

A deep learning based SCF decoding algorithm of Polar Codes

基於深度學習之極化碼 SCF 解碼演算法

組別: B78 組員: 王政皓 鄭凱鴻 指導教授: 翁詠祿

摘要

極化碼是一種用於訊號傳輸的錯誤更正編解碼技術，透過通道極化的處理，使得一部分的通道容量趨近於 1(完美通道)，另一部分通道容量趨近於 0(純雜訊通道)，並利用趨近完美通道的部分作為傳輸通道，藉此逼近香農極限。

當編碼長度愈長，通道極化的現象會變得愈明顯，所以在編碼長度有限的狀況下，基礎的逐次消除(Successive Cancellation, SC)解碼演算法的性能並不好。SCF (Successive Cancellation Flip)解碼演算法的出現，是為了提升極化碼的解碼性能(即降低訊框錯誤率(Frame Error Rate, FER))。SCF 解碼透過數次重新進行 SC 解碼並在過程中翻轉一個錯誤機率高的位元來避免錯誤傳遞。在本次專題中，會介紹 SCF 解碼演算法並且比較在 SCF 解碼演算法中使用與未使用深度學習決定翻轉位元的差異。

關鍵字: 第五代行動通訊技術、極化碼解碼、深度學習、通訊演算法

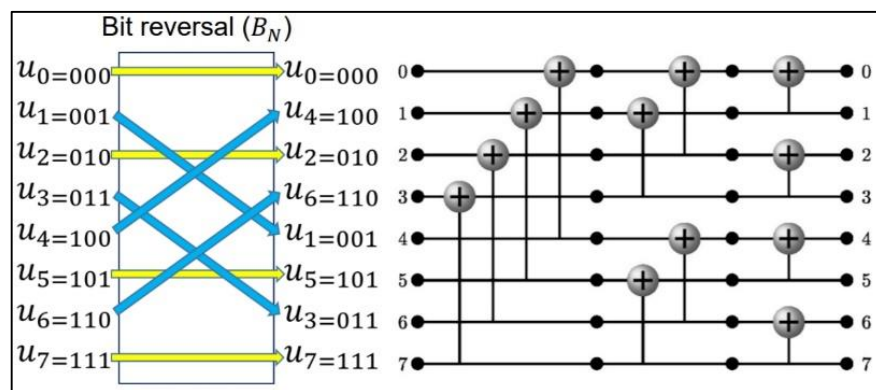
內容

1. 通道編碼

極化碼可以利用 $P(N, K)$ 來表示，其中 N 是碼長、 K 是用來傳送資訊的資訊位元數，不用來傳送資訊的 $(N-K)$ 個位元則當作凍結位元。原始訊息 $\mathbf{u}$ 為一長度為 N 的 row vector，經過編碼後得到 $\mathbf{x}$ ，兩者的關係如下：

$$\mathbf{x} = \mathbf{u} G_N$$

其中 $G_N = B_N F^{\otimes n}$ ， $F^{\otimes n}$ 是極化陣列 $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ 的 n 次 ($n = \log_2 N$) 克羅內克積 (Kronecker product)，以 $N=8$ 為例，其為圖一右側部分， B_N 是位逆序置換矩陣 (Bit-reversal permutation matrix)，為圖一左側部分。

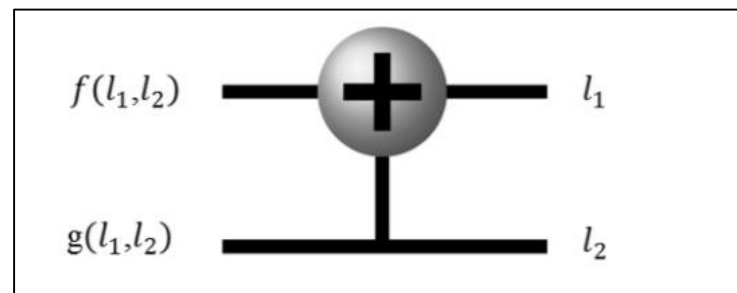


(圖一)編碼圖示(以 $N=8$ 為例)

2. 通道解碼(SC 解碼與 SCF 解碼演算法)

使用 BPSK 調變方式經過 AWGN 通道後得到的接收訊號 y ，再將 y 透過高斯近似的方式得到接收端各位元的對數似然比(log-likelihood ratio,LLR)。以 $N=8$ 為例，解碼圖就是編碼圖右邊灰色部分，而接收端各位元的對數似然比就是灰色部分中最右邊那個 column 從上到下 N 個節點的對數似然比； u 各個位元的 LLR 值就是灰色部分中最左邊那個 column 從上到下 N 個節點的對數似然比。

在 SC 演算法中，會將解碼圖中兩種節點的 LLR 分別用 f 函數與 g 函數計算，前一種節點是圖二中左上角的節點，而後一種節點則是左下角的節點。



(圖二)解碼架構單元

解碼的過程是從解碼圖中由右至左計算各節點的 LLR，接收端各位元的 LLR 就像樹狀圖的葉節點，而 u 各個位元的 LLR 值就像根節點，透過遞迴關係式可以計算得到 u 各個位元的 LLR 值。(例如：第一個解碼位元是用解碼圖最左邊 column 從上到下第一個節點(u_0)的 LLR 判斷，第二個解碼位元是用最左邊 column 從上到下第五個節點(u_1)的 LLR 判斷，依照 u 的順序當作解碼順序，以此類推。若將全數解碼位元解完還要再經過一次位逆序置換 (Bit-reversal permutation)才會還原為正確順序的 u 。)而判斷方法為如果該位元是資訊位元則當 $LLR > 0$ ，將該位元判斷成 0，反之判斷成 1；如果該位元是凍結位元，因為傳送的時候凍結位元都會被設成 0，所以就直接判斷成 0。

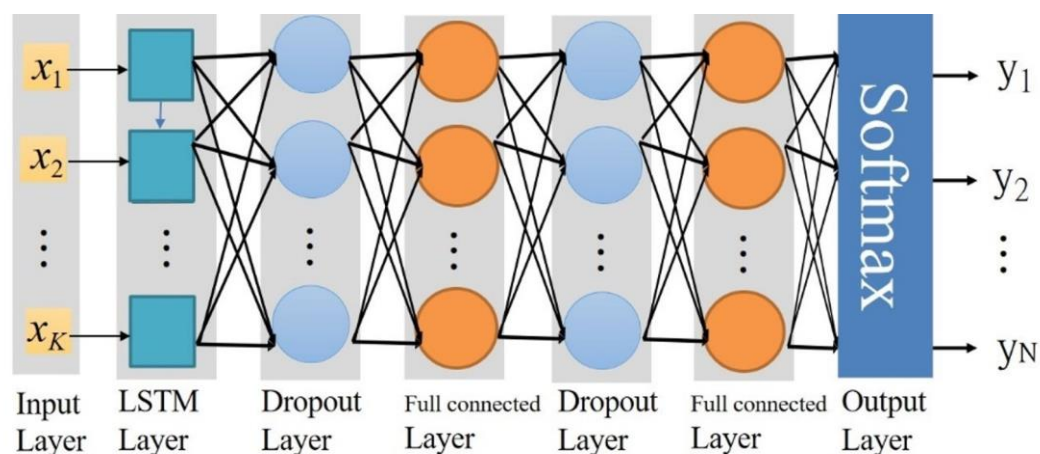
在節點 LLR 計算中， f 函數與 g 函數可寫作：

$$f(l_1, l_2) = \text{sign}(l_1) * \text{sign}(l_2) * \min(|l_1|, |l_2|)$$
$$g(l_1, l_2) = l_2 + l_1 * (-1)^p$$

其中 p 是從解碼圖中由左至右經邏輯運算到圖中左上角節點的結果，其值會被先前解出位元的判斷結果所決定，也就是說，若前面位元解錯就會造成錯誤傳遞。當 u 中資訊位元的 LLR 絕對值很小時，代表有很高的機率會發生判斷錯誤，SCF 演算法就是基於這樣的特性，若首次 SC 解碼錯誤，將資訊位元 LLR 絕對值前 T 小的索引(index)從小到大存成翻轉索引集合，之後將 SC 解碼重解 T 次，每次重解依照集合翻轉(意思是把原本解成 1 的結果改成 0，原本解成 0 改成 1)該個索引的解碼結果再繼續解碼下去，透過每次翻轉一個高機率錯誤的位元來降低 FER。此外，SCF 會將部份的資訊位元用來傳送 CRC 檢查碼，讓接收端可以藉由 CRC 檢查知道該次解碼是否正確，如果不正確就繼續重新解碼，並依照翻轉索引集合翻轉一個索引結果，直到成功(通過 CRC 檢查)或到重解上限 T 。

3. DL-SCF 解碼演算法

將 SCF 演算法中重新解碼才成功的情況當作訓練資料，利用以 LSTM 為核心的神經網路模型預測具有很高的機率會解碼錯誤的位元索引，其 optimizer 為 adam、loss function 為 categorical_crossentropy。在 DL-SCF 解碼演算法中，會將 softmax 輸出的機率分布排序，選出前 T 大的索引(index)從大到小存成翻轉索引集合，之後將 SC 解碼重解 T 次，每次重解依照集合翻轉該個索引的解碼結果再繼續解碼下去，透過每次翻轉一個高機率錯誤的位元來降低 FER。並將其結果與透過資訊位元的 LLR 絕對值來決定翻轉索引集合的 SCF 演算法比較。

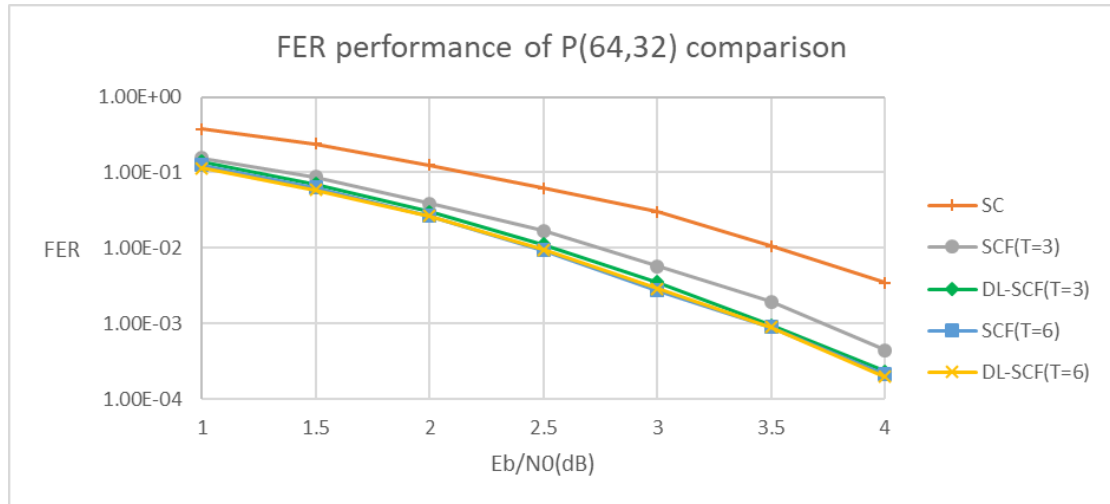


(圖三)神經網路架構

4. 實驗結果

每次實驗會隨機生成 24 位元(隨機的 1 或 0 透過均勻分布生成)傳送資料並生成 8 位元 CRC 檢查碼一起當作資訊位元，接收解碼後得到各位元的 LLR 值。若首次 SC 解碼失敗，將 LLR 絕對值與 CRC 檢查後的八位元徵狀值(Syndrome)當作神經網路的輸入，再由輸出排序得到翻轉索引集合(集合大小為 T)，進行 T 次，每次依照翻轉索引集合在解碼過程中翻轉一個位元判斷結果的 SC 解碼，結束解碼的條件為：若解碼次數 $< T$ 但通過 CRC 檢查或重解次數達到上限 T 次。

從實驗結果可以看出 DL-SCF($T=3$)和 SCF($T=3$)相比有較低的訊框錯誤率(FER)，而且在平均解碼次數(Average decoding time, ADT)的部分 DL-SCF($T=3$)也比 SCF($T=3$)來得少；DL-SCF($T=6$)和 SCF($T=6$)有著極為相近的訊框錯誤率(FER)，但是在平均解碼次數(Average decoding time, ADT)的部分 DL-SCF($T=6$)比 SCF($T=6$)少。附註：1.關於跑曲線的部分，低 SNR 會跑 1000 個錯誤來讓結果趨於穩定，高 SNR 則跑 100 個錯誤來讓結果趨於穩定 2.在實驗中，64 位元為一個訊框，訊框中只要錯一個位元就算是一個訊框錯誤。3.平均解碼次數代表著進行 SC 解碼的平均次數(包括第一次解碼與重新解碼(每次過程中需依照翻轉索引集合翻轉一個索引的結果))，當 SNR 很高時 ADT 趨近於 1，意思是在大多數情況下，進行第一次的 SC 解碼就解碼正確了。



(圖四)訊框錯誤率曲線



(圖五)平均解碼次數曲線

5. 結論

使用深度學習決定翻轉索引集合的 SCF 解碼演算法在不同 T 下存在不同表現，當 T=3，其 FER, ADT 都能得到改善；當 T=6，因為 SCF (T=6) 已涵蓋絕大多數翻一個位元就能解碼正確的情況，所以此時使用深度學習並不能改善 FER，而是能透過更精確地決定翻轉索引集合中的元素順序來減少 ADT。

6. 參考文獻

- [1] E. Arıkan, "Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels," IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, July 2009.
- [2] O. Afisiadis et al., "A low-complexity improved successive cancellation decoder for polar codes," Proc. IEEE 48th Asilomar Conf. on Sig., Sys. and Comp., pp.

2116-2120, Nov. 2014.

[3] Bi He, Shaohua Wu, Yajing Deng, Hao Yin, Jian Jiao, Qinyu Zhang: A Machine Learning Based Multi-flips Successive Cancellation Decoding Scheme of Polar Codes. VTC Spring 2020: 1-5

心得感想

王政皓：

從課程開始到海報比賽過了大約 9 個月的時間(包含暑假 2 個月)，我覺得非常充實，從研讀極化碼原理，到實作極化碼編碼與 SC 解碼，再從 SC 解碼延伸到 SCL 解碼，最後是實作海報比賽的 SCF 解碼。在實作專題課程中我學到了很多，除了極化碼編解碼演算法的知識外，還學到了 python 撰寫與神經網路的模型建立。(一開始的程式都是用 C++ 撰寫，但因為專題要用到深度學習的東西，所以會用到 python。)我在撰寫解碼程式的過程中遇到了不少困難，但在助教的協助與自己的努力下都一一克服。感謝助教對我提出的問題都耐心地給予回答，也感謝教授在每次開會上為我釐清觀念與給予建議。

鄭凱鴻：

從一開始修習專題到現在，我學會到很多新知，也體驗到將理論運用於實際的過程。我覺得整個課程雖然不輕鬆、也遇到許多的問題但卻非常的扎實。首先，在原理方面，因為當時還沒接觸過相關的課程，對於不同編碼與解碼的方式、極化碼背後數值的計算、理論的應用都不太了解，而這也大幅增加後續實作層面的難度。接著，由於自己 C++ 的基礎並不是很穩健，在打 code 時屢屢碰壁，遇到非常多的困難與挑戰，還好最後在老師、學長和團隊多次的幫助與討論下，這些問題都一一的解決。我很感謝他們的幫忙，因為有這些指導與協助，才能讓我更加理解背後相關的知識、克服遇到的挑戰，並強化自己解決問題的能力。