ECE 310 Fall 2018 Recitation 8.;

October 29, 2018

Convolution with the DFT

Circular Convolution

Multiplication in the time domain does not correspond to linear convolution in the discrete frequency domain (the domain after taking the DFT). Instead, because signals are periodic, multiplication corresponds to circular convolution. Assume we have two signals $x_1[n] = \{1, 2, 3, 4\}$ and $h_1 = \{5, 6, 7, 8\}$. To calculate a familiar linear convolution, we would perform the process as follows:

$$
\begin{array}{cccccccc}
 & & & 1 & 2 & 3 & 4 & \\
8 & 7 & 6 & 5 & & & & \rightarrow 5 \\
 & 8 & 7 & 6 & 5 & & & \rightarrow 6 + 10 = 16 \\
 & & & & & & & \vdots
\end{array}
$$

But because we assume the signals are periodic to work with the DFT, multiplication in the frequency domain $X_1[k]\, H_1[k]$ corresponds to a circular convolution instead:

$$
\begin{array}{cccccccc}
 & & & 1 & 2 & 3 & 4 & \\
8 & 7 & 6 & 5 & 8 & 7 & 6 & \rightarrow 5 + 16 + 21 + 24 = 66 \\
5 & 8 & 7 & 6 & 5 & 8 & 7 & \rightarrow 6 + 10 + 24 + 28 = 68 \\
6 & 5 & 8 & 7 & 6 & 5 & 8 & \rightarrow 7 + 12 + 15 + 32 = 66 \\
7 & 6 & 5 & 8 & 7 & 6 & 5 & \rightarrow 8 + 14 + 18 + 20 = 60
\end{array}
$$

The blue text indicates the changes from linear convolution due to the assumed periodicity of the signals. We can see that the next step would be the same as the first, so we stop here, having fully defined one period of the result of the convolution. The circular convolution of 2 signals of length N is also length N and periodic.

Linear Convolution

Generally, we are interested in performing the linear convolution between to signals, since this is useful for determining the output of LTI systems, not the circular convolution. But, computers are limited to using the DFT, which leads to circular convolution. To get around this, we perform zero padding when working with finite duration signals. Since we know a linear convolution result has length $L+M-1$, we zero pad both signals to this length, so their circular convolution will have the appropriate length: $x_2[n] = \{1,2,3,4,0,0,0\}$ and $h_2[n] = \{5,6,7,8,0,0,0\}$. Now the circular convolution looks like:

$$
\begin{array}{ccccccccccccccl}
 & & & & & & 1 & 2 & 3 & 4 & 0 & 0 & 0 & \\
0 & 0 & 0 & 8 & 7 & 6 & 5 & 0 & 0 & 0 & 8 & 7 & 6 & \rightarrow 5 \\
5 & 0 & 0 & 0 & 8 & 7 & 6 & 5 & 0 & 0 & 0 & 8 & 7 & \rightarrow 6+10=16 \\
6 & 5 & 0 & 0 & 0 & 8 & 7 & 6 & 5 & 0 & 0 & 0 & 8 & \rightarrow 7+12+15=34 \\
7 & 6 & 5 & 0 & 0 & 0 & 8 & 7 & 6 & 5 & 0 & 0 & 0 & \rightarrow 8+14+18+20=60 \\
 & & & & & & \vdots & & & & & & &
\end{array}
$$

We can see that one period of this circular convolution of x_2 and h_2 will correspond to the linear convolution of the signals x_1 and h_1. Thus we can use $X_2[k]\,H_2[k]$ to effectively perform a linear convolution of x_1 and h_1.

Other Effects of Padding

In addition to allowing the computation of linear convolution, zero padding (or otherwise changing the number of samples in the signal) also affects the granularity of the DFT. We know that the length N of the signal determines the length of the DFT, and that the N points of the DFT sample the DTFT between 0 and 2π with the relationship $\omega = \frac{2\pi}{N}k$. For finite length input, we can simply zero pad to increase the length of the DFT, thereby increasing the resolution with which we sample the DTFT. For infinite length input, we must instead increase the number of samples taken from the analog signal.

Decimation

Decimation is the process of reducing sampling rate of a signal. It is often used to reduce the amount of computation to be done when processing a signal. This is why it is employed in the FFT algorithm. The FFT is frequently implemented using decimation by 2, though it is possible to perform decimation by other factors. Here we look at the FFT computed by decimation in frequency of the signal $x[n]$ with length N that we will assume is a power of 2. We will calculate its DFT $X[k]$ by separately calculating the even and odd indices. The even indices are represented:

$$X[2k]\,, k = 0, 1, \ldots, \frac{N}{2} - 1$$

And the odd indices are represented:

$$X[2k+1]\,, k = 0, 1, \ldots, \frac{N}{2} - 1$$

We use the definition of the DFT to represent the even indices:

$$\begin{aligned} X[2k] &= \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}2kn} \\ &= \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} \end{aligned}$$

We split the summation into two separate summations (a common technique when working with decimation), one of the first half of $x[n]$ and one of the second half:

$$\begin{aligned} X[2k] &= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} + \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] e^{-j\frac{2\pi}{\frac{N}{2}}k\left(n+\frac{N}{2}\right)} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} + \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j\frac{2\pi}{\frac{N}{2}}k\frac{N}{2}} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} + \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j2\pi k} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} + \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] e^{-j\frac{2\pi}{\frac{N}{2}}kn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] + x\left[n+\frac{N}{2}\right]\right) e^{-j\frac{2\pi}{\frac{N}{2}}kn} \end{aligned}$$

A similar process of writing out the definition, splitting the sum, and recombining gives an expression for the odd indices of the original $X[k]$:

$$
\begin{aligned}
X[2k+1] &= \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}(2k+1)n} \\
&= \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}2kn} e^{-j\frac{2\pi}{N}n} \\
&= \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j\frac{2\pi}{N}n} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j\frac{2\pi}{N}n} + \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] e^{-j\frac{2\pi}{\frac{N}{2}}k\left(n+\frac{N}{2}\right)} e^{-j\frac{2\pi}{N}\left(n+\frac{N}{2}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j\frac{2\pi}{N}n} + \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j2\pi k} e^{-j\frac{2\pi}{N}n} e^{-j\pi} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j\frac{2\pi}{N}n} - \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j\frac{2\pi}{N}n} \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] - x\left[n+\frac{N}{2}\right]\right) e^{-j\frac{2\pi}{\frac{N}{2}}kn} e^{-j\frac{2\pi}{N}n}
\end{aligned}
$$

We have shown that a DFT of even length N can be broken into two smaller DFTs of length $\frac{N}{2}$, which can then be re-assembled into the original DFT by interleaving. If N is chosen to be a power of 2, this splitting can be performed recursively until the calculation of the sums is trivial ($N = 1$). For an example, the process is applied to an 8-point DFT of $x[n]$. We define 2 sequences $a[n]$ and $b[n]$ which correspond to the 4-point DFTs that $X[k]$ will be divided into:

$$
\begin{aligned}
a[n] &= x[n] + x\left[n+\frac{8}{2}\right] = x[n] + x[n+4] \\
b[n] &= \left(x[n] - x\left[n+\frac{8}{2}\right]\right) W_8^n = (x[n] - x[n+4])\, W_8^n
\end{aligned}
$$

DFTs of $a[n]$ and $b[n]$ will have the relationships:

$$
\begin{aligned}
X[0] &= A[0] \\
X[1] &= B[0] \\
X[2] &= A[1] \\
X[3] &= B[1] \\
X[4] &= A[2] \\
X[5] &= B[2] \\
X[6] &= A[3] \\
X[7] &= B[3]
\end{aligned}
$$

We divide the 4 point DFT $A[k]$ into 2 point DFTs $C[k]$ and $D[k]$ $B[k]$ is divided into $E[k]$ and $F[k]$. The corresponding sequences are defined below:

$$
\begin{aligned}
c[n] &= a[n] + a\left[n + \frac{4}{2}\right] = a[n] + a[n+2] \\
d[n] &= \left(a[n] - a\left[n + \frac{4}{2}\right]\right) W_4^n = \left(a[n] - a[n+2]\right) W_4^n \\
e[n] &= b[n] + b\left[n + \frac{4}{2}\right] = b[n] + b[n+2] \\
f[n] &= \left(b[n] - b\left[n + \frac{4}{2}\right]\right) W_4^n = \left(b[n] - b[n+2]\right) W_4^n
\end{aligned}
$$

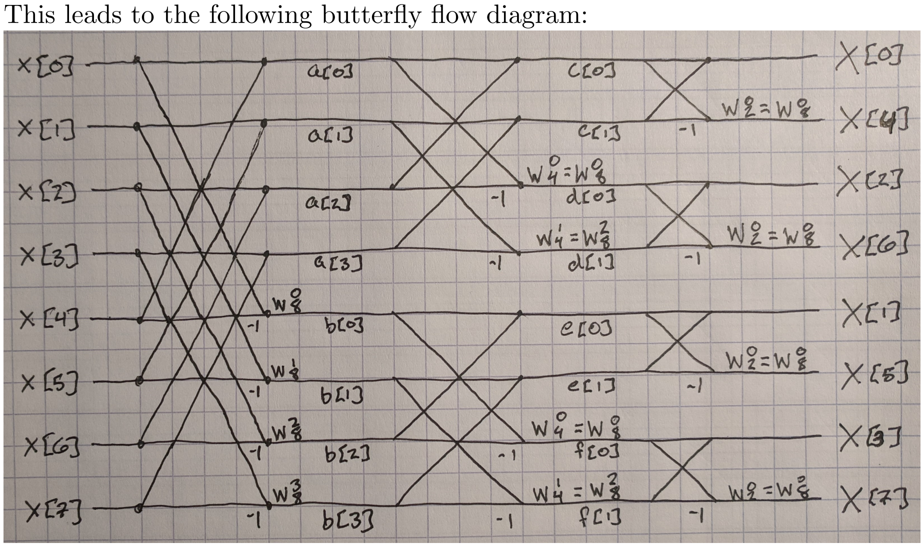
The DFTs now have the following replationships:

$$
\begin{aligned}
X[0] &= A[0] = C[0] \\
X[1] &= B[0] = E[0] \\
X[2] &= A[1] = D[0] \\
X[3] &= B[1] = F[0] \\
X[4] &= A[2] = C[1] \\
X[5] &= B[2] = E[1] \\
X[6] &= A[3] = D[1] \\
X[7] &= B[3] = F[1]
\end{aligned}
$$

The 2 point DFTs can be divided into 1 point DFTs, for which calculation is trivial because only $n = 0$ must be evaluated. This will result in the following relationships:

$$
\begin{aligned}
C[0] &= c[0] + c[1] \\
C[1] &= (c[0] - c[1])\, W_2^0 \\
D[0] &= d[0] + d[1] \\
D[1] &= (d[0] - d[1])\, W_2^0 \\
E[0] &= e[0] + e[1] \\
E[1] &= (e[0] - e[1])\, W_2^0 \\
F[0] &= f[0] + f[1] \\
F[1] &= (f[0] - f[1])\, W_2^0
\end{aligned}
$$

This leads to the following butterfly flow diagram:



Twiddle factors are typically rewritten with the base for the largest DFT being computed so it is more obvious that they can be re-used in computation. Twiddle factors of multiple bases are included in this diagram to make the pattern of their use easier to see.