

# Pruning을 이용한 그래프 신경망 기반 저복잡도 채널 복호기

장우람, 이나영, 박호성\*

전남대학교 컴퓨터정보통신공학과, \*ICT 융합시스템공학과

jwr0425@naver.com, eao3116@naver.com, \*hpark1@jnu.ac.kr

## Graph Neural Network-based Low-Complexity Channel Decoder using Pruning

Wooram Jang, Nayoung Lee, Hosung Park\*

Chonnam National University

요약

본 논문에서는 두 가지 pruning 기법을 사용하여 그래프 신경망(Graph Neural Network: GNN) 기반 복호기를 개선하는 방안을 제시하며, 성능을 비교 및 분석한다. 최근 딥러닝 기반의 복호 기술이 활발히 연구되고 있으며, 그래프 신경망(Graph Neural Network: GNN) 기반 복호 또한 제안된 바 있다. GNN 기반 복호기는 기존 신뢰 전파(Belief propagation: BP) 알고리즘 기반 복호기에 비해 상대적으로 높은 신뢰성을 보장하면서도 블록 길이에 대해 확장성을 가진다. pruning을 통해 오류를 성능 열화를 최소화하면서도 GNN 기반 복호기의 가중치를 약 50% 줄이는데 성공하였다.

### I. 서론

현재 딥러닝 기반 복호 기술의 연구가 활발히 진행되고 있으며, 다양한 선형 블록 코드 환경에서 GNN 기반 복호의 가능성이 발견되었다[1]. 복호기의 성능은 직접적으로 통신의 품질에 영향을 미친다. 복호기는 수신 측에서 받은 비트열을 해석하기 위한 필수적인 부분으로, 수신된 신호를 해독하여 오류를 검출 및 정정하여 송신된 신호를 예측한다. 복호기의 오류 정정 능력의 개선은 낮은 비트 오류율, 즉 높은 신뢰성을 의미한다. GNN 기반 복호기는 기존 BP 기반 디코더에 비해 적은 반복 횟수만으로 낮은 비트 오류율을 보여주었다. 또한 블록 길이와 차원성에 구애받지 않는다는 장점을 가지고 있다.

하지만 GNN 기반 복호기는 이웃 노드의 정보를 종합하는 aggregation 과정 특성상 높은 연산 복잡도를 가진다. 총 9640개의 학습 가능한 매개변수를 통해 복호를 수행한다. 하드웨어 자원이 제한된 환경에서의 적용을 위해선 모델의 경량화가 필수적으로 요구된다. 모델의 경량화의 목적은 최대한 성능을 유지하며 모델의 크기를 줄여내는 것이다. 본 논문에서는 unstructured와 structured, 두 가지 방식의 pruning을 사용하여 GNN 기반 복호기를 경량화하고자 하였다.

### II. 본론

BP 기반 복호기는 반복적인 합-곱 알고리즘(sum-product algorithm: SPA)을 통해 확률값을 갱신하며 복호를 수행한다. Tanner 그래프를 통해 복호 과정을 표현할 수 있다. Tanner 그래프는 체크 노드, 변수 노드, 각 노드를 잇는 간선으로 구성된다. 패리티 체크행렬의 행이 체크 노드, 열이 변수 노드를 의미한다. 패리티 체크행렬의 원소가 1이라면 해당 행과 열을 잇는 간선으로 표현할 수 있다. 복호는 체크 노드 갱신 후 변수 노드의 값을 갱신하는 순서로 진행된다.

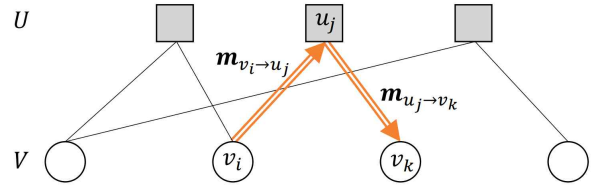


그림 1. GNN 기반 복호 과정 예시 [1]

### 2.1. GNN 기반 복호기 구조

그래프 신경망 기반 복호기와 BP 기반 복호기의 차이점은 SPA 알고리즘이 아닌 MLP(Multi layer perceptron)를 통해 복호를 수행하는데 있다.

- $m_{v_i \rightarrow u_j} = f^{F_m}([h_{v_i} || h_{u_j}], \theta_{m_{v \rightarrow u}})$  : 변수 노드에서 체크 노드로의 메시지 값을 업데이트하기 위한 함수
- $m_{u_j \rightarrow v_i} = f^{F_m}([h_{u_j} || h_{v_i}], \theta_{m_{u \rightarrow v}})$  : 체크 노드에서 변수 노드로의 메시지 값을 업데이트하기 위한 함수
- $h'_{u_j} = f^{F_n}([h_{u_j} || \frac{1}{|V(u_j)|} \sum_{v_i \in V(u_j)} m_{v_i \rightarrow u_j}], \theta_u)$  : 체크 노드의 임베딩 벡터를 업데이트하기 위한 함수
- $h'_{v_i} = f^{F_n}([h_{v_i} || \frac{1}{|U(v_i)|} \sum_{u_j \in U(v_i)} m_{u_j \rightarrow v_i}], \theta_v)$  : 변수 노드의 임베딩 벡터를 업데이트하기 위한 함수

$[x || y]$ 는  $x$ 와  $y$ 의 임베딩 벡터 연결,  $\theta$ 는 학습 가능한 매개변수를 의미한다. 체크 노드를 업데이트하기 위해 이웃한 변수 노드와의 메시지를 모두 계산 및 종합하여야 한다. 이후 자신의 임베딩과 연결하여 MLP를 통과시키면 임베딩 벡터를 구할 수 있다. 변수 노드 또한 동일하다.

## 2.2. pruning 방식

본 논문에서는 unstructured와 structured, 두 가지 방식을 사용했다.

- Unstructured method : 가중치를 개별적으로 제거한다. 절댓값을 기준으로 중요도를 판별하여 중요도가 낮은 것부터 비활성화한다. 학습 후 한 번 pruning 하여 BER 성능을 측정하였다.
- Structured method : 뉴런을 제거하는 방식으로 가중치를 비활성화한다. 가중치 절댓값의 합을 기준으로 중요도를 판별한다. 학습과 pruning을 반복하며 성능 하락을 최소한으로 유지하였다. 첫 번째 학습 후 3120개 가중치 제거, 두 번째 학습 후 1680개의 가중치를 제거하여 BER 성능을 측정하였다.

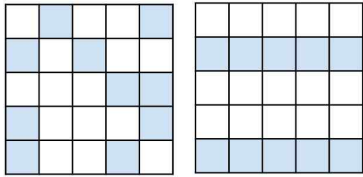


그림 2. Unstructured, Structured pruning 예시

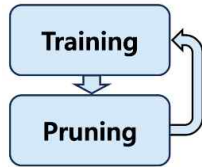


그림 3. Structured pruning 진행 방식

## 2.3. 실험 결과

(63, 45) BCH 코드에 대한 복호 성능을 평가하기 위해 이진 위상천이변조된 신호를 가산성 백색 가우시안 잡음 채널에 통과시켜 BER 성능을 비교하였다. 각각의 신호대잡음비 환경에서 BP, basic GNN, pruning GNN 복호기의 BER 값을 그래프로 확인할 수 있다.

### 2.3.1. Unstructured pruning

총 1024개, 10.6% 가중치를 비활성화하였다. 그래프 신경망 기반 복호기에 사용된 MLP의 종류에 따라 다른 비율을 적용하여 pruning을 수행하였다. 이때, 노드를 임베딩하는 MLP의 pruning 비율을 높이는 것이 성능 유지에 중요하였다.

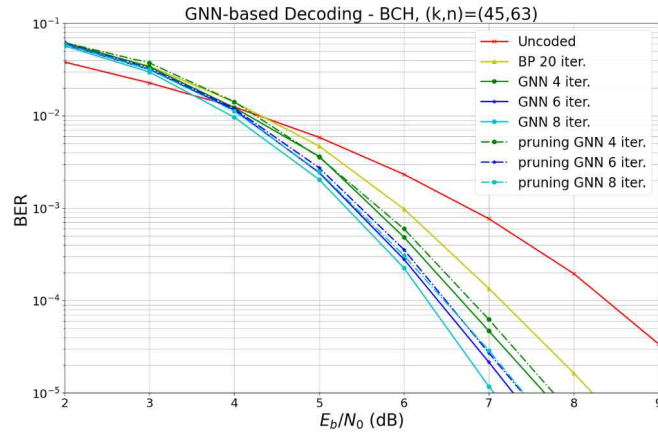


그림 4. Unstructured pruning 여부에 따른 BER 성능 비교

MLP 종류	layer number	pruning ratio(%)	pruning count
$MLP_{m_{v \rightarrow u}}$	1	10	160
	2	2	16
$MLP_{h'_u}$	1	20	320
	2	2	16
$MLP_{m_{u \rightarrow v}}$	1	10	160
	2	2	16
$MLP_{h'_v}$	1	20	320
	2	2	16

표1. Unstructured pruning, layer별 pruning 비율 및 가중치 수

## 2.3.2. Structured pruning

총 4800개, 49.7% 가중치를 비활성화할 수 있었다. 같은 비율로 pruning을 진행하더라도 큰 성능 하락은 관찰되지 않았다. Unstructured pruning과 비교하여 더 많은 가중치를 비활성화했음에도 불구하고 오히려 더 높은 성능을 관찰할 수 있었다.

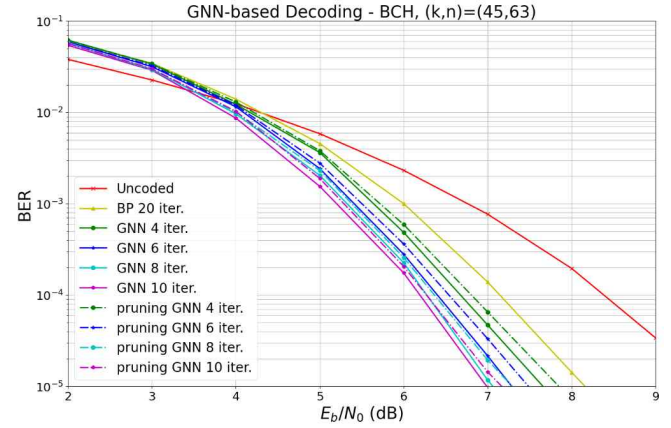


그림 5. Structured pruning 여부에 따른 BER 성능 비교

MLP 종류	layer number	pruning ratio(%)	pruning count
$MLP_{m_{v \rightarrow u}}$	1	50	800
	2	50	400
$MLP_{h'_u}$	1	50	800
	2	50	400
$MLP_{m_{u \rightarrow v}}$	1	50	800
	2	50	400
$MLP_{h'_v}$	1	50	800
	2	50	400

표2. Structured pruning, layer별 pruning 비율 및 가중치 수

## III. 결론

두 가지 방식의 pruning을 통해 실험 결과를 도출하고, structured pruning을 통해 최종 49.7% pruning을 성공하였다. 기존 대비 절반의 가중치만으로 거의 동일한 성능을 유지할 수 있었다. 행렬 계산 시 약 50%의 연산량 감소가 기대되며 GNN 기반 복호기의 높은 계산 복잡도를 효과적으로 개선했다고 볼 수 있다. 모델의 경량화를 통해 실질적인 학습 및 추론 시간의 향상, 제한된 하드웨어 자원 환경에서의 활용을 기대해볼 수 있다. 추후 pruning 기준과 비율, 학습 파라미터 변경 등의 조정을 통해 보다 효과적인 방법을 찾는 연구가 가능할 것이다.

## ACKNOWLEDGMENT

본 연구는 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 ICT혁신인재4.0 사업(IITP-2023-RS-2022-00156385), 6G/B5G xURLLC를 위한 유연한 신뢰도의 채널코딩(No. 2021001016)과 인공지능 혁신 허브 연구 개발(No.2021-0-02068)의 연구결과로 수행되었음.

## 참고 문헌

- [1] Cammerer, S., Hoydis, J., Aoudia, F. A., and Keller, A. "Graph neural networks for channel decoding", IEEE Globecom Workshops, pp. 486-491, Dec 2022.
- [2] Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. "Rethinking the value of network pruning", arXiv preprint arXiv:1810.05270, Mar 2019.