22. Fast Fourier Transform

Thursday, October 28, 2021   11:14 PM

frequency domain   signal, time domain

Define: The Fourier transform $X(\omega)$ of a function $x(t)$ is given by

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt$$

And the inverse Fourier transform is

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{i\omega t} d\omega$$

But in practice, we actually use the Discrete Fourier Transform, which works because of the Nyquist-Shannon sampling theorem.

Nyquist-Shannon sampling theorem

If a function $x(t)$ contains no frequencies B Hz or higher, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2B}$ sec apart.

proof. Let $X(\omega)$ be the spectrum of $x(t)$

Then $x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{i\omega t} d\omega = \frac{1}{2\pi} \int_{-2\pi B}^{2\pi B} X(\omega) e^{i\omega t} d\omega$

because $X(\omega) = 0$ outside the band $\left|\frac{\omega}{2\pi}\right| < B$.

Let $t = \frac{n}{2B}$ for $n \in \mathbb{Z}$.

Then $\underbrace{x\left(\frac{n}{2B}\right)}_{\text{sampling } x \text{ at } t = \frac{n}{2B}} = \frac{1}{2\pi} \int_{-2\pi B}^{2\pi B} X(\omega) e^{i\omega \cdot \frac{n}{2B}} d\omega$.

Recall the Fourier series of a periodic function $f(y)$ is given by

$$f(y) = \sum_{n=-\infty}^{\infty} c_n e^{i \cdot \frac{2\pi n y}{P}}, \quad \text{where } P = \text{period of } f(y).$$

$$c_n = \frac{1}{P} \int_P f(y) e^{-i\frac{2\pi n y}{P}}.$$

Let $P = 4\pi B$, $c_n = \frac{1}{4\pi B} \int_{-2\pi B}^{2\pi B} f(y) e^{-iy \cdot \frac{n}{2B}} dy$.

The RHS above is precisely coefficients of the Fourier series expansion of $X(\omega)$ when taken as a $4B$-periodic function.

$\Rightarrow X(\omega)$ is completely determined by sampling at $t = \frac{n}{2B}$, $n \in \mathbb{Z}$,

$\Rightarrow X(\omega)$ is completely determined by sampling at $t = \frac{n}{2B}$, $n \in \mathbb{Z}$, for $\omega \in [-2B, 2B]$, and $X(\omega) = 0$ outside that interval.

Hence, we know $X(\omega)$ after sampling.

But $x(t)$ is determined by $X(\omega)$ $\Rightarrow$ $x(t)$ determined by sampling $\frac{1}{2B}$ sec apart.

Hence DTFT can recover original signal.

DFT gives discrete-time-Fourier transform of N-periodic sequence with only discrete frequency components, i.e. is a sampling of the DTFT.

Define: The **Discrete Fourier Transform** (DFT) of a vector $\vec{a} = (a_0, a_1, \ldots, a_{N-1})$ into another vector $F(\vec{a}) = \vec{A} = (A_0, A_1, \ldots, A_{N-1})$, is given by

$$A_k = \sum_{n=0}^{N-1} a_n e^{-\frac{i2\pi}{N}kn}$$

And its inverse transform F^{-1} is given by $F^{-1}(\vec{A}) = \vec{a}$, where

$$a_n = \frac{1}{N}\sum_{k=0}^{N-1} A_k e^{\frac{i2\pi}{N}kn}.$$

Note:

$$F^{-1}(F(\vec{a}))_m = \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{n=0}^{N-1} a_n e^{-\frac{i2\pi}{N}kn}\right) e^{\frac{i2\pi km}{N}} = \frac{1}{N}\sum_{n=0}^{N-1}\sum_{k=0}^{N-1} a_n e^{\frac{i2\pi}{N}k(m-n)}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} a_n \underbrace{\sum_{k=0}^{N-1} e^{\frac{i2\pi}{N}k(m-n)}}_{\substack{=N \text{ if } m=n \\ =0 \text{ if } m\neq n}} = \sum_{n=0}^{N-1} a_n\, \delta(m-n) = a_m.$$

(sum of roots of unity)

Note: We will view $\vec{a}$ and $\vec{A}$ as periodic functions on $\mathbb{Z}$.

Linearity: $F(x\vec{a} + y\vec{b}) = xF(\vec{a}) + yF(\vec{b})$ (obvious from def)

Time-reversal: Let $\vec{a}^R = (a_0, a_{N-1}, a_{N-2}, \ldots, a_1)$. ← note: different kind of reversal than last time because we aren't reversing the 0 element.

Then $F(\vec{a}^R) = F(\vec{a})^R$.

proof.

$$F(\vec{a}^R)_k = \sum_{n=0}^{N-1} a_{N-n}\, e^{-\frac{i2\pi}{N}kn}$$

replace n by $N-n$

roots of unity

$$= \sum_{\ldots}^{N-1} a_{\ldots}\; e^{-\frac{i2\pi}{N}k(n-N)}$$

because $e^{-\frac{i2\pi}{N}\cdot kN} = e^{-i2\pi k} = 1$

$$= \sum_{n=0}^{N-1} a_{N-n} \, e^{\frac{-i2\pi}{N} k(n-N)}$$

because $e^{\frac{-i2\pi}{N}\cdot kN} = e^{-i2\pi k} = 1$

$$= \sum_{m=1}^{N} a_m e^{\frac{-i2\pi}{N} k(-m)}$$

let $m = N-n$

$n=0 \iff m=N$

$$= \sum_{m=0}^{N-1} a_m e^{-\frac{i2\pi}{N} m(N-k)} = \left(F(\vec{a})^R\right)_k$$

Note: Our vectors in the wavelet proof from yesterday need to be zero-padded to align definitions of reversal.

$$(0, a_0, a_1, \ldots, a_{N-1})^R = (0, a_{N-1}, a_{N-2}, \ldots, a_1, a_0)$$

Symmetric if real: Let $\vec{a} \in \mathbb{R}^N$.

Then $F(\vec{a})_k = \overline{F(\vec{a})_{N-k}}$.

proof.
$$F(\vec{a})_k = \sum_{n=0}^{N-1} a_n e^{\frac{-i2\pi}{N} nk} = \sum_{n=0}^{N-1} a_n e^{-\frac{i2\pi}{N} n(-N+k)}$$

$$= \overline{\sum_{n=0}^{N-1} a_n e^{\frac{i2\pi}{N} n(N-k)}} = \overline{\sum_{n=0}^{N-1} a_n e^{-\frac{i2\pi}{N} n(N-k)}} = \overline{F(\vec{a})_{N-k}}.$$

Note: Again, we need to 0-pad in the wavelet proof.

Convolution theorem

Let $(f * g)[k] := \sum_{m=0}^{N-1} f[m]\, g[k-m]$ be the circular convolution of two periodic functions $f, g : [N] \to \mathbb{R}$

i.e. $f[N] = f[0]$
$f[N+1] = f[1]$

Then $F(f*g) = F(f)\, F(g)$.

proof.
$$F(f*g)[k] = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f[m]\, g[n-m]\, e^{-\frac{i2\pi}{N} nk}$$

Let $p = n-m$

$$= \sum_{n=0}^{N-1} \sum_{p=n}^{n-N+1} f[m]\, g[p]\, e^{-\frac{i2\pi}{N}(p+m)k}$$

$$= \sum_{m=0}^{N-1} f[m] e^{\frac{-i2\pi}{N} mk} \sum_{p=0}^{N-1} g[p] e^{-\frac{i2\pi}{N} pk} = F(f)\, F(g)\, [k].$$

Note: If we pad our vectors with enough zeros, we can extract a linear convolution of finite support vectors from this periodic result.

Fourier transform of Kronecker delta

Let $\delta(k) = [1, 0, 0, \ldots, 0]$.

Then $F(\delta)_k = e^0 = 1 \quad \forall k$.

Naive Discrete Fourier Transform complexity

$$A_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}$$ $\Leftarrow$ $O(N)$ time for computing 1 coefficient

N coefficients $A_0, \ldots, A_{N-1}$, so $O(N^2)$ time for naive DFT.

Fast-Fourier Transform

Consider $N = 2^q$, for $q \in \mathbb{N}$.

Then $\vec{A} = F(\vec{a})$ is given by $A_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}$.

Let $\vec{e} = (a_0, a_2, \ldots, a_{N-2})$

Let $\vec{o} = (a_1, a_3, \ldots, a_{N-1})$

$\vec{O} = F(\vec{o}) \quad \vec{E} = F(\vec{e})$

$$\Rightarrow A_k = \sum_{m=0}^{\frac{N}{2}-1} a_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{\frac{N}{2}-1} a_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}$$

$$A_k = \underbrace{\sum_{m=0}^{\frac{N}{2}-1} a_{2m} e^{-\frac{2\pi i}{N/2} mk}}_{\text{Fourier transform of even } a_{2m}} + e^{-\frac{2\pi i}{N}k} \underbrace{\sum_{m=0}^{\frac{N}{2}-1} a_{2m+1} e^{-\frac{2\pi i}{N/2} mk}}_{\text{Fourier transform of odd } a_{2m+1}} = E_k + e^{-\frac{2\pi i}{N}k} O_k$$

for $k = 0, \ldots, \frac{N}{2} - 1$

(because for larger k, no longer in range of $\frac{N}{2}$ DFT)

$$A_{k+\frac{N}{2}} = \sum_{m=0}^{\frac{N}{2}-1} a_{2m} e^{-\frac{2\pi i}{N/2} m(k+\frac{N}{2})} + e^{-\frac{2\pi i}{N}(k+\frac{N}{2})} \sum_{m=0}^{\frac{N}{2}-1} a_{2m+1} e^{-\frac{2\pi i}{N/2} m(k+\frac{N}{2})}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} a_{2m} e^{-\frac{2\pi i}{N/2} mk} + e^{-\frac{2\pi i}{N}k} \underbrace{e^{-\pi i}}_{=-1} \sum_{m=0}^{\frac{N}{2}-1} a_{2m+1} e^{-\frac{2\pi i}{N/2} mk}$$

$$= E_k - e^{-\frac{2\pi i}{N}k} O_k .$$

Thus:

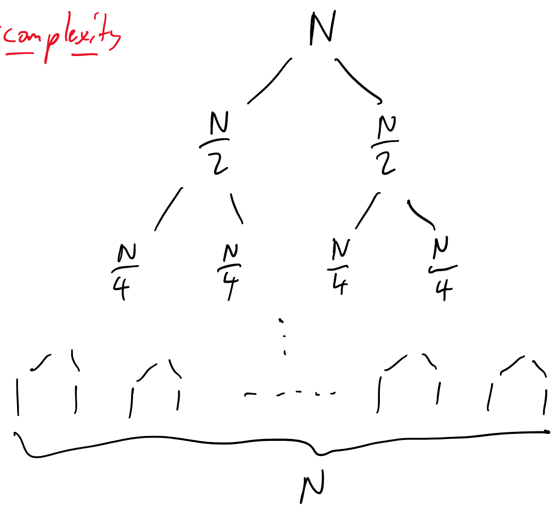
$$A_k = E_k + e^{-\frac{2\pi i}{N}k} O_k \quad \text{for } k = 0, \ldots, \frac{N}{2} - 1$$

$$A_{k+\frac{N}{2}} = E_k - e^{-\frac{2\pi i}{N}k} O_k \quad \text{for } k = 0, \ldots, \frac{N}{2} - 1.$$

Punchline: We can recursively compute the Fourier transform of a length N vector by computing two Fourier transforms of length $\frac{N}{2}$ vectors

Punchline: We can compute the Fourier transform of a length N vector
by computing two Fourier transforms of length $\frac{N}{2}$ vectors
Merging the two $\frac{N}{2}$ DFTs takes $O(N)$ time.

Time-complexity



$O(N)$ merging time

$O(N)$ merging time

$O(N)$ merging time

⋮

N length 1 DFTs $= N \cdot 1^2$ time

$\log_2(N)$ layers

$\Rightarrow O(N \log N)$ time.