Automatic Speech Recognition
Lecture Note 3: Discrete Fourier Transform

1. **Discrete Fourier Transform**

   The discrete Fourier transform (DFT) of a finite-duration sequence $x[n], 0 \leq n \leq N-1$, is defined as

   $$X[k] = \sum_{n=0}^{N-1} x[n] W^{nk},$$

   where $W = e^{-j(2\pi/N)}$. The *inverse* discrete Fourier transform (IDFT) is given by

   $$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W^{-kn}.$$

   Note that the DFT computation yields samples of the spectrum of a finite sequence at the points $\omega_k = \frac{2\pi k}{N}$.

2. **Periodicity of DFT and IDFT**

   The DFT is periodic in k with period N. The IDFT is periodic in n with period N and each period is identical to the original sequence.

3. **Convolution Theorem of DFT**

   Let $x_1[n]$ and $x_2[n]$ be N-point finite duration sequences. Let $X_1[k]$ and $X_2[k]$ be their DFTs. Let $X[k] = X_1[k]X_2[k]$. Then the IDFT of $X[k]$, say $x[n]$, corresponds to the *circular (or periodic)* convolution of $x_1[n]$ and $x_2[n]$, defined by

   $$x[n] = \sum_{m=0}^{N-1} \tilde{x}_1[m] \tilde{x}_2[n-m],$$

   where $\tilde{x}_i[n]$ is the periodic extension of $x_i[n]$, i.e.,

   $$\tilde{x}_i[n] = x_i[n \mod N].$$

proof: The DFT, say $Y[k]$, of $x[n]$ as defined above is

$$
\begin{aligned}
Y[k] &= \sum_{n=0}^{N-1} x[n]W^{nk} = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m]W^{nk} \\
&= \sum_{m=0}^{N-1} \tilde{x}_1[m] \sum_{n=0}^{N-1} \tilde{x}_2[n-m]W^{nk} \\
&= \sum_{m=0}^{N-1} \tilde{x}_1[m]W^{km} \sum_{n=0}^{N-1} \tilde{x}_2[n-m]W^{k(n-m)} \\
&= \cdots \sum_{r=-m}^{N-1-m} \tilde{x}_2[r]W^{kr} \\
&= \ldots \Big(\sum_{r=-m}^{-1} + \sum_{r=0}^{N-1-m} \Big) \; \tilde{x}_2[r]W^{kr} \\
&= \ldots \Big(\sum_{r=N-m}^{N-1} + \sum_{r=0}^{N-1-m} \Big) \; \tilde{x}_2[r]W^{kr} \\
&= \cdots \sum_{r=0}^{N-1} x_2[r]W^{kr} \\
&= \sum_{m=0}^{N-1} \tilde{x}_1[m]W^{km}X_2[k] \\
&= \sum_{m=0}^{N-1} x_1[m]W^{km}X_2[k] \\
&= X_1[k]X_2[k].
\end{aligned}
$$

4. **FIR Filter Implementation via DFT**

   The convolution of two finite-duration sequences with lengths N_1, N_2 is finite-duration with length $N_1 + N_2 - 1$. This can be implemented by DFT by augmenting the original sequences to a length $\geq N_1 + N_2 - 1$, and then compute the IDFT of the product of DFTs. This is practical because the DFT computation can be very fast.

5. **Fast Fourier Transform**

   The Fast Fourier Transform (FFT) is nothing more than a fast way to compute DFT. Consider an N-point sequence where $N = N_1N_2$.

The two-dimensional DFT of a matrix $N_1 \times N_2$ is computationally less expensive than the one-dimensional N-point DFT. Exploiting this idea, the DFT of $x[n], 0 \le n \le N-1$ can be computed by the following steps:

- Let $N = LM$, and put $x[n]$ into the (l, m)-th (index starting from 0) element of the $L \times M$ matrix with $n = Ml + m$.
- Compute the L-point DFT of each column with kernel W^M. This results in a new matrix $q(s, m)$.
- Multiply each element in $q(s, m)$ by W^{ms} where $W = e^{-j(2\pi/N)}$. Call the new matrix $h(s, m)$.
- Compute the DFT of each row of h, with W^L as the kernel. Call the final matrix $X(s, r)$.

Then $X[k] = X(s, r)$ where $k = Lr + s$ (index starting from 0). With such implementation, the time complexity of FFT is $O(N \log N)$.