I/O with NumPy

The ndarray objects can be saved to and loaded from the disk files. The IO functions available are –

- **load()** and **save()** functions handle /numPy binary files (with **npy**extension)
- **loadtxt()** and **savetxt()** functions handle normal text files

NumPy introduces a simple file format for ndarray objects. This **.npy** file stores data, shape, dtype and other information required to reconstruct the ndarray in a disk file such that the array is correctly retrieved even if the file is on another machine with different architecture.

numpy.save()

The **numpy.save()** file stores the input array in a disk file with **npy**extension.

```
import numpy as np

a = np.array([1,2,3,4,5])

np.save('outfile',a)
```

To reconstruct array from **outfile.npy**, use **load()** function.

```
import numpy as np

b = np.load('outfile.npy')

print b
```

It will produce the following output –

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

The save() and load() functions accept an additional Boolean parameter **allow_pickles**. A pickle in Python is used to serialize and de-serialize objects before saving to or reading from a disk file.

savetxt()

The storage and retrieval of array data in simple text file format is done with **savetxt()** and **loadtxt()** functions.

Example

```
import numpy as np


a = np.array([1,2,3,4,5])

np.savetxt('out.txt',a)

b = np.loadtxt('out.txt')

print b
```

It will produce the following output –

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

The savetxt() and loadtxt() functions accept additional optional parameters such as header, footer, and delimiter.