

기계학습을 활용한 부호화 기법 개발

Development of encoding techniques using machine learning.

팀 DC(Develop of enCoding)

20185218 김민혁 20185141 용권순

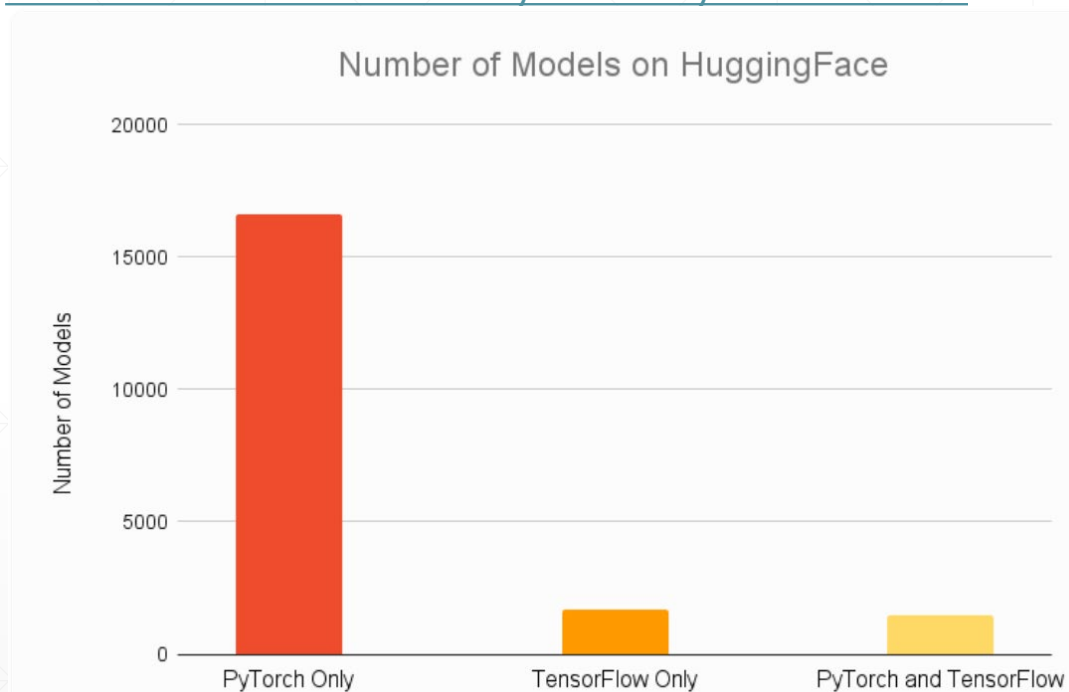
LAYOUT

- Introduction
 - Problem Environment
 - Simulation
 - Result
 - Expectation
-

Introduction

Introduction

- 본 프로젝트는 상용화된 5G통신의 LDPC Simulator를 pytorch로 구현 및 성능 검증을 목표로 한다.
- 현재 Keras기반의 물리 계층 오픈 소스 라이브러리가 등록되었음. [Sionna: An Open-Source Library for Next-Generation Physical Layer Research](#)



[그림 1] 딥러닝 Framework 사용자 비율

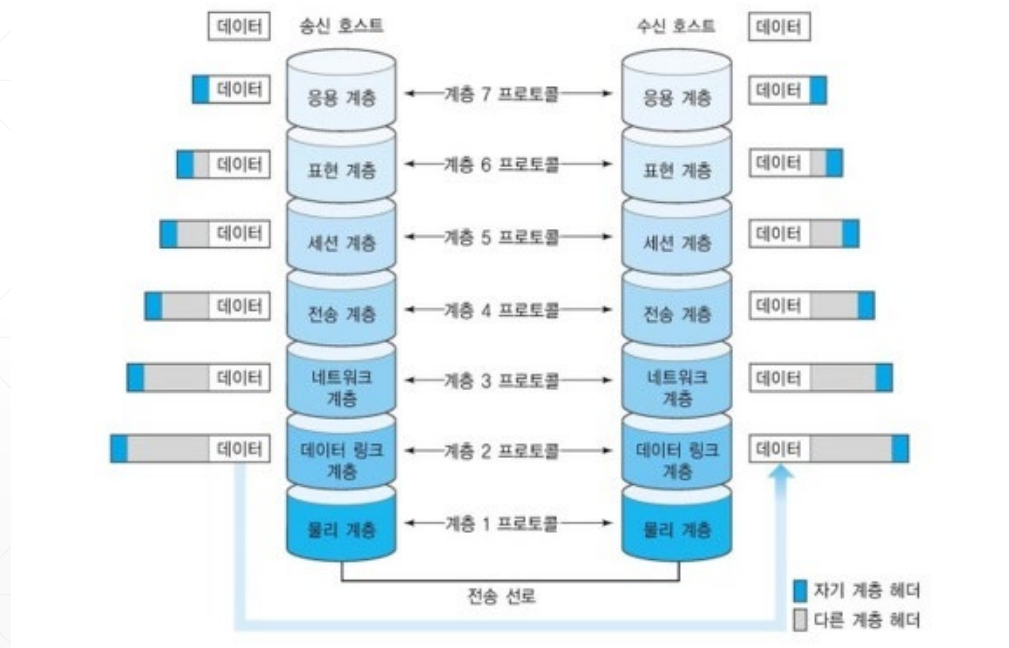
최근 딥러닝을 사용한 framework 조사 결과
pytorch를 사용한 연구가 tensorflow를 제치고
활발하게 연구되고 있다고 조사되어진다



Problem Environment

Problem Environment –OSI 7 layer

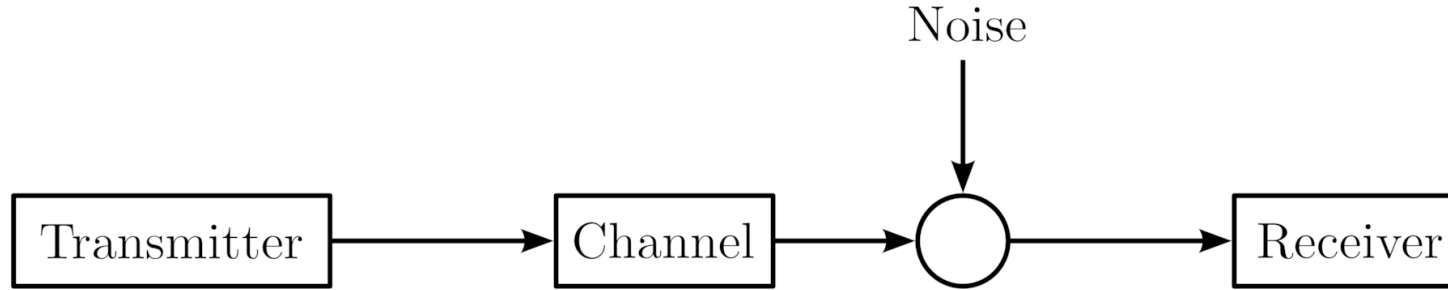
- 전송 데이터는 송신 호스트의 응용 계층에서 시작해 하위 계층으로 순차적으로 전달되어, 최종적으로 물리 계층에서 수신 호스트에 전달
- 본 과제는 최 하단의 물리 계층간 통신을 다룬다



[그림 2] OSI 7계층 모델

Problem Environment – Communication System

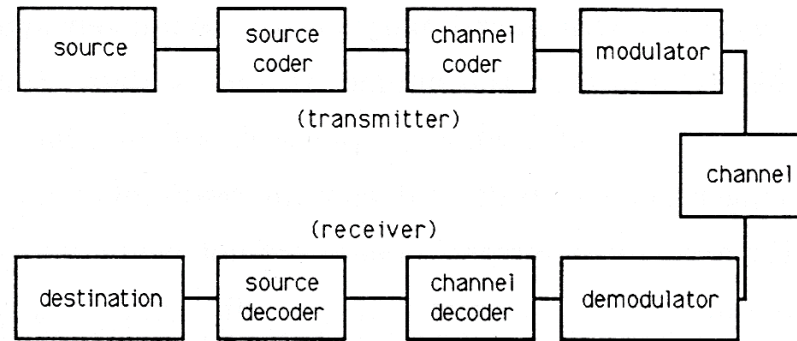
- 통신 시스템은 기본적으로 송신기(transmitter)와 전송매체(Channel), 수신기(Receiver)로 구성되어 있다.



- 메시지 M 을 송신기에 넣었을 경우 $s(t) = A(t)\cos[2\pi fc + \phi(t)]$ [그림 3] 통신 시스템
 - 채널을 통과한 신호 $r(t) = s(t) + n(t)$
 - 수신기에서 복조화를 진행한 원본 추정 신호 $\hat{m}(t)$ 를 원래 신호 $m(t)$ 복원
-

Problem Environment – Channel Coding

- 현재의 디지털 통신 시스템은 소스 코딩(source coding), 채널 코딩(channel coding)으로 나누어짐

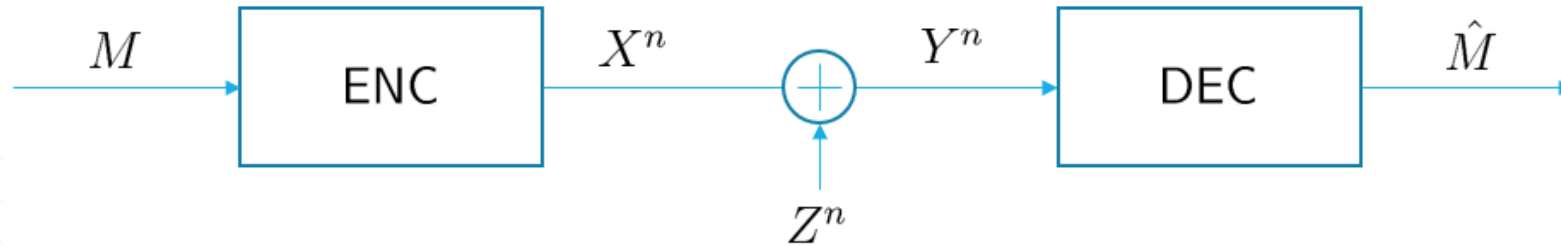


[그림 4] 디지털 통신 시스템 블록도

- 소스 코딩(source coding): 디지털 데이터를 압축하여 전송량을 줄이는 기법
 - 채널 코딩(Channel coding): 채널을 통과하면서 발생하는 오류를 수신기에서 검출하거나 정정할 수 있도록 송신기에서 의도적으로 데이터를 추가하는 기법
-

Problem Environment – AWGN Channel

- AWGN 채널: 대기중에 신호를 전송할 때 주로 표현되는 채널



[그림 5] AWGN 채널을 사용하는 무선 이동 통신 시스템

$$Y^n = X^n + Z^n$$

$$\text{Noise } Z^n: Z \sim N(0, \sigma^2)$$

Problem Environment – Evaluation

- 채널 코딩의 성능을 평가하는 척도인 SNR , BLER

$$SNR[dB] = 10\log \frac{S}{N} [dB]$$

S : 평균 신호 전력 $S = P_{s(avg)} = v_{s(rms)}^2$

N : 평균 잡음 전력 $N = P_{N(avg)} = v_{n(rms)}^2$

[그림 6] SNR

BLER

(Block Error Rate)

- 디지털 통신에서는 어떤 경우에도 통신을 위한 최소 SNR이 요구되며, SNR값이 커질 수록 BER는 증가
 - BLER(Block Error Rate) : 하나의 bit만 잘못 예측하더라도 전송한 메시지의 예측을 실패했다고 가정하는 오류 측정 기준
-

Simulation

Simulation– Hamming 7,4 code

- 1950년 미국 Bell 연구소의 Hamming이 고안한 간단한 선형 블록 부호

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

[그림 7] Hamming 7,4 code $G = [I, P], H = [P^T I]$

- Code word를 생성하는 G 와 오류 검출을 위한 H 행렬을 가짐

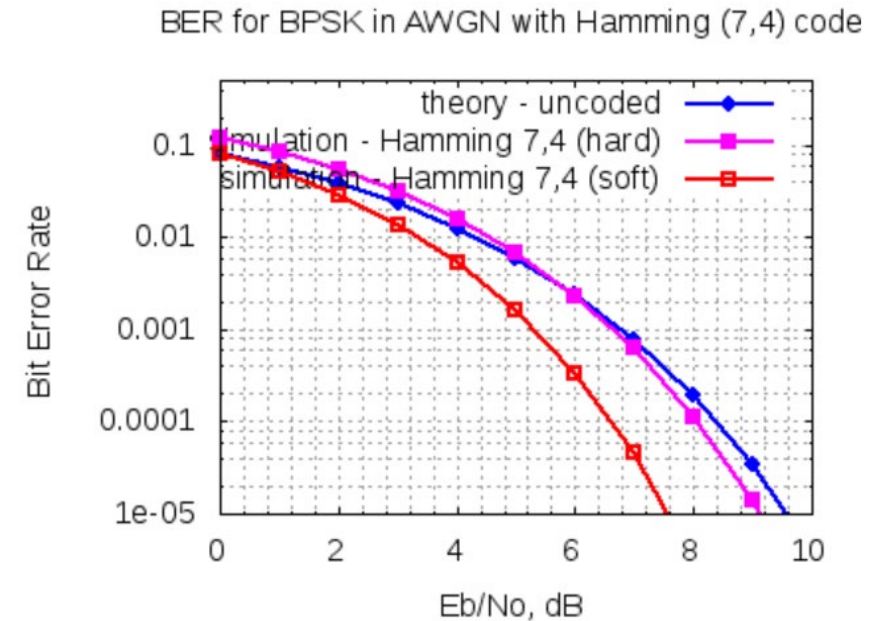
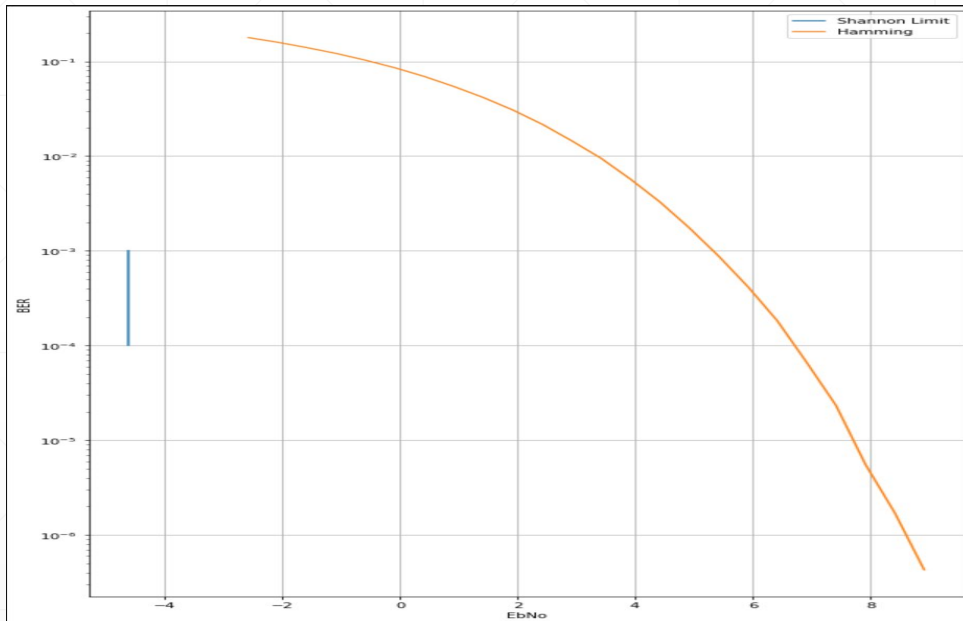
$$\begin{aligned} \hat{M} &= \operatorname{argmax}_{x \in \tilde{C}} \log p(y|x) \\ &= \operatorname{argmin}_{x \in \tilde{C}} |y - x|^2 \\ &= \operatorname{argmax}_{x \in \tilde{C}} y^T x \end{aligned}$$

[그림 8] ML Decoding

- ML(Maximum Likely hood)를 이용하여 수신 받은 신호 $r(t)$ 를 $\hat{m}(t)$ 메시지로 예측
-

Simulation- Hamming 7,4 code

- Pytorch를 이용하여 Hamming 7,4 code를 구현
- 성능의 유사도를 확인



[그림 9] pytorch를 이용해 구현한 Hamming 7,4 code vs MATLAB으로 구현된 Hamming (7,4) code

Simulation– 5G NR LDPC Code

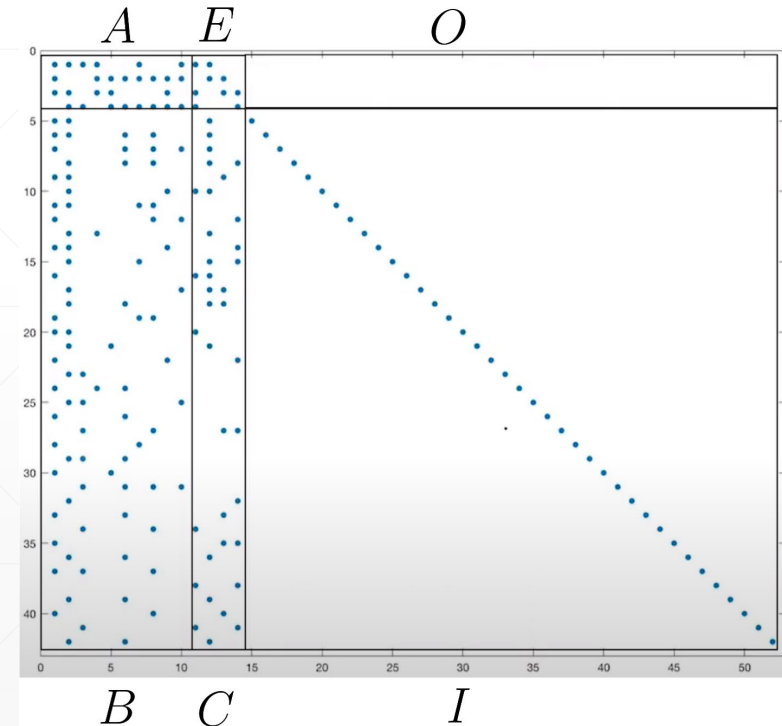
- 5G NR 이동 무선 통신의 표준 코드 중 하나
- 대부분의 원소가 0으로 이루어진 희소 행렬을 Parity check matrix로 사용.
- LLR을 사용해 효율적으로 bit확률을 계산 가능

$$L(\mu) = \log \frac{\mu(0)}{\mu(1)}$$

$$\mu(0) = \frac{e^{L(\mu)}}{1 + e^{L(\mu)}} \quad \text{and} \quad \mu(1) = \frac{1}{1 + e^{L(\mu)}}$$

$$L(\mu_{a \rightarrow b}) := L_{a \rightarrow b}$$

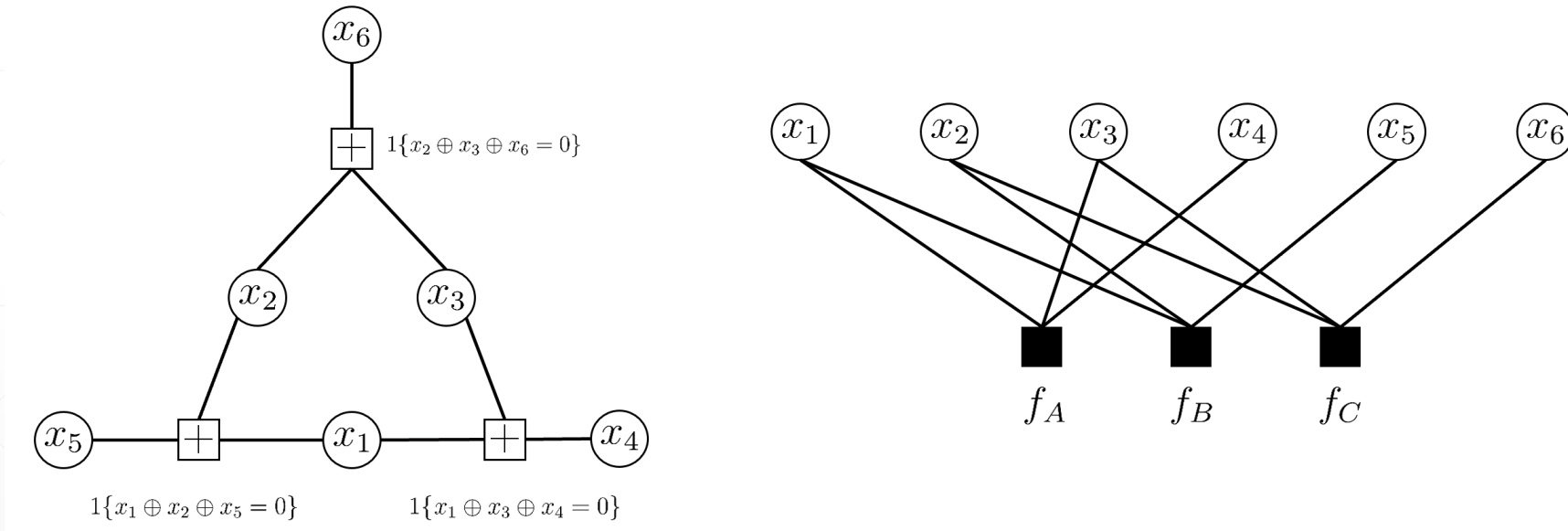
[그림 10] LLR(Log Likelihood Ratio)



[그림 11] LDPC Base Graph

Simulation– 5G NR LDPC Code

- Tanner Graph를 이용한 Sum Product Algorithm으로 LLR을 계속해서 update
- Sum Product Algorithm: Tanner graph에서 서로 연결되어 있는 node가 정보를 update 하는 방식으로 몇몇data가 소실 되어도 기존의 정보로 유추 가능

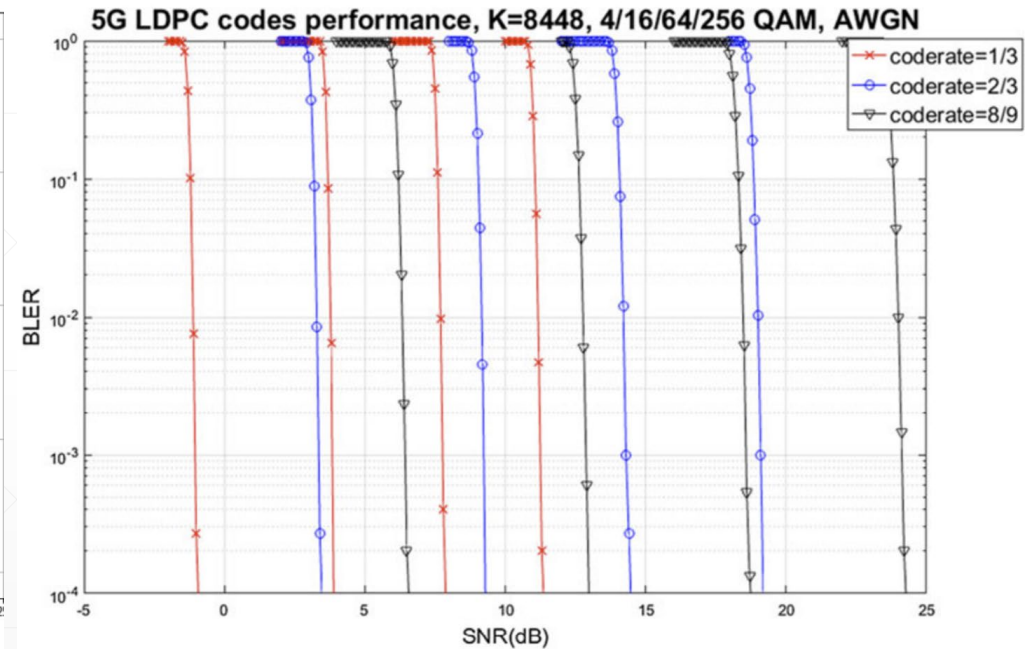
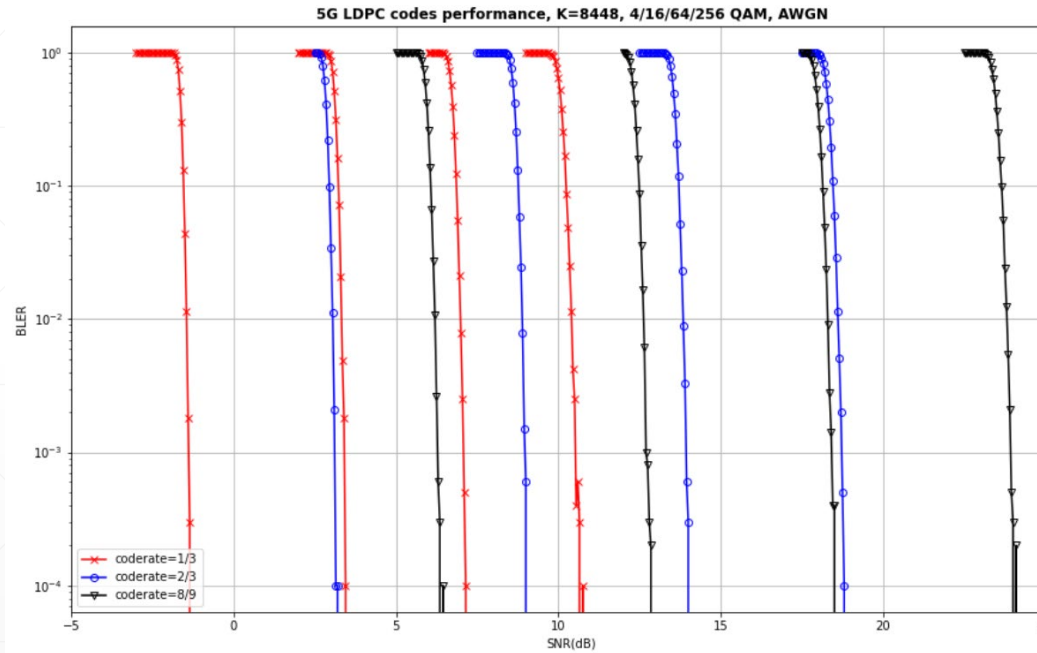


[그림 11] Hamming (7,4) code를 Tanner graph로 표현

Result

Result – 5G NR LDPC Code

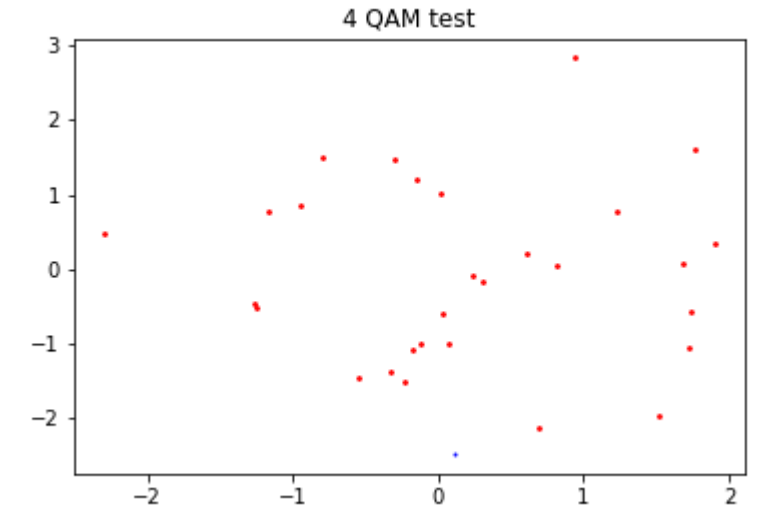
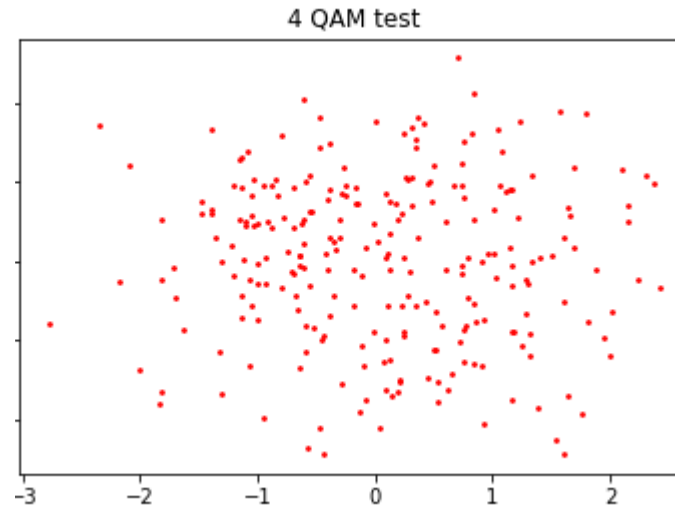
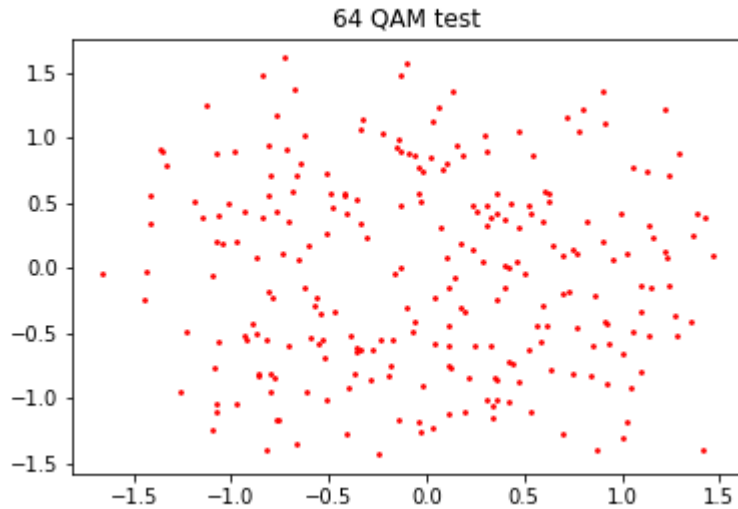
- Pytorch를 이용하여 LDPC Code를 구현
- BG1: 46×68 환경에서 각각 4,16,64,256 QAM의 BLER값을 도출



[그림 10 좌: pytorch로 구현된 LDPC Code 우: MATLAB으로 구현된 논문에서의 LDPC Code

Result– 5G NR LDPC Code

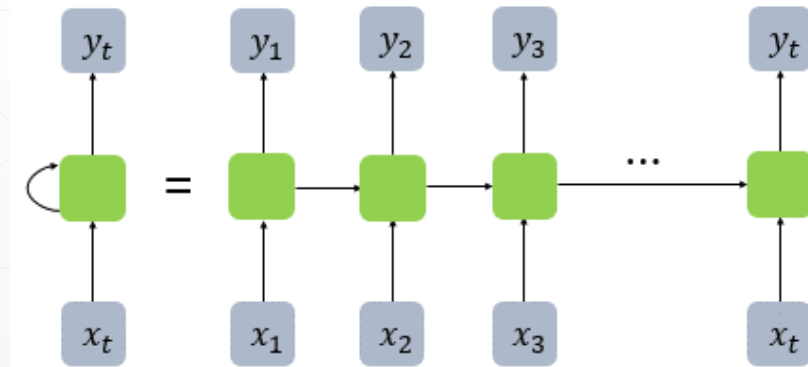
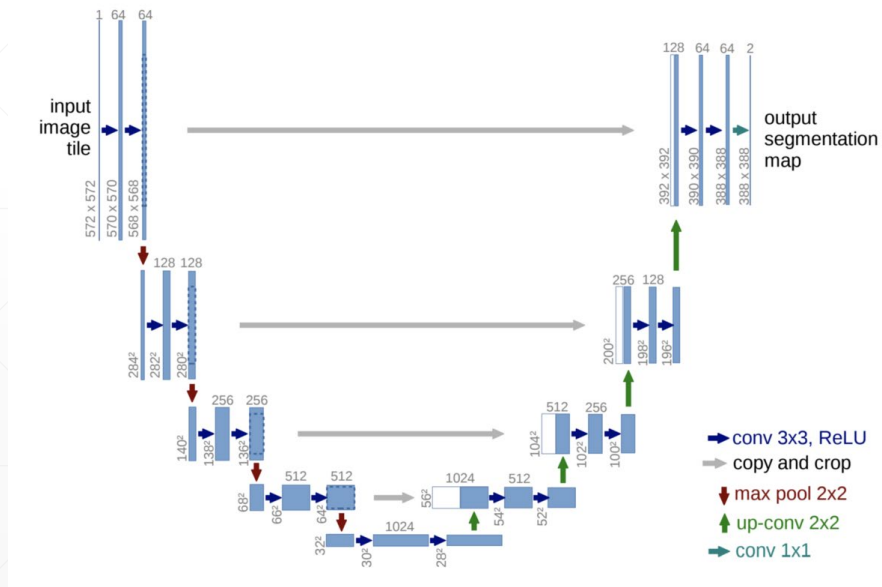
- 64 ,4QAM상황에서 고정된 SNR값을 기준으로 Sum product algorithm을 사용해 noise 섞인 message를 복원



Expectation

Expectation – Connection with DNN

- 본 프로젝트는 물리계층 시뮬레이션환경과 딥러닝 환경을 통일시켜 기존의 비효율적인 연구 환경 개선, python 환경에서 5G 무선 통신체계를 포함한 다양한 통신기술 연구의 기반기술을 확보한다는 의의를 가짐
- 구체적으로 현재 pytorch로 구현된 LDPC 시뮬레이터를 사용하여 DNN 모델인 Unet과 RNN을 사용하여 Code Rate을 낮추는 시도 중



[그림 13] DNN Unet구조와 RNN 구조

Thank you!