Neural Network Belief Propagation

GiGi Chou

Advisor: Prof. Jiun Hung Yu
Prof. Tofar Chih-Yuan Chang

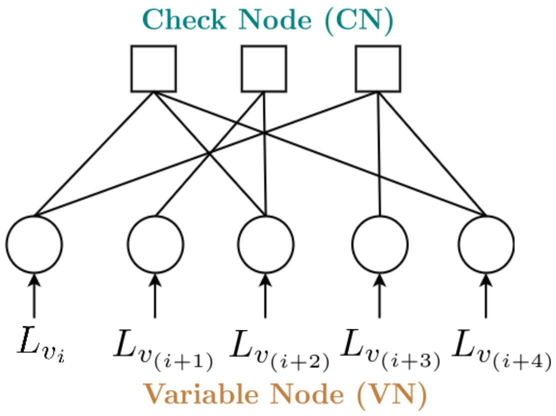
Transmission and Networking Technologies Laboratory,
Institute of Communications Engineering
National Chiao Tung University, Hsinchu, Taiwan

Outline

- Review
- Introduction
- NNBP decoding of BCH
- NNBP decoding of LDPC
- NNBP Decoding of Hard Decision
- NNBP decoding of LDPC without SNR
- Future Research Directions
- Reference

Review - Belief Propagation Algorithm

$$
\mathbf{H} = \begin{array}{c c}
\begin{matrix} \text{VN0} & \text{VN1} & \text{VN2} & \text{VN3} & \text{VN4} \end{matrix} & \\
\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} & \begin{matrix} \text{CN0} \\ \text{CN1} \\ \text{CN2} \end{matrix}
\end{array}
\xrightarrow{\text{Tanner Graph}}
$$



1. **Initialization:** For all VN, initialize all L_{v_i} according to (1) for the appropriate channel model. Then, set $L_{v_i \to c_j} = L_{v_i}(LLR)$ at first iteration.

$$L_{v_i} = L(r_i|y_i) = \log\left(\frac{\Pr(r_i = 0|y_i)}{\Pr(r_i = 1|y_i)}\right) = \frac{2y_i}{\sigma^2} \quad (1)$$

r_i represents the decoded codeword , y_i represents the channel value and σ is the Gaussian noise standard deviation.

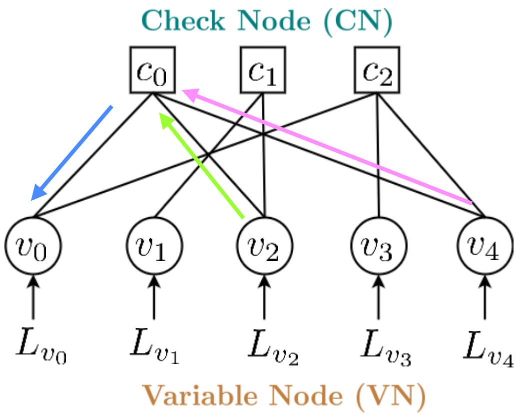
Review - Belief Propagation Algorithm

2. **CN update:** Compute outgoing CN messages $L_{c_j \to v_i}$ for each CN to be transmitted to the VNs.

$$L_{c_j \to v_i} = 2\tanh^{-1}\left(\prod_{v' \in N(c_j) \setminus v_i} \tanh\left(\tfrac{1}{2} L_{v' \to c_j}\right)\right) \quad (2)$$

Ex :

$$L_{c_0 \to v_0} = 2\tanh^{-1}\left(\tanh\left(\frac{1}{2} L_{v_2 \to c_0}\right) \times \tanh\left(\frac{1}{2} L_{v_4 \to c_0}\right)\right)$$



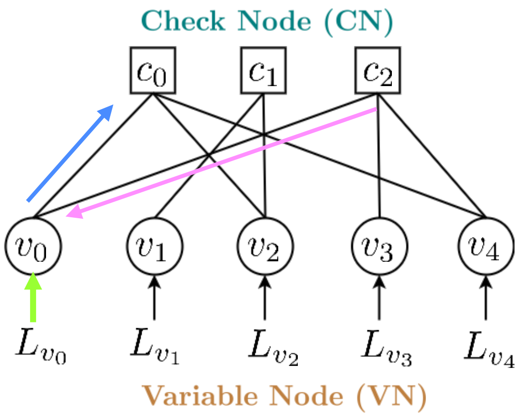
Review - Belief Propagation Algorithm

3. **VN update:** Compute outgoing VN messages $L_{v_i \to c_j}$ for each VN to be transmitted to the CNs.

$$L_{v_i \to c_j} = L_{v_i} + \sum_{c' \in N(v_i) \backslash c_j} L_{c' \to v_i} \tag{3}$$

Ex :

$$L_{v_0 \to c_0} = L_{v_0} + L_{c_2 \to v_0}$$



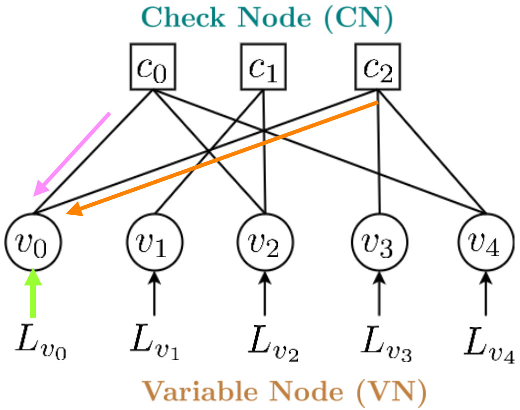
Review - Belief Propagation Algorithm

4. **LLR total:** For $i = 0, 1, \ldots, n-1$, compute

$$L_{v_i}^{\text{total}} = L_{v_i} + \sum_{c' \in N(v_i)} L_{c' \to v_i} \tag{4}$$

Ex :

$$L_{v_0}^{\text{total}} = L_{v_0} + L_{c_0 \to v_0} + L_{c_2 \to v_0}$$



Review - Belief Propagation Algorithm

5. **Stopping criteria:** For $i = 0, 1, \ldots, n-1$, set

$$\hat{r}_i = \begin{cases} 1 & \text{if } L_{v_i}^{\text{total}} < 0, \\ 0 & \text{else,} \end{cases}$$

to obtain $\hat{\boldsymbol{r}}$. If $\hat{\boldsymbol{r}}\boldsymbol{H}^T = \vec{\mathbf{0}}$ or the number of iterations equals the maximum limit, stop; else, go to Step **CN update**.

Outline

- Review
- **Introduction**
- NNBP decoding of BCH
- NNBP decoding of LDPC
- NNBP Decoding of Hard Decision
- NNBP decoding of LDPC without SNR
- Future Research Directions
- Reference

Introduction

- Bose, Chaudhuri, and Hocquenghem (BCH) codes, while well-established for error correction, lack efficient and practical soft-decision decoding algorithms. Traditional soft decoding methods such as maximum likelihood decoding are not only computationally expensive but also result in sub-optimal performance in real-world applications.
- Due to the absence of a robust soft decoding technique, neural network belief propagation (NNBP) has emerged as a promising alternative.
- Learns to compensate for short cycles in the Tanner graph by properly weighting messages.

Introduction - BP transform to NNBP

CN update:

$$L_{l,c_j \to v_i} = 2\tanh^{-1}\left(\prod_{v' \in N(c_j) \backslash v_i} L_{l-1,v' \to c_j}\right)$$

VN update:

$$L_{l,v_i \to c_j} = \tanh\left(\frac{1}{2}\left(w_{l,v_i} L_{v_i} + \sum_{c' \in N(v_i) \backslash c_j} w_{l,c' \to v_i} L_{l,c' \to v_i}\right)\right)$$

Total LLR :

$$o_{v_i} = \sigma\left(w_{l,v_i} L_{v_i} + \sum_{c' \in N(v_i)} w_{l,c' \to v_i} L_{l,c' \to v_i}\right)$$

$w_{l,c' \to v_i}$: Represents the weight of the CN (c') to VN (v_i) at the l^{th} iteration.

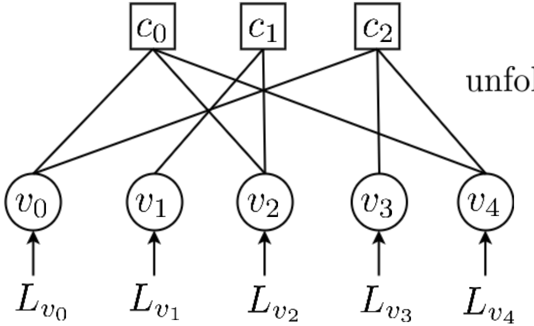
w_{l,v_i}: Represents the weight of the i^{th} channel value at the l^{th} iteration.

σ : Represents a sigmoid function with an output range of $[0,\ 1]$.

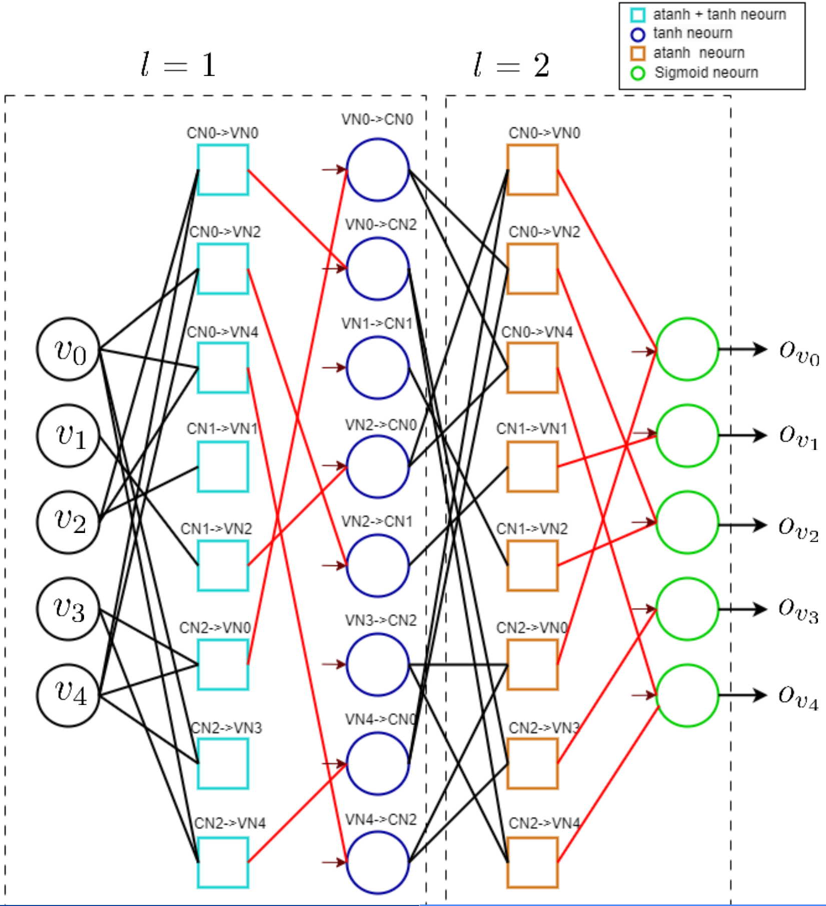
Introduction - BP transform to NNBP

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

Tanner Graph



unfold , iteration=2



Outline

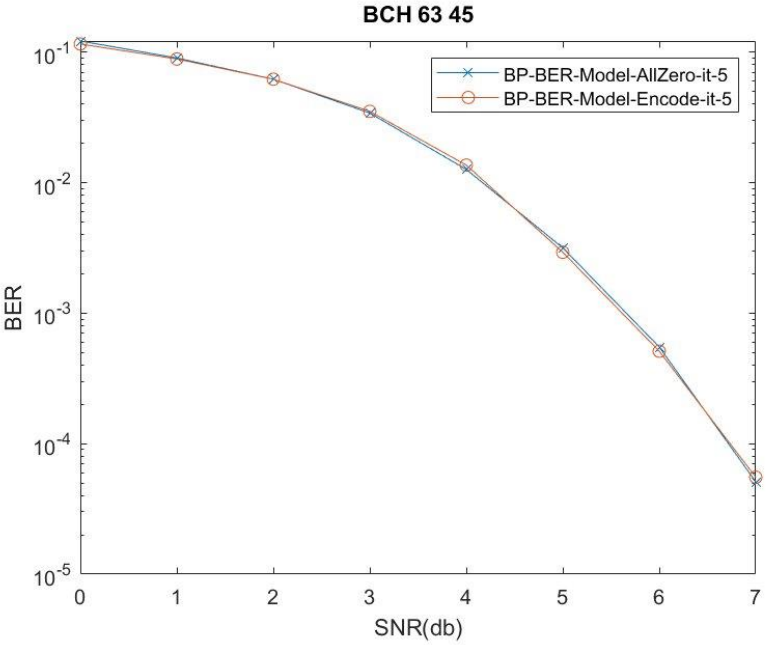
- Review
- Introduction
- **NNBP decoding of BCH**
- NNBP decoding of LDPC
- NNBP Decoding of Hard Decision
- NNBP decoding of LDPC without SNR
- Future Research Directions
- Reference

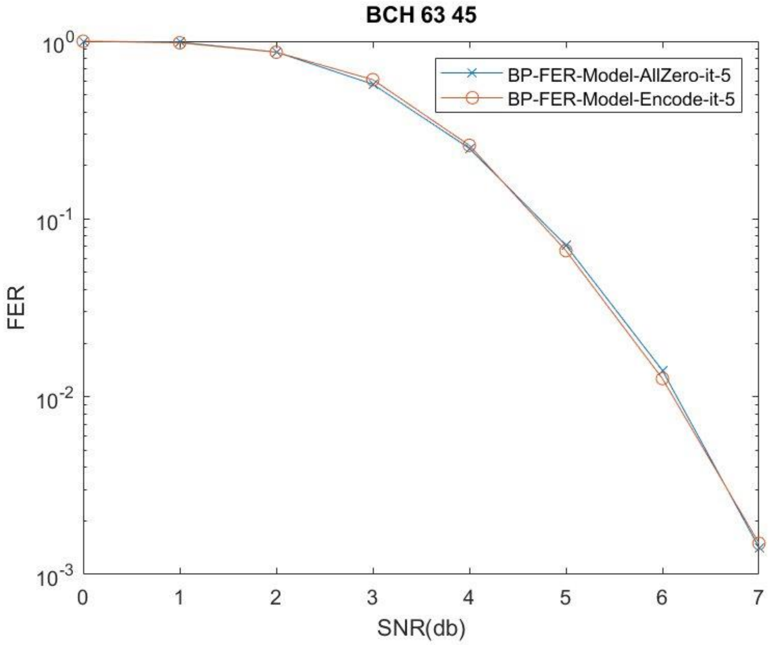
NNBP Decoding of BCH (63,45)

Model Information :

Model	BP with 5 iteration = 10 layer Neural Network
Parity check matrix	BCH (63,45)
Training data set	SNR:0∼5dB (step=0.25, Zero CodeWord)
Training data format	Log-Likelihood Ratio (LLR)
Loss Function	Mutiloss (BCE Loss)
Optimizer	Adam
Learning ratio	0.001

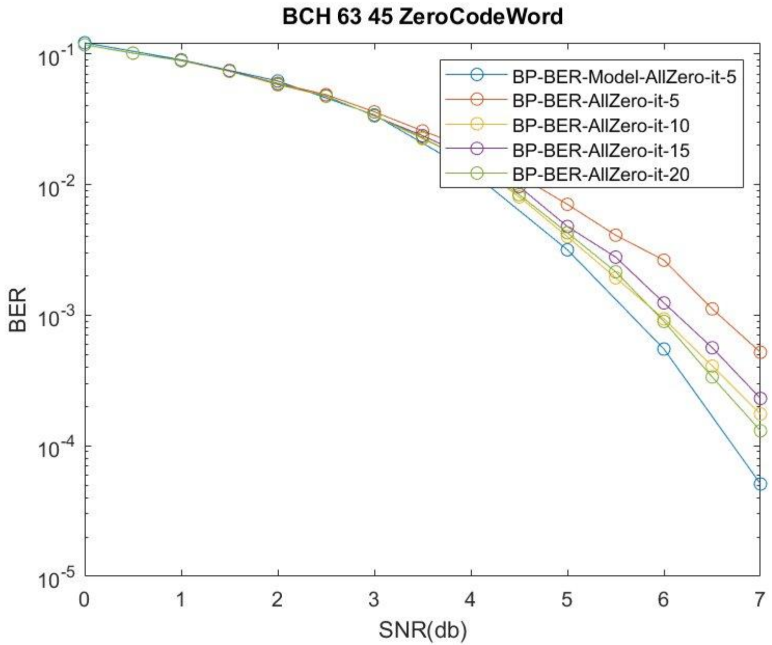
To ensure that the training results do not converge to zero.

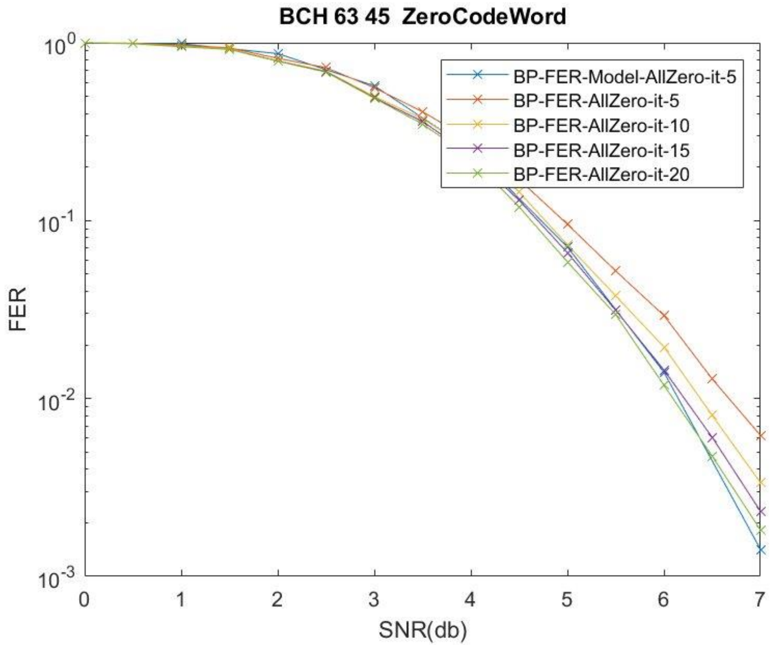




Simulation

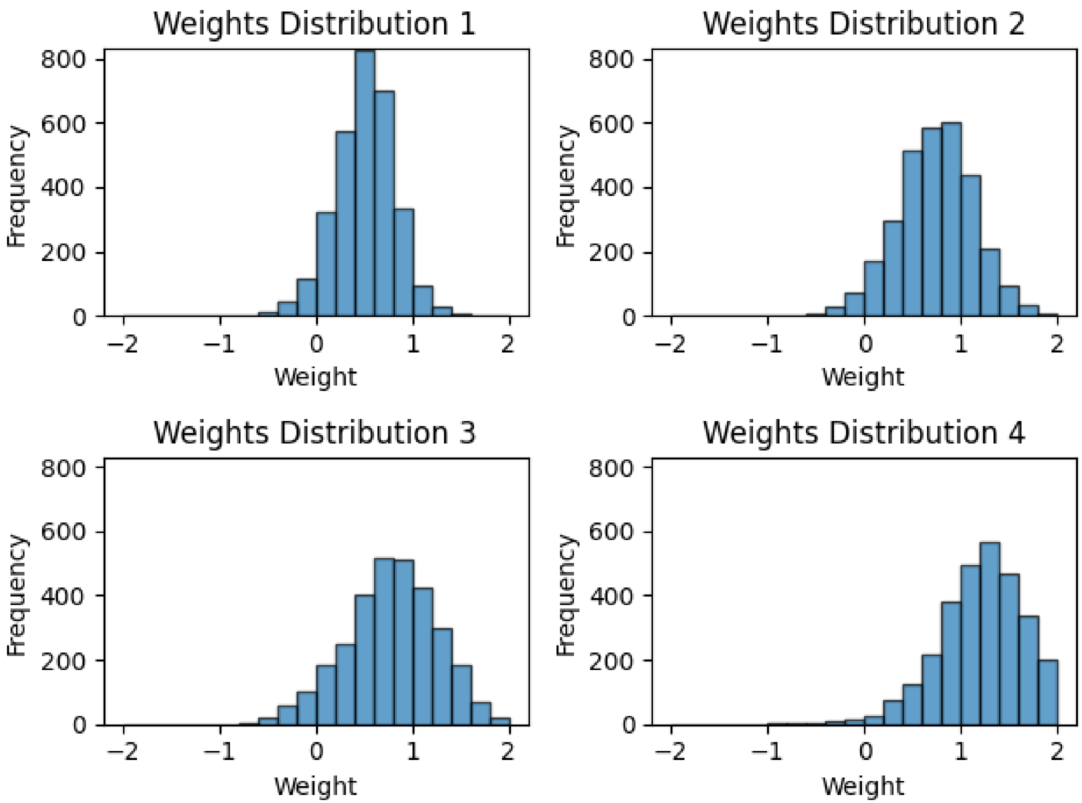
To compare with the conventional BP (CBP) algorithm using different numbers of iterations.





Simulation

weights histogram for the each iteration.



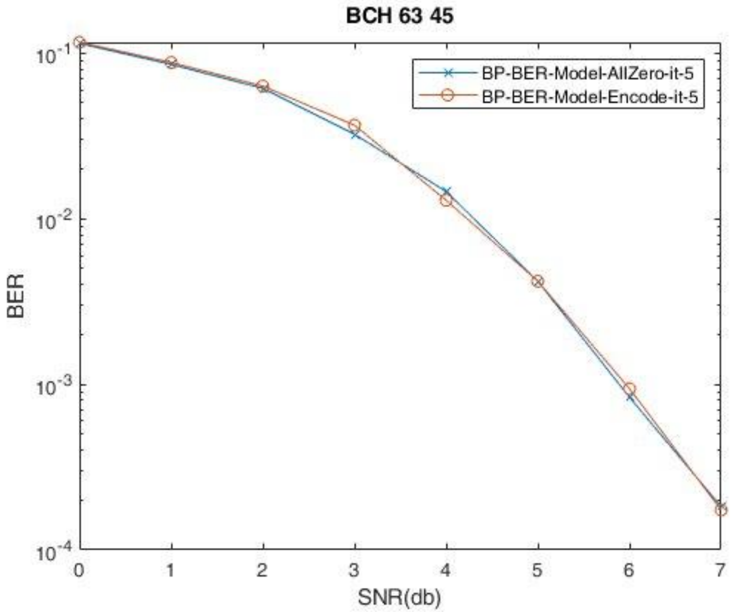
NNBP Decoding of BCH (63,45)

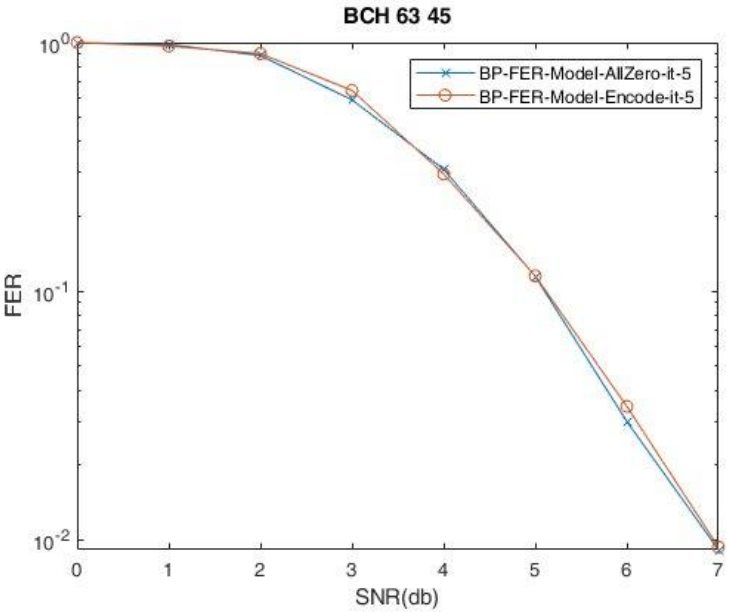
Model Information :

Model	BP with 5 iteration = 10 layer Neural Network
Parity check matrix	BCH (63,45)
Training data set	SNR:0~5dB (step=0.25, Encode CodeWord)
Training data format	Log-Likelihood Ratio (LLR)
Loss Function	Mutiloss (BCE Loss)
Optimizer	Adam
Learning ratio	0.001

Simulation

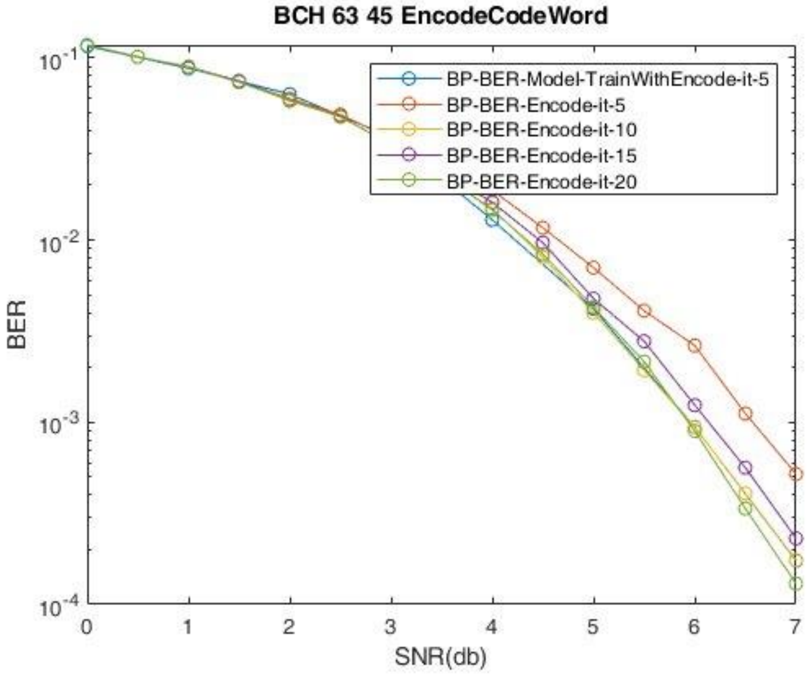
To ensure that the training results do not converge to zero.

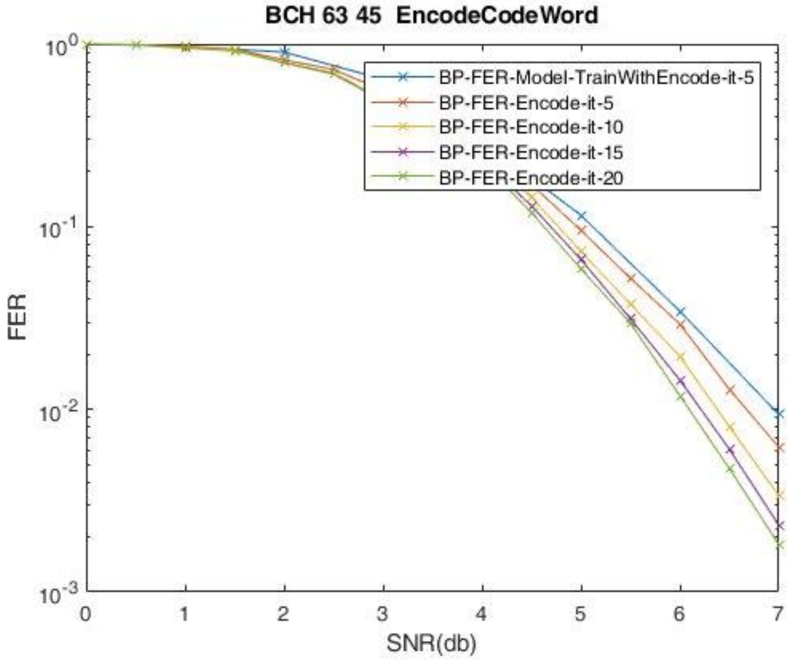




Simulation

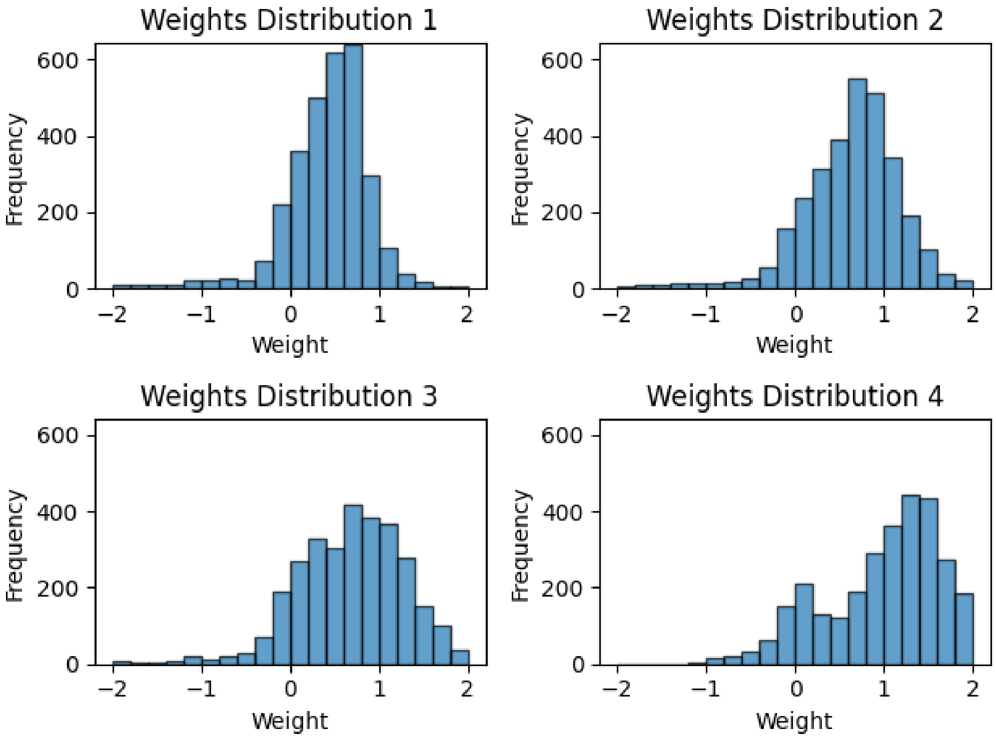
To compare with the conventional BP (CBP) algorithm using different numbers of iterations.





Simulation

weights histogram for the each iteration.



Conclusion(1)

- Training with all-zero codewords and encoded codewords, the result from training with all-zero codewords is more better.

- When using all-zero codewords to train a model, the absence of variability in the input could make it easier for the model to identify noise and potential cycles or interference patterns.

- Using NNBP for soft decoding of BCH codes reduces the number of iterations and decreases complexity. It efficiently handles soft information, approaching correct decoding faster, and automates parameter learning to simplify the process.

- Since NNBP has demonstrated advantages in HDPC, the next step is to explore its performance when applied to LDPC. It will be interesting to see how NNBP handles the structure and decoding challenges of LDPC codes.

Outline

- Review
- Introduction
- NNBP decoding of BCH
- **NNBP decoding of LDPC**
- NNBP decoding of Hard Decision
- NNBP decoding of LDPC without SNR
- Future Research Directions
- Reference

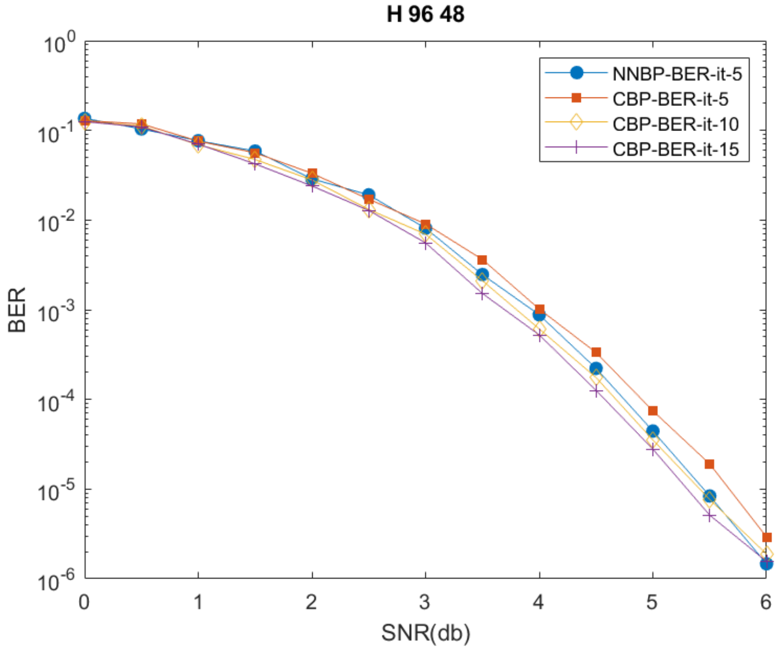
NNBP Decoding of LDPC (96,48)

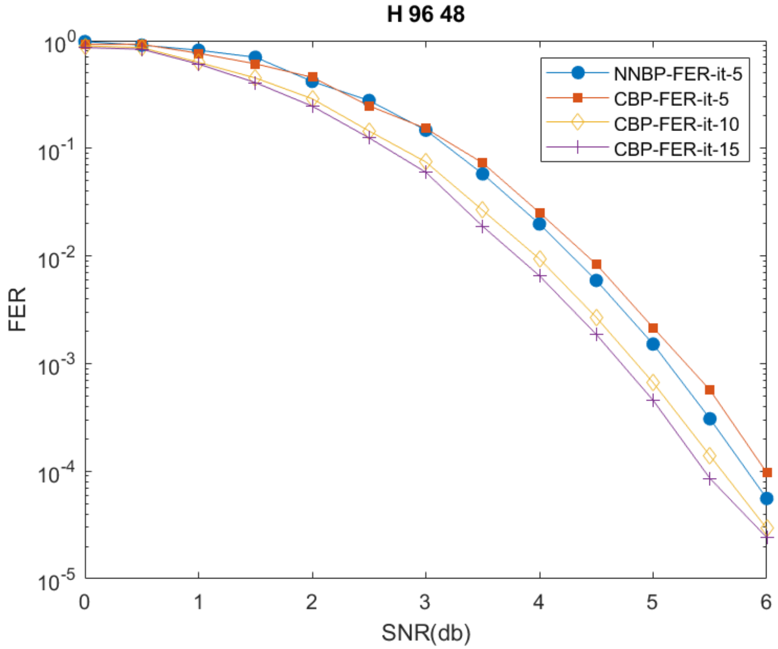
Model Information :

Model	BP with 5 iteration = 10 layer Neural Network
Parity check matrix	LDPC (96,48), girth=6
Training data set	SNR:3∼4.5dB (step=0.1, Zero CodeWord)
Training data format	Log-Likelihood Ratio (LLR)
Loss Function	Mutiloss (BCE Loss)
Optimizer	RMSprop
Learning ratio	0.001

Simulation

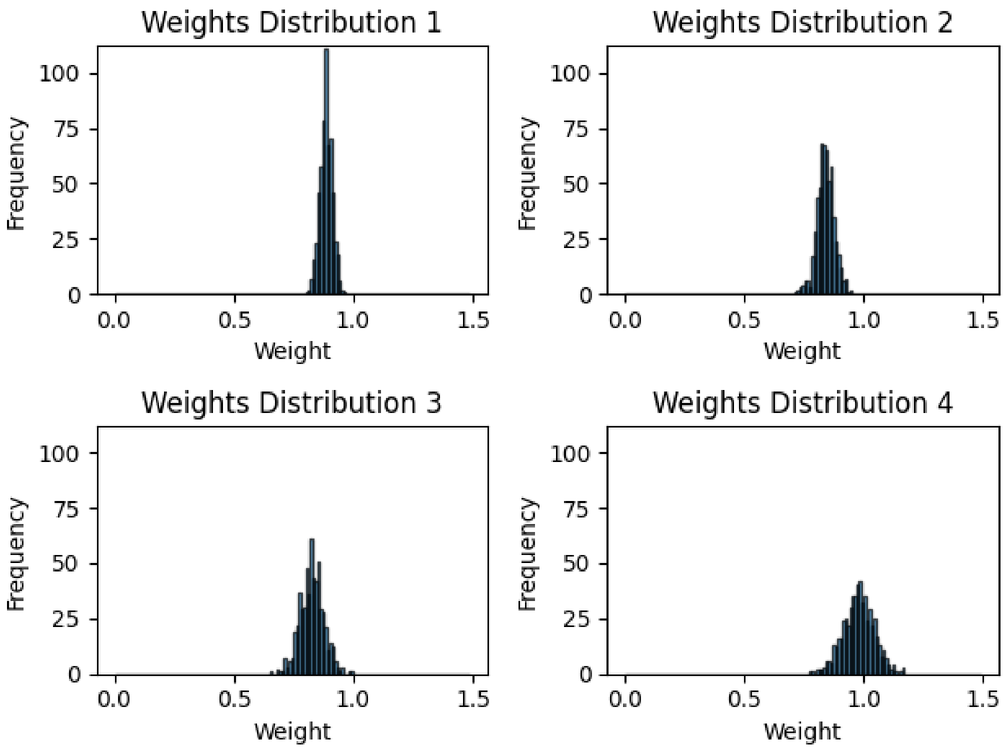
To compare with the conventional BP (CBP) algorithm using different numbers of iterations.





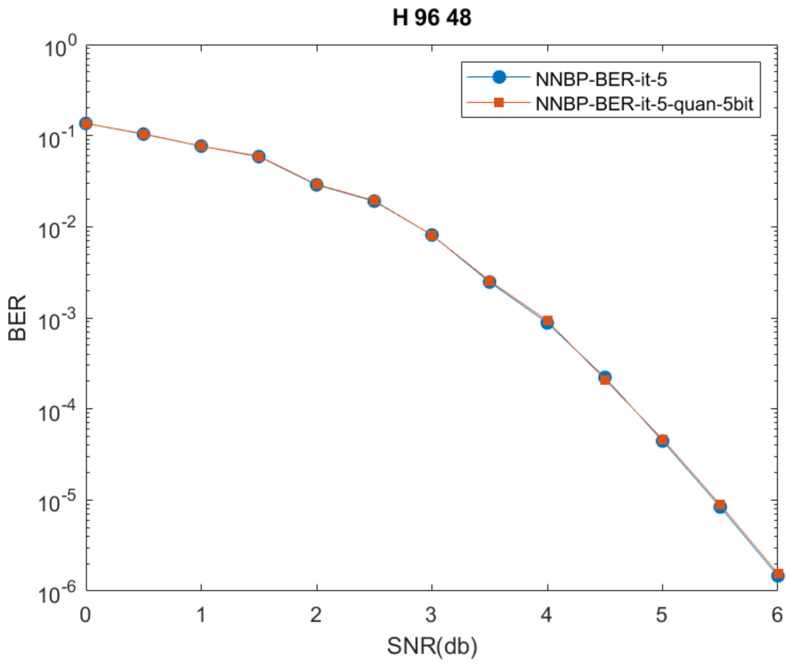
Simulation

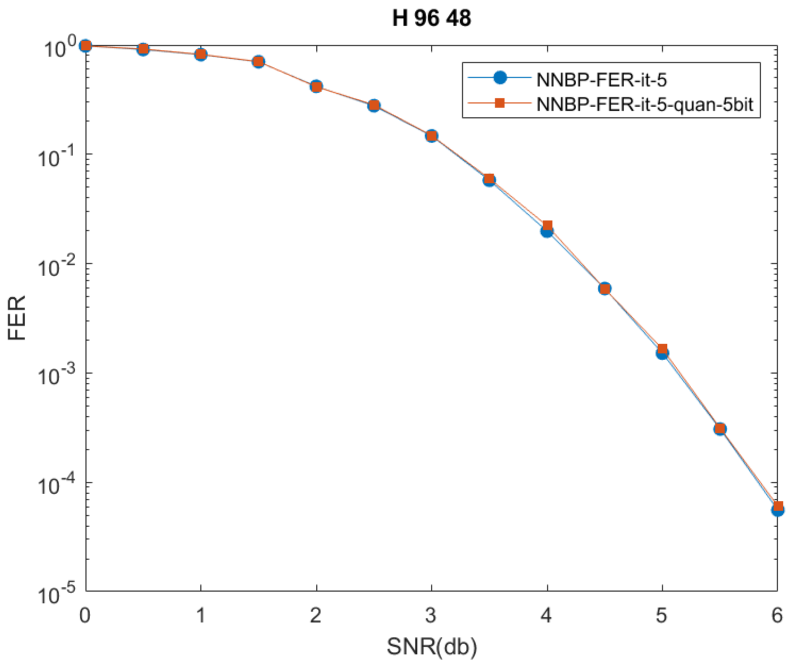
weights histogram for the each iteration.



Simulation

Comparison of the original model's performance and the performance after quantization.





Conclusion(2)

- NNBP shows only slight improvements over conventional BP at the same number of iterations, suggesting that conventional BP is already sufficiently effective.
- Compared to conventional BP, NNBP has higher complexity, making hardware implementation more complicated.

Outline

- Review
- Introduction
- NNBP decoding of BCH
- NNBP decoding of LDPC
- **NNBP decoding of Hard Decision**
- NNBP decoding of LDPC without SNR
- Future Research Directions
- Reference

NNBP Decoding of Hard Decision

Recall :

1. **Initialization:** For all VN, initialize all L_{v_i} according to (1) for the appropriate channel model. Then, set $L_{v_i \to c_j} = L_{v_i}(LLR)$ at first iteration.

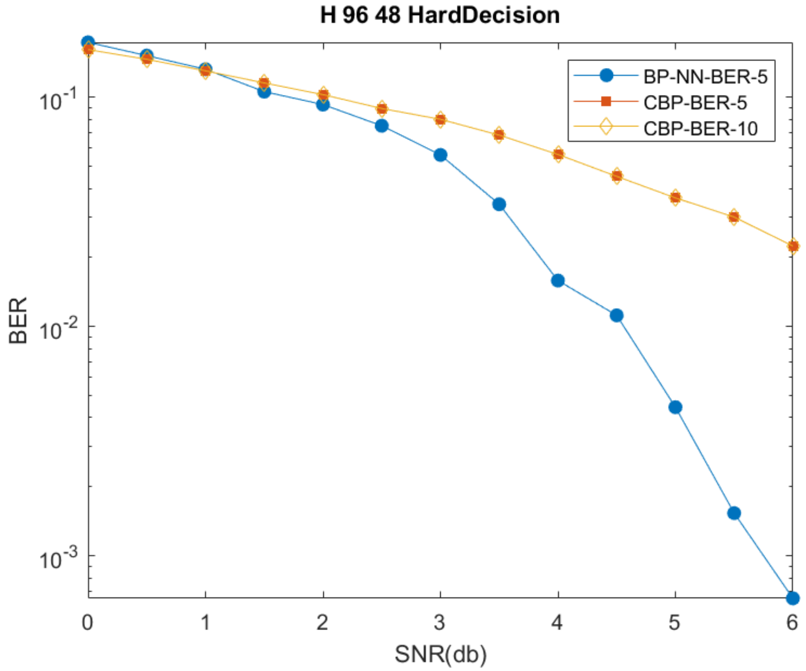
$$L_{v_i} = L(r_i|y_i) = \log\left(\frac{\Pr(r_i = 0|y_i)}{\Pr(r_i = 1|y_i)}\right) = \frac{2y_i}{\sigma^2} \tag{1}$$

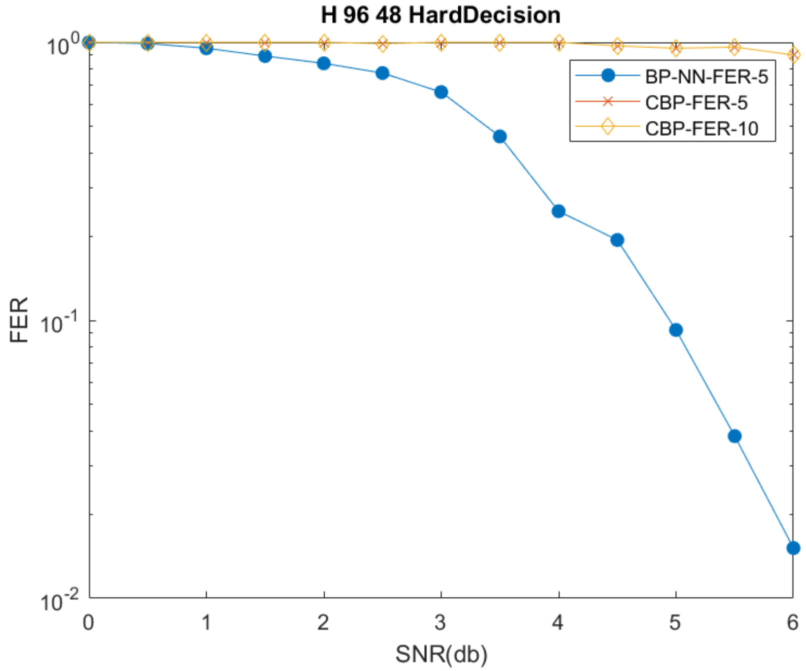
$$L_{v_i} = \begin{cases} 1 & \text{if } y_i > 0, \\ -1 & \text{else,} \end{cases}$$

r_i represents the decoded codeword , y_i represents the channel value and σ is the Gaussian noise standard deviation.

Simulation

To compare with the conventional BP (CBP) algorithm using different numbers of iterations.





NNBP Decoding of Hard Decision

Recall :

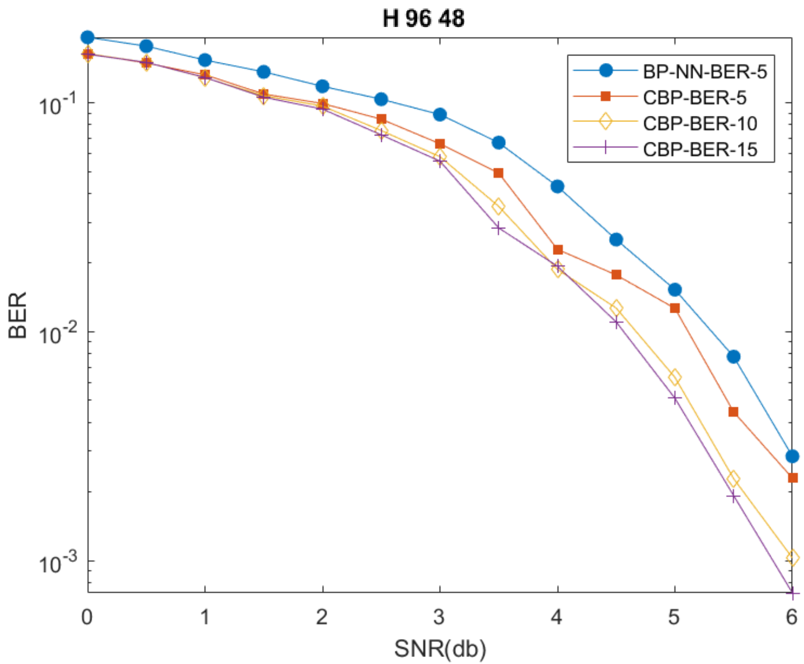
1. **Initialization:** For all VN, initialize all L_{v_i} according to (1) for the appropriate channel model. Then, set $L_{v_i \to c_j} = L_{v_i}(LLR)$ at first iteration.

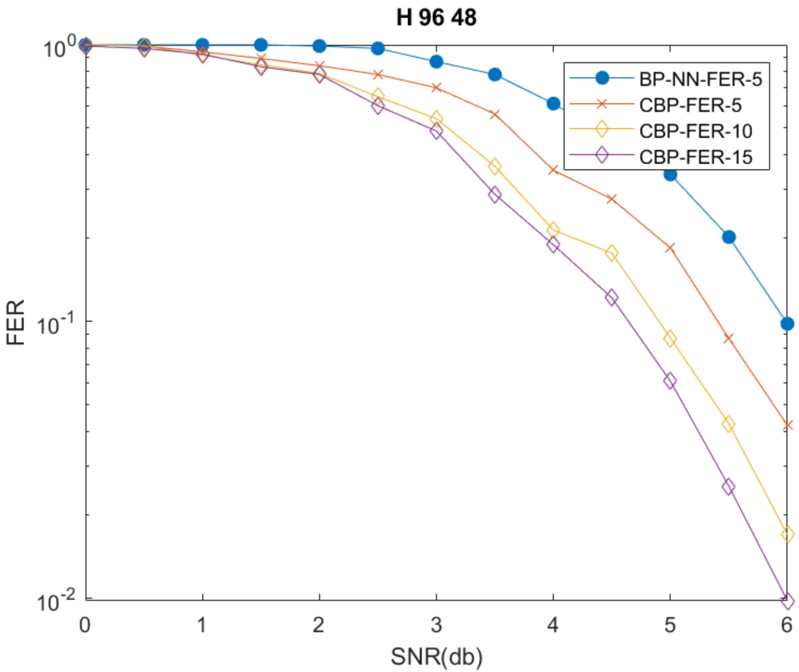
$$L_{v_i} = L(r_i|y_i) = \log\left(\frac{\Pr(r_i = 0|y_i)}{\Pr(r_i = 1|y_i)}\right) = \frac{2y_i}{\sigma^2} \tag{1}$$

$$L_{v_i} = \begin{cases} \frac{2}{\sigma^2} & \text{if } y_i > 0, \\ \frac{-2}{\sigma^2} & \text{else,} \end{cases}$$

r_i represents the decoded codeword , y_i represents the channel value and σ is the Gaussian noise standard deviation.

To compare with the conventional BP (CBP) algorithm using different numbers of iterations.





Outline

- Review
- Introduction
- NNBP decoding of BCH
- NNBP decoding of LDPC
- NNBP decoding of Hard Decision
- **NNBP decoding of LDPC without SNR**
- Future Research Directions
- Reference

NNBP Decoding without SNR

Recall :

1. **Initialization:** For all VN, initialize all L_{v_i} according to (1) for the appropriate channel model. Then, set $L_{v_i \to c_j} = L_{v_i}(LLR)$ at first iteration.

$$L_{v_i} = L(r_i|y_i) = \log\left(\frac{\Pr(r_i = 0|y_i)}{\Pr(r_i = 1|y_i)}\right) = \frac{2y_i}{\boxed{\sigma^2}} \quad (1)$$

r_i represents the decoded codeword , y_i represents the channel value and σ is the Gaussian noise standard deviation.

When performing soft decoding, the receiver needs Signal to Noise Ratio (SNR) for effective decoding. Without SNR, the decoding performance will degrade significantly.

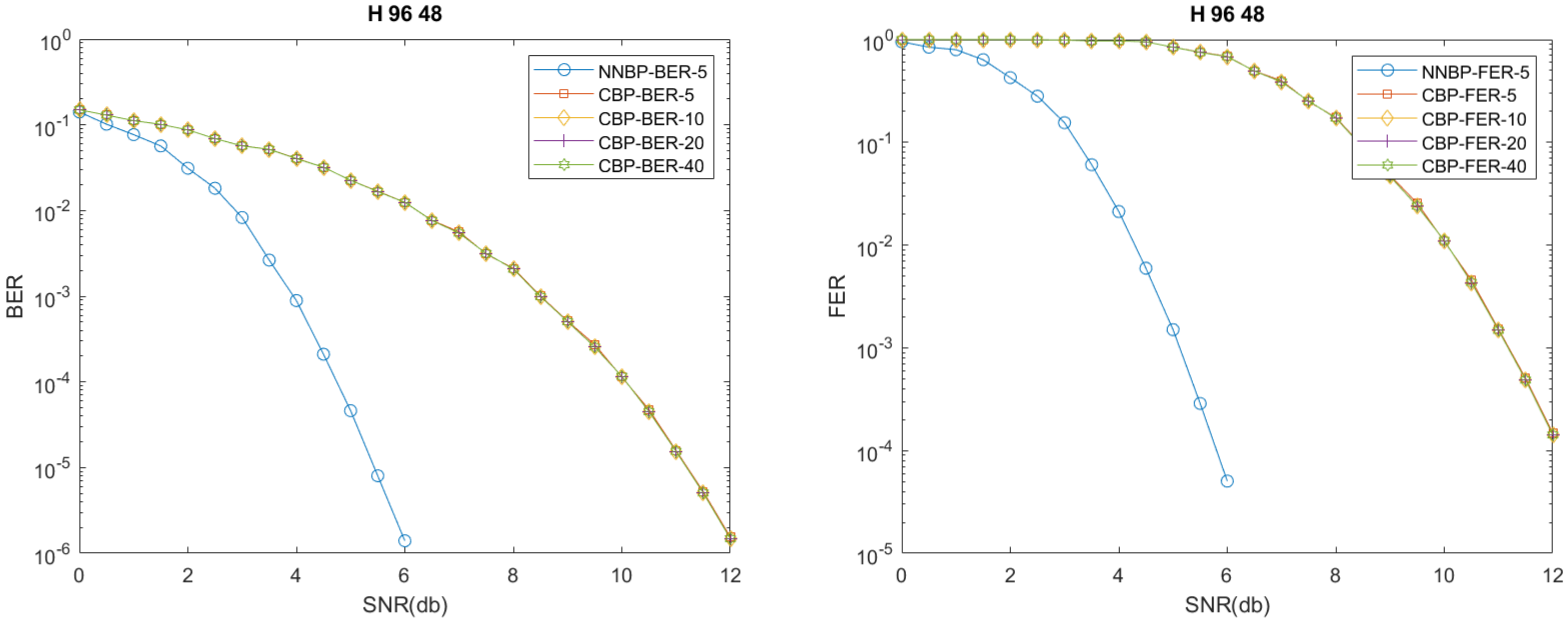
NNBP Decoding without CSI

Model Information :

Model	BP with 5 iteration = 10 layer Neural Network
Parity check matrix	LDPC (96,48), girth=6
Training data set	SNR:3∼4.5dB (step=0.1, Zero CodeWord)
Training data format	received signal value
Loss Function	Mutiloss (BCE Loss)
Optimizer	RMSprop
Learning ratio	0.001

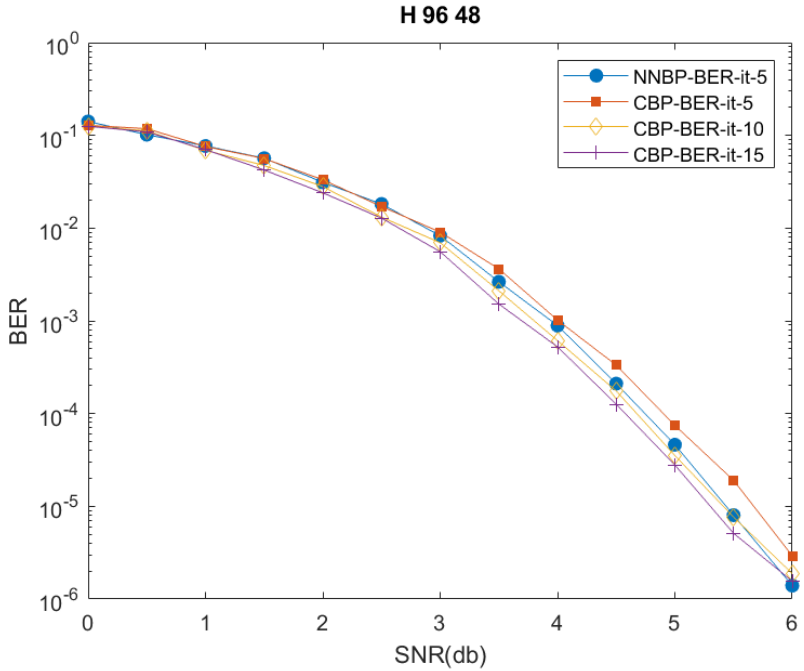
NNBP Decoding without SNR

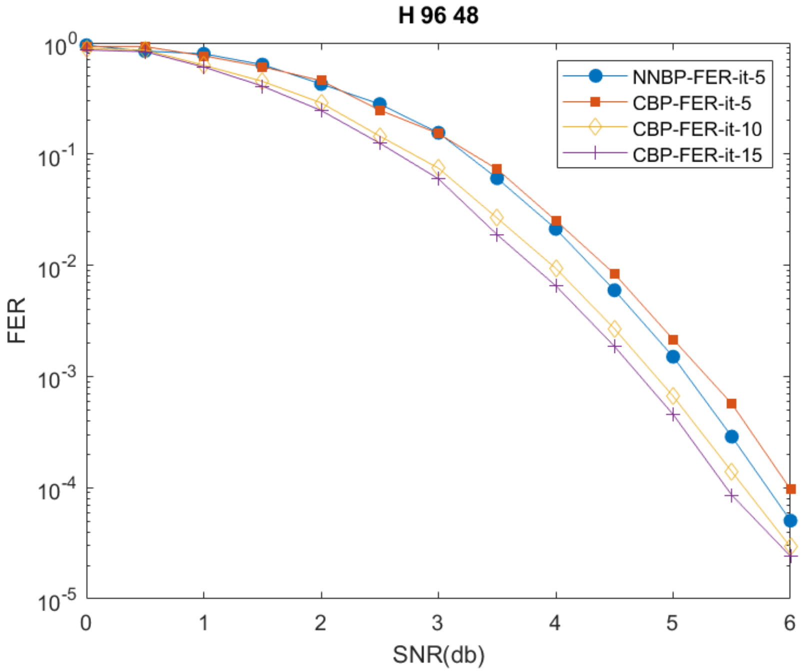
To compare with the conventional BP algorithm without SNR using different numbers of iterations.



NNBP Decoding without SNR

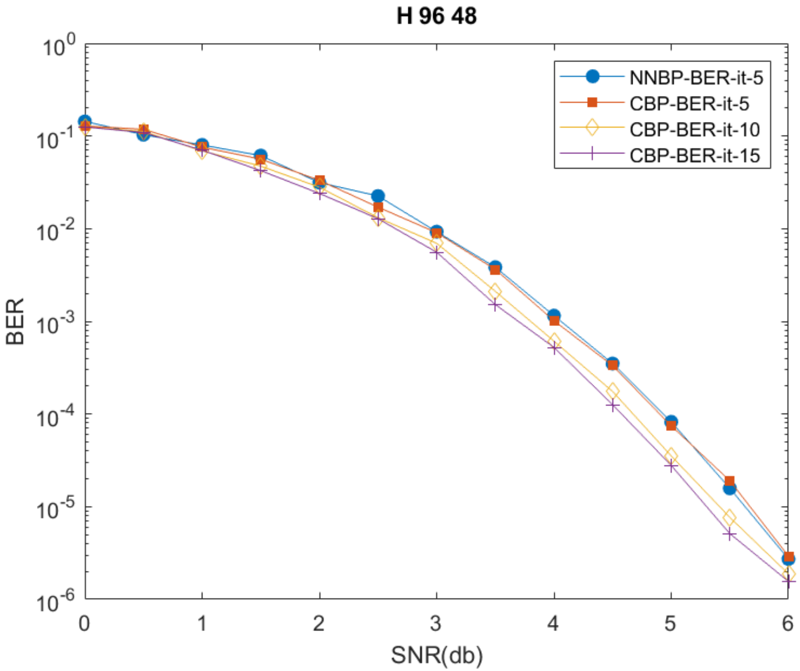
To compare with the conventional BP algorithm using different numbers of iterations.

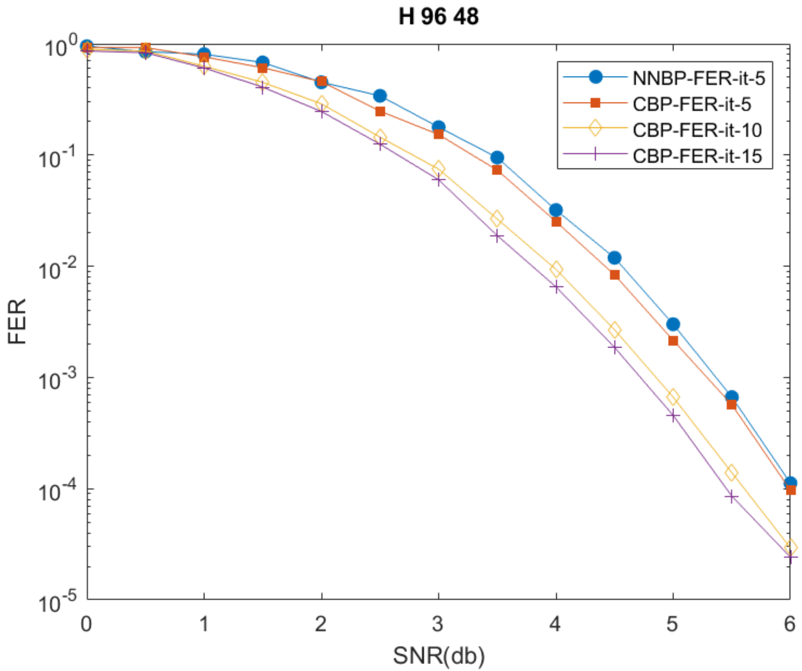




NNBP Decoding without SNR

Comparison of the original model's performance and the performance after quantization.





Conclusion(3)

- The model trained without SNR matches the performance of the conventional BP with known SNR. Additionally, using 5-bit quantization reduces the overall complexity.

Outline

- Review
- Introduction
- NNBP decoding of BCH
- NNBP decoding of LDPC
- NNBP decoding of Hard Decision
- NNBP decoding of LDPC without SNR
- **Future Research Directions**
- Reference

Future Research Directions

- Since the hidden layers of neural network belief propagation are composed of CN updates and VN updates, they can be transformed into Recurrent neural network (RNN) and adopt a weight-sharing mechanism to reduce overall complexity.
- The standard Belief Propagation (BP) implementation can be expensive due to the multiplications and hyperbolic functions needed for the check node function. As a result, the min-sum approximation is often used in practical decoders to reduce complexity.

Outline

- Review
- Introduction
- NNBP decoding of BCH
- NNBP decoding of LDPC
- NNBP decoding of Hard Decision
- NNBP decoding of LDPC without SNR
- Future Research Directions
- **Reference**

Reference

- E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein and Y. Be'ery, "Deep Learning Methods for Improved Decoding of Linear Codes," in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 119-131, Feb. 2018.
- E. Nachmani, Y. Be'ery and D. Burshtein, "Learning to decode linear codes using deep learning," 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 2016.
- Q. Wang et al., "Normalized Min-Sum Neural Network for LDPC Decoding," in IEEE Transactions on Cognitive Communications and Networking, vol. 9, no. 1, pp. 70-81, Feb. 2023.

Thanks