Quantum Federated Learning for Wireless Communications

Bhaskara Narottama* and Soo Young Shin†
Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea
e-mail: {*bhaskara,†wdragon}@kumoh.ac.kr

***Abstract*—This initial work presents a quantum federated learning (QFL) architecture for wireless communications. Different from centralized learning, federated learning offloads the learning tasks into local computation units to improve privacy. The presented QLF is expected to reduce complexity by utilizing quantum computation. The presented two-tier QLF consists of QNN operations in the access points and the cloud.**

***Index Terms*—Quantum federated learning, Quantum neural networks, Wireless communications**

I. Introduction

Future wireless communications is expected to utilize machine learning for various tasks, such as optimization [1]. Centralized learning, which uses a central processing unit, has several drawbacks such as security issue [2]. Federated learning aims to tackle these issues by distributing the burden of learning into edge computation units [3]. That is, instead of uploading all of the training data to the central processing unit, the data is kept in the edge devices, thus enhancing data privacy. Moreover, QNN gains research attentions amid performance gain offered by quantum computations [4], [5]. Motivated by these factors, this study inspects the usage of QFL for wireless communications, which utilizes multiple QNNs. *The presented QFL can be used for general optimization in wireless (such as optimizing power allocation in non-orthogonal multiple access [6]).*

II. Proposed Quantum Federated Learning

The operation of two-tier QNN circuits in QFL is presented in Fig. 1. Consider a wireless network where users are grouped into N_G groups. Each group, consists of $U_{m,1}$ and $U_{m,2}$ in particular, is served by an access point [1] .

Each access point $E_m, \forall m \in \{1, \ldots, N_G\}$, receives channel information of $U_{m,1}$ and $U_{m,2}$ ($\tilde{h}_{m,1}$ and $\tilde{h}_{m,2}$, respectively) from pilot signals. Subsequently, each $E_m, \forall m \in \{1, \ldots, N_G\}$, predicts the *solution* for G_m using a particular *shallow* QNN, Q_m^{local}. The local prediction of G_m, which is a set of optimization solutions in G_m, is denoted as $\boldsymbol{s}_m^{\text{local}} = \{\lambda_{m,1}, \lambda_{m,2}\}$, where $\lambda_{m,k}$ is the optimization value for $U_{m,k}$. *This study refers "solution" as a general term for wireless optimization* [2] . Subsequently, the cloud processing unit takes $\boldsymbol{s}_m^{\text{local}}$, $\forall m \in \{1, \ldots, N_G\}$, as an input for shared QNN (i.e., Q^{shared}). The output of Q^{shared} is used to optimize $E_m, \forall m \in \{1, \ldots, N_G\}$.

[1]This study refers to each user as $U_{m,k}$, where m and k indicates group and user indexes, respectively.

[2]In particular, it can be the transmit power level in NOMA [6].

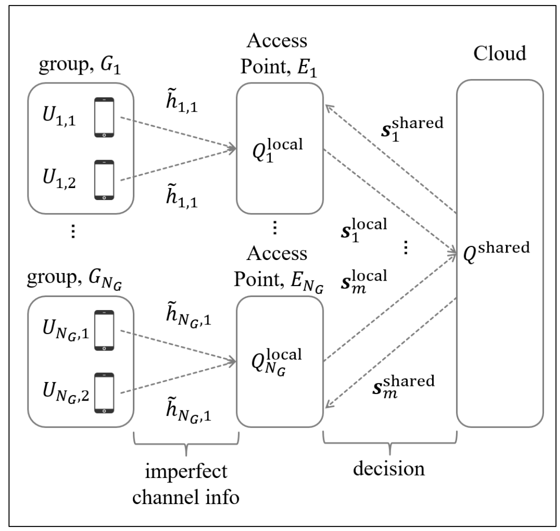

Figure 1: QFL for wireless optimization, which consists of two-tier operations: access point QNN (Q_m^{local}, $m \in \{1, \ldots, N_G\}$) and cloud QNN (Q^{shared}).

A. Local QNN

Figure 2 presents the circuit design for the local QNN. *This study combines the multi-layer QNN design of [8]* [3] *with activation function of [4] for local QNN* . The local QNN, which is computed by E_m to predict $\boldsymbol{s}_m^{\text{local}}$ based on $\tilde{\boldsymbol{h}}_m$, is defined as

$$Q_m^{\text{local}} = \underbrace{\text{CCX}(ancilla|q_1, q_2)}_{\text{"activation function"}} \underbrace{\text{X}^{\otimes N_G}}_{\text{"neuron connection"}} \text{H}^{\otimes N_G} U^{\text{encode}} U^{\text{local}} \text{H}^{\otimes N_G}, \tag{1}$$

where U^{encoding} and U^{local} are *encoding* [7] and unitary operation of shallow QNN.

Definition 1 (Shallow QNN)**.** The unitary operations of shallow QNN is defined as

$$U^{\text{local}} \triangleq \underbrace{\left(\text{R}_z(\theta_{m,k}^{\text{local}})\right)^{\otimes N_q^{\text{local}}}}_{\text{"weight"}} \text{CZ}(q_2|q_1), \tag{2}$$

where $\text{CCX}(ancilla|q_1, q_2)$ indicates X-axis rotation (applied on an *ancilla* qubit, controlled by q_1, q_2) and $\text{CZ}(q_2|q_1)$ indicates Z-axis rotation (applied on q_2, controlled by q_1).

[3]Notice that QNN neurons in [8] do not utilize activation functions.

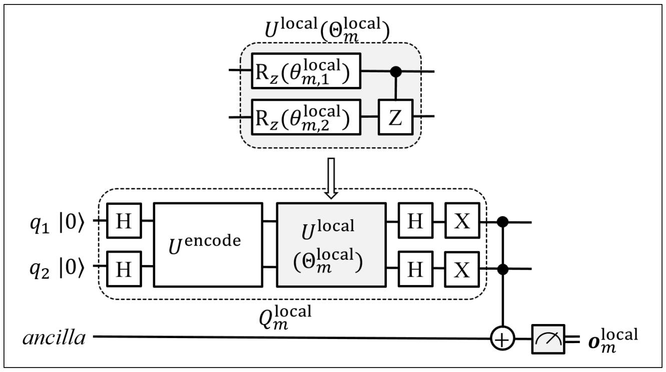

Figure 2: Unitary operation for local QNN. The multi-layer QNN design of [8] is combined with activation function in [4]

B. Shared QNN

Figure 3 presents the circuit design for the shared QNN QNN. *The design of shared QNN in this work is based on the multi-layer QNN design of [8] and activation function of [4].* The shared QNN, which is computed in cloud to predict $\mathbf{s}^{\text{shared}}$ is given as

$$Q^{\text{shared}} = \Big(\bigotimes_{i=((N_G-1)N_l)+1}^{N_G N_l} \text{XH} \Big) U^{\text{shared}} U^{\text{encode}} \text{H}^{\otimes N_G N_l}, \quad (3)$$

where N_l indicates the number of layers. Moreover, U^{shared} denotes the operation of deep QNN, which is defined as follows.

Definition 2 (Deep QNN)**.** The unitary operations of deep QNN is defined as

$$\begin{aligned} U^{\text{shared}} \triangleq & \underbrace{\Big(\prod_{l=1}^{N_l} \text{CX}(q_{N_G l} | q_{N_G (l-1)}) \Big)}_{\text{“inter-layer connection”}} \\ & \underbrace{\Big(\prod_{l=1}^{N_{\text{neuron}}} \text{CZ}(q_{N_G l} | q_{(N_G l)-1}) \Big)}_{\text{“inter-neuron connection”}} \qquad (4) \\ & \underbrace{\Big(\bigotimes_{l=1}^{N_l} \bigotimes_{m=1}^{N_G} \text{R}_z(\theta_{l,m}^{\text{local}}) \Big)}_{\text{“weight”}}, \end{aligned}$$

where $\text{CX}(q_{N_G l} | q_{N_G (l-1)})$ indicates X-axis rotation (applied on $q_{N_G l}$, controlled by $q_{N_G (l-1)}$).

III. Future Works

This initial work presents a QLF architecture for optimizing wireless communications. For future work, complexity can be analyzed. Supporting operations, e.g., encoding and decoding, will be further studied.

Acknowledgment

This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003).

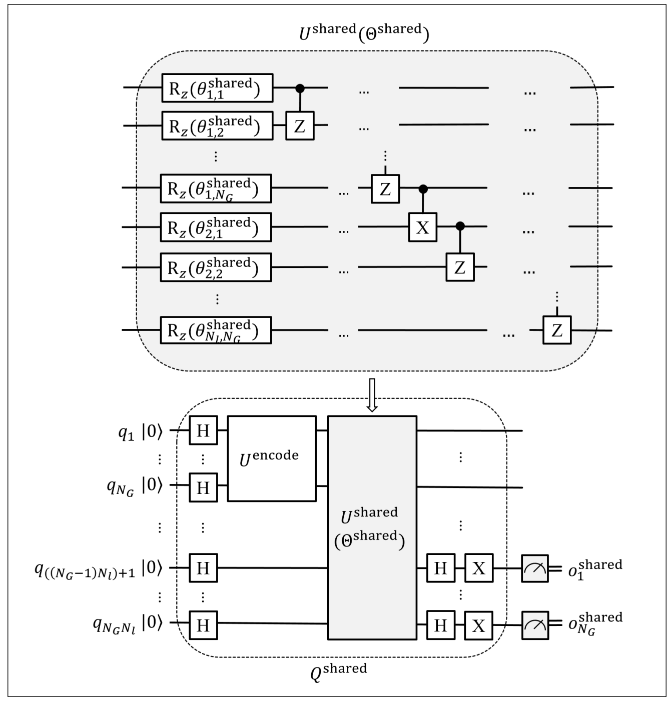

Figure 3: Unitary operation for shared QNN. Similar to local QNN, the multi-layer QNN design in [8] and activation function in [4] is utilized.

References

[1] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, “Ten challenges in advancing machine learning technologies toward 6G,” *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 96-103, Jun 2020.

[2] B. Brik, A. Ksentini, and M. Bouaziz, “Federated learning for UAVs-enabled wireless networks: use cases, challenges, and open problems,” *IEEE Access*, vol. 8, pp. 53841-53849, 2020.

[3] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, “Federated learning over wireless networks: optimization model design and analysis,” *Proc. IEEE Conf. Comp. Commun.*, Paris, France, 2019.

[4] F. Tacchino, C. Macchiavello, D. Gerace, *et al.*, “An artificial neuron implemented on an actual quantum processor,” *npj Quantum Inf.*, vol. 5, no. 26, Mar. 2019.

[5] K. Beer, D. Bondarenko, T. Farrelly, *et al.*, “Training deep quantum neural networks,” *Nature Commun.*, vol. 11, no. 808, Feb. 2020.

[6] B. Makki, K. Chitti, A. Behravan, and M. Alouini, “A survey of NOMA: current status and open research challenges,” *IEEE Open J. Commun. Soc.*, vol. 1, pp. 179-189, 2020.

[7] R. LaRose and B. Coyle, “Robust data encodings for quantum classifiers,” 2020, arXiv:2003.01695v1. [Online]. Available: https://arxiv.org/abs/2003.01695

[8] M. Watabe, K. Shiba, M. Sogabe, K. Sakamoto, and T. Sogabe, “Quantum circuit parameters learning with gradient descent using backpropagation,” 2019, arXiv:1910.14266v3. [Online]. Available: https://arxiv.org/abs/1910.14266