

一维卷积与池化实验报告

题目要求

1. 使用 C 语言（或伪代码）编写实现“一维卷积和池化”，对一维卷积后的结果进行池化操作。
2. 自己构建大于等于 100 个数据的一维数组，对其进行卷积和池化，卷积核长度需大于等于 3，池化过滤器长度需大于等于 3。
3. 讨论是否有快速进行卷积和池化的编程实现方法。

实现思路

方法一（模拟）

模拟卷积和池化过程。

对于卷积运算，对于每个答案进行一次卷积运算。

对于池化运算，用循环对于每个池化长度求最值。

若数据长度为 n ，卷积核长度为 m ，池化长度为 l 。算法复杂度 $O(nm + n)$ 。效率较低。

方法二（FFT 优化）

对于模拟算法，我们发现，效率较低的部分主要在卷积计算部分。所以我们希望优化卷积的计算。

设原数组为 a ，卷积核为 b ，卷积结果为 ans 。根据定义有：

$$ans_i = \sum_{j=0}^{j < m} a_{i+j} \times b_j$$

若我们将 b 反转产生新数组 c ，则有：

$$ans_i = \sum_{j=0}^{j < m} a_{i+j} \times c_{m-1-j}$$

这个式子很像多项式乘法（其实就是卷积的初始定义），所以可以用 FFT 优化。

把整个算法复杂度降低至 $O(n \log n + m \log m + n)$ 。效率很高。

总结与反思

FFT 可以高效地计算卷积，但也存在一定问题。比如运用复数计算和浮点运算，常数较大，存在误差。

进一步优化可以使用快速数论变换（NTT）。