

E/S com NumPy

Os objetos ndarray podem ser salvos e carregados nos arquivos do disco. As funções IO disponíveis são -

- **As funções load() e save()** lidam com arquivos binários /numPy (com extensão **np**)
- **As funções loadtxt() e savetxt()** lidam com arquivos de texto normais

NumPy apresenta um formato de arquivo simples para objetos ndarray. Este arquivo **.np** armazena dados, forma, tipo e outras informações necessárias para reconstruir o ndarray em um arquivo de disco, de modo que o array seja recuperado corretamente, mesmo se o arquivo estiver em outra máquina com arquitetura diferente.

numpy.save()

O arquivo **numpy.save()** armazena a matriz de entrada em um arquivo de disco com extensão **np** .

```
import numpy as np
a = np.array([1,2,3,4,5])
np.save('outfile',a)
```

Para reconstruir o array de **outfile.npy** , use a função **load()** .

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

Ele produzirá a seguinte saída -

```
array([1, 2, 3, 4, 5])
```

As funções save() e load() aceitam um parâmetro booleano adicional **Allow_pickle** . Um pickle em Python é usado para serializar e desserializar objetos antes de salvar ou ler um arquivo em disco.

salvartxt()

O armazenamento e recuperação de dados de array em formato de arquivo de texto simples é feito com as funções **savetxt()** e **loadtxt()** .

Exemplo

```
import numpy as np

a = np.array([1,2,3,4,5])
np.savetxt('out.txt',a)
b = np.loadtxt('out.txt')
print b
```

Ele produzirá a seguinte saída -

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

As funções `savetxt()` e `loadtxt()` aceitam parâmetros opcionais adicionais, como cabeçalho, rodapé e delimitador.