第二章 基礎理論與技術

2.1 語音辨識

2.1.1 語音辨識流程

語音辨識流程包括特徵向量的抽取、隱藏式馬克夫模型訓練和最後的語音辨識及由音節轉文字的模組，其全部流程如下圖所示。

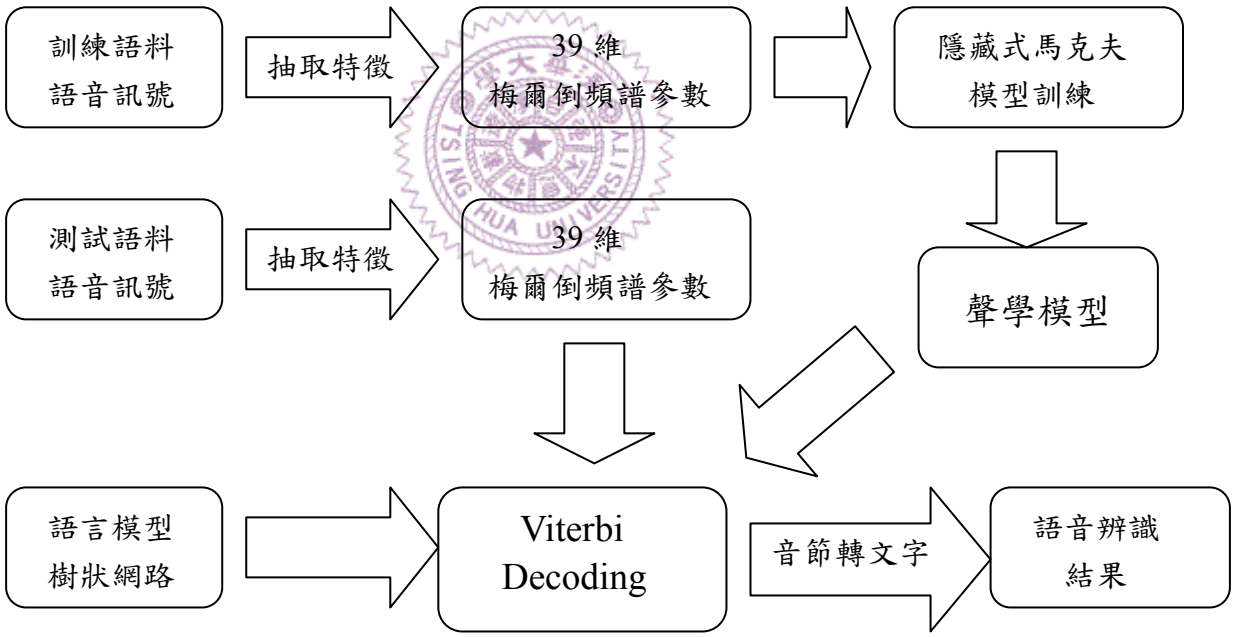

圖 2- 1 語音辨識流程圖

由圖 2- 1 可知在訓練過程中，我們先將訓練語料經特徵抽取，取出語音中的特徵，在此是利用 39 維的梅爾倒頻譜參數來作為特徵參數，之後再以隱藏式馬克夫模型訓練出聲學模型。接下來在辨識過程中，我們先抽取測試語料的特徵

後，將之前的聲學模型及語言模型經由維特比演算法(viterbi algorithm)找出最相似的音節，再把音節轉成文字就可得到語音辨識結果。

在整個流程圖中有特徵參數抽取、聲學模型(隱藏式馬克夫模型)、語言模型(樹狀網路)和維特比演算法的內容，都會在之後小節加以說明。

2.1.2 抽取特徵參數

梅爾倒頻譜係數(mel-scale frequency cepstral coefficients,簡稱 MFCC)，是在語音辨識上最有效的聲學特徵，其抽取流程如下所示。

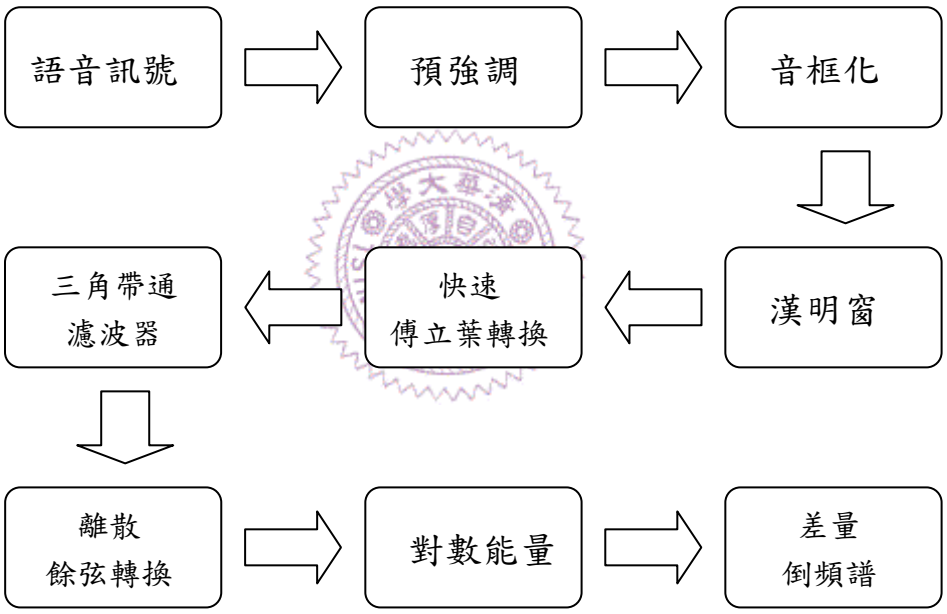

圖 2- 2 梅爾倒頻譜參數抽取流程

首先將 16KHz 語音訊號通過一高通濾波器 $H(z)=1-a\times z^{-1}\quad 0\le a\le 1$(取 a=0.975)，假設輸入訊號為 $S(n)$ ，經預強調後為 $S_2(n)=S(n)-a\times S(n-1)$ 。預強調目的是為了消除發聲過程中，聲帶與嘴唇的效應，並且補償語音訊號經由發音系統所抑制的高頻部分；之後取音框化，音框長度為 320 點，重疊 160 點，經音框化處理後的語音訊號，有時並非是一週期性的訊號，但是快速傅立葉轉換，又

只能作週期性訊號的運算，為了使音框化後的訊號看似為一週期性訊號，乘上漢明窗以增加此音框左端和右端的連續性。假設訊號為 $S(n), n = 0,1,..., N-1$ 乘上漢明窗後為 $S'(n) = S(n) \times W(n)$ ，其中 $W(n)$ 數學式如下：

$$W(n,a) = (1-a) - a \times \cos\left(\frac{2n\pi}{N-1}\right), a = 0.46,\ 0 \le n \le N-1$$

接下來作快速傅立葉轉換，求出音框的頻譜；再帶入一組 26 個三角帶通濾波器(triangular bandpass filter)求出每個頻帶輸出的對數頻譜；最後經離散餘弦轉換(discrete cosine transform)求出梅爾倒頻譜參數 C_m 。其數學式如下：

$$C_m = \sum_{k=1}^{M} E_k \times \cos\left(\frac{m \times (k-0.5) \times \pi}{M}\right), M = 26, m = 1,2,\ldots,L$$

訊號由離散餘弦轉換後可得到 12 維倒頻譜參數，再加上 1 維的對數能量，共可得到 13 維特徵參數，最後為了要顯示倒頻譜參數對時間的變化，會再取這 13 維的差量倒頻譜，其數學式如下：

$$\Delta C_m(t) = \frac{\sum_{\tau=-M}^{M} \tau \cdot C_m(t+\tau)}{\sum_{\tau=-M}^{M} \tau^2}$$

通常我們取 M=2，經一階及二階差量倒頻譜計算後可得到 26 維倒頻譜參數，再加上之前運算的 13 維參數總共有 39 維倒頻譜參數，這也就是一般語音辨識上常用的 39 維梅爾倒頻譜係數。

2.1.3 聲音單元

一個中文字即代表一個音節(syllable)，本論文中使用的聲音單元是音節內右相關(biphone)的聲學單位，例如「清華」的「華」字，漢語拼音為「hua」，其音節內右相關的聲音單元為「sil+h」，「h+u」，「u+a」，「a+sil」，此聲音單元將視為語音的最小單位，並為每一單元訓練其聲學模型，意即每單元都有一個模型(model)。

2.1.4 隱藏式馬可夫模型

在之前所用到的聲學模型是用隱藏式馬可夫模型(Hidden Markov Model , HMM)所訓練出來的。隱藏式馬可夫模型基本上是一種雙重隨機過程，因為其中有一組隨機過程是隱藏的，所以稱為隱藏式。另一組隨機過程稱為觀測序列(observation sequence)，是由狀態觀測機率(state observation probability)在每個狀態下觀測到各種語音特徵參數的機率分佈情況。

將 HMM 中隱藏隨機過程當作聲道正處於發某個聲音的組態，而在這發聲狀態下聽到各種可能的聲音當作狀態觀測機率。在作訓練時每一個聲音單元皆有一個 HMM，一個模型通常有 3 個狀態，在 HTK 的表示法如圖 2- 3 所示。

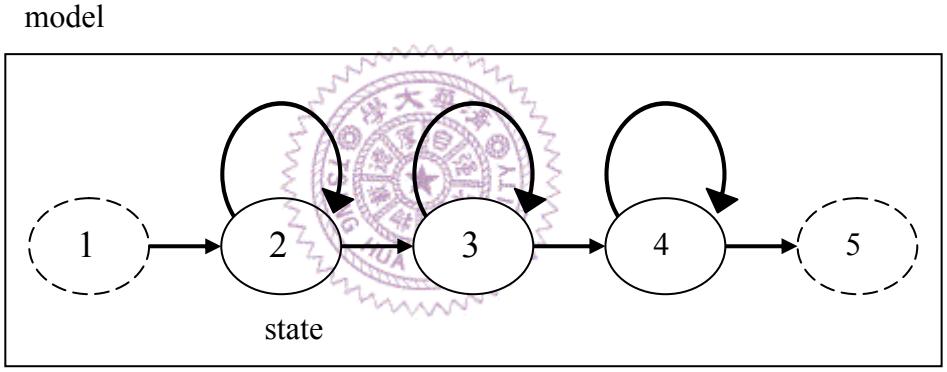

圖 2- 3 模型與狀態表示圖

HMM 的狀態觀測機率函數$b_j(o_t)$定義如下：

$$b_j(o_t)=\prod_{s=1}^{\#S}\left[\sum_{m=1}^{\#M_s} w_{jsm}G_{jsm}\right]^{r_s}$$

上式中#S 代表 Stream 的數量；r_s為 Stream 的權重(weight)取為 1；$\#M_s$表示 Stream 為 s 時 mixture 數量；w_{jsm}和G_{jsm}代表狀態在 j 時 Stream 為 s，且 mixture 為 m 時的高斯函數權重及高斯機率密度函數。

G_{jsm} 定義如下：

$$G_{jsm} = g(x;\mu\Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu) \right]$$

上式中 d 是維度，μ 和 Σ 代表此高斯機率密度函數的平均值(mean)及共變異矩陣(covariance matrix)。

本篇論文用 3 個 state，每個 state 有 3 個 Stream 分別為 6、2 及 2 個 mixture 來作聲學模型的訓練。其圖形化如圖 2- 4 所示。

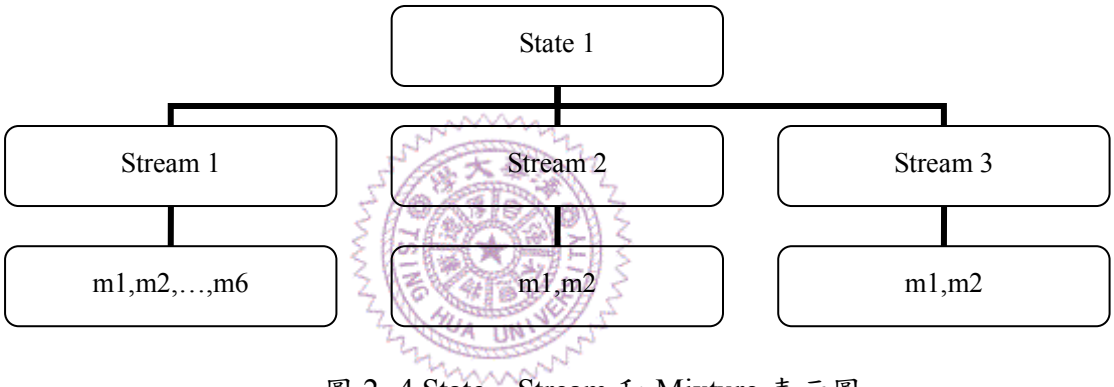

圖 2- 4 State、Stream 和 Mixture 表示圖

2.1.5 語音辨識法則

當我們將聲學模型訓練出來後，就可以和測試語料的特徵參數作比對，來得知辨識結果。而要如何辨識測試語料的結果，就是決定觀測序列是由哪些模型序列來描述是最合適的。本篇是以維特比演算法(viterbi algorithm)來比對出和觀測序列最相近的狀態序列。

狀態觀測機率 $b_j(o_t)$ 運算過程：

假設#S=3 代入狀態觀測機率 $b_j(o_t)$ 可得

$$b_j(o_t) = \prod_{s=1}^{3}\left(\sum_{m=1}^{\#M_s} w_{jsm} G_{jsm}\right)$$

將上式取對數(log)可得

$$\sum_{s=1}^{3}\log\left(\sum_{m=1}^{\#M_s} w_{jsm} G_{jsm}\right)$$

假設要求出 stream 1 的對數機率其運算過程如下[11]：

$$\log\left(w_1 G_1 + w_2 G_2 + w_3 G_3 + \ldots\right)$$

由高斯函數

$$G = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

則

$$\log(G) = -\frac{1}{2}\log\left((2\pi)^d |\Sigma|\right) - \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)$$

定義

$$GConst = \log\left((2\pi)^d |\Sigma|\right)$$

並依下式求出 *GConst*

$$\begin{aligned} GConst &= d\log(2\pi) + \log\left(|\Sigma|\right) \\ &= 13 \times \log(2\pi) + \sum_{i=1}^{13} |\mathrm{var}[i]| \end{aligned}$$

而 $(x-\mu)^T \sum^{-1}(x-\mu)$ 依下列矩陣求出

$$(x-\mu)^T \sum^{-1}(x-\mu) = \underset{1\times 13}{\begin{bmatrix} A & B & .. & .. \end{bmatrix}} \times \underset{13\times 13}{\begin{bmatrix} C & 0 & .. & 0 \\ 0 & D & 0 & .. \\ .. & 0 & .. & 0 \\ 0 & .. & 0 & .. \end{bmatrix}} \times \underset{13\times 1}{\begin{bmatrix} A \\ B \\ .. \\ .. \end{bmatrix}}$$

$$= \underset{1\times 13}{\begin{bmatrix} AC & BD & .. & .. \end{bmatrix}} \times \underset{13\times 1}{\begin{bmatrix} A \\ B \\ .. \\ .. \end{bmatrix}}$$

$$= A^2C + B^2D + \ldots$$

以上為狀態觀測機率 $b_j(o_t)$ 運算過程，以下為維特比演算法介紹：

首先假設觀測序列 $\vec{O} = \{o_1, o_2, \ldots, o_T\}$，其最佳狀態序列為 $\vec{q} = \{q_1, q_2, \ldots, q_r\}$，以 $\delta_t(i)$ 表示從頭開始，直到時間點 t 時的觀測值 o_t 為狀態 i 的最大機率。

其數學式表示如下(其中 λ 為 HMM)：

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P\left(q_1, q_2, \ldots, q_{t-1}, q_{t-1} = i, \vec{O} \mid \lambda\right)$$

再由[14]可知

$$\delta_{t+1}(i) = \left[\max_i \delta_t(i) \times a_{ij}\right] \times b_j(o_t)$$

其中 a_{ij} 為狀態 i 跳到狀態 j 的轉移機率(transition probability)，而 $b_j(o_t)$ 則為在狀態 j 時出現 o_t 的觀測機率值。維特比演算法[14]運算過程如下：

1. 初始化：

$$\delta_1(i) = \pi_i b_i(o_1)$$
$$\psi_1(i) = 0, \quad 1 \le i \le K$$

$\psi_t(j)$ 為回溯(backtracking)

2. 遞迴步驟：

$$\delta_t(j) = \left[\max_{1\le i\le K} \delta_{t-1}(i) \times a_{ij}\right] \times b_j(o_t)$$
$$\psi_t(j) = \arg\max_{1\le i\le K}\left[\delta_{t-1}(i) \times a_{ij}\right], \quad 2 \le t \le K, \quad 1 \le j \le K$$

3. 結束(Termination)：

$$P^* = \max_{1\le i\le K}[\delta_T(i)]$$
$$q_r^* = \arg\max_{1\le i\le K}[\delta_T(i)]$$

4. 回溯步驟(state sequence backtracking)：

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1$$

由上述步驟可找到最佳狀態序列。

2.1.6 辨識網路

將圖 2- 1 中測試語料抽取特徵值後，再和語言模型所用的樹狀網路及先前所訓練出來的聲學模型，利用上一節介紹的維特比演算法作比對，來辨識出測試語料的結果，樹狀網路圖形如圖 2- 5 所示。假設要辨識的詞為「台北市」、「台東縣」、「新竹市」和「新竹縣」共四個，前兩個都是以「台」為字根後面則接「北」和「東」，因為只有接兩個字所以搜尋空間少，使得辨識速度加快，並減少記憶體使用量。

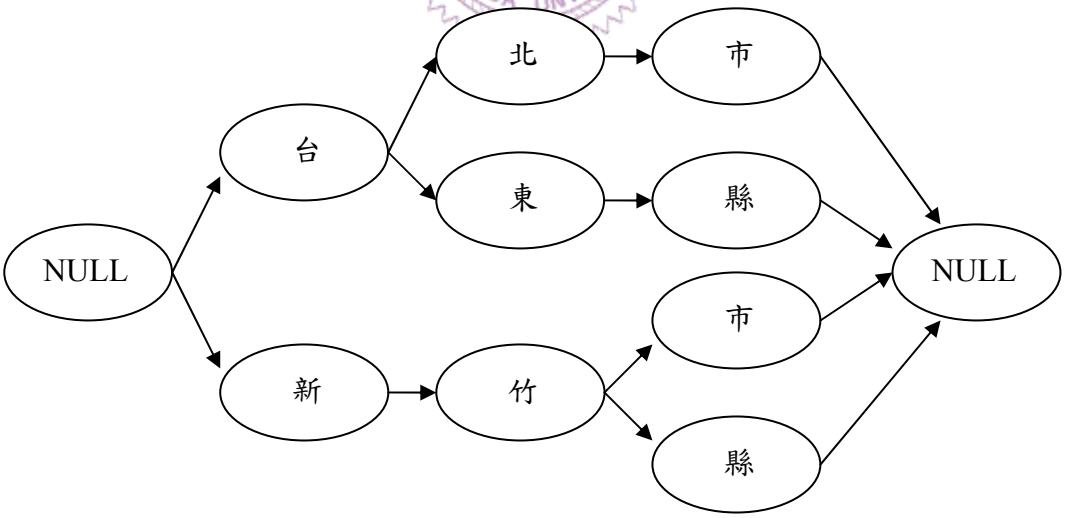

圖 2- 5 樹狀網路

2.2 快速傅立葉轉換

2.2.1 快速傅立葉轉換原理

因為訊號特性在時域(time domain)上的改變較難作分析，所以將訊號轉換到頻率域(frequency domain)上以方便作訊號特性分析。將此轉換過程稱作離散傅立葉轉換 (discrete fourier transform)，簡稱為 DFT 而其數學式如(2. 1)所示：

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{kn} \quad k=0,1,\ldots,N-1 \qquad (2.1)$$

其中 N 代表週期而 $W_N^{kn}=e^{-j(2\pi/N)kn}=\cos\left(-\frac{2\pi kn}{N}\right)+j\sin\left(-\frac{2\pi kn}{N}\right)$。

由(2. 1)可知需要 N^2 個乘法運法量，而由[10]提出方法可將乘法運算量減少至 $N\log N$ 個，使運算速度加快了許多。因此將此方法稱為快速傅立葉轉換(fast fourier transform)，簡稱為 FFT 而其數學式推導如下：

首先將(2. 1)分成奇數及偶數兩項如(2. 2)，假設 $n=2r$ 並提出奇數項的 W_N^k 如(2. 3)，將 $W_N^2=e^{-2j(2\pi/N)}=e^{-j2\pi/(N/2)}=W_{N/2}$ 代入(2. 3)後可簡化成(2. 4)。

$$X[k]=\sum_{n=even}x[n]W_N^{nk}+\sum_{n=odd}x[n]W_N^{nk} \qquad (2.2)$$

$$=\sum_{r=0}^{N/2-1}x[2r]W_N^{2rk}+\sum_{r=0}^{N/2-1}x[2r+1]W_N^{(2r+1)k}$$

$$=\sum_{r=0}^{N/2-1}x[2r]W_N^{2rk}+W_N^k\sum_{r=0}^{N/2-1}x[2r+1]W_N^{2rk} \qquad (2.3)$$

$$=\sum_{r=0}^{N/2-1}x[2r]W_{N/2}^{rk}+W_N^k\sum_{r=0}^{N/2-1}x[2r+1]W_{N/2}^{rk} \qquad (2.4)$$

接下來我們定義參數 x[k]+x[k+1]和 x[k]×W 等效圖形如圖 2- 6 所示，其中(b)圖等效於(a)圖，(d)圖等效於(c)圖。

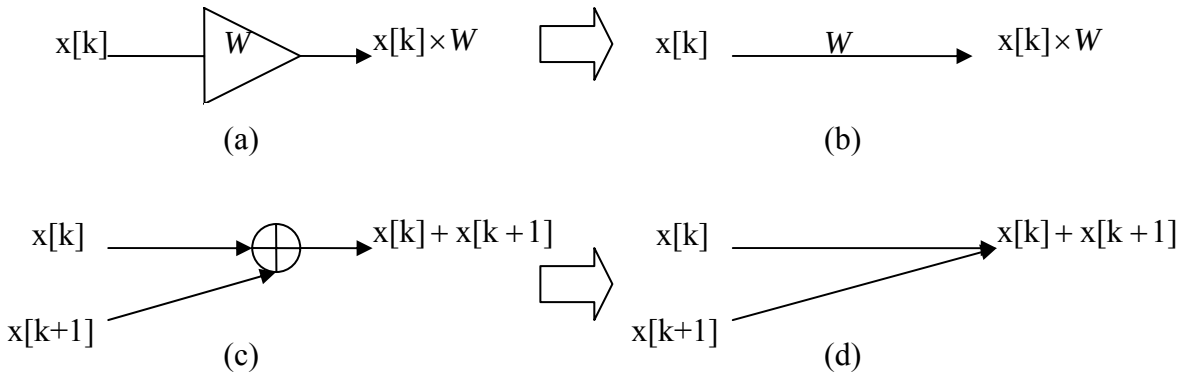

圖 2- 6 運算式等效圖形

假設(2. 4)中的 N=8 其圖形化後的結果如圖 2- 7 所示，由於圖形中交錯的箭頭很像蝴蝶一般，所以又稱為蝴蝶圖(butterfly graph)[3]。

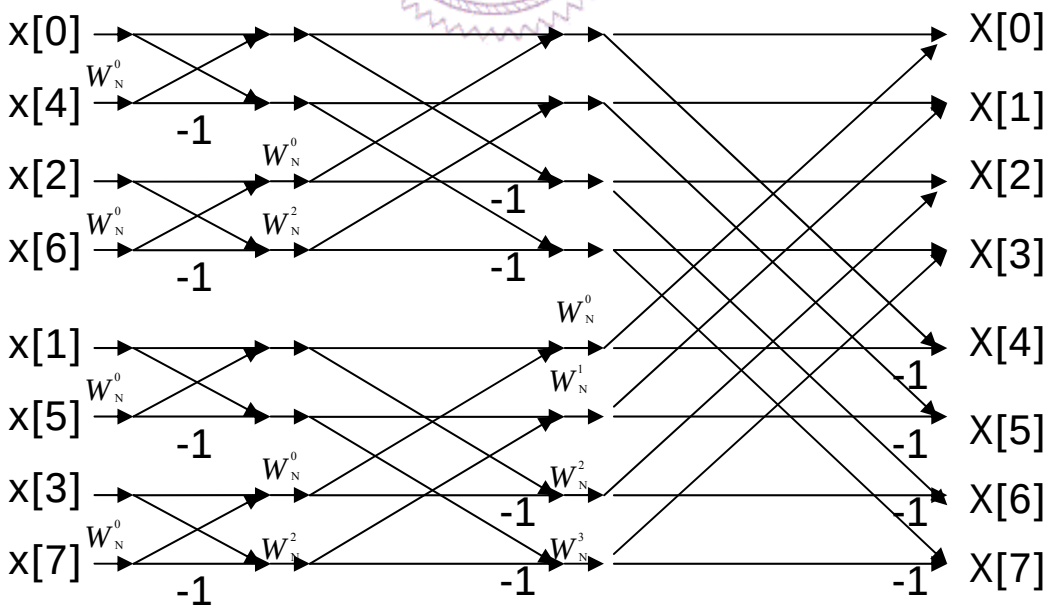

圖 2- 7 N=8 FFT 蝴蝶圖

2.2.2 浮點數轉整數型態方式

因為 $\sin\theta$ 和 $\cos\theta$ 值介於-1~1 之間，在整數型態化過程中必須乘上一放大係數(scaling-up factor ,簡稱 SF)，其數學式為 $\text{int}(SF\times\sin\theta)$，int( )代表取括弧內四捨五入的值；之後為了要方便使用整數化後的值，則會將其值存入一 Table 表中。首先假設 SF=1024，要建的值假設有 512 個，值為 $\sin\left(\frac{n\pi}{512}\right), n=0,1,2,...,511$ 共 512 個值，而 Table 表內的對應位置 β，就是 1,2,…,512。例如要求 $n=128$，則 $\sin\left(\frac{128\pi}{512}\right)=\sin\left(\frac{\pi}{4}\right)$，則對應到 β 的位置就是 129。整個流程如下圖所示。

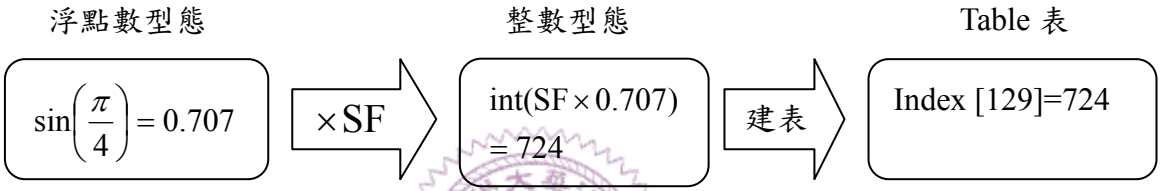

圖 2- 8 浮點數型態轉換整數型態範例

2.2.3 整數型態 FFT1

接下來介紹整數型態 FFT1 的方法，因為每級在運算時不需乘上 $\frac{1}{SF}$，所以優點為保留每級運算時的精確度。而付出的代價就是建表時的放大係數很小，使得建表值誤差很大。

首先在 FFT 運算過程中，經下圖的運算方式(其中 k=0,1,2,…,N-5)，稱為一個"級"。

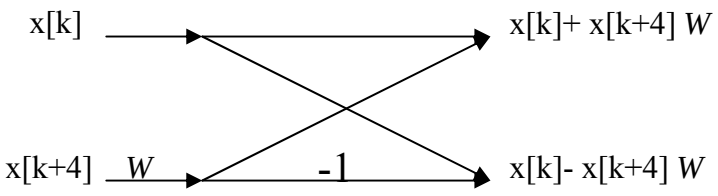

圖 2- 9 "級"表示圖形

FFT1 就是用上述方式，將圖 2- 7 分成三個級。由圖中可知每一級皆為相乘的關係，所以 FFT1 整體的放大係數是 $SF = (sf)^{\log_2 N}$，因此每級的放大係數為 sf，如下圖所示。

範例說明：假設 $SF = 2^{12} = 4096$，則每級所得到的 $sf = 2^{\frac{\log_2 4096}{\log_2 8}} = 2^4 = 16$。

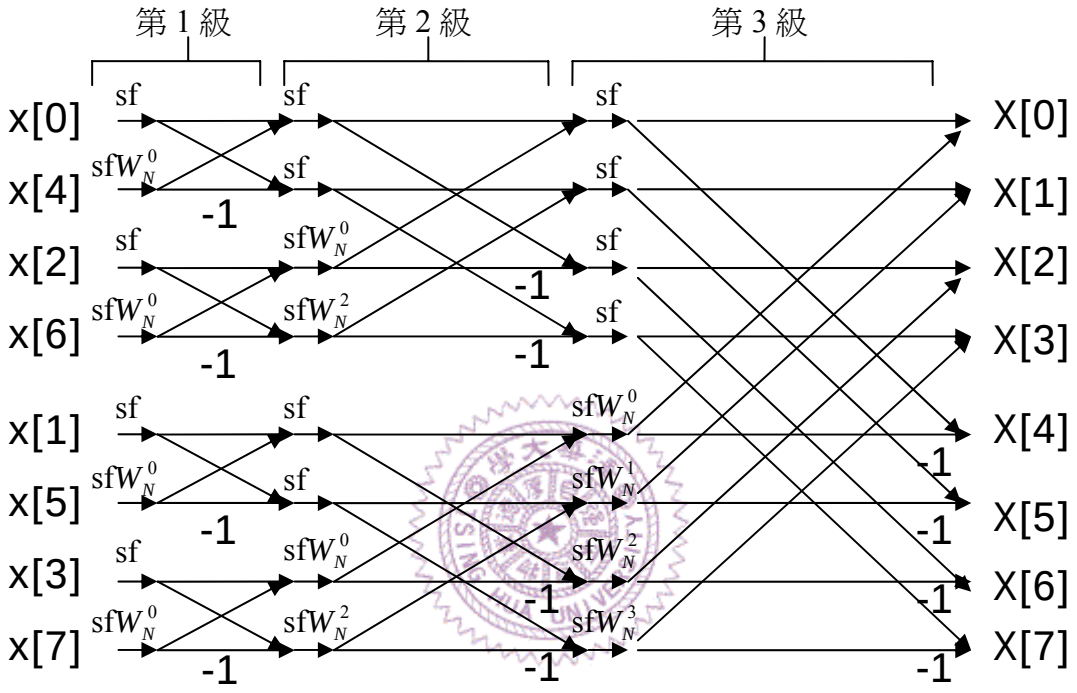

圖 2- 10 整數型態 FFT1 蝴蝶圖