

NumPy IO

NumPy 可以读写磁盘上的文本数据或二进制数据。

NumPy 为 ndarray 对象引入了一个简单的文件格式：`npz`。

npz 文件用于存储重建 ndarray 所需的数据、图形、dtype 和其他信息。

常用的 IO 函数有：

- `load()` 和 `save()` 函数是读写文件数组数据的两个主要函数，默认情况下，数组是以未压缩的原始二进制格式保存在扩展名为 `.npy` 的文件中。
- `savez()` 函数用于将多个数组写入文件，默认情况下，数组是以未压缩的原始二进制格式保存在扩展名为 `.npz` 的文件中。
- `loadtxt()` 和 `savetxt()` 函数处理正常的文本文件(.txt 等)

numpy.save()

`numpy.save()` 函数将数组保存到以 `.npy` 为扩展名的文件中。

```
numpy.save(file, arr, allow_pickle=True, fix_imports=True)
```

参数说明：

- **file**：要保存的文件，扩展名为 `.npy`，如果文件路径末尾没有扩展名 `.npy`，该扩展名会被自动加上。
- **arr**：要保存的数组
- **allow_pickle**：可选，布尔值，允许使用 Python pickles 保存对象数组，Python 中的 pickle 用于在保存到磁盘文件或从磁盘文件读取之前，对对象进行序列化和反序列化。
- **fix_imports**：可选，为了方便 Python2 中读取 Python3 保存的数据。

实例

```
import numpy as np
a = np.array([1,2,3,4,5])
# 保存到 outfile.npy 文件上
np.save('outfile.npy',a)
# 保存到 outfile2.npy 文件上，如果文件路径末尾没有扩展名 .npy，该扩展名会被自动加上
np.save('outfile2',a)
```

我们可以查看文件内容：

```
$ cat outfile.npy
?NUMPYv{'descr': '<i8', 'fortran_order': False, 'shape': (5,)}, }
$ cat outfile2.npy
?NUMPYv{'descr': '<i8', 'fortran_order': False, 'shape': (5,)}, }
```

可以看出文件是乱码的，因为它们是 Numpy 专用的二进制格式后的数据。

我们可以使用 load() 函数来读取数据就可以正常显示了：

实例

```
import numpy as np
b = np.load('outfile.npy')
print (b)
```

输出结果为：

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

np.savez

numpy.savez() 函数将多个数组保存到以 npz 为扩展名的文件中。

```
numpy.savez(file, *args, **kwds)
```

参数说明：

- **file**：要保存的文件，扩展名为 **.npz**，如果文件路径末尾没有扩展名 **.npz**，该扩展名会被自动加上。
- **args**：要保存的数组，可以使用关键字参数为数组起一个名字，非关键字参数传递的数组会自动起名为 **arr_0**, **arr_1**, ...。
- **kwds**：要保存的数组使用关键字名称。

实例

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
b = np.arange(0, 1.0, 0.1)
c = np.sin(b)
# c 使用了关键字参数 sin_array
np.savez("runoob.npz", a, b, sin_array = c)
r = np.load("runoob.npz")
print(r.files) # 查看各个数组名称
print(r["arr_0"]) # 数组 a
print(r["arr_1"]) # 数组 b
print(r["sin_array"]) # 数组 c
```

输出结果为：

```
['sin_array', 'arr_0', 'arr_1']
[[1 2 3]
 [4 5 6]]
[0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9]
[0.          0.09983342 0.19866933 0.29552021 0.38941834 0.47942554
 0.56464247 0.64421769 0.71735609 0.78332691]
```

savetxt()

savetxt() 函数是以简单的文本文件格式存储数据，对应的使用 loadtxt() 函数来获取数据。

```
np.loadtxt(FILENAME, dtype=int, delimiter=' ')\nnp.savetxt(FILENAME, a, fmt="%d", delimiter=",")
```

参数 delimiter 可以指定各种分隔符、针对特定列的转换器函数、需要跳过的行数等。

实例

```
import numpy as np\na = np.array([1,2,3,4,5])\nnp.savetxt('out.txt',a)\nb = np.loadtxt('out.txt')\nprint(b)
```

输出结果为：

```
[1. 2. 3. 4. 5.]
```

使用 delimiter 参数：

实例

```
import numpy as np\na=np.arange(0,10,0.5).reshape(4,-1)\nnp.savetxt("out.txt",a,fmt="%d",delimiter=",") # 改为保存为整数，以逗号分隔\nb = np.loadtxt("out.txt",delimiter=",") # load 时也要指定为逗号分隔\nprint(b)
```

输出结果为：

```
[[0. 0. 1. 1. 2.]\n [2. 3. 3. 4. 4.]\n [5. 5. 6. 6. 7.]\n [7. 8. 8. 9. 9.]]
```

← NumPy 线性代数

NumPy Matplotlib →

 点我分享笔记

