Defining Quantum Neural Networks via Quantum Time Evolution

Aditya Dendukuri[1]   Blake Keeling[1]   Arash Fereidouni[2,3]   Joshua Burbridge[1]

Khoa Luu[1]   Hugh Churchill[2,3]

*

Abstract

This work presents a novel fundamental algorithm for for defining and training Neural Networks in Quantum Information based on time evolution and the Hamiltonian. Classical Neural Network algorithms (ANN) are computationally expensive. For example, in image classification, representing an image pixel by pixel using classical information requires an enormous amount of computational memory resources. Hence, exploring methods to represent images in a different paradigm of information is important. Quantum Neural Networks (QNNs) have been explored for over 20 years. The current forefront work based on Variational Quantum Circuits is specifically defined for the Continuous Variable (CV) Model of quantum computers. In this work, a model is proposed which is defined at a more fundamental level and hence can be inherited by any variants of quantum computing models. This work also presents a quantum backpropagation algorithm to train our QNN model and validate this algorithm on the MNIST dataset on a quantum computer simulation.

1   Introduction

Deep Learning is a highly successful field of computer science, playing an integral role in cutting-edge technology like pattern recognition and self-driving cars. With recent developments in deep learning, fields like Computer Vision have been revolutionized with significant advancements in pattern recognition, classification, etc. In this work, we focus on the relevance of the quantum framework of information, namely *Quantum Computing*, to image classification using deep learning. Image classification is currently one of the most rapidly-changing research areas in deep learning. Representing image pixel by pixel using classical information requires enormous amounts of computational resources. Hence, exploring methods to represent images in a different paradigm is important. Deep Neural Networks in Quantum Information or Quantum Neural Networks (QNNs) have been getting a lot of attention in the past couple of years due to recent advancements in quantum computing. One of the most popular methods of QNN's is the Variational Quantum Circuit. Variational Quantum Circuits however have a disadvantage. The frontier models of variational quantum circuits are only limited to the Continous Variable Paradigm of quantum computers. Hence we need to build the quantum machine learning theory around more fundamental concepts of Quantum Theory which can universally apply to any quantum computing paradigm. In this work, one of the main focal points is to develop the notion of Variational Quantum Circuits in terms of fundamental concepts in quantum theory, specifically Hamiltonian operators and time evolution of quantum states. Our second focus is to test experimentally the performance of variational quantum circuits for the MNIST handwritten digits database. Our contributions can be compiled as follows.

Contributions of this Work This work defines a Quantum Neural Network (QNN) model as a

---

*[1]Computer Vision and Image Understanding Lab, Computer Science and Computer Engineering Department, [2]Department of Physics, *and* [3]Institute for Nanoscience and Engineering, University of Arkansas, Fayetteville, AR, 72701, Emails: *{adenduku, afereido}@email.uark.edu, {khoaluu, hchurch}@uark.edu*
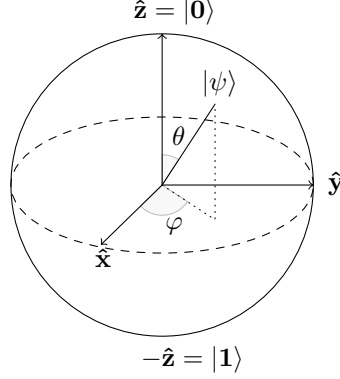
Figure 1: Visualization of Rotation and Measurement of quantum state using the Bloch Sphere. θ and φ are the parameters in this case. Tuning them would mean rotating the state $|\psi\rangle$ around the sphere and measuring in z direction means measuring in the computational basis ($|0\rangle, |1\rangle$).

Hamiltonian dictating the evolution of quantum states. This work also includes derivation of theoretical expression for quantum squared loss and a *Quantum Backpropogation algorithm* was developed based on the definition. This work then demonstrates how this fundamental definition can translate to a specific quantum system by experimentally testing our QNN model by training it to identify handwritten digits on MNIST database in a Quantum Computer Simulation library QuEST [1]. The proposed method is able to obtain 64% accuracy for MNIST on a Quantum Neural Network which is the highest accuracy for a large scale data-set to date.

1.1 Quantum Computation

Quantum Computing defines a non-deterministic approach to represent classical information using ideas from Quantum Theory. The idea of quantum information was first introduced in 1980 by Paul Benioff [2]. In the same year, Yuri Manin proposed a quantum computer in his textbook "Computable and Uncomputable" [3]. In the year 1982, the field was formalized and made popular by Richard Feynman in his paper about simulating physics in computers [4]. David Deutsch further advanced the field by formulating a Quantum Turing Machine [5]. Since then, there have been a number of algorithms developed such as the Grover's search algorithm [6] and Shor's factoring algorithm [7]. Shor's factoring algorithm is particularly significant since it demonstrated exponential speedup in factoring a large number. Since, modern day encryption techniques operate using huge numbers, the ideas sprouting from the Quantum Information are already implying effect on our world. The core idea of quantum computing is a *qubit*: the quantum analogue of a classical computer bit. A classical bit is capable of storing a determined value (0 or 1). A qubit (say ψ), can be represented using a superposition of both 0 and 1,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{1}$$

where, $|0\rangle$ and $|1\rangle$ are vectors $\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$ respectively. α and β are the *probability amplitudes*. These probability amplitudes are represented using complex numbers. Hence, getting a real valued probability would mean to take the modulus squared as follows,

$$P(\psi = |0\rangle) = |\alpha|^2; \quad P(\psi = |1\rangle) = |\beta|^2.$$

The probabilities are normalized, i.e. they add up to 1: $|\alpha|^2 + |\beta|^2 = 1$. This probabilistic model potentially allows us to define algorithms which can potentially be intrinsically parallel.

1.1.1 The Bloch Sphere and Quantum Operations

Given Eqn. (1), we can expand out the state of a qubit (ψ) into a vector as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right] + \beta\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right] = \left[\begin{smallmatrix}\alpha\\\beta\end{smallmatrix}\right]$$

The *Quantum Logic Gates* that can be applied to qubits come in the form of unitary matrices. Some of these gates are shown in Table 1. The function of quantum gates can be visualized in the Bloch

Name	Dirac Notation	Classical Notation	Function
PauliX	$\hat{\sigma_x}|\psi\rangle$	$\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} \alpha \\ \beta \end{smallmatrix}\right] = \left[\begin{smallmatrix} \beta \\ \alpha \end{smallmatrix}\right]$	Rotate in X direction by π
PauliY	$\hat{\sigma_y}|\psi\rangle$	$\left[\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} \alpha \\ \beta \end{smallmatrix}\right] = i\left[\begin{smallmatrix} \alpha \\ -\beta \end{smallmatrix}\right]$	Rotate in Y direction by π
PauliZ	$\hat{\sigma_z}|\psi\rangle$	$\left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]\left[\begin{smallmatrix} \alpha \\ \beta \end{smallmatrix}\right] = \left[\begin{smallmatrix} \alpha \\ -\beta \end{smallmatrix}\right]$	Rotate in Z direction by π
Hadamard	$\hat{H}|\psi\rangle$	$\frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]\left[\begin{smallmatrix} \alpha \\ \beta \end{smallmatrix}\right] = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} \alpha+\beta \\ \alpha-\beta \end{smallmatrix}\right]$	Puts the state in superposition

Table 1: Examples and Demonstration of Various Quantum Logic Gates.

spheres (as shown in Figure 1). The fundamental Pauli spin gates ($\Xi : \{\sigma_x, \sigma_y, \sigma_z, I\}$) form a complete basis. Using this basis we can form any unitary quantum gate (in the form $e^{-i\hat{H}t}$) where $\hat{H} = \sum_j a_j \Xi_j$ is the Hamiltonian and t is the time evolution. Any multi qubit quantum gates like the *Controlled NOT* (or CNOT) gate which flips the target qubit if the control qubit is 1 can be decomposed into this basis.

2 Neural Networks and Deep Learning

Classical neural networks are k-partite graphs which represent non-linear transformations. The nodes of the graph may or may not be fully connected. The first "layer" represents the input to the network as the nodes. To propagate through the network, we transform the input based on the bond strength between the nodes (weights). Let the input to the network be represented as $\vec{x}$, and the transformation matrix $\hat{W}$ hold the weights between the connections. The transformation for a layer L can be modelled as:

$$L_{\hat{W}}(\vec{x}) = \sigma(\hat{W}\vec{x} + \vec{b}) \tag{2}$$

The bias vector ($\vec{b}$) acts as the intercept of the linear model. The linear model is then passed into a non-linear logistic function to normalize the output. If there are multiple layers, this output will be the input to the next transformation and hence, a neural network can be represented as:

$$N(\vec{x}) = L_n \circ L_{n-1} \circ \ ......L_2 \circ L_1(\vec{x}) \tag{3}$$

We can use this function to model a variety of regression and classification problems. The weights and the biases act as tunable parameters which can be adjusted to compute the desired result. This fine-tuning process is termed as "training" the neural network. The training is carried out by an algorithm called *backpropogation*. The backpropogation algorithm minimizes a "loss" function which models the accuracy of the network. Mean Squared Error is a common loss function defined as follows:

$$\mathcal{L} = (N_{W,b}(\vec{x}) - y(\vec{x}))^2 \tag{4}$$

The training is carried out by updating the weights and biases such that Eqn. (4) is minimized as follows:

$$\hat{W} = \hat{W} - \eta\frac{\partial \mathcal{L}}{\partial \hat{W}}; \quad \vec{b} = \vec{b} - \eta\frac{\partial \mathcal{L}}{\partial \vec{b}} \tag{5}$$

The partial derivatives are calculated via the chain rule as it is a sequence of composite functions or layers. Hence, every transformation in the model like the linear and the logistic non-linear transformation of every layer has been fine-tuned to generate a desired output.

3 Neural Networks in Quantum Information

Remodelling the theory behind neural networks in quantum information is gaining popularity due to many advancements in quantum computers. The earliest ideas to model a neural network in quantum computing trace back to 1995 paper by Kak [8]. In the same year Menneer and Narayanan
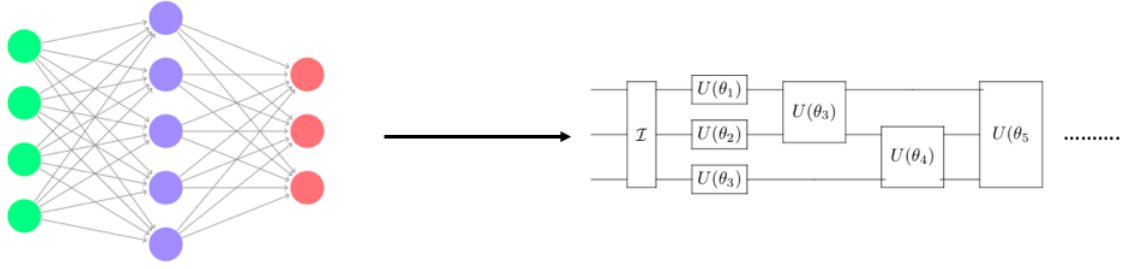
Figure 2: Differentiating between a classical neural network and a variational quantum circuit. Since, input is encoded in a quantum register, there is an exponential reduction in the input space. For the quantum circuit $\mathcal{I}$ is the encode image gate. In addition to that, the rotation gates are naturally computed in a quantum system.

proposed a neural network inspired by quantum processes [9]. Perus [10] suggested the advantage of quantum parallelism being applied to a neural network architecture. The first comprehensive study of a quantum neural network model was conducted by Menneer [11]. Altaisky introduced a quantum perceptron model in [12], but noted that the learning rule for this perceptron did not observe unitarity in general. Gupta and Zia derived a quantum neural network model from the Deutsch's model of quantum computational network [13]. A Quantum Neural Network based on qubits was introduced by Kouda [14]. Schlud et. al proposed a set of guidelines for developing quantum neural network models[15]. Specifically, a proposal for a generalized quantum neural network should (1) be able to encode some binary string of length N and produce as output some binary string of length M which is closest to N by some distance measure, (2) reflect one or more basic neural computing mechanisms, and (3) be based on quantum effects such as superposition and interference while remaining fully consistent with quantum theory. Quantum machine learning was experimentally tested for two class classification using quantum support vector machine [16]. Wiebe et al. introduced the term quantum deep learning in their paper [17]. Adachi proposed a quantum neural net model which applies Quantum Annealing [18]. A Quantum Boltzmann Machine model was introduced by Amin et al. A quantum recurrent neural network modelled after the Ising Spin Model was proposed in [19]. Another class of quantum neural networks called variational quantum circuits with tunable parameterized unitary gates implemented in the Continous Variable Model was introduced by Killioran et. al in [20]. A Quantum Generative Adversarial Network based on the Variational Circuit Model was introduced by Seth Lloyd in [21].

One obvious difficulty in transplanting quantum computing into machine learning is the general requirement for nonlinear activation functions. A quantum register stores a state vector, which contains the probability amplitudes associated with each possible state. Clearly, this vector is subject to the normalization condition, which means that any operator applied to the system must be unitary. A unitary operator is a square matrix with the property U*U = UU* = I, where U* is the conjugate transpose of U. Thus, the actions on a quantum register are constrained by linear dynamics, which makes quantum computing fundamentally incompatible with the activation function paradigm. However, the time evolution of quantum states itself, is intrinsically non-linear (as shown in equation 6).

3.1 Variational Quantum Circuit

There are a number of QNN models proposed based on parameterized quantum circuits, also called Variational Quantum Eigensolvers. The circuits composed of parameterized unitary gates that are optimized to produce the desired wave-function. This model can be inferred as a probability distribution similar to output of a softmax function. This is done by repeating and measuring the circuit multiple times and calculating the probability distribution of the basis states. To measure the circuit,observables such as the pauli-z spin matrix ($\sigma_z = \begin{bmatrix} 1,0 \\ 0,-1 \end{bmatrix}$) can be used to measure in the computational basis as shown in Figure 1). The state of any qubit can be visualized in a Bloch sphere. This kind of circuit with tunable parameters is called a Variational Quantum Circuit as shown in

Figure 2). Variational Quantum Circuits belong to a much larger family of *Hybrid* algorithm which require both classical and quantum components [22].

3.1.1 Advantages and Disadvantages of the Variational Quantum Circuit Model

There are a number of potential advantages of a Variational Quantum Circuits over classical deep models as follows:

- The first motivation is the notion of quantum parallelism. Since a state can be modeled as it is holding both possibilities, any operation potentially could naturally compute both probabilities at once.
- The natural ability of quantum gates to represent a rotation operator. There exists unitary quantum gates which represent rotations around the Bloch sphere.
- The superposition between '0' and '1' enables to encode N bit information in $log_2 N$ qubits.
- The reduction in features space also means reduction in number of layers and nodes needed.

However these advantages come with their caveats. Even though the notion of the quantum parallelism is theoretically sound, the "quantum advantage" by which a quantum algorithm outperforms a classical one has not yet been experimentally demonstrated. This is because it is extremely difficult to maintain quantum phenomena in real physical systems [23]. The present quantum systems are highly susceptible to external noise resulting in depletion of the superposition. This phenomenon is called *decoherence*. This issue is usually addressed by a phenomenon called error-correction, though decoherence must be suppressed below a certain threshold to achieve fault-tolerant operation and the physical resource demands of current error-correcting codes are significant.

3.1.2 Encoding Input in a Quantum Register

Indeed, the state of N qubits can be mathematically represented using 2^N dimensional Hilbert Space. We encode our $2^{\frac{N}{2}}$ by $2^{\frac{N}{2}}$ bit input image in this Hilbert space. Let us consider 2^N bits of classical input ($\vec{x}$) to be encoded in a quantum state $|\psi(x)\rangle$ with N qubits. Encoding the input in the coefficients of basis states is shown below:

$$|\psi(x)\rangle = \prod_{j=1}^{2^N} x_j |j\rangle$$

3.2 Quantum Neural Network based on Time Dependent Schrödinger Equation

Although the Variational Quantum Circuit model fairly mimics a quantum analogue to a classical neural network, they are not theoretically congruent. In other words, a fully connected neural network with multiple layers cannot be exactly converted to a Variational Quantum Circuit. Another important factor would be to focus on using unique quantum properties like entanglement as most of these quantum operations can be simulated in a classical system using supercomputers. Hence, we propose a quantum neural network model based on these important facts. Our main motivation behind modelling this quantum neural network is to utilize the quantum properties to the fullest.

3.3 The Proposed Model

Consider a quantum register with an initial state $|\psi\rangle(0)$ with an input encoded. The time evolution of any Quantum State can be defined as follows,

$$|\psi(t)\rangle = e^{-i\hat{H}t}|\psi(0)\rangle. \tag{6}$$

where, $\hat{H}$ is called the Hamiltonian of the system which dictates the evolution of the system. Hence, we will define the Evolution Structure of Quantum Neural Network in the Hamiltonian basis.

$$\hat{H} = \sum_{j=1}^{2^N} w_j \hat{\Xi}_j. \tag{7}$$

5

where w_j are real numbers defined as trainable weights. Ξ_j denotes any fundamental quantum gates like the pauliX and Identity gates. We denote the operation in Eqn. 6 as $\hat{\mathcal{N}}$. If we insert this Hamiltonian into Eqn. 6, we get,

$$
\begin{aligned}
\hat{\mathcal{N}}|\psi\rangle &= e^{-i\hat{H}\delta t}|\psi\rangle \\
&= e^{-i\sum_{j=1}^{2^N} w_j \hat{\Xi}_j \delta t}|\psi\rangle \\
&= \prod_{j=1}^{2^N} e^{-iw_j \Xi_j \delta t}|\psi\rangle \\
&= e^{-iw_1\Xi_1\delta t} * e^{-iw_2\Xi_2\delta t} \ldots e^{-iw_k\Xi_k\delta t} \ldots e^{-iw_{2^N}\Xi_{2^N}\delta t}|\psi\rangle.
\end{aligned}
\tag{8}
$$

There is no requirement to introduce any logistic non-linear function since this system is intrinsically non-linear. Eqn. (8) is the theoretical model of our proposed quantum neural network. We can translate any of these matrix exponentials to physical quantum gate matrices (Table 1) [24]. Consider a matrix exponential as shown below for a one-qubit case with pauliX operator and weight w:

$$
\begin{aligned}
e^{-iw\sigma_x\delta t} &= cos(w\delta t)I - isin(w\delta t)\sigma_x \\
&= \begin{bmatrix} cos(w\delta t) & 0 \\ 0 & cos(w\delta t) \end{bmatrix} + \begin{bmatrix} 0 & -isin(w\delta t) \\ -isin(w\delta t) & 0 \end{bmatrix} \\
&= \begin{bmatrix} cos(w\delta t) & -isin(w\delta t) \\ -isin(w\delta t) & cos(w\delta t) \end{bmatrix}
\end{aligned}
$$

Hence, we can construct parameterized quantum gates for any variational quantum circuit with this definition. This can be achieved to model any kind of rotation operation in any direction as long as the matrices defined in Ξ are unitary.

3.3.1 Defining Loss and Training

To optimize our QNN, we need to define a loss function to measure the performance of the network. We first need to encode the labels from the dataset in quantum states (say $|y\rangle$). Since we are working with complex numbers, squaring a complex number means multiplying with its conjugate. If the dataset size is m, we define the cost function as shown below:

$$
\begin{aligned}
\mathcal{L} &= (\langle\psi|\hat{\mathcal{N}}^\star - \langle y|) * (\hat{\mathcal{N}}|\psi\rangle - |y\rangle) \\
&= \langle\psi|\hat{\mathcal{N}}^\star\hat{\mathcal{N}}|\psi\rangle - \langle y|\hat{\mathcal{N}}|\psi\rangle - \langle\psi|\hat{\mathcal{N}}^*|y\rangle + \langle y|y\rangle \\
&= 2 - \langle y|\hat{\mathcal{N}}|\psi\rangle - \langle\psi|\hat{\mathcal{N}}^*|y\rangle
\end{aligned}
\tag{9}
$$

Where $\langle\psi|$ and $\hat{\mathcal{N}}^*$ refer to the complex conjugates of $|\psi\rangle$ and $\hat{\mathcal{N}}$ respectively. To optimize the neural network we need to minimize this cost function. This can be done by computing the gradient of the cost with respect to the weights and updating the weights to optimize the network as given below.

$$
W = W - \eta\nabla\mathcal{L}; \quad \nabla\mathcal{L} = [\frac{\partial\mathcal{L}}{\partial w_1}, \frac{\partial\mathcal{L}}{\partial w_2}, \frac{\partial\mathcal{L}}{\partial w_3} \ldots \ldots \frac{\partial\mathcal{L}}{\partial w_n}]
$$

We would like to compute the partial with respect to k^{th} weight. This process is shown below:

$$
\begin{aligned}
\frac{\partial\mathcal{L}}{\partial w_k} &= -\langle y|\frac{\partial\hat{\mathcal{N}}}{\partial w_k}|\psi\rangle - \langle\psi|\frac{\partial\hat{\mathcal{N}}^*}{\partial w_k}|y\rangle; \\
\frac{\partial\hat{\mathcal{N}}}{\partial w_k} &= e^{-i\alpha_1\Xi_1\delta t} * e^{-i\alpha_2\Xi_2\delta t} \ldots (-i\Xi_k\delta t) e^{-i\alpha_k\Xi_k\delta t} \ldots e^{-i\alpha_n\Xi_n\delta t}; \\
\frac{\partial\hat{\mathcal{N}}^*}{\partial w_k} &= e^{i\alpha_1\Xi_1\delta t} * e^{i\alpha_2\Xi_2\delta t} \ldots (i\Xi_k\delta t) e^{i\alpha_k\Xi_k\delta t} \ldots e^{i\alpha_n\Xi_n\delta t}.
\end{aligned}
\tag{10}
$$

The derivatives in equation 10 can also be computed numerically using finite difference method as shown in [20].
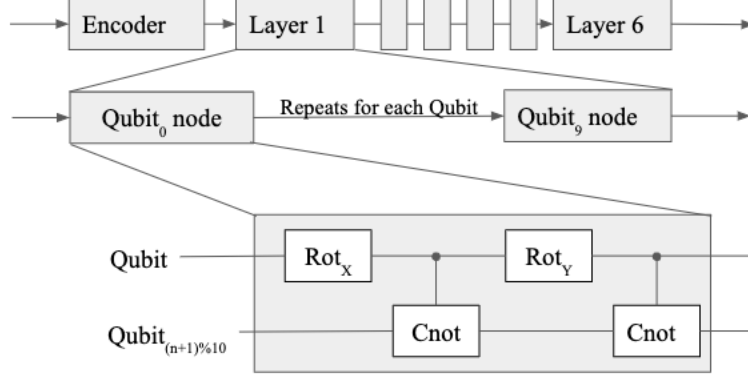
Figure 3: Schematic of the Variational Quantum circuit used for this case. Every layer is defined as a series of rotation gates in the x and y directions. Every qubit is linked to one other using CNOT gates.

4   Experiments

A fundamental model of QNNs has been defined in previous Section. In this section, we show how these can be converted to specific circuit models. The problem we tackled is classification of the MNIST handwritten digit database.

## 4.1   Variational Quantum Circuits on MNIST Dataset

The MNIST dataset contains 60,000 training samples and 10,000 testing samples. Each of the 28×28 grey scale images contains a single handwritten digit that the model must correctly identify. In order to encode 28×28 values in a quantum state, the model requires 10 qubits. So a high performance quantum circuit simulation library called QuEST has been used in this experiment. The only prepossessing that wasperformed on the images was to zero pad them to 32×32 to fill the 2^n quantum space. To model the quantum circuit, we arbitrarily choose a structure and fine tuned it based on the performance. This is similar to the notion of deciding the number of layers and nodes for classical ANNs. Therefore, the structure was defined in a completely in an ad-hoc fashion. The circuit consists of the image encoder followed by the modular structure shown in Figure 3. On a quantum computer, these probabilities could be estimated by running the circuit many times and forming a probability histogram. In our simulations, we measured these probabilities directly. We used the mean square error between the normalized probability and a one-hot representation of the label to calculate the loss as shown in Section 3.3.1. The weights were updated using mini-batch gradient descent, with a batch size of 10 images.

4.2   Results and Discussion

The proposed method experiments a number of trials based on number of quantum layers. The proposed method achieves 64% of the recognition accuracy, the highest among the current quantum neural network models. In this experiment, the learning rate set as $\rho = 0.03$ which decays by 0.99 every epoch. We see an increase in performance with increase in number of layers. The experiment results are laid out in Table 2. In Figure 4, the optimization process can be visualized as the evolution of network accuracy with every epoch. However, this was run in a simulation rather a real quantum system since there are very few accessible quantum computers with 10 usable qubits. This is an important point because simulating exponentially increasing dimensional Hilbert spaces in classical computers requires enormous amounts of resources. However, in an actual physical quantum computer most of the matrix transformations are computed naturally and require no computational power, potentially making QNN's much superior to classical ANN's in terms of performance.
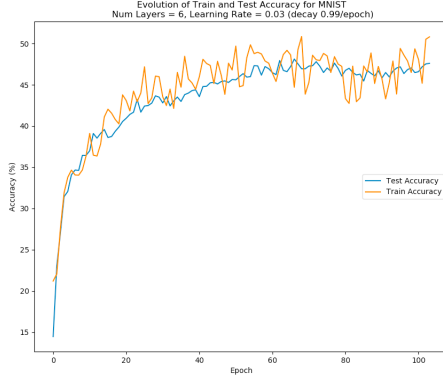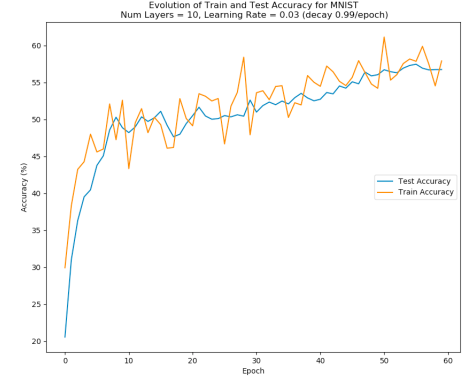
| | (a) | | | (b) | |

Figure 4: Visualizing the Training Process in relation to number of quantum layers. We see increased accuracy with number of layers. The gate complexity for each layer is $O(n)$ with n being number of quantum nodes in each layer.

Number of Layers	Train Accuracy (%)	Test Accuracy (%)	Convergence in (Epochs)
4	38.8	37.3	92
6	47.0	50.1	103
10	56.7	57.2	59
20	**64.08**	**64.74**	**10**

Table 2: Experimental Results.

5   Conclusion and Future Work

In this paper, we have presented a novel fundamental algorithm for for defining and training Neural Networks in Quantum Information based on time evolution and the Hamiltonian. A new deep learning based model has been introduced with more fundamental level and hence can be inherited by any variants of quantum computing models. In addition, we has proposed a new quantum backpropagation algorithm to train the new QNN model and validate this algorithm for the MNIST dataset on a quantum computer simulation. The future work will be highly focused on addressing the Information Loss of Quantum States due to measurement. This problem can be addressed by maximally entangling the quantum state. Another investigation will be conducted in running and testing the QNN model in short term Quantum Processors.

References

[1] Tyson Jones, Anna Brown, Ian Bush, and Simon Benjamin. Quest and high performance simulation of quantum computers. *arXiv:1802.08032*, 2018.

[2] Paul Beinoff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 1980.

[3] Manin Yu. Computable and uncomputable. *Sovetskoye Radio, Moscow*, 1980.

[4] Richard Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 1982.

[5] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society*, 1985.

[6] Lov K Grover. A fast quantum mechanical algorithm for database search. *arXiv preprint quant-ph/9605043*, 1996.

[7] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review 41.2*, 1999.

[8] Subhash C Kak. Quantum neural computing. In *Advances in Imaging and Electron Physics*, volume 94, pages 259–313. Elsevier, 1995.

[9] Tammy Menneer and Ajit Narayanan. Quantum-inspired neural networks. *Tech. Rep. R329*, 1995.

[10] Mitja Perus. Neuro-quantum parallelism in brain-mind and computers. *Informatica*, 20:173–183, 1996.

[11] T Meneer. Quantum artificial neural networks. Master's thesis, University of Exeter, 1998.

[12] M. V. Altaisky. Quantum neural network. *arxiv:quant-ph/0107012*, 2000.

[13] Sanjay Gupta and R. K. P Zia. Quantum neural networks. *Journal of Computer and System Sciences*, 2001.

[14] Vasile Palade, Robert J. Howlett, and Lakhmi Jain, editors. *Qubit Neural Network and Its Efficiency*. Springer Berlin Heidelberg, 2003.

[15] Sinayskiy I. Schuld, M. and F. Petruccione. The quest for a quantum neural network. *Quantum Information Processing,*, 2014.

[16] Zhaokai Li, Xiaomei Liu, Nanyang Xu, and Jiangfeng Du. Experimental realization of a quantum support vector machine. *Physical review letters*, 114(14):140504, 2015.

[17] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum deep learning. *arXiv preprint arXiv:1412.3489*, 2014.

[18] Maxwell P. Henderson Steven H. Adachi. Application of quantum annealing to training of deep neural networks. *arXiv:1510.06356*, 2015.

[19] Jason Rolfe Bohdan Kulchytskyy Mohammad H Amin, Andriyash Evgeny and Roger Melko. Quantum boltzmann machine. *Physical Review X 8, no. 2*, 2018.

[20] Nathan Killoran, Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *arXiv:1806.06871 [quant-ph]*, 2018.

[21] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *aPhys. Rev. Lett. 121, 040502*, 2018.

[22] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023–023, 2016.

[23] M. A. Nielsen and I. Chuang. *Quantum computation and quantum information.* Cambridge University Press, 2002.

[24] Arash Fereidouni Ghaleh Minab. Fastest scrambler. Master's thesis, University Of Minnesota, 2017.