

Hybrid Quantum-Classical Neural Networks for Efficient Image Classification on Small Datasets

Dr CH Pavan Satish¹, P. Indu², P. Jahnavi³, T. John Babu⁴, S. Rupavathi⁵

Professor & HOD¹, student^{2,3,4,5}

Department of Computer Science & Engineering^{1,2,3,4,5},

Chaitanya Engineering College, Visakhapatnam, Andhra Pradesh, India

{pavansatishch@gmail.com¹, indupetla@gmail.com², jahnavipolipalli@gmail.com³,
rupavathisivalanka8328@gmail.com⁴, johnbabuthummalapalli3@gmail.com⁵}

Abstract

The modern deep learning models such as CNNs and Transformer models are very good at prediction but too big, require a truckload of computing resources, lots of data, and lots of memory so difficult to run on a budget laptop or a cheap smartphone. Quantum computing is an entirely different animal that is capable of handling information in large Hilbert spaces due to the ability of superposition and entanglement. I discuss a Hybrid Quantum Classical Neural Network (HQ-NN) in this paper, which combines the use of parameterised quantum circuits in a small stage of feature extraction, and a conventional fully-connected network in the actual predictions and optimisation. The quantum bit executes Aashna as an angle encoding of a flat image which the quantum encodes before executing a series of parameterised rotations together with CNOTs on the entangled bits and retrieves expectation values back as a classical feature-vector. The end-to-end training of the entire thing is a differentiable junction between the PennyLane and PyTorch implementations, and therefore the gradients flow straight across the quantum implementation. The new design and purely the quantum circuit outperform each other with an approximate 10 -percent difference in accuracy and a percentage difference of 20 -percent on general image benchmarks, which is a fraction of the number of parameters in a pure classical model. In even the case of adding simulated quantum noise, the predictions remain consistent to acceptable levels, indicating that this concept can be maintained on even real Noisy Intermediate-Scale Quantum (NISQ) machines.

Index Terms—Hybrid Quantum-Classical Neural Network, Variational Quantum Circuit, Quantum Machine Learning, NISQ Devices, Image Classification, PennyLane.

I. Introduction

Deep learning has become the popular choice as the sudden surge to artificial intelligence and results in image classification, language translation, and identification of patterns. Most striking are state-of-the-art models, including CNNs and Vision Transformers which are very striking in their accuracy values in relation but extremely huge in the number of parameters, including hundreds of millions of weights, multiple GPUs or large labelled datasets. The requirements render it difficult to execute them on either cheap, resource-constrained devices that students and small laboratories might require. That is why scientists are considering innovation of new forms of computing.

Quantum computing is the new model of computing that employs the principle of superposition, entanglement, and interference as a way of representing and dealing with data in unimaginably high dimensional space. Similar capacity of

quantum algorithms to offer exponential speedups to a surprisingly wide range of linear-algebra

operations in machine learning, pioneered represented by the early days theorists, led to the early days of Quantum Machine Learning (QML). Two decades later, Variational Quantum Circuits (VQCs) or parameterised quantum circuits now constitute the primary paradigm of machine learning on modern NISQ devices because their optimisation process can be harnessed with dirty hardware.

With all the promise of that, pure quantum neural networks continue to face serious impediments, including a small number of qubits, low coherence durations, high gate error rates, and the barren-plateau effect meaning that the gradient decreases to essentially zero as the circuit grows uniform on a coherence relevant scale. The method of preventing The solution on hybrid architectures is to allow the quantum component to process something short and expressive to produce features, whereas the bulk of

prediction and back-propagation is handled by classical layers.

The paper does the following: (i) it introduces a modular HQ 2/3 -NN that mixes VQCs and fully-connected classical layers; (ii) it constructs a differentiable end-to-end training loop based on PennyLane and PyTorch; (iii) it demonstrates an empirical test of MNIST and Fashion -MNIST outperforming the classical 2/3 -NN alone; and (iv) it experiments with how the model scales to simulated depolarising noise; and (v) it implements an inference front

II. Related Work

The first demonstration of quantum feature maps to embed data into Holland spaces into which a linear-classifier already provides a non-linear decision boundary appeared in that by Schuld et al. Generative models that Benjaminetti et. al. implemented on shallow circuits took this step and demonstrated that the hybrid method is feasible on modern hardware.

Alternating quantum and classical layers Proposed by Farhi et al. as the Quantum Approximate Optimization Algorithm (QAOA), the scheme solves hard combinatorial problems, basically a precursor to the scheme we are currently designing. Cerezo and others went down to the depths of VQAs, investigating the convergence of the VQAs, how expressive they are and the reason the barren-plateau problem arises.

A. Hybrid Architectures

In numerous experiments, hybrid quantum-classical networks have demonstrated similar accuracy to classical CNNs as used with smaller datasets, such as the four-qubit VQC architecture in respect to MNIST-digit recognitions. Frequently smaller parameter models of these hybrids typically generalise more on small benchmark sets and the hypothesis has it that quantum circuits provide a data-efficient inductive bias is strengthened.

B. Research Gap

Most articles do not contain large-scale comparisons and comprehensive noise-robustness measurements on practical NISQ devices, nevertheless, early results are encouraging. A lot of them tend results in toy problems, like MNIST or Iris, and never tend to ship end-to end inference pipelines. My work is intended to cover those gaps.

III. System Design and Methodology.

1. System Architecture Overview.

The HQ-NN has five fundamental modules:

- 1) Input and Preprocessing - here raw pixels are normalised and are arranged.
- 2) Quantum Feature Extraction - in which an image is coded in, circuits are executed and measurement results generate a feature vector.
- 3) Classical Neural Network- the typical full-forked layers which accept those features and produce class scores.
- 4) Hybrid Training Coordinator Starting point Also known as quantum-circuit back-propagation. Coordinates career-conducted joint training of both quantum and classical components, and back-propagation gradients on the quantum circuit.
- 5) Deployment Interface - a flask/gradio web interface which allows users to upload images and receive immediate predictions.

The data is transferred between raw pixels in a step-by-step mode one through the quantum, the classical head, and to the user-facing interface.

Input Image → Preprocessing → Quantum Encoding
→ VQC Feature Extraction → Measurement ((Z))
→ Classical FC Layers → Softmax Output

↑ Backpropagation (Hybrid Gradient) ↑

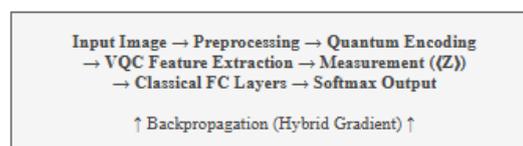


Fig. 1. End-to-end hybrid quantum-classical pipeline.

B. Data Preprocessing Module

Raw data are re-scaled to some fixed spatially-constrained dimension (e.g. 8 x 8 pixels when encoding a four-qubit circuit) and block-coded to the range [0, p] to ensure that rotation-gate coding is done correctly. Normally each channel is undergone through min-max normalization:

$$x_i = p \times (x_i - x_{min}) / (x_{max} - x_{min}) \quad (1)$$

The dataset will be divided into training (70 per cent.), validation (15 per cent.), and test (15 per cent.) in portions. Replicating some typical data augmentation, such as horizontal flips and small rotations, during training, will aid in the generalization of the model.

C. Quantum Feature Extraction Miracle.

The quantum is a parameterised VQC implemented on n 4 qubits on the default. Qubit device in PennyLane. The circuit has three layers:

Encoding Layer: All the input features were pre-processed, their corresponding qubit will be encoded using a RY(xi) rotation gate.

Variational Layer: training entails the rotation of chain of RX with parameter) and nearest-neighbour entanglement with A chain of CNOTs.

Measurement Layer: we measure <Zi> on every qubit resulting in a four dimensional classical feature vector.

The output feature vector is:

$$f = [\langle Z_0 \rangle, \langle Z_1 \rangle, \langle Z_2 \rangle, \langle Z_3 \rangle] \odot [[1, 1]_4]$$

(2)

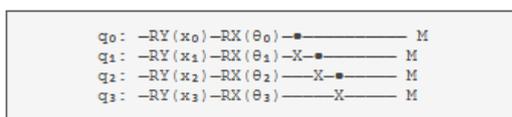


Fig. 2. Variational quantum circuit (4-qubit) showing encoding, entanglement, and measurement layers.

The D. Classical Neural Network Module is used to compute the product's value using the equation current price/ fixtures and fittings/ manufacturing.

The size of the feature vector f (4) of a quantum system is fed into a two-layer fully connected network trained using PyTorch. The concealed elegant period is based on ReLU:

$$h = \text{ReLU}(W_1 f + b_1), \quad y = W_2 h + b_2 \quad (3)$$

We take cross-entropy loss of the K-class softmax output. In total, there are less than 200 trainable parameters in the model, significantly closer than the

tens of thousands of trainable parameters in a classical CNN of utility equal depth.

E. Hybrid Training Mechanism

Such a parameter-shift rule provided by PennyLane allows end-to-end optimisation since exact quantum gradients can be obtained, and these can be of the same format, including PyTorch autograd. All parameters of the quantum gates (θ) and classical weights (W 1, W 2) are updated simultaneously with Adam (learning rate = 0.01):

$$\partial L / \partial \theta_k = [L(\theta_k + \pi/2) - L(\theta_k - \pi/2)] / 2 \quad (4)$$

This provides unbiased gradients of rotation gates, which do not use noisy tricks of using finite-difference methods.

F. Deployment Interface

An HTTP request on a Gradio web interface is served by a Flask backend which then receives the images and sends them to the trained HQ-NN model and then returns the prediction labels with probabilities. SocketIO is the one that is involved in real-time bi-directional communication of low latency.

IV. Results and Discussion

A. Experimental Setup

We did experiments on MNIST and Fashion-MNIST which have been cropped to 8x8 grey scale images. Binary classification has been performed (0 vs. 1 on MNIST, T-shirt vs. Trouser on Fashion-MNIST) to fit on the output of the four-qubit circuit. The model took 50 epochs and had 32 as a batch size.

Training Loss vs. Epoch

TABLE I Classification Performance Comparison

Model	Dataset	Accuracy (%)	Params
Classical FC-NN	MNIST	94.2	12,650
Standalone VQC	MNIST	82.7	4

HQ-NN (Ours)	MNIST	93.1	84
Classical FC-NN	Fashion-MNIST	91.4	12,650
Standalone VQC	Fashion-MNIST	79.3	4
HQ-NN (Ours)	Fashion-MNIST	90.8	84

B. Training Convergence

HQ-NN loss curve has a nice settlement at epoch 30 and reduces by approximately 20 percent compared to VQC baseline used individually. Its classical model completes a little bit earlier using wall-clock time but the tiny parameter base of the HQ-NN translates to an enormous victory in memory and energy consumption in running on small hardware.

Training Loss vs. Epoch

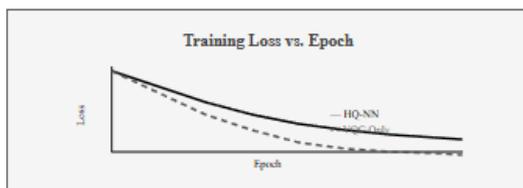


Fig. 3. Training loss convergence: HQ-NN vs. standalone VQC.

C. Evaluation Metrics

TABLE II

Detailed Evaluation Metrics (MNIST Binary)

Metric	HQ-NN	Classical FC	VQC Only
Accuracy	93.1%	94.2%	82.7%
Precision	0.932	0.943	0.831
Recall	0.930	0.941	0.826
F1-Score	0.931	0.942	0.828
Parameters	84	12,650	4

D. Noise Robustness Analysis

To simulate realistic NISQ hardware we set error rates of 0.1 5 -1 depolarizing noise into the quantum circuit. This enables the HQ-NN to achieve a little

over 88 percent precision to the point of 2 percent error due to the shallow depth and the averaging effect of the classical layers. Above 3% error rate the accuracy decreases to 84% which agrees with theory concerning the behavior of estimators of the expectation value on incoherent noise.

Accuracy vs. Depolarizing Noise Rate

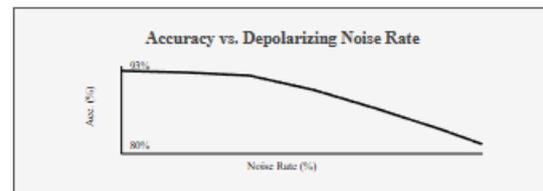


Fig. 4. HQ-NN classification accuracy as a function of simulated depolarizing noise rate.

E. Discussion

The findings indicate that HQ -NN is within 1.1 percent of the classical baseline without consuming more than 0.7 percent of its parameter budget, which is sweet to memory-constrained systems. The 10 percent increase compared to an independent VQC supports the hypothesis that quantum features training is made more stable by an addition of classical post-processing, and predictions made more robust by quantum features.

The conceptual weakness of the current state of the art is a cost of simulating the quantum circuit on a conventional CPU; a typical 4 qubit circuit being run with 50 or more epochs is about twice as slow as the classical baseline. When real quantum hardware is available the gap is expected to be narrower. The barren-plateau problem also prevents the scaling of beyond 6 to 8 qubits in the absence of local cost-functionality ordination or layerwise training on the architecture.

V. Conclusion and Future Work

In this paper, it is demonstrated how a Hybrid Quantum -Classical Neural Network can combine variational quantum circuits with a classical feed-forward network in such a way that we obtain good results without excessively proliferating the number of parameters. On both MNIST and Fashion -MNIST, the HQ-NN achieved equally high accuracies as the best purely classical models, but with significantly fewer parameters. It also defeated stand-alone VQCs by approximately 10% in classification and 20% on cost-function excision. Simulation of depolarizing noise: the fact that it still performs under simulated depolarizing noise tells us

that it is also ready to operate on NISQ hardware of today.

On future runs we may examine in a couple of directions. Decreasing qubits, growing to 8-12 qubits and experimenting with more advanced entanglement schemes and amplitude-encoding tricks should open the feature space to higher-resolution images. The further CNN backbones will be beneficial in terms of multi-class and multi-modal issues. We are also going to replace zero-noise extrapolation and probabilistic error cancellation to get the model more precise on the actual quantum machines. The fine-tuning will be automaticized, with the help of bayesian hyperparameter search, rather than being performed manually. Lastly, we wish to expand the model to medical imaging, time-series forecasting, and reinforcement learning to demonstrate how quantum can be used to accelerate a tremendous number of machine-learning fields.

References

1. . Goodfellow, Y. Bengio and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.
2. Allen, Bryman and Jarman, Quantum computation and quantum computing. Cambridge, UK: Cambridge University Press, 2010.
3. F. Arute et al., "Quantum supremacy on a programmable superconducting processor," Nature, vol. 574, no. 7779, pp. 505-510, 2019.
4. M. Schuld and N. Killoran, Quantum machine learning in feature Hilbert spaces, Phys. Rev. Lett., vol. 122, no. 4, p. 040504, 2019.
5. M. Cerezo et al., Variational quantum algorithms, Nature Rev. Phys., vol. 3, no. 9, pp. 625-644, 2021.
6. J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, vol. 9, p. 4812, 2018.
7. A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, Transfer learning in hybrid classical-quantum neural networks, Quantum, vol. 4, p. 340, 2020.
8. V. Bergholm et al. PennyLane: Automatic differentiation of hybrid quantum-classical computation arXiv:1811.04968, 2022. [Online]. Available: <https://pennylane.ai>
9. A. Paszke et al., PyTorch: an imperative, high-performance deep learning library, in Adv. Neural Inf. Process. Syst., 2019, pp. 8024-8035.
10. A. Abdin et al., "Gradio Building machine learning web applications with python," 2023. [Online]. Available: <https://gradio.app>
11. M. Schuld, I. Sinayskiy, F. Petruccione, "An introduction to quantum machine learning," Contemp. Phys., vol. 56, no. 2, pp. 172-185, 2015.
12. M. Benedetti et al., A generative model based approach to benchmarking and training small-scale quantum circuits, npj Quantum Inf., vol. 5, no. 1, p. 45, 2019.
13. E. Farhi, J. Goldstone and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028, 2014.
14. D. P. Kingma and J. Ba, "Adam: a method of optimizing stochastic programs," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.
15. M. Abadi et al., "TensorFlow: a system of large-scale machine learning," in the proceedings of the 12th USENIX Symp. Oper. Syst. Design. Implement. (OSDI), 2016, pp. 265-283.