NumPy stack()

Summary: in this tutorial, you'll learn how to use the NumPy `stack()` function to join two or more arrays into a single array.

Introduction to the NumPy stack() function

The `stack()` function two or more arrays (https://www.pythontutorial.net/python-numpy/create-numpy-array/) into a single array. Unlike the `concatenate()` (https://www.pythontutorial.net/python-numpy/numpy-concatenate/) function, the stack() function joins 1D arrays to be one 2D array and joins 2D arrays to be one 3D array.

The following shows the syntax of the `stack()` function:

```
numpy.stack((a1,a2,...),axis=0)
```

In this syntax, the (a1, a2, ...) is a sequence of arrays with `ndarray` type or array-like objects. All arrays a1, a2, .. must have the same shape.

The `axis` parameter specifies the axis in the result array along which the function stacks the input arrays. By default, the axis is zero which joins the input arrays vertically.

Besides the `stack()` function, NumPy also has `vstack()` (https://www.pythontutorial.net/python-numpy/numpy-vstack/) function that joins two or more arrays vertically and `hstack()` (https://www.pythontutorial.net/python-numpy/numpy-hstack/) function that joins two or more arrays horizontally.

NumPy stack() function examples

Let's take some examples of using the `stack()` function.

1) Using stack() function to join 1D arrays

The following example uses the `stack()` function to join two 1D arrays:

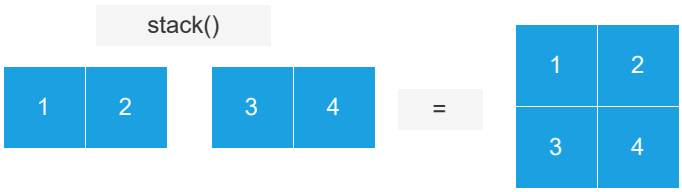
```
import numpy as np

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

c = np.stack((a, b))
print(c)
```

Output:

```
[[1 2]
 [3 4]]
```



The following example uses the `stack()` function to join two 1D arrays horizontally by using axis 1:

```
import numpy as np

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

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

c = np.stack((a, b), axis=1)
print(c)
```

Output:

```
[[1 3]
 [2 4]]
```

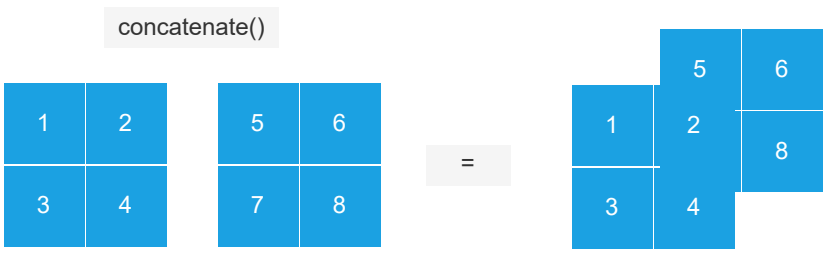
2) Using numpy stack() function to join 2D arrays

The following example uses the `stack()` function to join elements of two 2D arrays. The result is a 3D array:

```
import numpy as np

a = np.array([
    [1, 2],
    [3, 4]
])
b = np.array([
    [5, 6],
    [7, 8]
])

c = np.stack((a, b))
print(c)
print(c.shape)
```



Output:

```
[[[1 2]
  [3 4]]
 [[5 6]
  [7 8]]]
(2, 2, 2)
```

NumPy stack() vs. concatenate()

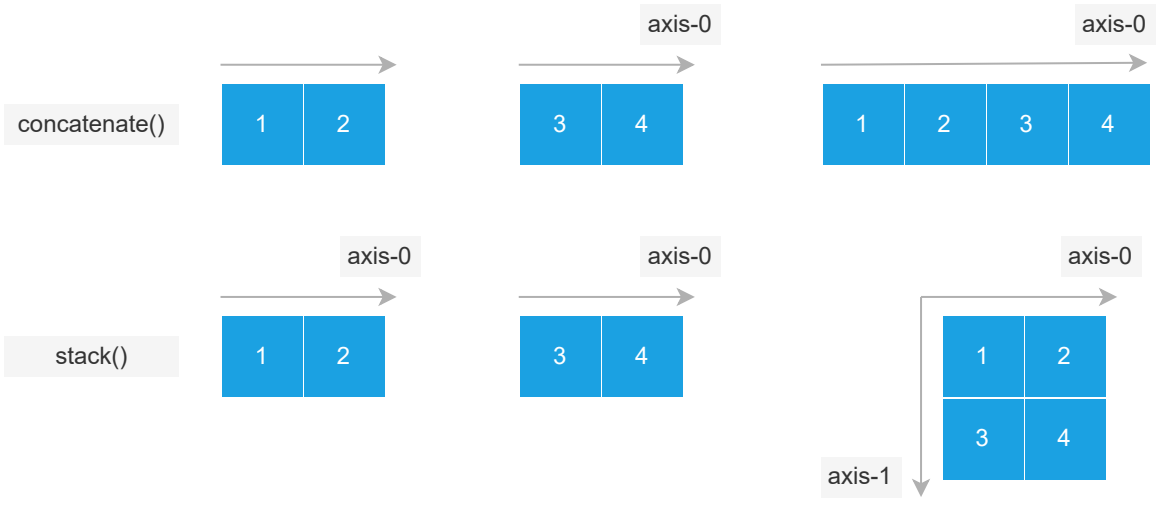
The following example illustrates the difference between `stack()` and `concatenate()` functions:

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

c = np.concatenate((a,b)) # return 1-D array
d = np.stack((a,b)) # return 2-D array
print(c)
print(d)
```

Output:

```
[1 2 3 4]
[[1 2]
 [3 4]]
```



In this example, the concatenate() function joins elements of two arrays along an *existing axis* while the `stack()` function joins the two arrays along a *new axis*.

Summary

- Use the numpy `stack()` function to join two or more arrays into one.