

Evaluating Accuracy and Adversarial Robustness of Quanvolutional Neural Networks

Korn Sooksatra ^{*}, Pablo Rivas [†], *Senior, IEEE*, Javier Orduz [‡]
School of Engineering and Computer Science
Department of Computer Science, Baylor University
Email: {*Korn_Sooksatra1, †Pablo_Rivas, ‡Javier_OrduzDucuaara}@Baylor.edu

Abstract—Machine learning can thrust technological advances and benefit different application areas. Further, with the rise of quantum computing, machine learning algorithms have begun to be implemented in a quantum environment; this is now referred to as quantum machine learning. There are several attempts to implement deep learning in quantum computers. Nevertheless, they were not entirely successful. Then, a convolutional neural network (CNN) combined with an additional quanvolutional layer was discovered and called a quanvolutional neural network (QNN). A QNN has shown a higher performance over a classical CNN. As a result, QNNs could achieve better accuracy and loss values than the classical ones and show their robustness against adversarial examples generated from their classical versions. This work aims to evaluate the accuracy, loss values, and adversarial robustness of QNNs compared to CNNs.

Index Terms—quanvolutional neural networks, quantum neural networks, convolutional neural networks

I. INTRODUCTION

Today, areas such as computer visions, natural language processing and autonomous cars have significant improvements thanks to the rise of deep learning (DL). However, DL performance is still unsatisfying in some applications, and some works explored its accuracy. Further, some examples show the DL vulnerabilities in adversarial examples [1], [2]. Thus, we aim to address those problems of deep learning.

After the rise of quantum computing (QC), Schuld *et al.* [3] implemented machine learning algorithms in QC context and introduced the new area, a.k.a. quantum machine learning. There are several existing works attempting to create deep learning models in quantum computing [4]–[7]. In particular, Henderson *et al.* [8] designed hybrid neural networks composed of a quanvolutional layer and classical layers and called them quanvolutional neural networks (and related contributions can be found in [9]).

Inspired by the idea in [8], we desire to evaluate if a QNN is more accurate and more adversarially robust than classical-CNN. Therefore, the contributions of this report can be summarized as follows:

- We discuss the idea of QNN and show our QNN layout focused on describing a quanvolutional layer.
- We extensively construct experiments to evaluate the performance and robustness of QNNs.

This paper is organized as follows: Section II provides a review of convolutional neural networks, adversarial examples and works related to our work; Section III explains how

we designed our quanvolutional neural network and how it functioned; Section IV describes how we constructed our experiments; Section V shows the results obtained from our experiments; finally, Section VI discusses the experimental results and concludes everything.

II. BACKGROUND

This section explains the preliminary knowledge needed for the other sections. First, we provide a background of convolutional neural networks and then discuss the literature of adversarial examples. At last, we briefly describe some works that are relevant to ours.

A. Convolutional neural networks

In general, convolutional neural networks (CNN) are composed of convolutional and dense layers. In particular, a convolutional layer performs a convolution operation on its input; therefore, it can extract a feature indicating spatial correlation among its input’s attributes. Furthermore, each layer can have multiple filters to extract multiple features, and the filters have the same kernel sizes and strides. In addition, a filter can be 1-dimension, 2-dimension or d -dimension where $d \geq 1$ and $d \in \mathbb{Z}$. For example, if an input is a 2-dimension image, a filter is usually 2-dimension to extract a feature for spatial correlation among the image’s pixels. Figure 1 shows an example of CNN where its components are listed as follows: the first layer is a convolutional layer with 8 filters and 64-by-64 kernel size; the second layer is a convolutional layer with 24 filters and 48-by-48 kernel size; the third layer is a convolutional layer with 24 filters and 16-by-16 kernel size; the fourth layer is a dense layer with 256 neurons; the last layer is a dense layer with 128 neurons. In the present, the examples of famous CNNs for images that are widely used are VGG [10], ResNet [11], Inception [12] and Xception [13].

B. Adversarial examples

Adversarial examples are samples that can mislead a target classifier to output predictions that contrast with humans’ perceptions. These examples were initially discovered in [1] and then were further analyzed why they could mislead classifiers in [2]. Moreover, they invented a one-step attack called Fast Gradient Sign Method (FGSM) and adversarial training to defend against the attack. Later, Kurakin *et al.* [14] modified FGSM to be iterative to create a more potent attack. In addition,

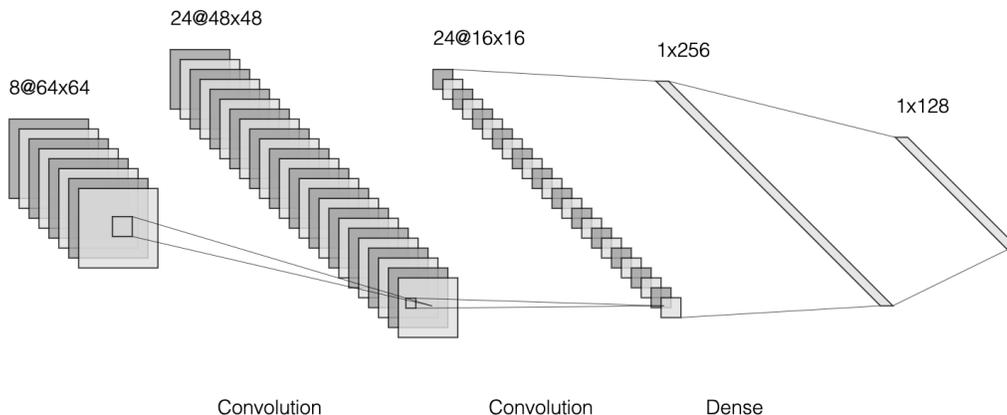


Fig. 1: An example of convolutional neural network that consists of three convolutional layers and two dense layers.

Papernot *et al.* [15] developed an attack altering only a few pixels of an image to create an adversarial example and called it Jacobian-based Saliency Map Attack (JSMA). Essentially, Kurakin *et al.* [16] showed that adversarial examples could be effective in the real world even though a camera could reduce small perturbation; thus, adversarial examples were hazardous in practice. Then, Carlini and Wagner [17] created the strongest adversarial attack by formulating an unconstrained optimization problem and using a gradient approach to solve it. More interestingly, Su *et al.* [18] constructed an adversarial example that perturbed only one pixel by using an evolutionary algorithm.

In our experiments, we utilized the attack from [17] to evaluate learning models in terms of adversarial robustness and called it Carlini and Wagner attack (CWA).

C. Related works

Several works have attempted to develop machine learning models in the quantum computing environment. Further, some tried to combine a quantum machine learning model and a classical one. Also, some works studies on adversarial robustness of quantum machine learning. Therefore, it is helpful to describe some of those works as the following briefly.

1) *Quantum neural networks*: Farhi *et al.* [4] introduced a quantum neural network consisting of a series of unitary operators, and its measurement is a Pauli operator. They applied this network to create a MNIST classifier. However, since they used a classical simulator, their networks could not handle a massive input, e.g., a MNIST input. Thus, they downsampled MNIST images from 28-by-28 images to 4-by-4 images and selected only the data with labels 3 and 6 because their network was simply a binary classifier. Then, Cong *et al.* [6] proposed quantum convolutional neural networks and used them for quantum phase recognition and quantum error correction. Later, Kamruzzaman *et al.* [19] explained quantum deep learning neural networks and described their advantages over classical deep learning neural networks. Oh *et al.* [7] also created a quantum convolutional neural network by imitating the structure of classical ones. Then, they trained it for classifying images in a downscaled MNIST dataset. As a result, the network

could achieve the same accuracy and loss value as its classical counterpart.

2) *Quantum convolutional neural networks*: A QNN is a hybrid network consisting of a quantum convolutional layer and classical layers, which will be explained later. This kind of network was proposed in Henderson *et al.* [8] and was used as a MNIST classifier. Consequently, the network could achieve a better accuracy and loss value than its classical counterpart. However, it is unfair since the QNN had an additional quantum convolutional layer. Hence, QNN had more layers than its classical counterpart. Later, in 2021, Orduz *et al.* [9] proposed a quantum convolutional autoencoder and compared it to its classical counterpart. As a result, the quantum convolutional autoencoder could offer early learning stability in the CIFAR-10 dataset [20].

3) *Adversarial robustness*: Liu and Wittek [21] recently discussed the trade-off between security and quantum advantage because when input was high-dimensional, a quantum classifier was significantly vulnerable to adversarial examples. Then, Lu *et al.* [22] analyzed adversarial robustness on quantum neural networks and found that they were vulnerable to adversarial examples generated from gradients of the networks. Further, they showed that adversarial examples generated from classical neural networks could be transferred to the quantum neural networks. Guan *et al.* [23] proposed a method to determine robustness bound and developed robustness verification algorithms for quantum classifiers.

III. QUANTUM CONVOLUTIONAL NEURAL NETWORKS (QNN)

These networks were first proposed by [8] and shown that they could improve the performance of classical CNN. In general, a QNN consists of two parts: a quantum convolutional layer and CNN. Specifically, this idea is to add one more quantum convolutional layer as the first layer of CNN, and this layer is simply a quantum circuit that mimics a convolutional layer. One circuit in a quantum convolutional layer is a filter; hence, if we desire to have n filters in the layer, we need n circuits. The number of qubits in each circuit can be determined as

$$k_w \cdot k_h \quad (1)$$

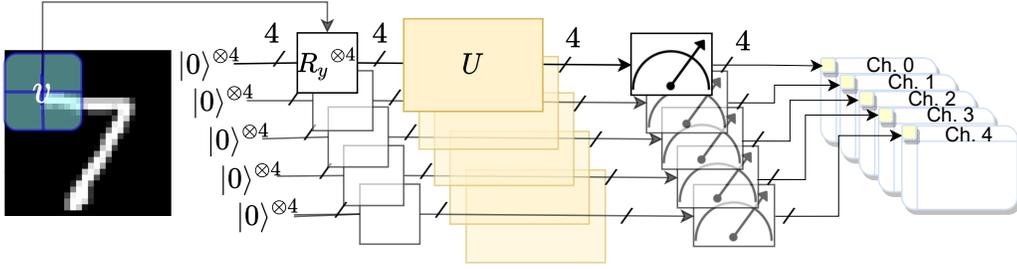


Fig. 2: A filter of quanvolutional layer with 5 filters where the kernel size is 2-by-2, and the output for each channel is the average of all the values after the measurements. Note that R_y has π multiplied by a value v from the input as its argument. The “ \otimes ” means more qubits.

where k_w is the width of the kernel, and k_h is the height of the kernel. For example, if the size of the kernel is 2-by-2, then the number of qubits is 4. Further, in a circuit, the initial states of the qubits are $|0\rangle$, and those states first pass R_y gates whose phases are π multiplied with the values from the kernel in the input. That is,

$$R_y(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad (2)$$

where $\theta = v\pi$, and v is a value from the kernel. Then, all outputs of R_y gates go to a U operator, a random quantum circuit consisting of several one-qubit and two-qubit entangling gates; hence, it can find spatially correlation among attributes of the input by using two-qubit entangling gates. After that, the U operator’s results are measured, and the average is the final output. This method and circuit can be illustrated in Figure 2 and summarized as follows.

- 1) We choose small kernels (2×2) as inputs with respect to strides.
- 2) We apply rotations onto $|0\rangle$, given by R_y operators whose parameters are π multiplied by values from the kernels.
- 3) We apply U , which is a random quantum circuit. It can come from different paradigms of quantum computing, e.g., variational quantum circuit [24] or from universal gates model [25].
- 4) We finally measure and obtain the average.

After the quanvolutional layer, we obtain the output with n channels where n is the number of filters. At last, this output is fed to CNN. Figure 3 shows the outputs after passing a quanvolutional layer with five filters, and each filter may focus on a different part of an image. For instance, according to the figure, the outputs of the quanvolutional layer for the image of 5 have white areas and dark areas in different places. Therefore, as being noticed, it is very similar to a convolutional layer. It simply applies a quantum circuit instead of a convolutional layer.

IV. EXPERIMENTAL DESIGN

We constructed these experiments on Google Colab [26] and used Python version 3 mainly with PennyLane module [27] for implementing quanvolutional layers and Tensorflow module [28] for implementing machine learning models (i.e., CNNs).

We use MNIST dataset [29] as our input, and this dataset is images of digits (i.e., zero to nine). Our models need to classify these images into digits. Fairly, we trained all models for 30 epochs with the same training dataset consisting of 50 images and tested them with the same 30 images. All the models are described as follows:

- 1) **Linear model.** This model has only the output layer, which has ten neurons for the outputs of ten digits.
- 2) **Quantum linear model.** The first layer of this model is a quanvolutional layer, and the other part is the linear model.
- 3) **CNN model.** The first layer of this model is a convolutional layer with 3 filters. Each filter’s kernel size and stride are 3×3 , and the rest is the linear model.
- 4) **Quantum CNN model.** This model is a quanvolutional neural network whose architecture is the same as the CNN model with an additional quanvolutional layer in its front.
- 5) **2-CNN model.** This model has two convolutional layers, and the first one has 5 filters with 3×3 kernel size and 1×1 stride. The rest of the model is the CNN model.

It is worth noting that our quanvolutional layer has 5 filters with 3×3 kernel and 1×1 stride, and we compute the average of the results in each filter as its output. Noticeably, without the first layer, the architecture of the quantum CNN model is the same as the one of the 2-CNN model because we aim to test if a quanvolutional layer can outperform a convolutional layer. All the experiments are listed as follows:

- 1) Comparison concerning accuracy and loss values between 1) the linear model and the quantum linear model, 2) the CNN model and the quantum CNN model and 3) the 2-CNN model and the quantum CNN model during training.
- 2) Evaluation of robustness of all models with adversarial examples generated from the linear model, CNN model and 2-CNN model. Note that we chose Carlini and Wagner attack from ref. [17] to find adversarial examples with 0.01 step size and picked only test samples that a target model correctly classified to generate adversarial examples.

V. RESULTS

This section discusses the results that we obtained from the experiments. In particular, we evaluated the learning models

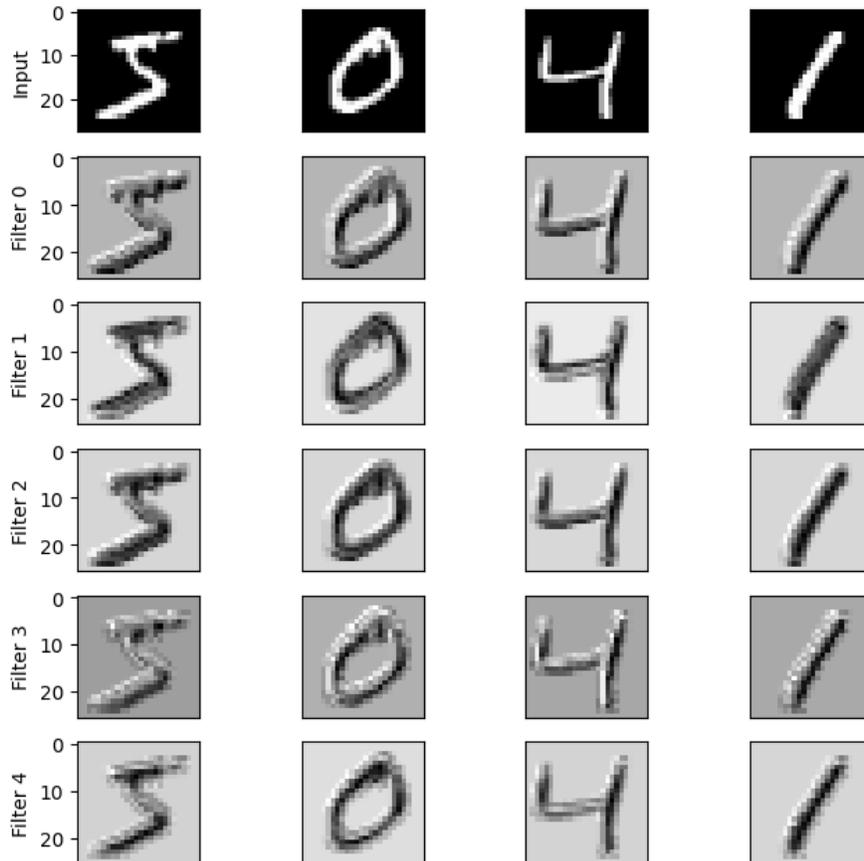


Fig. 3: The outputs of some images in MNIST dataset after passing a quanvolutional layer with 5 filters where the first row is original MNIST images, and the other rows are the outputs with respect to 5 filters.

in terms of three aspects: accuracy, loss value and adversarial robustness.

A. Accuracy and loss value

First, we compared the linear model to the quantum linear model. Then, we found that a linear model with an additional quanvolutional layer achieved higher accuracy than and lower loss values than the one without the quanvolutional layer as demonstrated in Figure 4a and 4b respectively. Further, we found the same explicit results when we compared the CNN model to the quantum CNN model as seen in Figure 4c and 4d. However, those results are not surprising since the quantum linear and quantum CNN models have more layers than the linear and CNN models. Then, we compared the quantum CNN model and the 2-CNN model and surprisingly discovered that the quantum CNN model achieved lower loss values than the 2-CNN model. However, they achieved the same accuracy as shown in Figure 4e and 4f.

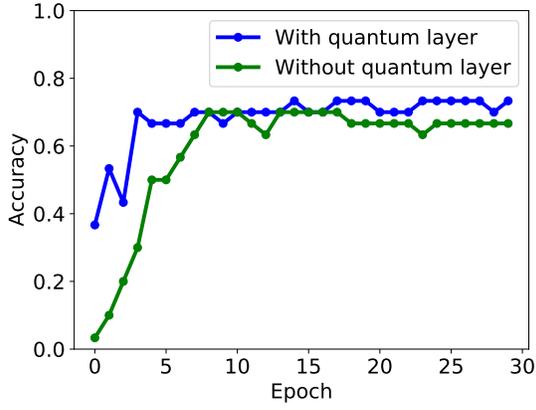
B. Adversarial robustness

In addition, we tested all the models in terms of adversarial robustness against CWA, which was a strong attack. According to Figure 5, most of the models were significantly vulnerable to adversarial examples generated from themselves. However,

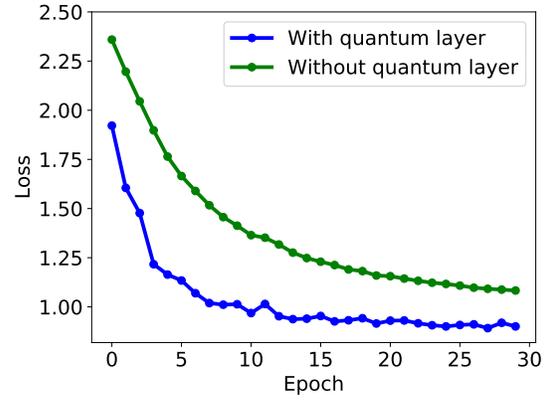
the quantum linear model could achieve very high accuracy on the adversarial examples created from itself. The reason is that we picked only test samples that it could correctly classify to create adversarial examples from the quantum linear model, and CWA could not alter the confidences of the model at all. Therefore, the quantum linear model is robust against CWA. In addition, the quantum CNN model also achieved high accuracy on adversarial examples created from the quantum linear model because the quanvolutional layer could extract features that were useful for specific images. Although those quantum models were slightly affected by adversarial examples created from the classical models, they were more robust than the classical models. Furthermore, according to Figure 5, we found that the more complex architecture of a model was, the more adversarially robust it became. Therefore, it followed a finding in [30].

VI. DISCUSSION AND CONCLUSION

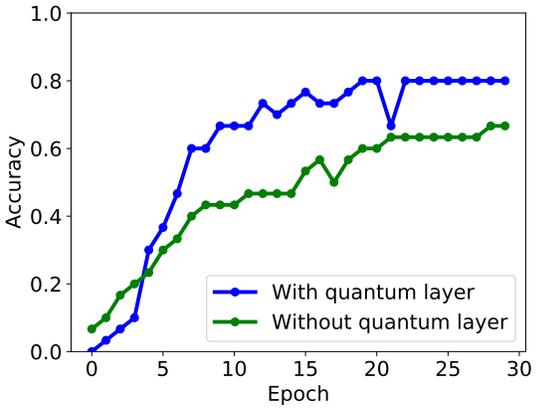
A quanvolutional neural network (QNN) is a convolutional neural network (CNN) with an additional quanvolutional layer composed of n quantum circuits for n filters. We observed that adding a quanvolutional layer to a model could increase its accuracy and reduce its loss value. Also, when we substituted



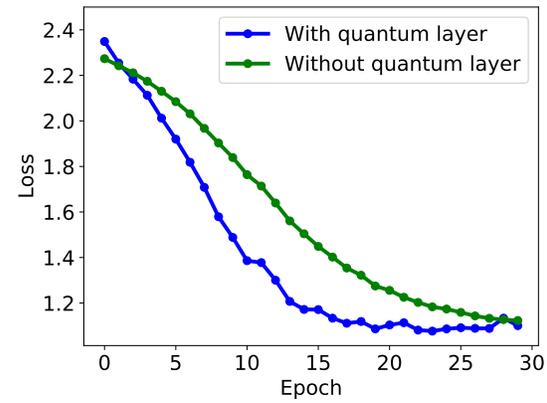
(a) Accuracy between the linear model and the quantum linear model.



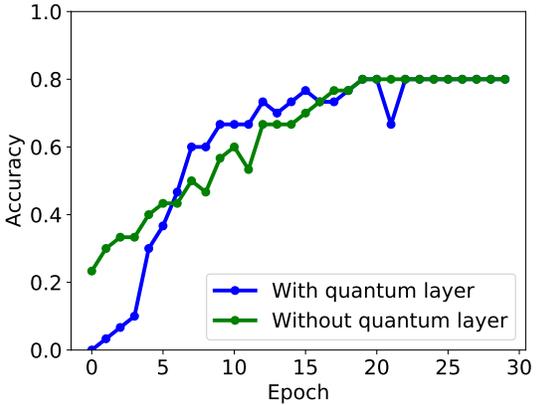
(b) Loss values between the linear model and the quantum linear model.



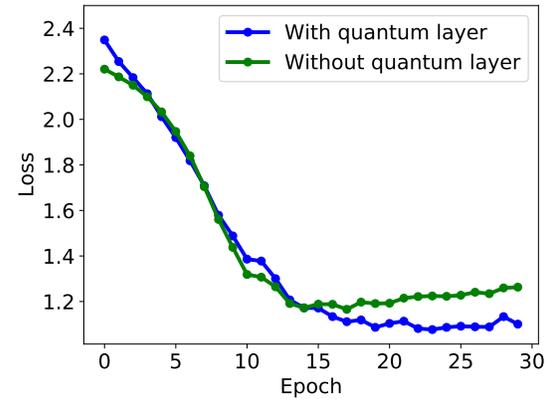
(c) Accuracy between the CNN model and the quantum CNN model.



(d) Loss values between the CNN model and the quantum CNN model.



(e) Accuracy between the 2-CNN model and the quantum CNN model.



(f) Loss values between the 2-CNN model and the quantum CNN model.

Fig. 4: Comparison of accuracy and loss values between (a, d) the linear model and the quantum linear model, (b, e) the CNN model and the quantum CNN model and (c, f) the 2-CNN model and the quantum CNN model respectively.

a convolutional layer with a quantum layer (in the case of the quantum CNN model and 2-CNN model), we noticed that they achieved the same accuracy. Nonetheless, a model with a quantum layer could achieve a lower loss value

than the one with a convolutional layer.

In addition to accuracy and loss value, adversarial examples created from one model could hardly transfer to that model with a quantum layer, and CWA could not harm the models

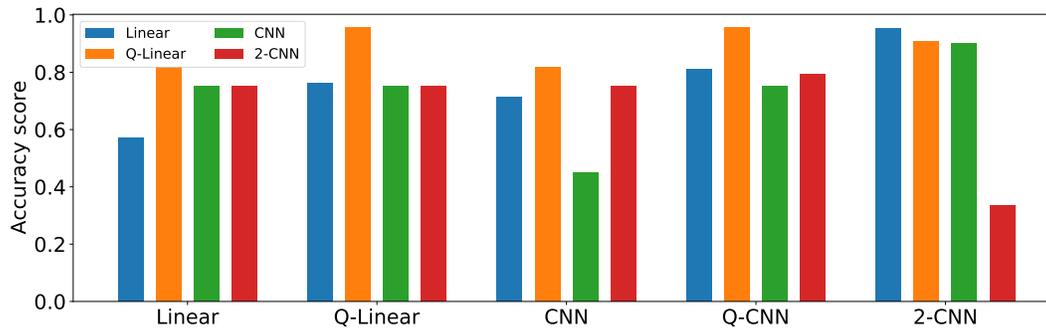


Fig. 5: Accuracy scores achieved by the models with adversarial examples generated from the linear model (blue), the quantum linear model (orange), the CNN model (green) and the 2-CNN model (red). Note that Q-Linear and Q-CNN are respectively the quantum linear model and quantum CNN model.

with a quanvolutional layer. Thus, a neural network can benefit from a quanvolutional layer. In the future, we plan to create a quanvolutional neural networks whose quanvolutional layer is trainable to determine its locally best parameters. We expect it to outperform this version of the quanvolutional neural network in terms of accuracy, loss value and adversarial robustness.

ACKNOWLEDGMENT

The work presented here was supported in part by the Baylor AI lab in Baylor University's Department of Computer Science.

REFERENCES

- [1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.
- [2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
- [3] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.
- [4] E. Farhi, H. Neven *et al.*, "Classification with quantum neural networks on near term processors," *Quantum Review Letters*, vol. 1, no. 2 (2020), pp. 10–37 686, 2020.
- [5] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," *Physical Review A*, vol. 98, no. 1, p. 012324, 2018.
- [6] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.
- [7] S. Oh, J. Choi, and J. Kim, "A tutorial on quantum convolutional neural networks (qcnn)," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2020, pp. 236–239.
- [8] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quanvolutional neural networks: powering image recognition with quantum circuits," *Quantum Machine Intelligence*, vol. 2, no. 1, pp. 1–9, 2020.
- [9] J. Orduz, P. Rivas, and E. Baker, "Quantum Circuits for Quantum Convolutions: A Quantum Convolutional Autoencoder," in *Transactions on Computational Science and Computational Intelligence*. Springer, 2021, accepted, to be published soon.
- [10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.
- [15] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.
- [16] A. Kurakin, I. Goodfellow, S. Bengio *et al.*, "Adversarial examples in the physical world," 2016.
- [17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.
- [18] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.
- [19] A. Kamruzzaman, Y. Alhwaiti, A. Leider, and C. C. Tappert, "Quantum deep learning neural networks," in *Future of Information and Communication Conference*. Springer, 2019, pp. 299–311.
- [20] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [21] N. Liu and P. Wittek, "Vulnerability of quantum classification to adversarial perturbations," *Physical Review A*, vol. 101, no. 6, p. 062331, 2020.
- [22] S. Lu, L.-M. Duan, and D.-L. Deng, "Quantum adversarial machine learning," *Physical Review Research*, vol. 2, no. 3, p. 033212, 2020.
- [23] J. Guan, W. Fang, and M. Ying, "Robustness verification of quantum classifiers," in *International Conference on Computer Aided Verification*. Springer, 2021, pp. 151–174.
- [24] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, pp. 1–20, 2021.
- [25] A. Y. Kitaev, "Quantum computations: algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, pp. 1191–1249, dec 1997. [Online]. Available: <https://doi.org/10.1070/rm1997v052n06abeh002155>
- [26] T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. Reboucas Filho, "Performance analysis of google colab as a tool for accelerating deep learning applications," *IEEE Access*, vol. 6, pp. 61 677–61 685, 2018.
- [27] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.
- [28] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.