

Giới thiệu numpy

Numpy là một thư viện rất hữu ích trong xử lý mảng, ma trận, ... Nó là một thư viện không thể thiếu khi làm việc với xử lý số học trên python

Để sử dụng ta phải import thư viện và alias tên, numpy thường được gọi với tên "np" cho ngắn gọn

```
In [6]: import numpy as np
```

Các thao tác cơ bản

```
In [7]: a = np.array([1, 2, 3])    # Create a rank 1 array
print(type(a))                    # Prints "<class 'numpy.ndarray'>"
print(a.shape)                   # Prints "(3,)"

<class 'numpy.ndarray'>
(3,)
```

```
In [8]: print(a[0], a[1], a[2])    # Prints "1 2 3"
a[0] = 5                          # Change an element of the array
print(a)                          # Prints "[5, 2, 3]"
c = np.array( [ [1,2], [3,4] ], dtype=complex ) # complex number

1 2 3
[5 2 3]
```

```
In [9]: c = np.array( [ [1,2], [3,4] ], dtype=complex ) # complex number
print(c)

[[ 1.+0.j  2.+0.j]
 [ 3.+0.j  4.+0.j]]
```

Truy xuất phần tử

Các thao tác truy xuất phần tử cơ bản giống python.

Ngoài ra, numpy còn cung cấp các thao tác phức tạp hơn để truy xuất các phần tử thuận tiện hơn

```
In [10]: a = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(a.shape, '\n', a)

# simple
print('a[0] -> ', a[0])
print('a[1][2] -> ', a[1][2])
print('a[1,2] -> ' , a[1,2])

(3, 3)
[[1 2 3]
 [4 5 6]
 [7 8 9]]
a[0] ->  [1 2 3]
a[1][2] ->  6
a[1,2] ->  6
```

Một số thao tác truy xuất phức tạp hơn về array

```
In [11]: # range
print(a[:2,:2])

[[1 2]
 [4 5]]
```

```
In [ ]: print(a[:2,...])
```

```
In [ ]: print(a[:, 1])
```

Thao tác có điều kiện

```
In [ ]: print(a[a>3])

cnd = (a > 3)
print(cnd)

print(a[cnd])
```

Một số thao tác khởi tạo gán giá trị hữu ích

```
In [ ]: a = np.zeros((2,2))
print(a)
```

```
In [ ]: b = np.ones((1,2))    # Create an array of all ones
print(b)                      # Prints "[[ 1.  1.]]"
```

```
In [ ]: c = np.full((2,2), 7) # Create a constant array
print(c)
```

```
In [ ]: e = np.random.random((2,2)) # Create an array filled with random values
print(e)                             # Might print "[[ 0.91940167  0.08143941]
#                                     [ 0.68744134  0.87236687]]"
```

```
In [ ]: # Create an empty array
print(np.empty((3,2)))
```

```
In [ ]: # Create an array of evenly-spaced values
        print(np.arange(10,125,5))

In [ ]: # Create an array of evenly-spaced values
        print(np.linspace(0,2,9))

In [ ]: print(np.eye(2))           # Create a 2x2 identity matrix

In [ ]: print(np.identity(3))
```

Thêm bớt phần tử vào array

```
In [5]: a = np.arange(24)
        a = a.reshape((2,3,4))
        a

-----
NameError                                Traceback (most recent call last)
<ipython-input-5-03b413f4e5f6> in <module>()
----> 1 a = np.arange(24)
      2 a = a.reshape((2,3,4))
      3 a

NameError: name 'np' is not defined
```

```
In [ ]: new_a = np.append(a, [0,1,2,3])
        new_a

In [ ]: a = np.arange(6).reshape((2,3))
        print(a)
        np.append(a, np.asarray([[1],[2]]), axis=1)

In [ ]: np.append(a, np.asarray([[8,9,10]]), axis=0)

In [ ]: np.append(a, np.asarray([[8,9,10], [11,12,13]]), axis=0)
```

Làm việc với file dữ liệu

Tải dữ liệu từ text file

Ví dụ file 'data.txt' ở thư mục data chứa nội dung như sau:

Value1	Value2	Value3
0.2536	0.1008	0.3857
0.4839	0.4536	0.3561
0.1292	0.6875	0.5929
0.1781	0.3049	0.8928
0.6253	0.3486	0.8791

```
In [ ]: # required file exist
x, y, z = np.loadtxt('data/data.txt',
                    skiprows=1,
                    unpack=True)

print('x = ', x)
print('y = ', y)
print('z = ', z)
```

Lưu dữ liệu vào file text

```
In [ ]: x = np.arange(0.0,5.0,1.0)
np.savetxt('data/test.out', x, delimiter=',')
```

NumPy Broadcasting

Đây là một trong những thao tác mạnh mẽ mà numpy hỗ trợ khi xử lý với dữ liệu dạng mảng.

Ví dụ dưới thực hiện cộng hai ma trận x và y để tạo ra ma trận z, trong trường hợp này x, y có cùng shape

```
In [ ]: # Initialize `x`
x = np.ones((3,4))

# Check shape of `x`
print(x.shape, '\n', x)

# Initialize `y`
y = np.random.random((3,4))

# Check shape of `y`
print(y.shape)

# Add `x` and `y`
z = x + y
print(z.shape) # same shape as x, y
print(z)
```

Trường hợp này thì khác

```
In [ ]: # print x info
print(x.shape, '\n', x)

# Initialize `y`
t = np.arange(4)
print(t.shape, t)

# Subtract `x` and `y`
print(x - y)
```

Phép toán này thì thế nào?

```
In [ ]: print(x + 5)  # what output?

# and
print(x.shape, '\n', x)
y1 = np.random.random((5,1,4))
print(y1)

# Add `x` and `y`
x + y1
```

Một số hàm hữu ích của numpy

`np.add()`, `np.subtract()`, `np.multiply()`, `np.divide()` and `np.remainder()`
`np.exp()` and `np.sqrt()`
`np.sin()` and `np.cos()`
`np.log()`
`np.dot()`

Làm việc với shape

```
In [ ]: a = np.arange(36)
print(a.shape, a)
b = np.resize(a, (9,4))
print(b.shape, '\n', b)
```

```
In [ ]: a.resize((4,9))
print('a info', a.shape, '\n', a)
```

```
In [ ]: d = a.reshape((36,))
print('d', d)
print('d size', d.size)
```

```
In [ ]: print(a)
print('flat ', a.ravel())
```

Một số nguồn tham khảo thêm về numpy:

- <http://www.numpy.org/> (<http://www.numpy.org/>)
- <https://docs.scipy.org/doc/numpy-dev/user/quickstart.html> (<https://docs.scipy.org/doc/numpy-dev/user/quickstart.html>)
- <https://www.datacamp.com/community/tutorials/python-numpy-tutorial> (<https://www.datacamp.com/community/tutorials/python-numpy-tutorial>)

Bài tập

Sinh viên được yêu cầu hoàn thành các bài tập sau đây và nộp cho thầy hướng dẫn

Bài nộp của sinh viên là chính là **file này** sau khi được đổi tên thành **MSSV.E8_introduction_numpy.ipynb** và đừng quên ghi thông tin sinh viên vào các ô ở dưới.

Địa chỉ nộp bài: <https://www.dropbox.com/request/Z45ugPBhZhVZDiApNp4m> (<https://www.dropbox.com/request/Z45ugPBhZhVZDiApNp4m>)

Deadline nộp bài: **22:00 thứ 2 tuần tiếp theo**

Điểm bài này sẽ được tổng hợp với điểm chấm trên lớp (nếu có) để ra điểm cuối cùng

Bài 1: Sinh viên hãy vào trang web sau để lấy dữ liệu và đọc mô tả về dữ liệu: <http://archive.ics.uci.edu/ml/datasets/Abalone> (<http://archive.ics.uci.edu/ml/datasets/Abalone>)

- 1) Viết đoạn chương trình để load dữ liệu từ file abalone.data vào
- 2) Đề xuất và viết code xử lý với trường dữ liệu nominal (trường sex)
- 3) Viết hàm để với mỗi dòng dữ liệu trả về dạng {'X': np.array, 't': np.int32} phù hợp với bài toán (yêu cầu hàm lazy evaluation)
- 4) Viết hàm get_batch(batch_size=16) sử dụng hàm ở 3 để trả về mỗi lần một "batch" với kích thước batch_size. Dạng của output {'X': np.array, 't': np.array}, với shape tương ứng là (batch_size, ?) và (batch_size,) (yêu cầu hàm lazy evaluation)

Sinh viên viết code tương ứng vào các phần bên dưới

Thông tin sinh viên:

```
In [1]: sid = 'MSSV'
        name = 'Tên Bạn'
```

```
In [2]: # Bai 2-1
```

```
In [3]: # Bai 2-2
```

```
In [4]: # Bai 2-3
```

```
In [ ]: # Bai 2-4
```

Câu hỏi/thảo luận