

학습 가능한 가중치를 활용한 레이어드 신경망 신뢰 전파 복호

조예지, 김정현, 송홍엽*

세종대학교, 연세대학교*

yezy0390@sju.ac.kr, j.kim@sejong.ac.kr, hysong@yonsei.ac.kr*

Layered Neural BP Decoding with Learnable Weights

Yeji Cho, Junghyun Kim, Hong-Yeop Song*

Sejong Univ., Yonsei Univ.*

요약

본 논문에서는 무선 통신 시스템에서의 향상된 복호 성능을 달성하기 위해, 딥러닝 기반 BP 디코더에 학습 가능한 추가 가중치를 도입하고, 레이어드(layered) 스케줄링 기법을 적용한 새로운 모델을 제안한다. 제안한 디코더는 기존 연구에서 고려되지 않았던 검사-변수 노드 간 메시지 업데이트에도 가중치를 부여하여, 더 정교한 메시지 전달 학습이 가능하도록 하였다. 또한 테스트 단계에서는 레이어드 스케줄링을 적용함으로써 수렴 속도를 높이고 복호 성능을 향상시켰다. 실험 결과, 제안 모델은 기존 디코더에 비해 우수한 비트 오류율(bit error rate)을 통해 향상된 성능과 빠른 수렴 속도를 동시에 달성함을 확인하였다.

I. 서론

최근 딥러닝은 객체 인식 및 자연어 처리와 같은 다양한 응용 분야에서 큰 성공을 거두고 있다[1]. 통신 시스템에서는 딥러닝의 성공에 영감을 받아 딥러닝을 활용하여 오래된 문제를 해결하려는 많은 노력이 있었다[2]. 특히 채널 코딩의 경우 신경망 belief propagation(BP) 디코더가 high density parity check(HDPC) 코드에 대해 기존 BP 디코더의 오류 정정 성능을 향상시킬 수 있음이 나타났다[3]. 이 논문에서는 태너(Tanner) 그래프 예지에 가중치를 부여하여 소프트 태너 그래프를 사용하였으며, 실험 결과 클래식 BP 디코더에 비해 향상된 성능을 보여주었다.

최근에는 BP 디코더의 수렴 속도를 높이고 더 적은 반복 횟수를 사용하여 처리량을 늘리기 위한 다양한 스케줄링 기법에 대한 관심이 높아지고 있다. BP 디코더에서 자주 사용되는 대표적인 스케줄링 기법에는 레이어드(layered) 기법이 있다. 일반적인 플러딩(flooding) 스케줄링은 모든 반복에서 먼저 모든 검사-변수 메시지를 업데이트한 다음 모든 변수-검사 메시지를 업데이트한다. 반면, 레이어드 스케줄링은 하나 또는 특정 수의 선택된 검사 노드에 연결된 변수 노드의 전체 집합을 업데이트하는 식으로 계층을 거듭한다. 이러한 기법을 통해 복잡도가 낮아지며, 수렴 속도는 증가한다[4].

본 논문에서는 선행 연구[3]의 신경망 BP 디코더에 추가적인 가중치를 도입하고, 레이어드 스케줄링[4]을 적용함으로써 디코딩 성능을 향상시킨 모델을 제안한다. 제안 모델은 학습 과정에서 기존 모델에서는 고려하지 않았던 검사-변수 노드 간 메시지 전달에 가중치를 부여하여, 이들 메시지의 중요도를 효과적으로 학습할 수 있도록 하였다. 또한 테스트 단계에서는 학습된 모델에 레이어드 스케줄링을 적용하여 디코딩 효율을 높였다. 실험 결과, 제안 모델은 BCH(63,36) 및 BCH(63,45) 코드에 대해 SNR 2dB부터 6dB 구간에서 기존 모델보다 높은 수렴 속도와 우수한 디코딩 성능을 보였다.

II. 본론

본 논문은 선행 연구와 동일한 무선 통신 환경을 가정한다. 훈련 과정에서는 전송 코드워드로 제로 코드워드(zero codeword)를 사용하고, 테스트

과정에서는 랜덤하게 생성된 유효한 코드워드를 사용한다. 각 코드워드는 binary phase shift keying(BPSK) 변조를 거친 후 additive white gaussian noise(AWGN) 채널을 통과하며, 수신 신호는 로그 우도비(log-likelihood ratio, LLR)로 변환되어 디코더의 입력으로 사용된다.

제안 모델은 훈련 단계에서 격자 형태의 태너 그래프 구조를 따르며, BP 알고리즘의 일반적인 스케줄링 방식인 플러딩 스케줄링을 적용하여 각 반복(iteration)마다 모든 메시지를 동시에 업데이트한다. 학습이 완료된 후 테스트 단계에서는 레이어드 스케줄링을 적용하여 디코딩의 성능과 수렴 속도를 향상시켰다. 이때 각 층의 업데이트는 순차적으로 이루어지며, 이는 플러딩 방식보다 더 빠른 수렴과 낮은 복잡도를 기대할 수 있다.

먼저 훈련 과정에서, 제안하는 모델은 은닉층 i ($i = 1, 2, \dots, 2L$)를 고려한다. 여기서 L 은 기존 BP 알고리즘의 반복 횟수를 의미한다. 각 은닉층에서 처리되는 요소 중 하나를 $e = (v, c)$ 로 나타낼 수 있으며, 이때 해당 처리 요소의 출력 메시지는 $x_{i,e}$ 로 표기한다. 이러한 구조에 따라, 홀수 은닉층 i 및 $e = (v, c)$ 를 갖는 처리 요소에 대해 디코더의 변수-검사 업데이트는 다음과 같이 정의될 수 있다.

$$x_{i,e} = \tanh\left(\frac{1}{2}(w_{i,v}l_v + \sum_{e'=(v,c'), c' \neq c} w_{i,e,e'}^{v2c} x_{i-1,e'})\right), \quad (1)$$

여기서 l_v 은 v 의 LLR 메시지이고, 출력 메시지 $x_{0,e'}$ 은 모든 예지 e' 에 대해 0으로 초기화된다. $w_{i,v}$ 및 $w_{i,e,e'}^{v2c}$ 는 l_v 및 $x_{i,e}$ 의 학습 가능한 가중치이다. 다음으로 짝수 은닉층 i 및 $e = (v, c)$ 에 대한 검사-변수 업데이트 과정은 다음과 같다.

$$x_{i,e=(v,c)} = 2\tanh^{-1}\left(\prod_{e'=(v',c), v' \neq v} w_{i,e,e'}^{c2v} x_{i-1,e'}\right), \quad (2)$$

선행 논문에서는 검사-변수 업데이트 과정에는 별도의 학습 가능한 가중치를 부여하지 않았지만, 제안 모델에서는 예지 가중치 $w_{i,e,e'}^{c2v}$ 를 부여한다. 마지막으로 네트워크의 출력은 다음과 같다.

$$o_v = \sigma(w_{i,v}l_v + \sum_{e'=(v,c')} w_{i,v,e'}^{out} x_{2L,e'}), \quad (3)$$

여기서 $\sigma(x) \equiv (1 + e^{-x})^{-1}$ 는 시그모이드 함수를 나타내고, 출력 레이어에도 학습 가능한 가중치 $w_{i,v,e'}^{out}$ 를 부여한다.

다음으로 테스트 과정에서 제안 모델은 효율적인 디코딩을 위해 레이어드 스케줄링을 사용한다. 레이어드 스케줄링은 선택된 검사 노드에 연결된 변수 노드를 즉시 업데이트하는 순차적 방식으로, 반복마다 부분적으로 테너 그래프를 갱신하기 때문에 전체 격자 구조를 고정할 필요가 없다. 따라서 반복 $t(t = 1, 2, \dots, L)$ 에서 선택된 검사 노드 c 에 연결된 에지 집합에 대해 변수-검사 업데이트는 다음과 같이 표현할 수 있다.

$$z_e^t = w_{i,e,e'}^{v2c} (l_v - m_{c \rightarrow v}^{(t-1)}), \quad (4)$$

다음으로 검사-변수 업데이트 과정은 다음과 같다.

$$m_{c \rightarrow v}^t = w_{i,e,e'}^{c2v} (2 \tanh^{-1}(\prod_{e'=(v',c), v' \neq v} \tanh(\frac{z_{e'}^{(t)}}{2}))), \quad (5)$$

마지막으로 네트워크의 출력은 다음과 같다.

$$o_v^t = \sigma(l_v + \sum_{e=(v,c)} w_{i,v,e}^{out} m_{c \rightarrow v}^t), \quad (6)$$

모델 훈련 과정에서 제안하는 구조는 격자 기반의 10개 은닉층으로 구성되며, 이는 BP 알고리즘의 전체 반복 5회에 해당한다. 훈련은 총 100 에포크(epoch) 동안 진행되며, 각 에포크는 20개의 미니배치(mini-batch)로 구성된다. 하나의 미니배치는 SNR 1dB부터 6dB까지 각각 20개씩, 총 120개의 제로 코드워드 샘플을 포함한다. 모델 최적화에는 학습률 0.001의 RMSProp 옵티마이저를 사용한다. 또한 제안 모델의 훈련을 위해 이진 교차 엔트로피(binary cross entropy) 손실 함수를 사용하였다.

$$L(o, y) = -\frac{1}{N} \sum_{v=1}^N y_v \log(o_v) + (1 - y_v) \log(1 - o_v), \quad (7)$$

여기서 o_v , y_v 는 각각 모델의 출력과 전송된 코드워드의 실제 값이다.

테스트 과정에서는 격자 구조를 사용하지 않고, 레이어드 스케줄링 기법을 적용하여 총 5회의 반복만 수행한다. 이때 디코딩 성능 평가는 무작위로 생성된 유효한 코드워드에 대해 비트 오류율(bit error rate)을 기준으로 이루어지며, 프레임 오류율(frame error rate)이 100개에 도달할 때까지 반복 수행된다. 채널 코드는 BCH(63,36), BCH(63,45)를 사용하였으며, SNR 1dB부터 6dB까지의 실험 결과는 그림 1과 그림 2에 제시하였다. 실험 결과, 제안 모델은 기존 모델에 비해 눈에 띄는 성능 향상을 보였으며, 특히 높은 SNR 구간에서 더욱 우수한 디코딩 성능을 보이는 것을 확인하였다. 또한 표 1은 SNR 6dB에서 하나의 코드워드에 대해 기존 모델과 제안 모델의 테스트 시간을 비교한 결과를 나타낸다. 이를 통해 제안 모델이 기존 모델에 비해 현저히 빠른 수렴 속도를 가지며, 연산 효율성 측면에서도 우수함을 확인할 수 있었다.

III. 결론

본 논문은 기존 모델에 검사-변수 에지 가중치를 도입하고, 테스트 과정에서 레이어드 스케줄링을 적용한 새로운 신경망 BP 디코더를 제안하였다. 제안 모델은 훈련 시 플러딩 스케줄링을 기반으로 격자 구조 내에서 가중치를 학습하고, 테스트 시에는 레이어드 방식으로 효율적인 디코딩을 수행한다. 실험 결과, 본 모델은 BCH(63,36), BCH(63,45)에 대해 SNR 1dB부터 6dB 구간에서 기존의 클래식 BP 디코더 및 신경망 BP 디코더 대비 우수한 비트 오류율 성능을 달성하였으며, 연산 효율성도 향상되었다. 향후 연구에서는 본 모델을 다양한 채널 환경 및 코드 구조로 확장하

고, 다양한 동적 스케줄링 기법을 접목하는 방향으로 발전시킬 수 있을 것으로 기대된다.

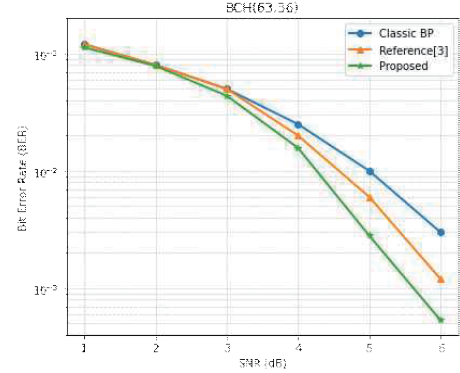


그림 1. BCH(63,36)에서 제안 모델과 기존 모델[3] 성능 비교

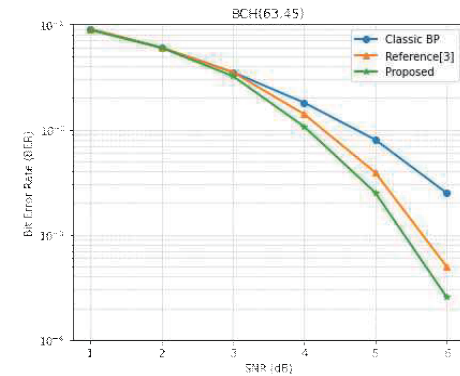


그림 2. BCH(63,45)에서 제안 모델과 기존 모델[3] 성능 비교

채널 코드	기존 모델[3]	제안 모델
BCH(63,36)	0.000035sec	0.000011sec
BCH(63,45)	0.000029sec	0.000008sec

표 1. SNR 6dB에서 제안 모델과 기존 모델[3]의 테스트 시간 비교

ACKNOWLEDGMENT

이 (성)과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.RS-2023-00209000).

참고 문헌

- [1] LeCun, Y. Bengio, Y., and Hinton, G. "Deep learning," nature, vol. 521, pp. 436-444, 2015.
- [2] Lee, H., Lee, B., Yang, H., Kim, J., Kim, S., Shin, W., Shim, B., and Poor, H. V., "Towards 6G Hyper-Connectivity: Vision, Challenges, and Key Enabling Technologies," Journal of Communications and Networks, vol. 25, no. 3, pp. 344-354, Jun. 2023.
- [3] Nachmani, E., Be'ery, Y., and Burshtein, D., "Learning to decode linear codes using deep learning," in Proc. 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, pp. 341-346, Sep. 27-30, 2016.
- [4] Rovini, M., Rossi, F., Ciao, P., L'Insalata, N., and Fanucci, L., "Layered Decoding of Non-Layered LDPC Codes," in Proc. 9th EUROMICRO Conference on Digital System Design (DSD'06), Cavtat, Croatia, pp. 537-544, Aug. 30-Sep. 01, 2006.