Kimheeseo_GEDU501-02_research_report_Natasha Powell

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 first reason, deep learning technology with decoding is applied to communication systems to achieve better performance. 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 re 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 research 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 'pruning'. It is a method of reducing the parameters of the model by removing the connection of the weight of the model that is less important. Through 'pruning', it reduces the number of parameters, which solves the computational volume and complexity problem, but has the disadvantage of losing information. A similar method, drop out, is performed for purpose of regularization. The difference is that this is applied by not using part for a while when learning, storing it in a way that it is used again later, and then using it again.

Nevertheless, proposed method was applied focusing on the advantages of improving complexity of pruning. 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 OMS.

메모 포함[NP1]: Run on sentence.

Kimheeseo_GEDU501-02_research_report_Natasha Powell

1. Result

Research goal is to further improve performance of conventional neural offset min-sum (NOMS) by utilizing "pruning". Prior decoding method was to train beta values of offset min-sum (OMS) to optimize. Since beta value changes depending on the importance of result, beta is used as an indicator of importance. Therefore, I proposed a new method, thinking that removing unimportant values would not significantly affect performance and would reduce amount of computation. The proposed method uses [1] "pruning", a method of reducing the number of parameters of model. The reason for using method is to reduce amount of computation to improve complexity. The novelty of proposed method is to present a new criterion for pruning.

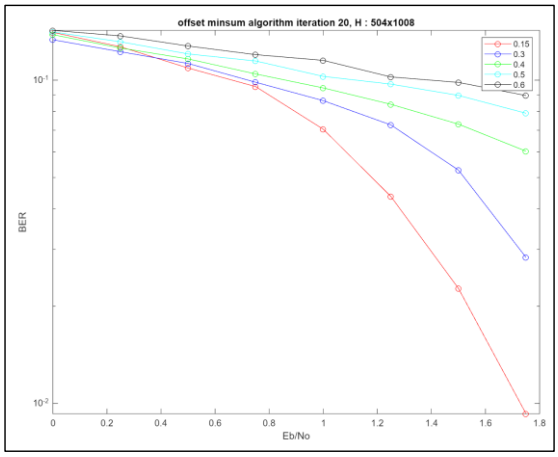

Fig 0. BER performance depending on beta value

The reason why I became interested in optimized beta value is that it has different performance depending on it as shown in fig 0., so it was conducted to find the optimal value of the value.

1.1) process

It is neural offset min-sum (NOMS) equation.

$$\mu_{c,v}^{t} = \mathrm{ReLU}\left(\min_{v'}\left(\left|\mu_{v',c}^{t}\right| - \beta_{c,v}^{t}\right)\right)\prod_{v'}\mathrm{sign}\left(\mu_{v',c}^{t}\right), \qquad (1)$$

where $\beta_{c,v}^{t}$ is a learnable offset parameter for the edge connecting CN c to VN v during iteration t. Simulation proceeds to the following process. First, to apply neural network to decoding process, it is made into an unrolled structure. And In Fig.1, the 'odd' th layer means variable node (VN) layer, and the 'even' th layer

means check node (CN) layer. In particular, I optimized beta value assigned to check node update and remove value based on a 'special criteria'. That is, making the small value zero is method of removing the unimportant value.

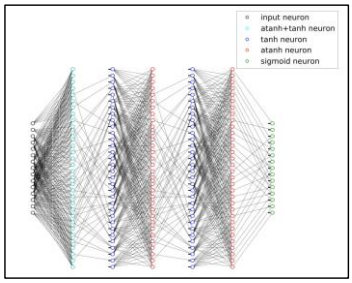

Fig 1. Deep Neural Network Architecture For BCH (15,11) with 5 hidden layers which correspond to 3 full BP iterations in [1]. It is noteworthy that this structure was used as a basic model for decoding algorithm research with deep learning.

1.2) setting

Simulation was conducted using [2] 'TensorFlow', using (200, 100), (120,60) LDPC code, AWGN channel and BPSK modulation, and all zero codeword. So, I inject noisy codewords after transmitting through an AWGN channel and measure the bit error rate (BER) in the decoded codeword at the network output. Training was conducted using Batch Gradient descent and used [3] Adam optimizer with a learning rate equal to 0.0001.

1.3) result

1.3.1) pruning

In simulation, it can show that it has similar performance to traditional research methods (OMS). In particular, it is remarkable to show that it has similar performance while reducing complexity compared to traditional methods.

Kimheeseo_GEDU501-02_research_report_Natasha Powell

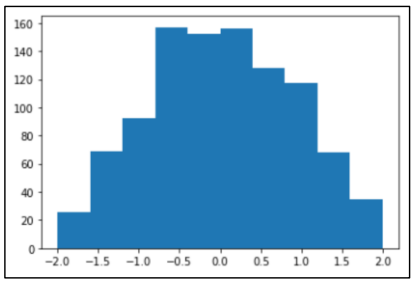

Fig 2. Distribution of beta values before training

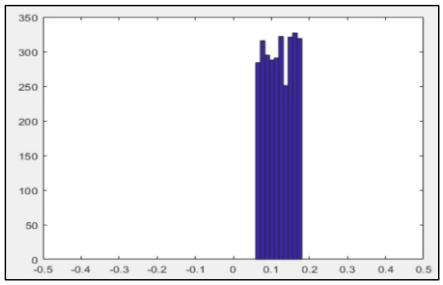

Fig 3. Distribution of beta values after training

Proposed method sets the optimal value for setting 0.15 as the beta value to the target value, and then creates an input value for setting the beta value randomly. So, Fig 2. shows that initialization value of beta value before training. Then, find the error value through difference from the target value, and find optimal value through the gradient descent value.

$$\beta = \beta - \eta \frac{\partial error}{\partial \beta},$$

where η is learning rate (0.0001) and error is difference between target value and estimated value obtained through training. During training process, I judged that the value of beta in last iteration is greater than or less than 15% of the target value and pruned it. As Fig 3. shows, "number of beta" is significantly reduced and the performance is evaluated with Bit Error Rate (BER) similar to the traditional method (OMS) in Fig 4. Therefore, by reducing the number of beta and reducing amount of computation, performance is shown to be superior through the fact that performance is similar to traditional methods while improving complexity. At this point, values greater than or less than 15% of 0.15 are considered insignificant in the result value, and zeroed out. This reduces the amount of computation by reducing the values you do not need as much as possible. Fig 4. can be seen that it has similar performance while improving complexity by comparing Bit Error Rate (BER) of proposed method and OMS.

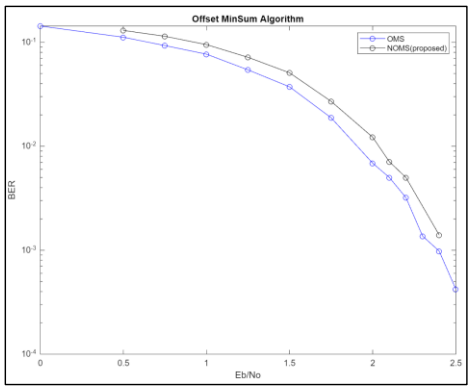

Fig 4. Bit error rate performance comparison between OMS and NOMS

And Table 5 shows the number of beta, and 3024 mean that 0.15 is set at all 1 positions of parity check matrix (H). Although beta was reduced by more than half in all dB, it was confirmed through Fig. 4 that the performance was similar.

	0 dB	0.5 dB	0.75 dB	1 dB	1.25 dB	1.5 dB	1.75 dB	2 dB	2.1 dB
OMS	3024	3024	3024	3024	3024	3024	3024	3024	3024
NOMS	1490	1484	1462	1479	1449	1442	1493	1464	1536

Table 5. Comparison of beta numbers of OMS and proposed methods according to dB

1.3.2) iteration and various codes

Since performance and complexity have a trade-off relationship, if Number of iterations or Size(length) of code are increased to improve performance, performance improves, but complexity increases. In this section, while performing a simulation by changing 'iteration' and 'code size', it is confirmed whether condition is also true in the proposed method.

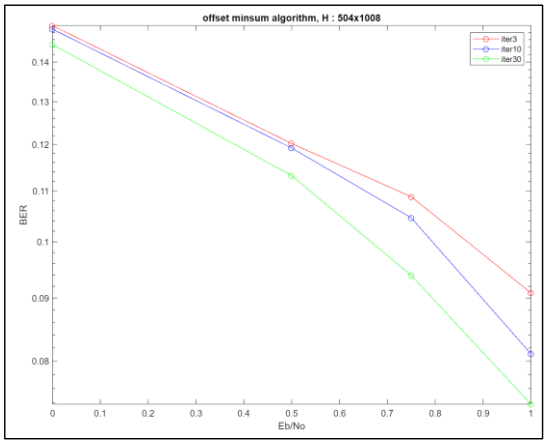

Figure 6. Result that depend on iteration

Fig 6. is a comparison of BER performance that varies for different iteration, and it is confirmed that performance improves as the iteration increases.

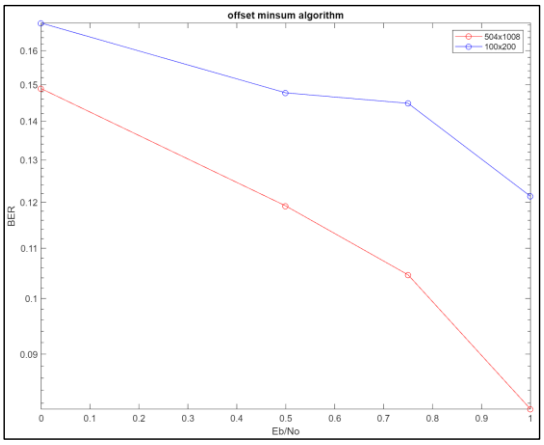

Figure 7. Result that depend on code length

Fig 7. compares BER performance of those with different code lengths, and also confirmed that performance improves as code length increases. Through this, it was confirmed that the existing law was established.

2. conclusion

[1] Recent research have shown performance improvement with the recent incorporation of deep learning into Channel Coding, which has led to the emergence of methods such as [4] neural offset minsum (NOMS), [5] neural normalized misum (NNMS), and shared neural normalized minsum (SNNMS) in prior researches. Prior research that proposed 'NOMS' and 'NNMS' mainly focused on performance improvement, and 'SNNMS' research focused on improving complexity. In particular, since there is a trade-off relationship between complexity and performance improvement, an appropriate value should be adjusted, and the proposed method focused on improving complexity. At this time, complexity was improved by reducing the amount of calculation through a method called 'pruning'. It is to reduce the amount of computation by training different beta for each edge between variable node (VN) and check node (CN) in OMS and removing small value per row. The reason is that the size of value represents the importance of decoding as indicator. Through simulation, it shown that performance of proposed method shows similar performance to prior one(OMS) while improving complexity. Through this, it is easier to apply hardware. The future topic of research is to apply proposed method in various code as well as in BCH, Polar code, and I will look at whether proposed method can be applied to NNMS, SPA.

reference

Kimheeseo_GEDU501-02_research_report_Natasha Powell

[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] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin et al., "Tensorflow: Large-scale machine learning on heterogeneous systems, 2015," Software available from tensorflow. org, vol. 1, 2015.

[3] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

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

[5] 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.