2. A Model-Driven Deep Learning Method for Normalized Min-Sum LDPC Decoding - Qing Wang

Abstract : With the applications of deep learning networks booming in physical layer communication, deep-learning-based channel decoding methods have become a research hotspot.(phy layer에서 딥러닝 네트워크의 적용에 의해, 채널 디코딩 방법을 기반으로 한 딥러닝은 연구의 관심사가 되었다.)

- In this paper, we propose a model-driven deep learning method for normalized min-sum(NMS) low-density parity-check(LDPC) decoding. First, we propose a neural normalized min-sum(NNMS) LDPC decoding network. By unfolding the iterative decoding progress between checking nodes(CNs) and variable nodes(VNs) into a feed-forward propagation network, we can make use of the advantages of both model-driven deep learning methods and conventional normalized min-sum(CNMS) LDPC decoding method.

Second, considering that the NNMS decoder needs large number of multipliers, we propose a shared neural normalized min-sum(SNNMS) decoding network to reduce the number of correction factors.

1. introduction

The belief propagation(BP) or min-sum(MS) algorithm are two main LDPC decoding approaches.
(BP알고리즘과 MS알고리즘은 LDPC 디코딩 방법이다.)

The BP decoding method achieves good performance but consumes a lot of logarithmic and multiplicative operations in the check nodes(CNs) computation. Comparing with the BP decoding method, the MS decoding method significantly reduces the computation and hardware complexity at a cost of performance degradation.(BP 디코딩은 성능이 좋지만, logarithmic과 곱셈 연상을 체크노드 계산에서 많은 양을 소비하는 tradeoff 관계가 있다. 따라서 BP와 달리 MS 디코딩은 상당히 계산 량을 줄이고, 성능 하락을 희생해서, 하드웨어의 복잡도를 줄인다.)

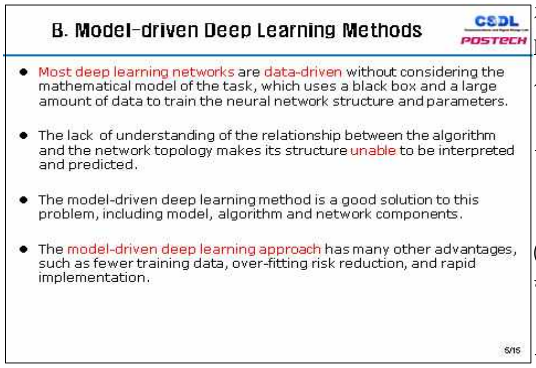
However, the modified MS decoding, a.k.s, normalized MS(NMS) or offset MS(OMS), that employs correction factors can offer competitive decoding performance with a low complexity.(BP 알고리즘 -> MS 알고리즘 -> NMS or OMS.)
As the code length grows, training the neural network will require a hug collection of code words dataset.(code 길이가 길어질수록, NN을 학습하는 것은 많은 양의 code words가 필요합니다.)

the complexity of computation hinders the application of neural networks for long code decoding. -> 계산의 복잡도는 long code 디코딩에 대한 NN의 적용을 방해한다.

저자가 제안한 방법

NNMS : novel neural normalized min-sum LDPC decoding network, where the iterative decoding progress between check nodes and variable nodes are unfolded into feed-forward propagation network to accelerate the decoding progress, which is actually the BP [1] unrolling methods for long code decoding. (NNMS는 체크노드와 변수노드 사이의 iterative decoding progress를 feed-forward propagation network로 unfold한 것으로, 디코딩 처리를 가속화한)

SNNMS : considering that the NNMS improves performance at the cost of a large number of multipliers, we then propose a shared neural normalized min-sum(SNNMS) decoding network by using the same correction facotrs in each layer of the SNNMS networks to reduce the variety of learnalbe correction factors.



저자가 제안한 방법은 'normalized min-sum(NMS)'와 'Model-driven Deep Learning Method'를 기반으로 작성한 것으로, 이때 'Model-driven Deep Learning Method'에 대해서 언급하겠습니다.

- Most deep learning networks are data-driven without considering the mathematical model of the task, which uses a black box and a large amount of data to train the neural network structure and parameters.
(대부분의 딥러닝 네트워크의 경우 data-driven 방법으로, 이 방법은 NN 구조와 파라미터를 학습하기 위해서 수학적 모델을 고려하지 않는다.)

- The lack of understanding(data-driven deep learning methods의 경우, 알고리즘과 네트워크 사이의 관계에 대해 이해가 부족하기에, 깊은 분석이 어렵고, 많은 양의 label data를 필요하지만, 얻기 어렵다.) of the relationship between the algorithm and the network topology makes its structure unable to be interpreted and predicted. Besides, a large amount of label data is not easy to obtain.

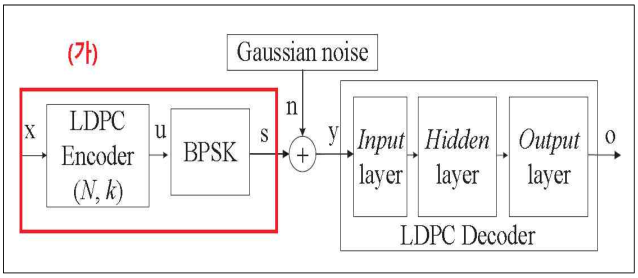
The model-driven deep learning method is a good solution to this problem, including model, algorithm and network components. : model-driven deep learning method의 경우, 이런 문제를 해결하는 좋은 해결책으로 특정 작업에 대한 도메인 지식, 물리적 매커니즘을 기반으로 구성 된 손실함수 같은 모델을 사용하는 방법.

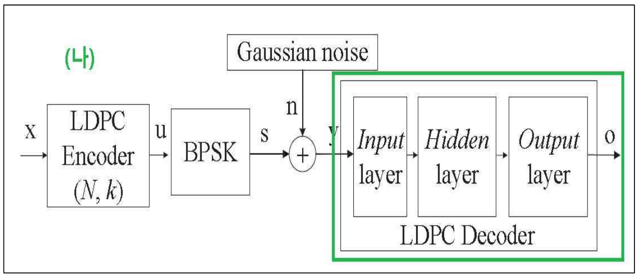
1) unrolling : 알고리즘 언롤링(unrolling) 또는 언폴딩(flending)이라는 새로운 기술은 신호 처리에 널리 사용되는 반복 알고리즘 과 심층 신경 네트워크 사이의 구체적이고 체계적인 연결을 제공함으로써 이러한 문제를 제거할 가능성을 제공한다. (An emerging technique called algorithm unrolling, or unfolding, offers promise in eliminating these issues by providing a concrete and systematic connection between iterative algorithms that are widely used in signal processing and deep neural networks.)

with the prior knowledge of the model and algorithm of a specific task, a interpretable network can be designed accordingly. Moreover, the model-driven deep learning approach has many other advantages, such as a fewer training data, over-fitting risk reduction, and rapid implementation.

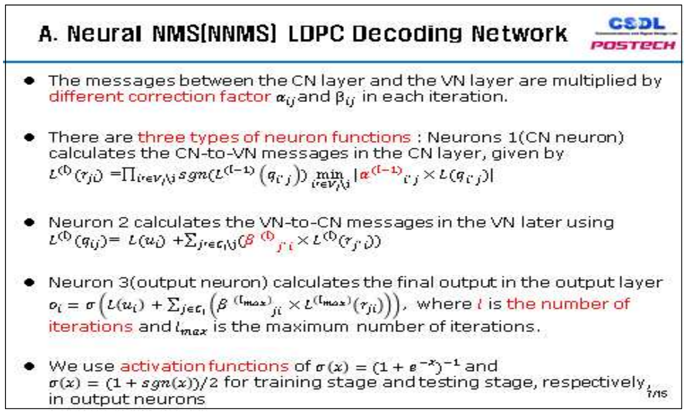
3. The Proposed Neural NMS(NNMS) LDPC Decoding Network

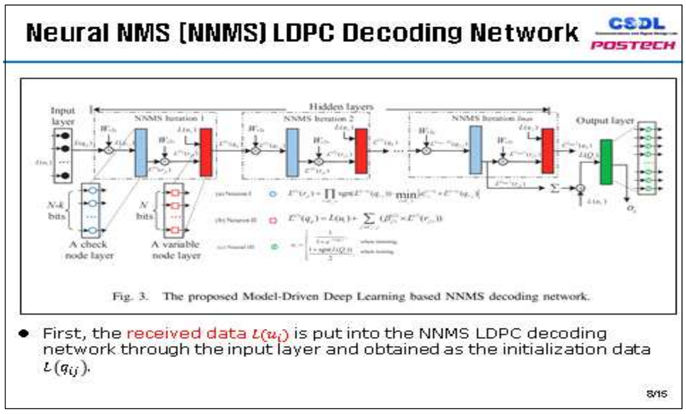
A. Neural NMS(NNMS) LDPC Decoding Network





(가) : 위의 그림은 제안된 NNMS LDPC decoding network의 block diagram이고, (가)부분의 x의 경우 k비트의 information bits인데, 이것은 LDPC Encoder에 의해 N 비트 code word 'u'로 인코딩 된다. 그리고 AWGN 채널을 통해 전송되고, 수신기에서 LLR값은 수신기 신호인 'y=s+n'을 사용하여, 계산되고, deep feed-forward neural network로 입력이 된다. (At the receiver, the LLR values are calculated using the receiver signal "y=s+n" and then are fed into the deep feed-forward neural network.) 그리고 LDPC decoder의 task는 전송된 information bits인 $\hat{u}= o$의 값을 추정하는 것. (The task of the LDPC decoder is to estimate the transmitted information bits $\hat{u}= o$.)



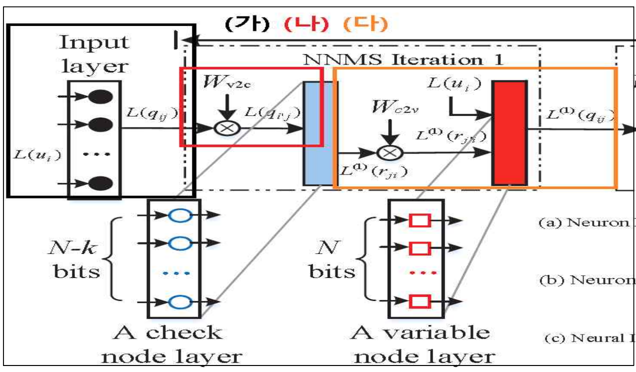


The iterative decoding algorithm is unfold into a forward-propagation network. In the Tanner graph, the parity check matrix H determines the edges of CNs and VNs connections.(parity check matrix H는 체크노드와 변수노드의 edge를 결정한다.)

The messages between the CN layer and the VN layer are multiplied by different correction factors α_{ij} and β_{ji} in each iteration, which is equivalent to adding weight parameters to the edges of Tanner graph.(체크노드 레이어와 변수노드 레이어 사이의 메시지는 서로 다른 correction facotr들이 매 반복마다 곱해지는 데, 이것은 테너 그래프의 edges에 weight parameters가 더해지는 것과 같다.)

NNMS와 SNNMS의 경우, 구조는 Neuron 1, 2, 3로 구성되어 있는데, correction factor를 어떻게 사용하는 지에 따라 구분이 된다.

B. The Signal Flow of the Proposed NNMS Network



(가) : the received data $L(u_i)$ is put into the NNMS LDPC decoding network through the input layer and obtained as the initialization data $L(q_{ij})$.(수신된 data는 input layer를 통해 NNMS LDPC decoding network로 들어가고, 초기화된 데이터 $L(q_{ij})$를 얻는다.)

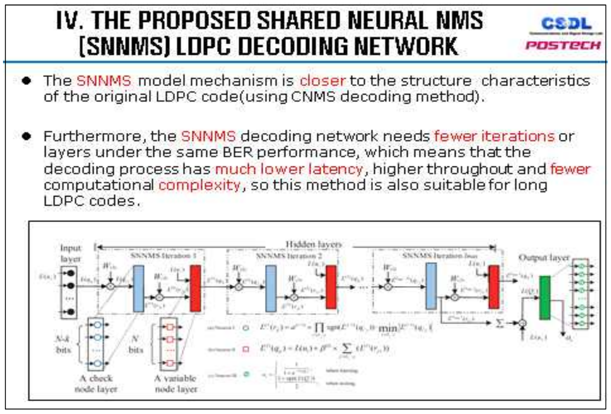
(나) l번째 hidden layer에서 'CN-to-VN messages'($L^{(l)}(r_{ji})$)를 계산하기 위해서, the data set $L^{(l-1)}(q_{i'j})$은 W_{v2c}에 의해서 추출된 $L^{(l-1)}(q_{i'j})$값이 얻어진다.

at the lth hidden layer, to calculate the CN-to-VN meesages $L^{(l)}(r_{ji})$, the data set $L^{(l-1)}(q_{i'j})$, which is connected to the CN_j except the VN_i, is extracted from $L^{(l-1)}(q_{ij})$ by W_{v2c}, where W_{v2c} is the row weight information of the parity check matrix H. $L^{(l)}(r_{ji})$ is calculated through the neuro 1 of the CN layer.

(다) l번째 hidden layer에서 'CN-to-VN messages'($L^{(l)}(r_{ji})$)를 계산하기 위해서, the data set $L^{(l-1)}(q_{i'j})$은 W_{v2c}에 의해서 추출된 $L^{(l-1)}(q_{i'j})$값이 얻어진다.

to calculate the VN-to-CN messages $L^{(l)}(q_{ij})$, the data set $L^{(l)}(r_{i'j})$, which is connected to the VN_i except the CN_j, is extracted from $L^{(l)}(r_{ji})$ by W_{c2v}, where W_{c2v} comes from the column weight information of the parity check matrix H. $L^{(l)}(q_{ij})$ is calculated in the neuron 2(VN neuron) of the VN layer.

- For (N, K) LDPC codes, the T iterations is unfolded into a deep neural network with $2l_{\max}+2$ layer, consisting of $l_{\max}$ CN layers, $l_{\max}$ VN layers, one input layer and one output layer.(CN, VN 각각 $l_{\max}$가 있고, input, output이 있기에 $2l_{\max}+2$가 된다.)

- 

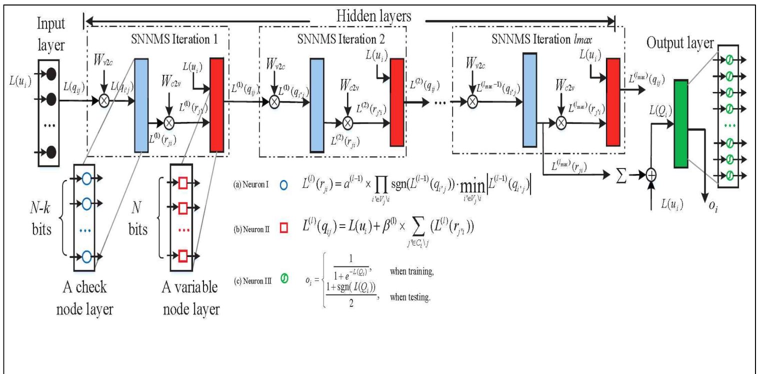
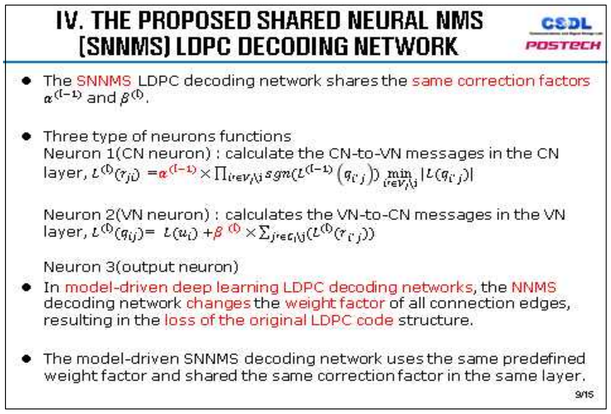


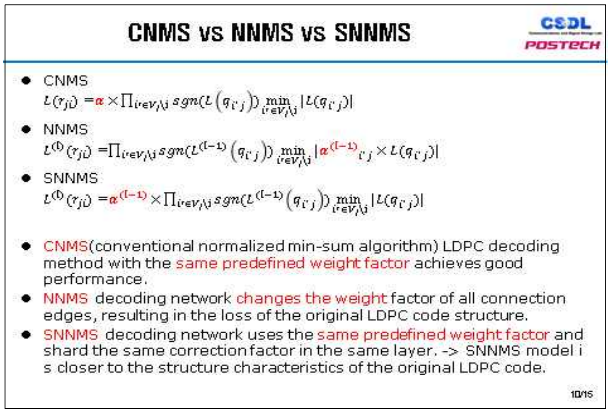
Fig. 4. The proposed Model-Driven Deep Learning based SNNMS decoding network.

-

4. The proposed Shared Neural NMS(SNNMS) LDPC Decoding Network

- The proposed NNMS LDPC decoding network use different weights for each connection edge between the CNs and the VNs in each iteration.(NNMS는 연결된 edge에 따라 다른 correction factor를 사용한다.) Although the parity check matrix H of LDPC code is a sparse matrix, this process will consume a lot of multipliers for long codes.(각각 다른 correction factor를 사용하다보니, 많은 multiplier를 사용하기에, long code에는 적합하지 않다.)
- In this section, we propose a network to reduce the number of correction factors. By sharing the same correction factors in the same layer, the SNNMS LDPC decoding network can achieve good performance with little increment of computation complexity.





A. Shared Neural NMS(SNNMS) LDPC Decoding Network : The SNNMS LDPC decoding network shares the same correction factors $\alpha^{(l-1)}$ and $\beta^{(l)}$, as shown in Fig. 4.

Comparing with NNMS, we can see that the two decoding networks seem to have similar network structure, but neurons are completely different actually, as shown in Fig.4 and Fig.3. (NNMS와 SNNMS는 비슷한 네트워크 구조를 갖지만, 서로 다른 correction factor 사용법에 따라 다른 결과 값을 갖는다.) 식의 $\alpha^{(l-1)}$, $\beta^{(l)}$가 각각 layer마다 같은 correction factor를 사용하는 것을 알 수 있습니다.

오른쪽 그림을 통해, CNMS, NNMS, SNNMS의 성능을 비교하고 있습니다.
- As the performance and complexity of LDPC codes, with different structures, is different. The CNMS LDPC decoding method with the same predefined weight factor(correction factor α) achieves good performance.

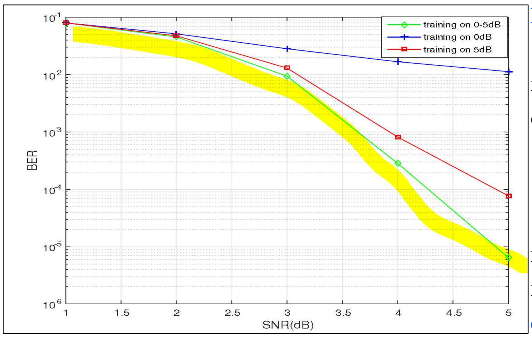
- However, in model-driven deep learning LDPC decoding networks, the NNMS decoding network changes the weight factor of all connection edges(the parity check matrix H has changed), resulting in the loss of the original LDPC code structure.(NNMS는 연결된 edge에 따라 weight factor를 바꾸기 때문에, 원래 LDPC 코드 구조에 대한 loss를 초래한다.)

- the model-driven SNNMS decoding network uses the same predefined weight factor and shared the same correction factor in the same layer. Thus the SNNMS model mechanism is closer to the structure characteristics of the original LDPC code(using CNMS decoding method), and it is more helpful for the network to capture the topological relationship between the nodes of the entire LDPC code when propagating forward.

- 5. Simulations Results

- The LDPC code is designed according to IEEE 802.16e standard with code block length 576 and rate 3/4, where the codeword is selected randomly and the parity check matrix H is from “Database of channel codes and ml simulation results : www.uni-kl.de/channel-codes = University of Kaiserslautern”.

The training data is generated under multiple signal-to-noise(SNRs) ranging from 0dB to 5dB. We use the min-batch gradient descent method to train the network. Each mini-batch contains 120 blocks of data and each SNR occupies the same proportion in one mini-batch.



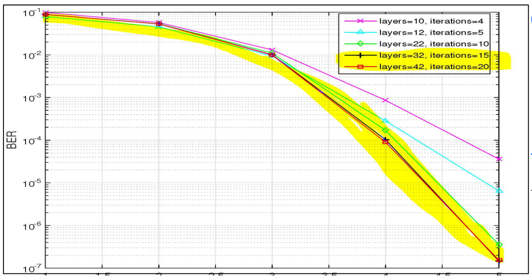
B) System Robustness

- The first experiment tests the system robustness. The system robustness can be measured using BER performance with respect to the quality of the training data(the signal to noise ratio(SNR) of the training data.)

- 0dB일 때, bad channel condition에서 training data는 생성되어, dataset에 error가 많이 존재하는 결과를 초래한다. (When the SNR of training data equals to 0db, which means that the training data is generated under a bad channel condition, resulting in a lot of errors in the labeled dataset, thus the network can not learn the decoder structure to correct errors.)

- when the SNR is high, there will be fewer erroneous data are fed into the network. The network system can not learn the ability to remove noise.
- comparing with the network trained using 0dB or 5dB data only, we can see that the more diverse the training data(training under multi-channel conditions of 0~5dB), the better BER performance the network achieves.

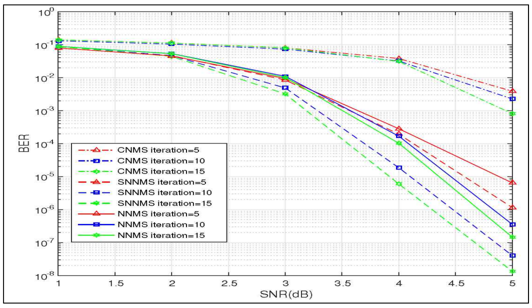
=> The diversity of the training data helps enhance the robustness of the data-driven neural network.



C. BER Performance vs. Number of Network Layers

- The second experiment analyzes the BER performance in terms of the number of network layers.
- within a certain number of network layers, with the number of layers increase, the BER performance gets improved(layer의 수가 많을수록, 깊이가 더 깊어지기에, 덧셈, 곱셈 연산을 더 많이 사용하고, BER의 성능이 향상된다.).
- It must point out that, the deeper network layers consume more multiplication and summation operations, thus the improvement of BER performance is at the cost of computation complexity.(그러나 tradeoff 관계로 성능 향상과 함께 복잡도가 늘어나고, 적정 수준의 깊이가 깊어지면, 성능향상은 극적으로 진행되지 않는다.)

When the number of network layers reaches a certain depth, the performance of the decoder will not be greatly improved, which also conforms to the nature of conventional LDPC decoding.(tradeoff between performance and complexity)



D. BER performance Comparsion between NNMS and SNNMS : in the case of 32 layers neural network with BER of 10^{-7}, the BER performance of SNNMS outperforms up to 0.4dB compared with the NNMS LDPC decoding network. SNNMS decoding network is closer to the original LDPC code structure in model mechanism, so it has better performance in model-driven deep learning method.

E. complexity : When compared with CNMS, it has better BER performance without changing much computational complexity.

6. Conclusion : By sharing the correction factors, the SNNMS LDPC decoding network is proposed to reduce the complexity. Simulation results shows the proposed SNNMS decoder can achieve better BER performance with lower computation complexity compared with the NNMS decoder.

In addition, the improved SNNMS decoder is acceptable for long codes application thank to the lower latency and less computational complexity, which is durable in practice.