [70]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一列的平均值(0表示列)
print("每一列的平均值为:")
result = np.mean(stus_score, axis=0)
print(result)

# 求每一行的平均值(1表示行)
print("每一行的平均值为:")
result = np.mean(stus_score, axis=1)
print(result)

每一列的平均值:
[ 81.4  81.6]
每一行的平均值:
[ 84.   81.5  79.5  84.5  78. ]
```

求平均值

求平均值



```
stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一行的平均值(0表示列)
print("每一列的平均值:")
result = np.mean(stus_score, axis=0)
print(result)

# 求每一行的平均值(1表示行)
print("每一行的平均值:")
result = np.mean(stus_score, axis=1)
print(result)
```

(/apps/redir
utm_source
banner-clic

• 方差 std

```
In [71]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一列的方差(0表示列)
print("每一列的方差:")
result = np.std(stus_score, axis=0)
print(result)

# 求每一行的方差(1表示行)
print("每一行的方差:")
result = np.std(stus_score, axis=1)
print(result)

每一列的方差:
[ 3.77359245  4.1761226 ]
每一行的方差:
[ 4.   0.5  4.5  1.5  3. ]
```

求方差

求方差

```
stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
# 求每一列的方差(0表示列)
print("每一列的方差:")
result = np.std(stus_score, axis=0)
print(result)

# 求每一行的方差(1表示行)
print("每一行的方差:")
result = np.std(stus_score, axis=1)
print(result)
```

数组运算

• 数组与数的运算

```
In [76]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
print("加分前:")
print(stus_score)

# 为所有平时成绩都加5分
stus_score[:, 0] = stus_score[:, 0]+5
print("加分后:")
print(stus_score)

加分前:
[[80 88]
 [82 81]
 [84 75]
 [86 83]
 [75 81]]
加分后:
[[85 88]
 [87 81]
 [89 75]
 [91 83]
 [80 81]]
```

数组与数运算:

加法

加法



```

stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
print("加分前:")
print(stus_score)

# 为所有平时成绩都加5分
stus_score[:, 0] = stus_score[:, 0]+5
print("加分后:")
print(stus_score)

```

(/apps/redir
utm_source
banner-clic

```

In [78]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
print("减半前:")
print(stus_score)

# 平时成绩减半
stus_score[:, 0] = stus_score[:, 0]*0.5
print("减半后:")
print(stus_score)

```

减半前:

```

[[80 88]
 [82 81]
 [84 75]
 [86 83]
 [75 81]]

```

减半后:

```

[[40 88]
 [41 81]
 [42 75]
 [43 83]
 [37 81]]

```

数组与数的运算:
乘法

乘法

```

stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])
print("减半前:")
print(stus_score)

# 平时成绩减半
stus_score[:, 0] = stus_score[:, 0]*0.5
print("减半后:")
print(stus_score)

```

• 数组间也支持加减乘除运算,但基本用不到

```

In [79]: a = np.array([1, 2, 3, 4])
b = np.array([10, 20, 30, 40])
c = a + b
d = a - b
e = a * b
f = a / b
print("a+b为", c)
print("a-b为", d)
print("a*b为", e)
print("a/b为", f)

```

a+b为 [11 22 33 44]
a-b为 [-9 -18 -27 -36]
a*b为 [10 40 90 160]
a/b为 [0.1 0.1 0.1 0.1]

数组间运算

image.png

```

a = np.array([1, 2, 3, 4])
b = np.array([10, 20, 30, 40])
c = a + b
d = a - b
e = a * b
f = a / b
print("a+b为", c)
print("a-b为", d)
print("a*b为", e)
print("a/b为", f)

```



矩阵运算 `np.dot()` (非常重要)

学生姓名	体育平时成绩(40%)	体育期末成绩(60%)	总成绩
老王	80	88	84.8
大毛	82	81	81.4
三毛	84	75	78.6
宝儿	86	83	84.2
徐四	75	81	78.6

根据权重计算成绩

(/apps/redir
utm_source
banner-clic

• 计算规则

$(M\text{行}, N\text{列}) * (N\text{行}, Z\text{列}) = (M\text{行}, Z\text{列})$

```
In [84]: stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])  
# 平时成绩占40% 期末成绩占60%，计算结果  
q = np.array([[0.4], [0.6]])  
result = np.dot(stus_score, q)  
print("最终结果为:")  
print(result)
```

矩阵运算:总成绩

```
最终结果为:  
[[ 84.8]  
 [ 81.4]  
 [ 78.6]  
 [ 84.2]  
 [ 78.6]]
```

矩阵计算总成绩

```
stus_score = np.array([[80, 88], [82, 81], [84, 75], [86, 83], [75, 81]])  
# 平时成绩占40% 期末成绩占60%，计算结果  
q = np.array([[0.4], [0.6]])  
result = np.dot(stus_score, q)  
print("最终结果为:")  
print(result)
```

• 矩阵拼接

◦ 矩阵垂直拼接




```
In [93]: print("v1为:")
v1 = [[0, 1, 2, 3, 4, 5],
      [6, 7, 8, 9, 10, 11]]
print(v1)
print("v2为:")
v2 = [[12, 13, 14, 15, 16, 17],
      [18, 19, 20, 21, 22, 23]]
print(v2)
# 垂直拼接
result = np.vstack((v1, v2))
print("v1和v2垂直拼接的结果为")
print(result)

v1为:
[[0, 1, 2, 3, 4, 5], [6, 7, 8, 9, 10, 11]]
v2为:
[[12, 13, 14, 15, 16, 17], [18, 19, 20, 21, 22, 23]]
v1和v2垂直拼接的结果为
[[ 0  1  2  3  4  5]
 [ 6  7  8  9 10 11]
 [12 13 14 15 16 17]
 [18 19 20 21 22 23]]
```

垂直拼接

垂直拼接

```
print("v1为:")
v1 = [[0, 1, 2, 3, 4, 5],
      [6, 7, 8, 9, 10, 11]]
print(v1)
print("v2为:")
v2 = [[12, 13, 14, 15, 16, 17],
      [18, 19, 20, 21, 22, 23]]
print(v2)
# 垂直拼接
result = np.vstack((v1, v2))
print("v1和v2垂直拼接的结果为")
print(result)
```

• 矩阵水平拼接

```
In [95]: print("v1为:")
v1 = [[0, 1, 2, 3, 4, 5],
      [6, 7, 8, 9, 10, 11]]
print(v1)
print("v2为:")
v2 = [[12, 13, 14, 15, 16, 17],
      [18, 19, 20, 21, 22, 23]]
print(v2)
# 水平拼接
result = np.hstack((v1, v2))
print("v1和v2水平拼接的结果为")
print(result)

v1为:
[[0, 1, 2, 3, 4, 5], [6, 7, 8, 9, 10, 11]]
v2为:
[[12, 13, 14, 15, 16, 17], [18, 19, 20, 21, 22, 23]]
v1和v2水平拼接的结果为
[[ 0  1  2  3  4  5 12 13 14 15 16 17]
 [ 6  7  8  9 10 11 18 19 20 21 22 23]]
```

水平拼接

水平拼接

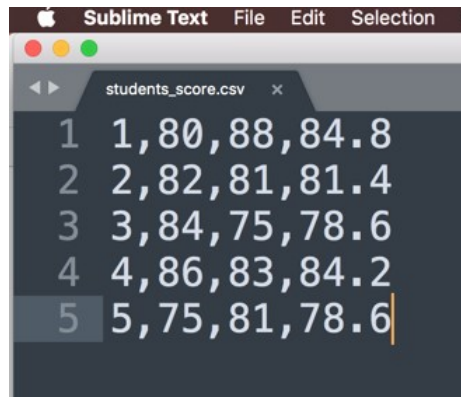
(/apps/redir
utm_source
banner-clic



```
print("v1为:")
v1 = [[0, 1, 2, 3, 4, 5],
       [6, 7, 8, 9, 10, 11]]
print(v1)
print("v2为:")
v2 = [[12, 13, 14, 15, 16, 17],
       [18, 19, 20, 21, 22, 23]]
print(v2)
# 垂直拼接
result = np.hstack((v1, v2))
print("v1和v2水平拼接的结果为")
print(result)
```

(/apps/redir
utm_source
banner-click

Numpy读取数据 np.genfromtxt



csv文件以逗号分隔数据

```
In [102]: # genfromtxt 设置csv文件读取路径, 设置分隔符
result = np.genfromtxt("../students_score.csv", delimiter=",")
print(result)

[[ 1.  80.  88.  84.8]
 [ 2.  82.  81.  81.4]
 [ 3.  84.  75.  78.6]
 [ 4.  86.  83.  84.2]
 [ 5.  75.  81.  78.6]]
```

读取csv格式的文件

如果数值有无法识别的值出现,会以 nan 显示, nan 相当于 np.nan ,为float类型.

< 上一篇 (/p/570c1acdd236)

目录

下一篇 > (/p/7414364992e4)

如果您喜欢这篇文章,请点赞或打赏 ~

赞赏支持





程序员 (/nb/9526199)

4.1万字 · 13.0万阅读 · 308人关注

日常开发分享集

+ 关注

(/apps/redir
utm_source
banner-click

喜欢 | 49



更多分享



下载简书 App ▶

随时随地发现和创作内容



(/apps/redirect?utm_source=note-bottom-click)



登录 (/sign-in?utm_source=desktop&utm_medium=not-signed-in-comm)

2条评论 只看作者 按时间倒序 按时间正序

- 


CK_eb83 (/u/fed795897cb1)
3楼 · 2018.08.12 11:27
(/u/fed795897cb1)
挺实用的!


赞  回复
- 


api1024 (/u/9b76e338f244)
2楼 · 2018.07.31 15:03
(/u/9b76e338f244)
大神，可以交流下py机器学习吗，我Q 35816146
我还建立了大数据自学交流群 829163554 请大神光临


赞  回复

被以下专题收入，发现更多相似内容

- 

程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)
- 

工具癖 (/c/2mvgxp?utm_source=desktop&utm_medium=notes-included-collection)
- 

程序猿阵线联盟... (/c/0bab91ded569?utm_source=desktop&utm_medium=notes-included-collection)
- 

进击的Python (/c/a623c2b10540?)



说明

- 将输入数据（列表、元组、数组或其他序列类型）转换成由 dtype 表示的特别指定 dtype，默认直接复制输入数据
- 将输入转换为 ndarray，如输入本身就是一个 ndarray，类似于内部的 range，但返回的是一个 ndarray 而不是根据指定的 dtype 和 dtype 创建的一个全 1 数组，one_like 数，并根据其形状和 dtype 创建一个全 1 数组
- 类似于 ones 和 ones_like，只不过产生的全是 0 数组
- 创建新数组，只分配内存空间但不填充任何值
- 用 fill value 中的所有值，根据指定的形状和 dtype 创建使用另一个数组，用相同的形状和 dtype 创建
- 创建一个正方的 $N \times N$ 单位矩阵（对角线为 1，其余为 0）

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation

章 数据规整：聚合、合并和重塑第9章...

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

MOBI格式 代码仓库 NumPy - 简介 NumPy 是一个 Python 包。它代表 “Numeric Pyt...

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

Numpy是Python的第三方模块，用于科学计算。

1. 属性 列表转化为数组
2. array的创建 指定数据类型dtype 创建一些特殊数组zeros,ones,empty,linspace,arange,reshape
3. Numpy基础运算 基本运算加减乘除, ...

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

一. NumPy的引入 标准安装的Python中用列表(list)保存一组值，可以用来当作数组使用，不过由于列表的元素可以是任何对象，因此列表中所保存的是对象的指针。这样为了保存一个简单的[1,2,3]，需要有3个指针和三个整数对象。对于数值运算来说这种结构显然比较浪费...

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

主 意

加賀澤蘭斯 4

东野圭吾最畅销系列
加賀澤蘭斯第4部

日本国民作家

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation

故事简介:畅销书作家日高邦彦在出国前一天被杀死家中,加贺参与此次事件调查,并根据现场的烟头以及死者的朋友野野口修的口供等等确定他就是凶手。本以为这样可以结案了,却因为加贺对于野野口的作案动机心存怀疑不断进行调查,最终揭开了野野口犯案的真实动机。野野口修第一次招认:加贺...

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation



[小别离 \(/p/d8786f709fb7?utm_campaign=maleskine&utm_content=not...](#)

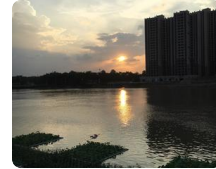
如果用一生换一天你最想成为的人，你会愿意吗？当一个人决定要跟另外一个人分开的时候，他的内心是矛盾的，一边果断地离开，一边拼命想对方挽留。不知道怎么写出了这样一段话。下午在看小别离的时候，一路哭过来，不知是我泪点低还是联想到以后的婚姻生活。我怕我的生活没有动力，怕当我...



简单Apirl (/u/9ec0aac1283f?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation banner-click

(/p/27486979e857?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation

Ways to Live Forever (/p/27486979e857?utm_campaign=maleskine&ut...

Can we? No. So please treasure what you have in present and live your life, really. Or, we do have some ways to live forever. That is to ...



theBigVivi (/u/cbef0cf19327?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

推动粮食烘干机械发展 确保食品安全问题 (/p/35be96c2d8b7?utm_campa...

食品安全是国家大事，那么保障粮食的安全也是我国的头等大事，在发展中也将是经济发展的重中之重。随着我国全力推进粮食生产的机械设备，以及全程机械化和农业生产机械化的进程，粮食烘干机设备在机械行业内得到了快速的发展。但是由于是刚刚起步，整个粮食烘干机设备的市场还处于比较混乱的阶段...



农业烘干工厂店 (/u/1e5f92ed25a3?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

(/p/ec05dd8304cc?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation

普法：员工请事假期间出国旅游，公司可解雇吗？ | 劳动法库 (/p/ec05dd83...

来源 劳动法库 杨小云于2012年5月28日进入宝丰公司从事销售助理工作，最后一份劳动合同期限为2015年7月1日至2018年6月30日。杨小云婆婆袁某某于2015年6月30日至上海市第六人民医院就诊，杨小云于2015年7月3日向宝丰公司提出事假申请，事假日期为当月6日至...



管東平律師 (/u/1fa573603cc5?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

