Quantum Federative Learning

Project Report

A.Goultiaev Tolstokorov - 18328823

1 Abstract

This report outlines a demonstration of federated learning using a hybrid quantum neural net to classify images of planes and cars from the CIFAR10 dataset. The approach used for the neural net is a hybrid, combining a pre-trained ResNet18 neural network with a quantum variational encoder. The project proves the applicability of federated learning for quantum machine learning and argues for its usefulness in the field.

2 Introduction

With the recent rise in popularity of machine learning (ML) and specifically deep learning (DL) the field has expanded massively to different applications. From speech-recognition[1, 2] and computer vision[3, 4] to finance[5] and playing video and board games[6, 7]. Recently thanks to companies such as IBM, Google, IonQ, D-Wave and software development kits such as Qiskit[8] and Pennylane[9], Quantum is being brought to the public en mass. Quantum computers can theoretically achieve exponential speedup in certain cases that conventional computers struggle with[10]. With this it was no surprise that ML and DL were applied on Quantum Computers with high degree of success rate and sometimes even performing better than their conventional counterparts. However, some applications require access to sensitive data such as medical imaging, text prediction or speech recognition which is to be protected from third parties or bad actors. In conventional ML and DL one of the surging solutions for this challenge is federative learning (FL)[11]. Some research has been conducted in quantum federative computing (QFC) with hybrid conventional-quantum machine learning systems[12] and fully quantum ones[13]. In this report I build and test a hybrid classical-quantum neural network to classify images of automobiles and aeroplanes from the CIFAR10[14] dataset. Then I train the built system in a federated way using a FL framework called Flower[15] and compare it to the conventionally trained network.

3 Federated Learning

As outlined before in the Introduction section, FL is an emerging solution to data privacy concerns in large scale deployments of deep learning where sensitive data from many clients is used to train the deep learning model. The main premise of FL is that each client device computes computations and updates to its own local model of the neural network and only sends its weight updates to the central server or body which then uses some strategy to aggregate those updates from the clients and construct the global model. This way no one body has access to the training data and theoretically only the client itself stores its own data with the server only having access to the model updates. The most general FL strategy is called FedAvg and is the one I will be employing in my implementation.

3.1 FedAvg

In FedAvg each training round the server will call a subset $\mathcal{C}$ clients out of all the $\mathcal{N}$ clients, these clients will then compute the model updates θ from training with batch size $\mathcal{B}$ for $\mathcal{E}$ epochs. Next the server will average out the θ updates from all the called clients and update its global model which will then send its updated weights to the next $\mathcal{C}$ subset of clients for the next round and repeat that same process.

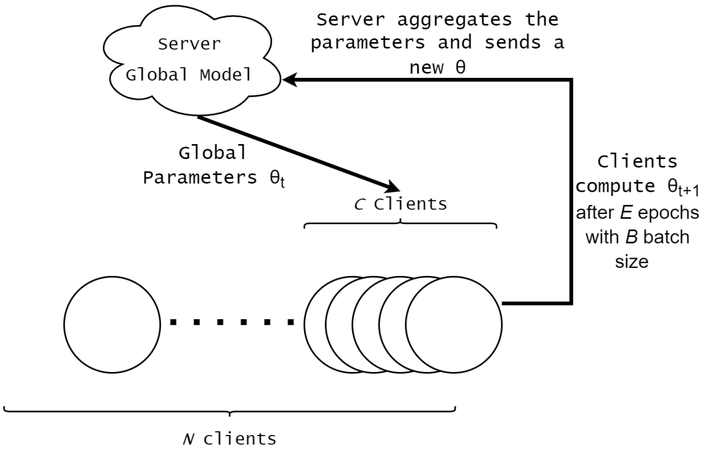

Server
Global Model
Server aggregates the parameters and sends a new θ
Global Parameters θt
C Clients
Clients compute θt+1 after E epochs with B batch size
N clients

Figure 1: *Federated Model.*

4 Quantum Machine Learning

Quantum neural networks are a special type of quantum circuits with adjustable parameters that can be optimized with machine learning. The most widely implemented circuit for this purpose is a variational quantum circuit (QVC), this circuit consists of an encoding routine of the classical data $E(x)$ followed by a variational circuit block $W(\theta)$ with learnable parameters ϕ.

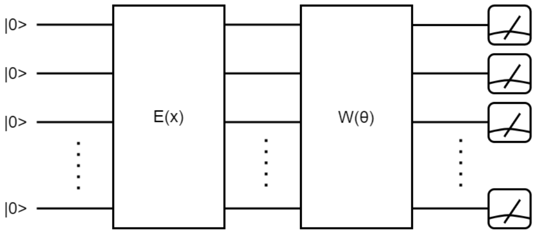
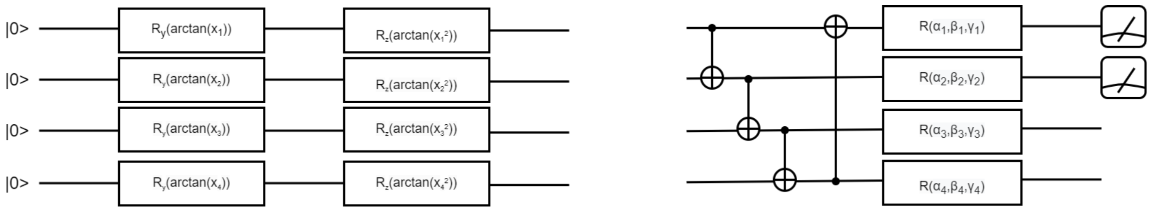


Figure 2: *Structure of a variational quantum circuit.*

In Figure 3 the QVC that I will be using is shown. The encoder in my QVC consists of a series of single-qubit gates R_y and R_z which are rotations on the y-axis and z-axis respectively. These



(a) Encoder. (b) Variational layer.

Figure 3: Encoder and variational layer of a 4-qubit variational quantum classifier.

rotation gates depend on the input values x_i and therefore encode the data with respect to the input. The variational layer consists of CNOT gates that entangle each neighbouring pair of qubits and single qubit unitary gates $R(\alpha, \beta, \gamma)$ that have parameters α, β and γ that are subject to iterative optimization. The quantum measurement at the end ill output the Pauli-Z expected values of those qubits, in our case we will be performing binary classification between images of cars or planes so we only measure the first two qubits.

4.1 Hybrid Model

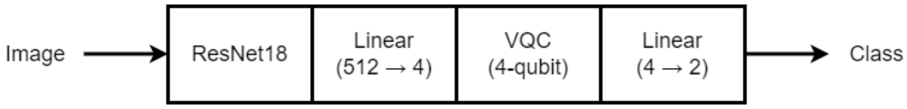


Figure 4: *Structure of a hybrid model.*

A classical-quantum hybrid model is essentially a classical neural net with some of its layers replaced with a quantum circuit. In my case I am dealing with image classification and thus will use a Convolutional Neural Network (CNN), specifically a pretrained network called ResNet18 [16]. In this CNN the last classifying layers will be replaced by the QVC shown in Figure 3 and the quantum measurement will be the readout of the model. The classical neural net then acts to reduce the tensor of the input image into 4 features as our QVC is 4-qubit.

5 Methodology

The process to build and implement the target hybrid network followed these steps:

I. Build the hybrid neural network following structure shown in Fig4 and using the QVC shown in Fig3.

II. Train and evaluate its performance on the CIFAR10 dataset classifying between planes and cars.

III. Train and evaluate the model in a federated setting using the Flower framework.

5.1 Part I

Following the outlined structure the hybrid neural network was written using PyTorch[17]. The CIFAR10 dataset had to be filtered as we were interested only in the images belonging to the classes car or plane, this left me with 10 thousand images for training and 2 thousand for testing. Finally the QVC was coded to be 6 deep (i.e. the variational layer in the QVC would repeat 6 times before the readout). These parameters were chosen based on existing implementations of such hybrid networks.

5.2 Part II

5.3 Part III

6 Discussion

6.1 Results

6.2 Conclusion

References

[1] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: an overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603, 2013.

[2] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE access*, 7:19143–19165, 2019.

[3] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. In *Science and information conference*, pages 128–144. Springer, 2019.

[4] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[5] JB Heaton, Nicholas G Polson, and Jan Hendrik Witte. Deep learning in finance. *arXiv preprint arXiv:1602.06561*, 2016.

[6] Niels Justesen, Philip Bontrager, Julian Togelius, and Sebastian Risi. Deep learning for video game playing. *IEEE Transactions on Games*, 12(1):1–20, 2019.

[7] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[8] A tA v, MD SAJID ANIS, Abby-Mitchell, Héctor Abraham, AduOffei, Rochisha Agarwal, Gabriele Agliardi, Merav Aharoni, Vishnu Ajith, Ismail Yunus Akhalwaya, Gadi Aleksandrowicz, and many others. Qiskit: An open-source framework for quantum computing, 2021.

[9] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[10] Aram W Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, 2017.

[11] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[12] Samuel Yen-Chi Chen and Shinjae Yoo. Federated quantum machine learning. *Entropy*, 23(4):460, 2021.

[13] Mahdi Chehimi and Walid Saad. Quantum federated learning with quantum data. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8617–8621. IEEE, 2022.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.