

Library - NumPy

Jin Wang

MSIS, Rutgers

1. Introduction
 2. Create a basic array
 3. list-like Indexing and slicing
 4. Operations - 1: Basic
 5. Operations - 2: Broadcasting
 6. Conditionals used in NumPy
 7. Statistical calculations
- References

1. Introduction

Install Numpy

Windows

```
bash pip install numpy
```

Macbook

```
bash pip3 install numpy
```

What is NumPy?

- NumPy stands for *Numerical Python*.
- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- It is an open source project and you can use it freely.

Why Use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- *NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.*
- The array object in NumPy is called `ndarray`, it provides a lot of supporting functions that make working with `ndarray` very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

2. Create a basic array

```
>>> import numpy as np
>>> np.array([1, 2, 3])
array([1, 2, 3])
```

create an array filled with 0's and 1's:

```
>>> import numpy as np
>>> np.zeros(2)
array([0., 0.])
>>> np.ones(2)
array([1., 1.])
```


`np.arange()`

```
>>> import numpy as np
>>> np.arange(4)
array([0, 1, 2, 3])
```

`np.linspace()` --- create an array with values that are spaced linearly in a specified interval by using `np.linspace()`. Note that the boundaries would be included in the output, and the length of the output equals the parameter `num`.

```
>>> import numpy as np
>>> np.linspace(0, 10, num=6)
array([ 0.,  2.,  4.,  6.,  8., 10.] )
```

3. list-like Indexing and slicing

4. Operations - 1: Basic

element-wise calculation

```
import numpy as np
data_1 = [1, 2, 3]
data_2 = [11, 21, 31]

data_1, data_2 = np.array(data_1), np.array(data_2)

print(data_1 + data_2)
print(data_1 - data_2)
print(data_1 * data_2)
```

5. Operations - 2: Broadcasting

```
>>> import numpy as np
>>> data = np.array([1, 2, 3])
>>> data * 1.6
array([1.6, 3.2, 4.8])
```

6. Conditionals used in NumPy

```
>>> import numpy as np
>>> data = np.array([23, 2, 9, 8, 54, 89, 23])
>>> data > 10
array([ True, False, False, False,  True,  True,  True])
>>> data[data>10]
array([23, 54, 89, 23])
```

- example: to find odd numbers from a list.

```
>>> data = [23, 2, 9, 8, 54, 89, 23]
>>> data = np.array(l)
>>> odd_number = data[data%2!=0]
>>> odd_number
array([23,  9, 89, 23])
```

7. Statistical calculations

- min
- max
- sum
- standard deviation - `std`
- percentiles (Q_1 , Q_2 (median), Q_3 ,) - `np.percentile(data, perc)`

References

- https://numpy.org/doc/stable/user/absolute_beginners.html
- https://www.w3schools.com/python/numpy/numpy_intro.asp

End