△ Fourier transform : $f: [0,1] \to \mathbb{C}$ $f(0)=f(1)$

(FT) $$a_n = \hat{f}(n) = \int_0^1 f(x)\, e^{2\pi i n x}\, dx$$

(FI) $$f(x) = \sum_{n\in\mathbb{Z}} a_n\, e^{-2\pi i n x}$$

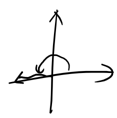
△ Divide $[0,1]$ into N pieces of equal length. $[\frac{j}{N}, \frac{j+1}{N})$

(DFT) $$a_n = \sum_{0\le j\le N-1} f\left(\frac{j}{N}\right) e^{2\pi i n \frac{j}{N}}$$

$\omega = e^{\frac{2\pi}{N} i}$ $\mathcal{F}_N$ is $N\times N$ matrix having ij entry $\omega^{(i-1)(j-1)}$

(DFT) : $$\begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} = \mathcal{F}_N \cdot \begin{pmatrix} f(0) \\ f(\frac{1}{N}) \\ \vdots \\ f(\frac{N-1}{N}) \end{pmatrix}$$

(DFI) $$\begin{pmatrix} f(0) \\ f(\frac{1}{N}) \\ \vdots \\ f(\frac{N-1}{N}) \end{pmatrix} = \mathcal{F}_N^{-1} \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix}$$



△ Example : $N=2$ $f(0),\ f(\frac{1}{2})$ $e^0=1$ $e^{\pi i} = -1$

$$a_0 = f(0)\, e^{2\pi i \frac{0\cdot 0}{N}} + f(\tfrac{1}{2})\, e^{2\pi i \frac{0\cdot 1}{N}} = f(0)+f(\tfrac{1}{2})$$

$$= \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} f(0) \\ f(\frac{1}{2}) \end{pmatrix}$$

$$a_1 = f(0)\, e^{2\pi i \frac{1\cdot 0}{N}} + f(\tfrac{1}{2}) \cdot e^{2\pi i \frac{1\cdot 1}{2}} = f(0) - f(\tfrac{1}{2})$$

$$= \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} f(0) \\ f(\frac{1}{2}) \end{pmatrix}$$

$$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} f(0) \\ f(\frac{1}{2}) \end{pmatrix}$$

$i^4 = 1$ $i^9 = i$

$i^6 = i^4 \cdot i^2 = -1$

$$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} f(0) \\ f(\frac{1}{2}) \end{pmatrix}$$

$i^4 = 1$

$i^6 = i^4 \cdot i^2 = -1$

Δ $N = 4$.

$$(F_N)_{ij} = w^{(i-1)(j-1)} \qquad w = e^{\frac{2\pi i}{N}} = e^{\frac{2\pi i}{4}} = e^{\frac{\pi}{2} i} = i \qquad i^2 = -1$$

$$\underline{F_4} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

$i = 2 \quad w^{(j-1)}$

$i = 3 \quad w^{2(j-1)} = (-1)^{j-1}$

$i = 4 \quad w^{3(j-1)}$

$$a_1 = f(0) \underbrace{e^{2\pi i \frac{1 \cdot 0}{4}}}_{1} + f\left(\frac{1}{4}\right) \underbrace{e^{2\pi i \frac{1 \cdot 1}{4}}}_{i} + f\left(\frac{2}{N}\right) \underbrace{e^{2\pi i \frac{1 \cdot 2}{4}}}_{i^2} + f\left(\frac{3}{N}\right) \underbrace{e^{2\pi i \frac{1 \cdot 3}{4}}}_{i^3}$$

Δ Lem: $\left(\frac{1}{\sqrt{N}} F_N\right)^* \cdot \left(\frac{1}{\sqrt{N}} F_N\right) = I$, So F_N is invertible $F_N^{-1} = \frac{1}{N} F_N^*$

Pf: it suffices to prove $\underline{F}^* \cdot \underline{F} = \underline{N \cdot I}$

$$(F)_{ij} = w^{(i-1)(j-1)} \qquad (F^*)_{ij} = (\overline{F_{ji}}) = w^{-(i-1)(j-1)}$$

fact: $|z| = 1$, then $\bar{z} = z^{-1}$, then $|w| = 1$ $\quad \bar{w} = w^{-1}$

fact: $(A \cdot B)_{ij} = \sum_{1 \le k \le N} A_{ik} \cdot B_{kj}$

$$(F^* \cdot F)_{ij} = \sum_k F^*_{ik} \cdot F_{kj} = \sum_k w^{-(i-1)(k-1)} \cdot w^{(k-1)(j-1)}$$

$$= \sum_k w^{(k-1) \cdot (j-i)}$$

Case 1: $j = i$ then $(F^* F)_{ii} = \sum w^0 = \sum 1 = N$

Case 2: $j \neq i$ $\quad (F^* F)_{ij} = \sum_k w^{(k-1)(j-i)}$, $\quad w^{j-i} \neq 1$

Expect to be 0 $\qquad w^N = 1$

Expect to be 0 $W^N = 1$

Trick: $x = w^{j-i}$ $x \neq 1$, $x^N = 1$

Claim $S = \sum_{k=1}^{N} x^{(k-1)} = 1 + x + x^2 + \cdots x^{N-1}$

Pf: $x \cdot S = x + x^2 + x^3 + \cdots + \underbrace{x^N}_{=1} = S$

$\begin{cases} x \neq 1 \\ xS = S \end{cases} \Rightarrow S = 0$

$(x-1)S = 0$

#

Δ Rmk: (1) $\underbrace{x \cdot y}_{}$ is more time consuming than addition $\underbrace{x+y}_{}$

N digits N^2-times N times

(2) Q: for DFT How efficient is direct computation $F \cdot \begin{pmatrix} f(0) \\ \vdots \\ f(\frac{N-1}{N}) \end{pmatrix}$?

How many of multiplications in this algorithm in terms of N?

rows $N \times N$ $O(N^2)$

Δ Fast FT can reduce $O(N \cdot \log(N)) = O(N^{1+\varepsilon})$

Focus on $N = 2^\ell$ $N=2$ $N=4$

$$F_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

$$F_4 \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{P} = \left(\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & i & -i \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -i & i \end{array}\right)$$

F_2, $D_2 = \begin{pmatrix} 1 & \\ & i \end{pmatrix}$, $\begin{pmatrix} 1 & \\ & i \end{pmatrix} F_2$, $-\begin{pmatrix} 1 & \\ & i \end{pmatrix} F_2$

$$= \left(\begin{array}{c|c} I & D_2 \\ \hline I & -D_2 \end{array}\right) \cdot \left(\begin{array}{c|c} F_2 & \\ \hline & F \end{array}\right)$$

$$= \left(\begin{array}{c|c} I & D_2 \\ \hline I & -D_2 \end{array}\right) \cdot \left(\begin{array}{c|c} F_2 & \\ \hline & F_2 \end{array}\right)$$

$$\underline{F_4 = \left(\begin{array}{cc} I & D_2 \\ I & -D_2 \end{array}\right)} \left(\begin{array}{c|c} F_2 & \\ \hline & F_2 \end{array}\right) \underline{P^{-1}}$$

△ In General $\underline{F_{2n} = \left(\begin{array}{cc} I_n & D_n \\ I_n & -D_n \end{array}\right) \left(\begin{array}{cc} F_n & 0 \\ 0 & F_n \end{array}\right) P^{-1}}$

$$D_n = \begin{pmatrix} 1 & & & \\ & w & & \\ & & w^2 & \\ & & & \ddots \\ & & & & w^{n-1} \end{pmatrix} \qquad \zeta = e^{\frac{2\pi i}{n}} \qquad w = e^{\frac{2\pi i}{2n}}$$

Ingredient :

i-th row of F_{2n} is $\underline{w^{(i-1)(j-1)}}$ w is $2n$-th root of unity

$j = 1, 3, 5, 7, \cdots$ $\quad w^{(i-1)(j-1)} = w^{(i-1)2k} = (w^2)^{(i-1)\cdot k} = \underline{\zeta^{(i-1)k}}$

$j = 2k+1$ $\quad w^2 = e^{\frac{2\pi i}{2n}\cdot 2} = e^{\frac{2\pi i}{n}} = \zeta$

these $\zeta^{(i-1)k}$ forms i-th row of F_n

$j = 2, 4, 6 \sim$ $\quad j = 2m$ $\quad w^{(i-1)(j-1)} = w^{(i-1)(2m-1)}$

$$= w^{(i-1)(2m-2+1)} = (\underline{w^2})^{(i-1)\cdot m} \cdot w^{i-1}$$

$$= \boxed{(\zeta)^{(i-1)m}} \; \underline{w^{i-1}} \leftarrow \text{related to } D_n$$

$$\underline{w^n} = e^{\frac{2\pi i}{2n}\cdot n} = e^{\pi i} = -1 \leftarrow \text{related to } -D_n$$

△ Efficiency of algorithm : $\Phi(2^n)$ Counting # multiplication for $F_{2^n} \cdot x$

Suppose $\Phi(2^{n-1})$ this is known

$$F_{2^n} x = \begin{pmatrix} I & D \\ I & -D \end{pmatrix} \begin{pmatrix} F_{2^{n-1}} & \\ & F_{2^{n-1}} \end{pmatrix} P^{-1} \cdot x$$

Step 1 no multiplications

Step 2: $2 \cdot \Phi(2^{n-1})$

Step 3 additional 2^n multiplications

$$\Phi(2^n) = 2\Phi(2^{n-1}) + 2^n$$

Check: $\Phi(2^n) = n \cdot 2^n$ is solution to this recursive relation

$$n \cdot 2^n \overset{?}{=} 2 \cdot \underbrace{(n-1)\, 2^{n-1}}_{} + 2^n$$

$$(n-1)\, 2^n + 2^n = n \cdot 2^n$$

$N = 2^n$ $\quad \Phi(N) = (\log_2 N) N \qquad (\quad N^2$

More efficient than previous approach

#

△ Dual vector space and tensor product

Motivation: generalize Matrix (2-dim datum) to higher dimension and make sense of multiplication

Given a vector space V

△ Definition a linear functional L on V is a linear operator

$$L: V \to \mathbb{C}$$

Denote by $\hat{V} = \{$ lineal functionals $\}$

which forms a vector space over $\mathbb{C}$

$$\begin{cases} a\cdot L : V \to \mathbb{C} \\ \quad v \mapsto aLv \\ L_1 + L_2 : V \to \mathbb{C} \\ \quad v \mapsto L_1 v + L_2 v \end{cases}$$

We call $\hat{V}$ the dual vector space of V

△ Explicitly, we pick a basis $\{e_1 \cdots e_n\}$ for V

then $L \in \hat{V}$ is determined by $(Le_1, \cdots Le_n)$

if $v = (e_1 \cdots e_n)\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ then $Lv = (Le_1 \cdots Le_n)\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$

Conversely for any n-tuple $(a_1 \cdots a_n)$ we can associate

a linear functional $L : V \to \mathbb{C}$

$e_i \mapsto a_i$

Cor: if $\dim V = n$, then $\dim \hat{V} = n$.

△ Cor: for $v \in V$, if $Lv = 0$ for any $L \in \hat{V}$, then $v = 0$

Pf: $v \neq 0$, then $v = (e_1 \cdots e_n)\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, for some $x_j \neq 0$

$$(a_1 \cdots a_n) = (0 \cdots \underset{\substack{\uparrow \\ j\text{-th}}}{1}, 0 \cdots 0)$$ then associated L

$$Lv = (a_1 \cdots a_n)\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = x_j \neq 0$$

△ Remark: we have a natural bijection $V \xrightarrow{\sim} \hat{\hat{V}}$

$v \longmapsto$ linear functional on $\hat{V}$

given by $L \longmapsto Lv$

This is true for finite-dim V.S.

but not true for ∞-dim V.S.

△ Def: Given a basis $\{e_i\}$ for V, the <u>dual basis</u> is a basis $\{E_i\}$ for $\hat{V}$ s.t. $E_i e_j = \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i\neq j \end{cases}$

In concise form, we can write $(\underline{E}_1 \cdots E_n)^T \cdot (\underline{e}_1 \cdots e_n) = I_{\text{size } n}$