如何通俗易懂地解释卷积?

palet

对卷积的困惑

卷积这个概念，很早以前就学过，但是一直没有搞懂。教科书上通常会给出定义，给出很多性质，也会用实例和图形进行解释，但究竟为什么要这么设计，这么计算，背后的意义是什么，往往语焉不详。作为一个学物理出身的人，一个公式倘若倘若给不出结合实际的直观的通俗的解释（也就是背后的 “物理” 意义），就觉得少了点什么，觉得不是真的懂了。

教科书上一般定义函数 f, g 的卷积 $f * g(n)$ 如下：

连续形式：

$$(f * g)(n) = \int_{-\infty}^{\infty} f(\tau)g(n-\tau)d\tau$$

离散形式：

$$(f * g)(n) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(n-\tau)$$

并且也解释了，先对 g 函数[o] 进行翻转，相当于在数轴上把 g 函数从右边褶到左边去，也就是卷积的 “卷” 的由来。

然后再把 g 函数平移到 n，在这个位置对两个函数的对应点相乘，然后相加，这个过程是卷积的 “积” 的过程。

这个只是从计算的方式上对公式进行了解释，从数学上讲无可挑剔，但进一步追问，为什么要先翻转再平移，这么设计有何用意？还是有点费解。

在知乎，已经很多的热心网友对卷积举了很多形象的例子进行了解释，如卷地毯、丢骰子、打耳光、存钱等等。读完觉得非常生动有趣，但过细想想，还是感觉有些地方还是没解释清楚，甚至可能还有瑕疵，或者还可以改进（这些后面我会做一些分析）。

带着问题想了两个晚上，终于觉得有些问题想通了，所以就写出来跟网友分享，共同学习提高。不对的地方欢迎评论拍砖。。。

明确一下，这篇文章主要想解释两个问题：

1. 卷积这个名词是怎么解释？“卷” 是什么意思？“积” 又是什么意思？

2. 卷积背后的意义是什么，该如何解释？

考虑的应用场景

为了更好地理解这些问题，我们先给出两个典型的应用场景：

1. 信号分析

一个输入信号*f(t)*，经过一个线性系统（其特征可以用单位冲击响应函数*g(t)*描述）以后，输出信号应该是什么？实际上通过卷积运算就可以得到输出信号。

2. 图像处理

输入一幅图像*f(x,y)*，经过特定设计的卷积核*g(x,y)*进行卷积处理以后，输出图像将会得到模糊，边缘强化等各种效果。

对卷积的理解

对卷积这个名词的理解：**所谓两个函数的卷积，本质上就是先将一个函数翻转，然后进行滑动叠加。**

在连续情况下，叠加指的是对两个函数的乘积求积分，在离散情况下就是 加权求和 ，为简单起见就统一称为叠加。

整体看来是这么个过程：

翻转——> 滑动——> 叠加——> 滑动——> 叠加——> 滑动——> 叠加.....

多次滑动得到的一系列叠加值，构成了卷积函数。

卷积的 “卷”，指的的函数的翻转，从 *g(t)* 变成 *g(-t)* 的这个过程；同时，“卷” 还有滑动的意味在里面（吸取了网友 李文清 的建议）。如果把卷积翻译为 “ 褶积 ”，那么这个 “褶” 字就只有翻转的含义了。

卷积的 “积”，指的是积分 / 加权求和。

有些文章只强调滑动叠加求和，而没有说函数的翻转，我觉得是不全面的；有的文章对 “卷” 的理解其实是“积”，我觉得是张冠李戴。

对卷积的意义的理解：

1. 从 “积” 的过程可以看到，我们得到的 叠加值 ，是个全局的概念。以信号

分析为例，卷积的结果是不仅跟当前时刻输入信号的响应值有关，也跟过去所有时刻输入信号的响应都有关系，考虑了对过去的所有输入的效果的累积。在图像处理的中，卷积处理的结果，其实就是把每个像素周边的，甚至是整个图像的像素都考虑进来，对当前像素进行某种加权处理。所以说，“积” 是全局概念，或者说是一种 “混合”，把两个函数在时间或者空间上进行混合。

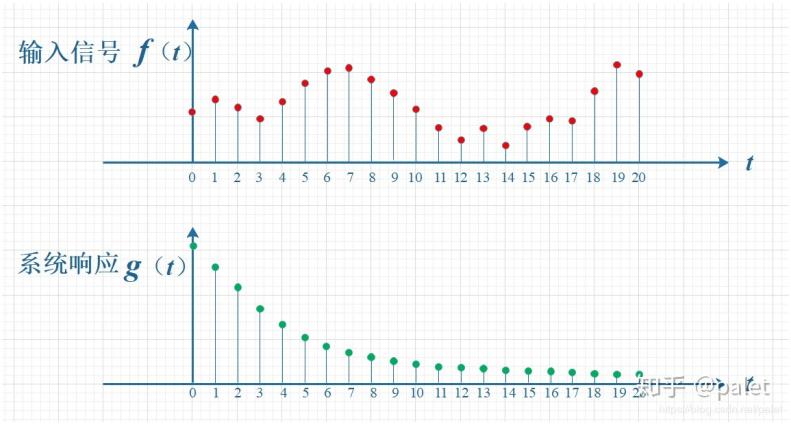
2. 那为什么要进行 “卷”？直接相乘不好吗？我的理解，进行“卷”（翻转）的目的其实是施加一种约束，它指定了在“积” 的时候以什么为参照。在信号分析的场景，它指定了在哪个特定时间点的前后进行“积”，在空间分析的场景，它指定了在哪个位置的周边进行累积处理。

举例说明

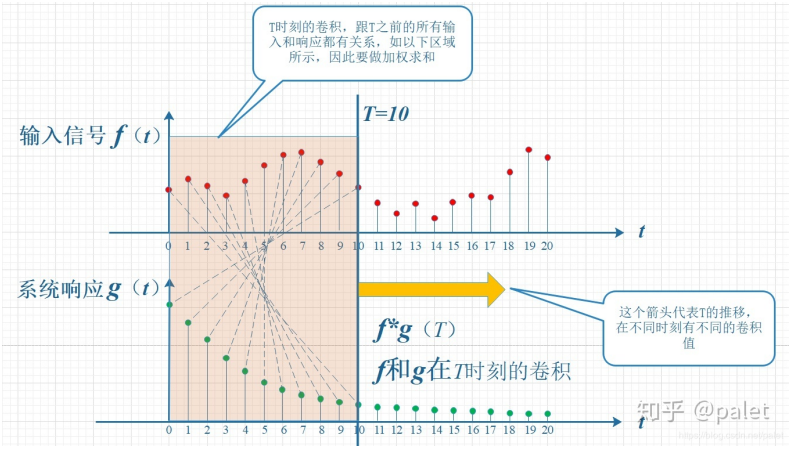
下面举几个例子说明为什么要翻转，以及叠加求和的意义。

例 1：信号分析

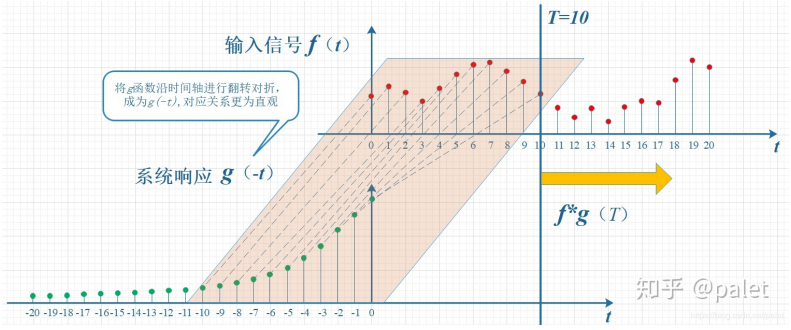
如下图所示，输入信号是 $f(t)$，是随时间变化的。系统响应函数是 $g(t)$，图中的响应函数是随时间指数下降的，它的物理意义是说：如果在 t=0 的时刻有一个输入，那么随着时间的流逝，这个输入将不断衰减。换言之，到了 t=T 时刻，原来在 t=0 时刻的输入 $f(0)$ 的值将衰减为 $f(0)g(T)$。



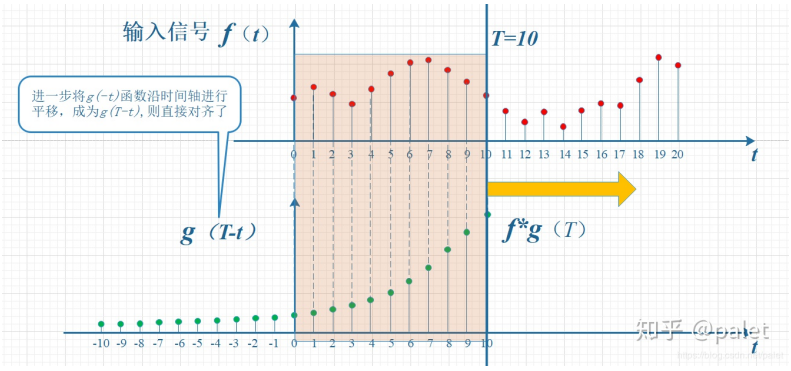
考虑到信号是连续输入的，也就是说，每个时刻都有新的信号进来，所以，最终输出的是所有之前输入信号的累积效果。如下图所示，在 T=10 时刻，输出结果跟图中带标记的区域整体有关。其中，f(10) 因为是刚输入的，所以其输出结果应该是 f(10)g(0)，而时刻 t=9 的输入 f(9)，只经过了 1 个时间单位的衰减，所以产生的输出应该是 f(9)g(1)，如此类推，即图中虚线所描述的关系。这些对应点相乘然后累加，就是 T=10 时刻的输出信号值，这个结果也是 f 和 g 两个函数在 T=10 时刻的卷积值。



显然，上面的对应关系看上去比较难看，是拧着的，所以，我们把 g 函数对折一下，变成了 g(-t)，这样就好看一些了。看到了吗？这就是为什么卷积要“卷”，要翻转的原因，这是从它的物理意义中给出的。



上图虽然没有拧着，已经顺过来了，但看上去还有点错位，所以再进一步平移 T 个单位，就是下图。它就是本文开始给出的卷积定义的一种图形的表述：



所以，在以上计算 T 时刻的卷积时，要维持的约束就是： $t+(T-t)=T$ 。这种约束的意义，大家可以自己体会。

例 2：丢骰子

在本问题 如何通俗易懂地解释卷积？ 中排名第一的 马同学 在中举了一个很好的例子（下面的一些图摘自马同学的文章，在此表示感谢），用丢骰子说明了卷积的应用。

要解决的问题是：有两枚骰子，把它们都抛出去，两枚骰子点数加起来为 4 的概率是多少?

分析一下，两枚骰子点数加起来为 4 的情况有三种情况：1+3=4， 2+2=4，3+1=4

因此，两枚骰子点数加起来为 4 的概率为：

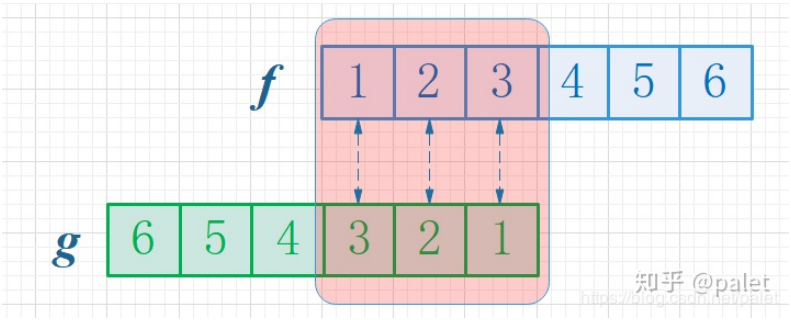
$$f(1)g(3) + f(2)g(2) + f(3)g(1)$$

写成卷积的方式就是：

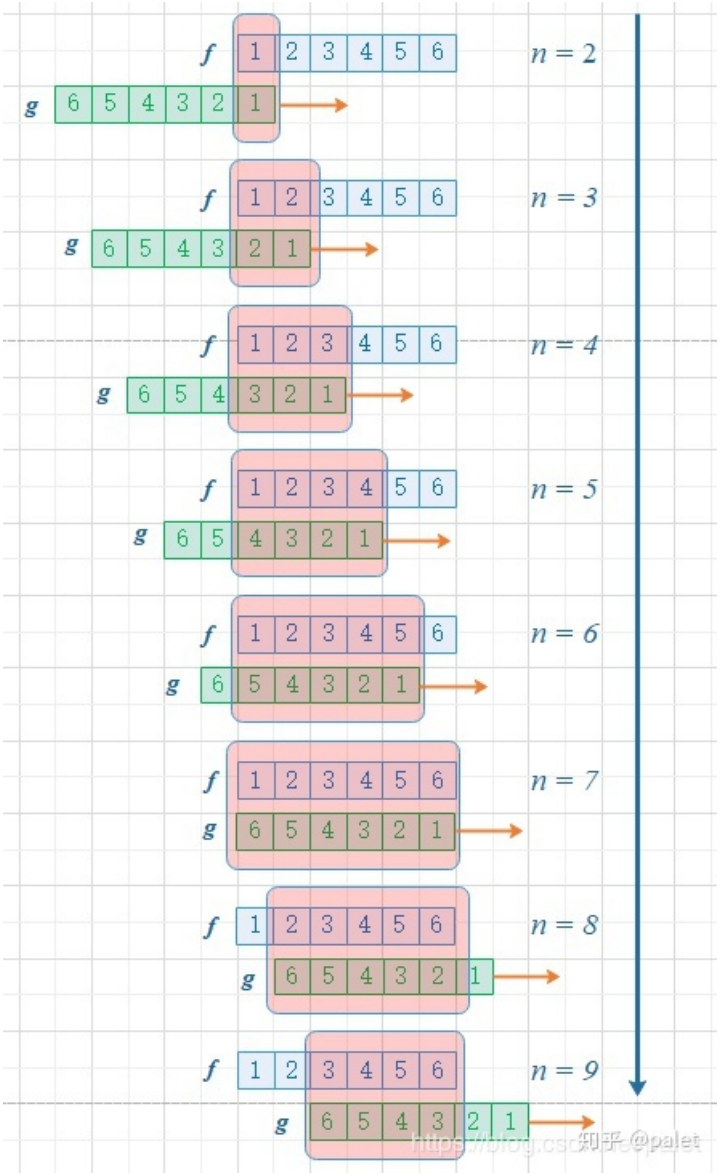
$$(f*g)(4) = \sum_{m=1}^{3} f(4-m)g(m)$$

在这里我想进一步用上面的翻转滑动叠加的逻辑进行解释。

首先，因为两个骰子的点数和是 4，为了满足这个约束条件，我们还是把函数 g 翻转一下，然后阴影区域上下对应的数相乘，然后累加，相当于求自变量为 4 的卷积值，如下图所示：



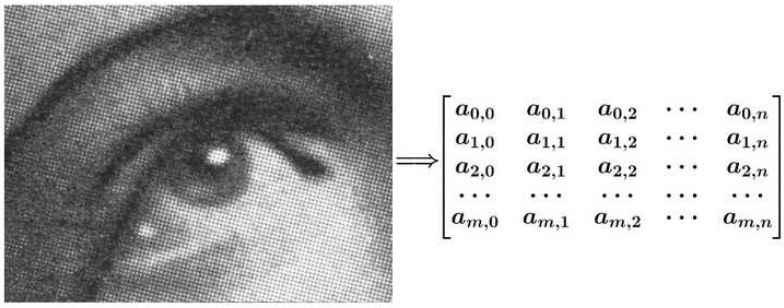
进一步，如此翻转以后，可以方便地进行推广去求两个骰子点数和为 *n* 时的概率，为*f* 和 *g*的卷积 *f*g(n)*，如下图所示：



由上图可以看到，函数 *g* 的滑动，带来的是点数和的增大。这个例子中对 f 和 g 的约束条件就是点数和，它也是卷积函数的自变量。有兴趣还可以算算，如果骰子的每个点数出现的概率是均等的，那么两个骰子的点数和 n=7 的时候，概率最大。

例 3：图像处理

还是引用知乎问题 如何通俗易懂地解释卷积？ 中 马同学 的例子。图像可以表示为矩阵形式（下图摘自马同学的文章）：



对图像的处理函数（如平滑，或者边缘提取），也可以用一个 g 矩阵 来表示，如：

$$g = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix}$$

注意，我们在处理平面空间的问题，已经是二维函数了，相当于：

$$f(x,y) = a_{x,y}$$

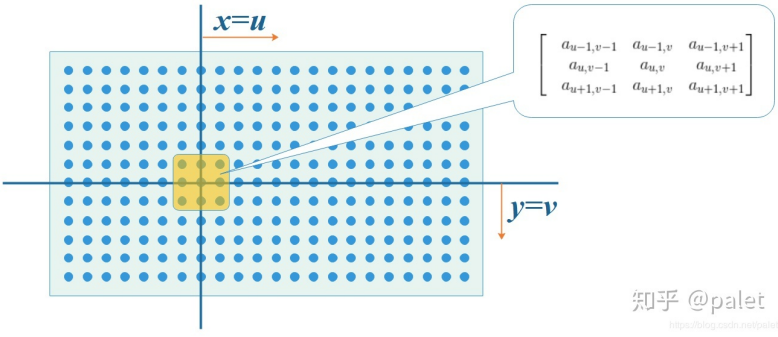
$$g(x,y) = b_{x,y}$$

那么函数 f 和 g 的在（u，v）处的卷积 $f * g(u,v)$ 该如何计算呢?

按卷积的定义，二维离散形式的卷积公式应该是：

$$(f*g)(u,v) = \sum_i \sum_j f(i,j)g(u-i,v-j) = \sum_i \sum_j a_{i,j} b_{u-i,v-j}$$

从卷积定义来看，应该是在 x 和 y 两个方向去累加（对应上面离散公式中的 i 和 j 两个下标），而且是无界的，从 负无穷 到正无穷。可是，真实世界都是有界的。例如，上面列举的图像处理函数 g 实际上是个 3x3 的矩阵，意味着，在除了原点附近以外，其它所有点的取值都为 0。考虑到这个因素，上面的公式其实退化了，它只把坐标（u,v）附近的点选择出来做计算了。所以，真正的计算如下所示：

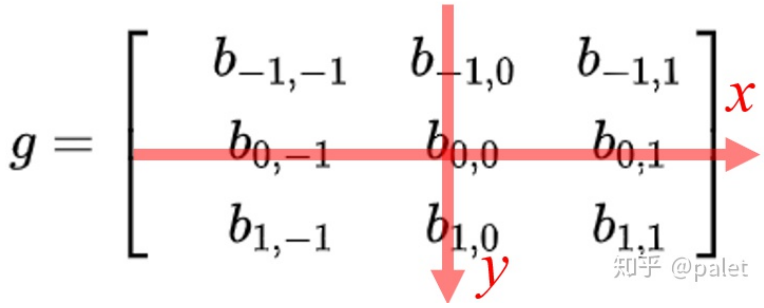


首先我们在原始图像矩阵中取出（u,v）处的矩阵：

$$f=\begin{bmatrix} a_{u-1,v-1} & a_{u-1,v} & a_{u-1,v+1} \\ a_{u,v-1} & a_{u,v} & a_{u,v+1} \\ a_{u+1,v-1} & a_{u+1,v} & a_{u+1,v+1} \end{bmatrix}$$

然后将图像处理矩阵翻转（这个翻转有点意思，可以有几种不同的理解，其效果是等效的：（1）先沿 x 轴翻转，再沿 y 轴翻转；（2）先沿 x 轴翻转，再沿 y 轴翻转；），如下：

原始矩阵：



翻转后的矩阵：$g'=\begin{bmatrix} b_{1,1} & b_{1,0} & b_{1,-1} \\ b_{0,1} & b_{0,0} & b_{0,-1} \\ b_{-1,1} & b_{-1,0} & b_{-1,-1} \end{bmatrix}$

（1）先沿 x 轴翻转，再沿 y 轴翻转

$$g=\begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} => \begin{bmatrix} b_{1,-1} & b_{1,0} & b_{1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{-1,-1} & b_{-1,0} & b_{-1,1} \end{bmatrix} => \begin{bmatrix} b_{1,1} & b_{1,0} & b_{1,-1} \\ b_{0,1} & b_{0,0} & b_{0,-1} \\ b_{-1,1} & b_{-1,0} & b_{-1,-1} \end{bmatrix} = g'$$

（2）先沿 y 轴翻转，再沿 x 轴翻转

$$g=\begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} => \begin{bmatrix} b_{-1,1} & b_{-1,0} & b_{-1,-1} \\ b_{0,1} & b_{0,0} & b_{0,-1} \\ b_{1,1} & b_{1,0} & b_{1,-1} \end{bmatrix} => \begin{bmatrix} b_{1,1} & b_{1,0} & b_{1,-1} \\ b_{0,1} & b_{0,0} & b_{0,-1} \\ b_{-1,1} & b_{-1,0} & b_{-1,-1} \end{bmatrix} = g'$$

计算卷积时，就可以用 f 和 g' 的内积：

$$f*g(u,v)=a_{u-1,v-1}\times b_{1,1}+a_{u-1,v}\times b_{1,0}+a_{u-1,v+1}\times b_{1,-1}$$

$$+a_{u,v-1}\times b_{0,1}+a_{u,v}\times b_{0,0}+a_{u,v+1}\times b_{0,-1}$$

$$+a_{u+1,v-1}\times b_{-1,1}+a_{u+1,v}\times b_{-1,0}+a_{u+1,v+1}\times b_{-1,-1}$$

请注意，以上公式有一个特点，做乘法的两个对应变量 a,b 的下标之和都是（u,v），其目的是对这种加权求和进行一种约束。这也是为什么要将矩阵 g 进行翻转的原因。以上矩阵下标之所以那么写，并且进行了翻转，是为了让大家更清楚地看到跟卷积的关系。这样做的好处是便于推广，也便于理解其物理意义。实际在计算的时候，都是用翻转以后的矩阵，直接求矩阵内积就可以了。

以上计算的是（u,v）处的卷积，延 x 轴或者 y 轴滑动，就可以求出图像中各个位置的卷积，其输出结果是处理以后的图像（即经过平滑、边缘提取等各种处理的图像）。

再深入思考一下，在算图像卷积的时候，我们是直接在原始图像矩阵中取了（u,v）处的矩阵，为什么要取这个位置的矩阵，本质上其实是为了满足以上的约束。因为我们要算（u，v）处的卷积，而 g 矩阵是 3x3 的矩阵，要满足下标跟这个 3x3 矩阵的和是（u,v），只能是取原始图像中以（u，v）为中心的这个 3x3 矩阵，即图中的阴影区域的矩阵。

推而广之，如果如果 g 矩阵不是 3x3，而是 7x7，那我们就要在原始图像中取以（u，v）为中心的 7x7 矩阵进行计算。由此可见，这种卷积就是把原始图像中的相邻像素都考虑进来，进行混合。相邻的区域范围取决于 g 矩阵的维度，维度越大，涉及的周边像素越多。而矩阵的设计，则决定了这种混合输出的图像跟原始图像比，究竟是模糊了，还是更锐利了。

比如说，如下图像处理矩阵将使得图像变得更为平滑，显得更模糊，因为它联合周边像素进行了平均处理：

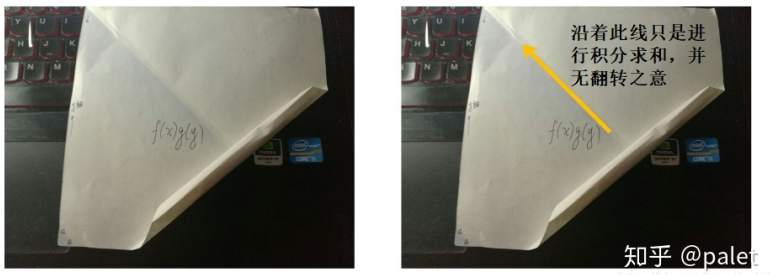
$$g = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

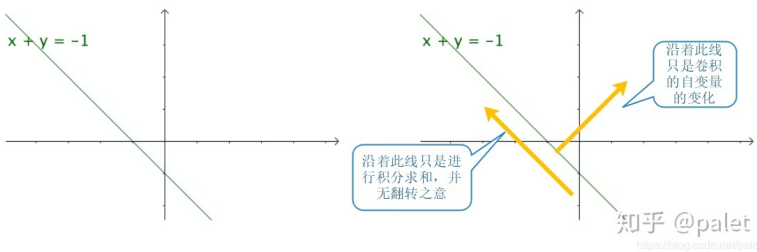
而如下图像处理矩阵将使得像素值变化明显的地方更为明显，强化边缘，而变化平缓的地方没有影响，达到提取边缘的目的：

$$g = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

对一些解释的不同意见

上面一些对卷积的形象解释，如知乎问题 卷积为什么叫「卷」积？ 中 荆哲，以及问题 如何通俗易懂地解释卷积？ 中 马同学 等人提出的如下比喻：





其实图中 “卷” 的方向，是沿该方向进行积分求和的方向，并无翻转之意。因此，这种解释，并没有完整描述卷积的含义，对 “卷” 的理解值得商榷。

一些参考资料

《数字信号处理（第二版）》 程乾生，北京大学出版社

《信号与系统引论》 郑君里，应启珩，杨为理，高等教育出版社

马同学

从数学上讲，卷积就是一种运算。

某种运算，能被定义出来，至少有以下特征：

- 首先是抽象的、符号化的
- 其次，在生活、科研中，有着广泛的作用

比如加法：

- $a+b$，是抽象的，本身只是一个数学符号
- 在现实中，有非常多的意义，比如增加、合成、旋转等等

卷积，是我们学习高等数学之后，新接触的一种运算，因为涉及到积分、级数，所以看起来觉得很复杂。

1 卷积的定义

我们称 $(f*g)(n)$ 为 f,g 的卷积

其连续的定义为：

$$(f*g)(n)=\int_{-\infty}^{\infty}f(\tau)g(n-\tau)d\tau$$

其离散的定义为：

$$(f*g)(n)=\sum_{\tau=-\infty}^{\infty}f(\tau)g(n-\tau)$$

这两个式子有一个共同的特征：

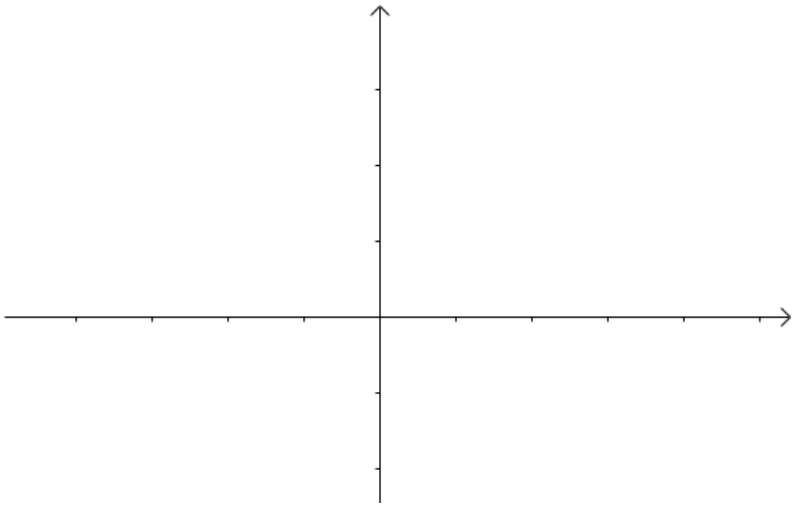
$$(f*g)(n)=\int_{-\infty}^{\infty}f(\tau)g(n-\tau)d\tau$$

$$n=\tau+(n-\tau)$$

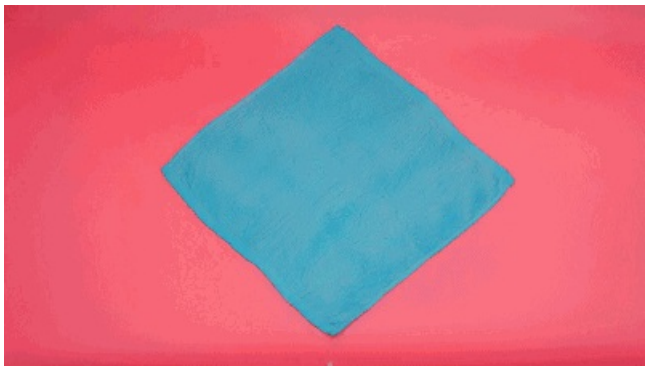
$$(f*g)(n)=\sum_{\tau=-\infty}^{\infty}f(\tau)g(n-\tau)$$

这个特征有什么意义?

我们令 $x=\tau, y=n-\tau$ ，那么 $x+y=n$ 就是下面这些直线：



如果遍历这些直线，就好比，把毛巾沿着角卷起来：



此处受到 荆哲：卷积为什么叫「卷」积? 答案的启发。

只看数学符号，卷积是抽象的，不好理解的，但是，我们可以通过现实中的意义，来**习惯**卷积这种运算，正如我们小学的时候，学习 加减乘除 需要各种苹果、糖果来帮助我们习惯一样。

我们来看看现实中，这样的定义有什么意义。

2 离散卷积 的例子：丢骰子

我有两枚骰子：

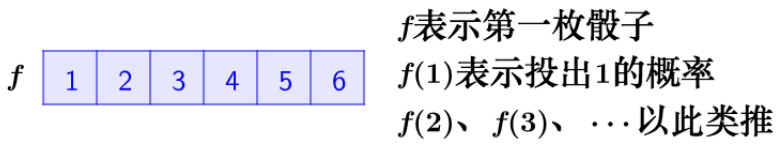
把这两枚骰子都抛出去：

求：

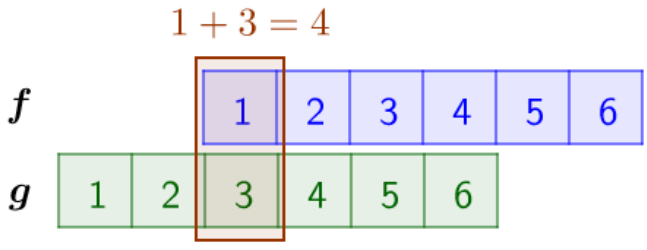
两枚骰子点数加起来为4的概率是多少？

这里问题的关键是，两个骰子加起来要等于 4，这正是卷积的应用场景。

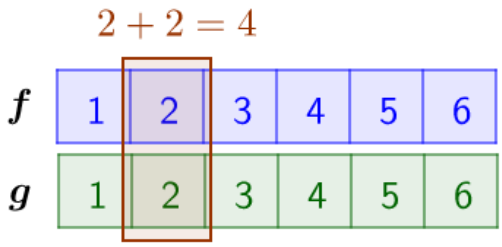
我们把骰子各个点数出现的概率表示出来：



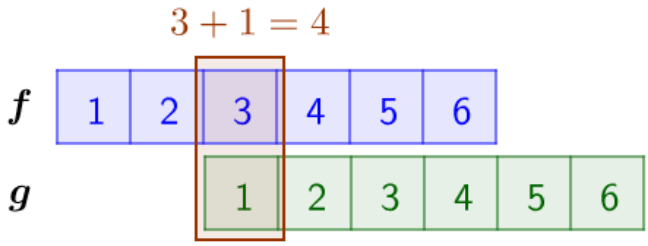
那么，两枚骰子点数加起来为 4 的情况有：



出现概率为： $f(1)g(3)$



出现概率为：$f(2)g(2)$



出现概率为：$f(3)g(1)$

因此，两枚骰子点数加起来为 4 的概率为：

$$f(1)g(3) + f(2)g(2) + f(3)g(1)$$

符合卷积的定义，把它写成标准的形式就是：

$$(f * g)(4) = \sum_{m=1}^{3} f(4-m)g(m)$$

3 连续卷积的例子：做馒头

楼下早点铺子生意太好了，供不应求，就买了一台机器，不断的生产馒头。

假设馒头的生产速度是 $f(t)$，那么一天后生产出来的馒头总量为：

$$\int_0^{24} f(t)dt$$

馒头生产出来之后，就会慢慢腐败，假设腐败函数为 $g(t)$，比如，10 个馒头，24 小时会腐败：

$$10 * g(t)$$

想想就知道，第一个小时生产出来的馒头，一天后会经历 24 小时的腐败，第二个小时生产出来的馒头，一天后会经历 23 小时的腐败。

如此，我们可以知道，一天后，馒头总共腐败了：

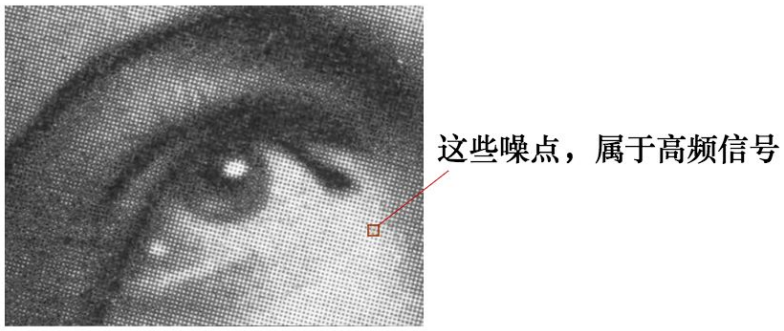
$$\int_0^{24} f(t)g(24-t)dt$$

这就是连续的卷积。

4 图像处理

4.1 原理

有这么一副图像，可以看到，图像上有很多噪点：



高频信号，就好像平地耸立的山峰：

看起来很显眼。

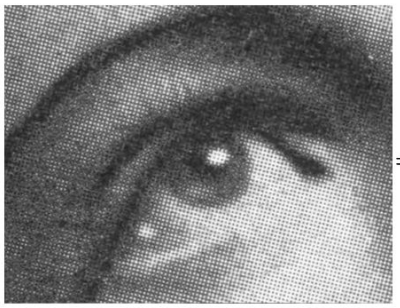
平滑这座山峰的办法之一就是，把山峰刨掉一些土，填到山峰周围去。用数学的话来说，就是把山峰周围的高度平均一下。

平滑后得到：

4.2 计算

卷积可以帮助实现这个平滑算法。

有噪点的原图，可以把它转为一个矩阵：



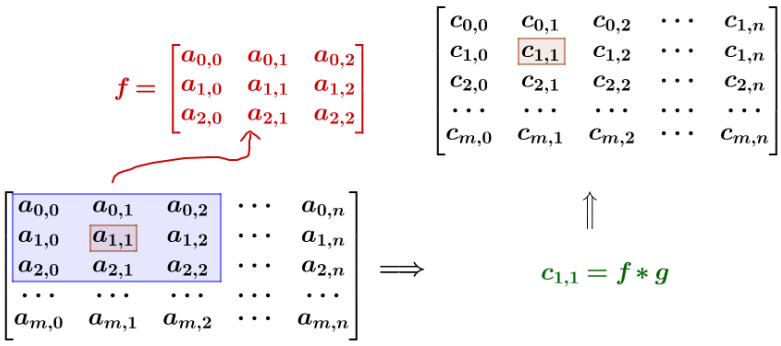
$$\Longrightarrow \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots & a_{0,n} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m,0} & a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

然后用下面这个平均矩阵（说明下，原图的处理实际上用的是正态分布矩阵，这里为了简单，就用了算术平均矩阵）来平滑图像：

$$g = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

记得刚才说过的算法，把高频信号与周围的数值平均一下就可以平滑山峰。

比如我要平滑 $a_{1,1}$ 点，就在矩阵中，取出 $a_{1,1}$ 点附近的点组成矩阵 f ，和 g 进行卷积计算后，再填回去：



要注意一点，为了运用卷积，g 虽然和 f 同维度，但下标有点不一样：

注意两者的下标不一样

$$f = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \qquad g = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix}$$

我用一个动图来说明下计算过程：

a, b的下标相加都为$1, 1$

$$f = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \qquad g = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix}$$

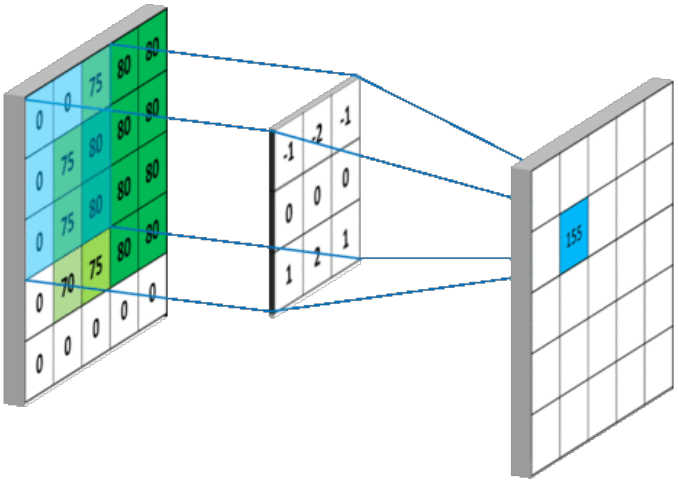
$$c_{1,1} = a_{0,0}b_{1,1}$$

写成卷积公式就是：

$$(f * g)(1,1) = \sum_{k=0}^{2}\sum_{h=0}^{2} f(h,k)g(1-h,1-k)$$

要求 $c_{4,5}$，一样可以套用上面的卷积公式。

这样相当于实现了 g 这个矩阵在原来图像上的划动（准确来说，下面这幅图把 g 矩阵旋转了 $180°$）：



此图出处： Convolutional Neural Networks - Basics

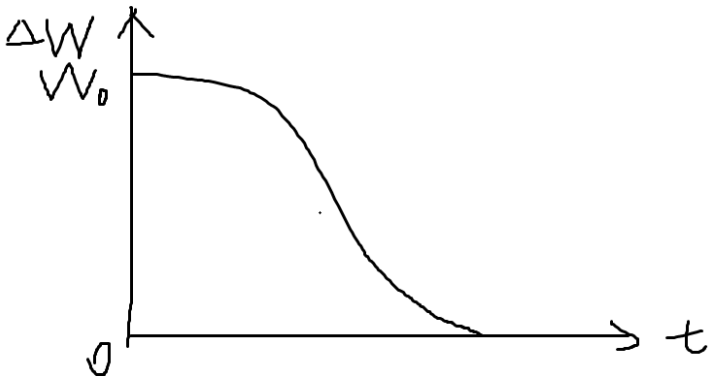
文章最新版本在（有可能会有后续更新）： 如何通俗地理解卷积?

1335

我来举个通俗易懂的例子吧。我大一是这么理解记忆的，到现在大四一直没忘记过。

要理解卷积，就必须树立起来 “**瞬时行为的持续性后果**” 这个概念。

举个例子。在一个时刻点，我以迅雷不及掩耳之势吃下了一个冰激凌，此时我的体重瞬间增加，之后随着消化吸收能量利用和排泄等生理活动的进行，我的体重又缓慢下降。如下图所示：

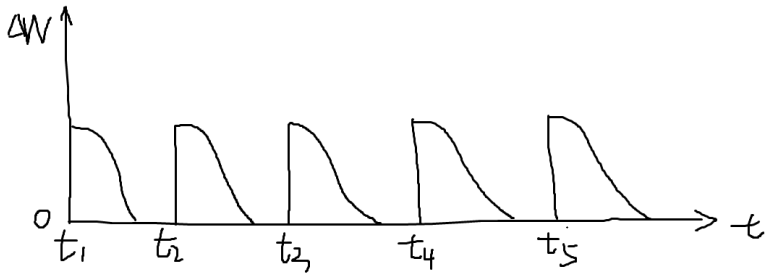


我们把这个函数记为 $f(t)$ 。我们把基础体重记为 0，即没吃冰淇凌的时候体重是 0，吃冰淇凌的效果过去了之后体重还是 0。我们记每一个冰淇凌带来的瞬间体重增加为 W_0 。易知， $f(0)=W_0, f(+\infty)=0$ 。

如何理解 “瞬时行为的持续性后果” 呢？在这个例子里，**吃冰激凌是瞬间完成的动作，是一个瞬时行为；吃完冰激凌之后的体重的缓慢下降是持续了一段时间的，因此是吃冰激凌这个瞬时行为的一个持续性后果。**

此时，只有在 0 时刻的瞬间吃了一个冰淇凌，在 0 时刻的瞬间，吃冰淇凌的速度是 $\frac{1}{\delta t}=+\infty$ ，其中 δt 表示极小的一个时间段；在其他时刻，吃冰淇凌的速度为 0。因此，我们可以
用一个冲击函数 $\delta(t)$ 来表示在这种情况下吃冰淇凌的速度。
$f(t)$表示的是，当吃冰淇凌的速度为冲击函数 $\delta(t)$的时候，对我的体重的影响。

接下来我们考虑，我吃冰淇凌的频率很低，且每次只在一个瞬间吃一个冰淇凌，每次都等到体重恢复到原来的程度了再吃一个，那么我的体重变化就是这样子的。



这种情况下，如果我想要知道每一个时刻的体重，只需要知道我吃每个冰淇凌的时刻 $t_1, t_2, \ldots t_5 \cdots$ ，再知道吃一个冰淇凌的效果 $f(t)$ ，很容易就能求出来了。

接下来，我们考虑，**如果我吃冰淇凌的速度恒定为 1（注意不是一瞬间吃一个了，不是冲击函数），且时时刻刻都在吃冰淇凌，那么，在我连续吃了T时间的冰淇凌之后，我的体重是多少呢?**

这个问题是不是有点不好算了呢？之前的冰淇凌增加的体重还没降到 0 呢，现在的冰淇凌带来的体重就又来了，还一直持续，还是连续的，想想就头疼。

这个时候，要引入两个个原理 。

第一，**线性原理。即，我在一瞬间吃冰淇凌的个数，会以线性的方式作用在冰淇凌对体重的影响函数 $f(t)$ 上。**我在一个瞬间吃了 1 个冰淇凌，之后我的体重变换是 $f(t)$，如果我在一个瞬间吃了 0.5 个冰淇凌，之后我的体重变换是 $0.5f(t)$，如果 n 个呢，那就是 $nf(t)$ 。

第二，**累加原理。即，冰淇凌的作用效果是可以累加的。**即，一段时间之前我吃了一个冰淇凌，经过了一段时间的体重下降，现在我的体重是 W_1 。现在我又吃了一个冰淇凌，体重又增加了。假设这个增加是可以累积的（直观上也是可以累积的），那么我的体重就会是 $W_1 + f(0) = W_1 + W_0$ 。这就是累加原理。

这时我们来试着计算，在从开始就不停地吃冰淇凌，且吃冰淇凌的速度恒定为 1 的情况下，在任意时刻 T 我的体重。

由于我在不停地吃冰淇凌，所以，我们先算，在某时刻 $\tau(\tau < T)$ 附近的一瞬间$d\tau$ ，我吃的冰淇凌对现在时刻 T 的我的体重的影响。因为，吃冰淇凌的速度是 1，时间是$d\tau$ ，因此，在$d\tau$这一瞬间我吃的冰淇凌的个数是 $1 * d\tau = d\tau$ 。那么根据线性原理，在$d\tau$这一瞬间，我吃的冰淇凌对现在时刻 T 的我的体重的影响就是 $f(T-\tau)d\tau$ 。

那么，根据累加原理，现在时刻 T 的我的体重就是：从 0 到 T时刻我吃的所有冰淇凌对我的体重的影响的累加，即为：

$$W(T) = \int_0^T f(T-\tau)d\tau$$

上面这个式子是不是有点像我们学过的卷积了呢?

我们上面的讨论基于我们吃冰淇凌的速度是常数 1，那么，**如果我吃冰淇凌的速度不是常数，而是一个连续变化的函数，如在 t 时刻，吃冰淇凌的速度是 $g(t)$ 。那么，在我连续吃了T时间的冰淇凌之后，我的体重是多少呢?**

同样，我们先算，在某时刻 $\tau(\tau < T)$ 附近的一瞬间$d\tau$ ，我吃的冰淇凌对现在时刻 T 的我的体重的影响。因为，吃冰淇凌的速度是 $g(\tau)$ ，时间是$d\tau$ ，因此，在$d\tau$这一瞬间吃的冰淇凌的个数是 $g(\tau) * d\tau = g(\tau)d\tau$ 。那么根据线性原理，在$d\tau$这一瞬间，我吃的冰淇凌对现在时刻 T 的我的体重的影响就是 $g(\tau)f(T-\tau)d\tau$ 。

再根据累加原理，现在时刻 T 的我的体重就是：从 0 到 T时刻我吃的所有冰淇凌对我的体重的影响的累加，即为：

$$W(T) = \int_0^T g(\tau)f(T-\tau)d\tau$$

这就是大家平时接触到的卷积了！

因此，在我的理解下，我将卷积解释为：

一个对象（本文中的吃冰淇凌）对一个系统（本文中的体重）的作用效果满足**线性原理、累加原理**。该对象对这个系统连续作用了一段时间后，求该系统的状态。这个时候，一个卷积就可以求出来了！

在卷积 $W(T)=\int_0^T g(\tau)f(T-\tau)d\tau$ 中，

第一个函数 $g(t)$ 表示这个对象对系统的作用速度。第二个函数 $f(t)$ 表示当作用速度为单位冲击函数时这个对象对系统的作用效果。

我们来验证一下第二个函数 $f(t)$ 的意义。取我吃冰淇凌的速度为单位冲击函数 $g(t)=\delta(t)$ ，则到时刻 T 我的体重就是：

$W(T)=\int_0^T \delta(\tau)f(T-\tau)d\tau=f(T)$ ，的的确确就是我吃冰淇凌的速度为单位冲击函数时，我的体重的变换。

最后，是一点说明。

课本上标准的卷积其实长成下面这个样子，积分区间是 $(-\infty,+\infty)$ 。

$$W(T)=\int_{-\infty}^{+\infty} g(\tau)f(T-\tau)d\tau$$

这个在我这个 case 里也比较好理解，主要是考虑到时间的物理意义。

第一，理解当 $t<0$ 时， $f(t)=0$ 恒成立。这个比较容易理解，因为，我在 $t=0$ 时刻吃的冰淇凌，对吃冰淇凌之前也就是 $t<0$ 时刻的我的体重是没有影响的。所以，当 $\tau>T$ 的时候， $T-\tau<0$ ， $f(T-\tau)=0$ 。

第二，理解当 $t<0$ 时，  恒成立。这个更好理解，就是时间非负性。我是从 $t=0$ 时刻开始吃冰淇凌的， $g(t)$ 表示我在 t 时刻吃冰淇凌的速度。$t<0$ 的时候，我还没吃冰淇凌呢 ，自然不存在吃冰淇凌的速度这个概念。

所以， $W(T)=\int_{-\infty}^{+\infty} g(\tau)f(T-\tau)d\tau=\int_0^T g(\tau)f(T-\tau)d\tau$

在其他的 case 里，情况就不一样了。

1、某一个对象的作用域可能不是 时间域，不必遵循时间上的因果律。因此，当 $t<0$ 时， $f(t)\neq 0$ 。

2、某一个对象的作用域可能不是时间域，作用域存在负数的可能性。因此，当 $t<0$ 时， $g(t)\neq 0$ 。

基于以上两点考虑，积分区间就是 $(-\infty,+\infty)$ ，也就是课本上标准的卷积形式了！

果程 C

应题主邀，把我在

卷积的物理意义是什么？ - 数学

的答案搬过来。

——————————————————————————————————————

对于初学者，我推荐用**复利**的例子来理解卷积可能更好理解一些：

小明存入 100 元钱， 年利率α 是 5%，按复利计算（即将每一年所获利息加入本金，以计算下一年的利息），那么在五年之后他能拿到的钱数是 $100(1+5\%)^5$，如下表所示：

本金	第一年	第二年	第三年	第四年	第五年
100	$100\times(1.05)^1$	$100\times(1.05)^2$	$100\times(1.05)^3$	$100\times(1.05)^4$	$100\times(1.05)^5$

将这笔钱存入银行的一年之后，小明又往银行中存入了 100 元钱，年利率仍为 5%，那么这笔钱按复利计算，到了第五年，将收回的钱数是 $100(1+5\%)^4$，我们将这一结果作为新的一行加入上面的表格中：

本金	第一年	第二年	第三年	第四年	第五年
+100	$100\times(1.05)^1$	$100\times(1.05)^2$	$100\times(1.05)^3$	$100\times(1.05)^4$	$100\times(1.05)^5$
	+100	$100\times(1.05)^1$	$100\times(1.05)^2$	$100\times(1.05)^3$	$100\times(1.05)^4$

以此类推，如果小明每年都往银行中存入新的 100 元钱，那么这个收益表格将是这样的：

本金	第一年	第二年	第三年	第四年	第五年
+100	$100\times(1.05)^1$	$100\times(1.05)^2$	$100\times(1.05)^3$	$100\times(1.05)^4$	$100\times(1.05)^5$
	+100	$100\times(1.05)^1$	$100\times(1.05)^2$	$100\times(1.05)^3$	$100\times(1.05)^4$
		+100	$100\times(1.05)^1$	$100\times(1.05)^2$	$100\times(1.05)^3$
			+100	$100\times(1.05)^1$	$100\times(1.05)^2$
				+100	$100\times(1.05)^1$
					+100

可见，最终小明拿到的钱将等于他各年存入的钱分别计算复利之后得到的钱数的总和，即：

复利计算（5年） 复利计算（4年） 复利计算（3年） 复利计算（2年） 复利计算（1年） 复利计算（0年）

$$100\times(1.05)^5+100\times(1.05)^4+100\times(1.05)^3+100\times(1.05)^2+100\times(1.05)^1+100\times(1.05)^0$$

第0年存入的钱 第1年存入的钱 第2年存入的钱 第3年存入的钱 第4年存入的钱 第5年存入的钱

用求和符号来简化这个公式，可以得到：

$$\sum_{i=0}^{5} f(i)g(5-i), \text{where } f(i)=100, g(5-i)=(1.05)^{5-i}$$

在上式中，$f(i)$为小明的**存钱函数**，而$g(i)$为存入银行的每一笔钱的**复利计算函数**。**在这里，小明最终得到的钱就是他的存钱函数和复利计算函数的卷积。**

为了更清晰地看到这一点，我们将这个公式推广到连续的情况，也就是说，小明在从0到t的这一段时间内，每时每刻都往银行里存钱，他的存钱函数为$f(\tau)\ (0 \le \tau \le t)$，而银行也对他存入的每一笔钱按复利公式计算收益：$g(t-\tau)=(1+5\%)^{t-\tau}$，则小明到时间$t$将得到的总 钱数o 为：

$$\int_0^t f(\tau)g(t-\tau)d\tau = \int_0^t f(\tau)(1+5\%)^{t-\tau}d\tau$$

这也就是卷积的表达式了，上式可以记为$(f*g)(t)$。

相信通过上面这个例子，大家应该能够很清晰地记住卷积公式了。下面我们再展开说两句：

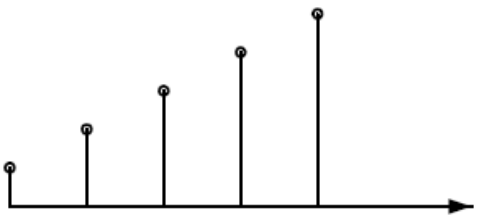
如果我们将小明的存款函数视为一个**信号发生（也就是激励）**的过程，而将复利函数$g(t-\tau)$视为一个**系统对信号的响应函数（也就是响应）**，那么二者的卷积$(f*g)(t)$就可以看做是在t时刻对系统进行观察，**得到的观察结果（也就是输出）**将是过去产生的所有信号经过系统的「处理 / 响应」后得到的结果的**叠加，**这也就是**卷积的物理意义**了。

THE SCIENTIST FACE

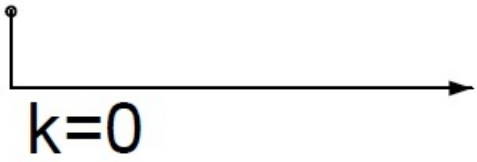
周赛克

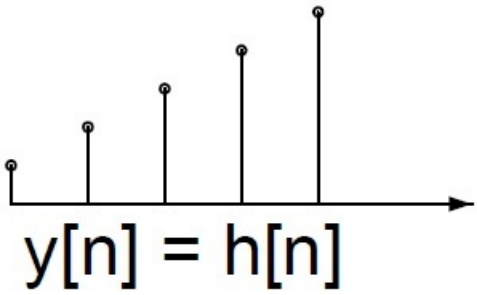
《卷积信号与 moba 游戏的伤害计算》

在打游戏的时候，假设你对敌方英雄造成了一次攻击。攻击的效果是，地方英雄会在接下来 5 秒内持续增长的掉血。每秒掉血分别为 1 2 3 4 5；那么你在 0 秒的时候攻击了一下敌方英雄，敌方英雄的掉血情况为：



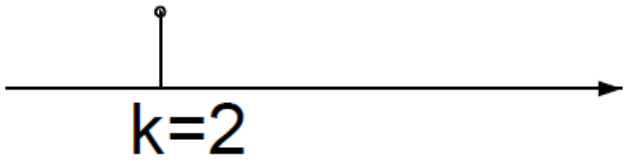
抽象一下。假设 x[k] 是一个单位信号（一次攻击），即仅当 k=0 时 x(k) = 1. 假设把这个信号输入到一个系统里面得到的响应为 h[n]：

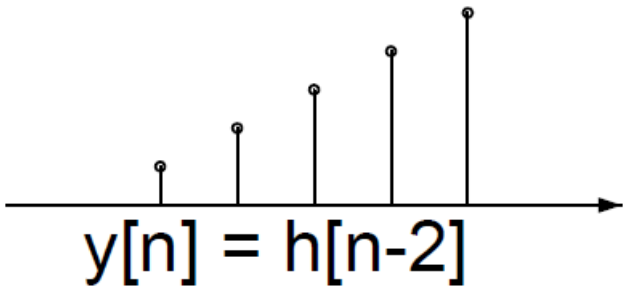




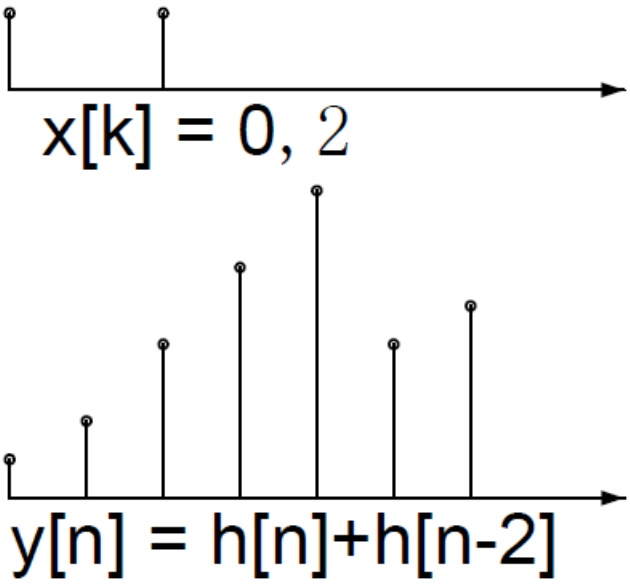
同样，假设你在第 2 秒对地方英雄进行了攻击，则敌方英雄会在地 2-7 秒持续掉血。

由于攻击不是在第 0 秒，而是在第二秒，因此，产生伤害也是从第二秒开始。使用函数位移的想法了解释，则本来第 0 秒产生的伤害右移了两个单位，变成了 h[n-2]：





那么，如果第 0 秒和第 2 秒都进行了攻击，伤害是怎样的呢？当然是第 0 秒的伤害加上第 2 秒的伤害，即：



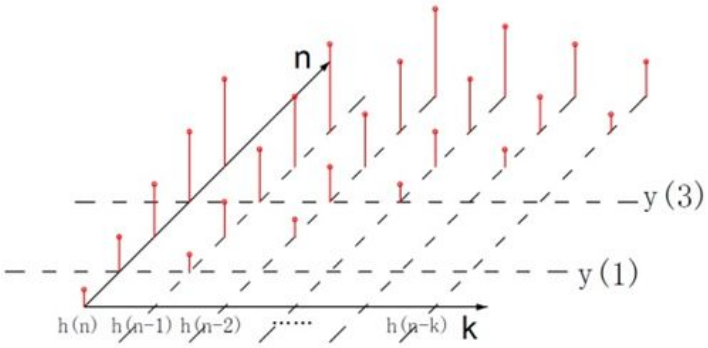
那么，对于任意形式的攻击 x[k]，很容易推广到，产生的伤害 y[n] 计算方法为：

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

这就是卷积的公式。解释很简单，就是对任意的攻击 x[k]，产生的伤害就是，把所有 k 时刻的攻击产生的伤害 h[n-k] 都叠加起来了。

那么，如果我想知道，对与任意的攻击方式，敌方英雄某一秒（比如第 3 秒）受到的伤害是怎样的呢?

即，对于输入 x[k]，任意一个 y[n] 点的伤害是怎样的呢?



我把不同 k 时候的关于 n 的函数 h[n-k] 全部画出来。如上图中不同的纵轴表示。那第三秒敌方英雄收到的伤害 y[3] 等于 x(0) h(3)+ x(1) h(3-1)+ x(2) h(3-2)+ x(3) h(3-3).

注意！！！ 在上图的表示中，我并未直接计算 ， 而是将其表示为不同的函数 h[n-k]与不同的 [k] 的组合！！

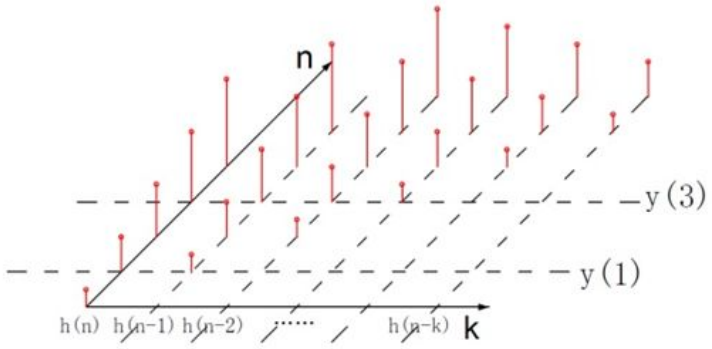
---------- 学术的分割线 ----------

也就是说，以离散信号为例，

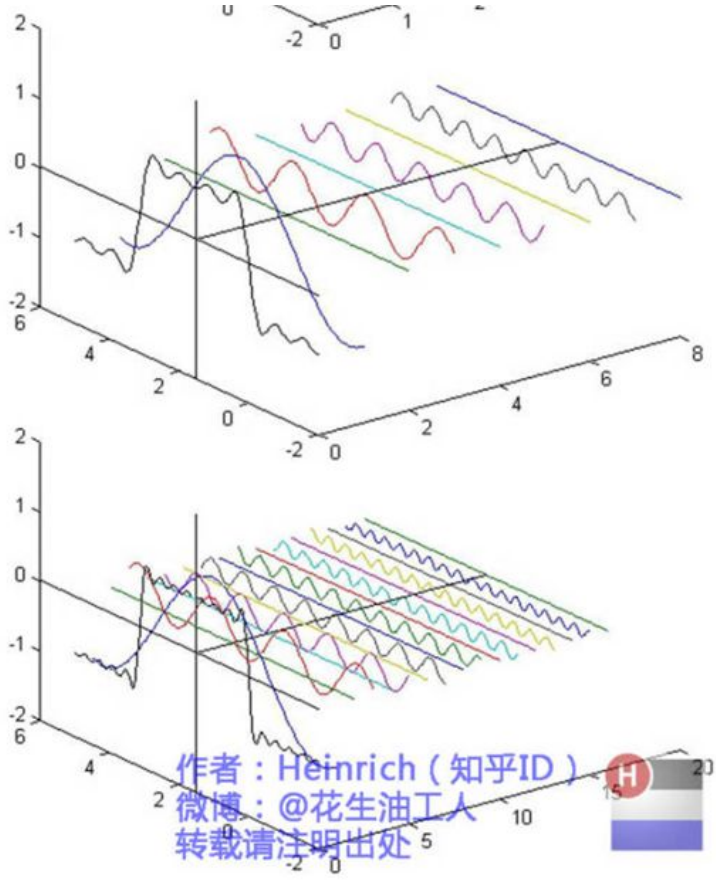
卷积 y[n] = x[n] * h[n]，可以理解为，将 y[n] 分为不同的 分量为 h[n-k]、系数为 x[k] 的**线性组合**。也就是说，**y[n] 是一系列关于 h[n] 移位后的 h[n-k]，每个函数 h[n-k] 乘上特定的系数 x[k] 组合而成的函数**。

这种 “用一组函数进行线性组合得到目标函数” 的理念，在傅里叶分析中也有应用：将原始信号 y[n], 分为一系列谐波信号 e^（jkwn）的函数的线性组合。

再来看这个图：



你想到了什么？？



上图解释参见： 如果看了这篇文章你还不懂傅里叶变换，那就过来掐死我吧

到这里会发现，卷积和傅里叶分析的一个相同的理念：

用不同的函数的线性组合，表示原信号。

傅里叶分析用的是不同的 正弦函数 的线性组合；

而卷积，用的是 h[n] 移位不同的 k 得到的一系列 h[n-k] 的线性组合。

换个角度，你会看到更多。