

CS677 Numpy Lesson 4 - Concatenate, Stack, and Split

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
```

vstack

```
In [2]: a = np.ones((4, 3))
b = np.zeros((2, 3))
```

```
In [3]: a
```

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

```
In [4]: b
```

```
Out[4]: array([[0., 0., 0.],
               [0., 0., 0.]])
```

```
In [5]: c = np.vstack([a, b])
c
```

```
Out[5]: array([[1., 1., 1.],
               [1., 1., 1.],
               [1., 1., 1.],
               [1., 1., 1.],
               [0., 0., 0.],
               [0., 0., 0.]])
```

```
In [6]: c.shape
```

```
Out[6]: (6, 3)
```

```
In [7]: # alternatively
        np.concatenate((a, b), axis = 0)
```

```
Out[7]: array([[1., 1., 1.],
               [1., 1., 1.],
               [1., 1., 1.],
               [1., 1., 1.],
               [0., 0., 0.],
               [0., 0., 0.]])
```

hstack

```
In [8]: a = np.ones((2, 4))
        b = np.zeros((2, 2))
```

```
In [9]: a
```

```
Out[9]: array([[1., 1., 1., 1.],
               [1., 1., 1., 1.]])
```

```
In [10]: b
```

```
Out[10]: array([[0., 0.],
                [0., 0.]])
```

```
In [11]: c = np.hstack((a, b))
        c
```

```
Out[11]: array([[1., 1., 1., 1., 0., 0.],
                [1., 1., 1., 1., 0., 0.]])
```

```
In [12]: c.shape
```

```
Out[12]: (2, 6)
```

```
In [13]: # alternatively
        np.concatenate([a, b], axis = 1)
```

```
Out[13]: array([[1., 1., 1., 1., 0., 0.],
                [1., 1., 1., 1., 0., 0.]])
```

In [14]: *# concatenating 1-D arrays*

```
x = np.array([1, 2, 3])
y = np.array([10, 20, 30, 40])

np.concatenate([x, y])
```

Out[14]: array([1, 2, 3, 10, 20, 30, 40])

Splitting of arrays

In [15]: `x = np.arange(10, 20)`

`x`

Out[15]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])

In [16]: *# into parts*

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

Out[16]: [array([10, 11, 12, 13, 14]), array([15, 16, 17, 18, 19])]

In [17]: `np.split(x, 5)`

Out[17]: [array([10, 11]),
array([12, 13]),
array([14, 15]),
array([16, 17]),
array([18, 19])]

In [18]: *# at index*

```
np.split(x, [4])
```

Out[18]: [array([10, 11, 12, 13]), array([14, 15, 16, 17, 18, 19])]

In [19]: *# Note array argument for splitting at index*

```
x1, x2 = np.split(x, [4])
```

In [20]: `x1`

Out[20]: array([10, 11, 12, 13])

```
In [21]: x2
```

```
Out[21]: array([14, 15, 16, 17, 18, 19])
```

```
In [22]: x1, x2, x3 = np.split(x, [4, 6])
```

```
In [23]: x1
```

```
Out[23]: array([10, 11, 12, 13])
```

```
In [24]: x2
```

```
Out[24]: array([14, 15])
```

```
In [25]: x3
```

```
Out[25]: array([16, 17, 18, 19])
```

hsplit

```
In [26]: x = np.arange(24).reshape((4, 6))  
x
```

```
Out[26]: array([[ 0,  1,  2,  3,  4,  5],  
                [ 6,  7,  8,  9, 10, 11],  
                [12, 13, 14, 15, 16, 17],  
                [18, 19, 20, 21, 22, 23]])
```

```
In [27]: x1, x2 = np.hsplit(x, 2)
```

```
In [28]: x1
```

```
Out[28]: array([[ 0,  1,  2],  
                [ 6,  7,  8],  
                [12, 13, 14],  
                [18, 19, 20]])
```

In [29]: `x2`

Out[29]: `array([[3, 4, 5],
[9, 10, 11],
[15, 16, 17],
[21, 22, 23]])`

In [30]: `# Note array argument for splitting at index`

`x1, x2 = np.hsplit(x, [2])`

In [31]: `x1`

Out[31]: `array([[0, 1],
[6, 7],
[12, 13],
[18, 19]])`

In [32]: `x2`

Out[32]: `array([[2, 3, 4, 5],
[8, 9, 10, 11],
[14, 15, 16, 17],
[20, 21, 22, 23]])`

In [33]: `x`

Out[33]: `array([[0, 1, 2, 3, 4, 5],
[6, 7, 8, 9, 10, 11],
[12, 13, 14, 15, 16, 17],
[18, 19, 20, 21, 22, 23]])`

In [34]: `x1, x2, x3 = np.hsplit(x, 3)`

In [35]: `x1`

Out[35]: `array([[0, 1],
[6, 7],
[12, 13],
[18, 19]])`

```
In [36]: x2
```

```
Out[36]: array([[ 2,  3],
               [ 8,  9],
               [14, 15],
               [20, 21]])
```

```
In [37]: x3
```

```
Out[37]: array([[ 4,  5],
               [10, 11],
               [16, 17],
               [22, 23]])
```

```
In [38]: x
```

```
Out[38]: array([[ 0,  1,  2,  3,  4,  5],
               [ 6,  7,  8,  9, 10, 11],
               [12, 13, 14, 15, 16, 17],
               [18, 19, 20, 21, 22, 23]])
```

```
In [39]: # Split at specified indices
```

```
x1, x2, x3 = np.hsplit(x, [1, 4])
```

```
In [40]: x1
```

```
Out[40]: array([[ 0],
               [ 6],
               [12],
               [18]])
```

```
In [41]: x2
```

```
Out[41]: array([[ 1,  2,  3],
               [ 7,  8,  9],
               [13, 14, 15],
               [19, 20, 21]])
```

```
In [42]: x3
```

```
Out[42]: array([[ 4,  5],
                [10, 11],
                [16, 17],
                [22, 23]])
```

```
In [43]: # alternatively
```

```
np.split(x, [1, 4], axis = 1)
```

```
Out[43]: [array([[ 0],
                [ 6],
                [12],
                [18]]), array([[ 1,  2,  3],
                [ 7,  8,  9],
                [13, 14, 15],
                [19, 20, 21]]), array([[ 4,  5],
                [10, 11],
                [16, 17],
                [22, 23]])]
```

vsplit

```
In [44]: x = np.arange(24).reshape((6, 4))
x
```

```
Out[44]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19],
                [20, 21, 22, 23]])
```

```
In [45]: x1, x2 = np.vsplit(x, 2)
```

```
In [46]: x1
```

```
Out[46]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11]])
```

In [47]: `x2`

Out[47]: `array([[12, 13, 14, 15],
[16, 17, 18, 19],
[20, 21, 22, 23]])`

In [48]: `# Note array argument for splitting at index`

`x1, x2 = np.vsplit(x, [2])`

In [49]: `x1`

Out[49]: `array([[0, 1, 2, 3],
[4, 5, 6, 7]])`

In [50]: `x2`

Out[50]: `array([[8, 9, 10, 11],
[12, 13, 14, 15],
[16, 17, 18, 19],
[20, 21, 22, 23]])`

In [51]: `x`

Out[51]: `array([[0, 1, 2, 3],
[4, 5, 6, 7],
[8, 9, 10, 11],
[12, 13, 14, 15],
[16, 17, 18, 19],
[20, 21, 22, 23]])`

In [52]: `x1, x2, x3 = np.vsplit(x, 3)`

In [53]: `x1`

Out[53]: `array([[0, 1, 2, 3],
[4, 5, 6, 7]])`

In [54]: `x2`

Out[54]: `array([[8, 9, 10, 11],
[12, 13, 14, 15]])`

In [55]: `x3`

Out[55]: `array([[16, 17, 18, 19],
[20, 21, 22, 23]])`

In [56]: `x`

Out[56]: `array([[0, 1, 2, 3],
[4, 5, 6, 7],
[8, 9, 10, 11],
[12, 13, 14, 15],
[16, 17, 18, 19],
[20, 21, 22, 23]])`

In [57]: `x1, x2, x3 = np.vsplit(x, [1, 4])`

In [58]: `x1`

Out[58]: `array([[0, 1, 2, 3]])`

In [59]: `x2`

Out[59]: `array([[4, 5, 6, 7],
[8, 9, 10, 11],
[12, 13, 14, 15]])`

In [60]: `x3`

Out[60]: `array([[16, 17, 18, 19],
[20, 21, 22, 23]])`

In [61]: `# alternatively`

`np.split(x, [1, 4], axis = 0)`

Out[61]: `[array([[0, 1, 2, 3]]), array([[4, 5, 6, 7],
[8, 9, 10, 11],
[12, 13, 14, 15]]), array([[16, 17, 18, 19],
[20, 21, 22, 23]])]`

Case Study - Image Merging

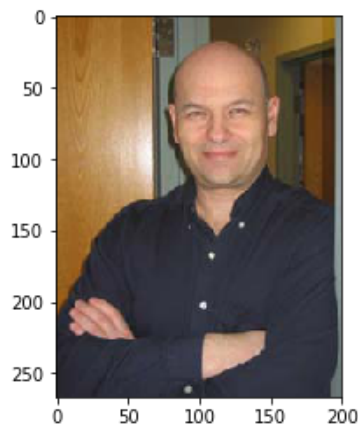
```
In [62]: # Python Imaging Library (Pillow)
```

```
from PIL import Image
import requests
```

```
In [63]: faculty = ['temkin', 'kalathur', 'braude', 'kanabar']
base_url = 'http://people.bu.edu/kalathur/datasets/'
images = [Image.open(requests.get(base_url + f + '.jpeg', stream=True).raw) for f in faculty]
images
```

```
Out[63]: [<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=200x267 at 0x113FE4EF0>,
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=155x210 at 0x113FE4F98>,
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=155x210 at 0x1140F16A0>,
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=155x210 at 0x113FF0CC0>]
```

```
In [64]: plt.imshow(np.asarray(images[0]));
```



```
In [65]: # sizes of the images
```

```
[img.size for img in images]
```

```
Out[65]: [(200, 267), (155, 210), (155, 210), (155, 210)]
```

```
In [66]: # resize all the images to be of the same size
```

```
sizes = sorted( [(np.sum(i.size), i.size) for i in images])
sizes
```

```
Out[66]: [(365, (155, 210)), (365, (155, 210)), (365, (155, 210)), (467, (200, 267))]
```

```
In [67]: desired_size = sizes[0][1]
         desired_size
```

```
Out[67]: (155, 210)
```

```
In [68]: # Now resize all images

         resized_images = [img.resize(desired_size) for img in images]
```

```
In [69]: # get 3-D array of each image
         image_arrays = [np.asarray(img) for img in resized_images]
```

```
In [70]: image_arrays[0].shape
```

```
Out[70]: (210, 155, 3)
```

```
In [71]: image_arrays[0][0][0]
```

```
Out[71]: array([145,  90,  26], dtype=uint8)
```

```
In [72]: merged_image = np.hstack(image_arrays)
         plt.imshow(merged_image)
         plt.xticks([])
         plt.yticks([]);
```



```
In [73]: Image.fromarray(merged_image).save('foo_h.jpeg')
```

```
In [74]: merged_image = np.vstack(image_arrays)
plt.imshow(merged_image)
plt.xticks([])
plt.yticks([]);
```



```
In [75]: Image.fromarray(merged_image).save('foo v.jpeg')
```