HW5 R10945004 孫欽鉉

1. Code 流程說明 (code 裡面也有寫 comment 助教也可從裡面確認)
   Step1 因為 alpha 的可能值為 2~M，所以利用迴圈 check 每一個 value，找出所有可能的 alpha 值
   Step2 找出所有的可能 alpha 值後，使用最小的 alpha 製作一個 mod table, a^1,a^2…..,a^N，利於我們後面 NTT 的計算
   Step3 計算 NTT 矩陣(N by N) 第一個 row 及第一個 column 直接全部填 1，如果 a^nk 的 nk 介在 (0,N) 之間 則可以直接填入我們 Step2 所算好的 value，若 nk > N 則先取 mod N 之後(必定介在 (0,N))在填入 Step2 所算好的 value，計算完成則得到所需要的 forward NTT matrix
   Step4 透過前面算出的最小 alpha 計算 inverse alpha，接著也計算 inverse N
   Step5 步驟如同 Step2，用 inverse alpha 製作一個新的 mod table
   Step6 步驟如同 Step3，算出 inverse NTT matrix 後乘上先前算出的 inverse N 再取 mod M 則為所需要的 inverse NTT matrix

R10945004
孫欽鉉

2\. (a) 若對兩個 real sequence $f_1[n]$, $f_2[n]$ 做 DFTs

Step1: $f_3[n] = f_1[n] + jf_2[n]$

Step2: $F_3[m] = DFT\{f_3[n]\}$

Step3: $F_1[m] = \dfrac{F_3[m] + F_3^*[N-m]}{2} \Rightarrow DFT(f_1[n])$

$F_2[m] = \dfrac{F_3[m] - F_3^*[N-m]}{2j} \Rightarrow DFT(f_2[n])$

∴ 只需一個 DFT

(b) if $x_1[n]$, $x_2[n]$ are real and even

$x_3[n]$, $x_4[n]$ are real and odd

$$y[n] = \underbrace{x_1[n] + x_3[n]}_{\Downarrow} + j(\underbrace{x_2[n] + x_4[n]}_{\Downarrow})$$

$$= y_1[n] + j\, y_2[n] \Rightarrow \begin{cases} Y_1[n] = \dfrac{Y[n] + Y^*[N-n]}{2} \\ Y_2[n] = \dfrac{Y[n] - Y^*[N-n]}{2} \end{cases}$$

$Y_1[m] = X_1[m] + X_3[m]$ $\quad \left(\begin{array}{l} \because X_1[m] = X_1[N-m] \\ X_3[m] = -X_3[N-m] \end{array} \right)$

$$\Rightarrow Y_1[m] + Y_1[N-m] = X_1[m] + X_1[N-m] + \cancel{X_3[m]} + \cancel{X_3[N-m]}$$

$$= 2X_1[m]$$

$$X_1[m] = \frac{1}{2}(Y_1[m] + Y_1[N-m]) = \frac{1}{4}(Y[n] + Y^*[n] + Y[N-n] + Y^*[N-n])$$

$$\Rightarrow Y_1[m] - Y_1[N-m] = X_1[m] - X_1[N-m] + X_3[m] - X_3[N-m]$$

$$= 2X_3[m]$$

$$X_3[m] = \frac{1}{2}(Y_1[m] - Y_1[N-m]) = \frac{1}{4}(Y[n] - Y^*[n] + Y^*[N-n] - Y[N-n])$$

$$Y_2[m] = X_2[m] + X_4[m]$$

$$\Rightarrow Y_2[m] + Y_2[N-m] = X_2[m] + X_2[N-m] + \cancel{X_4[m]} + \cancel{X_4[N-m]}$$

$$= 2X_2[m]$$

$$X_2[m] = \frac{Y_2[m] + Y_2[N-m]}{2} = \frac{1}{4}\left(Y[n] - Y^*[N-n] + Y[N-n] - Y^*[n]\right)$$

$$\Rightarrow Y_2[m] - Y_2[N-m] = X_2[m] - X_2[N-m] + X_4[m] - X_4[N-m]$$

$$= 2X_4[m]$$

$$X_4[m] = \frac{Y_2[m] - Y_2[N-m]}{2} = \frac{1}{4}\left(Y[n] - Y^*[N-n] - Y[N-n] + Y^*[n]\right)$$

3 (a) H_{32} 之 1^{st} row ~ 16^{th} row 為 H_{16} 之每一點重複，

17^{th} row $= x_1 - x_2$

18^{th} row $= x_3 - x_4$

19^{th} row $= x_5 - x_6$

20^{th} row $= x_7 - x_8$

21^{th} row $= x_9 - x_{10}$

22^{th} row $= x_{11} - x_{12}$

23^{th} row $= x_{13} - x_{14}$

$\therefore$ row 23^{th} $= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

(b) 可以用來偵測局部的影像特徵，像是 Adaboost face detection

4.

(a) LTI analysis → improper

因為在進行 logical convolution 得到的結果會有些許誤差，相較於 circular convolution

(b) Step-like signal expansion → proper

因為 walsh transform 的變化都為 -1 与 1 的變化，此變化如圖 ，所以適合做 step-like 的 signal

(c) modulation → proper

因為 walsh transform 相比 DFT 運算量少很多，且一樣且有 orthogonal 的性質，因此在調變與解調時都使用 walsh transform

(d) localized feature extraction → proper

因為 walsh transform 每個 raw 會有不同的 sign change 的數量，因此也能來做頻譜分析，而能做頻譜分析自然就能分析 localized feature extraction.（分析高頻區域 → local feature）

(5)

(a) ∵ 有 16-point 的 walsh transform

∴ 需 W_{16}，16x16 的 matrix

而因共有 16 個 row，

每個 row 需 7 次加減法，(☆減法 equal 加法)

∴ total 16-point walsh transform 需 $16 \times 7 = 112$ 個加法

↳(如果沒做任何 optimal 的處理)

(b) 因為 16-point 的 NTT

至多只需 36 個加法和減法，

所以可以使用 look-up table 來事先儲存 value

因此可以只花 0 個加法。

6.

advantage1： OFDM會有正交的性質，使於我們還原信號，相較於FDM。

advantage2： OFDM可以執行快速演算法，如同FFT可以使用的快速演算法OFDM都能使用，因為OFDM在discrete case其實就是IDFT。

7.

① 報告同學使用GAN的架構來當作模擬的DJ，透過生成器(DJ + Mixer)，然後會有分辨器(聽眾)，來訓練。

② 同學提到在1-D convolution without activation function的時候，其實是等效於一個可調的FIR filter，而RNN without activation function則如同可調式的IIR filter。

8.

(a) data [1 0 1] [0 1 0] [1 1 0] ⇒ [1 -1 1] [-1 1 -1] [1 1 -1]

1^{st} columns = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

5^{th} columns = [1 1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 -1 1 1]

10^{th} columns = [1 -1 -1 1 1 -1 -1 1 -1 1 1 -1 -1 1 1 -1]

⇒ [1 -1 1] modulated by channel 1

[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 : -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 :
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

⇒ [-1 1 -1] modulated by channel 5

[-1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 : 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 :
-1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1]

⇒ [1 1 -1] modulated by channel 10

[1 -1 -1 1 1 -1 -1 1 -1 1 1 -1 -1 1 1 -1 : 1 -1 -1 1 1 -1 -1 1 -1 1 1
-1 -1 1 1 -1 : -1 1 1 -1 -1 1 1 -1 1 -1 -1 1 1 -1 -1 1 1]

+)

result : [1 -1 1 3 3 1 -1 1 -1 1 3 1 1 3 1 -1 : 1 -1 -3 -1 -1 -3 -1
1 -1 1 -1 -3 -3 -1 1 -1 : -1 1 3 1 1 3 1 -1 1 -1 1 3 3 1 -1 1 1]

(b)

Yes, 適合使用NTT來做CDMA, 除了其本身具有orthogonal的性質之外, 還可根據其特性事先做出查找表(LUT)對比walsh transform不需任何計算。

Extra problem :

delay 1 ↓

[0 2 2 2 2 0 0 0 ⋮ 0 0 0 0 0 -2 -2 -2 ⋮ -2 0 0 0 0 2 2 2]

decode Ch2 [1 1 1 1 -1 -1 -1 -1] [1 1 1 1 -1 -1 -1 -1] [1 1 1 1 -1 -1 -1 -1]

channel 2 $6-2=4$ ⋮ $\frac{6}{8}=0.75$ ⋮ $-6-2=-8$

$\frac{4}{8}=0.5$ ⋮ $0.75>0 \Rightarrow 1$ ⋮ $\frac{-8}{8}=-1$

$0.5>0 \Rightarrow 1$ ⋮ ⋮ -1

$\therefore$ get $[1, 1, -1] \Rightarrow [1, 1, 0]$ #