

Optimization and Improvement of BP Decoding Algorithm for Polar Codes Based on Deep Learning

Li Ge

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail:1749018805@qq.com

Guiping Li

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 15693685@qq.com

Abstract—In order to solve the high latency problem of polar codes belief propagation decoding algorithm in the 5G and the dimension limitation problem of belief propagation decoding algorithm under deep learning, a multilayer perceptron belief propagation decoding (MLP-BP) algorithm based on partitioning idea is proposed. In this work, polar codes is decoded using neural networks in partitioning, and the right transfer message value of BP decoding algorithm is also set to complete the propagation process. Simulation results show that, compared with BP decoding algorithm, the proposed algorithm has better decoding performance, reducing the decoding latency, and it is also applicable to long polar codes.

Keywords-Polar Codes; Belief Propagation; Deep Learning

I. INTRODUCTION

In 2008, Professor Arikan first proposed polar codes [1], which was rigorously proved from the mathematical point of view and finally obtained polarization phenomenon of channel. Polar codes can reach the Shannon limit theoretically and construction method can be uniquely determined, so it has low coding complexity and simple implementation. It has been adopted as the control channel coding scheme for the 5th Generation Mobile Networks (5G). The basic decoding

algorithms of polar codes mainly include sequential cancellation (SC) [2] decoding algorithm and belief propagation (BP) [3] decoding algorithm. The SC decoding algorithm has low computational complexity, but due to its sequential decoding nature, there is a significant decoding delay problem in the long code decoding process. Compared with SC decoding algorithm, BP decoding algorithm is parallel decoding, with higher throughput rate and lower delay, so it can better meet the communication requirements of 5G.

With the development of neural networks, deep learning techniques have attracted much attention due to their good performance in many tasks, such as speech recognition, games, machine translation, and autonomous driving [4][5]. In communication systems, for the general channel decoding problem, it can be considered as a classification of information, so deep learning techniques [6] are applied to polar codes decoding as a way to improve the decoding performance of polar codes [7]. Compared to the traditional iterative decoding, deep learning techniques use a cascade of multiple layers of nonlinear processing units to extract and transform the features contained in the coding structure and noise features by pre-trained neural networks that pass through each layer only once to

calculate their estimates, called one-time decoding. Low latency decoding is achieved and the decoding performance is close to the maximum a posteriori probability (MAP) performance [8]. In addition, the high-speed requirements can be easily met by powerful hardware such as parallel computing and graphics processing units (gpu) using existing deep learning platforms, such as Tensorflow [9].

Although neural network decoding has led to a better decoding performance, the training complexity of decoding is exponentially related to the number of information bits, which makes neural network decoding limited by short block lengths [10]. The literature [11] proposed three types of neural network decoders built on multilayer perceptron (MLP), convolutional neural network (CNN), and recurrent neural network (RNN) with the same parameter size, and compared the performance of the three deep neural networks through simulation and concluded that each type of deep neural network has a saturation length, which is due to their limited learning capacity caused by their limited learning capacity. To solve this problem, the literature [12] proposes to divide the coding graph into sub-blocks and train them individually to approach the maximum a posteriori performance of each sub-block, and then connect these blocks through the remaining traditional confidence propagation decoding stage, the resulting decoding algorithm is non-iterative and essentially enables a high level of parallelization while showing a competitive BER performance. Meanwhile, literature [13] proposed a division-based belief propagation neural network algorithm that replaces the sub-blocks of belief propagation decoder with a neural network and connects them using a BP decoding framework, which improves the decoding performance in long codes case and reduces the computational

complexity and latency of the algorithm compared to the traditional BP algorithm.

In this paper, a partitioning method is combined with a multilayer perceptron to optimize polar codes belief propagation decoding algorithm. Mainly, polar codes are divided into several sub-blocks, and the last layers of BP decoding are replaced by trained neural network blocks, and the initial value of right message during BP iteration is set. The simulation results show that the proposed method has better performance in terms of BER and decoding delay compared with traditional BP decoding algorithm. And it can be applied to long codes.

II. BASIC THEORY OF POLAR CODES

A. Polar Codes Encoding

Polar codes can be described by (N, K) , where N is code length and K is number of bits of information. The core construction principle is "channel polarization" [14]. By polarizing channel, (N, K) polar codes divides the channel into two parts: K noiseless channels are used to transmit information bits, and remaining $N-K$ completely noisy channels transmit frozen bits. Let $u_1^N = (u_1, u_2, \dots, u_N)$ be the source vector, $x_1^N = (x_1, x_2, \dots, x_N)$ be the codes word vector, the generation matrix of polar codes is defined as follows:

$$x^N = u^N G_N = u^N F^{\otimes n} B_N \quad (1)$$

Where $n = \log_2 N$, B_N represents the bit-inversal permutation matrix and $F^{\otimes n}$ is the n -th Kronecker power of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Figure 1 shows the structure of polar codes, and it can be seen that polar codes has a recursive structure, for $N=8$ polar codes, it includes two $N=4$ polar codes structures and four $N=2$ polar codes structures, according to the structure of

Figure 1, polar codes can be decoded using BP decoding algorithm.

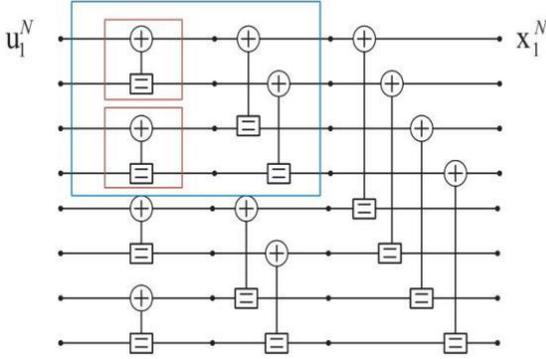


Figure 1. Structure of polar code

B. BP Decoding Algorithm for Polar Codes

BP decoding algorithm is a widely used message transmission algorithm, and one of the most important features compared with other decoding algorithms is that it can decode in parallel, which can solve the delay problem well, and it mainly realizes decoding by iterating the left and right cycles of the factor graph.

The factor graph of (N,K) polar codes consists of $n = \log_2 N$ stages and a total of $N \times (n+1)$ nodes. Each node includes two types of information, the left information propagated from right to left $L_{i,j}^{(t)}$ and the right information propagated from left to right $R_{i,j}^{(t)}$, where node (i, j) denotes the j -th input bit of the i -th stage and t denotes the t -th iteration. The BP decoding of polar codes is initialized as follows:

$$R_{i,j}^{(1)} = \begin{cases} 0, & \text{if } j \in A \\ +\infty & \text{if } j \in A^c \end{cases} \quad (2)$$

$$L_{n+1,j}^{(1)} = \ln \frac{P(y_j | x_j = 0)}{P(y_j | x_j = 1)} \quad (3)$$

Where A and A^c are the set of information bits and the set of freeze bits, respectively.

In the processing element, the updated information of a node of the outgoing processing element is determined by the information of the other three nodes of the incoming processing element. The information update formula as follows:

$$\begin{cases} L_{i,j} = g(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2^i}) \\ L_{i,j+N/2^i} = g(R_{i,j}, L_{i+1,2j-1}) + L_{i+1,2j} \\ R_{i+1,2j-1} = g(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2^i}) \\ R_{i+1,2j} = g(R_{i,j}, L_{i+1,2j-1}) + R_{i,j+N/2^i} \end{cases} \quad (4)$$

Among them:

$$g(x, y) = \ln \left(\frac{1+xy}{x+y} \right) \quad (5)$$

In order to reduce the computational complexity of BP decoding algorithm, the minimum sum (Min-Sum,MS) approximation[19] is used to approximate $g(x, y)$ as follows:

$$g(x, y) \approx \text{sign}(x) \text{sign}(y) \min(|x|, |y|) \quad (6)$$

After T iterations, the j -th estimated information bit can be obtained by hard decision on the left information log-likelihood ratio $L_{i,j}^{(T)}$ output from the last iteration:

$$\hat{u} = \begin{cases} 0, & \text{if } L_{i,j}^T \geq 0 \\ 1, & \text{if } L_{i,j}^T < 0 \end{cases} \quad (7)$$

C. Deep Learning Theory

Deep learning is one of the hottest technologies today, which allows systems to learn effective algorithms directly from training data. Deep Neural Networks (DNN), also known as deep feedforward neural networks, is a typical deep learning model. A deep neural network model can be abstracted as a function f that maps input $x_0 \in \mathbb{R}^{N_0}$ to output $y \in \mathbb{R}^{N_L}$:

$$y = f(x_0; \theta) \quad (8)$$

Where x_0 and y denote the input and output. θ Represents the optimal parameter solution of mapping between known input and expected output values. In general, DNN have a multilayer structure consisting of many functional units, as shown in Figure 2, with multiple hidden layers between the input and output layers.

In neural networks, the parameters in input and output mapping usually refer to weights and deviations. The weight reflects the degree of influence between neurons, while the deviation describes whether the neurons are activated. In order to minimize the loss function, the two parameters are iteratively adjusted using back propagation (BP) or random gradient descent (SGD) during the training phase until the DNN converges and stabilizes.

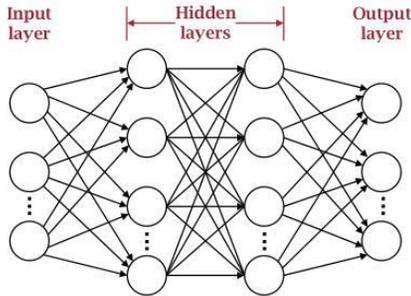


Figure 2. Multi-layer structure of deep neural network

Convolutional Neural Networks (CNN) is also a class of feedforward artificial neural networks, which have been widely used in computer vision tasks such as image classification, recognition, and segmentation in recent years. Recurrent Neural Network (RNN) is a class of neural networks with recursive structure, that is, the current output of a sequence is related to the previous output. Traditional RNN has serious disappearance and explosion gradient problems. Therefore, people usually use its variants, such as

Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

III. NEURAL NETWORK-BASED POLAR CODES DECODING

Neural network decoder is a classification problem in supervised learning. The current neural network decoder system diagram is shown in the figure 3, which mainly replaces the decoder in traditional receivers.

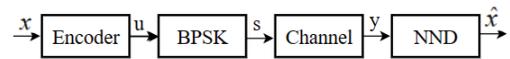


Figure 3. Block diagram of neural network based decoder system

At the transmitter, the information bit length of the source u is K , which is changed to x after polarization coding, and the code word length is set to N . The K information bits and the other $(N-K)$ frozen bits are respectively arranged on reliable bit channels and unreliable bit channels, and then multiplied by the coding matrix of polar codes G_N , this step satisfies equation (1). The encoded bits undergo Binary Phase Shift Keying (BPSK) to complete the modulation process into a modulation sequence s, s satisfying the formula:

$$s = -2x + 1 \quad (9)$$

After the modulated sequence passes through the channel, noise interference is added to obtain the noise-added sequence y , where the received sequence y satisfies the following formula:

$$y = s + n \quad (10)$$

Gaussian white noise is a common interference in the channel, so n is set here to meet the standard normal distribution of Gaussian white noise. At the receiving end, the sequence y is input into the neural network to complete the classification process and output the estimated information bits \hat{u} , which completes the decoding operation. In literature [6], it is pointed out that when training

neural network decoders, there is always an optimal SNR in the training dataset. Therefore, this article uses a normalized validation error (NVE) evaluation function to measure the impact of selecting SNR on network training results.

$$NVE(\rho_t) = \frac{1}{S} \sum_{s=1}^S \frac{BER_{NND}(\rho_t, \rho_{v,s})}{BER_{MAP}(\rho_{v,s})} \quad (11)$$

Where ρ_t and ρ_v represents the SNR of the training set and the SNR of the test set, and $BER_{NND}(\rho_t, \rho_v)$ represents the performance of the neural network trained under the ρ_t training set in the test set. $BER_{MAP}(\rho_v)$ represents the performance of the MAP decoding algorithm in the test set. S represents a total of s SNR test sets for testing.

The literature [7] and [16] used neural networks to improve BP decoding algorithm and achieved good performance. However, the complexity of the neural network and the corresponding training difficulty are positively correlated with the difficulty of decoding, and the size of the codebook set is exponentially related to the length of the information bits. When the length of information bits is long, it is difficult to train a suitable neural network. Below, some comparative simulations will be conducted to demonstrate this problem.

TABLE I. PARAMETERS SETTINGS

Parameters	Value
code length	8, 16, 32
code rate	0.5
batchsize	512
learning rate	0.001
training set size	106
epoch	103
network structure	32-16-8, 128-64-32, 512-256-128

Three polar codes with code lengths of 8, 16 and 32 with code rate R=0.5 are selected and trained with the same training set size and training period to compare their decoding performance. The relevant parameters are shown in the table I and II.

TABLE II. NETWORK STRUCTURE

	32-16-8	128-64-32	512-256-128
N=8	1024	11752	169846
N=16	1352	13488	174992
N=32	1352	13488	174992

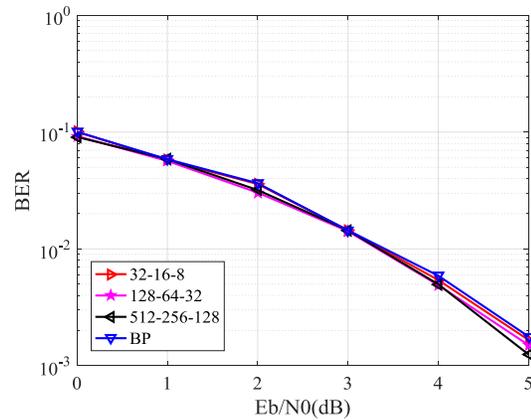


Figure 4. Performance of different network structures at N=8

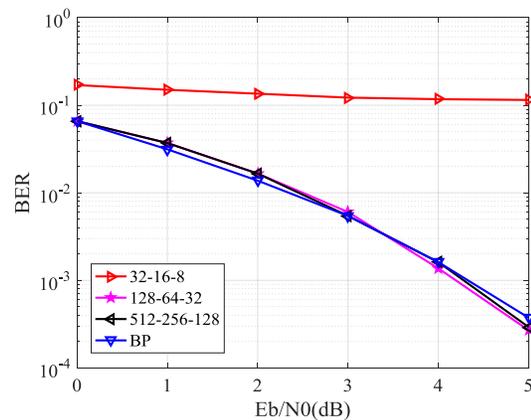


Figure 5. Performance of different network structures at N=16

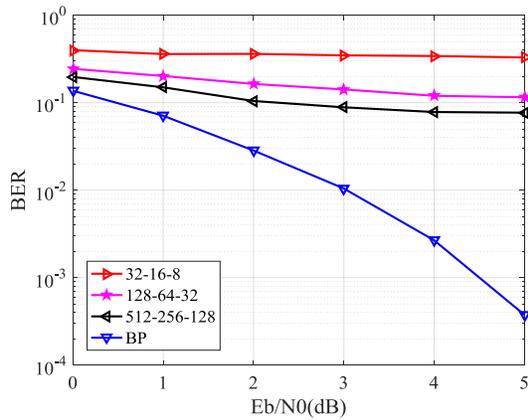


Figure 6. Performance of different network structures at N=32

The simulation results show that when the code length is very short, the decoding performance of the three network structures is similar, and their performance is better than the traditional BP decoding algorithm. For the case of N=16, the network structure of 32-16-8 cannot be decoded correctly. However, for the case of N=32, none of the three network structures can achieve the decoding function, requiring a more complex network structure to complete this task. However, the number of parameters for the 512-256-128 network structure is already very large. Therefore, it can be seen that the exponential growth of the complexity of the network structure with the length of information bits will seriously restrict the development of neural network decoder technology.

IV. PROPOSED MLP-BP DECODER MODEL

From the analysis in the previous section, it can be seen that although the emergence of neural network decoders is a significant breakthrough in traditional decoding, they still cannot overcome the problem of dimensional constraints. Therefore, in order to further expand the application of neural network decoders in long codes, this section will propose a partitioning based neural network

decoding scheme combining the coding characteristics of polar codes themselves.

According to the structure of polar codes in Figure 1, it is a recursive coding scheme, and long polar codes can be seen as a combination of several short polar codes. The traditional BP decoding algorithm is an iterative decoding algorithm, and its number of iterations is directly related to the performance and accuracy of decoding. The more iterations, the more accurate the result, but its computational complexity is also positively correlated with the number of iterations. The more iterations, the greater the decoding delay. Therefore, inspired by literature [12] [13] and literature [17] [18], this article introduces a neural network decoder to decode the sub blocks, which serves as a substitute for the last layers of BP decoder, while combining BP algorithm to complete the information exchange between the sub blocks. For example, dividing into two decoding blocks is shown in Figure 7:

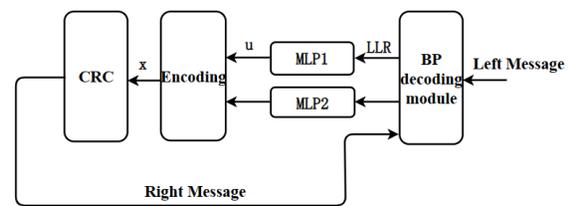


Figure 7. Structure diagram of the proposed MLP-BP

The interaction between BP component and NND component is shown in Figure 8.

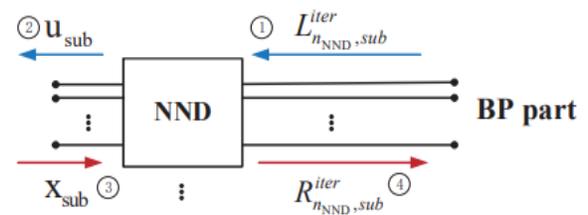


Figure 8. Interaction of BP and DNN blocks

The decoding is divided into two parts. The neural network block part contains multiple parallel MLP decoding blocks, each decoding block corresponding to polar codes sub block, arranged in parallel at the leftmost end of BP decoding. Replacing a part of BP decoding with neural network blocks combines the advantages of BP decoding algorithm and MLP decoding algorithm, maintaining parallel decoding. For the proposed algorithm, the size of NND portion is not fixed, but is determined by the training ability and performance requirements of the network. The larger the NND portion, the less the number of blocks, and the closer the entire system structure is to traditional neural network decoders, resulting in a significant increase in the difficulty of neural network training; On the contrary, the larger BP part, the closer its decoding performance will be to the performance of BP algorithm.

The whole module input is y_N , output is u_N , set the maximum number of iterations to $iter_{max}$, first initialize the rightmost L information to equation(3), then according to equation (4) will pass the information to the left, when it reaches the neural network module, using the pre-trained NND sub-block to calculate the corresponding u_{sub} , It should be noted that although current neural network decoders generally use directly acquired channel values as the actual data source for training data and decoding, the literature [4] points out that using the log-likelihood ratio LLR as the input to the network can also achieve good decoding results.

Since the MLP neural network decodes each sub-block, the information between the sub-blocks is not transmitted to each other. At this time, the BER curves of different sub-blocks u_{sub} vary greatly and the overall performance is also poor, therefore, it is necessary to continue to update the

R information from left to right. For the conventional BP decoding, as shown in equation (2), the R information of the information bits is initialized to 0; the frozen bits are regarded as check information, and their R values are initialized to infinity. For the scheme proposed in this section, since the left end of BP network is not docked to the original information sequence u, the decoding results of the sub-blocks need to be recoded to x_{sub} . In fact, in the first few iterations, most of the sub blocks cannot obtain completely accurate decoding results, so it is not possible to simply initialize x_{sub} according to the frozen bits. However, if all of them are initialized to 0, it will also cause the loss of verification information, making it impossible to obtain gain in subsequent iterations. Therefore, a setting for right transfer information is proposed to minimize information loss caused by it. As shown in the following formula:

$$R_{n_{NND},sub}^{iter} = \frac{iter}{R_i} * x_{sub} \quad (12)$$

$R_{n_{NND},sub}^{iter}$ is the initial value of R message in BP iteration, $R_{n_{NND},sub}^{iter}$ is related to the number of current iterations $iter$ and the code rate of sub-block R_i , the more iterations, the more reliable the initialized R message, the smaller the code rate, the more accurate the initialized R message. After completing the initialization of the R information, the network continues to transfer the rule of equation (4) to the rightmost stage of BP section. After multiple iterations, the overall error rate performance gradually converges to an ideal value. The summary of the entire MLP-BP decoding algorithm is shown in Algorithm 1.

After the CRC part is added to the recoding, because when the sub block is very small, CRC verification cannot be added. When the sub block

is larger, even if CRC verification can be added, it will reduce the code rate and cause performance losses. Therefore, the CRC check portion is added to all codewords.

Algorithm 1: Proposed MLP-BP decoding algorithm

- 1: **Enter.** y_0, y_1, \dots, y_{N-1}
 - 2: **Output.** u_0, u_1, \dots, u_{N-1}
 - 3: Initialization: Initialization using (2) $LLR(y_j)$
 4. *for* iter $\leftarrow 1$ to iter_{max} *do*
 5. *for* $i \leftarrow n+1$ to n_{NND} *do*
 - 6: Update using equation (3) $L_{i,j}^{iter}$
 7. *end for*
 - 8: After reaching NND use the sub-block NND_{sub} to calculate u_{sub}
 - 9: u_{sub} After recoding to get x_{sub}
 - 10: *if* after encoding x_{sub} by CRC checksum *do*
 - 11: Using equation (7) yields $R_{n_{NND},sub}^{iter}$
 12. *end if*
 - 13: Retransmission
 14. *for* $i \leftarrow n_{NND}$ to n *do*
 - 15: Update using equation (3) $R_{i+1,j}^{iter}$
 - 16: *end for*
 - 17: *end for*
-

V. SIMULATION RESULTS AND ANALYSIS

In the decoding problem of this section, the Gaussian approximation construction method is used, and the randomly generated data is used as the training set. For (8,4) polar code, the information bits are {3,5,6,7}; for {16,8} polar code, the information bits are {7,9,10,11,12,13,14,15}, and these two short code blocks are used as references to divide long codes. For (32,16) code, the information bits are {12,14,15,16,19,21,22,23,24,25,26,27,28,29,30}

From the table 3, it can be seen that (32,16) code can be divided into two 16 length codes, with the first block code rate of 0.25 and the second block code rate of 0.75. After block training, they are spliced.

TABLE III. POLAR(32,16) DIVIDED INTO TWO PARTS

Partition	Information bits	Code Rate
[0-15]	{11,12,14,15}	0.25
[16-31]	{19,21,22,23,24,25,26,27,28,29,30,31}	0.75

TABLE IV. POLAR(32,16) DIVIDED INTO FOUR PARTS

Partitioning	Information bits	Relative Location	Code Rate
[0-7]	None	None	0
[8-15]	{11,12,13,14}	{3,5,6,7}	0.5
[16-23]	{19,21,22,23}	{3,5,6,7}	0.5
[24-31]	{24,25,26,27,28,29,30,31}	{0,1,2,3,4,5,6,7}	1

By analyzing the information bits, not for a new code length to retrain a set of MLP decoding block, the (32,16) is divided into four 8-long code words: the first block code rate of 0, the fourth block code rate of 1, both of which do not require network training, using a hard judgment can be, for 2, 3 two intermediate blocks, through Table 4 compared with Table 3, equivalent to two (8,4) polar code neural network decoding block. The relative positions of the information bits are the same, and the two decoding blocks can be shared, realizing the reuse of the network decoding blocks. In general, when the code lengths are the same and the relative positions of the contained information bits are the same, the same decoding block can be used. When the MLP-BP decoder is applied to longer code words, the MLP block will have duplicate parts, which reduces the required neural network blocks and improves the reusability.

A. Simulation Parameter Setting

TABLE V. PARAMETER SETTING

Set options	Value
Test platform	Tensorflow
Encoding	Polar(32,16)、(64,128)
Signal to noise ratio	1~5dB
loss function	Cross Entropy Loss
Optimizer	Adam

B. Decoding Performance Analysis

Firstly, the change trend of the loss value with the training period is analyzed. From Figure 9, it can be seen that the loss value decreases and gradually stabilizing at a certain value as the number of training times increases, indicating that the proposed algorithm MLP-BP can effectively converge through training. In addition, Figure 10 also analyzes the performance changes of BER and FER when SNR is 4 dB are analyzed, and it can be seen that as the number of training increases, the performance of the algorithm gradually decreases and is superior to traditional BP decoding algorithms.

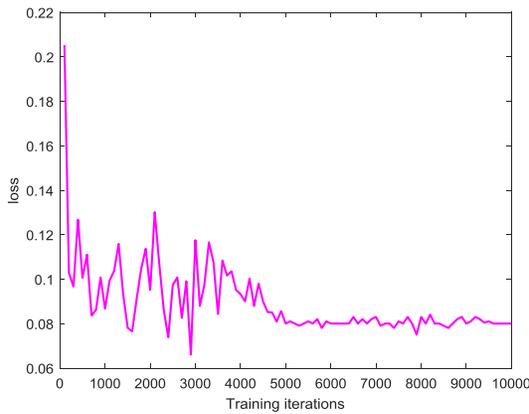


Figure 9. Change of MLP-BP training loss value when N=128

Next, we discuss the advantages of the proposed algorithm compared to traditional BP decoding algorithms.

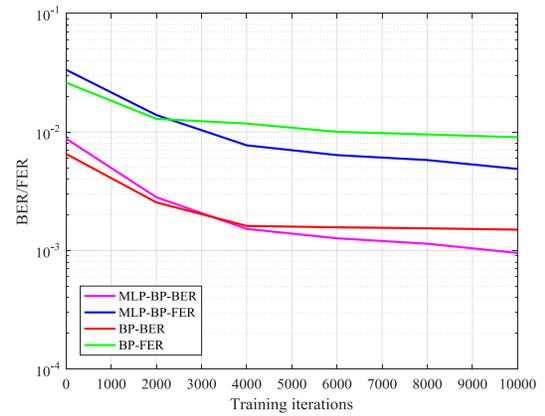


Figure 10. Evolution of MLP-BP and BP BER when N=128

Experiments are first conducted using code word N=32, and the code word is divided into 4 blocks, each of which had 8 bits. The trained model is saved, and after the training of the sub blocks is completed, it is cascaded to the end of BP decoding to replace the last three layers of BP. The performance comparison with BP decoding both using 30 iterations is shown in Figure 11.

Then, the model is applied to a polar code with code length of N=128, and the code is divided into 16 blocks, each containing 8 code words for decoding, the decoding results are shown in Figure 12:

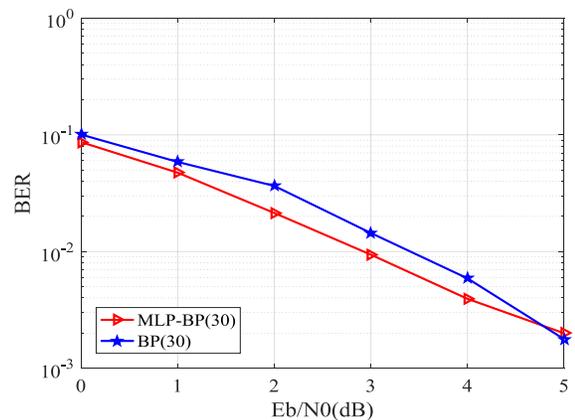


Figure 11. BER performance comparison of two decoding methods at N=32

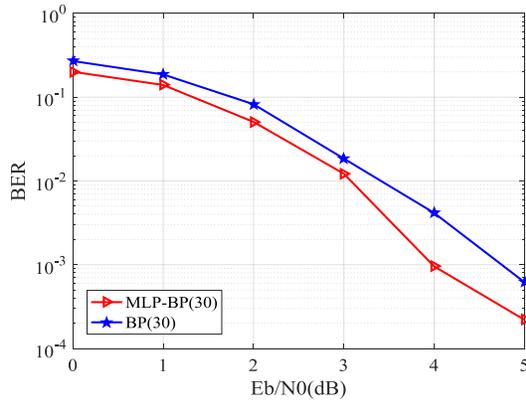


Figure 12. BER performance comparison of two decoding methods at $N=128$

For (128, 64) polar code, the MLP-BP decoding model has a performance improvement over BP decoding, with MLP-BP having a 0.2 dB gain over BP for BER 0.01 to 0.2, and MLP-BP having a 0.4 dB gain over BP for BER 0.001.

C. Decoding Time Delay Analysis

The decoding delay of the BP decoding algorithm for polar codes of code length N with a BP iteration number of l can be expressed by the time step as

$$\tau_{BP} = 2l \log_2 N \quad (13)$$

And the decoding delay required by the MLP-BP decoding algorithm proposed in this paper is determined by the code length N and the number of iterations l and the size of the sub-block N_{sub} and the number of hidden layers of the network m . The expression is:

$$\tau_{MLP-BP} = l \cdot \left(2 \log_2 \frac{N}{N_{sub}} + m \right) \quad (14)$$

For (32,16) polar code, the decoding delay of each algorithm is shown in Table 6, when the number of iterations is 30, the proposed

block-based MLP-BP decoding algorithm reduces the time step by 81.1% compared with BP decoding algorithm.

TABLE VI. DECODING TIME DELAY

Algorithm	BP	MLP-BP
Decoding time delay	380	72

VI. CONCLUSIONS

In this paper, we propose an MLP-BP decoding algorithm for polar codes based on the idea of partitioning, which combines the advantages of BP algorithm and MLP to maintain parallel decoding. At the same time, a method for setting the right value information during intermediate iterations is set, and the MLP neural network blocks are nested into the final layers of BP decoding. Simulation results show that compared to traditional BP decoding algorithm, the proposed MLP-BP decoding algorithm improves the bit error rate, reduces the number of iterations required, and reduces decoding delay in disguised form under the same performance conditions. In future research, further research should be conducted on the structural optimization of neural networks and the design of network parameters, with the aim of improving the decoding performance of polar codes BP decoding algorithm.

ACKNOWLEDGMENTS

This paper is supported in part by the Science and Technology Plan Project of Xi'an Beilin District (GX2214) under Grant, and in part by the Plan Project of the Xi'an Science and Technology (22GXFW0047) under Grant.

REFERENCES

- [1] Arikan E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric

- Binary-Input Memoryless Channels [J]. IEEE Transactions on Information Theory, 2009, 55(7):3051-3073. [J].
- [2] Alamdar-Yazdi A, Kschischang F R. A Simplified Successive-Cancellation Decoder for Polar Codes [J]. IEEE Communications Letters, 2011, 15(12):1378-1380. [J].
- [3] Bo Yuan, Keshab K. Parhi. Early Stopping Criteria for Energy-Efficient Low-Latency Belief-Propagation Polar Code Decoders. [J]. IEEE Trans. Signal Processing, 2014, 62(24).
- [4] K. Fithriasari and U. S. Nuraini, Face identification using multi-layer perceptron and convolutional neural network, ICIC Express Letters, vol.15, no.2, pp.157-164, 2021.
- [5] C. Lee and B.-D. Lee, Enhancement for automatic extraction of RoIs for bone age assessment based on deep neural networks, ICIC Express Letters, vol.14, no.2, pp.163-170, 2020.
- [6] Gruber T, Cammerer S, Hoydis J, et al. On Deep Learning-Based Channel Decoding [J]. IEEE, 2017. [J].
- [7] Xu W, Wu Z, Ueng Y L, et al. Improved polar decoder based on deep learning[C]// 2017 IEEE International Workshop on Signal Processing Systems (SiPS). IEEE, 2017. [J]. [1] Seo J, Lee J ,
- [8] Kim K. Decoding of Polar Code by Using Deep Feed-Forward Neural Networks[C]// 2018 International Conference on Computing, Networking and Communications (ICNC). IEEE Computer Society, 2018.
- [9] Abadi M, Barham P, Chen J, et al. TensorFlow: A system for large-scale machine learning [J]. USENIX Association, 2016. [J].
- [10] Xiao-An Wang, Wicker S.B.. An artificial neural net Viterbi decoder [J]. IEEE Transactions on Communications, 1996, 44(2).
- [11] Lyu W, Zhang Z, Jiao C, et al. Performance Evaluation of Channel Decoding with Deep Neural Networks [J]. IEEE, 2018.
- [12] Cammerer S, Gruber T, Hoydis J, et al. Scaling Deep Learning-based Decoding of Polar Codes via Partitioning[C]// GLOBECOM 2017 - 2017 IEEE Global Communications Conference. IEEE, 2017.
- [13] Chen W, Jian X, Lin G, et al. A BP-NN Decoding Algorithm for Polar Codes[C]// 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2019.
- [14] Liu Rongke, Sun He, Feng Baoping, et al Overview of research on polarization codes Telemetry and remote control [J]
- [15] Yuan B, Parhi K K. Architecture optimizations for BP polar decoders[C]// Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013. [J].
- [16] Teng C F, Chen-Hsi, Ho K S, et al. Low-complexity Recurrent Neural Network-based Polar Decoder with Weight Quantization Mechanism [J]. 2018.
- [17] Xu Xiang, Research on Polar Codes Decoding Algorithm Based on Deep Learning [D]. Beijing Jiaotong University, 2019
- [18] Han Xiaoyi, Research on Polar Codes Decoding Algorithm Based on Neural Network [D]. Xi'an University of Electronic Science and Technology, 2021