Wireless Personal Communications (2022) 122:737–753
https://doi.org/10.1007/s11277-021-08923-0

Deep Learning Based Decoding for Polar Codes in Markov Gaussian Memory Impulse Noise Channels

Shu-Ming Tseng[1] · **Wei-Cheng Hsu**[2] · **Der-Feng Tseng**[3]

Accepted: 8 August 2021 / Published online: 24 August 2021

Abstract
In previous papers, decoding schemes which did not use machine learning considered additive white Gaussian noise or memoryless impulse noise. The decoding methods applying deep learning to reduce computational complexity and decoding latency didn't consider the impulse noise. Here, we apply the Long Short-Term Memory (LSTM) neural network (NN) decoder for Polar codes under the Markov Gaussian memory impulse noise channel, and compare its bit error rate with the existing Polar code decoders like Successive Cancellation (SC), Belief Propagation (BP) and Successive Cancellation List (SCL). In the simulation results, we first find the optimal training SNR value 4.5 dB in the Markov Gaussian memory impulse noise channel for training the proposed LSTM based Polar code decoder. The optimal training SNR value is different from that 1.5 dB in the AWGN channel. The bit error rate of the propose LSTM based Polar code decoder is one third that of the previous non-deep-learning-based decoder SC/BP/SCL in Markov Gaussian memory impulse noise channels. The execution time of the proposed LSTM-based method is 5 ~ 12 times less and thus has much less decoding latency than that of SC/BP/SCL methods because the proposed LSTM-based method has inherent parallel structure and has one shot operation.

Keywords Markov Gaussian channel · Memory impulse noise · Polar code · Long short-term memory

1 Introduction

Polar code was proposed in [1]. It, concatenated with Cyclic Redundancy Check (CRC), is adopted as the channel code of the 5G control channel [2]. Its decoding methods include Successive Cancellation (SC) [1], Belief Propagation (BP) [3] and Successive Cancellation list (SCL) [4]. SC decoding is simpler but it does not have parallel structure. BP decoding is an iterative message passing procedure and has parallel structure

✉ Shu-Ming Tseng
shuming@ntut.edu.tw

[1] Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan

[2] Pegatron Corporation, Taipei, Taiwan

[3] Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

and doesn't need many iterations. In addition to the additive white Gaussian noise (AWGN) considered in most Polar code decoding papers, the non-Gaussian noise such as the impulse noise has a more serious impact on communication systems.

The impulse noise exists in wired and wireless communications channels by experimental measurements [5–7] such as urban outdoor/indoor mobile [8, 9], power-line communications [10, 11] because of man-made electromagnetic interference etc.

The impulse noise is different from the AWGN. The energy of impulse noise is often tens of times that of the AWGN. Impulse noise models can be divided into memory noise channels like Markov-Gaussian channel models [10, 12–15], and memoryless noise channels like Bernoulli-Gaussian (BG) [16–18] and Additive White (Middleton) Class A Noise (AWAN) [19–22]. The works on the Polar code decoding in the impulse noise channels are [19, 23, 24]. Tseng et al. [19, 23] used SC to decode the Polar code in the memoryless impulse noise channels with different cases like known statistical features, position or not [24]. Compared the SC and BP on Polar code with memoryless impulse noise with different situation like [23].

Deep learning is popular in many areas like image classification [25], speech recognition [26], and radio resource allocation [27, 28]. Machine learning needs separate feature extraction. Deep learning has embedded feature extraction and uses deep nueural network (DNN) composed of multiple layers of nonlinear processing units. The DNN learns data representation with multiple abstraction levels. Deep learning can learn complex structures in training sets to adjust the neuron weights [29]. There are special cases of DNN. The convolutional neural network (CNN) realizes spatial correlations [30, 31]. The recurrent neural network (RNN) [32] realizes the temporal correlations and is suited to sequential data [33]. To solve the gradient explosion problem of RNN, a Long Short-Term Memory (LSTM) model [34] that can be remembered for a long time emerges from RNN.

Recent years, many researchers try to use deep learning for channel coding [35–41]. Kim et al. [35] used RNN and Neural Recursive Systematic Convolutional (N-RSC) to train convolutional codes and Turbo codes on the AWGN channel and test it on the non-AWGN channel which show better performance than convolutional Turbo decoder. Liang et al. [36] proposed LDPC decoder using BP-CNN, CNN extracts the feature of the color noise (correlated noise is similar to spatial correlation of an image) and removes the BP decoding error in a way similar to image denoising. It showed BP-CNN is better than the BP decoder under the general color noise environment. Cammerer et al. [37] partitioned the 128 bit long Polar code encoding graph in to blocks and trained separately. These neural network decoders are connected to BP decoding. Gruber et al. [38] used deep neural network for decoding Polar and random code with 16–64 bit codeword length and showed the performance of Polar code is better than random code. Irawan et al. [39] extended [38] to fading channels for (16, 8) Polar code. Lyu et al. [40] compared DNN, CNN, and RNN for Polar code decoding. Gross et al. [41] proposed to partition Polar encoding graph too but to connect to SC decoding instead. It showed the same performance and 43.5% reduction in latency than [37] for a (128, 64) Polar code. The above deep learning Polar cod decoding papers, however, did not consider the impulse noise.

The main objective or research problem that this paper want to address is deep learning based Polar code decoding in the memeory impulse noise channels. We considered short Polar code (16,8), the same as the prior works [38, 39] for comparison. For longer code length, we could add more layers in DNN/CNN, or stack more LSTM cells in every time step [40]. The impulse channels exist in realistic environment [5–11] but it requires more complicated statistics for Polar code decoding like known statistical

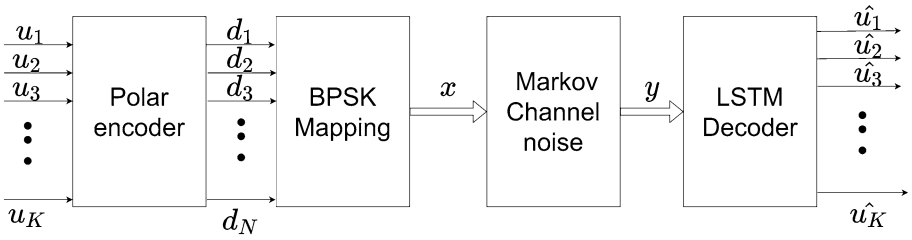

Fig. 1 Polar code decoder system model

features (average number of impulses in a code word), the positions of the impulses in a codeword, the state transition probability to and from the states with the impulse noise or not, etc. [15, 19, 23], so the other existing works do not consider the impulse noise channel.

The reasons why deep learning or LSTM is selected for Polar code decoding are as follows. First, non-deep learning-based Polar code decoding required iterative operations and has no parallel structure, so it has high computation complexity and thus high decoding latency [38]. The deep learnning for Polar code decoding has inherently parallel structure and is a one-shot operation [38]. That is, there is no iterative steps and thus reduce the complexity by 30% or more [42, 43]. Second, Polar decoding is a sequential decoding problem [43]. That is, there is long time dependency within the codeword [44] and LSTM is built for exploiting the temporal correlation. Finally, the memory impulse noise we considered in this paper has temporal memory, this is one more reason to select LSTM which is naturally fit to deal with temporal correlation for Polar code decoding.

In this paper, we propose LSTM (deep learning) based decoder for Polar code in the memory impulse noise channel. The contribution is as follows:

1. Previous papers [19, 23, 24] which did not use deep learning only considered the memoryless impulse noise and AWGN.
2. Previous papers [38–44] which decoded using deep learning didn't consider the impulse noise.
3. We find the optimal training SNR value 4.5 dB in the memory impulse channel, which is different from that 1.5 dB in the AWGN channel in [38, 40]. Based on the optimal training SNR value we found, the bit error probabilityt of the proposed LSTM-based decoding method is one third of that of the previous non-deep-learning based schemes for the testing SNR range $0 \sim 6$ dB.
4. The proposed LSTM-based method is $5 \sim 12$ times less and thus has much less decoding latency than that of SC/BP/SCL methods because the proposed LSTM-based method has inherently parallel structure and has one shot operation- no iterative operations.

2 System Model

The Polar code encoder and decoder block diagram is shown in Fig. 1. The decoder is replaced by LSTM based Polar coder decoder.

2.1 Markov-Gaussian (MG) Impulse Noise Channel Model

The MG noise model is shown in Fig. 2. The MG channel is a hybrid two state Markov chain and generated by a Gaussian process, while state1 (j = 1) represents the Gaussian noise only, state2 (j = 2) represents the impulse noise. MG channel model describes the burst channel, we assume that the received signal:

$$y = x + \omega \tag{1}$$

where x is the transmitted signal with bit energy E_b, ω is the noise, and it has two states, $state_1$ is the AWGN, and $state_2$ is the impulse noise. Then, the probability density functions (PDF) of ω is:

$$P(\omega|state_1) = \frac{1}{\sqrt{2\pi\sigma_G^2}} exp\left\{\frac{|\omega|^2}{2\sigma_G^2}\right\} \tag{2}$$

$$P(\omega|state_2) = \frac{1}{\sqrt{2\pi R\sigma_G^2}} exp\left\{\frac{|\omega|^2}{2R\sigma_G^2}\right\} \tag{3}$$

where σ_G^2 is the variance of the Gaussian noise, and R is the impulse noise power over the Gaussian noise power. The SNR is defined as $\frac{E_b}{\sigma_G^2}$.

The channel state transition probabilities are expressed as:

$$\begin{aligned} &P_{state^*|state} = P(state^*|state) \\ &state, state^* \in \{state_1, state_2\} \end{aligned} \tag{4}$$

$state^*$ is the next state.

The state transition probability matrix of the MG impulse noise channel is as follows:

$$\mathrm{M} = \begin{bmatrix} P_{state1|state1} & P_{state2|state1} \\ P_{state1|state2} & P_{state2|state2} \end{bmatrix} \tag{5}$$

The matrix element must satisfy:

$$P_{state1|state1} + P_{state2|state1} = 1 \tag{6}$$

$$P_{state1|state2} + P_{state2|state2} = 1 \tag{7}$$

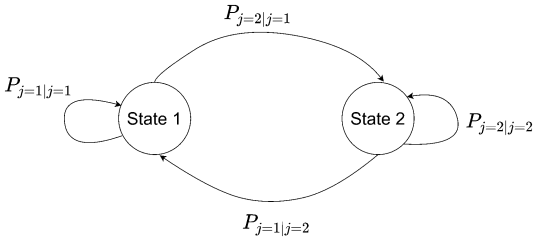

Fig. 2 State diagram of Markov-Gaussian noise model

and P_{state1}, P_{state2} can be expressed as:

$$P_{state_1} = \frac{P_{state1|state2}}{P_{state2|state1} + P_{state1|state2}} \tag{8}$$

$$P_{state_2} = \frac{P_{state2|state1}}{P_{state2|state1} + P_{state1|state2}} \tag{9}$$

According to [12, 15, 16], we set the matrix M as:

$$\mathrm{M} = \begin{bmatrix} 0.99 & 0.01 \\ 0.2 & 0.8 \end{bmatrix} \tag{10}$$

2.2 Polar Code

The Binary Discrete Memoryless Channel (B-DMC) is assumed channel splitter and channel combining to make channel polarization [1]

2.2.1 Channel Polarization

Channel polarization is the repeated use of any B-DMC in a recursive manner, and through channel splitting and channel combination. The original independent B-DMC is turned into a polarized channel with a certain relationship, and so many the sub -channels will have different reliabilities. When the encoding length N is larger, the more polar sub-channels, the channel capacity of some polar sub-channels will approach 1 and become the perfect channel. The rest sub-channels are the opposite. Their channel capacity will approach zero, making it a poor channel for pure noise. The transmission of the polarization code is to use the channel whose channel capacity is close to 1 to transmit the message, and the channel closer to 0 will transmit the frozen bits.

2.2.2 Encoder

The encoder is to form a butterfly architecture for channel merging and select a sub-channel with better channel capacity for message transmission. The polarization code encoding method with length N which is $\boldsymbol{u}$ through the generator matrix form G_N and then through each channel:

$$\mathrm{d} = \mathrm{u}G_N \tag{11}$$

G_N is the matrix with code length N, G_N is generated using the matrix $\mathrm{F} = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ via Kronecker power:

$$G_N = F^{\otimes n} \tag{12}$$

$F^{\otimes n}$ Kronecker power is define as:

$$A_{m*n} \otimes \mathrm{B} = \begin{bmatrix} A_{11}B \ldots A_{1n}B \\ \vdots \ddots \vdots \\ A_{m1}B \ldots A_{mn}B \end{bmatrix} \tag{13}$$

Only the sub-channel with the better channel capacity is selected for transmission, and the remaining sub-channels transmit frozen bits known in advance by the encoding and decoding end, so (11) can be rewritten in more details:

$$\begin{aligned} \mathrm{d} &= u_A G_N(\mathrm{A}) + u_{A^c} G_N(A^c) \\ \mathrm{u} &= \left(u_A, u_{A^c}\right) \end{aligned} \tag{14}$$

A is the set of original signals, A^c is set of frozen bits (=zeros).

2.2.3 Decoder

Assume a parameter (N, K, A, A^c), The encoder makes u_i into d_i and transmits it through the channel C. The decoder uses the two parameters known in advance with the encoder, The first one is the transmits message A, the second is the frozen bit A^c, and estimate u_i with the received signal y_i, then get the predicted result $\widehat{u}_i$. The decoder has already known the location and information of the frozen bit, so the $\widehat{u}_{A^c}$ must be equal to u_{A^c}, we only need to estimate the remaining information to get $\widehat{u}_A$.

Three common decoding methods will be used in this work to compare: Successive Cancellation (SC), Belief Propagation (BP), Successive Cancellation List (SCL).

Due to the limitation of space, we only describe SC in details. Please refer to references [3, 4] for details of BP and SCL.

SC decoding method was proposed in [1]. It can be known from the encoding principle that the construction of the Polar code is the choice of the channel, and this selection method is selected according to the optimal SC performance as the standard. The channel has the channel reliability, SC decoding is to judge the log likelihood ratio (LLR) of each bit from small to large, each step of decoding will need to use the previous results, so under the condition of correct decoding, the channel capacity will be reached, and the longer the Polar code, the easier the channel capacity is reached.

$$\widehat{u}_i \triangleq \begin{cases} u_i, i \in A^C \\ h_i\left(y_1^N, \hat{u}_1^{i-1}\right), i \in A \end{cases} \tag{15}$$

The estimated value $\widehat{u}_i$ during decoding will be determined according to (15).y_1^N is the first bit of receive signal with length N. As mentioned before, when it is a frozen bit, u_i must be equal to $\widehat{u}_i$. Where h_i is the decision equation decoded for this:

$$h_i\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \begin{cases} 0, L_{1,i} \geq 0 \\ 1, otherwise \end{cases} \tag{16}$$

$L_{1,i}$ is the LLR of $\widehat{u}_i$:

$$LLR(y) = ln \frac{W(\mathrm{y}|\mathrm{u}=0)}{W(\mathrm{y}|\mathrm{u}=1)} \tag{17}$$

W is the set of channel after channel polarization, $W(\mathrm{y}|\mathrm{u}=0)$ is the channel transmission probability.

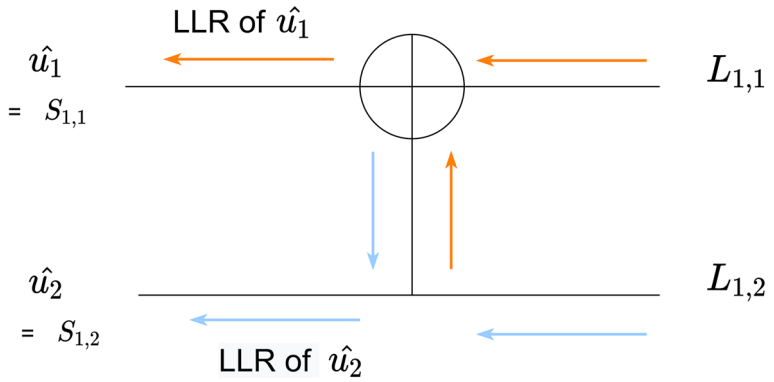

Fig. 3 Unit structure SC decoding

The SC decoding recursion can be expressed as (18):

$$L_{j,i} = \begin{cases} 2tanh^{-1}\left[\tanh\left(\frac{L_{j+1,i}}{2}\right)\tanh\left(\frac{L_{j+1,i+2^{j-1}}}{2}\right)\right], \frac{i-1}{2^{j-1}} mod 2 = 0 (18a) \\ \left(1-2\hat{u}_{j,i-2^{j-1}}\right)\left(L_{j+1,i-2^{j-1}}\right) + L_{j+1,i}, otherwise (18b) \end{cases} \tag{18}$$

L is LLR, $L_{j,i}$'s j and i respectively represent the jth recursion in the coding structure and the ith bit in this structure. $1 \le j \le n, 1 \le i \le N$.

We define two functions:

$$f(x,\ y) = \text{sign}(x)\text{sign}(y)\min(|x|, |y|) \tag{19}$$

$$g(x,\ y,\ z) = (-1)^z * x + y \tag{20}$$

and (18) can be rewritten as:

$$L_{j,i} \approx \begin{cases} f\left(L_{j+1,i}, L_{j+1,i+2^{j-1}}\right) \\ g\left(L_{j+1,i-2^{j-1}}, L_{j+1,i}, \hat{u}_{j,i-2^{j-1}}\right) \end{cases} = \begin{cases} sign\left(L_{j+1,i}\right)sign\left(L_{j+1,i+2^{j-1}}\right)\min\left(\left|L_{j+1,i}\right|, \left|L_{j+1,i+2^{j-1}}\right|\right), \frac{i-1}{2^{j-1}} mod 2 = 0 (21a) \\ (-1)^{\hat{u}_{j,i-2^{j-1}}}\left(L_{j+1,i-2^{j-1}}\right) + L_{j+1,i}, otherwise (21b) \end{cases} \tag{21}$$

when (18) updates, it is estimated one by one, and the result of previous bit will use in the next bit.

$$S_{i+1,j} \triangleq S_{i,j} \otimes S_{i,j+2^{i-1}}, \frac{i-1}{2^{j-1}} \text{mod} 2 = 0 \tag{22}$$

$$S_{i,j+2^{i-1}} \triangleq S_{i,j} \tag{23}$$

After receiving y_1^N, the decoder can first calculate the initial LLR $L_{n,i}$, and it is updated by the operation of (18) to obtain $L_{1,i}$ and use (15, 16) to estimate $\hat{u}_i$. After estimating $\widehat{u_{i,j}} = S_{i,j}$, The decoder could use (22, 23) to push back $S_{i+1,j}$ and $S_{i,j+2^{i-1}}$,. After the calculation and update of formula (18), the decoder could calculate $L_{1,i}$ of the next message, and use (15) and (16) to estimate the next $\hat{u}_i$.

As shown in Fig. 3, first go the orange line using f function defined in (19) with $L_{1,1}$ and $L_{1,2}$ to calculate the LLR of $\hat{u}_1$ and use (15), (16) to estimate the $\hat{u}_1$. Next, we go the blue

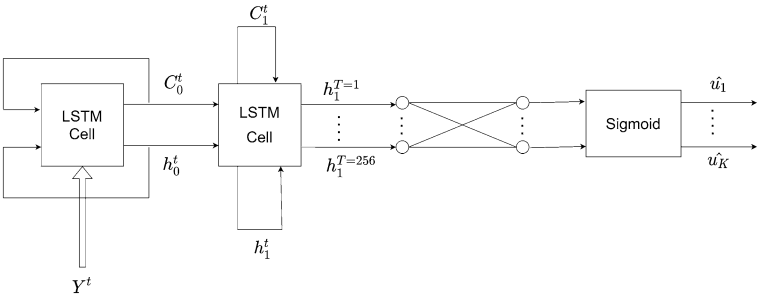

Fig. 4 Two layer LSTM model

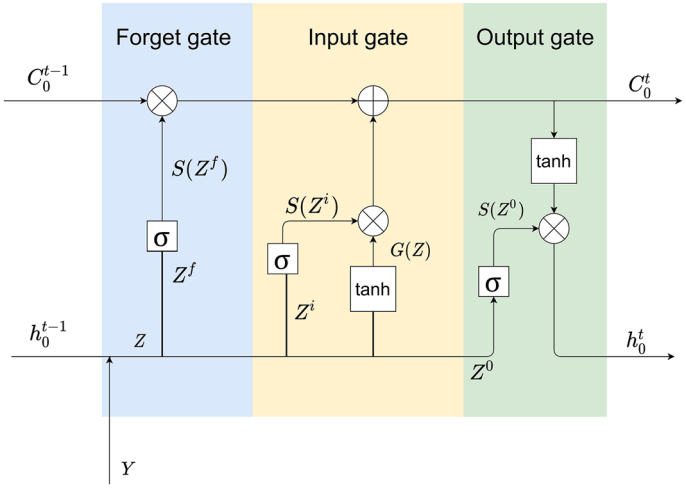

Fig. 5 LSTM cell model

line using g function defined in (20) with $\hat{u_1}$ and $L_{1,1}$ and $L_{1,2}$ to get the LLR of $\hat{u_2}$, then put it to (15), (16) to estimate $\hat{u_2}$.

3 The Proposed Deep-learning Based Decoding for Polar Code

3.1 Long Short-Term Memory(LSTM)

In this work, we use a 2-layer LSTM with a hidden layer size set to 256 and use sigmoid as the output process as shown in Fig. 4. Y^t is the training data (the received signal after impulse noise, and the orginal imformation bit) with different SNR and t is the time. A LSTM cell in Fig. 4 is shown in Fig. 5.

C_0^t is the value in the memory cell (cell state) of the first layer LSTM cell in the time t which is a vector, h_0^t is the first output solution of first LSTM cell (hidden state) in the time t. Next, C_0^t and h_0^t will put into second layer of LSTM Cell to help training. h_0^t size is

(128,256) which mean the size of data 128 and the hidden layer size of prediction output 256.

Finally, We get the hidden state outputh_1^T, T mean the final time of t. We do the matrix multiplication with h_1^T and put it to the sigmoid to get the the$\widehat{u}$, which is (128, 8).

$$\sigma(Z) = \frac{1}{1+e^{-z}} \tag{24}$$

This is sigmoid function, the output is 0 ~ 1.

$$\mathrm{Z} = weight_{h^{t-1}}^{y^t} = [W, U] * \left[h^{t-1}, y^t\right] + bias \tag{25}$$

Z is the real input of LSTM, which is y and h multipe with weight W and weight U and add bias.

$$\mathrm{G(Z)} = \tanh\left(weight_{h^{t-1}}^{y^t}\right) = tanh\left([W, U] * \left[h^{t-1}, y^t\right] + bias\right) \tag{26}$$

G(Z) is the value into the input gate.

$$\mathrm{S}\left(Z^i\right) = \sigma(inputweight_{h^{t-1}}^{y^t}) = \sigma\left(\left[W_i, U_i\right] * \left[h^{t-1}, y^t\right] + bias_i\right) \tag{27}$$

Z^i is the control value, which decide whether G(Z) can put in memory cell, the value of $\mathrm{S}(Z^i)$ will approximate 0 or 1

$$\mathrm{S}(Z^o) = \sigma(outputweight_{h^{t-1}}^{y^1}) = \sigma\left(\left[W_o, U_o\right] * \left[h^{t-1}, y^1\right] + bias_o\right) \tag{28}$$

Z^o is the control value, which decide whether the tanh multiply memory cell value can be output, the value of $\mathrm{S}(Z^o)$ will approximate 0 or 1

$$\mathrm{S}(Z^f) = \sigma(forgetweight_{h^{t-1}}^{y^t}) = \sigma\left(\left[W_f, U_f\right] * \left[h^{t-1}, y^t\right] + bias_f\right) \tag{29}$$

Z^f is the control value, which decide whether the value can be forgot or not, the value of $\mathrm{S}(Z^f)$ will approximate 0 or 1

$$c^t = \left(\left(S\left(Z^f\right)\right) * c^{t-1}\right) + \left(S\left(Z^i\right) * G(Z)\right) \tag{30}$$

(30) is the value in the memory cell, cell state, we can see if $\mathrm{S}(Z^f) = 0$, then the value of input will replace the value of cell (forget), if $\mathrm{S}(Z^f) = 1$ it will add up with input become new value.

$$h^t = S(Z^o) * \tanh\left(c^t\right) \tag{31}$$

(31) is the hidden state output, which is the real output, it is control by Z^o if $S(Z^o)$ is 0, then we can't transmit the output.

3.2 Training Data

The purpose of the decoding is to find the optimal map function, $f^* = Y \rightarrow X$, Y is the set of all possible y and U is the set of all possible of u. f^* should satisfy maximum a posteriori (MAP) criterion:

$$f^* = \underset{u \in U}{argmax} \quad P(u|y) \tag{32}$$

In deep learning, if want to train the neural network, we need a lot of data and then define the loss function.

I. Generating training samples: To train the NN, we generate a large number of received signals **y**, and the real information bits **x**. The received vector **y** is x plus the impulse noise. Taking y as the training data and x as the labeled output to train the model to estimate y to obtain $\widehat{u}_i$.
II. Loss function: It is a very important parameter in NN. It measures the difference between the labeled output and the neural network (NN) output. In this paper, we define loss function as:

$$L_{MSE} = \frac{1}{K} \sum_{i=0}^{K-1} \left(u_i - \hat{u}_i\right)^2 \tag{33}$$

where $u_i = \{0, 1\}$ is the i-th labeled output (correct information bit). $\hat{u}_i$ is the i-th of NN output (estimated information bit).

3.3 Validation

During training, the signal-to-noise ratio (SNR) ρ_t during training must be defined. Because in the actual decoding stage, the SNR is unknown and will change with time, the performance of our NN decoder will be greatly affected by the SNR during training, so we use a performance metric—the normalized validation error (NVE) [38]

$$NVE\left(\rho_t\right) = \frac{1}{S} \sum_{S=1}^{S} \frac{BER_{NN}\left(\rho_t, \rho_{v,s}\right)}{BER_{MAP}\left(\rho_{v,s}\right)} \tag{34}$$

where $\rho_{v,s}$ represents the s-th SNR in a set of S different verification samples, $BER_{NND}(\rho_t, \rho_{v,s})$ represents the BER obtained by NN decoder training SNR ρ_t. $BER_{MAP}(\rho_{v,s})$ is the BER of MAP decoding at the SNR $\rho_{v,s}$. The intention of this method is measuring the performance of NN decoder trained on a specific SNR compared to MAP decoding in different ranges of SNR. From (34), we can know that the less the NVE, the better the NN decoder, and there will be an optimal ρ_t. So, we train the NN decoder with datasets of different ρ_t in our work, and choose the optimal ρ_t which results in the least NVE.

In Fig. 6, we can see the SNR of 4.5 dB has the best performance, we choose this SNR as our NN decoder's training SNR, so there is only one NN model and training data are of SNR 4.5 dB, and testing data are of SNR in 0 ~ 6 dB (13 different SNR values). The optimal training SNR in the impulse noise channel is different from that in the AWGN channel. In the prior work [38], the optimal training SNR is 1.5 dB, not 4.5 dB, for the same testing SNR range 0–6 dB.

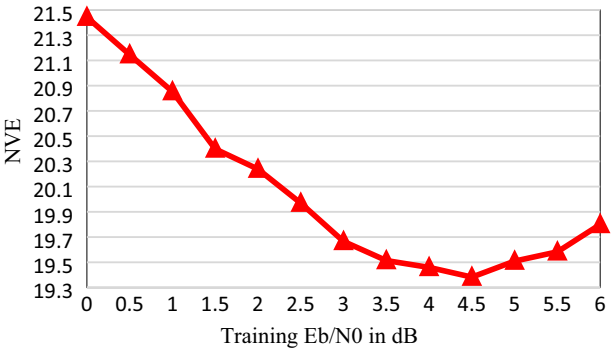

Fig. 6 NVE training E_b/N_0 for 16 bit length codes in impulse noise

4 The Numerical Results

4.1 Setting Environment of the Proposed Method and Comparing Methods

We compare the following schemes: NN (proposed), SC (comparing), BP (comparing), SCL (comparing), MAP (comparing). The common setting environment/ parameters of all schemes are listed in Table 1. We use the same (16,8) Polar code as prior works without impulse noise [38, 39].

The unique setting environment/ parameters for each method is as follows. The number of iterations for BP method is 100. We try 50,100, 200 iterations [24] and found 100 and 200 iterations have similar performance, so we set 100 iterations for BP method. The list size for SCL method is 2. We try list size 2,4,8,16 [4] and find they have similar performance so I select list size 2 for SCL method. In this work we don't set the CRC for SCL. The hyperparameters for the proposed NN method are listed in Table 2.

One important finding is that the optimal training SNR in the impulse noise channel is different from that in the AWGN channel. In the prior work [38], the optimal training SNR is 1.5 dB, not 4.5 dB in Fig. 6, for the same testing SNR range 0–6 dB (Table 1).

Table 1 Common setting environment of the proposed (NN) and comparing methods (SC, BP, SCL, MAP)

Parameter	Value
N (Polar code codeword length)	16
K (The number of information bits in a Polar code codeword)	8
SNR range	0 ~ 6 dB
Markov Gaussian memory impulse noise channel state transition probability matrix **M**	$\begin{bmatrix} 0.99 & 0.01 \\ 0.2 & 0.8 \end{bmatrix}$
CPU	I7–8700
GPU	Nvidia GeForce RTX 2080Ti

Table 2 The hyperparameters of the proposed method (NN)

Parameter	Value
Learning rate	0.0001
Dropout rate	0.2
Activation Function	Sigmoid
Loss function	Mean squared error
Epochs	10^5
Training data	10^6
Testing data	10^5
Batch Size (in codewords)	128
Number of LSTM cell	2
Hidden layer size	256
Forget bias	1
Initialization Method	Xavier initialization
Optimization method	Adam
Training SNR	4.5 dB (from Fig. 6)

4.2 Simulation Results

4.2.1 In the AWGN

Figure 7 shows the BERs of the proposed deep learning based decoding scheme (NN) and previous non-deep-learning based schemes SC/BP/SCL and the MAP bound are close to each other's.

4.2.2 In the Impulse Noise

For Markov Gaussian memory impulse noise channel with state transition matrix in (10), we compare the BERs of the proposed deep learning based decoding scheme (NN) and previous non-deep-learning based schemes SC/BP/SCL and the MAP bound in Fig. 8.

We can see all decoders under the impulse noise has a BER gap with the MAP bound, but the BER of proposed NN decoder is about one third of that of SC, SCL, and BP. One finding is that the NN decoder could learn to a degree the memory impulse noise. For comparison SC/BP/SCL/MAP perform poorly when the noise model in not exactly AWGN. Thus, the proposed NN decoded is more suited to memory impulse noise channel than previous SC/BP/SCL decoding schemes.

Another finding is that the proposed LSTM-based method is $5 \sim 12$ times less and thus has much less decoding latency than that of SC/BP/SCL methods because the proposed LSTM-based method has inherent parallel structure and has one shot operation- no iterative operations.

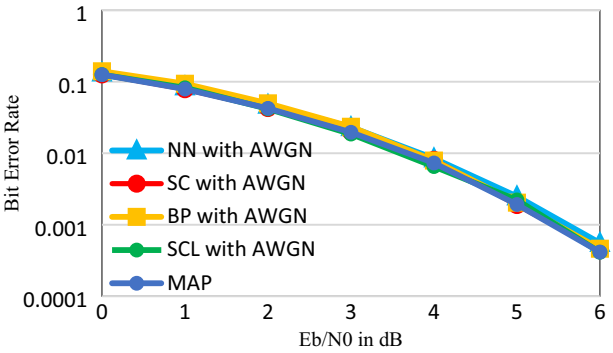

Fig. 7 AWGN

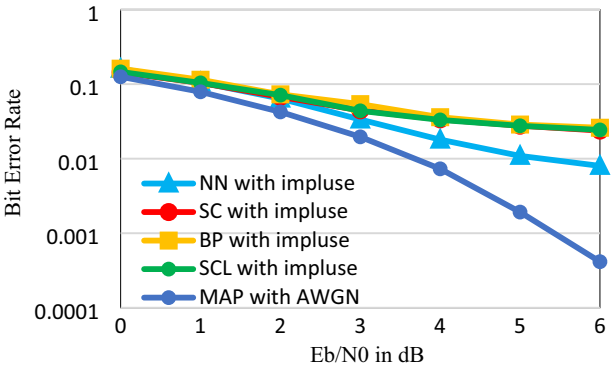

Fig. 8 Impulse noise

5 Conclusion

Deep learning-based Polar code decoding method has parallel structure in nature and has lower execution time and thus decoding latency. However, prior works do not consider memory impulse channels which has more complicated statistical model about average number of impulses in a codeword and the positions of the impulses in a codeword etc. In this paper, we propose the use LSTM for the Polar code decoding under the Markov Gaussian memory impulse noise channels. For the SNR range 0 ~ 6 dB, we find the optimal SNR value 4.5 dB, which is different from the 1.5 dB in the AWGN channel found in [38, 40]. The data set with training SNR 4.5 dB is used to train the proposed LSTM-based Polar code decoding method. The bit error probability of the proposed LSTM-based method is only one-third that of the conventional SC/BP/SCL decoding schemes under Markov Gaussian channels, and 5 ~ 12 times faster in execution time and decoding latency.

Funding This study was funded by the Ministry of Science and Technology, Taiwan, (Grant Number MOST 109–2221-E-027-087).

Availability of data and material Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Arikan, E. (2009). Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory, 55*(7), 3051–3073.
2. Hui, D., Sandberg, S., Blankenship, Y., Andersson, M., & Grosjean, L. (2018). Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE. *IEEE Vehicular Technology Magazine, 13*(4), 60–69.
3. Arikan, E. (2010). Polar codes: A pipelined implementation. In: Proceedings of 4th ISBC, (pp. 11–14).
4. Tal, I., & Vardy, A. (2015). List decoding of polar codes. *IEEE Transactions on Information Theory, 61*(5), 2213–2226.
5. Wang, X., & Rong, C. (2001). Blind turbo equalization in Gaussian and impulsive noise. *IEEE Transactions on Vehicular Technology, 50*(4), 1092–1105.
6. Axell, E., Wiklundh, K. C., & Stenumgaard, P. F. (2015). Optimal power allocation for parallel two-state Gaussian mixture impulse noise channels. *IEEE Wireless Communications Letters, 4*(2), 177–180.
7. Axell, E., Eliardsson, P., Tengstrand, S. Ö., & Wiklundh, K. (2017). Power control in interference channels with class a impulse noise. *IEEE Wireless Communications Letters, 6*(1), 102–105.
8. Blackard, K., & RappaportBostian, T. C. (1993). Measurements and models of radio frequency impulsive noise for indoor wireless communications. *IEEE Journal on Selected Areas in Communications, 11*(7), 991–1001.
9. Blankenship, T., et al. (1997). Measurements and simulation of radio frequency impulsive noise in hospitals and clinics. In: *Proceedings of 1997 IEEE 47th vehicular technology conference* (VTC), vol.3, (pp. 1942–1946).
10. Tseng, D.-F., Mengistu, F. G., Han, Y. S., Mulatu, M. A., Chang, L.-C., & Tsai, T.-R. (2014). Robust turbo decoding in a Markov Gaussian channel. *IEEE Wireless Communications Letters, 3*(6), 633–636.
11. Tseng, S.-M., Lee, T.-L., Ho, Y.-C., & Tseng, D.-F. (2017). Distributed space-time block codes with embedded adaptive AAF/DAF elements and opportunistic listening for multihop power line communication networks. *International Journal of Communication Systems, 30*(1), e2950.
12. Fertonani, D., & Colavolpe, G. (2009). On reliable communications over channels impaired by bursty impulse noise. *IEEE Transactions on Communications, 57*(7), 2024–2030.
13. Cheffena, M. (2012). Industrial wireless sensor networks: Channel modeling and performance evaluation. *EURASIP Journal of Wireless Communications Networking, 2012*(297), 1–8.
14. Alam, M., Labeau, F., & Kaddoum, G. (2016). Performance analysis of DF cooperative relaying over bursty impulsive noise channel. *IEEE Transactions on Communications, 64*(7), 2848–2859.
15. Tseng, S.-M., Wang, Y. C., Hu, C. W., & Lee, T. C. (2018). Performance analysis of CDMA/ALOHA networks in memory impulse channels. *Mathematical Problems in Engineering, 2018*, 9373468.
16. Ghosh, M. (1996). Analysis of the effect of impulse noise on multicarrier and single carrier QAM systems. *IEEE Transactions on Communications, 44*(2), 145–147.
17. Shongwe T, Vinck, AJH, Ferreira, H. C. (2014). On impulse noise and its models. In *Proceedings of 18th IEEE international symposium on power line communications and its applications*, (pp.12–17).
18. Han B, Schotten, H. D. (2018). A fast blind impulse detector for Bernoulli-Gaussian noise in underspread channel. In: *Proceedings of 2018 IEEE international conference on communications (ICC)* (pp. 1–6).
19. Tseng D.-F, Lin, Y-D, Tseng, S-M. (2020) Practical polar code construction over memoryless impulse noise channels. In: *Proceedings of 2020 IEEE 91st vehicular technology conference (VTC2020-Spring)*, (pp. 1–5,).
20. Middleton, D. (1977). Statistical-physical models of electromagnetic interference. *IEEE Transactions on Electromagnetic Compatibility, EMC-19*(3), 106–127.

21. Laere V Bette, S., Moeyaert, V. (2016). Poster: ITU-T G. 9903 performance against middleton class-A impulsive noise. In: *Proceedings of 2016 symposium on communications and vehicular technologies* (SCVT) (pp. 1–6).
22. Al-Rubaye, G., Tsimenidis, C. C., & Johnston, M. (2018). Performance evaluation of T-COFDM under combined noise in PLC with log-normal channel gain using exact derived noise distributions. *IET Communications, 13*(6), 766–775.
23. Chen W. (2018). Polar code over memoryless impulse noise channels. M.S. paper, Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan.
24. Lin Y.-D. (2019), Polar code over different decoding method for performance analysis comparison. M.S. paper, Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan.
25. Krizhevsky A., Sutskever, I, Hinton, G. E.,. (2012). ImageNet classification with deep convolutional neural networks. In: *Proceedings of advances in neural information processing systems,(*pp. 1090–1098).
26. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine, 29*(6), 82–97.
27. Tseng, S.-M., Chen, Y.-F., Tsai, C.-S., & Tsai, W.-D. (2019). Deep-learning-aided cross-layer resource allocation of OFDMA/NOMA video communication systems. *IEEE Access, 7*, 157730–157740.
28. Tseng, S.-M., Tsai, C.-S., & Yu, C.-Y. (2020). Outage-capacity-based cross layer resource management for downlink NOMA-OFDMA video communications: Non-deep learning and deep learning. *IEEE Access, 8*, 140097–140107.
29. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436–444.
30. LeCun Y, Denker J. S, Henderson D, Howard R. E, Hubbard W, Jackel L. D. (1990). Handwritten digit recognition with a back-propagation network. In: *Proceedings of advances in neural information processing systems*, (pp. 396–404).
31. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of IEEE, 86*(11), 2278–2324.
32. Werbos, P. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE, 78*(10), 1550–1560.
33. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *Proceedings of conference on empirical methods in natural language processing*, (pp. 1724–1734,).
34. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
35. Kim, H, Jiang, Y., Rana, R., Kannan, S., Oh, S., Viswanath, P. (2018). Communication algorithms via deep learning. In: *Proceedings of international conference on learning representations* (ICLR).
36. Liang, F., Shen, C., & Wu, F. (2018). An iterative BP-CNN architecture for channel decoding. *IEEE Journal of Selected Topics in Signal Processing, 12*(1), 144–159.
37. Cammerer, S, Gruber, T., Hoydis, J., ten Brink, S. (2017) Scaling deep learning-based decoding of polar codes via partitioning. In: *Proceedings of 2017 IEEE Global Communications Conference*, (pp. 1–6).
38. Gruber, T, Cammerer, S., Hoydis, J., ten Brink, S. (2017). On deep learning-based channel decoding. In: *Proceedings of 2017 51st annual conference on information sciences and systems (CISS)*, (pp. 1–6).
39. Irawan, A., Witjaksono, G., Wibowo, W. K. (2019). Deep learning for polar codes over flat fading channels. In: *Proceedings of 2019 international conference on artificial intelligence in information and communication (ICAIIC)* (pp. 488–491).
40. Lyu, W., Zhang, Z., Jiao, C., Qin, K., Zhang, H. (2018). Performance evaluation of channel decoding with deep neural networks. In: *Proceedings of 2018 IEEE international conference on communications (ICC)*, (pp. 1–6).
41. Doan, N., Hashemi, S. A., Gross, W. J. (2018). Neural successive cancellation decoding of polar codes. In: *Proceedings of 2018 IEEE 19th international workshop on signal processing advances in wireless communications (SPAWC)*, (pp. 1–5).
42. Gross, W., Doan, N., Mambou, E. N., & Hashemi, S. A. (2020). Deep learning techniques for decoding polar codes. In F.-L. Luo (Ed.), *Machine learning for future wireless communications* (pp. 287–301). Wiley.

43. Chen, C. -H., Teng, C-F, Wu, A-Y (2020). Low-complexity LSTM-assisted bit-flipping algorithm for successive cancellation list polar decoder. In: *Proceedings of 2020 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, (pp. 1708–1712).
44. He, B., Wu, S., Deng, Y., Yin, H., Jiao, J., Zhang, Q. (2020). A machine learning based multi-flips successive cancellation decoding scheme of polar codes. In: *Proceedings of 2020 IEEE 91st vehicular technology conference (VTC2020-Spring)*, (pp. 1–5).

Shu-Ming Tseng received a B.S. degree from National Tsing Hua University (highest honors), Taiwan and M.S. and Ph.D. degrees from Purdue University, IN, USA, all in electrical engineering, in 1994, 1995, and 1999, respectively. He was with the Department of Electrical Engineering, Chang Gung University, Taiwan, from 1999 to 2001. Since 2001, he has been with the Department of Electronic Engineering, National Taipei University of Technology, Taiwan where he is currently a Professor. From 2014-2015, he was a visiting scholar with the Department of Electrical and Computer Engineering, Universirty of California San Diego, CA, USA. He is the author of 54 SCI journal papers. His research interests include AI, 5G, radio resource allocation, cross layer design, video temporal segmentation, network performance evaluation, and optical sensors. He has been an Editor for KSII Transactions on Internet and Information Systems since 2013. He was a TPC Co-Chair of IEEE WOCC 2021. He was a TPC member for IEEE IWCMC 2020 Low Power Wide Area Networks Technologies for Internet of Things Symposium He took part in the academic survey of QS World University Rankings and Times Higher Education World University Rankings since 2019.

Wei-Cheng Hsu received an M.S. degree from the Department of Electronic Engineering at National Taipei University of Technology in 2020. His research interests are deep learning, polar codes, and wireless communications. He is currently a Software Engineer in Pegatron Corporation, Taipei, Taiwan.

Der-Feng Tseng received the B.S. degree in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, and the M.S. degree in electrical engineering and the Ph.D. degree from Purdue University, West Lafayette, IN, USA. He was with TRW-ESL, Sunnyvale, CA, USA, and Motorola Mobility, Pistcataway, NJ, USA, as a Senior Systems Engineer engaged in embedded soft- ware development for mobile devices before joining the National Taiwan University of Science and Technology, Taipei, Taiwan, where he has been since February 2003. From June to September 2013, he was a Visiting Research Scholar with the Department of Electrical Engineering, University of Houston, Houston, TX, USA. His research interests include communication systems, statistical signal processing, and channel coding techniques. Dr. Tseng received the IEEE Fred W. Ellersick Award for best paper in the unclassified technical program at the 1998 IEEE Military Communications Conference.