Fast Fourier Transform

Vajradevam

I. Introduction to Complex Matrices

Matrices with real entries can have complex eigenvalues, making it necessary to work with complex numbers. The most important complex matrix is the Fourier matrix F_n, used for Fourier transforms. While normal multiplication by F_n requires n^2 multiplications, the Fast Fourier Transform (FFT) reduces this to roughly $n \log_2 n$ multiplications - a revolutionary improvement.

II. Complex Vectors

A. Length

For a complex vector $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \in \mathbb{C}^n$, the standard definition $\mathbf{z}^T \mathbf{z}$ is inadequate as it can be zero for nonzero vectors. The correct definition is:

$$|\mathbf{z}|^2 = \overline{\mathbf{z}}^T \mathbf{z} = |z_1|^2 + |z_2|^2 + \cdots + |z_n|^2$$

We write this as $|\mathbf{z}|^2 = \mathbf{z}^H \mathbf{z}$, where $\mathbf{z}^H = \overline{\mathbf{z}}^T$ (the Hermitian transpose).

B. Inner Product

The inner product of two complex vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$\mathbf{y}^H \mathbf{x} = \overline{\mathbf{y}}^T \mathbf{x} = \overline{y}_1 x_1 + \overline{y}_2 x_2 + \cdots + \overline{y}_n x_n$$

III. Complex Matrices

A. Hermitian Matrices

A complex matrix A is called *Hermitian* if $A^H = A$ (where $A^H = \overline{A}^T$). The diagonal entries of Hermitian matrices must be real. For example:

$$A = \begin{bmatrix} 2 & 3+i \\ 3-i & 5 \end{bmatrix}$$

Hermitian matrices have real eigenvalues and perpendicular eigenvectors.

B. Unitary Matrices

A collection of complex vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ is orthonormal if:

$$\overline{\mathbf{q}}_j^T \mathbf{q}_k = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

A *unitary matrix* $Q = [\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_n]$ satisfies $Q^H Q = I$, making it the complex analog of orthogonal matrices.

IV. Discrete Fourier Transform

The discrete Fourier transform decomposes finite data sets into frequency components. The Fourier matrix F_n is defined as:

$$F_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{n-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \cdots & w^{(n-1)^2} \end{bmatrix}$$

where $w = e^{i \cdot 2\pi/n}$ (so $w^n = 1$) and $(F_n)_{jk} = w^{jk}$ for $j, k = 0, 1, \ldots, n-1$.

A. Example: F_4

For $n = 4$, $w = e^{2\pi i/4} = i$, giving:

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

The columns are orthogonal but not orthonormal (each has length 2). The normalized matrix $\frac{1}{2} F_4$ is unitary.

V. Fast Fourier Transform

Fourier matrices can be decomposed efficiently. The key relationship between F_n and F_{2n} uses the fact that $w_{2n}^2 = w_n$:

$$F_{2n} = \begin{bmatrix} I & D \\ I & -D \end{bmatrix} \begin{bmatrix} F_n & 0 \\ 0 & F_n \end{bmatrix} P$$

where:

- D is a diagonal matrix: $D = \text{diag}(1, w, w^2, \ldots, w^{n-1})$
- P is a permutation matrix that separates even and odd components

This decomposition allows a size $2n$ Fourier transform to be computed using:

- Two size n Fourier transforms ($2n^2$ operations)
- Simple matrix multiplications ($\mathcal{O}(n)$ operations)

By recursively applying this decomposition, the computational complexity reduces from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$.

A. Example Efficiency

For $n = 1024 = 2^{10}$:

- Direct multiplication: $n^2 = 1{,}048{,}576$ operations
- FFT: $\frac{1}{2} n \log_2 n = 5{,}120$ operations
- Speedup: $\approx 200\times$ faster