Complexity of Filtering and the FFT

Dr. Deepa Kundur
University of Toronto

Digital Filtering in the Time Domain

Let $x(n)$ and $h(n)$ be real signals.

Let the support of $x(n)$ be $n = 0, 1, \ldots, N - 1$. We are interested in determining $y(n)$ for $n = 0, 1, \ldots, N - 1$.

\[
y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)
\]

\[
= \sum_{k=0}^{N-1} x(k)h(n-k)
\]

Complexity of doing a brute-force convolution is given by:

- For fixed $n$:
  \[
y(n) = \sum_{k=0}^{N-1} x(k)h(n-k)
  \]
  - $N$ real multiplications
  - $N - 1$ real additions

- For all $n$ ($n=0, 1, \ldots, N-1$):
  - $N \cdot N = N^2 = O(N^2)$ real multiplications
  - $(N - 1) \cdot N = N(N - 1) = O(N^2)$ real additions

Is $O(N^2)$ high?

- Yes.

Idea: Maybe filtering in the frequency domain can reduce complexity.
Discrete Fourier Transform (DFT)

- Frequency analysis of discrete-time signals is conveniently performed on a DSP.
- Therefore, both time-domain and frequency-domain signals must be discrete.
  - $x(t) \xrightarrow{\text{sampling}} x(n)$
  - $X(\omega) \xrightarrow{\text{sampling}} X(\frac{2\pi k}{N})$ or $X(k)$
- What happens when we sample in the frequency domain?

Fourier Duality

<table>
<thead>
<tr>
<th>Time Domain</th>
<th>Frequency Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>sinc</td>
<td>rectangle</td>
</tr>
<tr>
<td>rectangle</td>
<td>sinc</td>
</tr>
<tr>
<td>sinc$^2$</td>
<td>triangle</td>
</tr>
<tr>
<td>triangle</td>
<td>sinc$^2$</td>
</tr>
<tr>
<td>ringing</td>
<td>truncation</td>
</tr>
<tr>
<td>truncation</td>
<td>ringing</td>
</tr>
<tr>
<td>discrete</td>
<td>periodic</td>
</tr>
<tr>
<td>periodic</td>
<td>discrete</td>
</tr>
<tr>
<td>continuous</td>
<td>aperiodic</td>
</tr>
<tr>
<td>aperiodic</td>
<td>continuous</td>
</tr>
</tbody>
</table>

DFT Intuition

- aperiodic + dst in time $\xleftarrow{\mathcal{F}}$ cts + periodic in freq
- periodic + dst in time $\xleftarrow{\mathcal{F}}$ dst + periodic in freq
- periodic + discrete $\xrightarrow{\text{DTFS}}$ periodic + discrete

- one period of dst samples $\longleftrightarrow$ one period of dst samples
  - $n = 0, 1, \ldots, N - 1$
  - $k = 0, 1, \ldots, N - 1$

Example

- periodic + dst in time-domain
- periodic + dst in freq-domain

note: signal examples are artificial
Frequency Domain Sampling

- Recall, sampling in time results in a periodic repetition in frequency.
  \[ x(n) = x_a(t)_{|t=nT} \iff X(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a(\omega + \frac{2\pi}{T}k) \]

- Similarly, sampling in frequency results in periodic repetition in time.
  \[ x_p(n) = \sum_{l=-\infty}^{\infty} x(n + lN) \iff X(k) = X(\omega)_{|\omega = \frac{2\pi}{N}k} \]

Frequency Domain Sampling and Reconstruction

- Therefore,
  \[ x(n) \iff X(\omega) \]
  \[ x_p(n) \iff X(k) \]

- Implications:
  - The samples of \( X(\omega) \) can be used to reconstruct \( x_p(n) \).

Q: Can we reconstruct \( x(n) \) from the samples of \( X(\omega) \)?
  - Can we reconstruct \( x(n) \) from \( x_p(n) \)?
A: Maybe.
  \[ x_p(n) = \left[ \sum_{l=-\infty}^{\infty} x(n + lN) \right] \]
**Frequency Domain Sampling and Reconstruction**

*Complexity of Filtering and the FFT*

**DTFT, DTFS and DFT**

- \( x(n) \) for all \( n \) \( \xrightarrow{\text{DTFT}} \) \( X(\omega) \) for all \( \omega \)
- \( x_p(n) \) for all \( n \) \( \xrightarrow{\text{DTFS}} \) \( X(k) \) for all \( k \)
- \( \hat{x}(n) \) \( \xrightarrow{\text{DFT}} \) \( \hat{X}(k) \)

where

\[
\hat{x}(n) = \begin{cases} 
  x_p(n) & \text{for } n = 0, \ldots, N-1 \\
  0 & \text{otherwise}
\end{cases}
\]

and

\[
\hat{X}(k) = \begin{cases} 
  X(k) & \text{for } n = 0, \ldots, N-1 \\
  0 & \text{otherwise}
\end{cases}
\]

**The Discrete Fourier Transform Pair**

- **DFT and inverse-DFT (IDFT):**
  \[
  X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \quad k = 0, 1, \ldots, N-1
  \]
  \[
  x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N-1
  \]

Note: we drop the \( \hat{\cdot} \) notation from now on.
Important DFT Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Time Domain</th>
<th>Frequency Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation: ( x(n) )</td>
<td>( X(k) )</td>
<td></td>
</tr>
<tr>
<td>Periodicity: ( x(n) = x(n + N) )</td>
<td>( X(k) = X(k + N) )</td>
<td></td>
</tr>
<tr>
<td>Linearity: ( a_1 x_1(n) + a_2 x_2(n) )</td>
<td>( a_1 X_1(k) + a_2 X_2(k) )</td>
<td></td>
</tr>
<tr>
<td>Time reversal: ( x(N - n) )</td>
<td>( X(N - k) )</td>
<td></td>
</tr>
<tr>
<td>Circular time shift: ( x((n - l)N) )</td>
<td>( X(k)e^{-j2\pi k/N} )</td>
<td></td>
</tr>
<tr>
<td>Circular frequency shift: ( x(n)e^{j2\pi n/N} )</td>
<td>( X((k - l)N) )</td>
<td></td>
</tr>
<tr>
<td>Complex conjugate: ( x^*(n) )</td>
<td>( X^*(N - k) )</td>
<td></td>
</tr>
<tr>
<td>Circular convolution: ( x_1(n) \otimes x_2(n) )</td>
<td>( X_1(k)X_2(k) )</td>
<td></td>
</tr>
<tr>
<td>Multiplication: ( x_1(n)x_2(n) )</td>
<td>( \frac{1}{N}X_1(k) \otimes X_2(k) )</td>
<td></td>
</tr>
<tr>
<td>Parseval’s theorem: ( \sum_{n=0}^{N-1} x(n)x^*(n) )</td>
<td>( \frac{1}{N} \sum_{k=0}^{N-1} X(k)X^*(k) )</td>
<td></td>
</tr>
</tbody>
</table>

Complexity of the DFT

\[
X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N - 1
\]

Straightforward implementation of DFT to compute \( X(k) \) for \( k = 0, 1, \ldots, N - 1 \) requires:

- \( N^2 \) complex multiplications
  - 1 complex mult = \( (a_R + ja_I) \times (b_R + jb_I) = (a_R \times b_R - a_I \times b_I) + j(a_R \times b_I + a_I \times b_R) \)
    - 4 real mult + 2 real add
  - \( 4N^2 = O(N^2) \) real multiplications

Complexity of the DFT (and IDFT)

\[
x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N - 1
\]

New notation: \( W_N = e^{-j2\pi/N} \)

\[
X(k) = \sum_{n=0}^{N-1} x(n)W_n^{kn}, \quad k = 0, 1, \ldots, N - 1
\]

\[
x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_n^{-kn}, \quad n = 0, 1, \ldots, N - 1
\]
Complexity of the DFT

- Is $O(N^2)$ high?
  - Yes. A linear increase in the length of the DFT increases the complexity by a power of two.
  - Given the multitude of applications where Fourier analysis is employed (linear filtering, correlation analysis, spectrum analysis), a method of efficient computation is needed.

---

Complexity of the DFT

- How can we reduce complexity?
  - Exploit **symmetry** of the complex exponential.

\[
W_N^{k+N} = W_N^k \\
LHS = W_N^{k+N} = e^{-j2\pi \frac{k+N}{N}} = e^{-j2\pi \frac{k}{N}} e^{-j2\pi} \\
= e^{-j2\pi \frac{k}{N}} e^{-j2\pi} = e^{-j2\pi \frac{k}{N}} (\cos(-\pi) + j \sin(-\pi)) \\
= e^{-j2\pi \frac{k}{N}} (-1) \\
= -e^{-j2\pi \frac{k}{N}} = -W_N^k = RHS
\]

---

Complexity of the DFT

- How can we reduce complexity?
  - Exploit **periodicity** of the complex exponential.

\[
X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \\
= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn} \\
= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{k(2m)} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)} \\
= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2km} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{2km} W_N^k \\
\equiv f_1(m) + f_2(m)
\]

Radix-2 FFT: Decimation-in-time

\[
X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \\
= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn} \\
= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{k(2m)} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)} \\
= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2km} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{2km} W_N^k \\
\equiv f_1(m) + f_2(m)
\]
Radix-2 FFT: Decimation-in-time

Note: $W_N^2 = e^{-j \frac{2\pi}{N}} = e^{-j \frac{2\pi}{N_2}} = W_{N/2}$

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2km} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{2km} W_N^k = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km}$$

$$= F_1(k) + W_N^k F_2(k), \quad k = 0, 1, \ldots, N - 1$$

Repeating the decimation-in-time for $f_1(n)$ and $f_2(n)$, we obtain:

$$X(k) = F_1(k) + W_N^k F_2(k) \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k) \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Note: since $F_1(k)$ and $F_2(k)$ are $\frac{N}{2}$-DFTs:

$$F_1(k) = F_1(k + \frac{N}{2})$$

$$F_2(k) = F_2(k + \frac{N}{2})$$

we have,

$$X(k) = F_1(k) + W_N^k F_2(k)$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) + W_N^{k+N} F_2(k)$$

since $W_N^{k+N} = e^{-j \frac{2\pi}{N} (k+N)} = e^{-j \frac{2\pi}{N} k} \cdot e^{-j \frac{2\pi}{N} N} = e^{-j \frac{2\pi}{N} k} (-1) = -W_N^k$

Consisting of $N/4$-DFTs.
Radix-2 FFT: Decimation-in-time

For $N = 8$.

- Combine 4-point DFTs
- Combine 2-point DFTs

FFT Complexity

- Each butterfly requires:
  - one complex multiplication
  - two complex additions

- In total, there are:
  - $\frac{N}{2}$ butterflies per stage
  - $\log N$ stages

- Order of the overall DFT computation is: $O(N \log N)$

Convolution using FFT

To compute the convolution of $x(n)$ (support: $n = 0, 1, \ldots, L - 1$) and $h(n)$ (support: $n = 0, 1, \ldots, M - 1$):

1. Assign $N$ to be the smallest power of 2 such that $N = 2^r \geq M + L - 1$.
2. Zero pad both $x(n)$ and $h(n)$ to have support $n = 0, 1, \ldots, N - 1$.
3. Take the $N$-FFT of $x(n)$ to give $X(k)$, $k = 0, 1, \ldots, N - 1$.
4. Take the $N$-FFT of $h(n)$ to give $H(k)$, $k = 0, 1, \ldots, N - 1$.
5. Produce $Y(k) = X(k) \cdot H(k)$, $k = 0, 1, \ldots, N - 1$.
6. Take the $N$-IFFT of $Y(k)$ to give $y(n)$, $n = 0, 1, \ldots, N - 1$. 
**Convolution using FFT**

To compute the convolution of $x(n)$ (support: $n = 0, 1, \ldots, L - 1$) and $h(n)$ (support: $n = 0, 1, \ldots, M - 1$):

1. Assign $N$ to be the smallest power of 2 such that $N = 2^r \geq M + L - 1$.
2. Zero pad both $x(n)$ and $h(n)$ to have support $n = 0, 1, \ldots, N - 1$.
3. Take the $N$-FFT of $x(n)$ to give $X(k)$, $k = 0, 1, \ldots, N - 1$.
4. Take the $N$-FFT of $h(n)$ to give $H(k)$, $k = 0, 1, \ldots, N - 1$.
5. Produce $Y(k) = X(k) \cdot H(k)$, $k = 0, 1, \ldots, N - 1$.
6. Take the $N$-IFFT of $Y(k)$ to give $y(n)$, $n = 0, 1, \ldots, N - 1$.

**Complexity of Convolution using FFT**

Therefore, the overall complexity of conducting convolution via the FFT is:

$$O(N \log N)$$

which is lower than $O(N^2)$ through direction computation of convolution in the time-domain.