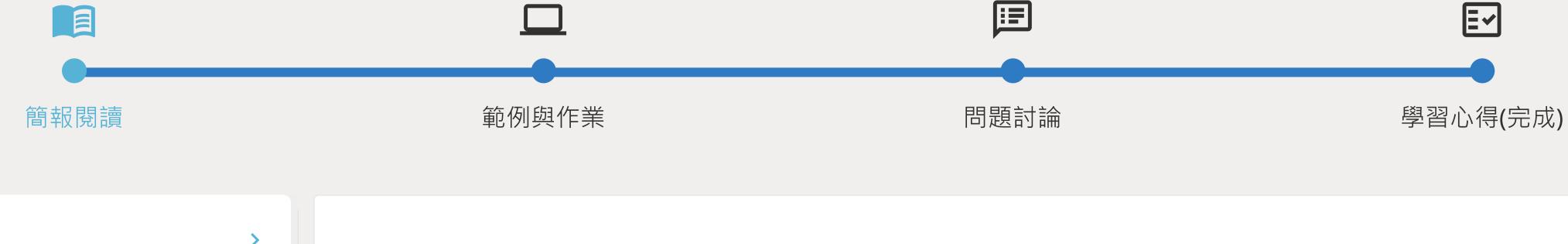
AI共學社群 > Python資料科學 > D08 NumPy 結構化陣 ... ctured Arrays)

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
- 種型別及其對應。

字母

В

boolean signed byte b

資料型別

unsigned byte

signed integer int np.int_ unsigned integer np.uint u

Python資料型別

bool

bytes

bytes

NumPy通用資料型別

np.bool_

np.bytes_

np.bytes_

floating-point float np.float_ complex complex-floating point np.cfloat С timedelta np.timedelta64 m datetime.timedelta datetime Μ datetime.datetime np.datetime64 S, a string str np.str_ Unicode string U str np.str_ NumPy 資料型別dtype 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

在讀取文字檔時,可將定義的資料型別指定給 dtype 引數。

dt = np.dtype('U5, i8, i8, U3')

使用字母代表的資料型別

資料型別也可以使用 NumPy 的資料型別,例如 np.int32。

```
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。

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

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

也可以用Column名稱,取得Column所有元素值。

b['Name']

dt = np.dtype({'names':('Name', 'num1', 'num2', 'True'), 'formats':((np.str_, 5), np.int32, int, 'U3')})

b[0] ('Jay', 1, 2, 'Yes')

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

取得單筆資料的欄位值。 b[1]['True']

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

'No'

也可以進行邏輯操作,取得對應的結果。

將清單資料餵入結構化陣列中。

 $num_1 = [11, 12, 13]$

c['Name'] = name

 $c['num1'] = num_1$

 $c['num2'] = num_2$

c['True'] = check

print(c)

c_rec

array(['James', 'Joe'], dtype='<U5') 新建立一個結構化陣列,方式跟建立陣列非常類似。

array([('', 0, 0, ''), ('', 0, 0, ''), ('', 0, 0, '')],
dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])

[('Chloe', 11, 14, 'Y') ('Charl', 12, 15, 'Y') ('Clara', 13, 16, 'N')]

dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])</pre>

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

NumPy 結構化陣列:RecordArray 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

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

c_rec.Name

● 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 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

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

Table of Contents Structured arrays Introduction Introduction Structured Datatypes Structured Datatype Creation

 Automatic Byte Offsets and Alignment Field Titles Union types Indexing and Assignment to Structured arrays Assigning data to a Structured Array Assignment from Python Native Types (Tuples) Assignment from Scalars Assignment from other Structured Assignment involving subarrays

NumPy index next previous NumPy.org Docs NumPy v1.19 Manual NumPy Tutorials NumPy basics Structured arrays Structured arrays are ndarrays whose datatype is a composition of simpler datatypes organized as a sequence of named fields. For Manipulating and Displaying Structured >>> x = np.array([('Rex', 9, 81.0), ('Fido', 3, 27.0)], dtype=[('name', 'U10'), ('age', 'i4'), ('weight', 'f4')]) array([('Rex', 9, 81.), ('Fido', 3, 27.)], dtype=[('name', 'U10'), ('age', '<i4'), ('weight', '<f4')])</pre> 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'. If you index x at position 1 you get a structure: >>> x[1] Indexing Structured Arrays ('Fido', 3, 27.0) Accessing Individual Fields Accessing Multiple Fields You can access and modify individual fields of a structured array by indexing with the field name: Indexing with an Integer to get a Structured Scalar >>> x['age'] Viewing Structured Arrays Containing array([9, 3], dtype=int32) Objects >>> x['age'] = 5 Structure Comparison Record Arrays array([('Rex', 5, 81.), ('Fido', 5, 27.)], Recarray Helper Functions dtype=[('name', 'U10'), ('age', '<i4'), ('weight', '<f4')]) Previous topic Structured datatypes are designed to be able to mimic 'structs' in the C language, and share a similar memory layout. They are Byte-swapping meant for interfacing with C code and for low-level manipulation of structured buffers, for example for interpreting binary blobs. Next topic 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 Ouick search Users looking to manipulate tabular data, such as stored in csv files, may find other pydata projects more suitable, such as xarray.

表長度 5 以下的 unicode 字串。

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

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

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 們混用了3種型別的表示方式:

建立陣列後,可以用索引的方式存取元素資料。

下例使用 zeros() 初始化陣列,並指定 dtype。 c = np.zeros(3, dtype=dt)

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:

c_rec = c.view(np.recarray)

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。

('Clara', 13, 16, 'N')],

rec.array([('Chloe', 11, 14, 'Y'), ('Charl', 12, 15, 'Y'),

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

知識點回顧

延伸閱讀

網站:<u>numpy</u>

下一步:閱讀範例與完成作業







