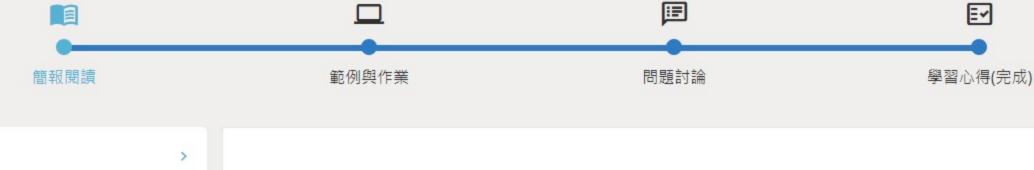
AI共學社群 > Python資料科學 > D08 NumPy 結構化陣 ... ctured Arrays)

D08 NumPy 結構化陣列 (Structured Arrays)









- 在NumPy中指定資料型別時,可以用字母、Python型別、NumPy通用資料型別來表示,下表列出各

字母

?

資料型別

boolean

signed byte b bytes np.bytes_ unsigned byte np.bytes_ В bytes

Python資料型別

bool

NumPy通用資料型別

np.bool_

unsigned integer	u		np.uint
floating-point	f	float	np.float_
omplex-floating point	С	complex	np.cfloat
timedelta	m	datetime.timedelta	np.timedelta64
datetime	М	datetime.datetime	np.datetime64
string	S, a	str	np.str_
Unicode string	U	str	np.str_

們混用了3種型別的表示方式:

• 資料型別也可以使用 NumPy 的資料型別, 例如 np.int32。

```
array([('Jay', 1, 2, 'Yes'), ('James', 3, 4, 'No'), ('Joe', 5, 6, 'Yes')], dtype=[('f0', '<U5'), ('f1', '<i8'), ('f2', '<i8'), ('f3', '<U3')])
```

a = np.genfromtxt("structured.txt", delimiter=',', dtype=dt)

NumPy 結構化陣列 (Structured Arrays)

建立結構化陣列可透過 dictionary 型別的資料建立 np.dtype 物件,並指定 dtype 給陣列。

dt = np.dtype({'names':('Name', 'num1', 'num2', 'True'), 'formats':((np.str_, 5), np.int32, int, 'U3')}) b = np.genfromtxt("structured.txt", delimiter=',', dtype=dt)

資料型別可以使用 Python 的資料型別、NumPy 的資料型別、或是字母代表的型別皆可。在範例中我

建立陣列後,可以用索引的方式存取元素資料。 b[0]

array(['Jay', 'James', 'Joe'], dtype='<U5')

也可以進行邏輯操作,取得對應的結果。

c = np.zeros(3, dtype=dt)

 $num_1 = [11, 12, 13]$

c['True'] = check

C

b['Name']

b[b['num2'] >= 3]['Name']

array(['James', 'Joe'], dtype='<U5')

dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])</pre> 將清單資料餵入結構化陣列中。

c['Name'] = name $c['num1'] = num_1$ $c['num2'] = num_2$

[('Chloe', 11, 14, 'Y') ('Charl', 12, 15, 'Y') ('Clara', 13, 16, 'N')] NumPy 結構化陣列:RecordArray

RecordArray 雖然方便但是在效能上會比原來的陣列差。使用方法如下:

('Clara', 13, 16, 'N')], dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])

原先我們是透過索引或是名稱存取元素值,但是 RecordArray 可以使用屬性的方式來取得。

RecordArray 與 Structured Array 非常類似,但是提供更多的屬性可以用來存取結構化陣列。不過

知識點回顧

array(['Chloe', 'Charl', 'Clara'], dtype='<U5')

使用,建議可以參照內容中提供的對照表。 除了數值資料之外,NumPy 陣列也可以儲存複合式資料,也就是包含不同資料型別的元素。這 就是結構化陣列 (Structured Arrays) 的功能,進行後續的資料存取及處理。

NumPy Structured arrays 官方文件

• 資料型別常在陣列中用到,NumPy的 dtype 使用彈性很大,並且可以與 Python 資料型別交互

index next previous Structured arrays Table of Contents Introduction Introduction Structured Datatype Creation Structured arrays are indarrays whose datatype is a composition of simpler datatypes organized as a sequence of named fields. For >>> x = np.array[[["Rex', 9, 81.9], ["Pido", 3, 27.0]], ... dtype=[["name", "ULO"], ["age", "14"], ["weight", "f4"]]] dtype=(('name', '010'), ('age', '<14'), ('weight', '<f4')])

Here x is a one-dimensional array of length two whose datatype is a structure with three fields: 1. A string of length 10 or less

named 'name', 2. a 32-bit integer named 'age', and 3. a 32-bit float named 'weight'.

dtype={('name', 'UlG'), ('age', '<14'), ('weight', '<f4')})

You can access and modify individual fields of a structured array by indexing with the field name:

If you index is at position 1 you get a structure:

array({('Res', 5, 81.), ('Fido', 5, 27.)],

array((9, 3), dtype=int32)
>>> x['age'] = 5

Previous topic Structured datatypes are designed to be able to mimic 'structs' in the C language, and share a similar memory layout. They are For these purposes they support specialized features such as subarrays, nested datatypes, and unions, and allow control over the memory layout of the structure. Writing custom array containers Chrick sparch Users looking to manipulate tabular data, such as stored in csy files, may find other pydata projects more suitable, such as xarray.

重要知識點

- NumPy 資料型別dtype及對應
- 種型別及其對應。
 - signed integer i int np.int_
- 透過 numpy.dtype 物件,指定要讀入各 Column 的資料型別,例如 f8 代表 float64 (8 bytes),U5 代 表長度 5 以下的 unicode 字串。 # 使用字母代表的資料型別 dt = np.dtype('U5, i8, i8, U3')

在讀取文字檔時,可將定義的資料型別指定給 dtype 引數。

array([('Jay', 1, 2, 'Yes'), ('James', 3, 4, 'No'), ('Joe', 5, 6, 'Yes')], dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])

('Jay', 1, 2, 'Yes') 也可以用Column名稱,取得Column所有元素值。

b[1]['True'] 'No'

新建立一個結構化陣列,方式跟建立陣列非常類似。 下例使用 zeros() 初始化陣列,並指定 dtype。

array([('', 0, 0, ''), ('', 0, 0, ''), ('', 0, 0, '')],

 $num_2 = [14, 15, 16]$ check = ['Y', 'Y', 'N']

name = ['Chloe', 'Charlotte', 'Clara']

print(c)

c_rec = c.view(np.recarray) rec.array([('Chloe', 11, 14, 'Y'), ('Charl', 12, 15, 'Y'),

c_rec.Name

網站: numpy

延伸閱讀

 Assigning data to a Structured Array
 Assignment from Python Native Types (Tuples) Assignment from Scalars
 Assignment from other Structured

NumPy

 Indexing Structured Arrays
 Accessing Individual Fields Accessing Multiple Fields
 Indexing with an integer to get a Structured Scalar Record Arrays

下一步:閱讀範例與完成作業