Performance on Autoencoder-Based MIMO Quantize-Forward Relay System for Various Learning Parameters

Juin Shin
Division of Electronic Engineering
Jeonbuk National University
Jeonju, Korea
juin0128@gmail.com

Yifan Yao
Division of Electronic Engineering
Jeonbuk National University
Jeonju, Korea
yaoyifan1995@gmail.com

Xianglan Jin
Division of Electronic Engineering
Jeonbuk National University
Jeonju, Korea
jinxl77@jbnu.ac.kr

***Abstract*—In multiple input multiple output (MIMO) quantize-forward (QF) relay systems, an autoencoder comprising an encoder, a decoder, and a channel component has been employed, demonstrating commendable performance. In the QF relaying, the relay quantizes the phases of received signals and forwards them to the destination. A neural network is subsequently integrated into the relay after quantization, introducing a non-linear beamforming effect. In assessing the efficacy of the autoencoder-based MIMO QF relay system applying phase quantization with neural network at the relay, we conduct a comprehensive analysis of bit error rates. This evaluation compares system performance related to diverse learning parameters, such as batch size, number of epochs, and neural network size at the relay. Simulation results clearly illustrate that these learning parameters significantly influence the overall performance of the system.**

***Index Terms*—Autoencoder, deep learning, multi-input multi-output (MIMO), quantize-forward, relay**

I. Introduction

Recently, deep learning [1] has found widespread applications across various research fields. In communications, deep learning has been leveraged to enhance system performance and computational efficiency [2]–[5]. An impact application of deep learning in communication systems is the implementation of autoencoders [5]. This approach views the communication system as an end-to-end autoencoder, optimizing both the neural-network transmitter and the neural-network receiver jointly. Notably, this method is advantageous for channel models lacking known optimal solutions. Extending this study to multiple input multiple output (MIMO) systems has demonstrated superior performance compared to conventional MIMO systems [6].

Notable advancements have been achieved in relay cooperative communication systems, where a relay node supports communication between a transmitter (source) and a receiver (destination) [7]. An amplify–forward (AF) relaying, a simplest relay algorithm, amplifies the received signal at the relay and forwards it to the destination. However, the AF relaying introduces noise amplification along with the signal, and the analog nature of received signals in a half-duplex AF relay demands high memory for storing continuous signals before forwarding. To address these challenges, the quantize–forward (QF) algorithm has been proposed as an alternative relay method [8]. The QF relaying quantizes the received signal at the relay and then transmits the quantized signal to the destination. Consequently, the QF relaying with reduced memory requirements, achieves performance comparable to AF relaying.

This paper focuses on the autoencoder-based QF relay system with phase quantization (PQ) at the relay, incorporating an encoder (source), a decoder (destination), and a channel consisting of a source-destination (SD) channel and a source-relay-destination (SRD) channel [9]–[11]. The introduction of a neural network after quantization at the relay results in a non-linear beamforming effect, forming an algorithm named PQ with neural network (PQNN) [11]. The encoder, decoder, and the neural network at the relay are jointly optimized. This paper evaluates the system performance using PQNN at the relay across various learning parameters, including batch size, number of epochs, and the neural network size at the relay, through simulations. Simulation results unequivocally show that these learning parameters have a significant impact on the overall performance of the system.

II. A Relay Communication System

As shown in Fig. 1, a relay communication system including a source (S), a destination (D), and a relay (R) is considered. In this system, the source, the destination, and the relay are equipped with $N_{\mathrm{S}}, N_{\mathrm{D}}$, and N_{R} antennas, respectively. The links connecting each node are categorized as source-relay (SR), SD, and relay-destination (RD) links, respectively.

In the first time slot, the source broadcasts a symbol vector $\mathrm{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{N_{\mathrm{S}}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{C}^{N_{\mathrm{S}}}$ with $\|\mathrm{x}\|^2 = 1$ to the relay and the destination. Accordingly, the received signals at

This research has been supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant 2022R1F1A1063048.

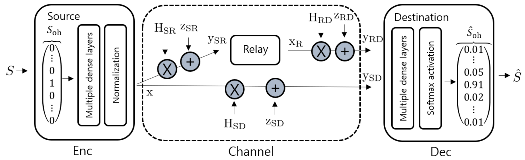


Fig. 1. An autoencoder-based relay system

the relay and the destination are expressed as

$$\mathrm{y}_{\mathrm{SR}} = \mathrm{H}_{\mathrm{SR}}\mathrm{x} + \mathrm{z}_{\mathrm{SR}} \tag{1}$$

$$\mathrm{y}_{\mathrm{SD}} = \mathrm{H}_{\mathrm{SD}}\mathrm{x} + \mathrm{z}_{\mathrm{SD}} \tag{2}$$

where $\mathrm{H}_{\mathrm{SR}} \in \mathbb{C}^{N_{\mathrm{R}} \times N_{\mathrm{S}}}$ and $\mathrm{H}_{\mathrm{SD}} \in \mathbb{C}^{N_{\mathrm{D}} \times N_{\mathrm{S}}}$ are channel coefficient matrices for the SR and the SD links, respectively. $\mathrm{z}_{\mathrm{SR}} \sim \mathcal{CN}(0, \sigma^2 I_{N_{\mathrm{R}}})$ and $\mathrm{z}_{\mathrm{SD}} \sim \mathcal{CN}(0, \sigma^2 I_{N_{\mathrm{D}}})$ are the noise vectors at the relay and the destination, respectively.

In the second time slot, the received signal y_{SR} at the relay is reconstructed into a new signal $\mathrm{x}_{\mathrm{R}} = f(\mathrm{y}_{\mathrm{SR}})$ with $\|\mathrm{x}_{\mathrm{R}}\|^2 = 1$ following the relay algorithm, and then the signal is transmitted to the destination. As a result, the received signal at the destination is

$$\mathrm{y}_{\mathrm{RD}} = \mathrm{H}_{\mathrm{RD}}\mathrm{x}_{\mathrm{R}} + \mathrm{z}_{\mathrm{RD}} \tag{3}$$

where $\mathrm{H}_{\mathrm{RD}} \in \mathbb{C}^{N_{\mathrm{D}} \times N_{\mathrm{R}}}$is the RD channel coefficient matrix. $\mathrm{z}_{\mathrm{RD}} \sim \mathcal{CN}(0, \sigma^2 I_{N_{\mathrm{D}}})$ is a noise vector at the destination. The destination estimates the signal x using the two received signals y_{SD} and y_{RD}.

III. Autoencoder-Based QF Relay System

A. Autoencoder

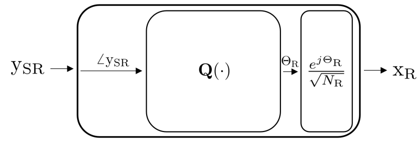
An autoencoder comprises an encoder and a decoder. The encoder reduces the input dimension, producing a compressed value, which the decoder uses to reconstruct the input. This autoencoder structure closely resembles that of a conventional communication system. Consequently, the transmitter and receiver of the communication system are optimized as an autoencoder in an end-to-end fashion. In this setup, the encoder generates a transmission signal corresponding to the information message, while the decoder reconstructs and estimates the information message from the received signal.

The autoencoder-based MIMO relay communication system illustrated in Fig. 1 consists of an encoder (source), a decoder (destination), and a channel.

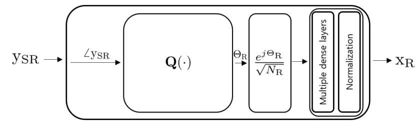
In the transmitter, a message $s \in \mathcal{M} = \{1, 2, \ldots, M^{N_{\mathrm{S}}}\}$ is encoded to a $M^{N_{\mathrm{S}}} \times 1$ one-hot vector S_{oh}, passing through two dense layers and a normalization layer. After the normalization layer, an encoded symbol vector x is generated.

Referring to the system described in Section II, the source broadcasts x to the relay and the destination in the first time slot, with the relay transmitting x_{R} to the destination in the second time slot. Consequently, the destination receives signals y_{SD} and y_{RD} over two time slots.

At the destination, the two received signals pass through two dense layers and a softmax activation function. This network estimates the original message by producing an output $\hat{S}_{\mathrm{oh}}$.



(a) PQ



(b) PQNN

Fig. 2. The PQ and PQNN algorithms at the relay

Within the elements of $\hat{S}_{\mathrm{oh}}$, the index corresponding to the element with the largest value is determined as an estimation message $\hat{s}$. To reduce the difference between the transmitted message s and the estimated message $\hat{s}$, we use the categorical cross-entropy loss function between $\hat{S}_{\mathrm{oh}}$ and S_{oh} as

$$L = -\sum_{i=1}^{M} [S_{\mathrm{oh}}]_i \log([\hat{S}_{\mathrm{oh}}]_i).$$

B. QF algorithm

The AF strategy exhibits commendable performance, but the demand for memory at the relay is substantially high due to the storage of the analog signal. Conversely, the QF approach quantizes the phase of the signal received at the relay, effectively utilizing a limited amount of memory. Additionally, unlike AF, the QF method eliminates the necessity for channel information H_{SR} of the SR link, thereby alleviating the computational burden at the relay. For specific details we first explain the phase quantization (PQ) and subsequently introduce PQ with neural network (PQNN).

Applying a $q(\geq \log_2 M)$-bit uniform phase quantization [8], the result for the quantization is $[\Theta_{\mathrm{R}}]_i = \mathbf{Q}(\angle[\mathrm{y}_{\mathrm{SR}}]_i) = \phi_k = \dfrac{2\pi k}{2^q}$, when $\frac{(2k-1)\pi}{2^q} < \angle[\mathrm{y}_{\mathrm{SR}}]_i \leq \frac{(2k+1)\pi}{2^q}$ for $k = 0, 1, \ldots, 2^q$. Then, x_{R} in (3) is denoted as

$$\mathrm{x}_{\mathrm{R}} = \frac{1}{\sqrt{N_{\mathrm{R}}}} e^{j\Theta_{\mathrm{R}}} = \frac{1}{\sqrt{N_{\mathrm{R}}}} e^{j\mathbf{Q}(\angle \mathrm{y}_{\mathrm{SR}})} \tag{4}$$

where $\frac{1}{\sqrt{N_{\mathrm{R}}}}$ arises from the power constraint $\|\mathrm{x}_{\mathrm{R}}\|^2 = 1$. This is referred to as PQ, and it has been introduced for both the SISO system [9] and the MIMO case [10].

If the relay possesses the capability to accommodate additional computational load, further enhancements to the system performance can be achieved by introducing an additional neural network, particularly one that includes a hidden layer, after quantization. This extended approach is termed PQNN [11]. The PQ and PQNN algorithms are succinctly depicted in Fig. 2.

IV. Simulation Result and Discussion

In this section, we evaluate the system performance for the autoencoder-based QF relay system applying the PQNN at

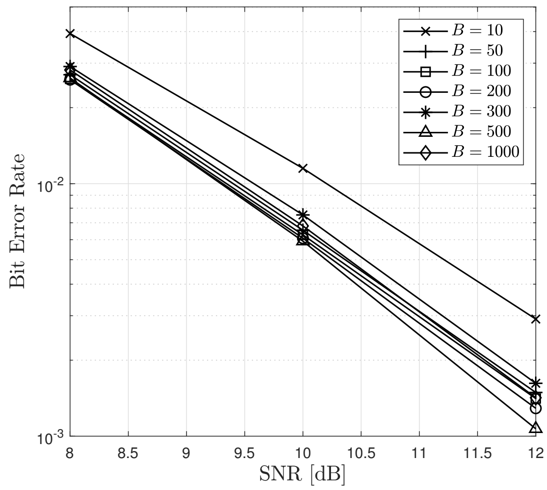


Fig. 3. BER performance for various batch sizes.

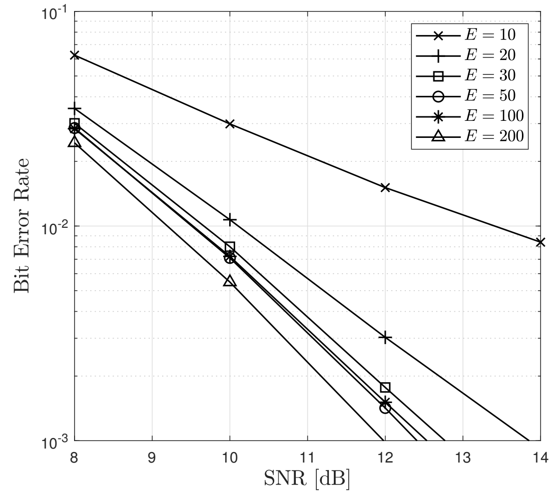


Fig. 4. BER performance for various numbers of the epochs.

the relay, where $N_{\mathrm{S}} = N_{\mathrm{R}} = N_{\mathrm{D}} = 4, M = 4$, and $q = \log_2 M + 2$. We explore various learning parameters, including the batch size B, the number of epochs E, and the number of nodes in the hidden layer of the neural network at the relay, through simulations.

For training, we employ Adam optimizer [12] on TensorFlow frameworks [13]. Unless explicitly specified, the learning parameters for the simulations are set as follows: the number of training data $N = 10000$, the number of test data $N_{\mathrm{test}} = 1000$, the batch size $B = 300$, the number of epochs $E = 100$, and the number of the nodes for the hidden layer at the relay $= 8N_{\mathrm{R}}$. The simulation process is conducted repeatedly for 1000 Rayleigh-distributed fading channel scenarios to calculate the average bit error rate (BER).

A. Batch size

The batch size is a critical parameter in the training phase, representing the number of data samples processed in parallel during each training iteration. A too-small batch size allows for a detailed examination of individual data samples during training, but it may also lead to significant deviations from the optimal optimization path during the weight update process. Conversely, an excessively large batch size increases the number of simultaneously handled data, reducing the frequency of weight updates. While this can shorten the training time, it may result in large memory usage and performance degradation due to fewer opportunities for weight updates. Determining the appropriate batch size is crucial as it significantly impacts memory usage, training time, and performance.

To determine the optimal batch size, a performance comparison is conducted across various batch sizes, as illustrated in Fig. 3. Notably, the model with a batch size of $B = 10$ stands out as the most suboptimal, exhibiting poor performance by a substantial margin compared to other batch sizes, which tend to cluster together with relatively similar outcomes.

The choice of batch size is a trade-off between memory usage and training time. Larger batch sizes increase memory usage during training but reduce the overall training time due to fewer training iterations. Conversely, smaller batch sizes necessitate a longer training time due to a greater number of training iterations. Given these characteristics, it is advisable to choose a batch size $B \geq 50$ for better model performance.

B. Number of epochs

Epochs represent the number of training iterations completed on a given training dataset and constitute a crucial parameter in the training phase, alongside the batch size. If the number of epochs is excessively large, it can result in long training times, while too few epochs may hinder convergence to a stable stage.

In Fig. 4, a comparison of BERs across various numbers of epochs is presented. Generally, an increase in the number of epochs tends to improve performance, with the exception of the case $E = 50$. Interestingly, the performance at $E = 50$ shows slightly better performance than the case with $E = 100$. However, this observation does not alter the broader trend of performance improvement with an increased number of epochs. From the figure, it is evident that the system requires more than 30 epochs for satisfactory performance. Considering the trade-off between training time and performance, an appropriate number of epochs ($E \geq 30$) can be chosen to ensure convergence to a stable stage without excessively extending the training duration.

C. Number of nodes in the hidden layer

This simulation explores the influence of the number of nodes in the hidden layer for the PQNN algorithm. Considering the real and imaginary parts for N_{R} received signals at the relay, y_{SR}, the sizes the input layer and the output layer of the neural network at the relay are $2N_{\mathrm{R}}$. Therefore, the number of nodes in the hidden layer can be set as a value proportional to $2N_{\mathrm{R}}$.

In Fig. 5, the BER performance is depicted for cases where the number of nodes are multiples of $2N_{\mathrm{R}}$. The graph reveals

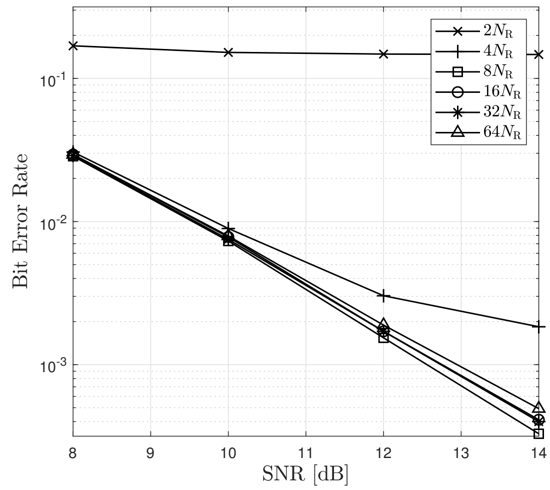


Fig. 5. BER performance for various numbers of nodes for the hidden layer.

that there is no significant difference in performance among cases with sizes $8N_\mathrm{R}$, $16N_\mathrm{R}$, $32N_\mathrm{R}$, and $64N_\mathrm{R}$. However, $2N_\mathrm{R}$ exhibits inadequate performance, and $4N_\mathrm{R}$ also shows unsatisfactory performance. Therefore, it appears that the number of nodes in the hidden layer needs to reach a certain level, $8N_\mathrm{R}$ or higher to achieve satisfactory performance.

V. Conclusion

In conclusion, our study focuses on enhancing the performance of the autoencoder-based MIMO QF relay communication system by incorporating the PQNN at the relay. We systematically evaluate the system performance under varying learning parameters, including the batch size, the number of epochs, and the neural network size at the relay. The simulation results explicitly demonstrate that these learning parameters have a significant impact on the system's overall performance. This observation underlines the importance of selecting these learning parameters to achieve satisfactory results in deep learning-based communication systems.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[2] C. K. Wen, S. Jin, K. K. Wong, J. C. Chen, and P. Ting, "Channel estimation for massive MIMO using gaussian-mixture bayesian learning," IEEE Trans. Wireless Commun., vol. 14, no. 3, pp. 1356–1368, Mar. 2015.

[3] D. Neumann, T. Wiese and W. Utschick, ''Learning the MMSE channel estimator," IEEE Trans. Signal Process., vol. 66, no. 11, pp. 2905-2917, Jun. 2018.

[4] N. Samuel, T. Diskin and A. Wiesel, ''Learning to detect," IEEE Trans. Signal Process., vol. 67, no. 10, pp. 2554-2564, May. 2019.

[5] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," IEEE Trans. Cogn. Commun. Netw., vol. 3, no. 4, pp. 563-575, Dec. 2017.

[6] T. J. O'Shea, T. Erpek and T. C. Clancy, "Deep learning-based MIMO communications," July 2017. [Online]. Available: arXiv:1707.07980

[7] J. N. Laneman, D. N. C. Tse and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," IEEE Trans. Inf. Theory, vol. 50, no. 12, pp. 3062 – 3080, Dec. 2004.

[8] M. R. Souryal and H. You, "Quantize-and-forward relaying with M-ary phase shift keying," in Proc IEEE Wireless Commun. Netw. Conf., 2008, pp. 42-47.

[9] J. Shin and X. Jin. 'An application of autoencoder in the quantize-and-forward relay system," Summer Conference on KICS (the Korean Institute of Communications and Information Sciences), Jeju, Korea, Jun. 2022, pp. 1781 – 1782.

[10] J. Shin and X. Jin, "Quantize-and-forward relay system with autoencoder using multiple antennas," in Proc IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2022, pp. 1-3.

[11] J. Shin, "Autoencoder-based MIMO quantize-forward relay system," M.S. thesis, Jeonbuk National University, 2023.

[12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[13] 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 distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467