1. Introduction

With its recent development, deep learning has shown great achievements when applied to areas outside the computer science domain. Therefore, deep learning has also been applied to physical layer communication research such as channel coding and antenna technology. There are two reasons why deep learning is used in communication systems: performance improvement and complexity reduction. Based upon the,with decoding is applied to communication systems Before the integration of deep learning, noise, which is added when a message passes through a channel, can prevent the receiver from receiving the message. To improve this, there is a process called encoding and decoding in the classical communication domain. There is a popular decoding method, known as the Belief Propagation Algorithm (BPA), also named the Sum Product Algorithm (SPA). This algorithm has good performance, but it is composed of many multiplication operations. Therefore, the longer the length of the message used as an input value, the more complicated calculation is. In such circumstances, the second reason to use deep learning, reducing complexity is important. To solve the complexity problem, there is a solution called the min-sum algorithm (MSA). In this min-sum algorithm, the complexity problem was improved, but performance loss degradation occurred. To properly adjust this trade-off relationship between performance and complexity, there are two algorithms: 1. The Normalized min-sum algorithm (NMSA) that multiplies correction factor value, which is a constant value, from the check node update process 2. The Offset min-sum algorithm (OMSA) that adds or subtracts correction factor value from check node update process. By incorporating deep learning in this way into existing communication systems, the correction factor is optimized. Remarkably, results from the use of these algorithms showed better performance than MSA and improved complexity than BPA, respectively. This is a point.

<related work>

Recently, many researchers have been actively researching methods to incorporate deep learning into channel coding. A formative study, by Nachmani, used deep learning in the decoding process. By setting different weight values at the edges connecting check nodes(CN) and variable nodes(VN), [1] improved performance by reducing the effect of small cycles in tanner graphs during the decoding process. In [2], unlike deep learning’s application to BPA, Lugosch applied it to OMS. Deep Learning was used to obtain OMS’s optimized correction factor value. More computations result in more complexity, more load, higher hardware device temperatures, and permanent performance degradation. Therefore, to improve this, complexity is one of the problems that hardware must solve. Particularly, OMS is an algorithm consisting of addition and subtraction, not a more complex multiplication calculation, so it is a suitable method for an algorithm to hardware because of its low complexity. This algorithm is called a neural offset min-sum (NOMS). [3] conducted similar research where a neural normalized min-sum (NNMS) was proposed using an optimized correction factor through deep learning. Alternatively, Wang et. al suggested a different method. In order to improve the complexity problem, they used a sharing method that uses the same correction factor value for each iteration, unlike recent studies that used different correction factor values for each iteration and node. This algorithm is called a Shared Neural NMS (SNNMS).

The aforementioned research used deep learning to optimizing correction factor. However, there is a study focusing on refining the deep learning architecture for this application. Deep Learning has several architectures such as Deep Neural Network (DNN), Convolutional Neural Network(CNN), and Recurrent Neural Networks(RNN). [4] is a seminal work using ‘RNN’ called a ‘circular neural network’. This RNN utilizes past data for learning through the concept of a recurrent. In other words, it is an algorithm that utilizes not only current inputs but also past data for learning. Remarkably, this first study to incorporate ‘RNN’ into the decoding process showed similar performance to prior studies using fewer parameters. Subsequently, researcher improved performance by incorporating “relaxation” into the RNN architecture in [5]. The purpose of using this relaxation concept was to determine how much previous data to use. Notably, this method optimized the decoder relaxation factor through deep learning as opposed to the previous method, brute force simulation.

However, ‘RNN’ has two limitations. To be specific, input vectors are entered sequentially to enable sequential data processing, but ‘parallelization operation’ is not possible. The derivative value of tanh, activation function of RNN, is used in this case. However, there is a disadvantage that back propagation information is rarely transferred because a vanishing gradient occurs. To solve these problems, our method used Long Short Term Memory (LSTM, a special case of RNN) along with the concept of relaxation. In order to show excellent performance from the proposed method, simulations were conducted in BCH code, which is a high-density parity check code (HDPC) and low density parity check code (LDPC) with different lengths. Also, the semiconductor company ‘NVIDIA’ recently announced an open source simulator ‘Sionna’ [6] to help research 5G communication using deep learning. In channel coding, there are many methods to compare performance, and I used Bit Error Rate (BER), which is number of bits that have errors in the process of being transmitted to the number of bits received. In the simulation part, performance was compared through simulation of BER and it will be shown that performance is better when using our proposed method than when using RNN.

<reference>

[1] Nachmani, Eliya, Yair Be'ery, and David Burshtein. "Learning to decode linear codes using deep learning."2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2016.

[2] Lugosch, Loren, and Warren J. Gross. "Neural offset min-sum decoding." 2017 IEEE International Symposium on Information Theory (ISIT). IEEE, 2017.

[3] Wang, Qing, et al. "A model-driven deep learning method for normalized min-sum LDPC decoding." 2020 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2020.

[4] Nachmani, Eliya, et al. "RNN decoding of linear block codes," arXiv preprint arXiv:1702.07560(2017).

[5] Nachmani, Eliya, et al. "Deep learning methods for improved decoding of linear codes." IEEE Journal of Selected Topics in Signal Processing 12.1 (2018): 119-131.

[6] Hoydis, Jakob, et al. "Sionna: An Open-Source Library for Next-Generation Physical Layer Research." arXiv preprint arXiv:2203.11854 (2022).