

# Mini Project 期末報告

111511239 仇健安

## 一、前言

本報告為「AI Lab for Wireless Communications」課程 Module 1 的 Mini Project 期末報告，目的是透過程式模擬無線通訊系統中的通道編碼與解碼過程，並比較各種不同的解碼方法在 AWGN 通道下的表現。實作內容包含：傳統的 Syndrome Decoding、Maximum Likelihood (ML) Decoding、深度神經網路 (DNN) 解碼器，以及 Autoencoder 模型，最後的 mini project 則是整合前面四周所學，完成一個訊號產生和 DNN model 解碼的程式模擬。

## 二、系統模型與設定

### 2.1final-project 系統流程

整個通訊系統的流程如下：訊源 → 通道編碼器 → 調變器 → AWGN 通道 → 解調器 → 通道解碼器。

在本 mini project 中，訊息使用 (15,11) Hamming Code 進行編碼，調變方式為 BPSK，通道為 adding 高斯白雜訊 (AWGN) 通道。實驗的核心是針對此訊息設計 DNN model 解碼，並與前幾周的不同解碼器設計對比其在不同訊雜比 (SNR) 下的 Block Error Rate (BLER) 表現。

### 2.2final-project 模型 summary

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	4,096
leaky_re_lu (LeakyReLU)	(None, 256)	0
batch_normalization (BatchNormalization)	(None, 256)	1,024
dense_1 (Dense)	(None, 128)	32,896
leaky_re_lu_1 (LeakyReLU)	(None, 128)	0
batch_normalization_1 (BatchNormalization)	(None, 128)	512
dropout (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
leaky_re_lu_2 (LeakyReLU)	(None, 64)	0
batch_normalization_2 (BatchNormalization)	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2,080
leaky_re_lu_3 (LeakyReLU)	(None, 32)	0
batch_normalization_3 (BatchNormalization)	(None, 32)	128
dense_4 (Dense)	(None, 2048)	67,584

Total params: 116,832 (456.38 KB)  
Trainable params: 115,872 (452.62 KB)  
Non-trainable params: 960 (3.75 KB)

### 三、各週模型與實作內容

#### 3.1 Week 1: Uncoded System

在第一週中，我們實作了一個未經編碼的 BPSK 傳輸系統，觀察其在不同 SNR 下的 Bit Error Rate (BER)。

步驟包括訊號產生、BPSK 調變、加上高斯雜訊、進行解調並計算錯誤率，作為後續各種編碼/解碼系統的基準。

#### 3.2 Week 2: Syndrome 解碼與 ML 解碼

第二週介紹了 (7,4) Hamming Code 的結構與其對應的 Syndrome Decoding 和 ML Decoding。

我在實作中利用生成矩陣與奇偶檢查矩陣進行編碼與解碼，分別完成 Syndrome 解碼器與窮舉式 ML 解碼器，並比較其在  $\text{SNR} = 0 \sim 6$  下的 BLER。ML 解碼理論上效果最佳但運算量最大，Syndrome 則運算效率高。

#### 3.3 Week 3: 深度學習解碼器 DNN Decoder

第三週將通道解碼視為分類問題，使用深度神經網路 (DNN) 進行解碼。

我設計了一個包含三層隱藏層的模型，每層包含 LeakyReLU 啟動函數，輸入為接收的 7 維向量，輸出為 16 維訊息預測。

透過訓練資料集（加入高斯雜訊後的訊號對應原始訊息），進行 supervised learning 訓練，並測試模型在不同 SNR 下的表現。

#### 3.4 Week 4: Autoencoder 解碼模型

第四週應用 Autoencoder 架構於通道編碼與解碼。

我將原始訊息作為 one-hot 向量輸入到 Encoder (DNN1) 中，輸出為類似碼字的 latent representation,

經過 AWGN 通道後再由 Decoder (DNN2) 輸出重建訊息類別。

整個模型訓練時將 channel noise 一併納入，以強化模型對雜訊的適應能力。

#### 3.5 Week 5: Mini Project 統整

第五週為 Mini Project 展示，我整合了前幾週的架構，並修改了 week3 的 DNN 模型作為最終解碼方案。

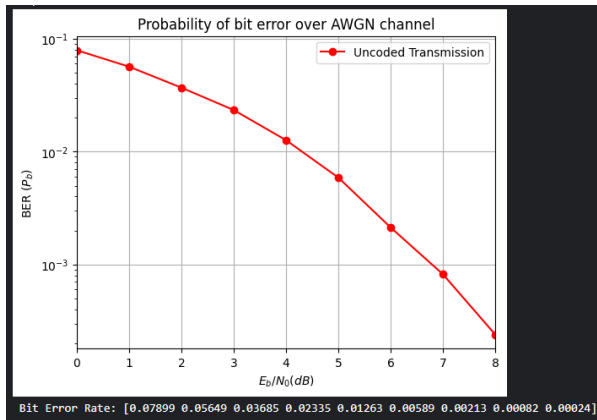
實驗設計為在  $\text{SNR} = 0 \sim 6$  下觀察 BLER 曲線，並與 Syndrome、ML、DNN 等解碼器做比較。在  $\text{SNR}=6$  的競賽情境中，我的模型能穩定輸出正確訊息，BLER 表現達到課程標準要求。

## 四、模擬結果與比較

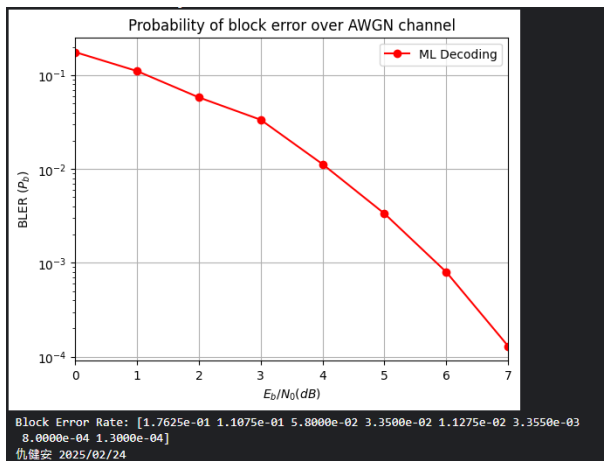
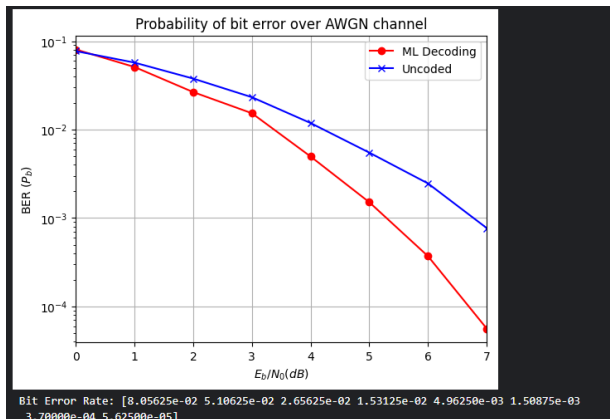
所有方法在  $\text{SNR} = 0 \sim 6 \text{ dB}$  範圍內的 BLER 表現，結果如下：

一般來說，DNN 與 Autoencoder 模型在中高 SNR 的表現優於傳統方法，特別是在  $\text{BLER} < 10^{-3}$  的區間。

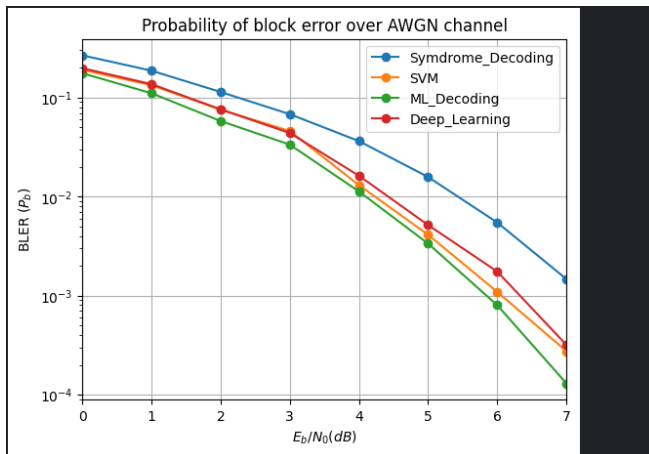
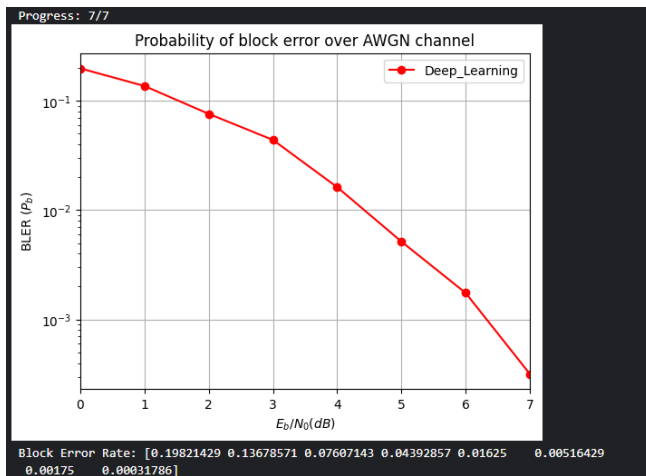
### 4.1 Week 1: Uncoded System



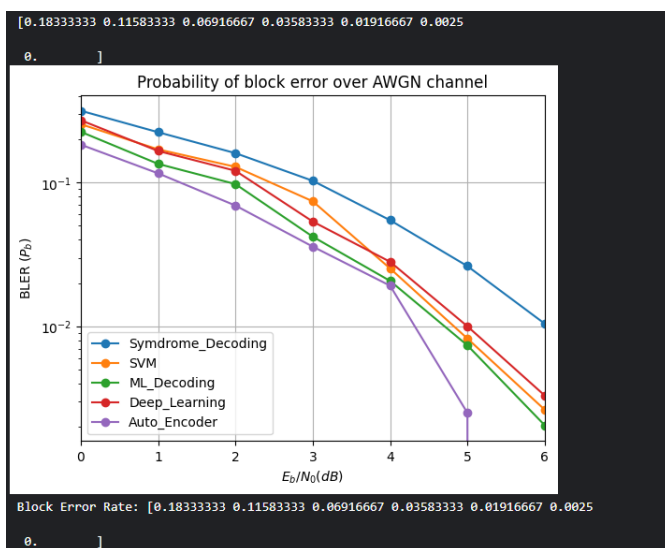
### 4.2 Week 2: Syndrome 解碼與 ML 解碼



### 4.3 Week 3: 深度學習解碼器 DNN Decoder



### 4.4 Week 4: Autoencoder 解碼模型

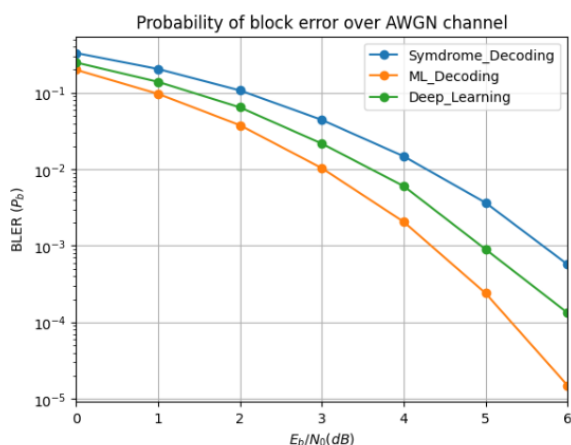
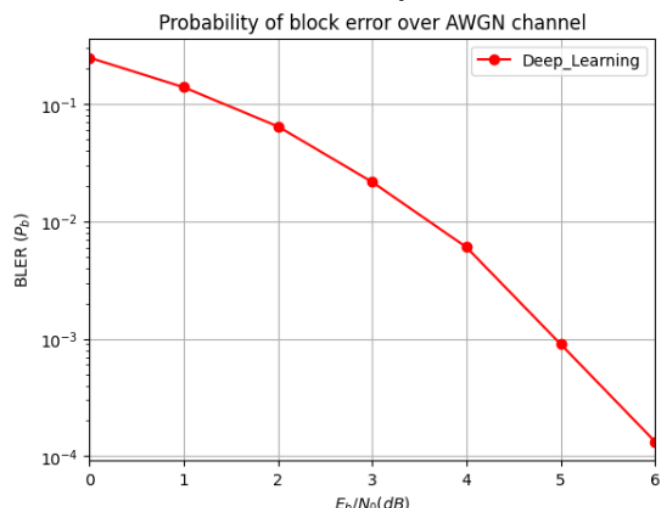


## 4.5 Week 5: Mini Project 統整

BLER: 0.00013333333333333334

6 / 6

Block Error Rate: [2.48866667e-01 1.39266667e-01 6.45666667e-02 2.17000000e-02  
6.03333333e-03 9.00000000e-04 1.33333333e-04]



CPU times: user 11min 56s, sys: 55.5 s, total: 12min 51s  
Wall time: 13min 40s

```
] print("111511239仇健安")
```

111511239仇健安

## 五、結論與心得

透過這次 Mini Project，我實際實作了從傳統到現代的多種解碼技術，並學習到如何將機器學習模型應用於通訊系統中的錯誤更正問題。特別是在訓練深度學習與 Autoencoder 模型時，我更了解如何設計合適的網路架構與加入通道雜訊以達到強健的效能。

整體來說，本專案大大加深了我對通訊系統與 AI 應用的理解與實作能力。