Pruning을 적용한 Transformer 기반 채널 복호기

황우진, 조은우, 박호성*

전남대학교 컴퓨터정보통신공학과, *ICT 융합시스템공학과

ruddy_1632@hanmail.net, twins2937@naver.com, *hpark1@jnu.ac.kr

Transformer-based Channel Decoder using Pruning

Woojin Hwang, Eunwoo Jo, Hosung Park*

Chonnam National University

요 약

본 논문에서는 Transformer 기반 복호기의 복잡도를 감소시키기 위해 Pruning 기술을 도입하고, 이를 통한 성능 비교 및 분석을 제시한다. 현재까지 딥러닝을 기반으로 한 복호 기술이 적극적으로 연구되고 있으며, Transformer 기반 복호기 역시 연구된 바 있다. 해당 연구에서 기존의 신뢰 전파(Belief propagation)알고리즘을 기반으로 한 복호기와 비교하여 Transformer 기반 디코더가 높은 신뢰성을 보여준다는 점을 확인했다. 우리는 pruning을 통해 transformer 기반 복호기의 가중치를 최대 40% 감소시키면서도 성능 저하를 최소화하는 데에 성공하였다.

Ⅰ. 서 론

차세대 통신 시스템에서의 고속 데이터 전송을 위해서는 빠르고 정확한 오류정정부호가 필수적이다. 최근 딥러닝 기반 복호기들의 연구가 진행되고 있는데, Transformer 기반 복호기도 연구된 바 있다. Transformer는 self-attention을 통해 단어 간의 상호작용 잘 파악할 수 있어 자연어 처리에서 높은 성능을 보이고 있다. 이러한 Transformer의 특성을 채널 복호기에 적용하면 수신된 비트들 간의 복잡한 관계를 잘 파악할 수 있다. 하지만 Transformer 기반 복호기는 self-attention layer에서 아주 많은 parameter를 사용하기 때문에 계산 복잡도와 메모리 사용량을 키우고, 이는 Transformer 기반 복호기를 실제 통신 기술에 적용하기 어렵게 한다. Transformer 기반 복호기가 통신 시스템에 적용되기 위해서는 모델의 경량화를 통해 계산 복잡도와 메모리 사용량을 줄이면서, 신뢰성을 보장해야 한다. 본 논문에서는 복호기의 multi-head self-attention layer와 feed forward 신경망에 pruning 기술을 적용하여 Transformer 기반 복호기를 경량화하고자 하였다.

Ⅱ. 본론

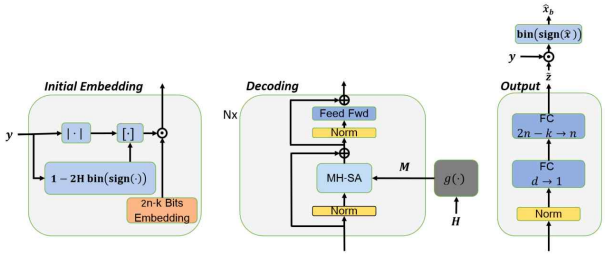

그림 1. Transformer 기반 복호기 구조

2.1 Transformer 기반 복호기 구조

Transformer 기반 복호기의 구조는 initial embedding, decoding layer, output layer 세 가지로 나뉜다.

$\langle \phi_i, \phi_j \rangle = \begin{cases} |y_i||y_j|\langle W_i, W_j \rangle & i,j \le n \\ |y_i|(1-2(s(y))_{j-n+1})\langle W_i, W_j \rangle, & i \le n < j \end{cases}$: 채널을 거친 입력 값의 magnitude와 syndrome의 결합한 값을 2n-k 길이로 임베딩한 함수

$$A(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d}})V$$

-query, key, value를 이용한 attention 함수

$$g(H) : \{0,1\}^{(n-k)\times k} \rightarrow \{-\infty, 0\}^{2n-k\times 2n-k}$$

-패리티 체크 행렬의 특성을 반영한 마스킹

$$A_H(Q,K,V) = \text{Softmax}\left(\frac{QK^T + g(H)}{\sqrt{d}}\right)V.$$

- 마스킹을 추가한 attention 계산

초기 임베딩 후 decoding layer의 입력 값이 MH-SA(Multi Head-Self-attention) 연산을 통해 비트들 간의 상관관계를 학습하고 feed forward 신경망을 거친다. 이 과정에서 잔차 연결과 정규화를 거쳐서 기울기 소실을 최소화하고 학습을 빠르게 진행시킨다.

$$\hat{x_b} = bin(sign(f_\theta(y) \bullet y))$$

- 추정한 잡음이 제거된 함수

f_θ는 모델의 함수 값을 의미한다. output layer에서 두 번의 FC layer(fully-connected layer)를 통해 차원의 크기를 축소한다. 첫 번째 layer에서는 1차원의 2n-k 벡터로 축소하고, 두 번째 layer에서는 soft decoding 된 잡음 z 값인 n 벡터를 추출한다. 이후 cross entropy loss function을 통해 학습을 진행한다.

2.2 Pruning 방식

우리는 ECC Transformer 모델을 경량화하기 위해 decoder 구조에 pruning을 적용하였다. pruning은 중요한 파라미터는 유지하고 0에 가까운 파라미터를 제거하는 기법으로 네트워크 성능이 크게 저하되지 않는 선에서 파라미터 개수를 줄인다. decoder 구조의 multi-head self-attention과 feed forward network에 대하여 pruning을 amount 10%부터 50% 까지 10%씩 증가시키며 적용하였다. pruning ratio 는 제거할 파라미터의 백분율이며 10%부터 50%까지 적용했을 때 모델 전체 파라미터의 8%, 16%, 24%, 32%, 40%를 제거할 수 있다. 모델 전체에서 파라미터를 사용하는 부분은 decoder와 output layer이다. decoder layer에만 pruning을 적용한 이유는 output layer의 경우 decoder layer에 비해 파라미터 개수가 적고 압축되어진 중요한 정보를 담고 있기 때문에 pruning을 적용하면 모델의 성능이 하락할 수 있기 때문이다.

실험은 구체적으로 LDPC 부호 중 전체 비트 수의 길이가 49, 데이터 비트 수가 24인 부호에 대하여 실험을 진행하였다. pruning ratio 값 이외의 하이퍼파라미터 값은 동일하게 진행했으며 모델을 학습할 때 epoch은 50으로 하고 batch 사이즈는 128로 하여 실험했다. multi-head attention을 할 때 head의 수는 8개로 하였고, attention을 수행하는 decoder layer는 2개, 각 self-attention의 특성 차원은 32개로 하였다.

2.3 실험 결과

(49,24) LDPC 코드에 대한 복호 성능을 평가하기 위해 BER 성능을 비교하였다. SNR (신호 대 잡음비)를 4, 5, 6으로 설정하여 해당 수치만큼 잡음을 생성하여 송신비트에 더하면 수신비트가 만들어지고 이러한 비트를 오류 정정하도록 하였다. (49,24)에 대해서만 실험을 적용하면 장부호에서의 pruning 모델의 성능을 증명할 수 없으므로 (121,60) LDPC 부호에 amount 값을 30%로 하여 전체 파라미터의 24%를 제거하는 실험을 한 차례 더 진행하였다.

	n=49 k=24						N=121, k=60	
SNR	기존 모델	8%	16%	24%	32%	40%	기존 모델	24%
4	0.0144	0.0146	0.0145	0.0146	0.0148	0.0150	0.0121	0.0109
5	0.00385	0.00389	0.00390	0.00393	0.00402	0.00413	0.001250	0.000999
6	0.0006	0.000609	0.000612	0.000626	0.000661	0.000688	0.0000398	0.0000263

표1. 파라미터 제거에 따른 BER 값

아래는 기존 모델과 비교하여 BER 감소 값을 백분율로 정리한 표이다. 감소 값이기 때문에 음수 값은 오히려 BER이 감소하여 모델의 성능이 향상됨을 의미한다. (49,24) LDPC 부호에서 파라미터는 최대 40%의 파라미터를 감소시켜 모델의 복잡도를 감소했지만 SNR에 따라 BER 감소가 적게 일어남을 확인할 수 있다. 추가 실험을 진행한 (121,60) 부호에서는 오히려 모델의 성능이 향상된 결과를 확인할 수 있다.

	n=49 k=24					n=121 k=60
SNR	8%	16%	24%	32%	40%	24%
4	1.389%	0.694%	1.389%	2.778%	4.167%	-9.917%
5	1.039%	1.299%	2.078%	4.416%	7.273%	-20.080%
6	1.500%	2.000%	4.333%	10.167%	14.667%	-33.920%

표2. 파라미터 제거에 따른 BER 감소 백분율

아래 그림은 표1을 BER curve로 나타낸 것이다. pruning을 전체 파라미터의 8%, 16%, 24%, 32%, 40% 진행했을 때 계산 복잡도는 감소하지만 BER curve는 비슷하게 유지된다는 사실을 확인할 수 있다.

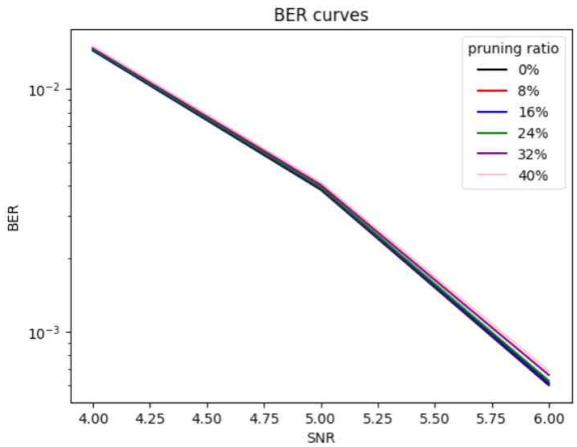

그림2. (49,24) LDPC 부호의 BER 그래프

Ⅲ. 결론

Pruning ratio를 변경하며 pruning을 진행하여 기존 대비 약 40%의 가중치로 성능이 거의 동일하게 유지되었다. 또한 한 차례 진행한 장부호의 pruning에서는 overfitting이 감소하여 성능이 증가하는 결과도 나왔다. 이는 pruning 등을 통한 신경망 가속화가 Transformer 기반 복호기의 높은 계산 복잡도와 메모리 요구량을 감소시킬 뿐 아니라 성능을 향상시킨다고 볼 수 있다. 이러한 모델의 경량화를 통해 실제 통신기술에의 적용을 기대해 볼 수 있다. 추후 fine-tuning을 통한 전이 학습을 적용하여 빠르게 학습을 진행 시키거나, 더 효과적인 Transformer 모델의 구조를 연구하여 성능을 향상시키는 연구가 기대된다.

ACKNOWLEDGMENT

본 연구는 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 ICT혁신인재4.0사업 (IITP-2023-RS-2022-00156385), 6G/B5GxURLLC를 위한 유연한 신뢰도의 채널코딩(No. 2021001016)과 인공지능혁신허브연구개발(No.2021-0-02068)의 연구결과로 수행되었음.

참 고 문 헌

[1] Yoni Choukroun,Lior Wolf,"Error Correction Code Transformer",arXiv:2203.14966, 2022.

[2] Amir Bennatan, Yoni Choukroun, and Pavel Kisilev. Deep learning for decoding of linearcodes-a syndrome-based approach. In 2018 IEEE International Symposium on InformationTheory (ISIT), pages 1595-1599. IEEE, 2018.

[3] 박호성, 노종선, "차세대 통신 시스템을 위한 오류 정정 부호," 『한국통신학회지』, 제 29권, 8호, pp. 26-33. 2012년 8월.

[4] Seong-Joon Park, Hee-Youl Kwak, Sang-Hyo Kim, Sunghwan Kim, Yongjune Kim, Jong-Seon No, How to Mask in Error Correction Code Transformer: Systematic and Double Masking, arXiv:2308.08128, 2023.