Fast Fourier Transform

Matt McCarthy

Christopher Newport University

CPSC 621
November 30, 2015

Discrete Fourier Transform

Definition 1 (Discrete Fourier Transform)

Let $\mathbf{X} = (x_0, x_1, \ldots, x_{n-1}) \in \mathbb{C}^n$. Then the *Discrete Fourier Transform of* $\mathbf{X}$ is defined as $\mathbf{Y} = (y_0, y_1, \ldots, y_{n-1})$ where

$$y_j := \sum_{k=0}^{n-1} x_k \omega^{jk}$$

with $\omega = e^{2\pi i/n}$. Furthermore, we denote $\mathbf{Y} = \mathcal{F}(\mathbf{X})$.

Discrete Fourier Transform

Definition 1 (Discrete Fourier Transform)

Let $\mathbf{X} = (x_0, x_1, \ldots, x_{n-1}) \in \mathbb{C}^n$. Then the *Discrete Fourier Transform of* $\mathbf{X}$ is defined as $\mathbf{Y} = (y_0, y_1, \ldots, y_{n-1})$ where

$$y_j := \sum_{k=0}^{n-1} x_k \omega^{jk}$$

with $\omega = e^{2\pi i/n}$. Furthermore, we denote $\mathbf{Y} = \mathcal{F}(\mathbf{X})$.

Complexity: $\Theta(n^2)$.

Slightly Faster Fourier Transform

If we assume n is even, then by symbol pushing we get

$$
\begin{aligned}
y_j &= \sum_{k=0}^{n/2-1} x_{2k}\omega^{(2k)j} + \sum_{k=0}^{n/2-1} x_{2k+1}\omega^{(2k+1)j} \\
&= \sum_{k=0}^{n/2-1} x_{2k} e^{2(2\pi i/n)jk} + \sum_{k=0}^{n/2-1} x_{2k+1}\omega^j e^{2(2\pi i/n)jk} \\
&= \sum_{k=0}^{n/2-1} x_{2k} \left(e^{2\pi i/n}\right)^{2jk} + \omega^j \sum_{k=0}^{n/2-1} x_{2k+1} \left(e^{2\pi i/n}\right)^{2jk}
\end{aligned}
$$

Slightly Faster Fourier Transform

If we assume n is even, then by symbol pushing we get

$$
\begin{aligned}
y_j &= \sum_{k=0}^{n/2-1} x_{2k}\omega^{(2k)j} + \sum_{k=0}^{n/2-1} x_{2k+1}\omega^{(2k+1)j} \\
&= \sum_{k=0}^{n/2-1} x_{2k}e^{2(2\pi i/n)jk} + \sum_{k=0}^{n/2-1} x_{2k+1}\omega^{j}e^{2(2\pi i/n)jk} \\
&= \sum_{k=0}^{n/2-1} x_{2k}\left(e^{2\pi i/n}\right)^{2jk} + \omega^{j}\sum_{k=0}^{n/2-1} x_{2k+1}\left(e^{2\pi i/n}\right)^{2jk}
\end{aligned}
$$

Let $\tilde{\omega} = \omega^2$ and we have

$$
y_j = \sum_{k=0}^{n/2-1} x_{2k}\tilde{\omega}^{jk} + \omega^{j}\sum_{k=0}^{n/2-1} x_{2k+1}\tilde{\omega}^{jk}.
$$

If $n = 2^k$ for some $k \in \mathbb{Z}^+$, then we can iterate this process using the following algorithm.

Fast Fourier Transform

If $n = 2^k$ for some $k \in \mathbb{Z}^+$, then we can iterate this process using the following algorithm.

The one-dimensional, unordered, radix 2, FFT algorithm.

function $\mathrm{R\text{-}FFT}(\mathbf{X},\mathbf{Y},n,\omega)$
    if $n{=}1$ **then**
        $y_0 = x_0$
    else
        Let $\mathbf{Q} = \mathbf{0}, \mathbf{T} = \mathbf{0} \in \mathbb{C}^n$
        $\mathrm{R\text{-}FFT}((x_0, x_2, \ldots, x_{n-2}),(q_0, q_2, \ldots, q_{n-2}),n/2,\omega^2)$
        $\mathrm{R\text{-}FFT}((x_1, x_3, \ldots, x_{n-1}),(t_1, t_3, \ldots, t_{n-1}),n/2,\omega^2)$
        for all $j \in \{0, 1, \ldots, n-1\}$ **do**
            $y_j = q_{j \mod n/2} + \omega^i t_{j \mod n/2}$
        end for
    end if
end function

Fast Fourier Transform: Serial Analysis

- Since $n = 2^k$, we do $\lg n = k$ steps
- At the mth level of recursion we do 2^m FFTs of size $n/2^m$
  - Each level is $\Theta(n)$
- Thus, FFT is $\Theta(n \lg n)$.

Iterative Formulation

function $\text{I-FFT}(\mathbf{X},\mathbf{Y},n)$
    $t := \lg n$
    $\mathbf{R} = \mathbf{X}$
    for $m = 0$ to $t-1$ **do**
        $\mathbf{S} = \mathbf{R}$
        for $l = 0$ to $n-1$ **do**
            Let $(b_0 b_1 \ldots b_{t-1})$ be the binary expansion of l
            $j := (b_0 \ldots b_{m-1} 0 b_{m+1} \ldots b_{t-1})$
            $k := (b_0 \ldots b_{m-1} 1 b_{m+1} \ldots b_{t-1})$
            $r_i := s_j + s_k \omega^{(b_m b_{m-1} \ldots b_0 0 \ldots 0)}$
        end for
    end for
    $\mathbf{Y} := \mathbf{R}$
end function

References I

Author
Title
where

Thank you!