NUMPY

Contents

- Why?
- Creating Numpy Arrays
- Attributes
- Subsetting, Slicing, Indexing
- Array Manipulation
- Vectorize function
- Mathematical Operations,Broadcasting
- Statistical Functions,Trigonometric Functions, Logarithmic Functions
- Linear Algebraic Functions
- Reading image data into a numpy array
- Reading sklearn datasets as numpy array

Why Numpy

- Provides efficient storage
- Meant for creating homogeneous n dimensional array
- Better ways of handling data for Mathematical Operations
- Linear algebra, Statistical operations, Matrix operations
- Shapes manipulation.
- Random number generation
- Building block for other packages

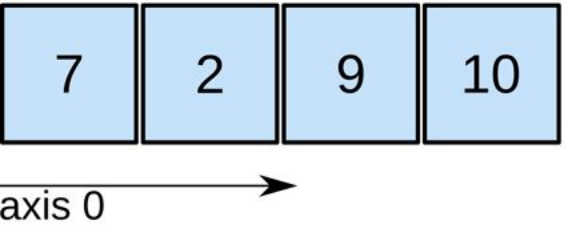
Importing Package

- Importing the package
  - *import numpy*
- Importing the package with alias
  - *import numpy as np*
- Importing a function from package
  - *from numpy import mean*
- Importing a packages from packag*e*
  - *from numpy import random*
  - *from numpy import linalg*
- Importing all
  - *from numpy import **

Array Creation

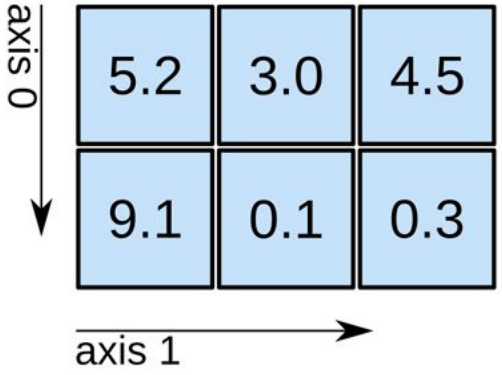
- From List
  - *np.array([7,2,9,10])*
- From Tuple
  - *np.array((20,30,14))*
- Generating array elements in Given Range
  - *np.arange(10)*
  - *np.arange(2, 3.2, 0.1)*
  - *np.linspace(1, 3, 15)*
- Multidimensional arrays

1D array



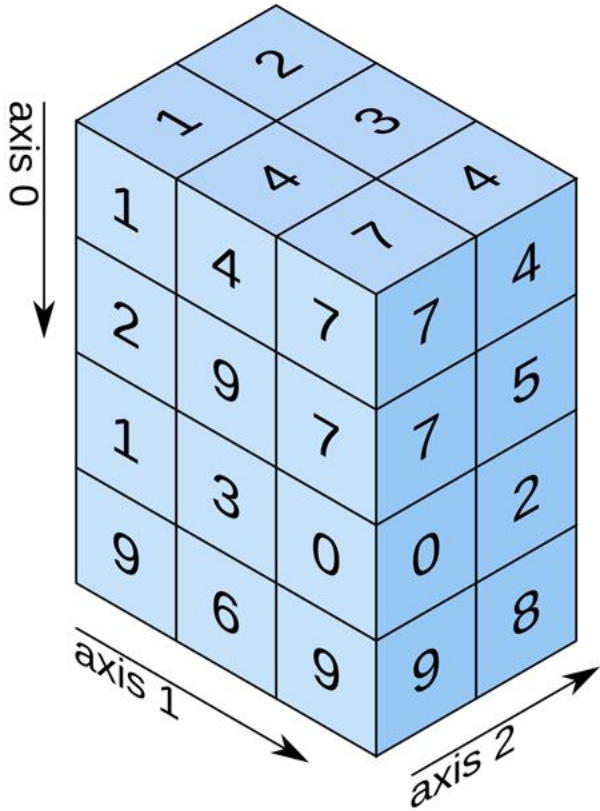
shape: (4,)

2D array



shape: (2, 3)

3D array



shape: (4, 3, 2)

Array creation - using numpy functions

- Initialized
  - *np.zeros((2,2))*
  - *np.ones((1,2))*
  - *np.full((2,2), 7)*
- Uninitialized
  - *np.empty((2,3))*
- Diagonal values
  - *np.eye(2)*
  - *np.diag([4,5,6,8])*

0.	0.
0.	0.

1.	1.

7	7
7	7

1.	0.
0.	1.

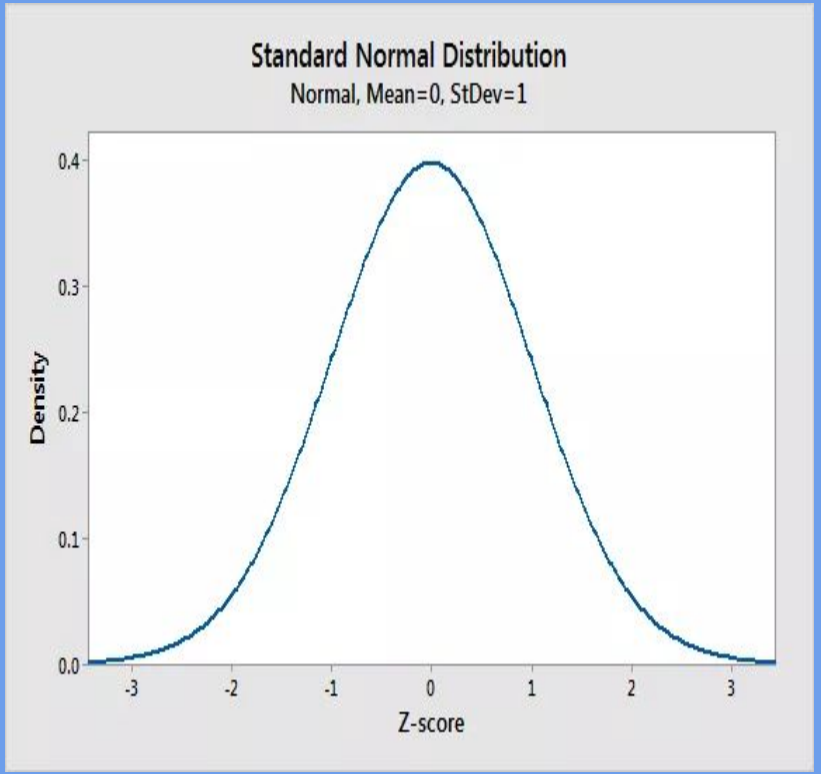
4	0	0	0
0	5	0	0
0	0	6	0
0	0	0	8

NumPy dtypes

Basic Type	Available NumPy types	Comments
Boolean	`bool`	Elements are 1 byte in size
Integer	`int8, int16, int32, int64, int128, int`	int defaults to the size of int in C for the platform
Unsigned Integer	`uint8, uint16, uint32, uint64, uint128, uint`	uint defaults to the size of unsigned int in C for the platform
Float	`float32, float64, float, longfloat,`	Float is always a double precision floating point value (64 bits). longfloat represents large precision floats. Its size is platform dependent.
Complex	`complex64, complex128, complex`	The real and complex elements of a complex64 are each represented by a single precision (32 bit) value for a total size of 64 bits.
Strings	`str, unicode`	Unicode is always UTF32 (UCS4)
Object	`object`	Represent items in array as Python objects.
Records	`void`	Used for arbitrary data structures in record arrays.

Normal(or Gaussian) Distribution or Bell curve

- The curve is symmetric about mean.
- The mean, median, and mode are all Equal.
- Half of the population is less than the mean and half is greater than the mean.



Array creation - using random package

- *np.random.randint(20), np.random.randint(2, 20)*
- *np.random.randint(2, 20, 7), np.random.randint(2, 20, (3,2))*
- *np.random.random((2,2))*
- *np.random.rand(4), np.random.rand(4,3)*
- *np.random.randn(2), np.random.randn(2, 4)*

Attributes

- *array.shape*
- *array.size*
- *array.ndim*
- *array.itemsize*
- *array.nbytes*
- *array.dtype*

Subsetting, Slicing & Indexing with 1d array

```
# Third element
print(array_1d[2])
# Specific elements
print(array_1d[[2, 5, 6]])
# Slice third element onwards
print(array_1d[2:])
# Slice first three elements
print(array_1d[:3])
# Slice third to seventh elements
print(array_1d[2:7])
# Subset starting 0 at increment of 2
print(array_1d[0::2])
```

Subsetting, Slicing & Indexing with 2d array

Fancy Indexing

array_1d[[True,False,False,True,True,False,False,True,True,False]]

 # Fancy Indexing Boolean array as index

array_1d[array_1d>5]

Array Manipulation

- reshape
- flatten
- expand_dims
- squeeze
- stacking - hstack,vstack
- resize

Reference

https://docs.scipy.org/doc/numpy/reference/routines.array-manipulation.html

Mathematical operations

- array + 1
- 4.5 * array_1 + 3
- array % 2
- a1 + a2
- add.multiply,divide,mod,power

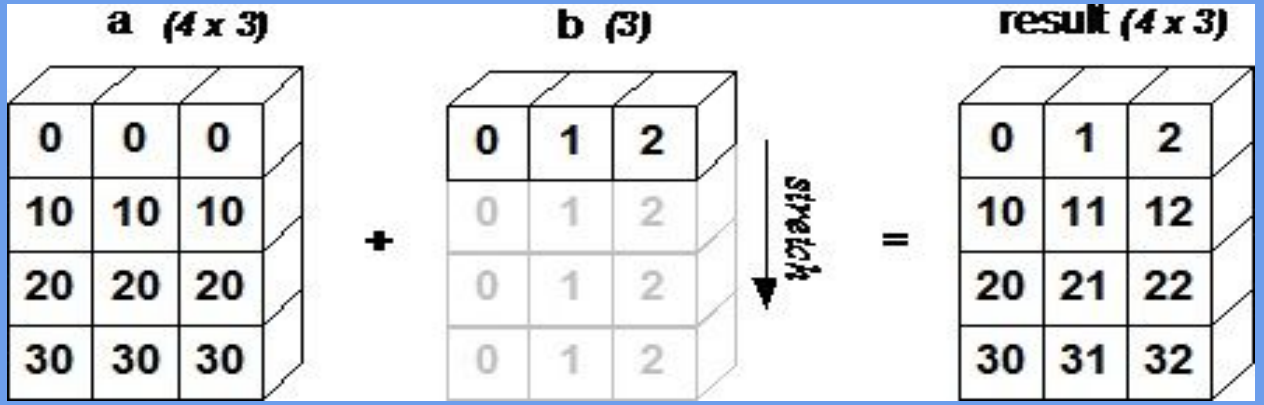
Broadcasting

how numpy treats arrays with different shapes during arithmetic operations.

the smaller array is “broadcast” across the larger array so that they have compatible shapes

Two dimensions are compatible when

- they are equal, or
- one of them is 1



Statistical Functions

Max,min,mean,std,median,quantile

Trigonometric Functions

Sin,cos,tan,arctan,arcsin,arccos,sinh,arcsinh,degrees,rad2deg

Logarithmic Functions

Log,log2,log10,log1p

Exponential Functions

exp,exp2,expm1

Reading Image data as ndarray

```
from matplotlib import pyplot as plt

img = plt.imread("ml.jpg")

img.shape

type(img)

plt.imshow()
```

Reading sklearn Dataset as ndarray

```
from sklearn.datasets import load_breast_cancer

data = load_breast_cancer()

print(data)

print(data['data'])

print(data['data'].shape
```

THANK YOU