GEDU501: Scientific Writing **POSLEC**

Name: kimheeseo Date: 2022.09.12

Article Annotation Practice

Preparing for an Annotated Bibliography

Instructions (설명)

1. *Insert information from one research article you will read in the* ***Article Information*** *table.*
2. *Answer the annotation questions in the* ***Before Reading*** *and* ***After Skimming*** *tables.*
3. *In* ***Article Notes and Highlighting****, copy and paste annotations you've made directly on the article. (and delete the example)*

Article Information

Title	Learned Decimation for Neural Belief Propagation Decoders : Invited Paper
Author (s)	Andreas Buchberger, Christian Häger, Henry D. Pfister, Laurent Schmalen, Alexandre Graell i Amat
Journal Title	ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
Year of Publishing	06-11 June 2021
Volume/Issue	
Pages	p.5
Keywords / Search Terms	LDPC, NBP(neural belief propagation), decimation

Article Notes (a.k.a. Annotation)

Before Reading the Article	
Question	Answer
What key information are you hoping to get from this paper? Or What questions are you expecting to be answered by reading this paper?	Recent research trend is to use deep learning in many research field. Therefore, there is a question of how to use the neural network architecture. What algorithms or deep neural network method(architecture) do researcher use?
Which section of the paper is the best place to find that information (or answer	'section 3. Decimated neural belief propagation decoder' deals with the neural network(NN), an important part of the paper. Total decoding process consists of a two

those questions)?	steps, ‘List-based Decimation Stage’ and ‘Learned Decimation Stage’, and each stage process and order are described.
How does this paper relate to other research/articles in the field? *(Is the paper a foundation paper that explains basic methods that are now used by all researchers, is the paper using a novel technique or perspective from prior research?)*	Recent research trend is using deep learning. (in the paper) In particular, deep learning is used to improve performance and complexity. The main goal of this paper is to improve complexity issue, so I think it will be helpful for future research. This paper is using a novel technique.

After skimming the article (reading abstract, figures, reading the beginning and/or end of key paragraphs/section)	
Question	Answer
What info in this article is still useful to you?	My interest subject is channel coding with deep learning. So, It was helpful because it was one of the various methods of deep learning. (In other paper) Many researchers focused on check node updates, but it is characterized by focusing on variable nodes in this paper. It is interesting part of this paper.
What is the purpose or aim of this study(the research done in this paper)?	adjust tradeoff of performance and complexity. Reducing complexity compared to previous studies (paper or research), and making similar performance as possible, i.e. adjusting tradeoff appropriately.
What is/are the main/key outcomes (results/conclusions)?	adjust tradeoff of performance and complexity.
What was/were important aspect(s) of their procedure or methods?	Since researcher don't know sign, researcher set it to +- infinity as a method for decimation and used 'learned decimation stage' to use NN(neural network). In particular, the parameters were optimized through deep learning. However, limitation thing is that It didn't mention what parameter is specifically, but I think it's weight. And it had other limitation thing that there was no standard for how many decimation times(number) each of the two stages.
Were there any limitations of the study? What were those limitations? *(perspectives the research didn’t cover, missing or unclear information, . . .)*	‘Pruning neural belief propagation decoders’ is a paper that studies similar methods. The performance was compared with 'previous paper', a similar study. But the performance did not seem to be compared under the same conditions(situation).

Article Notes and Highlighting

In the empty space below, input an image of one page (of a hand annotated) or screen capture (of a digitally annotated) the journal article.

LEARNED DECIMATION FOR NEURAL BELIEF PROPAGATION DECODERS

(Invited Paper)

Andreas Buchberger, Christian Häger*, Henry D. Pfister†, Laurent Schmalen‡, and Alexandre Graell i Amat**

* Dept. of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
† Dept. of Electrical and Computer Engineering, Duke University, Durham, North Carolina, USA
‡ Communications Engineering Lab, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

arXiv:2011.02161v1 [cs.IT] 4 Nov 2020

ABSTRACT

We introduce a two-stage decimation process to improve the performance of neural belief propagation (NBP), recently introduced by Nachmani *et al.*, for short low-density parity-check (LDPC) codes. In the first stage, ① we build a list by iterating between a conventional NBP decoder and guessing the least reliable bit. The ② second stage iterates between a conventional NBP decoder and learned decimation, where we use a neural network to decide the decimation value for each bit. For a (128,64) LDPC code, the proposed NBP with decimation outperforms NBP decoding by 0.75 dB and performs within 1 dB from maximum-likelihood decoding at a block error rate of 10^{-4}.

1. INTRODUCTION

Belief propagation (BP) decoding can be formulated as a sparse deep neural network (NN) where instead of iterating between check nodes (CNs) and variable nodes (VNs), the messages are passed through unrolled iterations in a feed-forward fashion [1, 2]. For short block codes, the weights associated with the edges may counteract the effect of short cycles in the graph by scaling the messages accordingly. This is commonly referred to as neural belief propagation (NBP) and can be seen as a generalization of weighted BP decoding where all individual messages are scaled by a single coefficient [3]. For a given code, the performance of BP and NBP can be improved by using redundant parity-check matrices at the cost of increased complexity [4–10]. A pruning-based NBP decoder was introduced in [10], which improves the performance of NBP at a lower decoding complexity.

edge에 각각 다른 가중치 값을 할당.

It can be observed that the gain of NBP over BP for low-density parity-check (LDPC) codes is mostly in reducing the number of decoding iterations—the performance of NBP and BP converges for a sufficiently large number of iterations and a fundamental gap between (N)BP and maximum-likelihood (ML) decoding remains. Indeed, the error rate of (N)BP decoding of LDPC codes is dominated by absorbing sets from which the (N)BP decoder cannot recover. One way to improve performance is to run multiple decoders with different hard guesses for the least reliable bits [11–13]. Each guess naturally splits the decoding process into two parallel decoders, one for each guess [13]. Similarly, in problems where multiple codewords can have similar posterior probabilities, such as lossy compression, one may make hard decisions for the most reliable bits without splitting the decoder, helping the BP algorithm to converge to a codeword [14–17]. We refer to both of these operations as *decimation*, though that term is typically associated with the second case.

In this work, we propose the use of decimation for NBP decoding, thereby introducing a neural belief propagation with decimation (NBP-D) decoder. Our proposed decimation scheme consists of a ① *list-based decimation stage* and a ② *learned decimation stage*. We start with the list-based decimation stage and run a conventional NBP decoder for $\ell_{\max}$ iterations. We identify the least reliable VN, i.e., the VN with the lowest absolute *a posteriori* log-likelihood ratio (LLR) and decimate it to $\pm\infty$. Choosing the correct sign is essential as the correct sign will aid convergence whereas the incorrect sign will hinder it. As the correct sign is unknown, we proceed with two graphs—one where the VN is decimated to $+\infty$ and one where the VN is decimated to $-\infty$. We iterate between decimating and decoding using the NBP decoder for the desired number of times and end up with a list of codeword candidates. After the list-based decimation stage is complete, we continue with a learned decimation stage. Similar to the list-based decimation stage, we run a conventional NBP decoder for $\ell_{\max}$ iterations. For each VN, we then use an NN to decide to which value each VN is decimated to. The sign of the VN is not changed. We then iterate between the conventional NBP decoder and the learned decimation for a desired number of decimation steps. We apply our proposed NBP-D decoder to an LDPC code from the CCSDS standard and demonstrate a performance within 1 dB from ML decoding.

* 신뢰성↓ VN값 ⇒ ±∞. vs pruning

「신뢰성↓

check node value ⇒ 0

list-based

neural network = learned' decimation stage

decimation stage. 어떤 값으로 decimate할지 결정」

2. PRELIMINARIES

Consider a linear block code $\mathcal{C}$ of length n and dimension k with parity-check matrix $\boldsymbol{H}$ of size $m \times n$, $m \geq$

This work was partially funded by the Swedish Research Council under grant 2016-04253 and the Knut and Alice Wallenberg Foundation under grant 2018.0090.

「gain」

NBP > BP

(reduce iteration수↓)

「list-based decimation stage

↓「codeword candidate: list」

learned decimation stage」

NN(neural network) 어떤 값으로 decimate?

Highlight : before watching video.

Red box : after watching video.

Bibliographic Management Tools

After deleting the example image below, take a screen capture of a list (at least 3 research papers) of articles you are reading or have read in one of the softwares (your preferred software choice) to organize your bibliography.