

[首页](#) [JAVA](#) [PHP](#) [HTML](#) [CSS](#) [CSS3](#) [JAVASCRIPT](#) [Sass](#) [MYSQL](#)
[SQL](#) [PL/SQL](#) [SQLITE](#) [REDIS](#)

[首页](#) > [编程笔记](#) > [python](#) > [Python scipy.ndimage.convolve实例讲解](#)

Python scipy.ndimage.convolve实例讲解

时间:2022-04-07

本文章向大家介绍Python scipy.ndimage.convolve实例讲解，主要分析其语法、参数、返回值和注意事项，并结合实例形式分析了其使用技巧，希望通过本文能帮助到大家理解应用这部分内容。

用法:

```
scipy.ndimage.convolve(input, weights, output=None, mode='reflect', cval=0.0, origin=0)
```

多维卷积。

该数组与给定的内核卷积。

参数:

input: array_like

输入数组。

weights: array_like

权重数组，与输入的维数相同

output: 数组或数据类型，可选

放置输出的数组，或返回数组的 dtype。默认情况下，将创建一个与输入具有相同 dtype 的数组。

mode: { 'reflect' , 'constant' , 'nearest' , 'mirror' , 'wrap' }, 可选

mode 参数确定输入数组如何扩展至其边界之外。默认为 'reflect' 。每个有效值的行为如下：

'reflect' (d c b a | a b c d | d c b a)

通过反射最后一个像素的边来扩展输入。此模式有时也称为half-sample 对称模式。

'constant' (k k k k | a b c d | 呸呸呸呸)

通过使用 cval 参数定义的相同常量值填充边之外的所有值来扩展输入。

'nearest' (啊啊啊啊啊 | a b c d | 啊啊啊啊啊)

通过复制最后一个像素来扩展输入。

'mirror' (d c b | a b c d | c b a)

通过反射最后一个像素的中心来扩展输入。此模式有时也称为whole-sample 对称模式。

'wrap' (a b c d | a b c d | A B C D)

通过环绕到相对边来扩展输入。

为了与插值函数保持一致，还可以使用以下模式名称：

'grid-mirror'

这是 'reflect' 的同义词。

'grid-constant'

这是 'constant' 的同义词。

'grid-wrap'

这是 'wrap' 的同义词。

cval: 标量，可选

如果模式为 'constant' , 则填充过去输入边的值。默认值为 0.0

origin: int 可选

控制输入信号的原点, 这是滤波器产生输出第一个元素的中心位置。正值将过滤器向右移动, 负值将过滤器向左移动。默认值为 0。

返回:

result: ndarray

输入与权重的卷积结果。

注意:

结果中的每个值都是 $C_i = \sum_j I_{i+k-j} W_j$, 其中 W 是权重内核, j 是 N-D 空间索引 W , i 是输入 k 是 W 中心的坐标, 由下式指定起源在输入参数中。

例子:

也许最容易理解的情况是 `mode='constant'`, `cval=0.0`, 因为在这种情况下边界(即, 权重内核, 以任何一个值为中心, 延伸到边之外输入) 被视为零。

```
>>> a = np.array([[1, 2, 0, 0],
...               [5, 3, 0, 4],
...               [0, 0, 0, 7],
...               [9, 3, 0, 0]])
>>> k = np.array([[1, 1, 1], [1, 1, 0], [1, 0, 0]])
>>> from scipy import ndimage
>>> ndimage.convolve(a, k, mode='constant', cval=0.0)
array([[11, 10, 7, 4],
       [10, 3, 11, 11],
       [15, 12, 14, 7],
       [12, 3, 7, 0]])
```

环境 `cval=1.0` 相当于填充的外边输入 1.0 (然后只提取结果的原始区域)。

```
>>> ndimage.convolve(a, k, mode='constant', cval=1.0)
array([[13, 11, 8, 7],
```

```
[11, 3, 11, 14],
[16, 12, 14, 10],
[15, 6, 10, 5]])
```

和mode='reflect' (默认), 外部值反映在边输入填写缺失值。

```
>>> b = np.array([[2, 0, 0],
...               [1, 0, 0],
...               [0, 0, 0]])
>>> k = np.array([[0, 1, 0], [0, 1, 0], [0, 1, 0]])
>>> ndimage.convolve(b, k, mode='reflect')
array([[5, 0, 0],
       [3, 0, 0],
       [1, 0, 0]])
```

这包括在角落的对角线。

```
>>> k = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])
>>> ndimage.convolve(b, k)
array([[4, 2, 0],
       [3, 2, 0],
       [1, 1, 0]])
```

和mode='nearest', 中的单个最接近的值输入根据需要重复多次以匹配重叠权重。

```
>>> c = np.array([[2, 0, 1],
...               [1, 0, 0],
...               [0, 0, 0]])
>>> k = np.array([[0, 1, 0],
...               [0, 1, 0],
...               [0, 1, 0],
...               [0, 1, 0],
...               [0, 1, 0]])
>>> ndimage.convolve(c, k, mode='nearest')
array([[7, 0, 3],
       [5, 0, 2],
       [3, 0, 1]])
```

随机文章

[具有张量流的混合密度网络](#)

[SDL系列讲解（八） SDL_ttf教程](#)

[一步一步创建ASP.NET MVC5程序\[Repository+A](#)

[android am 命令简单讲解\(一\)](#)

[android dumphsys 快速入门](#)

[新年伊始，微软再遭叙利亚电子军挑衅](#)

[android调用dialog.hide\(\)引起的输入事件派发错](#)

[SDL系列讲解（七） SDL_image教程](#)

[一步一步创建ASP.NET MVC5程序\[Repository+A](#)

[android SDL系列讲解（十三） 播放音乐库 SDL_I](#)

[渗透测试专用版Linux：BackBox3.13发布](#)

[不容小觑的金融诈骗手段 – 419诈骗](#)

[任天堂3DS游戏机烧录卡蓝屏事件](#)

[互联网世界的毒瘤——僵尸网络](#)



本站知识点必读

[JavaScript 教程](#)

[JavaScript 与Java](#)

[JavaScript 特殊数据类型](#)

[JavaScript 编辑工具](#)

[JavaScript 数据结构](#)

[JavaScript 运算符](#)

[JavaScript 与HTML](#)

[JavaScript 基本数据类型](#)

[JavaScript typeof 运算符](#)

JavaScript 表达式	JavaScript 类型转换	JavaScript 基本语法
JavaScript 注释	Javascript 基本处理流程	Javascript 选择结构
Javascript if 语句	Javascript if 语句的嵌套	Javascript switch 语句
Javascript 循环结构	Javascript 循环结构实例	Javascript 跳转语句
Javascript 控制语句总结	Javascript 函数介绍	Javascript 函数的定义
Javascript 函数调用	Javascript 几种特殊的函数	JavaScript 内置函数简介
Javascript eval() 函数	Javascript isFinite() 函数	Javascript isNaN() 函数
parseInt() 与 parseFloat()	escape() 与 unescape()	Javascript 字符串介绍
Javascript length属性	javascript 字符串函数	Javascript 日期对象简介
Javascript 日期对象用途	Date 对象属性和方法	Javascript 数组是什么
Javascript 创建数组	Javascript 数组赋值与取值	Javascript 数组属性和方法

最近更新

Roarctf 2019 easy_pwn

[V&N公开赛]simpleHeap

PWN Chunk Extend

python测试开发django-77.ORM如何添加 DateTimeField 不显示毫秒

python测试开发django-75.ORM根据日期查询(__range)

python测试开发django-74.auth认证之is_active

号主无聊时间做了个Win10家庭中文版ISO镜像，支持写入U盘做启动安装盘等！

在PowerBI中创建时间表（非日期表）

python测试开发django-76.ORM查询之Q查询

Zipack初体验：我的开源标准！

Python 中的元类到底是什么？这篇恐怕是最清楚的了

[CVE-2020-1948] Apache Dubbo Provider反序列化漏洞复现及分析

Pytorch optimizer.step() 和loss.backward()和scheduler.step()的关系与区别

看看有哪些 Web 攻击技术.

jupyter lab最强代码补全插件

