

17

Manipulating NumPy Arrays

NumPy 数组规整

重塑数组的维数、形状



我不能教任何人任何东西。我只能让他们思考。

I cannot teach anybody anything. I can only make them think.

—— 苏格拉底 (Socrates) | 古希腊哲学家 | 470 ~ 399 BC



```
▶ XXXXX
▶ XXXXX
▶ XXXXX
▶ XXXXX
▶ XXXXX
▶
```



本书前文介绍的 `numpy.swapaxes()`、`numpy.reshape()`、`numpy.resize()`、`numpy.transpose()`、`numpy.squeeze()`、`numpy.ravel()` 等等都算是对 NumPy 数组进行规整的函数。本章将介绍其他几种常用规整函数。

17.1 堆叠

沿行堆叠

用 `numpy.arange()` 产生如图 1 所示的两个一维等长数组。图 2 所示为三种办法将两个等长一维数组沿行 `axis = 0` 方向堆叠，结果为二维数组。

`numpy.stack()` 函数将沿着指定轴将多个数组堆叠在一起，返回一个新的数组；默认轴为 `axis = 0`。`numpy.row_stack()` 函数将多个数组沿着行方向进行堆叠，生成一个新的数组。`numpy.vstack()` 将多个数组沿着垂直方向（行方向）进行堆叠，生成一个新的数组。



图 1. 两个等长一维数组

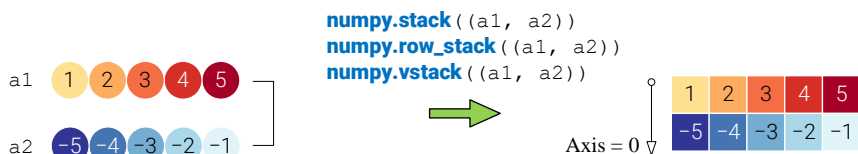


图 2. 沿行 `axis = 0` 方向堆叠

沿列堆叠

图 3 所示为沿列 `axis = 1` 方向堆叠两个一维等长数组。图中给出两种办法。

`numpy.column_stack()` 将多个一维数组沿着列方向进行堆叠，生成一个新的二维数组。

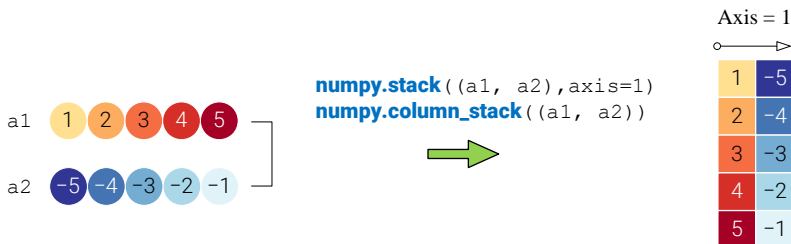


图 3. 沿列 `axis = 1` 方向堆叠

如图 4 所示，用 `numpy.hstack()` 堆叠一维数组的结果还是一个一维数组。`numpy.hstack()` 将多个数组沿着水平方向（列方向）进行堆叠，生成一个新的数组。为了获得图 3 结果，需要先将两个一维数组变形为列向量，然后用 `numpy.hstack()` 函数沿列堆叠，具体如图 5 所示。

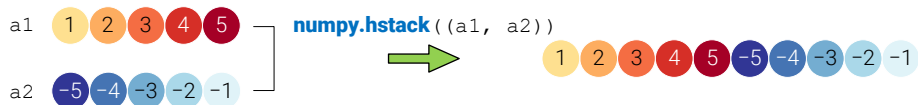


图 4. 沿列 axis = 1 方向堆叠，用 `numpy.hstack()`

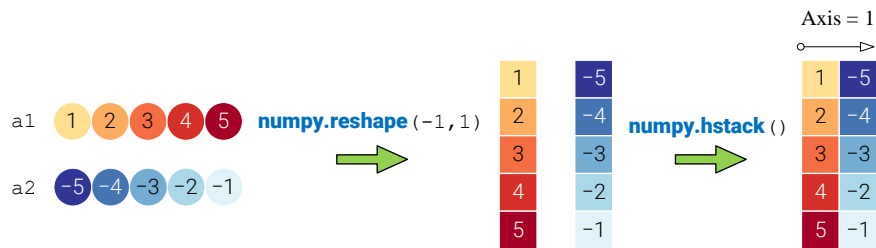


图 5. 沿列 axis = 1 方向堆叠，两个列向量

拼接

我们还可以用 `numpy.concatenate()` 完成数组拼接。如所示，利用 `numpy.concatenate()`，我们可以分别完成沿行、列方向数组拼接。

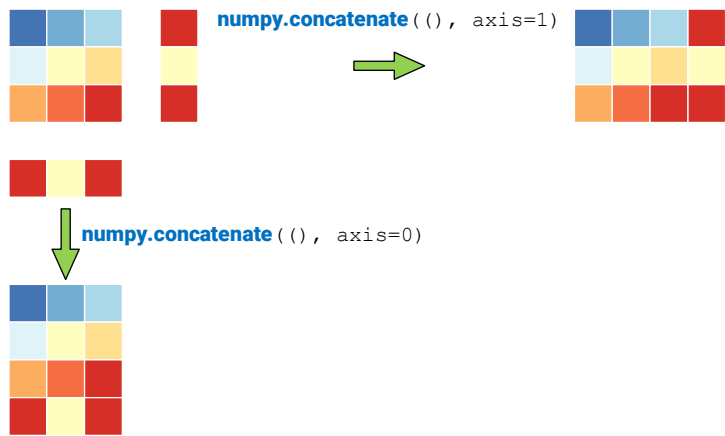


图 6. 用 `numpy.concatenate()` 拼接

堆叠结果为三维数组

此外，利用 `numpy.stack()`，我们还可以将二维数组堆叠为三维数组。图 7 所示为沿三个不同方向堆叠结果的效果图。

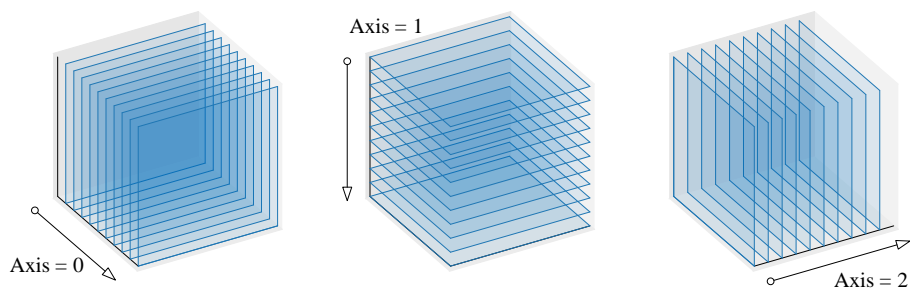


图 7. 沿着三个不同方向堆叠

举个例子，给定图 8 所示两个形状相同的二维数组。它俩按图 7 所示为沿三个不同方向堆叠的结果如图 9 所示。

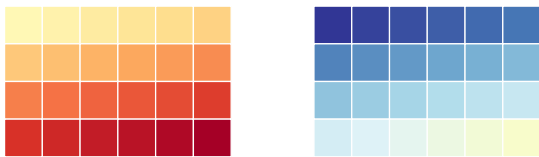


图 8. 两个形状相同的二维数组

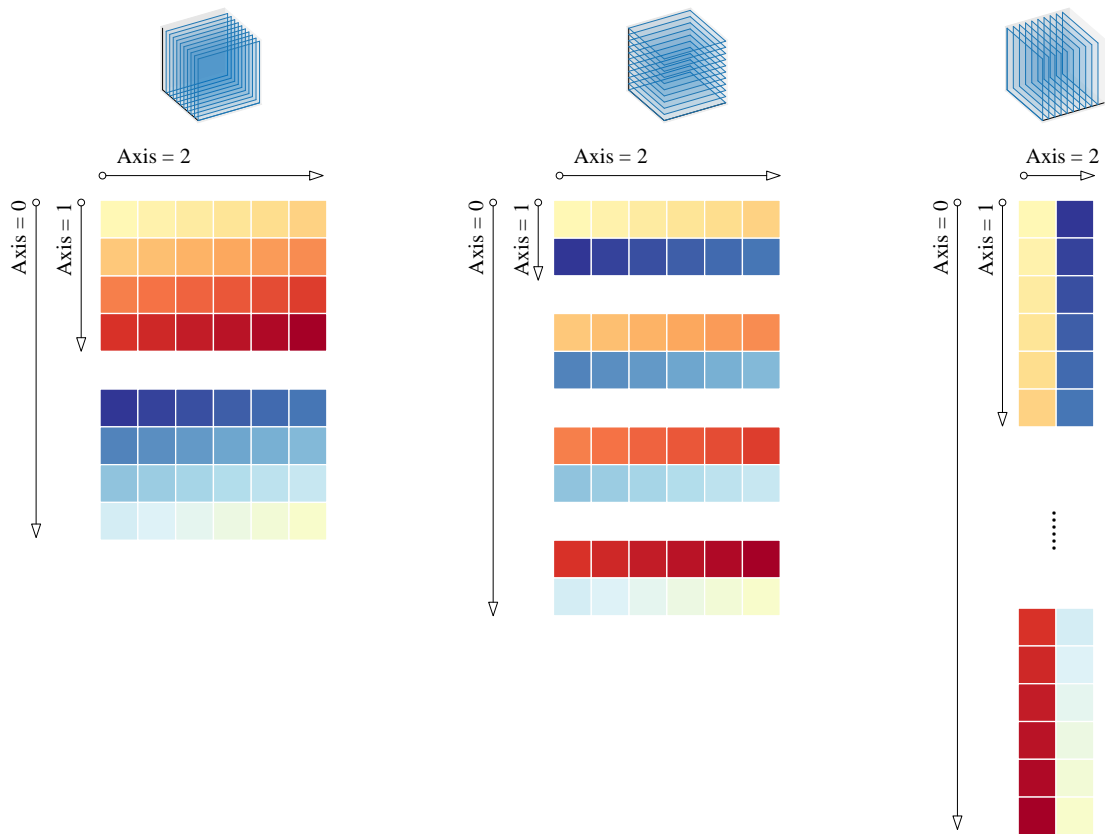


图 9. 得到三个不同的三维数组

17.2 重复

`numpy.repeat()` 和 `numpy.tile()` 都可以用来重复数据。`numpy.repeat()` 和 `numpy.tile()` 的区别在于重复的对象不同。`numpy.repeat()` 重复的是分别数组中的每个元素。`numpy.repeat()` 还可以指定具体的轴，以及不同元素重复的次数，请大家参考其技术文档。

`numpy.tile()` 重复的是整个数组，如图 11 所示。本章配套 Jupyter Notebook 还提供其他示例，请大家自行练习。

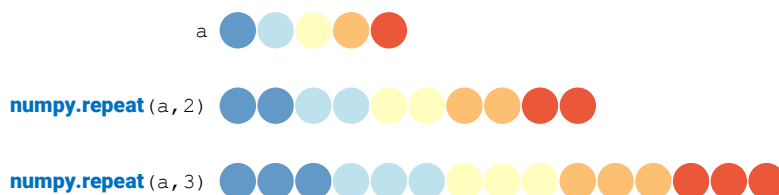


图 10. 利用 `numpy.repeat()` 重复一维数组

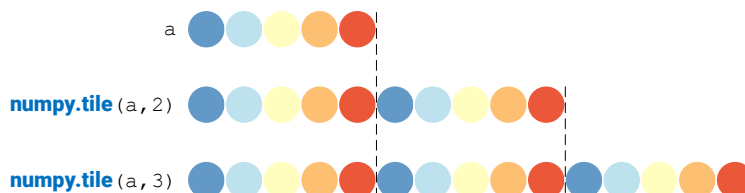


图 11. 利用 `numpy.tile()` 重复一维数组

17.3 分块矩阵

合成

`numpy.block()` 函数用于将多个数组沿不同的轴组合成一个分块矩阵。它接受一个嵌套列表作为输入，每个列表代表一个块矩阵，然后根据指定的轴将这些块矩阵组合在一起。

在图 12 给出的例子中，我们创建了四个小的矩阵，并使用 `numpy.block()` 函数将它们组合成一个分块矩阵 M 。

分块矩阵经常用来简化某些线性代数运算，鸢尾花书《矩阵力量》将专门介绍分块矩阵。

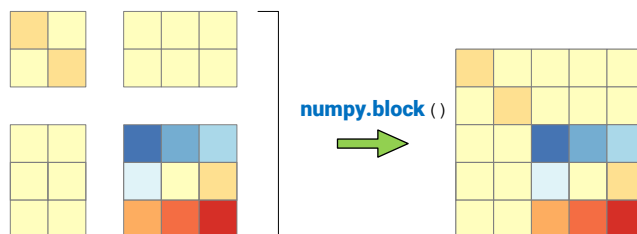


图 12. 四个二维数组合成一个矩阵



什么是分块矩阵?

分块矩阵是由多个小矩阵组合而成的大矩阵。它将一个大的矩阵划分为若干个小矩阵，这些小矩阵可以是实数矩阵、向量矩阵或者其他的矩阵形式。通常情况下，分块矩阵可以使用一个方括号将小矩阵组合在一起，然后按照一定的规则排列。分块矩阵可以简化一些复杂的矩阵计算，同时也常常用于表示具有特定结构的矩阵，例如对角矩阵或者上下三角矩阵等。

切割

numpy.split() 函数可以将一个数组沿指定轴分割为多个子数组。**numpy.split()** 接受三个参数：要分割的数组、分割的索引位置、沿着哪个轴进行分割。图 13 所示为将一个一维数组三等分得到三个子数组。本章配套的 Jupyter Notebook 中，大家可以看到如何设定分割索引位置，请自行练习。

图 14 所示为利用 **numpy.split()** 将二维数组沿不同轴三等分。大家也可以分别尝试使用 **numpy.hsplit()** 和 **numpy.vsplit()** 完成类似操作。本章配套 Jupyter Notebook 中还介绍如何使用 **numpy.append()**、**numpy.insert()**、**numpy.delete()** 完成附加、插入、删除操作，请大家自行学习。

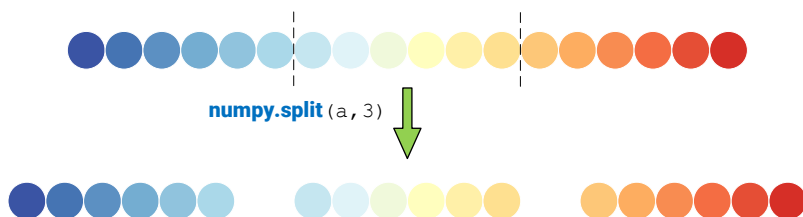


图 13. 将一维数组三等分

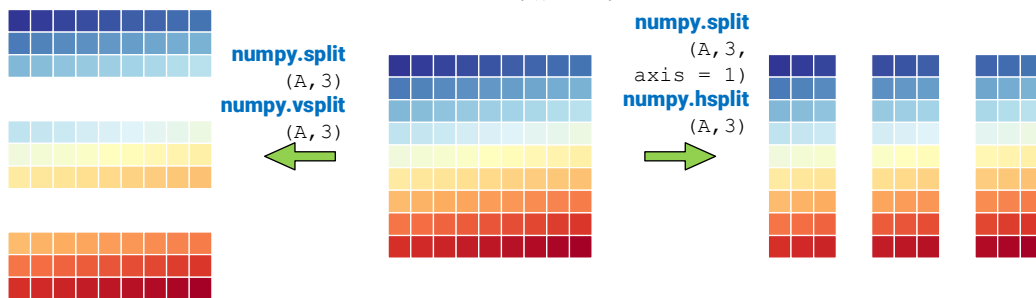


图 14. 将二维数组三等分，沿不同轴



下面，是有关 NumPy 数组规整的三道习题，请大家完成。

Q1. 请生成 $[0, 1]$ 区间内的连续均匀两个随机数数组，数组形状为 $(10,)$ 。将它俩分别按行、按列堆叠起来形成二维数组。

Q2. 请生成 $[0, 1]$ 区间内的连续均匀一个随机数数组，数组形状为 $(12, 12)$ 。将它分别按行、按列三等分。

Q3. 请生成 $[0, 1]$ 区间内的连续均匀两个随机数数组，数组形状分别为 $(8, 5)$ 、 $(3, 5)$ 。用几种不同办法将它们拼接成一个数组。

* 这三道题目很基础，本书不给答案。