Recurrent Neural Network-Aided BP Decoder Based on Bit-flipping for Polar Codes

Guiping Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: liguiping@xatu.edu.cn

Xiaojie Liu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 2719130786@qq.com

Chang Yun
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: llxghp@sina.com

***Abstract*—Compared with SC decoding, BP decoding with the parallel mechanism has higher throughput and lower latency, which is more suitable for the demand of 5G scene. To further improve its FER performance and reduce the memory overhead, a recurrent neural network-aided bit-flipping BP decoding of polar codes is proposed. Firstly, it uses bit flip to correct the wrong decoded bits during the decoding iteration. And then, the offset min-sum approximation is used to replace multiplication operation. Lastly the improved recurrent neural network architecture is adopted to realize parameter sharing. The simulation shows that the proposed scheme has a better error correction ability with fewer flipping times, and can effectively reduce the computational resource consumption and extra memory overhead of BP decoding.**

Keywords-Polar Codes; Belief Propagation; Recurrent Neural Network; Bit-flip

I. INTRODUCTION

With the rapid development of Internet of Things (IoT), a growing number of physical devices are being connected at an unprecedented rate. The emerging 5G-IoT-centric applications like augmented reality, high-resolution video streaming, self-driven cars, smart environment and e-health care etc all require low latency, high throughput and ultra-reliable communication. Since polar codes proposed by E. Arikan in 2008 based on the polarization phenomenon of channel [1] are low complexity codes and can be proved to achieve Shannon capacity theoretically under successive cancellation (SC) decoding when the code length approaches infinity. Due to these excellent characteristics, polar codes are selected as the control channel coding in the enhanced Mobile Broadband (eMBB) scenario in the 5G mobile communication system plan. Since 2016, polar codes have been from theoretical research to engineering application less than 10 years after they were proposed.

In the recent academic research, the decoding algorithms of polar codes mainly fall into the following two categories, namely the Successive Cancellation (SC) [2] decoding and the Belief Propagation (BP) [3] decoding. Among them, SC decoding include its derivative decoding Successive Cancellation List (SCL) [4,5] and BP decoding are two kinds of commonly used algorithms in polar code decoding algorithm. The SC decoding algorithm has low computational complexity, but is limited by the nature of serial decoding, resulting in low throughput and high latency in the decoding process. Compared with the SC decoding algorithm, the BP decoding algorithm has higher throughput and lower decoding latency due to its parallel decoding characteristics, but it also leads to an increase in computational complexity, at the same time, the BP algorithm

DOI: 10.2478/ijanmc-2025-0004

iteratively the characteristics of decoding also make the decoding delay increase with the increase of the number of iterations.

In recent years, with the rapid development of deep learning, neural network technology has injected fresh blood into many fields. Due to the similarities between the neural network and the decoding network structure, many scholars have tried to apply the neural network to polar code decoding to reduce the decoding latency compared with the existing traditional decoding methods. Tobias Gruber [6] proposed a polar codes decoding scheme based on Deep Neural Network (DNN) in 2017, using DNN to learn the mapping relationship of information before and after decoding can effectively replace the traditional complex decoding algorithm at the receiving end, simulation proved that its decoding performance can reach the maximum a posteriori probability (MAP) performance close to that of the CA-SCL decoding algorithm. At present, the polar codes decoding algorithm based on neural network reduces the decoding latency to a certain extent, at the same time, the neural network decoding algorithm [7,8] still has certain limitations, the performance of the code has a great impact, and the memory overhead of the neural network BP decoding algorithm, the decoding error correction performance and the difficulty of learning the codeword structure mapping of SC decoding still need to be improved. Therefore, improving the generalization ability of neural network decoding to achieve optimal decoding performance with lower complexity is of great significance in future research.

Compared with BP decoding, SC decoding and the improved SC algorithm can achieve better channel capacity and lower block error rate (BLER), however, SC decoding has low throughput due to its sequential processing characteristics, while BP algorithm has outstanding performance in parallelization and low decoding latency. In recent years, the research of BP decoding is also trying to improve the decoding performance of BP while maintaining its advantages. One way to optimize the performance of BP decoding algorithm is to add neural network to assist the BP decoding process, expand the BP iterative decoding structure into the neural network architecture [9], and optimize the minimum and approximate scaling factors to compensate for the performance loss through deep learning. In reference [10,11], the BP decoding algorithm based on neural network reduces the total number of iterations and the overall complexity before convergence by scaling the messages with trainable weights, but it does not solve the problem of the lack of error correction performance of BP decoding. In order to solve a large number of multiplication operations caused by BP neural network decoding, and further reduce the block error rate of the BP decoding algorithm, this paper proposes a recurrent neural network-assisted polar codes bit-flipping (BF) BP decoding algorithm [12-15].

Firstly, this paper introduces some basic theories of polar codes and deep learning, then introduces the construction of bit-flipping and critical set of polar codes BP decoding algorithm. Then mainly introduces the polar codes bit flip BP decoding algorithm based on RNN and makes simulation analysis, and finally, a summary is made and the future of BP decoding is prospected.

II. KEY THEORIES AND ALGORITHMS

A. Polar Code Decoding Theory

Through the phenomenon of channel polarization, it is found that for any $N = 2^n$ independent channels, the channel capacity of some of the polarized channels tends to be polarized, besides, the coding complexity of polar codes is very low, which is mathematically a matrix multiplication operation. At the same time, the proposal of polar codes provides a factual basis for the proof of Shannon's noisy channel coding theorem, which further improves Shannon's theorem. Therefore, polar codes are considered to be an important breakthrough in the coding theorem, the process of polar codes channel polarization conversion is shown in Figure 1.

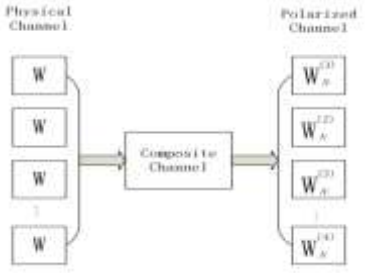

Figure 1. Polarization Transformation.

In the process of constructing the polar code of (N,K), first the information bit length of the source u is K, after polar coding, it becomes x, and the code word length is set to N. The K information bits and other frozen bits of $(N-K)$ will be arranged to the reliable bit channel and the unreliable bit channel respectively, and then multiplied by the coding matrix $\mathbf{G}_N$ of the polar code, which satisfies the formula:

$$\mathbf{x}^N = \mathbf{u}^N \mathbf{G}_N = \mathbf{u}^N \mathbf{F}^{\otimes n} \mathbf{B}_N \tag{1}$$

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

B. BP Decoding Algorithm

SC decoding algorithm is a serial decoding algorithm, which has high latency and low throughput in the decoding process, in order to better solve the decoding latency problem of serial decoding, BP decoding is applied to polar codes decoding. The main process of BP decoding is to adaptively adjust the check matrix according to the reliable information in each iteration, and sparse it to obtain enhanced reliability information and decide the decoding, the BP factor diagram of a (N,K) polar codes has $N \times (n+1)$ nodes and $n = \log_2 N$ stages, the factor diagram of a $(8,4)$ polar codes is shown in Figure 2 and Figure 3.

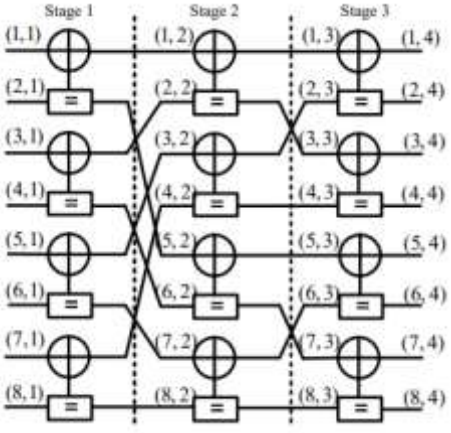

Figure 2. Factor diagram of BP decoding algorithm for $(8,4)$ polar codes.

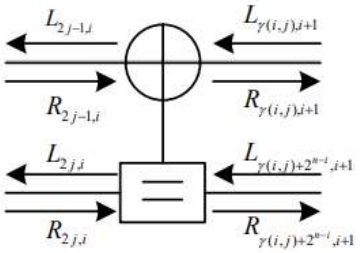

Figure 3. Unit factor diagram of BP decoding algorithm for polar codes.

When performing the BP iterative operation, each node (i,j) has two iterative calculation formulas from right to left and from left to right, by converting the information received by the receiver into log-likelihood ratio information, the for the iterative decoding operation of soft information. The information from left to right is expressed as $R_{i,j}^{(t)}$, and the information from right to left is expressed as $L_{i,j}^{(t)}$, the specific iterative rule for iteratively updating the likelihood ratio information is the following equation:

$$\begin{cases} L_{i,j}^{(t)} = g\left(L_{i+1,2j-1}^{(t)}, L_{i+1,2j}^{(t)} + R_{i,j+\frac{N}{2}}^{(t)} \right) \\ L_{i,j+\frac{N}{2}}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t)} \right) + L_{i+1,2j}^{(t)} \\ R_{i+1,2j-1}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,2j}^{(t)} + R_{i,j+\frac{N}{2}}^{(t)} \right) \\ R_{i+1,2j}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t)} \right) + R_{i,j+\frac{N}{2}}^{(t)} \end{cases} \tag{2}$$

First, the log-likelihood ratio information is passed from the rightmost to the leftmost, and then from the leftmost to the rightmost, to realize an iterative decoding operation, and the above-mentioned likelihood ratio information is updated in this process. Among them, $g(x,y) = \ln \dfrac{1+xy}{x+y}$, since the logarithmic calculation process is more complicated, the min-sum(MS) approximation [9] is generally used to simplify $g(x,y)$:

$$g(x,y) \approx sign(x) \times sign(y) min(|x|,|y|) \tag{3}$$

At the beginning of iteration, log-likelihood ratio information is initialized. The initial values $R_{1,j}^{(1)}$

and $L_{n+1,j}^{(1)}$ of right information log-likelihood ratio and left information log-likelihood ratio can be obtained by initializing log-likelihood ratio, which are expressed as follows:

$$R_{1,j}^{(1)} = \begin{cases} 0, & \text{if } j \in A \\ \infty, & \text{if } j \in A^c \end{cases} \tag{4}$$

$$L_{n+1,j}^{(1)} = \ln \frac{P(y_j | x_j = 0)}{P(y_j | x_j = 1)} \tag{5}$$

Since the BP algorithm is an iterative decoding algorithm, the number of iterations is involved, the superscript of the letter of the initialization formula above represents the number of iterations, because it is initialization, the superscript is 1, indicating the first iteration. Among them, A represents the set of information bits, and A^c represents the set of frozen bits, when the number of iterations reaches the set T times, the next step is the decision stage in the decoding process, the source data u_1^N is judged according to the likelihood ratio information of the leftmost node, and then the estimated value of the bit $\hat{u}_1^N$ is obtained, the specific judgment rules are as follows:

$$\hat{u}_j^N = \begin{cases} 0, & \text{if } L_{1,j}^T \geq 0 \\ 1, & \text{if } L_{1,j}^T < 0 \end{cases} \tag{6}$$

C. Deep Learning Theory

After artificial intelligence has achieved extensive applications and good results in various fields, the concept of deep learning is more and more familiar to people. The most common model of deep learning is deep neural network, as a popular learning model in the academic world, various academic fields try to bring deep learning into the original field, hoping to obtain better performance.

Each neuron of the deep neural network is not only connected with all the neurons in the previous layer to receive the information from them, but also connected with the neurons in the latter layer to transmit the current information. The number of neuron nodes in the input layer is affected by the data to be learned and needs to be consistent with the dimension of the learning data, connected to the input layer is the hidden layer, which receives the information from the input layer. A DNN model can be abstracted as a function f that maps the input $x_0 \in \square^{N_0}$ to the output $\mathbf{y} \in \square^{N_L}$:

$$\mathbf{y} = f(\mathbf{x}_0; \theta) \tag{7}$$

Convolutional Neural Network(CNN) is a deep neural network with a convolutional structure, the reason why the convolutional structure can offset the memory requirements as the number of network layer increases, three of which are crucial. First is the local receptive field, for a two-dimensional image, the connection between each other is usually only local, and it can be considered that the characteristics of each block will only affect the characteristics of the image around it, features are only affected by the features of the images connected to it, so low-level neurons learn local low-level features, and high-level connections are used to statistically synthesize each low-level feature, and gradually learn global features. The second is weight sharing, a convolution kernel is equivalent to a filter, which is used to extract features of one dimension, when the same feature appears in different dimensions, the previous convolution kernel is used to extract this feature, so that different dimensions a convolution kernel is shared, which not only reduces the parameters of the convolution kernel, but also allows the redundant convolution kernel to extract more features. The third is the pooling layer, pooling is a form of down sampling, there are many forms of nonlinear pooling functions, of which max pooling and average pooling are two common methods, through the pooling layer, the feature dimension can be reduced, so that learning fewer parameters is conducive to the establishment of the model, reducing the task of learning, and avoiding the occurrence of overfitting to a certain extent. Therefore, the convolutional neural network is a neural network algorithm that can effectively split the data.

Recurrent Neural Network(RNN) are also called feedback neural networks, the existence of

feedback makes the recurrent neural network a nonlinear dynamic system that can be used to solve problems such as associative memory and optimization calculation, which is also the biggest difference between the recurrent neural network and the feedforward neural network. The recurrent neural network with fixed weights, external input and internal state can be divided into two network models: static field neural network model and local field neural network model. Among them, a typical well-known local field neural network model is called Hopfield-type neural network, and the optimization neural network mainly used to solve linear variational inequality and linear complement problem is one of the more famous static field neural network models.

As the excellent performance of deep neural network has been widely concerned by scholars from all walks of life, the application of neural network in the field of communication has gradually increased, the decoding process can be regarded as the process of signal classification, so the features contained in the coding structure can be learned by training the neural network. Some scholars have applied DNN to the decoding of linear codes and obtained better performance results, using the neural network decoder method, since the network training is completed offline, the decoding latency is not involved in the actual calculation. At present, the polar codes decoding algorithm based on neural network reduces the decoding latency to a certain extent, at the same time, the neural network decoding algorithm still has certain limitations, however, the memory overhead and decoding error correction performance of the neural network-based BP decoding algorithm still need further optimization. Therefore, improving the generalization ability of neural network decoding to achieve optimal decoding performance with lower complexity is of great significance in future research.

III. PRELIMINARY WORKS

A. *Bit-flip*

Incorrect decoding of information bits due to the BP decoding message passing algorithm can lead to error propagation, negatively affecting the reliability and accuracy of many other bits. Bit flipping is an auxiliary mechanism to the decoding process that guesses and flips potentially incorrect decoded bits before restarting the decoding process, therefore, precise bit flipping can effectively improve the BLER performance of polar codes. In order to solve the problem of error propagation, the BF mechanism flips the previously estimated value of u_j^N and sets the prior knowledge of $\hat{u}_j^N$ to infinity, then, the error-propagated decoding information is corrected by using infinite prior LLR instead of directly modifying the hard decision value of the flipped bit, therefore, the initialized value of R_0 in formula (4) is modified as:

$$R_{0,j}^{(1)} = \begin{cases} 0, & \text{if } j \in \{A \setminus F\} \\ \infty \times (2\hat{u}_j^N - 1), & \text{if } j \in F \\ +\infty, & \text{if } j \in A^c \end{cases} \tag{8}$$

where F is the set of flip positions, set to infinity.

B. *Critical set*

During the transmission of polar codes, the main reason for decoding errors is error propagation, by flipping the first error, the decoding performance can be improved. However, the biggest defect of the traditional flip set is that if the first wrong bit is not found, the performance will hardly be improved, and most of the time, there is more than one wrong bit, how to select unreliable information and build the correct flip set is the biggest difficulty. However, the search range of the first error bit is the whole unfrozen bit set, in order to narrow the search range of the first error unfrozen bit in decoding, the critical set (CS) composed of most error-prone bits was proposed and adopted in reference [8-14]. The critical set is constructed according to the structure of polar codes, in which the first node of all information sub-trees is high risk, only by selecting bits for BF from CS, fewer flip attempts can be achieved and lower latency can be achieved. In addition, the reference [12] puts forward the ω-order CS to flip ω-easy dislocations at the same time, at the cost of increasing 2^{ω}-times, with better error correction ability, in the literature [14,15], the BP decoding algorithm of polar codes

based on recurrent neural network and the BP decoding algorithm of polar codes bit-flip based on convolutional neural network were proposed, which achieved good decoding and error correction ability, but have large memory overhead and flipping attempts.

$E\left\{L_N^{(i)}\right\}$ represents the expected value of log-likelihood ratio of the i_{th} polarized channel, then the error rate of the ith polarized channel can be expressed as

$$P_e(u_i) = Q\left(\sqrt{E\left\{L_N^{(i)}\right\}/2}\right) \tag{9}$$

Where $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{+\infty} e^{-\frac{\alpha^2}{2}}\mathrm{d}\alpha$, selecting the indexes of k polarized channels with the largest $E\left\{L_N^{(i)}\right\}$ to form the information bit set A.

The polar codes structure can be represented by a complete binary tree, as shown in Figure 4 [12]. The black node indicates that all its leaf nodes are information bits, while the white node indicates that all its leaf nodes are frozen bits, and the gray node indicates that its leaf nodes contain both information and frozen bits, black nodes are nodes with coding rate of 1, and the critical set needs to take the index of the first bit in each node with coding rate of 1.

$$CS = \bigcup_i \mathcal{R}_i[1] \tag{10}$$

Where $\mathcal{R}_i$ represents the i_{th} node with a rate of 1, and $\mathcal{R}_i[1]$ represents the index of the first bit in $\mathcal{R}_i$.

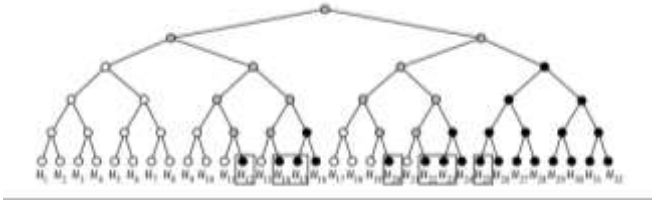

Figure 4. Binary Tree Structure Diagram(polar $^{(32,16)}$).

The bit inversion of CS-1 is the same one-dimensional vector as CS, however, it contains a large number of error-prone exponents (about $|CS|^{\omega}$), which need to be tried by bit inversion decoding. In order to avoid the bit-flip decoding index attempt and obtain lower decoding latency, the truncated version of CS-ω was proposed in [12], CS-ω is a $|CS|\times\omega$ matrix, and one row of CS-ω represents the ω index of ω flip bits in one BP decoding. Each line of CS-ω is shown as follows, first of all, according to the error rate $P_e(u_i)$ and $i \in CS$, the elements in CS are sorted in descending order:

$$\begin{aligned} &\left\{\mathrm{CS}(k_1), \mathrm{CS}(k_2), \ldots, \mathrm{CS}\left(k_{|CS|}\right)\right\} \\ &\text{s.t. } P_e\left(u_{\mathrm{CS}(k_i)}\right) \geq P_e\left(u_{\mathrm{CS}(k_{i+1})}\right) \end{aligned} \tag{11}$$

Next, the i_{th} line in $CS-\omega$ is expressed as:

$$\begin{aligned} &CS-\omega(i) = (j_1, j_2, \ldots, j_\omega), 1 \leq i \leq |CS| \\ &j_1 = CS(k_i) \\ &j_2, j_3, \ldots, j_\omega \in \mathcal{A} \\ &j_1 < j_2 < \cdots < j_\omega \end{aligned} \tag{12}$$

Based on the above description of the binary tree structure of polar codes, reference [12] constructs the critical set CS according to formula (10) and the binary tree structure of polar codes, for more detailed information about bit flip and critical set, please refer to reference [8-12].

Neural network BP decoding algorithm

The min-sum approximation function avoids the complicated logarithmic operation, simplifies the calculation process of the algorithm, and uses the early termination criterion to reduce the redundant iterative process and the decoding time of BP algorithm. However, the above improvements still can't change the characteristics that BP algorithm is an iterative decoding algorithm, and can't achieve considerable decoding performance under the limited number of iterations.

In recent years, due to the excellent performance of deep neural network, which has been widely concerned by scholars from all walks of life. The application of neural network in the field of communication has gradually increased, the decoding process can be regarded as the

process of signal classification, so the features contained in the coding structure can be learned by training the neural network. Some scholars have applied DNN to the decoding of linear codes and obtained better performance results. Using the neural network decoder method, since the network training is completed offline, the decoding latency is not involved in the actual calculation. This neural network decoding model has a highly parallel structure, so it is also called a one-shot decoding process, in addition, with the emergence of a large number of open-source neural network frameworks, such as Tensorflow, Pytorch, etc, deep learning can easily carry out high-speed calculation.

In the literature [7], the author first proposed a DNN-based polar codes BP decoding algorithm (DNN-BP), the learnable parameters are set in, so as to make up for the performance loss caused by the minimum sum approximation. The specific iterative rule formula (2) for updating the iterative likelihood ratio information is updated to the following formula:

$$\begin{cases} L_{i,j}^{(t)} = \alpha_{i,j}^{(t)} \cdot g'\left(L_{i+1,2j-1}^{(t)}, L_{i+1,2j}^{(t)} + R_{i,j+N/2}^{(t)}\right) \\ L_{i,j+N/2}^{(t)} = \alpha_{i,j+N/2}^{(t)} \cdot g'\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t)}\right) + L_{i+1,2j}^{(t)} \\ R_{i+1,2j-1}^{(t)} = \beta_{i+1,2j-1}^{(t)} \cdot g'\left(R_{i,j}^{(t)}, L_{i+1,2j}^{(t-1)} + R_{i,j+N/2}^{(t)}\right) \\ R_{i+1,2j}^{(t)} = \beta_{i+1,2j}^{(t)} \cdot g'\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t-1)}\right) + R_{i,j+N/2}^{(t)} \end{cases} \quad (13)$$

Where $\alpha_{i,j}^{(t)}$ and $\beta_{i,j}^{(t)}$ respectively represent the scaling factors of the log-likelihood ratio of left information transmitted from right to left and the log-likelihood ratio of right information transmitted from left to right by nodes (i, j) in the t iteration of polar codes BP decoding.

Although DNN-BP can better solve the approximate loss problem of min-sum, there are still two problems in this deep neural network architecture. First of all, the scaling factor set by this algorithm to reduce the approximate loss brings a lot of multiplication operations to the algorithm, which leads to a significant increase in the computational complexity of BP decoding algorithm based on deep neural network. In addition, the storage space is occupied by the storage of a large number of scaling factors in the deep neural network architecture. Although the BP decoding algorithm based on DNN reduces the total number of iterations before the convergence of the decoding algorithm by scaling the messages with trainable weights, it does not solve the problem of error propagation in the BP decoding algorithm. The incorrect decoding of information bits in the BP decoding message transmission algorithm may lead to error propagation, which will have a negative impact on the reliability and accuracy of many other bits, in order to further solve the problems existing in the BP decoding algorithm of polar codes based on DNN and reduce the BLER of BP decoding algorithm, this paper proposes a BP decoding algorithm of polar code bit-flip based on recurrent neural network.

IV. BP Decoder of Polar Codes Bit-flip Based on Recurrent Neural Network

There is a big difference between deep neural network and recurrent neural network, DNN cannot model changes in time series. In an ordinary fully connected network or CNN, the processing is independent at each moment, while in RNN, the output of neurons can directly act on itself in the next time period, the input of neurons in the i_{th} layer at m time, except the output of neurons in the $(i-1)$ layer at that time, the output of the neuron can directly act on itself in the next time period. Therefore, the RNN-based BP decoding algorithm will reuse the weights in different iterations, thereby realizing parameter sharing among multiple iterations of the neural network, inspired by the recurrent neural network architecture in literature [15], a BP decoding architecture of recurrent neural network based on minimum and approximate scaling offset is proposed. Therefore, the specific iterative rule formula (13) for the previous iteration to update the log-likelihood ratio information is replaced by the following formula:

$$\begin{cases} L_{i,j}^{(t)} = \alpha_{i,j} \cdot g\left(L_{i+1,j}^{(t-1)}, L_{i+1,j+N/2^i}^{(t-1)} + R_{i,j+N/2^i}^{(t)}\right) \\ L_{i,j+N/2^i}^{(t)} = \alpha_{i,j+N/2^i} \cdot g\left(R_{i,j}^{(t)}, L_{i+1,j}^{(t-1)}\right) + L_{i+1,j+N/2}^{(t-1)} \\ R_{i+1,j}^{(t)} = \beta_{i+1,j} \cdot g\left(R_{i,j}^{(t)}, L_{i+1,j+N/2^i}^{(t-1)} + R_{i,j+N/2^i}^{(t)}\right), \\ R_{i+1.i+N/2^i}^{(t)} = \beta_{i+1.i+N/2^i}^{(t)} \cdot g\left(R_{i,j}^{(t)}, L_{i+1,j}^{(t-1)}\right) + R_{i.i+N/2^i}^{(t)} \end{cases} \quad (14)$$

The network architecture of BP decoding

algorithm based on RNN is shown in Figure 5 and Figure 6, in order to reduce the large number of multiplication operations caused by the minimum sum approximation, the Offset Min-Sum (OMS) approximation proposed in the literature [13] is used to improve the BP decoding algorithm, and the $g(x, y)$ function is updated as the following formula:

$$g(x,y) \approx \operatorname{sgn}(x)\operatorname{sgn}(y) \times \max\left(\min(|x|,|y|) - \beta, 0\right) \quad (15)$$

Compared with the scaling factor minimum and approximation algorithm, the minimum and approximation algorithm is different in that a linear rectification function $f_{\mathrm{ReLU}}(x)$ and a learnable offset β are added, which play the roles of nonlinear activation and substitution of multiplication respectively, this function is designed to provide better nonlinear fitting ability for neural networks, at the same time, similar to the scaling factor minimum sum algorithm. In order to improve the system performance, different offsets are applied to different nodes in the factor graph, the specific iterative rule formula (14) for iteratively updating the log-likelihood ratio information is further updated as:

$$\begin{cases} L_{(i,j)}^{(t)} = g\left(L_{i+1,j}^{(t-1)}, L_{i+1,j+\frac{N}{2^i}}^{(t-1)} + R_{i,j+\frac{N}{2^i}}^{(t)}, \beta_{L_{i,j}^t} \right) \\ L_{i,j+\frac{N}{2^i}}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,j}^{(t-1)}, \beta_{L_{i,j+\frac{N}{2^i}}^{(t)}} \right) + L_{i+1,j+\frac{N}{2}}^{(t-1)} \\ R_{i,j}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,j+\frac{N}{2^i}}^{(t-1)} + R_{i,j+\frac{N}{2^i}}^{(t)}, \beta_{R_{i+1,j}^t} \right) \\ R_{i,j+\frac{N}{2^i}}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,j}^{(t-1)}, \beta_{R_{i+1,j+\frac{N}{2^i}}^t} \right) + R_{i,j+\frac{N}{2^i}}^{(t-1)} \end{cases} \quad (16)$$

Where $\beta_{L_{i,j}^t}$ and $\beta_{R_{i,j}^t}$ represent the offset of the left information $L_{(i,j)}^{(t)}$ and the right information $R_{i,j}^{(t)}$ in the i_{th} iteration process, respectively added during calculation. When the offsets are all set to 0, the algorithm is simplified to a common min-sum approximation BP decoding algorithm, by replacing the multiplication operation with simple sign, comparison and subtraction operations, the problem of high computational complexity caused by setting the scaling factor is solved.

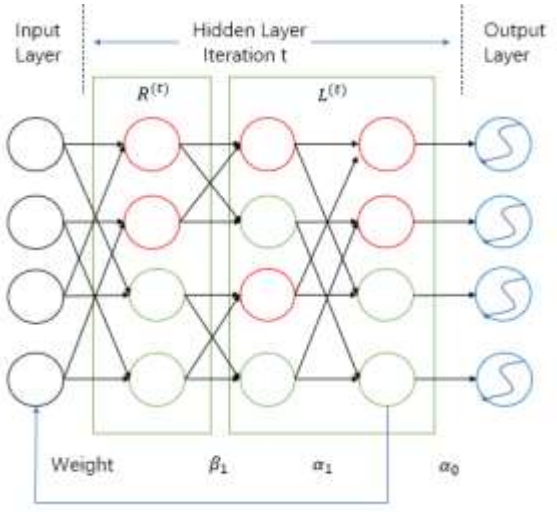

Figure 5. A complete decoding iteration in a recurrent neural network with polar $^{(4,2)}$.

Figure 5 shows a complete iterative decoding process of the $(4,2)$ polar codes in the recurrent neural network, which is expanded according to the polar codes factor graph shown in Figure 2, in the recurrent neural network architecture of Figure 5, $N=4, n=\log_2 N=2$, so the input layer consists of $N=4$ neurons, and the initial value of the channel log-likelihood ratio is input, the transfer process of the right information and the left information are expanded into $(n-1)=1$ layer and $n=2$ layers, which consists of $N=4$ neurons .Therefore, a complete iteration translates to $(2n-1)=3$ hidden layers, the last hidden layer for right-to-left information transfer computes the output of the leftmost node of the original factor graph. At the end of T iterations, in order to make the codeword estimate obtained by the neural network decoder in the range of $[0,1]$, the output layer of the last layer of the network needs to use the sigmoid function to rescale the output to the range of $[0,1]$:

$$o_j = \sigma\left(L_{1,j}^{(T)}\right) = \left(1 + e^{-L_{1,j}^{(T)}}\right)^{-1} \quad (17)$$

After T iterations are completed, calculate the error between the output o and the transmitted bit u:

$$L(\mathbf{u},\mathbf{o}) = -\frac{1}{N}\sum_{i=1}^{N} u_i \log(o_i) + (1-u_i)\log(1-o_i) \quad (18)$$

Compared with other neural network architectures, this recurrent neural network architecture has obvious optimization. Firstly, the linear rectification function is added in the design of the minimum offset and approximation algorithm, which is equivalent to adding the linear rectification function as the activation function in the calculation process of each neuron in the hidden layer, which plays a role in speeding up the training speed and avoiding the gradient from disappearing. Secondly, after the operation of every $(2n-1)$ hidden layer is completed, the n-bit LLR value is circularly input to realize the reuse of offset parameter set β.

After the CS is established, the bit-flipping BP decoding process assisted by the recurrent neural network can be start. If the RNN-OMS-BP decoder fails to decode with cyclic redundancy check (CRC), as shown in Figure 7, a candidate is selected from the critical set according to the descending index corresponding to the error rate in the critical set, that is, u_i with a larger value $P_e(u_i)$ is flipped first, such sorting can be done offline without introducing the latency caused by sorting. After bit flipping, BP decoding based on RNN is performed, if the result meets CRC, the decoding process is complete, otherwise, other candidate nodes in the critical set are tried until the CRC passes successfully. In this paper, critical set bit flipping is adopted to minimize error propagation, the BP decoding algorithm of polar codes bit flipping based on recurrent neural network is summarized in Algorithm 1, this algorithm is a combination of training and decoding, and the algorithm is as follows:

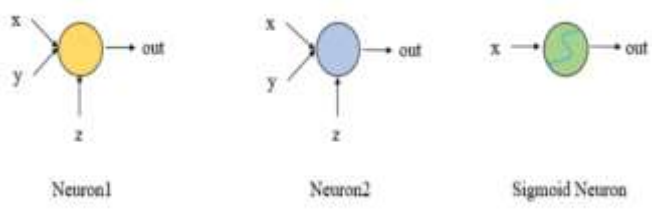

Figure 6. Three types of neurons in network architecture.

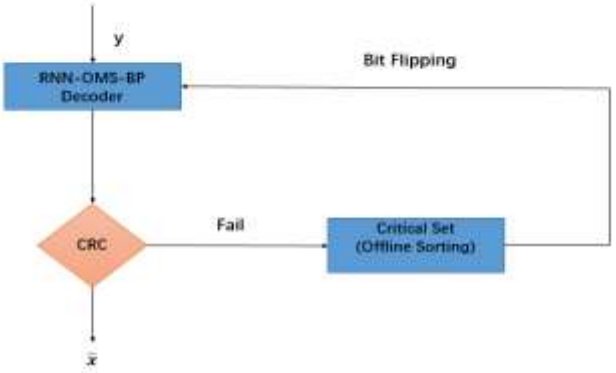

Figure 7. Polar codes bit-flip BP decoder based on RNN.

Algorithm 1 RNN-OMS-BPF algorithm

Input: maximum number of flips $\mathbf{T_{max}}$, critical set **CS**, N, R, SNR, learning rate η, etc.

Output: $\hat{\mathbf{u}}^{\mathbf{N}}$, $\mathbf{o}$, $BLER$, etc.

1: **for** i=0: $\dot{\mathbf{SNR}}$ **do**
2: Generate: $\mathbf{x} = \mathbf{llr}, \mathbf{y} = \mathbf{u}$ // generating a codeword set for training and testing, where x is the generated codeword sequence and y is the transmission bit sequence
3: **end for**
4: **for** itr=1: epoch (100) **do**
5: gendata $(\mathbf{x}, \mathbf{y})$ //get training data
6: tf.train.AdamOptimizer($\eta = 0.001$).minimize(loss)
7: **for** i=0: $\dot{\mathbf{SNR}}$ **do**
8: **for** j=0: $\mathbf{2^{SNR}-1}$ **do**
9: L, R L,R $\leftarrow$Initialie the RNN-OMS-BP decoder using (4) and (5)
10: calculate the $\mathbf{L}_{\mathbf{i,j}}^{(\mathbf{t})}$ and $\mathbf{R}_{\mathbf{i,j}}^{(\mathbf{t})}$ using (16)
11: $\hat{\mathbf{u}}^{\mathbf{N}} \leftarrow$RNN-OMS-BP decoder $(\mathbf{L}, \mathbf{R})$
12: $\mathbf{t} \leftarrow \mathbf{1}$ // the number of bit flip decoding attempts
13: **while** $\hat{\mathbf{u}}^{\mathbf{N}}$ does not pass CRC && $\mathbf{t} \le \mathbf{T_{max}}$ **do**
14: $\mathbf{i} \leftarrow \mathbf{CS(t)}$ //the index in CS
15: $\mathbf{R}(\mathbf{A}^{\mathbf{c}}, \mathbf{1}) \leftarrow +\infty$ // modify R using (8)
16: $\mathbf{R}_{\mathbf{i,j}}^{(\mathbf{l})} = \infty \times \left(\mathbf{2}\hat{\mathbf{u}}_{\mathbf{j}}^{\mathbf{N}} - \mathbf{1}\right), \mathbf{j} \in \mathbf{CS}$
17: $\hat{\mathbf{u}}^{\mathbf{N}} \leftarrow$1 RNN-OMS-BP decoder $(\mathbf{L}, \mathbf{R})$
18: **if** $\hat{\mathbf{u}}^{\mathbf{N}}$ pass CRC **then**

19: **return** $\hat{\mathbf{u}}^{\mathbf{N}}$
20: **else**
21: **if** $t = T_{\max}$ **then**
22: The bit-flip BP decoding fails
23: **end if**
24: **end if**
25: $\mathbf{t} \leftarrow \mathbf{t}+\mathbf{1}$
26: **end while**
27: **end for**
28: $\mathbf{o}, \mathbf{Loss} = \mathbf{sess.run}(\mathbf{x}, \mathbf{y}, \mathbf{optimizer}, \mathbf{loss})$ //run the session output prediction bit sequence and loss function value
29: **end for**
30: **end for**

V. Simulation Results and Analysis

The RNN-OMS-BPF decoder model used in this paper is trained on the deep learning framework Tensorflow 1.8.0 by the adam gradient descent optimization algorithm with a learning rate of 0.001. In order to evaluate the decoding performance of the bit-flip BP decoding algorithm based on the recurrent neural network, a simulation experiment was carried out by the Monte Carlo method, and the all-zero codeword was selected as the training data, the signal-to-noise ranges from 0dB to 3dB, and 3600 codewords are selected as the mini-batch size. The test data is random binary information obtained after polarization coding, binary phase shift keying modulation and transmission on a Gaussian channel, and the weights of the neural network are initialized with random values in a standard normal distribution.

In order to evaluate the BLER performance of the proposed RNN-OMS-BPF algorithm, it is compared with other BP decoding algorithms through simulation, respectively, Table 1 lists the experimental parameters used in the simulation.

TABLE I. Simulation Parameter

Set options	Value
Test platform	Tensorflow1.8.0
Encoding	(64,32)
CRC Generator Polynomial	x^6+x^5+1
Training codeword	7.8×10^4
Testing codeword	3.4×10^3
Batch_size	3600
Loss function	Cross entropy
Optimizer	Adam

A. Performance analysis

With the same code length and code rate, the BLER performance of different algorithms is compared with that of the RNN-OMS-BPF decoding algorithm proposed in this paper, the number of iterations of the RNN-OMS-BPF decoding algorithm is 5, and the simulation results are shown in Figure 8.

When the simulation code length is 64 and the code rate is 0.5, the BLER performance of traditional BP, RNN-BP, CS-BF, CNN-Tree-MBF, CA-SCL and RNN-OMS-BPF decoding algorithm proposed in this paper are compared and analyzed. First of all, we can observe that for $(64,32)$ polar codes, the $BLER$ performance of traditional BP decoding algorithm and RNN-BP decoding algorithm is poor, the RNN-OMS-BPF decoding algorithm proposed in this paper has achieved good decoding performance and the BLER performance is better than that of CA-SCL decoding algorithm with $L=4$. When the number of iterations of BP decoder reaches the maximum and the current decoding result fails the CRC check, the bit flip decoding starts. The bit flip CS-BF algorithm realizes bit flip in BP decoding by setting infinite prior LLR for unreliable information bits in CS set. The simulation results show that, compared with the traditional BP decoding algorithm of polar codes, the bit-flip CS-BF decoding algorithm with bit-flip $\omega=3$ has a BLER performance gain of about 2dB when the BLER is 1×10^{-1}.

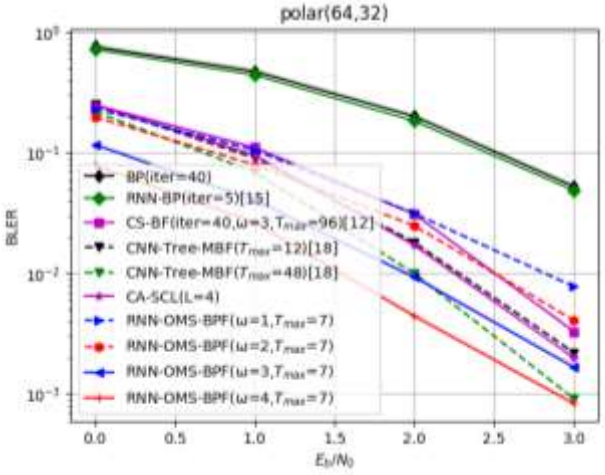

Figure 8. BLER performance of different algorithms.

In order to speed up the convergence rate of the

decoding algorithm, reduce the number of iterations required to achieve the convergence effect, and improve the computational resource consumption and extra memory overhead of the traditional BP decoding algorithm, this paper designs a recurrent neural network architecture on the basis of bit flipping, and combines the critical set to perform bit flipping, and selects a candidate bit from the critical set to flip according to the index in descending order of the corresponding error rate in the critical set. Based on the BP algorithm, the BP decoding algorithm is improved by adopting the Offset Min-Sum OMS. According to the similarity between the polar codes BP decoding factor graph and the neural network structure, the polar codes BP decoding factor graph is expanded on the basis of RNN architecture to form a recurrent neural network decoder, and all zero codewords are generated. The recurrent neural network decoder is trained by using the back propagation and Adam gradient descent optimization algorithm in deep learning technology. Compared with CS-BF decoding algorithm, the RNN-OMS-BPF algorithm proposed in this paper has a performance gain of about 0.50dB when the BLER is 3.407×10^{-3}, however, compared with CS-BF decoding algorithm and CNN-Tree-MBF decoding algorithm, RNN-OMS-BPF decoding algorithm has a smaller maximum number of flips, the RNN-OMS-BPF decoding algorithm proposed in this paper can significantly improve the error correction ability with fewer flips.

B. *Computational complexity analysis*

The computational complexity of several decoding algorithms recorded in Table 2, taking $(64,32)$ polar codes as an example, the computational complexity of multiplication of various algorithms in the process of iterative updating of core information is listed when the SNR is 0, the number of iterations of RNN-BP decoding algorithm, RNN-OMS-BPF decoding algorithm are 5 times, and CS-BF decoding algorithm is 40 times, the RNN-OMS-BPF decoding algorithm proposed in this paper replaces all multiplication operations at the cost of increasing some addition operations.

TABLE II. COMPUTATIONAL COMPLEXITY ANALYSIS

Decoding algorithm	Multiplication
RNN-BP	$2IN\log_2 N = 3840$
CS-BF	$2IN\log_2 NT_{avg} = 144080$
CNN-Tree-MBF	$2IN\log_2 NT_{avg} + 9.8M\times T_{CNN}$
CA-SCL	$L\left(N\log_2 N+\frac{3N}{2}+\frac{5K}{2}\right)-\frac{1}{2}K = 4464$
Proposed	0

C. *Memory overhead analysis*

The memory overhead analysis in Table 3 takes the $(64,32)$ polar codes as an example, the memory overhead of various algorithms is listed.

TABLE III. MEMORY OVERHEAD ANALYSIS

Decoding algorithm	Memory overhead
RNN-BP	$N(\log_2 N+1) = 448$
CS-BF	$N(\log_2 N+1)+\omega T_{max} = 736$
CNN-Tree-MBF	$2IN(\log_2 N+1)+T_{max}+K+0.3M \approx 0.3M$
CA-SCL	$N+3NL-L = 1592$
Proposed	$N(\log_2 N+1)+\omega T_{max} = 469$

Table 3 uses the number of parameters to evaluate the memory overhead, where the flip bit ω is 3 and the SNR is 0dB. The RNN-BP decoding algorithm stores L, R messages for computing updated messages, approximately $N(\log_2 N+1)$ messages, while the CS-BF decoding algorithm requires additional memory to store the flipped table, approximately ωT_{max} However, CNN-tree-MBF decoding algorithm needs to store L and R messages of each iteration as input data, which is $2I$ times of

RNN-BP, in addition, additional memory is needed for prediction results, decoding bit values required parameters. The RNN-OMS-BPF decoding algorithm proposed in this paper significantly improves the error correction ability and effectively reduces the memory overhead under the condition of fewer rollover times.

VI. Conclusions

In this paper, a BP decoding algorithm of polar codes bit flipping based on recurrent neural network is proposed, by selecting bits from the critical set of bit flipping constructed by most error-prone bits, the error correction ability can be significantly improved with less flipping times, and the BLER of BP decoding algorithm can be effectively reduced. Moreover, the multiplication operation is replaced by the Offset Min-Sum approximation algorithm, and the improved recurrent neural network architecture is adopted to realize parameter sharing, which effectively reduces the memory overhead and improves the computational resource consumption of traditional BP decoding algorithm. Next, the stop set will be combined to reduce unnecessary flip bits to reduce decoding delay, so as to obtain better performance gain.

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.

[2] Liu C, Dai W, Guo R. Syndrome Check Aided Fast-SSCANL Decoding Algorithm for Polar Codes [J]. KSII Transactions on Internet & Information Systems, 2024, 18(5). DOI: 10.3837/tiis.2024.05.014

to determine the flip direction and storage of CNN

[3] Liu H, Gunawan E, Yaoyue H, et al. BP-Based Sparse Graph List Decoding of Polar Codes [J]. IEEE communications letters: A publication of the IEEE Communications Society, 2023(5):27.

[4] Tal I, Vardy A. List Decoding of Polar Codes[C]// IEEE. IEEE, 2011.

[5] Ren, Y. Shen, Y. Zhang, L. Kristensen, A.T.; Balatsoukas-Stimming, A. Boutillon, E. Burg, A. Zhang, C. High-Throughput and Flexible Belief Propagation List Decoder for Polar Codes [J]. IEEE Trans. Signal Process. 2024, 72, 1158–1174.

[6] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, on deep learning based channel decoding, in The 51st Annual Conference on Information Sciences and Systems (CISS). IEEE, pp. 16, 2017.

[7] Lyu W, Zhang Z, Jiao C, et al. Performance Evaluation of Channel Decoding with Deep Neural Networks[C]// 2018 IEEE International Conference on Communications (ICC). IEEE, 2018.

[8] Teng C F, Chen-Hsi, Ho K S, et al. Low-complexity Recurrent Neural Network-based Polar Decoder with Weight Quantization Mechanism. 2018.

[9] Zhang W, Wu X. Low-Latency SCL Bit-Flipping Decoding of Polar Codes [J]. ArXiv, 2023, abs/2306.02629.DOI:10.48550/arXiv.2306.02629.

[10] Teng C F, Ho K S, CH Wu, et al. Convolutional Neural Network-aided Bit-flipping for Belief Propagation Decoding of Polar Codes. 2019.

[11] Xu W, Tan X, Be'Ery Y, et al. Xu W, Tan X, Be'Ery Y, et al. Deep Learning-Aided Belief Propagation Decoder for Polar Codes[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2020, 10(2): 189-203.

[12] Yinyou M, Dong Y, Xingcheng L, et al. Improved Segmented Belief Propagation List Decoding for Polar Codes with Bit-Flipping [J]. China communications, 2024, 21(3):19-36.

[13] Zhouyu, Yue L I, Shuangshuang Z, et al. Polar codes based OFDM-PLC systems in the presence of impulsive noise [J]. High Technology Letters, 2024, 30(2):188-198.

[14] Chen C H, Teng C F, Wu A Y. Low-Complexity LSTM-Assisted Bit-Flipping Algorithm for Successive Cancellation List Polar Decoder [C]// ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

[15] Zhang J, Wang M. Belief Propagation Decoder with Multiple Bit-Flipping Sets and Stopping Criteria for Polar Codes [J]. IEEE Access, 2020, PP (99):1-1.