ECE 438 lecture 2/27/2023

Announcements

- Office Hours today!

  11:30a – 12:30p EST

  4:00p – 5:00p EST

- Quiz No. 2
  available at 6:00p EST today
  you have 30 minutes to work your solution and upload to Gradescope by 11:59p EST
  will cover ZT vs DFT or FFT

Continue with development of FFT algorithm

Completed an example for $N = 8 = 2^3$
(radix 2 FFT)

Computation

$C_{DIRECT} = N^2$ C.O. — 1 C.O. = 1 complex multiply + 1 complex addition

$C_{FFT} = N \log_2 N$ C.O.

N	C_{DIRECT}	$\log_2(N)$	C_{FFT}	$\eta = \frac{C_{DIRECT}}{C_{FFT}}$ (gain)
2	4	1	2	2
16	256	4	64	4

1024	16,384	10	10,240	102.54
$(1024)^2$	2^{20}	20	10×2^{10}	$109.4 \times 10^6 \cong 10^3$

Ordering of input data $N = 8$

Binary output	Output order of Stage 3	Input to Stage 2	Input to Stage 1	Binary output	Decimal ordering of inputs
000	0	$x[0]$	$x[0]$	000	0
001	1	$x[2]$	$x[4]$	100	4
010	2	$x[4]$	$x[2]$	010	2
011	3	$x[6]$	$x[6]$	110	6
100	4	$x[1]$	$x[1]$	001	1
101	5	$x[3]$	$x[5]$	101	5
110	6	$x[5]$	$x[3]$	011	3
111	7	$x[7]$	$x[7]$	111	7

Output ordering is bit-reversed relative to the input

Closing remarks for FFT

1. What we have derived is the decimation-in-time FFT for radix 2
2. There is also a decimation in frequency algorithm

(flip flow diagram for DIT to get DIF $\Rightarrow$ input ordering is bit reversed. output order is normal

There is more!

③ Can derive FFT algorithms for any highly composite N, i.e.

$$N = N_0 \cdot N_1 \cdot N_2 \cdots N_{M-1}$$

called mixed radix algorithm

④ S. Winograd developed FFT algorithms based on "small convolutions"

⑤ Very general and efficient FFT algorithms exist in Matlab and Numpy library

---

Application of the FFT: Efficient convolution

Filter an N-length signal with an M-length filter, where the filter is FIR $M << N$

We have $y[n] = \sum_{m=0}^{N-1} x[m]\, \underbrace{h[n-m]}_{h[-(m-n)]}$

How much computation is needed?

$$C_{DIRECT} = MN \text{ c.o.}$$

Now let's see what the FFT would require:

① $x[n] \xrightarrow{N\text{-pt DFT}} X^{(N)}[k],\ k=0,\ldots,N-1$

$N \log_2 N$ c.o.'s

② $h[n] \xrightarrow{N\text{-pt. DFT}} H^{(N)}[k],\ k=0,\ldots,N-1$

$N \log_2 N$ c.o.'s

note: we have zero-padded the filter to be length N

③ $Y^{(N)}[k] = H^{(N)}[k]\,X^{(N)}[k], \quad k = 0, \ldots, N-1$

$\underline{\frac{N}{2} \text{ c.o.'s}}$

④ $y[n] \xleftarrow{N\text{ pt. } DFT^{-1}} Y^{(N)}[k], \quad k = 0, \ldots, N-1$

$\underline{N \log_2 N \text{ c.o.}}$

Total

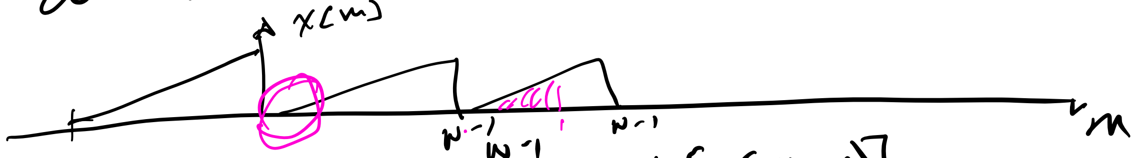
$$C_{FFT} = 3N\log_2(N) + \frac{N}{2}$$

gain

$$\eta = \frac{C_{DIRECT}}{C_{FFT}} = \frac{MN}{3N\log_2(N) + \frac{N}{2}} = \frac{M}{3\log_2(N) + \frac{1}{2}}$$

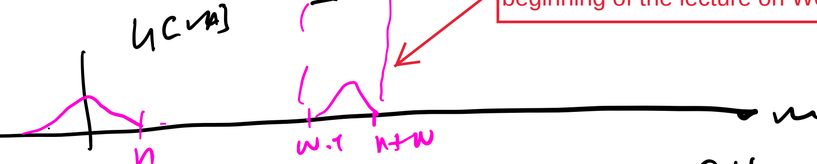
does this work?

Problem: DFT assumes that signals & spectra are periodic with period N



remember: $y[n] = \sum_{m=0} x[m]\,h[-(m-n)]$

This drawing is not correct. I will fix it at the beginning of the lecture on Wednesday 1 March.



Solution: zero-pad both signal $x[m]$ and filter $h[m]$ to length $N+M-1$

First office hour: 2:30p EST

Second office hour: 4:00p EST