

Numpy

Chen Chen

College of Computer Science

Nankai University

November 22, 2020



Outline

- **NDarray**
- Create
- Data Type
- Random
- Index
- Matrix
- Operation
- Analysis
- Linear Algebra



NDarray

N-Dimensional array (ndarray) in Numpy Array in Numpy is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers.

属性	说明
ndim	返回 int。表示数组的维数
shape	返回 tuple。表示数组的尺寸，对于 n 行 m 列的矩阵，形状为 (n,m)
size	返回 int。表示数组的元素总数，等于数组形状的乘积
dtype	返回 data-type。描述数组中元素的类型
itemsize	返回 int。表示数组的每个元素的大小（以字节为单位）。

Outline

- NDarray
- Create
- Data Type
- Random
- Index
- Matrix
- Operation
- Analysis
- Linear Algebra



Create

```
numpy.array(object, dtype=None,  
copy=True, order='K', subok=False,  
ndmin=0)
```

参数名称	说明
object	接收array。表示想要创建的数组。无默认。
dtype	接收data-type。表示数组所需的数据类型。如果未给定，则选择保存对象所需的最小类型。默认为None。
ndmin	接收int。指定生成数组应该具有的最小维数。默认为None。

Code Example

```
import numpy as np
```

```
nda1 = np.array([1,2,3,4])
```

```
nda2 = np.array([[1, 2, 3, 4],[4, 5, 6, 7], [7,  
8, 9, 10]])
```

```
nda1.ndim
```

```
nda2.shape
```

```
nda2.dtype
```

```
nda2.size
```

Create Cntd.

```
arange([start,] stop[, step,], dtype=None)
```

```
linspace(start, stop, num=50, endpoint=True,  
retstep=False, dtype=None, axis=0)
```

```
logspace(start, stop, num=50, endpoint=True,  
base=10.0, dtype=None, axis=0)
```

Code Example

```
np.arange(0,1,0.1)
```

```
np.arange(1,10,1)
```

```
np.linspace(0,10,11)
```

```
np.logspace(0,3,4)
```

```
np.logspace(0,4,5,base=2.0)
```



Create Cntd.

```
zeros(shape, dtype=float, order='C')
```

```
eye(N, M=None, k=0, dtype=<class 'float'>,  
order='C')
```

```
identity(n, dtype=None)
```

```
diag(v, k=0)
```

```
ones(shape, dtype=None, order='C')
```

Code Example

```
np.zeros((2,3))
```

```
np.eye(3)
```

```
np.identity(3)
```

```
np.diag([1,2,3,4,5]) np.ones((3,5))
```

```
np.ones((3,5))
```



Outline

- NDarray
- Create
- Data Type
- Random
- Index
- Matrix
- Operation
- Analysis
- Linear Algebra



Data Type

A numpy array is homogeneous, and contains elements described by a dtype object. A dtype object can be constructed from different combinations of fundamental numeric types.

类型	描述
bool	用一位存储的布尔类型（值为TRUE或FALSE）
int	由所在平台决定其精度的整数（一般为int32或int64）
int8	整数，范围为-128至127
int16	整数，范围为-32768至32767
int32	
.....

Data Type Cntd

创建一个存储餐饮企业库存信息的数据类型。其中，用一个长度为40个字符的字符串来记录商品的名称，用一个64位的整数来记录商品的库存数量，最后用一个64位的单精度浮点数来记录商品的价格。

```
df = np.dtype([("name", np.str_, 40), ("numitems",  
np.int64), ("price", np.float64)])
```

```
items = np.array([("tomatoes", 42, 4.14), ("cabbages", 13,  
1.72)], dtype=df)
```

Code Example

```
np.int("32")
```

```
np.float64(42)
```

```
np.bool(3)
```

```
np.float(10)
```

```
np.str(10)
```



Outline

- NDarray
- Create
- Data Type
- **Random**
- Index
- Matrix
- Operation
- Linear Algebra



Random

`random(size=None)`

`random_sample(size=None)`

`rand(d0, d1, ..., dn)`

`randn(d0, d1, ..., dn)`

`randint(low, high=None, size=None, dtype='N')`



Code Example

```
np.random.random(5)
```

```
np.random.random_sample(5)
```

```
np.random.random((2,5))
```

```
5 * np.random.random_sample((3, 2)) - 5
```

```
np.random.rand(3,2)
```

```
np.random.randint(2, size=10)
```

```
np.random.randint(1, 10, size=10)
```

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

```
np.random.choice(ary,10,p =  
[0.1,0.1,0.1,0.1,0.6])
```



Random



函数	说明
seed	确定随机数生成器的种子。
permutation	返回一个序列的随机排列或返回一个随机排列的范围。
shuffle	对一个序列进行随机排序。
binomial	产生二项分布的随机数。
normal	产生正态（高斯）分布的随机数。
beta	产生beta分布的随机数。
chisquare	产生卡方分布的随机数。
gamma	产生gamma分布的随机数。
uniform	产生在 $[0,1)$ 中均匀分布的随机数。

Outline

- NDarray
- Create
- Data Type
- Random
- **Index**
- Matrix
- Operation
- Analysis
- Linear Algebra



Index-One Dimensional Array

```
import numpy as np
```

```
ary = np.arange(10)
```

```
ary[4]
```

```
ary[-4]
```

```
ary[1:3]
```

```
ary[:3]
```

```
ary[1:]
```

```
ary[1:9:2]
```



Index-Multi Dimensional Array

```
ary2 = np.array([[1, 2, 3, 4, 5],[4, 5, 6, 7, 8],  
[7, 8, 9, 10, 11]])
```

```
ary2[1,2]
```

```
ary2[1,3:5]
```

```
ary2[1:,2:]
```

```
ary2[:,2:]
```

Index-Condition

```
arr = np.random.randint(1,20,10)
```

```
print(arr[(arr>1)&(arr<7)&(arr%2==0)])
```



Index-Reshape

```
a.reshape(shape, order='C')
```

```
a.ravel([order])
```

```
a.flatten(order='C')
```



Index-Reshape

```
ary = np.arange(12)
```

```
ary.reshape(3,4)
```

```
ary = ary.reshape(2,6)
```

```
ary.ravel()
```

```
ary.flatten()
```

```
ary.flatten()
```



Index-Stack

```
np.hstack((array1, array2))
```

```
np.vstack((array1, array2))
```

```
concatenate((a1, a2, ...), axis=0, out=None)
```

Code Example

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

```
b = np.array([[4,5,6]])
```

```
np.hstack((a,b))
```

```
np.vstack((a,b))
```

```
np.concatenate((a,b),axis=1)
```

```
np.concatenate((a,b),axis=0)
```



Index-Split

`hsplit(ary, indices_or_sections)`: Split an array into multiple sub-arrays horizontally (column-wise).

`vsplit(ary, indices_or_sections)`: Split an array into multiple sub-arrays vertically (row-wise).

`split(ary, indices_or_sections, axis=0)`: Split an array into multiple sub-arrays.

Code Example

```
x = np.arange(16.0).reshape(4, 4)
```

```
np.vsplit(x, 2)
```

```
np.hsplit(x,2)
```

```
np.split(x, 2, axis=1)
```

```
np.split(x, 2, axis=0)
```



Outline

- NDarray
- Create
- Data Type
- Random
- Index
- **Matrix**
- Operation
- Analysis
- Linear Algebra



Matrix-create

`asmatrix(data, dtype=None):`

Interpret the input as a matrix.

`matrix(data, dtype=None, copy=True):`

Returns a matrix from an array-like object or from a string of data.

属性	说明
T	返回自身的转置
H	返回自身的共轭转置
I	返回自身的逆矩阵
A	返回自身数据的2维数组的一个视图

Code Example

```
x=np.matrix([[1, 2], [3, 4]])
```

```
m = np.asmatrix(x)
```

```
m.T
```

```
m.I
```

```
np.asmatrix("1 2 12 13; 3 4 5 16; 17 8 9 10")
```

```
matrix1 +/- matrix2
```

```
matrix * 3
```

```
matrix1 * matrix2
```

```
np.multiply(matrix1, matrix2)
```

Matrix-ufunc

- `class ufunc(builtins.object)`
 - Functions that operate element by element on whole arrays.
- One dimensional array broadcasting

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} + [1 \ 2 \ 3] \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}$$

Matrix-ufunc



$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 3 & 3 & 3 \\ 5 & 5 & 5 \\ 7 & 7 & 7 \end{bmatrix}$$



Outline

- NDarray
- Create
- Data Type
- Random
- Index
- Matrix
- Operation
- Analysis
- Linear Algebra



Opertation

- Math
- I/O
- Char



Operation-Math(np.math)



fun	description	fun	description
ceil	Return the ceiling of x as an Integral	floor	Return the floor of x as an Integral.
radians	Convert angle x from degrees to radians.	degree	Convert angle x from radians to degrees.
cos	Return the cosine of x (measured in radians).	tan	Return the tangent of x (measured in radians).
sin	Return the sine of x (measured in radians).	exp	Return e raised to the power of x.
log	Return the logarithm of x to the given base.	pow(x,y)	Return $x^{**}y$ (x to the power of y).
fabs	Return the absolute value of the float x.	sqrt	Return the square root of x.
trunc	Truncates the Real x to the nearest Integral toward 0.	isnan	Return True if x is a NaN and False otherwise.

Operation-I/O

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

- Save an array to a binary file in NumPy ``.npy`` format

`load(file, mmap_mode=None,
allow_pickle=False, fix_imports=True,
encoding='ASCII')`

- Load arrays or pickled objects from ``.npy``, ``.npz`` or pickled files.

Operation-I/O

```
savetxt(fname, X, fmt='%.18e', delimiter=' ',  
newline='\n', header='', footer='',  
, comments='# ', encoding=None)
```

- Save an array to a text file.

```
loadtxt(fname, dtype=<class 'float'>,  
comments='#', delimiter=None, converters=None,  
one, skiprows=0, usecols=None,  
unpack=False, ndmin=0, encoding='bytes',  
max_rows  
=None)
```

- Load data from a text file.

Code Exmample

```
x = np.arange(10)
```

```
np.save(outfile, x)
```

```
np.load(outfile)
```

```
x = np.arange(0.0,5.0,1.0)
```

```
np.savetxt('test.out', x, delimiter=',')
```



Code Example Cntd

```
c = StringIO(u"1,0,2\n3,0,4")
```

```
x, y = np.loadtxt(c, delimiter=',', usecols=(0, 2), unpack=True)
```

x

```
array([1., 3.])
```

y

```
array([2., 4.])
```


Operation-Char(np.char)



fun	description	fun	description
capitalize	Return a copy with only the first character of each element capitalized.	isalpha	Returns true for each element if all characters in the string are alphabetic
count	Returns an array with the number of non-overlapping occurrences of substring	join	Return a string which is the concatenation of the strings in the sequence `seq`.
endswith	Returns a boolean array which is `True` where the string element in `a` ends with	lower	Return an array with the elements converted to lowercase.
find	For each element, return the lowest index in the string	partition	Partition each element in `a` around `sep`.
replace	For each element in `a`, return a copy of the string with all occurrences of substring `old` replaced by `new`.	strip	For each element in `a`, return a list of the words in the string, using `sep` as the delimiter string.

Outline

- NDarray
- Create
- Data Type
- Random
- Index
- Matrix
- Operation
- **Analysis**
- Linear Algebra




Analysis

- Sort
- Set
- Statistics



Analysis-Sort

- `numpy.sort(a, axis=-1, kind=None, order=None)`
 - Return a sorted copy of an array.
 - `ndarray.sort(axis=-1, kind=None, order=None)`
 - Sort an array in-place
 - `numpy.argsort(axis=-1, kind=None, order=None)`
 - Returns the indices that would sort this array.
- 

Code Example

```
a = np.array([1,3,2,4])  
print(np.sort(a))  
print(a)
```

```
a = np.array([1,3,2,4])  
a.sort()  
print(a)
```



Code Example

```
>>> a = np.array([[1,4],[3,1]])
```

```
>>> np.sort(a)
```

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

```
>>> np.sort(a, axis=None)
```

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

```
>>> np.sort(a, axis=0)
```

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



Analysis-Set

`intersect1d(ar1, ar2, assume_unique=False, return_indices=False)`

- Find the intersection of two arrays. Return the sorted, unique values that are in both of the input arrays.

`in1d(ar1, ar2, assume_unique=False, invert=False)`

- Test whether each element of a 1-D array is also present in a second arr

Analysis-Set

● `setdiff1d(ar1, ar2, assume_unique=False)`

- Find the set difference of two arrays.
- Return the unique values in `ar1` that are not in `ar2`.

● `union1d(ar1, ar2)`

- Find the union of two arrays.
- Return the unique, sorted array of values that are in either of the two input arrays.

● `unique(ar, return_index=False, return_inverse=False, return_counts=False, axis=None)`

- Find the unique elements of an array.

Analysis-Set

`union1d(ar1, ar2)`

- Find the union of two arrays. Return the unique, sorted array of values that are in either of the two input arrays.

`unique(ar, return_index=False, return_inverse=False, return_counts=False, axis=None)`

- Find the unique elements of an array.

Code Example

```
np.intersect1d([1, 3, 4, 3], [3, 1, 2, 1])
```

```
array([1, 3])
```

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

```
b = np.array([3, 4, 5, 6])
```

```
np.setdiff1d(a, b)
```

```
array([1, 2])
```



Code Example Cntd.

```
test = np.array([0, 1, 2, 5, 0])
```

```
states = [0, 2]
```

```
mask = np.in1d(test, states)
```

```
mask
```

```
<<<array([ True, False,  True, False,  True])
```

```
test[mask]
```

```
<<<array([0, 2, 0])
```

Code Example Cntd.

```
>>> a = np.array([1, 1, 2, 1])
```

```
>>> np.unique(a)
```

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

```
>>> np.union1d([-1, 0, 1], [-2, 0, 2])
```

```
array([-2, -1, 0, 1, 2])
```



Analysis-Statistics

当axis=0时，表示沿着纵轴计算。当axis=1时，表示沿着横轴计算。

函数	说明
sum	计算数组的和
mean	计算数组均值
median	计算中位数
std	计算数组标准差
var	计算数组方差
min	计算数组最小值
max	计算数组最大值
argmin	返回数组最小元素的索引
argmax	返回数组最大元素的索引
Cumsum	计算所有元素的累计和
cumprod	计算所有元素的累计积

Outline

- NDarray
- Create
- Data Type
- Random
- Index
- Matrix
- Operation
- Analysis
- Linear Algebra



Linear Algebra



Method	Description
norm	Vector or matrix norm
inv	Inverse of a square matrix
solve	Solve a linear system of equations
det	Determinant of a square matrix
lstsq	Solve linear least-squares problem
pinv	Pseudo-inverse using a singular value decomposition
matrix_power	Integer power of a square matrix
eig	Eigenvalues and vectors of a square matrix
eigh	Eigenvalues and eigenvectors of a Hermitian matrix
eigvals	Eigenvalues of a square matrix
eigvalsh	Eigenvalues of a Hermitian matrix
qr	QR decomposition of a matrix
svd	Singular value decomposition of a matrix
cholesky	Cholesky decomposition of a matrix

Code Example

```
A = np.array([[1, 2, 3], [2, -1, 1], [3, 0, -1]])  
b = np.array([9, 8, 3])  
x = np.linalg.solve(A, b)  
print(x)
```

```
A = np.random.randint(1,10,[3,3])  
np.linalg.svd(A)
```


End

Thanks for listening

