LOW-COMPLEXITY LSTM-ASSISTED BIT-FLIPPING ALGORITHM FOR SUCCESSIVE CANCELLATION LIST POLAR DECODER

Chun-Hsiang Chen, Chieh-Fang Teng, and An-Yeu (Andy) Wu
Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, 106, Taiwan.
{johnny, jeff}@access.ee.ntu.edu.tw, andywu@ntu.edu.tw

ABSTRACT

Polar codes have attracted much attention in the past decade due to their capacity-achieving performance. The higher decoding capacity is required for 5G and beyond 5G (B5G). Although the cyclic redundancy check (CRC)-assisted successive cancellation list bit-flipping (CA-SCLF) decoders have been developed to obtain a better performance, the solution to error bit correction (bit-flipping) problem is still imperfect and hard to design. In this work, we leverage expert knowledge in communication systems and adopt deep learning (DL) techniques to obtain a better solution. A low-complexity long short-term memory network (LSTM)-assisted CA-SCLF decoder is proposed to further improve the performance of conventional CA-SCLF and avoid complexity and memory overhead. Our test results show that we can effectively improve the BLER performance by 0.11dB compared to prior work and reduce the complexity and memory overhead by over 30% of the network.

Index Terms—Polar codes, successive cancellation list, bit flipping, long short-term memory network

1. INTRODUCTION

Polar codes, since proposed by Arikan in 2009, have attracted lots of attention from both academia and industry due to their capacity-achieving performance [1]. Further, in 2016, polar codes have been adopted as the official channel coding scheme for enhanced mobile broadband (eMBB) uplink and downlink control channels [2].

Recently, as deep learning (DL) has many revolutionary breakthroughs in the field of computer vision and natural language processing, many researchers have also been dedicated to applying this powerful technique to enhance decoding algorithms [3]-[7]. However, their decoding capacity is still worse than the state-of-the-art CRC-assisted successive cancellation list (CA-SCL) [8]-[10]. Although CA-SCL benefits from the list structure and CRC, it still suffers from large storage overhead and computational complexity. As a result, bit-flipping decoding was adopted in CA-SCLF to release hardware complexity with the flexible tradeoff between the decoding performance and latency as shown in Fig. 1(a) [11]-[13].

To further improve decoding capacity and reduce the decoding latency, the flipping trials of error bits in CA-SCLF, we take advantage of DL to enhance the error path prediction accuracy. Besides, to avoid the severely increased memory overhead and additional computations from the DL model, we apply our domain knowledge in communication systems with carefully designed DL architecture. The proposed long short-term memory network (LSTM)-assisted CA-SCLF is shown in Fig. 1(b). The main contributions of our work are as follows:

1) We apply LSTM to the CA-SCLF decoder, which can effectively extract the features from the sequential decoding process and enhance the error bit prediction. Therefore, the proposed design can improve the block error rate (BLER) performance and decoding latency by 0.11dB and 41.25%, respectively.

2) By taking advantage of domain knowledge for input data preprocessing and output dimension reduction, we can significantly reduce the memory requirements and computational complexity by over 30%, making our design more feasible for hardware implementation.

The rest of this paper is organized as follows. Section 2 briefly reviews the concept of polar codes, CA-SCLF, and the prior work. Section 3 illustrates the proposed architecture of LSTM-assisted CA-SCLF with novel input data preprocessing and output design. The numerical experiments and analyses are shown in Section 4. Finally, Section 5 concludes our work.
2. PRELIMINARIES

2.1. Polar Codes
In this paper, we use \( u_j^i \), with \( j < i \), to denote the vector \( \{ u_j, u_{j+1}, \ldots, u_i \} \). To construct a \((N,K)\) polar code, the mathematical foundations lay on the polarization effect of the \( 2 \times 2 \) transformation matrix \( F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \). \( N \) virtual bit-channels are constructed from recursive polarizing transformation by \( \log_2 N \) times. As \( N \) approaches infinity, the channels separate into the noisy channels (unreliable) and noiseless channels (reliable). According to the reliabilities of each channel, the \( K \) information bits are assigned to the \( K \) most reliable channels in the \( N \)-bit message \( u_1^N \) and the remaining \((N-K)\) bits are taken as frozen bits with the assignment of zeros. The set of information bit indices is denoted by \( A \). Then, the \( N \)-bit transmitted codeword \( x_1^N \) is calculated by multiplying \( u_1^N \) with generator matrix \( G_N \) as follows:

\[
x_1^N = u_1^N G_N = u_1^N F^\otimes n B_N, \quad n = \log_2 N,
\]

where \( F^\otimes n \) is the \( n \)-th Kronecker power of \( F \), and \( B_N \) represents the bit-reversal permutation matrix.

In past years, there are several decoding algorithms developed to meet various scenarios, such as belief propagation (BP), CRC-assisted successive cancellation bit flip (CA-SCF) [11], and CA-SCL. BP iteratively decodes the message over the corresponding factor graphs. On the other hand, both CA-SCF and CA-SCL decode the message based on SC but with different mechanisms to correct the error decoding bits.

2.2. Prior Work: CA-SCLF Decoding [12]-[13]
CA-SCL decoder is recently developed due to its high performance and flexibility. The decoding process of CA-SCLF shown in Fig. 2, is described as the following. Firstly, given the received signal \( y \), the CA-SCL algorithm with determined list size \( L \) is implemented. The concept of CA-SCL is to reserve the \( L \) most reliable paths (red lines) from \( 2L \) temporary paths through the path metric (PM) values \( PM_i \), where \( l \) and \( i \) denote the index of \( 2L \) temporary paths and the bit index. The path metric is defined as

\[
PM_i = \sum_{j=1}^{L} \ln(1 + e^{-(l-2\bar{y}_j[l]+L)})^{LL_i(l)}(l),
\]

where \( \bar{y}_j[l] \) and \( LL_i[l] \) denote the \( j \)-th decoding bit at \( l \)-th path and the decoding LLR of bit \( \bar{y}_j[l] \) given \( y \) and decoding trajectory \( \hat{u}_1^{j-1}[l] \), respectively. Without loss of generality, we assume that the \( 2L \) temporary paths satisfy \( PM_1 < PM_2 < \ldots < PM_{2L} \). Besides, \( l' \) and \( l' + L \) stand for the reserved paths (first \( L \) paths) and discarded paths (last \( L \) paths), respectively.

Secondly, CA-SCLF attempts to correct the first error path selection, where the correct path (orange arrow line) is not among the reserved paths (red lines). In Fig. 2., the first error happens at \( \hat{u}_6 \). As a result, there will be a set for bit flipping (BFS). In [13], the BFS is based on the self-defined error metric \( M(i) \):

\[
M(i) = \sum_{l=1}^{L} \prod_{j=1, j \neq i}^{J} (1 + e^{-(l-2\bar{y}_j[l]+L)})^{LL_i(l)}(l) \}
\]

Next, we retry CA-SCL from the discarded paths (blue lines) by following the bit-flipping order. We should note that there is only one bit flipped for every trial. The decoding continues until the CRC passes or the trial number has reached the maximum number \( T_{max} \).

We can observe that the error metric from Eq. (3) only contains the information from discarded paths. It results in information loss from the reserved paths, which will degrade the performance. In our work, we look into the information from both the reserved paths and discarded paths. Also, we are the first to utilize LSTM for CA-SCLF as far as we know.

3. PROPOSED LOW COMPLEXITY LSTM ASSISTED CA-SCLF POLAR DECODER

3.1. Proposed Feature Extraction for Neural Network
From the original implementation of CA-SCL in the first step of CA-SCLF decoding, we can receive the LLRs and PM values for each decoding bit. However, if we set the whole LLRs or PM values as the neural network inputs, the cost of memory and complexity will be unaffordable. As we know, the path metric represents the probability of each decoding path given \( y \). As a result, we propose the novel error metric based on PM. The error metric can represent the error path selection probability. The formula takes the ratio of the discarded paths and reserved paths without any information loss and is written as

\[
\hat{M}(i) = \frac{\sum_{l=l' + L}^{\infty} e^{-(PM_i)},}{\sum_{l=1}^{\infty} e^{-(PM_i)}}.
\]
Algorithm 1: Proposed LSTM-Assisted CA-SCLF Decoding

- **Input:** $y, L, T_{max}, A$
- **Output:** $\hat{u}_t^n$
- **Computation**
  
  for $i = 1, 2, ..., N$ do
  
  Do the conventional CA-SCL
  
  $\{\hat{u}_t^n, PM_1^n, PM_2^n, ..., PM_L^n\} \leftarrow CA-SCL(\emptyset)$
  
  Calculate $\hat{M}(i)$ by Eq. (8)
  
  end
  
  if $CRC(\hat{u}_t^n) = $ success then
  
  Return $\hat{u}_t^n$ and break
  
  end

outputs, the fully connected network with softmax as its activation function is connected with the LSTM. We take the concept of critical set (CS) for the outputs in our work. The CS consists of the bits which can be taken as the first bit in every sub-block of polar codes with code rate equal to 1. These bits are proven to have higher error rate [11]. As a result, we propose to use a one-hot vector with the length $S$ whose first $S - 1$ bits represent the information bit indices in CS, excluding the first log$_2(L)$ information bit indices. The last bit represents the other information bit indices not in the critical set (NCS) as shown in Fig. 3(b). Relying on the specific design, the complexity and memory can be reduced. The loss function is cross-entropy between neural network output $\hat{o}$ and true label $o$, which is written as

$$L(\hat{o}, o) = \frac{-1}{S} \sum_{i=1}^{S} \hat{o}_i \log(o_i) + (1 - \hat{o}_i) \log(1 - o_i).$$

The decoding procedure is shown in Algorithm 1.

4. SIMULATION RESULTS

The simulation setup is summarized in Table I.

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encoding</strong></td>
<td>Polar code (512,256), CRC = 24, $S = 62$</td>
</tr>
<tr>
<td><strong>Eb/N0 (dB)</strong></td>
<td>0, 0.5, 1, 1.5, 2</td>
</tr>
<tr>
<td><strong>Training/Validation data size</strong></td>
<td>500,000/100,000</td>
</tr>
<tr>
<td><strong>Mini-batch size</strong></td>
<td>200</td>
</tr>
<tr>
<td><strong>Optimizer</strong></td>
<td>Adam</td>
</tr>
<tr>
<td><strong>Simulation Environment</strong></td>
<td>DL library of MATLAB with NVIDIA RTX 2080 GPU</td>
</tr>
</tbody>
</table>
4.1. Performance of Proposed LSTM Design for CA-SCL

In the first experiment, we examine the block error rate (BLER) performance of the proposed LSTM design with preprocessing (WP) named as WP-LSTM. Similar to the output design of WP-LSTM with the path metrics taken as inputs, we also simulate for the performance of LSTM without preprocessing (NP) named as NP-LSTM. The proposed design is compared with the prior work [13], and the simulation results are shown in Fig. 4. Except for CA-SCL, all the others are implemented with \( L = 8 \), \( T \) and \( H \) denote the \( T_{\text{max}} \) and hidden layer size in the figure, respectively.

Firstly, we can observe that WP-LSTM with \( T_{\text{max}} = 10 \) can outperform the CA-SCL decoder with \( L = 16 \) by 0.06 dB and 0.13 dB with \( T_{\text{max}} = 20 \). Also, WP-LSTM has 0.11 dB better performance than [13] when \( T_{\text{max}} = 20 \). Secondly, we compare WP-LSTM with NP-LSTM with the same hidden layer size 500 and \( T_{\text{max}} = 20 \). The results show that WP-LSTM can achieve 0.018 dB better performance, which tells the truth that the domain knowledge cannot only reduce the complexity and memory overhead but also help to improve the performance under limited resources. Furthermore, we reduce the hidden layer size of WP-LSTM to 100. We can observe that the performance just slightly degrades 0.024 dB, which means the design is robust.

In the second experiment, we examine the average bit-flipping attempts (latency). The simulation results are shown in Fig. 5. We can observe that the average bit-flipping attempts of the proposed design are relatively small. Compared with the conventional design [13], the proposed design can reduce 41.25% of average extra attempts with \( T_{\text{max}} = 20 \) at EbN0 = 1.5 dB.

From the simulations above, we can know that the proposed design cannot only enhance the BLER performance but also reduce the latency.

4.2. Complexity Analysis of Neural Network Design

In Table II, we compare the hardware complexity of the proposed design and the LSTM design without domain knowledge, whose inputs are the LLRs of N-bit message in \( L \) paths and the outputs are the whole information set. The design rule of the latter design is similar to [7].

In Table II, the list size \( L \) and hidden layer size are set to 8 and 100, respectively. We can see that by taking advantage of domain knowledge for input data preprocessing and output dimension reduction, the proposed LSTM design can effectively reduce about 32% of complexity and memory usage.

TABLE II. COMPLEXITY ANALYSIS OF DIFFERENT NETWORK DESIGN

<table>
<thead>
<tr>
<th></th>
<th>LSTM network w/o domain knowledge</th>
<th>Proposed LSTM network w/ domain knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>( \sim H(4H + 4L + K + 5) ) 69300 (100%)</td>
<td>( \sim H(4H + S + 9) ) 47100 (67.9%)</td>
</tr>
<tr>
<td>Multiplication</td>
<td>( \sim H(4H + 4L + K + 3) ) 69100 (100%)</td>
<td>( \sim H(4H + S + 7) ) 46900 (67.8%)</td>
</tr>
<tr>
<td>Memory</td>
<td>( \sim H(4H + 4L + K + 6) ) 69400 (100%)</td>
<td>( \sim H(4H + S + 10) ) 47200 (68.0%)</td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this paper, we present the novel LSTM-assisted CA-SCL decoder. The bit-flipping order can be learned from the LSTM network, and it can improve the performance and the latency compared to the conventional algorithms [13]. Also, we utilize the domain knowledge to reduce the complexity and memory requirements so that the hardware can be friendly. In the future, we will research into the design for supporting multiple code rates.
6. REFERENCE


