DFT 与 FFT

陈柏均

2025 年 4 月 25 日

1 DFT

1.1 信号

$\{x_k : k = 0, \ldots, N-1\}$，记 $\omega_N = e^{+2\pi i/N}$，其中 $z^n = 1$

$$y_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{nk}{N}} = \frac{1}{N}\sum_{n=0}^{N-1} x_n \omega_N^{-nk} \tag{1.1.1}$$

称 $\{y_0, \ldots, y_{N-1}\}$ 为 $\{x_0, \ldots, x_{N-1}\}$ 的 DFT。

记 $\mathcal{F}_N : \mathbb{C}^N \to \mathbb{C}^N, \quad \mathcal{F}_N(x) = y$。

1.2 逆公式

$$x_k = \sum_{n=0}^{N-1} y_n e^{2\pi i \frac{nk}{N}} = \sum_{n=0}^{N-1} y_n \omega_N^{nk}, \text{即} \mathcal{F}_N^{-1}(y) = x. \tag{1.2.1}$$

证明.

$$\begin{aligned}
\sum_{n=0}^{N-1} y_n \omega_N^{nk} &= \sum_{n=0}^{N-1} \left(\frac{1}{N}\sum_{r=0}^{N-1} x_r \omega_N^{-rn} \right) \omega_N^{nk} \\
&= \sum_{r=0}^{N-1} x_r \frac{1}{N}\sum_{n=0}^{N-1} \omega_N^{n(k-r)} \\
&= \sum_{r=0}^{N-1} x_r \delta_{k,r} = x_k.
\end{aligned} \tag{1.2.2}$$

其中

$$\delta_{k,r} = \begin{cases} 1, & k = r \\ 0, & k \neq r \end{cases} \tag{1.2.3}$$

□

$$\frac{1}{N}\sum_{n=0}^{N-1} \omega_N^{n(k-r)} = \delta_{k,r} \tag{1.2.4}$$

证明. 当 $k = r$ 时，显然成立。

当 $k \neq r$ 时，

$$\begin{aligned} \frac{1}{N}\sum_{n=0}^{N-1} \omega_N^{n(k-r)} &= \frac{1}{N} \cdot \frac{1-\omega_N^{N(k-r)}}{1-\omega_N^{k-r}} \cdot \omega_N^{k-r} \\ &= \frac{1}{N} \cdot \frac{1-\omega_N^{N(k-r)}}{1-\omega_N^{k-r}} \\ &= \frac{1}{N} \cdot \frac{1-(\omega_N^{N})^{k-r}}{1-\omega_N^{k-r}} = 0 \end{aligned} \tag{1.2.5}$$

这里用到了 $\omega_N^N = 1$。 □

1.3 Fourier 矩阵

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = F_N \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}, \tag{1.3.1}$$

$$F_N = \left(\omega_N^{-nk}\right)_{n,k=0}^{N-1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix} \tag{1.3.2}$$

1.3.1 性质 1

F_N 是酉矩阵，$F_N^* F_N = N I_{N\times N}$

考虑第 j 行 $(\omega_N^{j\times 0},\omega_N^{j\times 1},\omega_N^{j\times 2},\dots,\omega_N^{j(N-1)})\equiv J$

对于第 k 行 $(\omega_N^{k\times 0},\omega_N^{k\times 1},\omega_N^{k\times 2},\dots,\omega_N^{k(N-1)})\equiv K$

证明. 不同行正交

$$\begin{aligned}J^T\overline{K}&=\omega_N^{j\times 0}\overline{\omega_N^{k\times 0}}+\omega_N^{j\times 1}\overline{\omega_N^{k\times 1}}+\cdots+\omega_N^{j(N-1)}\overline{\omega_N^{k(N-1)}}\\&=\omega_N^{(j-k)\times 0}+\omega_N^{(j-k)\times 1}+\cdots+\omega_N^{(j-k)(N-1)}\\&=\sum_{m=0}^{N-1}\omega_N^{(j-k)m}=\frac{1-\omega_N^{(j-k)N}}{1-\omega_N^{j-k}}=0\end{aligned}\tag{1.3.3}$$

这里用到了 $\omega_N^N=1$。 □

证明. 不同列正交

由上同理可得 □

证明. 同一行 (列) 的内积为 N。

$$\begin{aligned}J^T\overline{J}&=\omega_N^{j\times 0}\overline{\omega_N^{j\times 0}}+\omega_N^{j\times 1}\overline{\omega_N^{j\times 1}}+\cdots+\omega_N^{j(N-1)}\overline{\omega_N^{j(N-1)}}\\&=1+\cdots+1=N.\end{aligned}\tag{1.3.4}$$

□

1.3.2 性质 2

$$F_N^{-1}=\frac{1}{N}F_N^*=\left(\omega_N^{nk}\right)\tag{1.3.5}$$

1.4 二进制形式下的 Fourier 矩阵 ($N=2^p$)

$$F_2=\begin{pmatrix}1&1\\1&\omega_2^{-1\times 1}\end{pmatrix}=\begin{pmatrix}1&1\\1&e^{-i\frac{2\pi}{2}}\end{pmatrix}=\begin{pmatrix}1&1\\1&-1\end{pmatrix}\tag{1.4.1}$$

$$\begin{cases}y_0=\dfrac{1}{2}(x_0+x_1\omega_2^{-0\times 1})\\ y_1=\dfrac{1}{2}(x_0+x_1\omega_2^{-1\times 1})\end{cases}\Leftrightarrow\begin{pmatrix}y_0\\y_1\end{pmatrix}=\frac{1}{2}F_2\begin{pmatrix}x_0\\x_1\end{pmatrix}=\begin{pmatrix}\frac{1}{2}(x_0+x_1)\\\frac{1}{2}(x_0-x_1)\end{pmatrix}\tag{1.4.2}$$

表 1: 计算量

total cost	**矩阵计算**
乘法次数 (multiplications): 0 次 加法次数 (additions): 2 次	除法及计算: Add 2^1 multi 2^2

$$F_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega_4^{-1\times1} & \omega_4^{-2\times1} & \omega_4^{-3\times1} \\ 1 & \omega_4^{-1\times2} & \omega_4^{-2\times2} & \omega_4^{-3\times2} \\ 1 & \omega_4^{-1\times3} & \omega_4^{-2\times3} & \omega_4^{-3\times3} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \tag{1.4.3}$$

$$\omega_4 = e^{\frac{2\pi i}{4}} = i, \tag{1.4.4}$$

1.5 DFT 的计算量 (cost of computation)

$$y_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n \omega_N^{-nk} \tag{1.5.1}$$

每个 y_k 需要 $(N-1)$ 个乘法，N 个加法。

总乘法次数：$(N-1)^2$

总加法次数：$N(N-1)$

注记 1.5.1. 计算 y_k 时不需要乘法，故不是 N^2；用矩阵形式 $y = F_N x$ 理解时，乘法次数为 N^2；计算复杂度：$O(N^2)$！

2 FFT

1965 年，Cooley 和 Tukey 提出 FFT 算法。(An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation 19(90): 297-301, Jan 1965. DOI: 10.1090/S0025-5718-1965-0178586-1)

2.1 FFT $(N = 2^2 = 4)$

$$F_4 : \{x_0, x_1, x_2, x_3\} \to \{y_0, y_1, y_2, y_3\} \tag{2.1.1}$$

$$y_k = \frac{1}{4}\left(x_0\omega_4^{-0\times k} + x_1\omega_4^{-1\times k} + x_2\omega_4^{-2\times k} + x_3\omega_4^{-3\times k}\right), \quad k = 0, 1, 2, 3 \tag{2.1.2}$$

奇偶重排 (rearrange) y_k:

$$\begin{aligned} J^T\overline{K} &= \omega_N^{j\times 0}\overline{\omega_N^{k\times 0}} + \omega_N^{j\times 1}\overline{\omega_N^{k\times 1}} + \cdots + \omega_N^{j(N-1)}\overline{\omega_N^{k(N-1)}} \\ &= \omega_N^{(j-k)\times 0} + \omega_N^{(j-k)\times 1} + \cdots + \omega_N^{(j-k)(N-1)} \\ &= \sum_{m=0}^{N-1}\omega_N^{(j-k)m} = \frac{1-\omega_N^{(j-k)N}}{1-\omega_N^{j-k}} = 0 \end{aligned} \tag{2.1.3}$$

$$P_k = \frac{x_0 + x_2\omega_2^{-k}}{2}, \quad I_k = \frac{x_1 + x_3\omega_2^{-k}}{2}, \quad k = 0, 1, 2, 3. \tag{2.1.4}$$

显然：

$$P_{k+2} = P_k, \quad k = 0, 1 \quad 和 \quad I_{k+2} = I_k, \quad k = 0, 1. \tag{2.1.5}$$

因此：

$$\begin{pmatrix} P_0 \\ P_1 \end{pmatrix} = F_2\begin{pmatrix} x_0 \\ x_2 \end{pmatrix}, \quad \begin{pmatrix} I_0 \\ I_1 \end{pmatrix} = F_2\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} \tag{2.1.6}$$

则：

$$\begin{cases} y_k = \frac{1}{2}\left(P_k + \omega_4^{-k}I_k\right), \quad k = 0, 1 \\ y_{k+2} = \frac{1}{2}\left(P_{k+2} + \omega_4^{-(k+2)}I_{k+2}\right) = \frac{1}{2}\left(P_k - \omega_4^{-k}I_k\right), \quad k = 0, 1 \end{cases} \tag{2.1.7}$$

表 2: 计算量

	$P_k\ (k=0,1)$	$I_k\ (k=0,1)$	$y_k\ (k=0,1)$	$y_{k+2}\ (k=0,1)$
乘法	0 次	0 次	1 次	1 次
加法	2 次	2 次	2 次	2 次

至此，$\{y_0,y_1,y_2,y_3\}$ 全部算出。

计算 $\{b_0,b_1,b_2,b_3\}$ 过程中的计算量：乘法 1 次（表面上 2 次），加法 8 次

对比直接按公式计算的计算量：乘法 $(4-1)^2=9$，加法 $4\times(4-1)=12$ 次

2.2 FFT ($N=2^3=8$)

$$F_8:\{x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7\}\to\{y_0,y_1,\dots,y_7\} \tag{2.2.1}$$

2.2.1 递推公式

$$\begin{aligned}
y_j &= \frac{1}{8}\left(x_0+x_1\omega_8^{1\times j}+x_2\omega_8^{2\times j}+\cdots+x_6\omega_8^{6\times j}+x_7\omega_8^{7\times j}\right)\\
&=\frac{1}{2}\left(\frac{1}{4}\left(x_0+x_2\omega_8^{2\times j}+x_4\omega_8^{4\times j}+x_6\omega_8^{-6\times j}\right)\right.\\
&\quad\left.+\frac{1}{4}\left(x_1\omega_8^{3\times j}+x_3\omega_8^{3\times j}+x_5\omega_8^{5\times j}+x_7\omega_8^{-7\times j}\right)\right)\\
&=\frac{1}{2}\left(\frac{1}{4}\left(x_0+x_2\omega_4^{2\times j}+x_4\omega_4^{0\times j}+x_6\omega_4^{-0\times j}\right.\right.\\
&\quad\left.\left.+x_1\omega_4^{1\times j}+x_3\omega_4^{-1\times j}+x_5\omega_4^{2\times j}+x_7\omega_4^{-2\times j}\right)\omega_8^{j}\right)\\
&=\frac{1}{2}\left(P_j^{(3)}+I_j^{(3)}\omega_8^{j}\right)
\end{aligned} \tag{2.2.2}$$

其中：

$$P_j^{(3)}=\frac{1}{4}\left(x_0+x_2\omega_4^{2\times j}+x_4\omega_4^{0\times j}+x_6\omega_4^{-0\times j}\right) \tag{2.2.3}$$

$$I_j^{(3)}=\frac{1}{4}\left(x_1+x_3\omega_4^{2\times j}+x_5\omega_4^{0\times j}+x_7\omega_4^{-0\times j}\right) \tag{2.2.4}$$

可证:

$$\begin{cases}P_j^{(3)}=P_{j+4}^{(3)}, & j=0,1,2,3\\ I_j^{(3)}=I_{j+4}^{(3)}\end{cases} \tag{2.2.5}$$

2.2.2 计算复杂度分析

第一层 (A)

已知$P_j^{(3)}, I_j^{(3)} (j=0,1,2,3)$, 计算$P_j^{(3)} (j=0,1,2,3)$ 的计算量

- 乘法 3 次 (当 $j=0$ 时无乘法)
- 加法 8 次

第二层 (B)

下面讨论$P_j^{(3)}$ 的计算量 $(j=0,1,2,3)$

一方面

$$\begin{aligned} P_j^{(3)} &= \frac{1}{4}(x_0,x_2,x_4,x_6)\cdot(1,w_4^{-j},w_4^{-2j},w_4^{-3j}) = \mathrm{DFT}(x_0,x_2,x_4,x_6) \\ &= \frac{1}{2}\left\{\frac{1}{2}\left(x_0+x_4w_4^{-2j}\right)+\frac{1}{2}\left(x_2+x_6w_4^{-2j}\right)w_4^{-j}\right\} \\ &= \frac{1}{2}\left\{P_j^{(2)}+I_j^{(2)}w_4^{-j}\right\} \quad (\text{同样地有}I_j^{(3)}=P_j^{(2)}+I_j^{(2)}w_4^{-j} \quad j=0,1) \end{aligned} \tag{2.2.6}$$

如果已知 $P_j^{(2)}, I_j^{(2)}, j=0,1,2$，计算 $P_j^{(3)}$ 需要 1 次乘法，4 次加法。

同样地

$$\begin{aligned} I_j^{(3)} &= \frac{1}{4}(x_1,x_3,x_5,x_7)\cdot(1,w_4^{-j},w_4^{-2j},w_4^{-3j}) \quad (j=0,1,2,3) \\ &= \frac{1}{2}\left\{\frac{1}{2}\left(x_1+x_5w_4^{-2j}\right)+\frac{1}{2}\left(x_3+x_7w_4^{-2j}\right)w_4^{-j}\right\} \\ &= \frac{1}{2}\left\{P_j^{(2)}+I_j^{(2)}w_4^{-j}\right\} \quad (j=0,1) \end{aligned} \tag{2.2.7}$$

三层 (C)

计算 $P_j^{(2)}, I_j^{(2)}, P_j^{(2)}, I_j^{(2)}$ ($j=0,1$) 分别需要 0 次乘法，2 次加法。

结论：(A)、(B)、(C) 三层共花费：乘法 $3+2+0=5$ 加法 $8+8+6=24$

注记 2.2.1. 直接计算时，乘法 $4\times 3=12$ 次，加法 $4\times 3=12$ 次

2.3 FFT ($N=2^8$, general case)

2.3.1 定义和基本公式

$$F_{2^q}:\mathbb{C}^{2^q}\to\mathbb{C}^{2^q} \tag{2.3.1}$$

$$(x_0, x_1, \ldots, x_{2^q-1}) \to (y_0, y_1, \ldots, y_{2^q-1}) \tag{2.3.2}$$

$$y_k = \frac{1}{2^q}(x_0, x_1, \ldots, x_{2^q-1}) \cdot \left(1, w_{2^q}^{-k}, w_{2^q}^{-2k}, \ldots, w_{2^q}^{-(2^q-1)k}\right) \tag{2.3.3}$$

$$y_k = \frac{1}{2^q}\sum_{j=0}^{2^q-1} x_j \left(w_{2^q}\right)^{-jk}, \quad k = 0, 1, \ldots, 2^q - 1 \tag{2.3.4}$$

2.3.2 递推公式

$$y_k = \frac{1}{2}\left\{\frac{1}{2^{q-1}}\sum_{j=0}^{2^{q-1}-1} x_{2j}\left(w_{2^{q-1}}\right)^{-jk} + \frac{1}{2^{q-1}}\sum_{j=0}^{2^{q-1}-1} x_{2j+1}\left(w_{2^{q-1}}\right)^{-jk}\left(w_{2^q}\right)^{-k}\right\} \tag{2.3.5}$$

$$y_k = \frac{1}{2}\left(P_k^{(q-1)} + w_{2^q}^{-k} I_k^{(q-1)}\right), \quad k = 0, 1, \ldots, 2^q - 1 \tag{2.3.6}$$

其中：

$$P_k^{(q-1)} = \frac{1}{2^{q-1}}\sum_{j=0}^{2^{q-1}-1} x_{2j}\left(w_{2^{q-1}}\right)^{-jk}, \quad k = 0, 1, \ldots, 2^{q-1} - 1 \tag{2.3.7}$$

$$I_k^{(q-1)} = \frac{1}{2^{q-1}}\sum_{j=0}^{2^{q-1}-1} x_{2j+1}\left(w_{2^{q-1}}\right)^{-jk}, \quad k = 0, 1, \ldots, 2^{q-1} - 1 \tag{2.3.8}$$

利用 $\left(w_{2^q}\right)^{-(k+2^{q-1})} = -\left(w_{2^q}\right)^k$，得到：

$$\begin{cases} P_{k+2^{q-1}}^{(q-1)} = P_k^{(q-1)} \\ I_{k+2^{q-1}}^{(q-1)} = I_k^{(q-1)} \end{cases} \quad k = 0, 1, \ldots, 2^{q-1} - 1 \tag{2.3.9}$$

因此有公式：

$$\begin{cases} y_k = \dfrac{1}{2}\left[P_k^{(q-1)} + \left(w_{2^q}\right)^{-k} I_k^{(q-1)}\right], \quad k = 0, 1, \ldots, 2^{q-1} - 1 \\ y_{k+2^{q-1}} = \dfrac{1}{2}\left[P_k^{(q-1)} - \left(w_{2^q}\right)^{-k} I_k^{(q-1)}\right], \quad k = 0, 1, \ldots, 2^{q-1} - 1 \end{cases} \tag{2.3.10}$$

特别地：

$$P_k^{(q-1)} = \mathrm{DFT}\left\{x_{2j}\right\}_{j=0}^{2^{q-1}-1}, \quad k = 0, 1, \ldots, 2^{q-1} - 1 \tag{2.3.11}$$

$$I_k^{(q-1)} = \mathrm{DFT}\left\{x_{2j+1}\right\}_{j=0}^{2^{q-1}-1}, \quad k = 0, 1, \ldots, 2^{q-1} - 1 \tag{2.3.12}$$

故有：

$$\begin{cases} P_k^{(q-1)} = P_k^{(q-2)} + \left(w_{2^{q-1}}\right)^{-k} I_k^{(q-2)} \\ I_k^{(q-1)} = P_k^{(q-2)} + \left(w_{2^{q-1}}\right)^{-k} I_k^{(q-2)} \end{cases} \quad k = 0, 1, \ldots, 2^{q-2} - 1 \tag{2.3.13}$$

2.3.3 FFT 算法中乘法和加法次数的计算

M_q : 在 FFT 算法中的乘法次数, 数据长度为 $\mathrm{N} = 2^q$
A_q : 在 FFT 算法中的加法次数

已知：

$$\begin{cases} M_1 = 0 \quad A_1 = 2 \\ M_2 = 1 \quad A_2 = 8 \\ M_3 = 5 \quad A_3 = 24 \end{cases} \tag{2.3.14}$$

下面计算 M_q 和 A_q：

首先讨论 M_q 和 M_{q-1} 的关系以及 A_q 和 A_{q-1} 的关系：

计算 $P_k^{(q-1)}$ (DFT of even)　乘法次数：M_{q-1}　加法次数：A_{q-1}

计算 $I_k^{(q-1)}$　乘法次数：M_{q-1}　加法次数：A_{q-1}

乘法次数：$2^{q-1} - 1$　加法次数：2^q

$$\begin{cases} M_q = 2M_{q-1} + 2^{q-1} - 1 \\ M_1 = 0 \end{cases} \tag{2.3.15}$$

$$\begin{cases} A_q = 2A_{q-1} + 2^q \\ A_1 = 2 \end{cases} \tag{2.3.16}$$

表 3: FFT 算法中乘法和加法次数的计算步骤

计算步骤	乘法次数	加法次数
计算 $P_k^{(q-1)}$	M_{q-1}	A_{q-1}
计算 $I_k^{(q-1)}$	M_{q-1}	A_{q-1}
公式 (1) 乘法	$2^{q-1} - 1$	2^q

例题 2.3.1.

$$\begin{cases} A_q = 2A_{q-1} + 2^q \\ A_1 = 2 \end{cases} \tag{2.3.17}$$

解 2.3.1. 由(2.3.17)式知 $A_{q-j} = 2A_{q-j-1} + 2^{q-j}$

则

$$2^j A_{q-j} = 2^{j+1} A_{q-j-1} + 2^q \tag{2.3.18}$$

令(2.3.18)中 $j = 0, 1, \ldots, q-2$，然后相加：

$$\begin{cases} A_q = 2A_{q-1} + 2^q \\ 2A_{q-1} = 2^2 A_{q-2} + 2^q \\ 2^2 A_{q-2} = 2^3 A_{q-3} + 2^q \\ \vdots \\ 2^{q-2} A_2 = 2^{q-1} A_1 + 2^q \end{cases} \tag{2.3.19}$$

由上可得

$$A_q = 2^{q-1} A_1 + (q-1)2^q \tag{2.3.20}$$

则

$$A_q = 2^q + (q-1)2^q = q2^q = N \log_2 N \tag{2.3.21}$$

因此加法计算量为 $N \log_2 N = O(N \log N)$

例题 2.3.2.

$$\begin{cases} M_q = 2M_{q-1} + 2^{q-1} - 1 \\ M_1 = 0 \end{cases} \tag{2.3.22}$$

解 2.3.2. 由(2.3.22)式知

$$M_{q-j} = 2M_{q-j-1} + 2^{q-j-1} - 1, \quad j = 0, 1, \ldots, q-2 \tag{2.3.23}$$

则

$$2^j M_{q-j} = 2^{j+1} M_{q-j-1} + 2^{q-1} - 2^j \tag{2.3.24}$$