University of Mumbai
2020-2021

M.H. SABOO SIDDIK COLLEGE OF ENGINEERING
8, Saboo Siddik Road, Byculla, Mumbai - 400 008
Department of Computer Engineering

Project Synopsis Report

On

"Neural Networks through quantum Perspective "

By

Mr. Faisal Sayed
Mr. Ashhar Shaikh
Mr. Faraz Shaikh

Guided By
Prof. Ahlam Ansari

M.H. SABOO SIDDIK COLLEGE OF ENGINEERING
8, Saboo Siddik Road, Byculla, Mumbai -400 008.

This is to certify that,

Roll No	Name
CS3117042	Faisal Sayed
CS3117052	Shaikh Md Ashhar
CS3117053	Faraz Shaikh

Of Final Year/Second year (B.E./S.E Semester VII/III) degree course in Computer Engineering, have completed the specified project synopsis on,

"Neural Networks through quantum Perspective"

As a partial fulfillment of the project work in a satisfactory manner as per the rules of the curriculum laid by the University of Mumbai, during the Academic Year July 2019 — June 2020.

Internal Guide

Internal Examiner

External Examiner

Head of Department

Principal

Abstract

Artificial neural networks, usually just called neural networks, arecomputing systems indefinitely inspired by the biological neural networks and they are extensive in both research and industry. It is critical to design quantum Neural Networks for complete quantum learning tasks. In this project we suggest a computational neural network model based on principles of quantum mechanics which form a quantum feed forward neural networks capable of universal quantum computation. This structure takes input as from one layer of qubits and passes that input onto another layer of qubits. This layer of qubits evaluates this information and passes on the output to the next layer. Eventually the path leads to the final layer of qubits.

The layers do not have to be of the same width, meaning they don't have to have the same number of qubits as the layer before or after it. This structure is trained on which path to take similar to classical artificial neural networks.

The proposed project can be summarized by the following points given below:

- The efficient training of the quantum neural network using the fidelity as a cost function, providing both classical and efficient quantum implementations.

- Use of methods that allows for fast optimization with reduced memory requirements.

- Benchmarking our proposal for the quantum task of learning an unknown unitary and find remarkable generalization behavior and a striking robustness to noisy training data.

List of Figures

Contents

Chapter 1: INTRODUCTION

1. Introduction

The concept of artificial neural network has been proposed around 1950s mainly to mimic the different activities of human brain. an artificial neural network (ANN) is a parallel distributed information processor made up of identical units (neurons) capable of storing information and make it available for use. Over the years, quantum computing has seen exceptional development which has a great impact on faster computing. Like artificial neural network (ANN), an unprecedented, useful and applicable concept has been proposed which is known as quantum neural network (QNN).

QNN has been developed combining the basics of ANN with quantum computation paradigm which is superior to the traditional ANN. quantum computers assure significant advantages over classical computers for a number of different applications. We show that the complete loss function landscape of a neural network can be represented as the quantum state output by a quantum computer. A quantum computer harnesses some of the almost-mystical phenomena of quantum mechanics to deliver huge leaps forward in processing power. quantum machines promise to outstrip even the most capable of todays—and tomorrows—supercomputers. Machine learning (ML), particularly applied to deep neural networks via the backpropagation algorithm, has enabled a wide spectrum of revolutionary applications ranging from the social to the scientific.

Regardless of quick theoretical and practical progress, ML training algorithms are computationally expensive and, now that Moore's law is faltering, we must anticipate a future with a slower rate of advance. QNN is being used in computer games, function approximation, handling big data etc. Algorithms of QNN are also used in modelling social networks, associative memory devices, and automated control systems etc. Different models of QNN has been proposed by different researchers throughout the world but systematic study of these models have not been done till date. Moreover, application of QNN may also be seen in some of the related research papers. As such, this paper includes different models which have been developed and further the implement of the same in various applications. In order to understand the powerfulness of QNN, few results and reasons are incorporated to show that these new models are more useful and efficient than traditional ANN.

Discovering a suitable set of weights for a neural network has become one of the most studied problems of modern machine learning. It has presented a significant challenge to computer scientists for whom few successful alternatives to back-propagation are available. It can be difficult to explore very large search spaces efficiently and, worse, optimization may converge to a local minima far from global optimum.

Basic quantum concepts:

Quantum bits are the fundamental units of information in quantum information processing in much the same way that bits are the fundamental units of information for classical processing. the space of possible polarization states of a photon is an example of a quantum *bit*, or *qubit*. A qubit has a continuum of possible values: any state represented by a unit vector $a\,|\uparrow\rangle + b\,|\rightarrow\rangle$ is a legitimate qubit value. the amplitudes a and b can be complex numbers, even though complex amplitudes were not needed for the explanation of the experiment. (In the photon polarization case, the imaginary coefficients correspond to circular polarization.) In general, the set of all possible states of a physical system is called the *state space* of the system. Any quantum mechanical system that can be modeled by a two-dimensional complex
Vector space can be viewed as a qubit. Such systems, called two-state quantum systems, include photon polarization, electron spin, and the ground state together with an excited state of an atom. the two-state label for these systems does not mean that the state space has only two states but it has infinitely many—but rather that all possible states can be represented as a linear combination, or superposition, of just two states Paul Dirac's bra / ket notation is used throughout quantum physics to represent quantum states and their transformations. the notation |.>, <.| j is called ''Dirac notation'' mainly used in quantum computation, which represents the standard notation for the states in the quantum mechanics given after the name of famous theoretical physicist Paul Dirac. |.> i is a *ket vector* which in general a column vector and <.| is a *bra vector* which is complex conjugate transpose of ket vector, represents a row vector. If we operate a matrix with ket vector, we get a ket vector again. Together bra and ket give an inner product that is combining <x| and |y> as <x|y> denotes an inner product of two vectors which always gives a scalar quantity.

As Moore's law meets its end, two new computing paradigms have been explored, *neuromorphic* and quantum *computers*. Quantum computing is based on quantum bits (or qbits) obeying the laws of quantum physics as opposed to the classical bits of today that are based on classical physics. Note that in physics the term classical is used to mean non-quantum and we use this terminology throughout. quantum machine learning aims to find an advantage in applying quantum computing to machine learning. Current research into quantum machine learning falls into one of two categories. Some quantum algorithms promise a revolution in machine learning in theory, but contain many gaps in their implementation in practice. In contrast, others are more realistic in their method, but struggle to justify a place amongst the well-established methods of machine learning.

In this paper, it is shown that a quantum computer can output a quantum state that represents the entire cost landscape for a given neural network. the method is shown to be versatile and has some remarkable properties, as the ability to generalize from very small data sets and a remarkable tolerance to noisy training data.

Chapter 2: LITERATURE REVIEW

2. Literature Survey

Over the past decades, the term has been used to describe a variety of ideas, ranging from quantum computers emulating the exact computations of neural nets, to general trainable quantum that bear only little resemblance with the multi-layer perceptron structure.

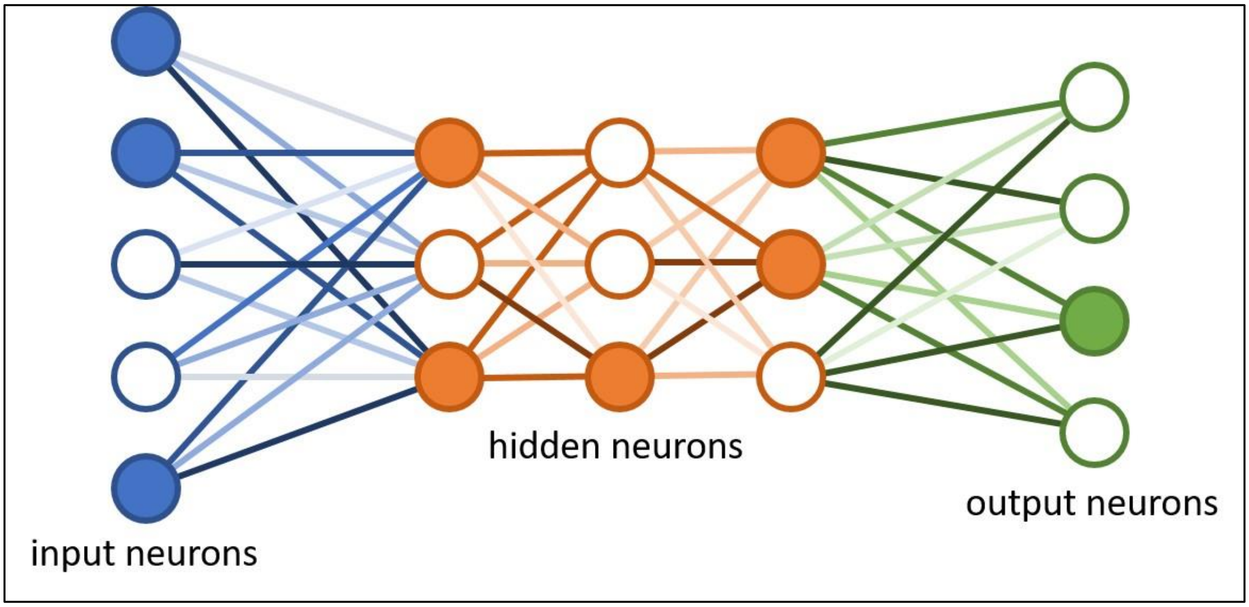


Figure 1 - NEURAL NET ARCHITECHTURE

Already in the 1990s, quantum physicists have tried to come up with “quantum versions” of recurrent and feed-forward neural networks. the models were attempts to translate the modular structure as well as the nonlinear activation functions of neural networks into the language of quantum algorithms. However, one could argue that chains of linear and nonlinear computations are rather “unnatural” for quantum computers. More recent research has tackled this problem, suggesting special measurement schemes or modifications of the neural nets that make them more amenable to quantum computing, but the advantage of these models for machine learning is still not conclusively established.

Abdulah Fawaz and Paul Klein’s research on Training and Meta-Training Binary Neural Networks with quantum computing show that quantum superposition can be used to represent many parameters of a neural network at once and efficiently encode entire loss and meta-training. As a training method it possesses significant advantages as it is landscape-independent, has a quadratic speedup over a classical search of the same kind, and would be able to solve statistically neutral problems such as parity problems

Chapter 3: AIM OBJECTIVE AND OUTCOME

3.3 Aim Objective and Outcome

Here we propose a truly quantum analogue of classical neurons, which form quantum feedforward neural networks capable of universal quantum computation. We describe the efficient training of these networks using the fidelity as a cost function, providing both classical and efficient quantum implementations. Our method allows for fast optimization with reduced memory requirements: the number of qubits required scales with only the width, allowing deep-network optimization. We benchmark our proposal for the quantum task of learning an unknown unitary and find remarkable generalization behavior and a striking robustness to noisy training data.

With the aim of building a fully quantum deep neural network capable of universal quantum computation we have found it necessary to modify the extant proposals somewhat. In this paper we define a quantum perceptron to be a general unitary operator acting on the corresponding input and output qubits, whose parameters incorporate the weights and biases of previous proposals in a natural way. Furthermore, we propose a training algorithm for this quantum neural network that is efficient in the sense that it only depends on the width of the individual layers and not on the depth of the network. It is also an important observation that there is no barren plateau in the cost function landscape. We find that the proposed network has some remarkable properties, as the ability to generalize from very small data sets and a remarkable tolerance to noisy training data.

Chapter 4: SCOPE

Scope

4.1 Justification:

A series of hurdles face the designer of a QML algorithm for quantum data. These include, finding the correct quantum generalization of the perceptron, (deep) neural network architecture, optimization algorithm, and loss function. In this paper we meet these challenges and propose a natural quantum perceptron which, when integrated into a quantum neural network (QNN), is capable of carrying out universal quantum computation. Our QNN architecture allows for a quantum analogue of the classical backpropagation algorithm by exploiting completely positive layer transition maps. We apply our QNN to the task of learning an unknown unitary, both with and without errors. Our classical simulation results are very favorable and suggest the feasibility of our procedure for noisy intermediate scale (NISQ) quantum devices, although one would still have to study how noise in the network itself influences the performance.

In this paper we define a quantum perceptron to be a general unitary operator acting on the corresponding input and output qubits, whose parameters incorporate the weights and biases of previous proposals in a natural way. Furthermore, we propose a training algorithm for this quantum neural network that is efficient in the sense that it only depends on the width of the individual layers and not on the depth of the network. It is also an important observation that there is no barren plateau in the cost function landscape. We find that the proposed network has some remarkable properties, as the ability to generalize from very small data sets and a remarkable tolerance to noisy training data.

4.2 Product scope description:

The Network Architecture

The smallest building block of a quantum neural network is the quantum perceptron, the quantum analogue of perceptrons used in classical machine learning. In our proposal, a quantum perceptron is an arbitrary unitary operator with *m* input qubits and *n* output qubits. Our perceptron is then simply an arbitrary unitary applied to the $m + n$ input and output qubits which depends on (2m+n)2−1(2m+n)2−1 parameters. the input qubits are initialised in a possibly unknown mixed state ρ^{in} and the output qubits in a fiducial product

state $|0\cdots0\rangle\mathrm{out}|0\cdots0\rangle\mathrm{out}$ (note that this scheme can easily be extended to qudits). For simplicity in the following we focus on the case where our perceptrons act on m input qubits and one output qubit, i.e., they are $(m + 1)$-qubit unitaries
Now we have a quantum neuron which can describe our quantum neural network architecture. Motivated by analogy with the classical case and consequent operational considerations we propose that a QNN is a quantum circuit of quantum perceptrons organized into L hidden layers of qubits, acting on an initial state ρ^{in} of the input qubits, and producing an, in general, mixed state ρ^{out} for the output qubits according to

$$\rho^{\mathrm{out}} \equiv \mathrm{tr}_{\mathrm{in,hid}}\left(\mathcal{U}(\rho^{\mathrm{in}} \otimes |0\cdots0\rangle_{\mathrm{hid,out}}\langle 0\cdots0|)\mathcal{U}^{\dagger}\right)$$

Where,
$U\equiv U\mathrm{out}ULUL-1\cdots U1$ is the QNN quantum circuit,
U^{l} are the layer unitaries, comprised of a product of quantum perceptrons acting on the qubits in layers $l - 1$ and l.
It is important to note that, because our perceptrons are arbitrary unitary operators, they do not, in general, commute, so that the order of operations is significant.

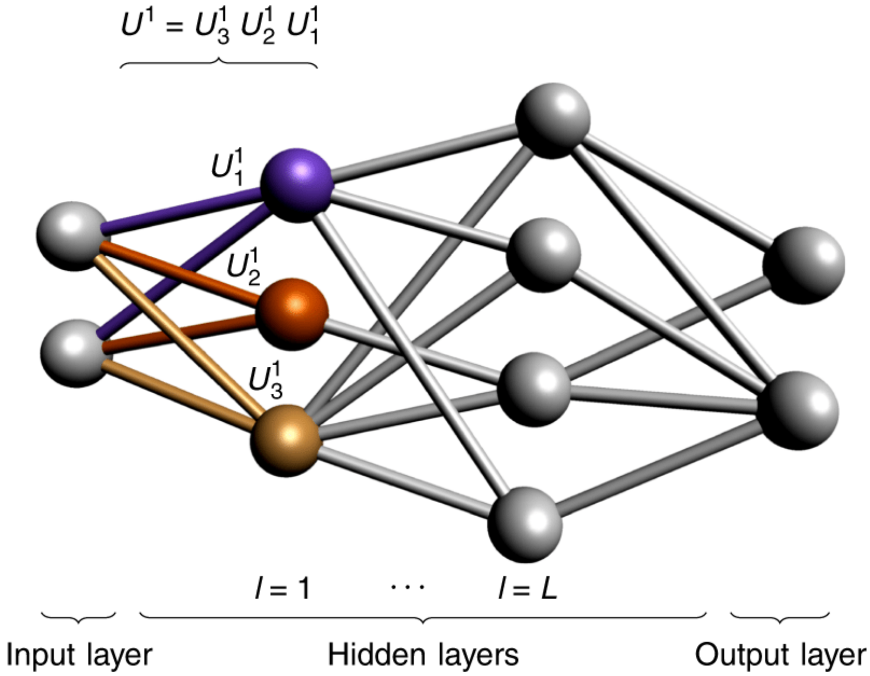


Figure 2- Feed Forward QNN architecture

A quantum neural network has an input, output, and L hidden layers. We apply the perceptron unitaries layer-wise from top to bottom (indicated with colors for the first layer): first the violet unitary is applied, followed by the orange one, and finally the yellow one.

4.3 Acceptance criteria

The cost function takes a slightly more complicated form when the training data output states are not pure, which may occur if we were to train our network to learn a quantum channel. the cost function varies between 0 (worst) and 1 (best).

4.4 Deliverables

- To minimize cost function for QNN
- Achieve generalization of quantum neural network against noisy (random) pairs and evaluate corresponding cost function for it.
- Once generalized, check the robustness of QNN to noisy data.

4.5 Assumptions

- A qubit cannot be copied like a classical bit
- Computer has high performance GPU such as NVIDIA 1080TI
- The quantum computing library has density matrices and ket states for quantum operations.
- Computer has dedicated RAM for training the QNN.

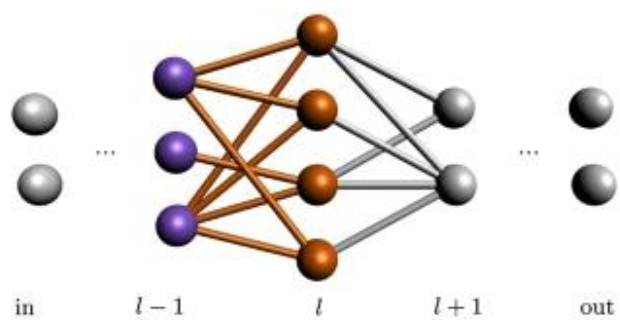
Chapter 5: PROPOSED SYSTEM

5. Proposed System

5.1 Analysis/Framework/Algorithm

1. Initialize:
Choose the initial U_j^l randomly for all j and l.

2. Feedforward: For every training pair $\left(|\phi_x^{\text{in}}\rangle, |\phi_x^{\text{out}}\rangle\right)$ and every layer l, perform the following steps:
2a. Apply the channel $\mathcal{E}^l$ to the output state of layer $l-1$: Tensor ρ_x^{l-1} with layer l in state $|0\dots0\rangle_l$ and apply $U^l = U_{m_l}^l \dots U_1^l$:



2b. Trace out layer $l-1$ and store ρ_x^l.

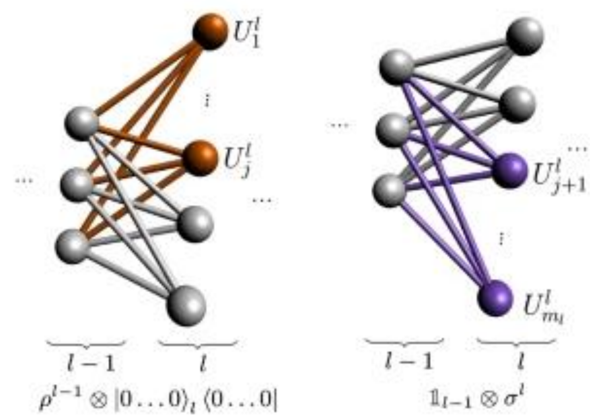
3. Update the network:
3a. Calculate the parameter matrices given by

$$K_j^l = \eta \frac{2^{m_{l-1}}}{N} \sum_{x=1}^{N} \text{tr}_{\text{rest}} M_j^l$$

where the trace is over all qubits that are not affected by U_j^l, η is the learning rate and

$$M_j^l = \Big[\prod_{\alpha=j}^{1} U_\alpha^l \left(\rho_x^{l-1,l}\right) \prod_{\alpha=1}^{j} U_\alpha^{l\,\dagger}, \prod_{\alpha=j+1}^{m_l} U_\alpha^{l\,\dagger} \left(\mathbb{1}_{l-1} \otimes \sigma_x^l\right) \prod_{\alpha=m_l}^{j+1} U_\alpha^l \Big],$$

where $\rho_x^{l-1,l} = \rho_x^{l-1} \otimes |0\ldots0\rangle_l \langle 0\ldots0|$,
$\sigma_x^l = \mathcal{F}^{l+1}\left(\ldots\mathcal{F}^{\text{out}}\left(|\phi_x^{\text{out}}\rangle\langle\phi_x^{\text{out}}|\right)\ldots\right)$ and $\mathcal{F}^l$ is the adjoint channel to $\mathcal{E}^l$, i.e. the transition channel from layer $l+1$ to layer l. Below, the two parts of the commutator are depicted:



3b. Update each unitary U_j^l according to $U_j^l \rightarrow e^{i\epsilon K_j^l} U_j^l$.

4. Repeat: Repeat step 2. and 3. until the cost function reaches its maximum.

Figure 3- Training Algorithm

5.2 Details of Hardware & Software

5.2.1 Hardware Requirements

IBM quantum Computer – 16 GB RAM

5.2.2 Software Requirements

Language – Python 3.x

Python Libraries - scipy, qutip, time, random, matplotlib.pyplot

5.2.3 Technology Used

Quantum Computing

Neural Network

5.3 Design details

5.3.1 Flowchart

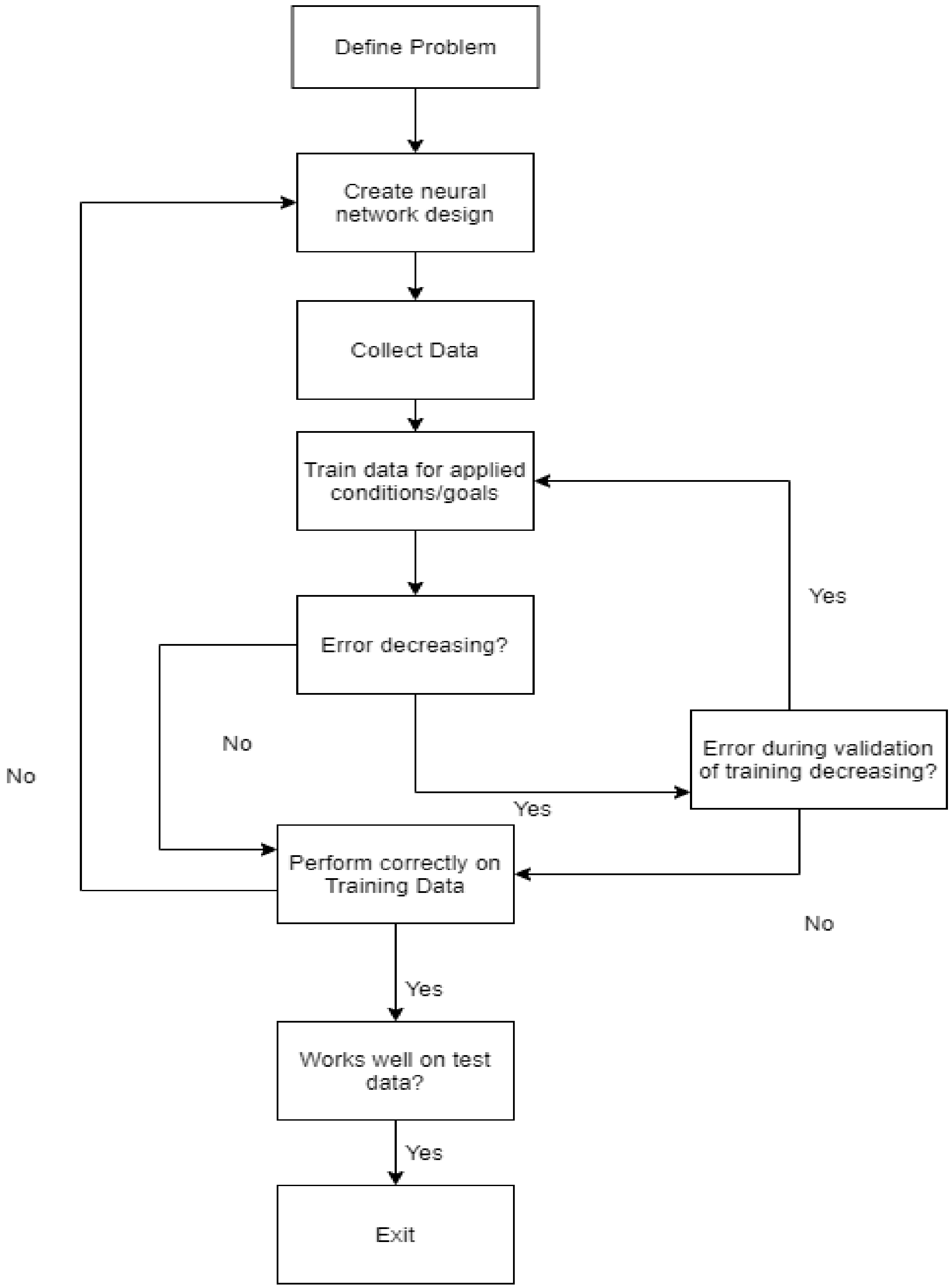


Figure 4- Flow chart

5.3.2 Simulation

It is impossible to classically simulate deep QNN learning algorithms for more than a handful of qubits due to the exponential growth of *Hilbert space*. To evaluate the performance of our QML algorithm we have thus been restricted to QNNs with small widths. We have carried out pilot simulations for input and output spaces of $m = 2$ and 3 qubits and have explored the behavior of the QML gradient descent algorithm for the task of learning a random unitary V.We focussed on two separate tasks: In the first task we studied the ability of a QNN to generalise from a limited set of random training pairs (|∣ϕinx⟩,V|∣ϕinx⟩)(|ϕxin⟩,V|ϕxin⟩), with $x = 1,\ldots, N$, where N was smaller than the Hilbert space dimension. the results are displayed. Here we have plotted the (numerically obtained) cost function after training alongside a theoretical estimate of the optimal cost function for the best unitary possible which exploits all the available information (for where n is the number of training pairs, N the number of test pairs and D the Hilbert space dimensions). Here we see that the QNN matches the theoretical estimate and demonstrates the remarkable ability of our QNNs to generalize.

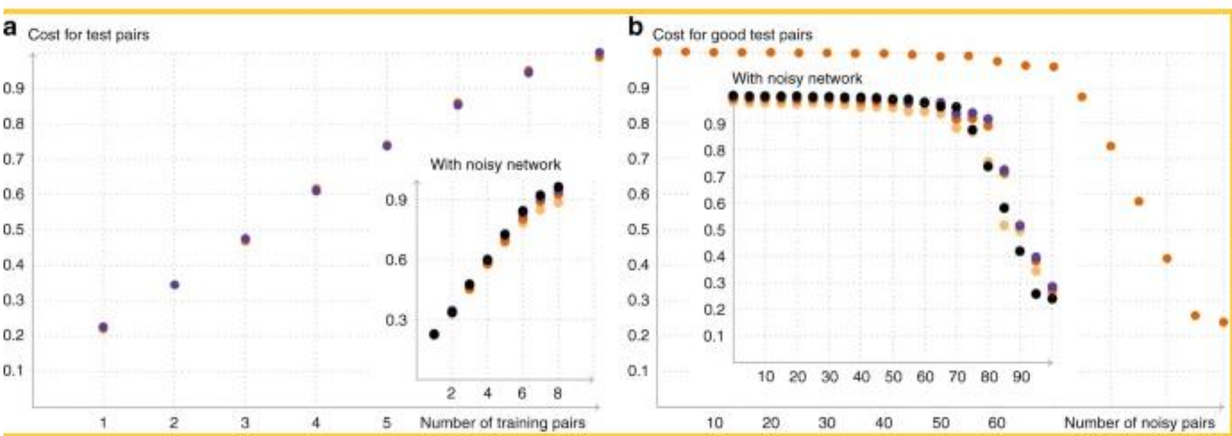


Figure 5- Simulation

In both plots, the insets show the behavior of the quantum neural network under approximate depolarizing noise. the colors indicate the strength t of the noise: black $t = 0$, violet $t = 0.0033$, orange $t = 0.0066$, yellow $t = 0.01$. and ***Panel (a)*** *shows the ability of the network to generalize.* We will be training a 3-3-3 network with for 1000 rounds with $n = 1$, 2. 8 training pairs and evaluated the cost function for a set of 10 test pairs afterwards. We averaged this over 20 rounds (orange points) and compared the result with the estimated value of the optimal achievable cost function (violet points)*.* ***Panel (b)*** *shows the robustness of the QNN to noisy data.* We trained a 2-3-2 network with for 300 rounds with 100 training pairs. In the plot, the number on the x-axis indicates how many of these pairs were replaced by a pair of noisy (i.e. random) pairs and the cost function is evaluated for all “good” test pairs.

Chapter 6: TIMELINE

6.Timeline

These two phases of operations are known as iteration. Neural Networks repeat the two steps until the desired output and accuracy is generated.

1. Training of networks: To train a network of data, we collect a large number of data and design a model that will learn the features. But the process is slower in case of a very large number of data.

2. Feature Extraction: After all the layers are trained about the features of the object, features are extracted from it and output is predicted with accuracy.

3. Evaluate Network: Once the network is trained, it can be evaluated.The network can be evaluated on the training data, but this will not provide a useful indication of the performance of the network as a predictive model, as it has seen all of this data before.

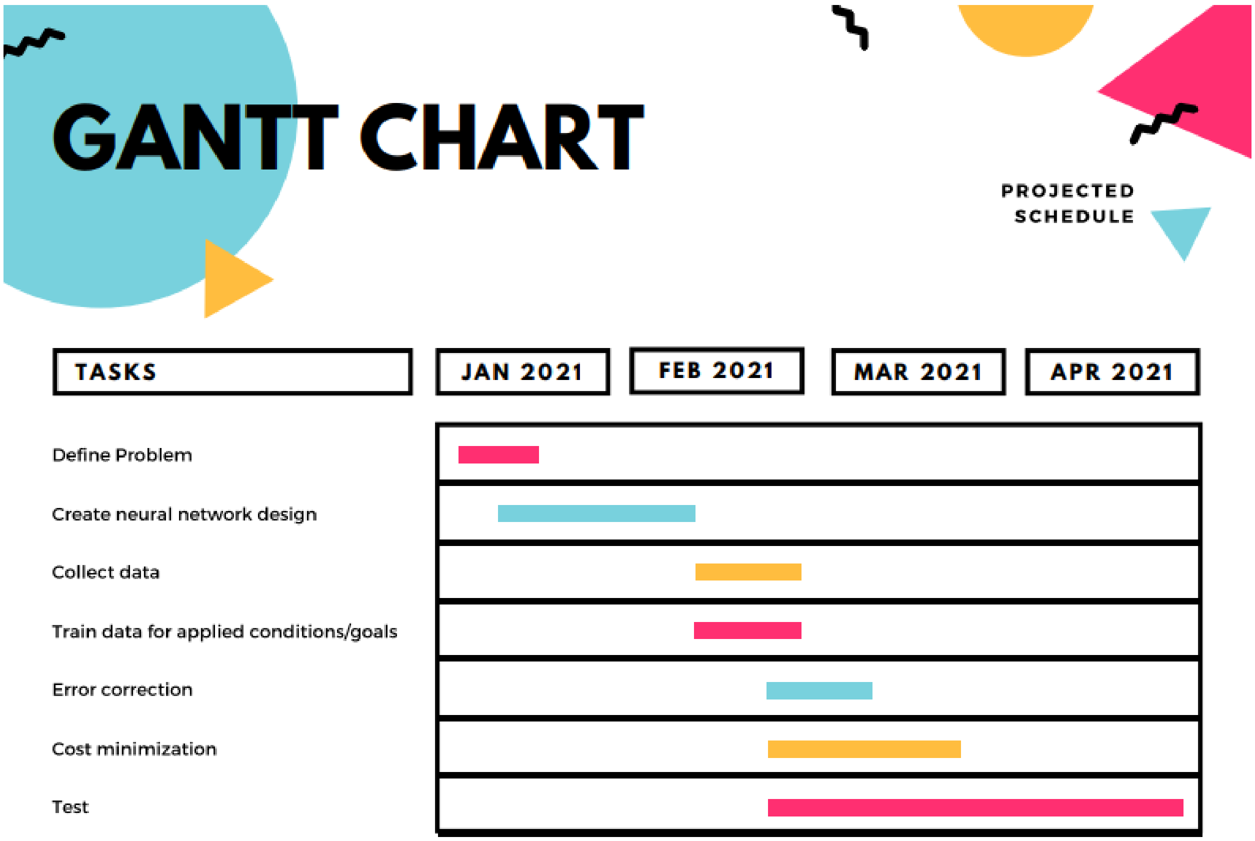


Figure 6- Gantt Chart

Chapter 7: EMPIRICIAL ANALYSIS

6. Empirical Analysis

Here are the various types of QNN model proposed till date to the best of our knowledge:

i). quantum feedforward neural network.
Ii). quantum Competitive Neural Network
iii). Quantum-Inspired Neural Network
iv). quantum Dot Neural Network
v). quantum Cellular Neural Network
vi). Qubit Neural Network
vii).Quantum Associative Neural Network

1) Abdullah Fawaz, Paul Klein, Simone Severini Peter Mountney (2019) - Training and Meta-Training Binary Neural Networks with quantum computing. They construct two toy problems, both of which are a binary classification on three binary features of eight data points corresponding to every 2^3 arrangement of those features. They construct a quantum circuit equivalent to the BNN, known as the quantum Binary Neural Network (QBNN),every operation in the implementation of a BNN is mapped to a quantum equivalent.

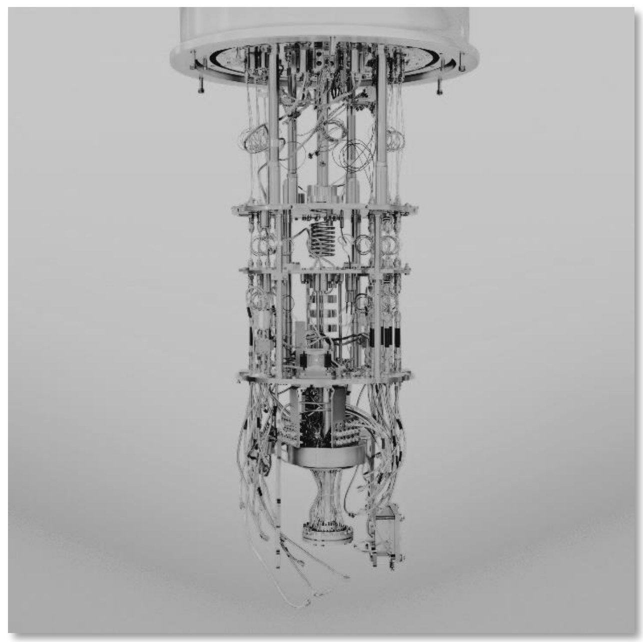


Figure 7- quantum COMPUTER

2) A quantum competition neural network (QCNN) has been presented by Zhong and Yuan . Because of the presence of the pseudo states, QCNN is capable of quantum memory. They have presented a competitive algorithm based on quantum concept and an application problem related to pattern recognition. With simulations,they concluded that the presented algorithm is far superior to its classical counterpart.Robert Salzmann,and Ramona Wolf.
3) Menneer and Narayanan proposed a hypothetical neural network model known as quantum inspired neural network. Introducing quantum rotation gates to the back propagation network, a novel QINN model has been proposed. quantum Inspired Neural Network has been discussed by different authors

It’s crucial to emphasize that it will probably be a long time before we have fault-tolerant quantum computers (having copies of qubit) for solving hard problems. Nevertheless, we can make significant progress in the near term by developing better methods and hardware for implementing quantum error correction, guided by relatively small-scale experiments with quantum error-correcting codes. We expect that, in the next few years, enhanced control of an error-corrected qubit will be demonstrated convincingly in the lab for the first time.

In traditional neural networks arguably, the best-known disadvantage is their “black box” nature. Simply put, you don’t know how or why your NN came up with a certain output. For example, when you put an image of a cat into a neural network and it predicts it to be a car, it is very hard to understand what caused it to arrive at this prediction. When you have features that are human interpretable, it is much easier to understand the cause of the mistake. By comparison, algorithms like decision trees are very interpretable. This is important because in some domains, interpretability is critical.

Chapter 8: CONCLUSION

5. Conclusion

the network architecture enables a reduction in the number of coherent qubits required to store the intermediate states needed to evaluate a QNN. Thus we only need to store a number of qubits scaling with the width of the network. This remarkable reduction does come at a price, namely, we require multiple evaluations of the network to estimate the derivative of the cost function. In this paper we have introduced natural quantum generalizations of perceptron and (deep) neural networks, and proposed an efficient quantum training algorithm. the resulting QML algorithm, when applied to our QNNs, demonstrates remarkable capabilities, including, the ability to generalize, tolerance to noisy training data, and an absence of a barren plateau in the cost function landscape. There are many natural questions remaining in the study of QNNs including generalizing the quantum perceptron definition further to cover general CP maps (thus incorporating a better model for decoherence processes), studying the effects of overfitting, and optimized implementation on the next generation of NISQ devices.('Noisy Intermediate-Scale Quantum' devices)

REFERENCES

References:

[1] Abdulah Fawaz, Paul Klein, Simone Severini, Peter Mountney - Training and Meta-Training Binary Neural Networks with quantum computing – **Journal KDD '19, August 4–8, 2019, Anchorage, AK, USA**

[2] A Survey on quantum computing technology - **Journal Gyongyosi, Laszlo, Imre, Sandor - February,2019**

[3] Wikipedia contributors. quantum machine learning—Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php? title=Quantummachinelearning&oldid=883275117 (2019)

[4] Altaisky, M. V. quantum neural network. Preprint at https://arxiv.org/abs/ quant-ph/0107012 (2001).

[5] Carleo, G. & Troyer, M. Solving the quantum many-body problem with **artificial neural networks. Science 355, 602–606 (2017)**

[6] Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning **(MIT Press, 2016).**

[7] Schuld, M., Bocharov, A., Svore, K. & Wiebe, N. Circuit-centric quantum classifiers. Preprint at https://arxiv.org/abs/1804.00633 (2018).

Articles :

[1] *Arrazola, J. M. et al.* Machine learning method for state preparation and gate synthesis on photonic quantum computers. *quantum* **Sci. Technol. 4, 024004 (2019).**

[2] McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscape**. Nat. Commun. 94812 (2018)**

[3] Arute, F. et al. quantum supremacy using a programmable superconducting processor. **Nature 574, 505–510 (2019)**

[4] Sasaki, M. & Carlini, A. quantum learning and universal quantum matching **machine. Phys. Rev. A 66, 022303 (2002)**

[5] Goncalves, C. quantum neural machine learning: backpropagation and dynamics. NeuroQuantology 15, 22–41 (2016)

[6] Verdon, G., Pye, J. & Broughton, M. A universal training algorithm for quantum deep learning. **Preprint at https://arxiv.org/abs/1806.09729 (2018).**

[7] Monràs, A., Sentís, G. & Wittek, P. Inductive supervised quantum learning. **Phys. Rev. Lett. 118, 190503 (2017).**

[8] Grant, E. et al. Hierarchical quantum classifiers. npj quantum **Inf. 4, 65 (2018).**

[9] Torrontegui, E. & García-Ripoll, J. J. Unitary quantum perceptron as efficient universal approximator **EPL 125, 30004 (2019).**

Acknowledgement

We wish to express our sincere thanks to our director Dr. Mohiuddin Ahmed and our principal Dr. Ganesh Kame, M.H. Saboo Siddik College of Engineering for providing us all the facilities, support and wonderful environment to meet our project requirements.

We would also take the opportunity to express our humble gratitude to our Head of Department of Computer Engineering Dr. Zainab Pirani for supporting us in all aspects and for encouraging with her valuable suggestions to make our project success.

We are highly thankful to our internal project guide Prof. Ahlam Ansari whose valuable guidance helped us understand the project better, her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the project successfully.

We would also like to acknowledge with much appreciation the role of the staff of Computer Department, especially the Laboratory staff, who gave the permission to use the labs when needed and the necessary material to complete the project.

We would like to express our gratitude and appreciate the guidance given by other supervisors and project guides, their comments and tips helped us in improving our presentation skills.

Although there may be many who remain unacknowledged in this humble note of appreciation but there are none who remain unappreciated.