第 8 次 OJ 提示

林越山

November 2022

1 线性卷积

假设有长度为 N 的序列 $x[n], n = 0, 1, ..., N-1$ 与长度为 M 的序列 $h[n], n = 0, 1, ..., M-1$，二者进行线性卷积得到长度为 $N+M-1$ 的新序列，这里记为 $y[n] = h[n] \star x[n]$。

我们以样例为例，序列 x 为 $[1, 0, 4, 2, 3, 1, 5]$，序列 h 为 $[3, 1, 4]$，给出线性卷积的具体过程如下。

第 0 步：

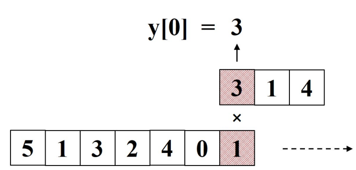


图 1: 第 0 步

第 1 步：

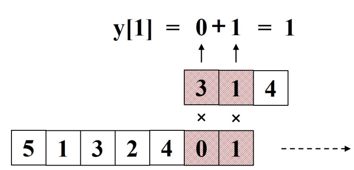


图 2: 第 1 步

类似的经过第 2-7 步，到第 8 步：

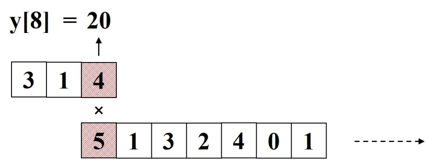


图 3: 第 8 步

由此我们得到序列 y，为 $[3,1,16,10,27,14,28,9,20]$。

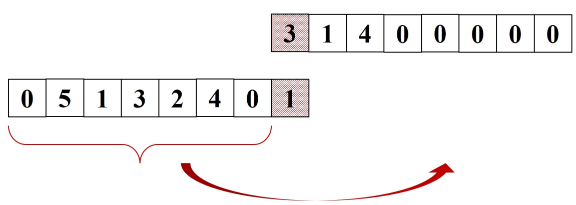
综上，线性卷积的输入是长度为$\boldsymbol{N}$ 与$\boldsymbol{M}$ 的两个序列，经过上述反转平移、乘积求和过程，输出长度为$\boldsymbol{N+M-1}$ 的序列。

2 循环卷积

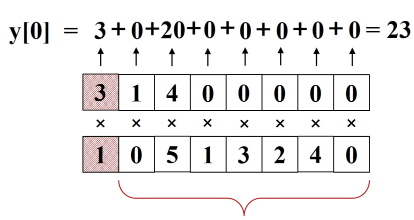
假设有长度为 N_1 的序列 $x_1[n], n=0,1,...,N_1-1$ 与相同长度的序列 $h_1[n], n=0,1,...,N_1-1$，二者进行循环卷积得到长度为 N_1 的新序列，这里记为 $y_1[n]=h_1[n]\otimes x_1[n]$。

我们同样给出一个例子，序列 x_1 为 $[1,0,4,2,3,1,5,0]$，序列 h_1 为 $[3,1,4,0,0,0,0,0]$，给出循环卷积的具体过程如下。

第 0 步：



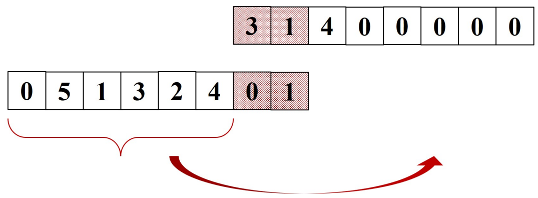
(a)



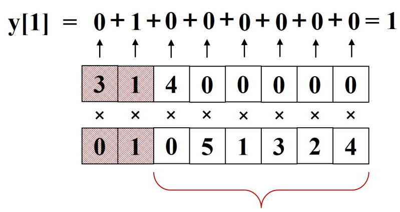
(b)

图 4: 第 0 步

第 1 步:



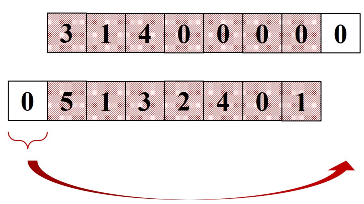
(a)



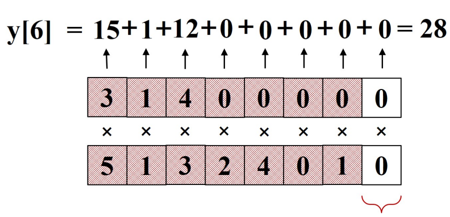
(b)

图 5: 第 1 步

类似的经过第 2-5 步，到第 6 步:



(a)



(b)

图 6: 第 6 步

最后，第 7 步:

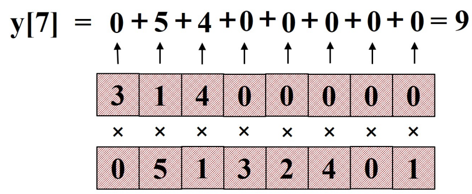


图 7: 第 7 步

由此我们得到序列 y_1，为 $[23, 1, 16, 10, 27, 14, 28, 9]$。

综上，循环卷积相比线性卷积，多了循环移位的一步（图中的粗箭头），循环卷积的输入是长度均为$\boldsymbol{N_1}$ 的两个序列，输出同样是长度为$\boldsymbol{N_1}$ 的序列。

注意：可以观察到，我们在举的循环卷积例子中的输入序列 x_1 与 h_1 是从线性卷积例的输入序列 x 与 h 经过一定数量的补零至相同长度得到的。最终得到的输出序列 y_1 与线性卷积例的输出序列 y 并不相同，但二者之间也存在关联。请思考，如果我们要对两个序列 x 与 h 进行线性卷积，能否先对这两个序列进行一定数量的补零得到 x_1 与 h_1，再对补零后序列进行循环卷积，使得线性卷积的结果 y 是循环卷积的结果 y_1 中的一部分？如果可以，需要至少补多少个零才能实现？

3　离散傅里叶变换 (Discrete Fourier Transform, DFT)

离散傅里叶变换 (Discrete Fourier Transform, DFT) 能将时域信号变换到频域进行处理，变换的物理意义会在《信号与系统》、《数字信号处理》等课程中详细讲述。我们这里仅给出 DFT 的定义如下：假设有长度为 N 的复数序列 $x[n], n = 0, 1, ..., N-1$，对其进行 DFT，得到一个同样长度为 N 的复数序列 $X[k], k = 0, 1, ..., N-1$，具体计算方式为：

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn},\ k = 0, 1, ..., N-1 \tag{1}$$

DFT 也存在其逆变换，能从序列 $X[k], k = 0, 1, ..., N-1$ 计算得到原复数序列 $x[n], n = 0, 1, ..., N-1$，具体计算方式为：

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot \mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn},\ n = 0, 1, ..., N-1 \tag{2}$$

我们之所以介绍 DFT，是因为 DFT 有一个很重要的性质：设有长度为 N 的两个复数序列 $x[n], n = 0, 1, ..., N-1$ 与 $h[n], n = 0, 1, ..., N-1$，二者循环卷积的结果是长度同样为 N 的复数序列 $y[n], n = 0, 1, ..., N-1$，即 $y[n] = h[n] \otimes x[n]$，三个序列进行 DFT 后得到的序列分别为 $X[k], k = 0, 1, ..., N-1$、$H[k], k = 0, 1, ..., N-1$ 与 $Y[k], k = 0, 1, ..., N-1$，那么以下等式成立：

$$Y[k] = X[k] \cdot H[k],\ k = 0, 1, ..., N-1 \tag{3}$$

这个性质给我们的解循环卷积问题（即 $y[n] = h[n] \otimes x[n]$，已知序列 y 与 h，求序列 x）提供了一种新思路：先分别对序列 y 与 h 进行 DFT，再将结果 Y 与 H 逐元素相除得到 X，再逆变换得到 x。请思考：该方法的时间复杂度为多少？与求解线性方程组的复杂度相比如何？

4 快速傅里叶变换 (Fast Fourier Transform, FFT)

快速傅里叶变换 (Fast Fourier Transform, FFT) 是 DFT 的一种快速算法，能够实现更低的时间复杂度。基 2 的 FFT 的实现方式可以参考网址 `https://blog.csdn.net/weixin_40106401/article/details/106128194`。

5 其他

其他需要注意的点：

- 实现 FFT 函数后，假设序列 x 经 FFT 得到序列 X，只需要对序列 X 经过一定的变换，再输入到 FFT 函数，就能得到序列 x，实现逆变换。请思考具体需要怎样的变换。
- 实现 FFT 需要引入复数与复数运算，在 C 语言中有对应的库，可以考虑采用。