PATTERN STORAGE IN QUBIT ARRAYS USING ENTANGLEMENT AND QUANTUM ANNEALING

A Thesis by

Bathiya Samarakoon

Bachelor of Science, University of Sri Jayawardenepura, 2015

Submitted to the Department of Mathematics, Statistics, and Physics
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

May 2019

PATTERN STORAGE IN QUBIT ARRAYS USING ENTANGLEMENT AND QUANTUM ANNEALING

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Physics.

Elizabeth Behrman, Committee Chair

Terrance Figy, Committee Member

Jason W. Ferguson, Committee Member

Saideep Nannapaneni, Committee Member

DEDICATION

To my Father and Mother, I am truly grateful for your continued support, for without it, I would not have had the opportunity to finish another chapter of my education

"A physical law must possess mathematical beauty."
-Paul Dirac

ACKNOWLEDGEMENTS

First and foremost I would like to respectfully thank Professor Elizabeth Behrman for advising and supporting this research and encouraging me to develop the entire task. And I must thank to Nam Nguyen for helping me to conduct this research. I should respectfully mention here, Dr. Mathew Muether, and all my friends in the Physics department. I appreciate all your words of encouragement.

ABSTRACT

Multiqubit arrays can be prepared in any state by adjusting the parameters in the Hamiltonian that governs their time evolution. In this work we show that these arrays can be used for pattern storage and recall, using techniques of machine learning. Any shape of a character or letter can be encoded as a collection of line segments, represented by the pairwise entanglement between neighboring qubits. We did this in two ways: first, by training the real time evolution; and second, by training to a ground state as a quantum annealing (QA) algorithm, by lowering the effective temperature of the system. In the real time training we succeeded in creating the letters X, M, N, and O, in a four-qubit system, with 2.09 % RMS error. With the QA we trained fifteen different characters on a four- five-, and then six-qubit system, with comparable RMS errors. We also showed that the pattern storage was robust to both classical and quantum noise.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\|\psi\rangle$	Ket-Vector
$\langle\psi\|$	Bra-Vector
ρ	Density Operator
Pr	Probability
tr	Trace of a Matrix
E_F	Entanglement of Formation
σ_z	Pauli Matrix
C	Concurrence
L	Lagrangian
w	Weights of the Neural Network
R	Real Numbers
η	Learning Rate
$\hbar$	Planck Constant
H	Hamiltonian
κ	Tunneling Parameter
ζ	Coupling Constant
ϵ	Bias Parameter

CHAPTER 1

INTRODUCTION

Pattern recognition problem is one of the most challenging problems in artificial intelligence. Neural Networks have the ability to perform cognitive processes like pattern recognition and pattern storage in the human brain [24]. The main task of pattern recognition is to classify patterns into different classes. Generally, pattern recognition canbe split in to two main components, memorization and recall. When constructing a model, in order to store patterns in a classical way, it can be based on Hopfield Network [28]. However, other algorithmic approaches exist, such as Grover's algorithm, that utilize the principles of quantum computing to perform pattern recognition [25].

A Hopfiled network is single-layered and recurrent network: the neurons are fully connected. When using the classical Hopfield Network to solve optimization problems we must choose weight w_{ij} and external inputs. Therefore, we will find the global minima for the solution of the problem. So the minimization is based on gradient descent method [32]. Hopfield networks have a scalar value, related to each neuron. When any neuron changes its state, the energy of the network will be decreased [31,32]. If we update the network weight for store a pattern, the energy of the system either will be decreased or will remained same. The network that minimizes the energy becomes stable and reaches to the training pattern and stops evolving [32].

A neural network that contains a single artificial neuron is the perceptron, which is used to pattern classification. Quantum mechanically, the perception depends on the unitary evaluation of a density matrix [39] according to $\rho_{final} = U\rho^{input}U^{\dagger}$. If we consider any given N patterns, the input states can be prepared as $\rho_i^{in}; i = 1, .., N$. Now the input and the output are assumed to be independent. In this approach, the computation is depending on non-linear components but evolution is linear. In this way, when the error exceeds a threshold value, then it is considered as the information is not being stored in the Perceptron. The

patterns that can be obtained through devices (such as interference patterns) can be modified by Grover's search algorithm [14,16]. Also, there is another algorithmic application base on the Fourier transform, which has been developed to recognize a pattern in a rectangular shape of $N \times M$ array of unit cells [23].

Quantum Associative Memory is playing a vital role in Quantum Pattern recognition, and patterns are stored as basis states of memory quantum states [24,25]. However, the global optimum for Quantum Associative Memory can be obtained experimentally by Adiabatic Quantum Computing (AQC) [4,21,24]. In AQC, Hamiltonian $H(s)$ is controlled by the anneling parameter (s), $s = \frac{t}{T}$, $(0 < \frac{t}{T} < 1)$, where T is running time. According to AQC, the system reaches its ground state when it is set to anneal. Therefore, in AQC, the Hamiltonian is changed from the initial Hamiltonian is changed from the initial Hamiltonian H_i to the ground state of the final Hamiltonian.

To determine a state of a spin system, we would be able to control the individual spins and their couplings, and therefore we need a programmable artificial spin system [48]. There are few main implementations for the qubits and those are used in computations such as [35],

(1). Trapped ions : Ions in a magnetic trap, manipulated by lasers.

(2). Quantum dot : An artificial atom has an electronic and optical properties which is manipulated using electrical current.

(3). Optical lattices : Neutral atoms that are trapped in a optical cavities. These atoms are controlled by laser.

(4). Magnetic Nuclear Resonance (MNR) Quantum computers : Qubits are simulated by manipulating the spin state of molecule using magnetic resonance technique.

For a liquid state nuclear magnetic resonance system, the Hamiltonian as, an addition of the memory Hamiltonian, which is related to the knowledge of the stored pattern in the associative memory, and the Hamiltonian of the computational input with a weight factor (retrieval Hamiltonian). The memory Hamiltonian can be defined as the coupling strengths

between qubits. The retrieval Hamiltonian, which defines an external field, creates a metric that is proportional to the Hamming distance between the input state and the memory patterns. Similar patterns will take its lowest energy by changing the energy of the memory Hamiltonian. As classical pattern recognition, AQC represent minimization of the cost function [21,47,31,32].

If we consider a two qubit system, a quantum gate can be built by changing the coupling constant using an external field. Therefore, the system can be used to map from a given initial state (input state) to a specific output [2]. According to this work learning algorithm can be developed for learning of an entanglement witness for an input state. So, we use this method in real time storage. In our work, we use another algorithmic approach "Learning Quantum Annealing", which is the reverse mode of AQC [4]. In an adiabatic process, the conditions of a system are changed gradually, so as to allow the system to adopt its configuration. If the system starts in the ground state of an initial Hamiltonian, after the adiabatic change, it will end up in the ground state of the final Hamiltonian. Here, we are using an electrostatically coupled two-SQUID (superconducting quantum interference device) system as the model [3,4]. Instead of specifying the Hamiltonian, we can specify the desired state (the pattern) and find the Hamiltonian at the ground state , which gives the final state (reverse problem of the traditional annealing). We can show that the QNN outputs are robust to noise and decoherence.

CHAPTER 2

ENTANGLEMENT

Entanglement is one of the famous topics in Quantum Mechanics (QM). One of the most unusual and fascinating aspect of QM is the fact that any particle or system can become entangled. Let A and B be two quantum systems. Values of certain properties of system A are correlated with the values that those properties will assume for system B. The properties can become correlated even when the two systems are spatially separated. This situation is called "Spooky Action". The word "Entanglement" was introduced by E. Schrodinger in 1935. In the same year, Einstein, Podolsky, and Rosen (EPR) published a paper [45] titled "Can the Quantum mechanical description of reality be considered complete?" Actually, EPR designed to show that Quantum Theory is incomplete. Einstein and his colleagues were believing that properties of a physical system have definite value ("an objective reality"). Whether you observe the system or not a given property of the system has a sharply defined value before a measurement is made. However QM tells a different story [35,45,46].

2.1 Qubit

A qubit is a quantum system whose state lies in a two dimensional Hilbert space. For an example, a qubit is can be a spin $\frac{1}{2}$ particle, which has two states : the spin-up state $\binom{1}{0}$ which we label as $|0\rangle$, and the spin-down state $\binom{0}{1}$ which we label as $|1\rangle$. But quantum mechanics allows also a superposition of these states. Suppose that we have a qubit in the state $|\psi\rangle = |0\rangle$. QM tells that, prior to measure a property (let's say the position) of the system, it does not have a definite or sharply defined value. Assume that we want to measure X. We know that the state is in a superposition of the eigen-states of the X operators as,

$$|\psi\rangle = |0\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}} \tag{1}$$

The probability amplitude of $|\psi\rangle$ in the $|+\rangle = \frac{1}{\sqrt{2}}\binom{1}{1}$ and $|-\rangle = \frac{1}{\sqrt{2}}\binom{1}{-1}$ states is $1/\sqrt{2}$, so, measurement of X will find the system (probability) in $|+\rangle$, 50% of the time and in $|-\rangle$ 50% of the time. Actually, this is very similar to the Schrodinger's cat thought experiment.

$$|\psi\rangle = \begin{cases} |+\rangle\,; 50\% \\ |-\rangle\,; 50\% \end{cases}$$

After measurement the system does assume a definite state either $|+\rangle$ or $|-\rangle$. But, according to QM, before we take the measurement this is not the case. In addition, quantum mechanical operators may not commute. For example, momentum and position operator has a non zero commutator, $[X, P] = i\hbar$. This tells us that momentum and position of a quantum particle cannot be measured simultaneously. Let's consider the two quantum systems A and B again. We will denote the position and momentum for the particle in possession of A by X_A and P_A, respectively.

Now, EPR proposed,

1. The particles interact, then the particles spatially separate

2. There are no more interactions between the particles. We can even assume that they are so far away from each other that no signal (not even light) can connect them over the time span when measurement made.

EPR system has definite values for the following properties,

1. X_A-X_B

2. P_A+P_B

According to QM these parameters cannot be determined prior to measurement. If an observer measures the momentum of particle-A, he/she obtain definite value for momentum of A as P_A. Since $P_A + P_B$, the observer knows the exact value of P_B (momentum of particle-B) even though he/she has not made any measurement B-particle in any way. Therefore, EPR believed that because this observer can determine the values of momentum (or position) of particle-B, these properties have definite values regardless of whether or not we measure them (elements of reality). However, quantum theory tells us that the wave functions of each particle exist in superposition and each property does not have a definite value until

we measure it.

The argument of EPR centered on the fact that quantum mechanics violated either the principle of reality or principle of non-locality. Therefore QM must be incomplete and there should be a hidden variable(s) we do not know yet. Hence we have locality and realism.

Locality: Measurement of particle A in no way disturbs the state of spatially separated particle B.

Realism: The values of measurable properties of each particle are objectively real. They have definite values prior to measurement (regardless of whether or not they are observed).

In 1964 John S Bell came up with an inequality, is called Bell's inequality [44].

$$Pr(+a;+b) \leq Pr(+a;+c) + Pr(+c;+b) \tag{2}$$

Bell's calculations were based on Einstein's local realism. If this inequality is violated then QM is correct [46]. In 1982 Alain Aspects experimentally proved that QM violates the Bell's inequality [46].

When the system consist of two subsystems we say it is a bipartite system. The Hilbert space of the composite system is the tensor product of the Hilbert space that describe system of A and the Hilbert space that describes system of B

$$H = H_A \otimes H_B \tag{3}$$

2.2 Definition of Quantum Entanglement

Let Q_1, Q_2, Q_3 be quantum systems with underlying Hilbert spaces H_1, H_2, H_3 respectively. Then the global quantum system Q consisting of the quantum systems Q_1, Q_2, Q_3 is said to be entangled if its state $|\psi\rangle \in H = \oplus_{j=1}^{n} |H_j\rangle$ can not be written in the form

$$|\psi\rangle = \otimes_{j=1}^{n} |\psi_j\rangle \tag{4}$$

2.2.1 Composite System

The state space of a composite system is the tensor product of the states of the component of a physical system and it represents overall of the coupled and entangled particles of the system [27]. The state vector of a composite system can be written as,

$$|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B . \tag{5}$$

2.3 Density Matrix

The density matrix is an operator determined by the outer product of a state vector,

$$\rho = |\psi\rangle \langle\psi| . \tag{6}$$

If we can write the density matrix as above form, then it is called a pure state. Even this state is a pure state, the state vector could be in a superposition. Since, quantum systems are inherently probabilistic, a pure state can be considered as a pure ensemble. For a pure state has a following properties.

(1). $\rho^2 = \rho$

(2). $tr(\rho^2) = 1$

(3). $tr(\rho) = 1$

(4). $\rho^\dagger = \rho$

(5). $\langle\phi| \rho |\phi\rangle \geq 0$ (Positive semi-definite)

Not all the quantum states are pure states. If we write a state as a linear combination of projectors onto pure state , it is called as a mixed state. A mixed state describes a system of identical particles but portions are in different configurations. A mixed state can be written as,

$$\rho_{mixed} = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right| . \tag{7}$$

and it has following properties.

(1). $\rho^2_{mixed} \neq \rho_{mixed}$

(2). $tr(\rho_{mixed}) = 1$

(3). $tr(\rho^2_{mixed}) \leq 1$

(4). $\rho^\dagger_{mixed} = \rho_{mixed}$

(5). Positive semi-definite

This is not the general form of the density matrix. Any state of any quantum system whether pure or mixed, entangled or not, can be written in this form [37,38].

2.4 Entanglement of Formation

Entanglement is the potential for quantum states to exhibit correlations that cannot be accounted for classically. It's one of the main properties that allows quantum computer to be more powerful than a classical computer. Similarly, a mixed state, ρ, is entangled if it cannot be written as the convex combination of product (separable) states, i.e.

$$\rho \neq \sum_i p_i \rho^i_A \otimes \rho^i_B \quad 0 \leq p_i \leq 1, \quad \sum_i p_i = 1$$

A natural question arises is how can we quantify entanglement in a particular state? It turned out, there are multiple ways one can quantify or define a measure of entanglement on a particular state. For a pure state of 2 qubit, a natural entanglement measure is defined as the Von-Neumann Entropy of either reduced density operator (subsystem) ρ_A or ρ_B, i.e.

$$E(|\psi\rangle) = -tr(\rho_A log_2 \rho_A) = -tr(\rho_B log_2 \rho_B). \tag{8}$$

This is equivalent to checking the mixed-ness of the reduced density operator. That is, if the reduced density operator, ρ_A or ρ_B is a completely mixed state, then $\rho = |\psi\rangle\langle\psi|$ is

a maximal entangled state; And if it's pure state, then ρ is separable (zero entangled) state. This is the direct result of the Schmidt decomposition.

Example: Consider the composite pure state

$$\psi = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

then

$$|\psi\rangle\langle\psi| = \rho = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Consider, ρ^A, the reduced density operator, by taking the partial trace of ρ over B:

$$Tr_B(\rho) = \frac{1}{2}\begin{bmatrix} Tr\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & Tr\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ Tr\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} & Tr\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{I}{2} = \rho^A$$

Note that ρ^A is a completely mixed state, since

$$Tr\left[\left(\rho^A\right)^2\right] = \frac{1}{2}$$

Thus, we can say that ψ is the maximal entangled state. One would get an identical result from computing the Von-Neumann entropy of the reduced density operator,

$$E(\psi) = -Tr\left(\frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\log_2\left(\frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)\right) = 1$$

Note, the entanglement value of a state is always between 0 and 1.

For a mixed state of a bipartite system, defining the entanglement value as the Von-Neumann entropy of the reduced density operator is no longer valid, because each reduced system can now have a non-zero entropy on its own even if there is no entanglement. However, this question has been answer fully by W.K Wootters [27]. Wootters defined an entanglement measure called 'Entanglement of Formation'. It can be thought as quantifying entanglement as a physical resources. Entanglement of Formation takes a fully entangled state, $|\psi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$, called the singlet state, as a basis, and defined entanglement of any other pure state by relating it to this basis state. In particular, it quantifies the number of basis states needed to create another state. For example, if it takes 4 copies of the basis state, ψ, to create 9 copies of certain state ψ', then the entanglement value of ψ' is $\frac{4}{9}$. This can be generalize to mixed state, ρ, in the following way,

$$E(\rho) = min \sum_{i=1}^{M} \rho_i E(|\psi_i\rangle) \tag{9}$$

.

The above formula states that the entanglement of a mixed state ρ is the minimum of the average entanglement of its pure state decomposition. Note that this minimum is taken over all decomposition of ρ.

We can introduce the spin-flip transformation before do the further calculations. The entanglement defined as in equation (8), can be written as,

$$E(|\psi_i\rangle) = E_F(C(|\psi\rangle)) \tag{10}$$

where, C is defined as "concurrence" for pure state.

$$C(\psi_i) = |\langle\psi|\psi_{sf}\rangle| \tag{11}$$

where, $|\psi_{sf}\rangle = \sigma_y |\psi^*\rangle$.

The entanglement of formation is given by as a function of concurrence [42],

$$E_F(C) = -\frac{1}{2}(1+\sqrt{1-C^2})log_2(\frac{1}{2}[1+\sqrt{1-C^2}]) - \frac{1}{2}(1-\sqrt{1-C^2})log_2(\frac{1}{2}[1-\sqrt{1-C^2}]) \quad (12)$$

The function $E_F(C)$ monotonically increasing, and rages from 0 to 1 as C goes from 0 to 1 [27]. We use entanglement of formation in order to train the four qubit system for realtime storage.

CHAPTER 3

NEURAL NETWORK

A neural network is a network of units, designed to take inputs to produce outputs. These units are called as neurons. The neural network consist with only one neuron is a perceptron. If the network consist with more than one neuron,

(1). The signal propagates through the Network starting from the input units or,

(2). Propagates through the one way towards the output units or,

(3). Circulate in the network.

We can understand a neural network by a mathematical way as below.

A neural network can be considered as a directed multipartite composed graph consisting of layers $k = 0, ..., k$. The vertices (Apex) in layer k are connected to vertices in the following layer $k + 1$. Their vertices of the network are said to be "neurons". Each neuron computes a prime non-linear function $f_{w,\beta} : R^{\alpha} \to R$ for which the input values are given by neurons of the previous layer connected it. Where, $w \in R^{\alpha}$ and $\alpha \in R$ and w can be differ for each neuron.

Consider the example in Figure 1. This neural network consist of 3 layers. Suppose, this network takes input $x \in \{0, 1\}^m$ which is represented as the first layer of neurons. The values of neurons in the hidden layer and the output layer are given by the function $f_{w,\alpha} : R^{\alpha} \to [0, 1]$ evaluated on their output. The parameters $w \in R$ and $\alpha \in R$ are called **weights** and **bias**. (This can be differ from each other). Output of the network $F^n(x) \in R^n$ is given by the values of neurons in the last layer (see the Figure 2).

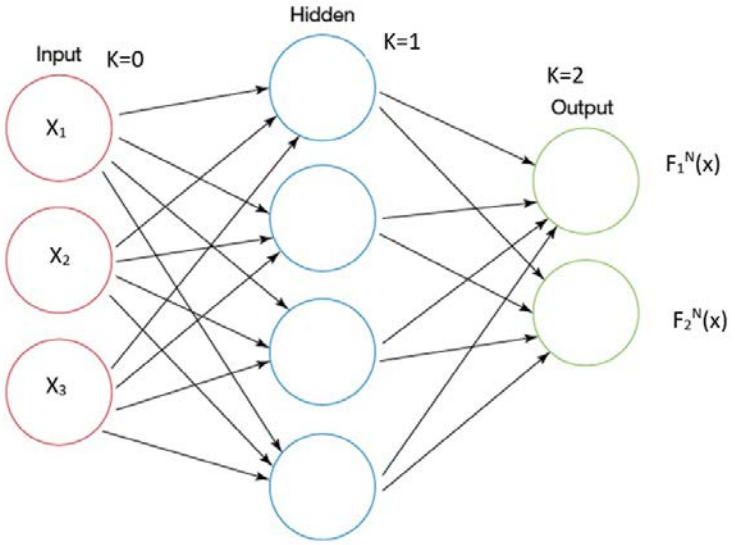

Figure 1: The hidden layers, input Layer and the output layer of a classical neural network.

The neurons in the first layer do not have any predecessors and their output is set to be the input of the network. $x \in \{0,1\}^m$. These values of the neurons in the last layer are interpreted as output of the network.

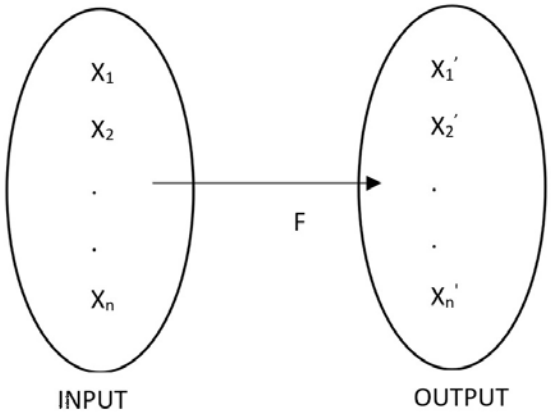

Figure 2: The inputs and outputs of the classical neural network. Here F is the unknown function to be approximated. The neural network gives an approximated function for the hidden layers of the neural network.

Above network describes a function $F : [0,1]^m \to [0,1]^n$. According to Figure 1, $k = 1$ is set to be the input layer and $k = k$ is set be the output layer. Rest of all $k = 2, .., k-1$ are said to be hidden layers (the internal structure of the network).

Then this neural network defined a linear map $ws_1 + \alpha$, where s is the input of the neuron. The function that each component has a form $f_{w,\alpha} : R^{\alpha}[0,1], y \mapsto \sigma(ws+\alpha)$, where, σ is the sigmoid function (discriminatory function).

$$[0,1]^n = [0,1] \times ....[0,1] \times .. = [0,1]^n \subset R^{\alpha} \tag{13}$$

Thus, neural network generates mapping from one space, to another space and sometimes from input space back to itself. Multilayered networks that associate vectors from one space for vectors of another space called "hetaroassociators". Simply, we can say, the neural network takes a vector in R^n and maps it in to a point in R^m ($f : R^n \to R^m$).

3.1 Perceptron

The simplest type of a neural network is said to be the perceptron (Figure 3), which is used to pattern classification. In this, the free parameters of the perceptron are adjusted.

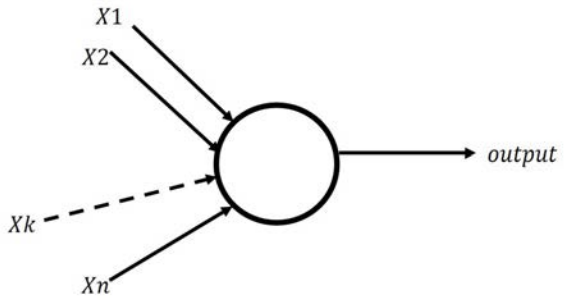

Figure 3: This perceptron has n inputs, here Xk implies the i-th input, and it has only one output .

The single Neuron in the perceptron produces +1 or -1 as the output , (as a binary classifier). It maps an input $x \in R^n$ to a binary values as,

$$f(x) = \begin{cases} 0, & \text{if } w^T x + \alpha > 0 \\ 1, & \text{otherwise} \end{cases}$$

where, w are the weights and α is the bias term. This function is said to be an

activation function of the Neural Network. In training of the perceptron, the weight vector and the bias term are obtained. However, if the training set is not linearly separable, the perceptron will not converge. The perceptron can be converged by minimizing the error term (cost function) .

$$E = \frac{1}{2}\sum(y_i - f(x_i))^2 \tag{14}$$

The weights and threshold values should be initialized. Weights can be initialized to a random value.

$$\frac{\partial E}{\partial w_j} = -(y_i - f(x_i))x_j \tag{15}$$

Now, change in w , Δw_j is proportional to $\frac{\partial E}{\partial w_j}$ and it gives,

$$\Delta w_j = \eta(y_i - f(x_i))x_i \tag{16}$$

where η is the learning rate.

3.2 Hopfield Network

A Hopfield network is an assemble of perceptron, which was implemented to solve XOR problem. Let's consider the Figure 4 given below. It shows a Hopfield network with 4 nodes and 4 input. In this circuit, neurons transmit signals back and forth to each other is a closed loop. In this network we assumed that weights of each interconnections are symmetric ($w_{ij} = w_{ji}$).

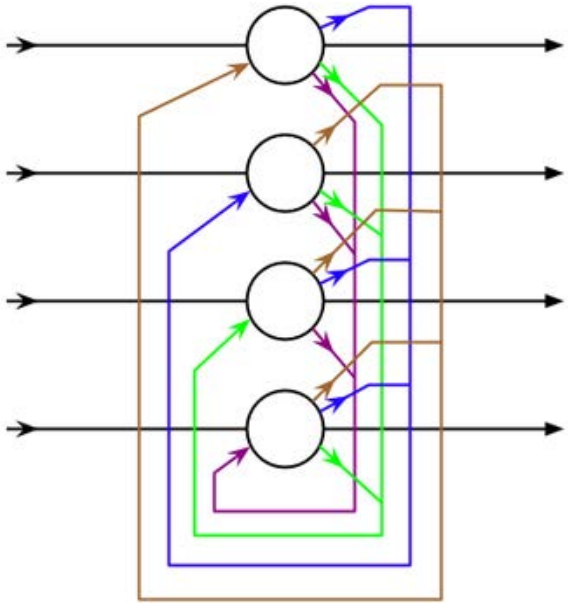

Figure 4: This Hopfield Network has 4 inputs and 4 nodes.

Each node has a state and it can be shown as,

$$s_i = \begin{cases} +1, & \text{if } \sum w_{ij} s_j \geq \theta_i \\ -1, & \text{otherwise} \end{cases}$$

where s_i is the state of a node and θ_i is a threshold value corresponding to the node. The nodes can be activated by either asynchronously and synchronously. In synchronously mode all the neurons are updated at the same time.

Hopfield network has a scalar value, related to each neuron. When a neuron changes its state, the energy of the network will be decreased.

$$E = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j + \sum_i \theta_i s_i \tag{17}$$

If we update the network weight to store a pattern, the energy of the system either decreases or remain the same. While the network minimizing energy, as it becomes stable, it reaches to the training pattern and stops evolving.

CHAPTER 4

REALTIME STORAGE

The main goal is to calculate the entanglement witness for each pattern. In realtime storage, the system is prepared in an initial state, and allowed to evolve for a fixed time. Then the measure is determined at the final time. The patterns are implemented on a four qubit system and here we are considering the map,

$$f : [0, 1]^{16} \mapsto [0, 1]. \tag{18}$$

This is a modification of the Quantum Neural Network's performance on the function approximation, for use in pattern storage.

4.1 Dynamic Learning for the Quantum Pattern Storage

A simple quantum gate can be built by using some properties of the interactions between two qubits. The interactions between these two qubits with the coupling can be externally adjusted as a function of time. In this work, we are dealing with a four qubit system and we use the knowledge [8] (bootstrapping) of the two qubit system as an extension of the future work in pattern storage. We can develop a general method for dynamic learning for a multiqubit system in real time. We can find values of the free parameters of the system such that the chosen measurement of the qubits are mapped to the desired functions, for example, classical or quantum logic gates. After that, we find those values such that the ground state of the system has certain desired qualities. Before the implementation of pattern storage, we will derive a dynamic learning technique starting with the density operator.

4.1.1 Time Evolution of Density Matrix

The time evolution of the density matrix is determined by the time evolution of each of the state vector $|\psi^{(\mu)}\rangle$. The time dependent density operator can be written as [38],

$$\rho(t) = \sum_{\mu=i}^{z} |\psi^{(\mu)}(t)\rangle \langle\psi^{(\mu)}(t)| \,. \tag{19}$$

An equation of motion for $\rho(t)$ can be determined by taking the time derivative of both side of equation.

$$\frac{\partial\rho(t)}{\partial t} = \sum_{\mu=i}^{z} \frac{\partial |\psi^{(\mu)}(t)\rangle}{\partial t} \langle\psi^{(\mu)}| \, t \, |\psi^{(\mu)}\rangle \frac{\partial \langle\psi^{(\mu)}(t)|}{\partial t} \tag{20}$$

However, since Schrodinger Equation (SE) gives us,

$$\frac{\partial |\psi^{(\mu)}(t)\rangle}{\partial t} = \frac{1}{i\hbar} H |\psi^{(\mu)}(t)\rangle \tag{21}$$

Now we can rewrite the equation (20) as,

$$\frac{\partial\rho(t)}{\partial t} = \frac{1}{i\hbar} \sum_{\mu=i}^{z} [H |\psi^{(\mu)}(t)\rangle \langle\psi^{(\mu)}(t)| - |\psi^{(\mu)}(t)\rangle H(\langle\psi^{(\mu)}(t)|)] \tag{22}$$

This is equal to,

$$\frac{\partial\rho(t)}{\partial t} = \frac{1}{i\hbar}(H\rho - \rho H) \tag{23}$$

Therefore, we will have the relationship between the Hamiltonian and the evolution of the desity matrix.

$$\frac{\partial\rho(t)}{\partial t} = -\frac{1}{i\hbar}[H, \rho] \tag{24}$$

This equation is known as Quantum Liouville equation [2]. However, ρ does not actually represent a physical observable. The Quantum Liouvulle equation can be solved formally as,

$$\rho(t) = e^{-\frac{iHt}{\hbar}} \rho(0) e^{\frac{iHt}{\hbar}}. \tag{25}$$

By introducing the Liouville operator ($\hat{L}$) as a a formal solution for the Quantum Liouville equation, $\hat{L} = \frac{1}{i\hbar}\big[\cdots, H\big]$, equation (23) can be rewritten as

$$\frac{\partial \rho}{\partial t} = -i\hat{L}\rho \tag{26}$$

which has a formal solution,

$$\rho(t) = e^{-i\hat{L}t}\rho(0). \tag{27}$$

The Hamiltonian for N qubit system an be written as,

$$H = \sum_{\alpha=1}^{N} K_\alpha \sigma_{x\alpha} + \varepsilon_\alpha \sigma_{z\alpha} + \sum_{\alpha \neq \beta = 1}^{N} \zeta_{\alpha\beta}\sigma_{z\alpha}\sigma_{z\beta}, \tag{28}$$

where $\{\sigma\}$ are the Pauli matrices related to each of the qubits ($\sigma_{xA} = \sigma_x \otimes I \ ... \ \otimes I$, for N qubit system there are (N-1) outer products), K are the tunneling amplitudes, ε are the biases, and ζ, the qubit-qubit coupling. In our calculations up state and down state are denoted by $|0\rangle$ and $|1\rangle$, respectively. For N qubit states, there are 2^N states, each labelled by a bit string each of whose numbers corresponds to the state of each qubit, in order. Above parameters in the system Hamiltonian are interactions in a physical qubit system such as SQUID arrays, and those those are adjustable according to changes in time.

Time evolution of the system is the function of the parameters $\{K, \epsilon, \zeta\}$. Therefore, if one of them (or more) is changed then the state also is changed because time is changing according to the equations (24) , (27), (28). There is an isomorphism between equation (29) and the equation for information propagation in a neural network

$$\phi_i = \sum_j \omega_{ij} f_j(\phi_i), \quad \phi_{out} = F_W \phi_{in} \tag{29}$$

where F_W the neural network operator and this is depending on neuron connecting weight matrix W, ϕ_{in} is the input vector and ϕ_{out} is the output vector of the Neural Network.

Note that $W \in C^n$ since our network takes on complex valued weights. The goal of learning in this context is to control the system via above parameters (K, ε, ζ) in order to calculate target output in response to given output. In a quantum neural computer, the input vector is; $\rho(0), t = 0$ initial density matrix when $t = t_f$ the final density matrix $\rho(t_f)$ will give the output. As we mentioned above, the parameters (K, ε, ζ) are the weights of the network. By changing these parameters, system can be changed to evolve the time to a particular final state (at $t = t_f$). So we use this set of parameter functions from generalization of three or four qubit systems. We use a quantum machine learning method using a quantum variant of backpropagation [2,8] in time to find these parameter functions that produce desired quantum states. These parameter functions map the input state to an approximated value of the entanglement of formation at the final time $\langle \sigma_{zA}(t_f)\sigma_{zB}(t_f)\rangle$.

A learning rule can be derived for a quantum system based on dynamic back propagation for time dependent networks.

$$L = \frac{1}{2}\big[d - \langle O(t_f)\rangle\big]^2 + \int_0^{t_f} \lambda^\dagger(t)\left(\frac{\partial \rho}{\partial t} - \frac{i}{\hbar}[\rho, H]\right)\gamma(t)dt. \tag{30}$$

Where L is Lagrangian should be minimized, d is target output and $\lambda^\dagger(t)$, $\gamma(t)$ are Lagrange multiplier vectors. Using above relationship, we want to develop a weight update rule based on gradient descent method.

The output of the neural network,

$$\langle O(t_f)\rangle = tr[\rho(t_f)O] = \sum_i p_i \langle \psi_i(t_i)| O |\psi_i(t_i)\rangle = Output. \tag{31}$$

Now we take the first variation of the Lagrangian with respect to the density matrix and set it equal to zero in order to obtain the learning rule. Therefore,

$$\delta L_{\delta\rho} = 0 \tag{32}$$

and then we can have,

$$\gamma_i \frac{\partial \gamma_j}{\partial t} + \frac{\partial \lambda_i}{\partial t}\gamma_j - \frac{i}{\hbar}\sum_k \lambda_k H_{ki}\gamma_j + \frac{i}{\hbar}\sum_k \lambda_i H_{jk}\gamma_k = 0 \tag{33}$$

with the boundary conditions at the final time t_f given by,

$$-[d - \langle O(t_f)\rangle]O_{ij} + \lambda_i(t_f)\gamma_j(t_f) = 0 \tag{34}$$

By using gradient descent method (see the Figure 5), we can minimize L by iteratively in direction of steepest as defined by the negative of the gradient.

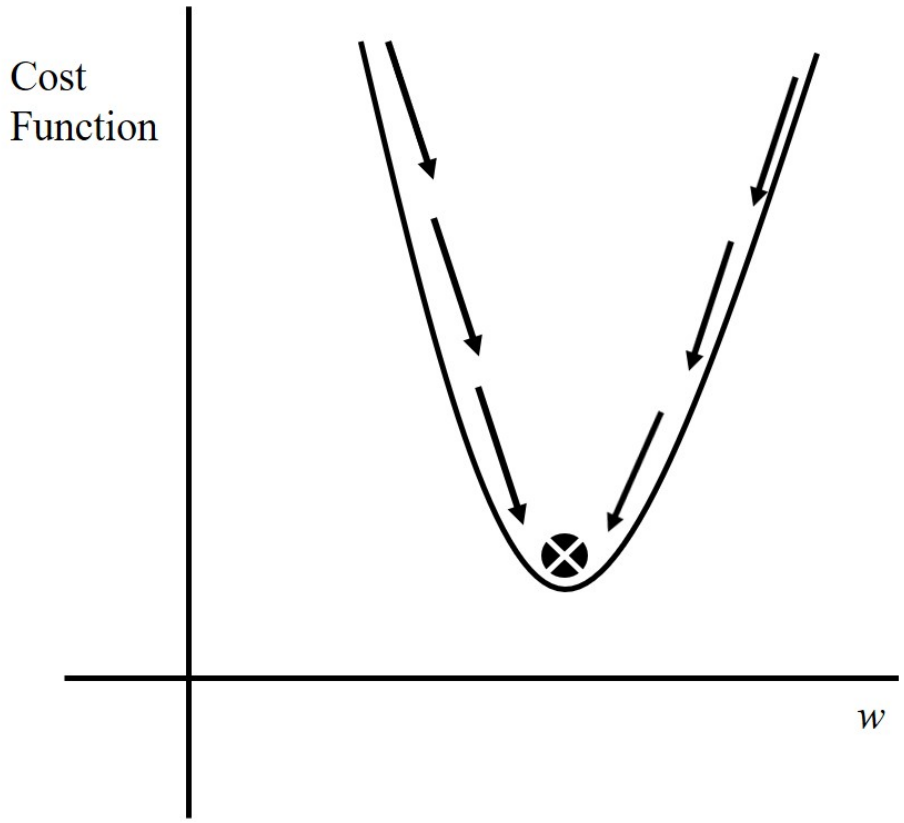

Figure 5: Gradient descent is an optimization algorithm used to minimize a given function by iteratively moving in the direction of steepest descent. The direction of steepest descent is defined by the negative of the gradient. In machine learning, gradient descent is used to update the parameters of the model.

Hence, we can derive the gradient descent learning rule to minimize the L with respect to w as below in equation 35,

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w} \tag{35}$$

where, η is the learning rate. If the learning rate is low, then training is more reliable but the optimization takes a lot of time. If the learning rate is high, then training may not converge or even diverge. Now we may write,

$$\frac{\partial L}{\partial w} = \frac{i}{\hbar} \int_0^{t_f} \lambda^{\dagger}(t) \left[\frac{\partial H}{\partial w}, \rho \right] \gamma(t) dt \tag{36}$$

Because of the Hermiticity of the Hamiltonian , H and the density matrix ρ, $\lambda_i \gamma_j = \lambda_j \gamma_i$ and $\frac{\partial L}{\partial w}$ given by the equation (39) will be real numbers,

$$\frac{\partial L}{\partial w} = \frac{i}{\hbar} \int_0^{t_f} \sum_{ijk} (\lambda_i(t) \frac{\partial H_{ik}}{\partial w}, \rho_{kj} \gamma_j - \lambda_i(t) \rho_{ik} \frac{\partial H_{kj}}{\partial w} \gamma_j) dt. \tag{37}$$

when we are considering the Hamiltonian of the four qubit (A, B, C, D) system, we can train the system for a symmetric measure. Therefore, we can observe the $K_A = K_B = K_C = K_D$, as well as for the ϵ and ζ parameters. But we have to distinguish among entanglement between qubits, A and B, between B and C, between A and D, between B and C, between B and D, and between C and D. Therefore, for four qubit system, there are six possible pairs.

We can use two qubit system to have values for entanglement witness by using the Wootters equation (equation 12) and those are the target values. However, for N qubit system, it should be trained on four input states for each of the $\binom{N}{2}$ pairs. The four input states are,

(1). The Bell State, $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T$. Entanglement = 1

(2). P- state, $\frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}^T$. Entanglement = 0.44

(3). Flat state, $\frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$. Entanglement = 0

(4). Product state, $\frac{1}{\sqrt{1.25}} \begin{bmatrix} 0 & 0 & 0.5 & 1 \end{bmatrix}^T$. Entanglement = 0

Now, we have $(6 \times 4) = 24$ training pairs (Figure 6). For example, the $Bell_{AB}$ state can be taken as,

$$Bell_{AB} = \frac{1}{\sqrt{2}}(|0000\rangle + |1100\rangle), \tag{38}$$

and was trained to give an output of 1 for the network output measure $O_{AB} = \langle \sigma_{zA}(t_f)\sigma_{zB}(t_f)\rangle^2$. And the other output measures $O_{\alpha\beta}$; $\alpha\beta \neq AB$ should be zero.

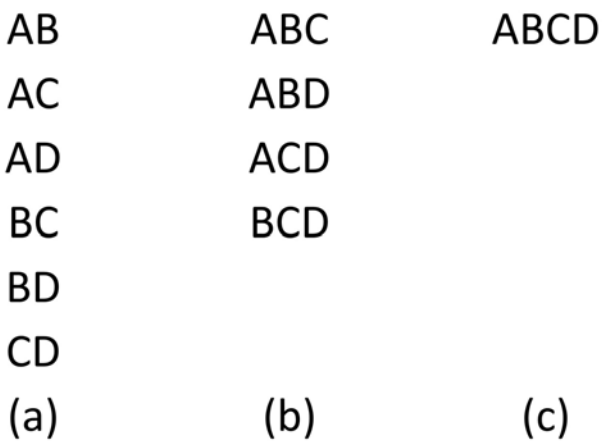

Figure 6: The possible entanglement states for a four qubit system. (a) There are 6 pairwise entanglements, which we are using in future calculations in realtime storage. (b) four 3-way entanglements. one GHZ (Greenberger-Horne-Zeilinger state) state.

For pattern storage, we find the parameters necessary such that a state in which the desired pattern, as represented by a set of line segments connecting entangled qubits, is stored via pairwise entanglements of those qubits (Figure 7).

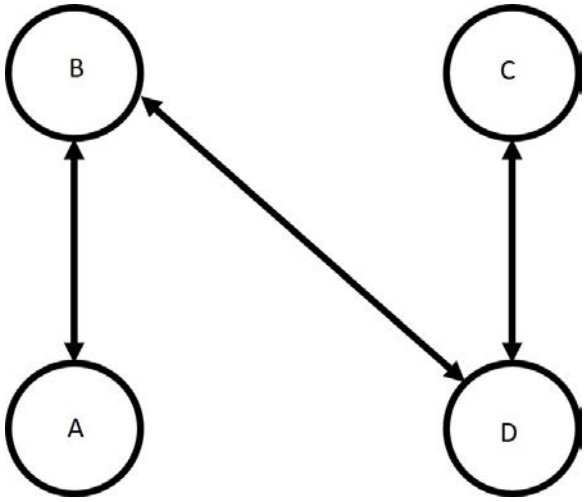

Figure 7: Encode the letter N using pairwise entanglement.

Now we can divide Letter "N" in to three segments as shown in figure 8. The Letter "N" is a combination of AB , BD and CD line segments.

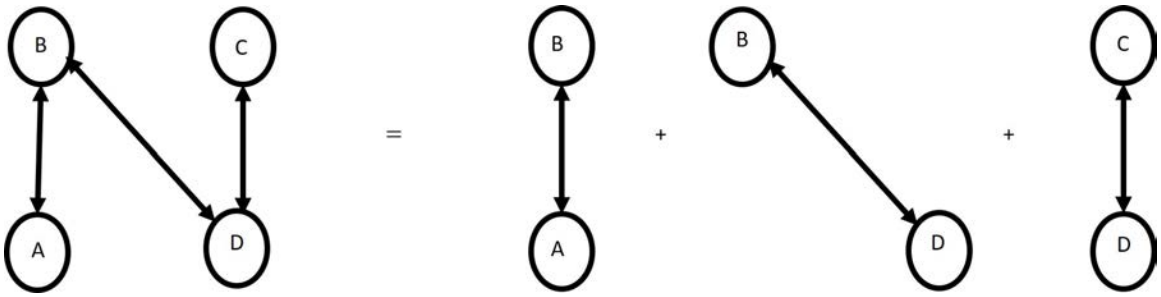

Figure 8: Encode the letter N using pairwise entanglement.

If we use Bell state to encode the letter "N", we can write it as,

$$|N\rangle = \frac{1}{\sqrt{6}}(|AB\rangle + |BD\rangle + |CD\rangle) \tag{39}$$

$$|N\rangle = \frac{1}{\sqrt{6}}(|0000\rangle + |1100\rangle + |0000\rangle + |0101\rangle + |0000\rangle + |0011\rangle) \tag{40}$$

After we trained the spin system, determining the parameter functions, those results can be used to test for other states. The Hamiltonian that we are determining in this approach , is said to be the Hamiltonian of the entanglement witness (H_{ew}). In the testing

process, we use H_{ew} to calculate the correlation between two qubits relate to the input state. For example, the correlation between A and B in letter "N" state is given by $tr(|N(t_f)\rangle \langle N(t_f)| M_{AB})^2$.

4.1.2 Training Results

Training of the four qubit system required 200 epochs (to train the entire training) and the learning rate η was taken as 6.2×10^{-6} with 251.46 ns total time. The asymptotic error was found as 2.09% as shown in Figure 9. Figure 10 shows the evolution of the qubit-qubit coupling as a function of time and we found that $\zeta(t_f = 251.4648ns) = 10^{-4}J$. We found that, the tunneling parameter at final time as $K(t_f = 251.4648ns) = 2.5 \times 10^{-3}J$, and Figure 11 shows the evolution of the tunneling parameter as a function of time. Finally, the evolution of energy bias is shown in Figure 12 and its value at final time was found as $\varepsilon(t_f = 251.4648ns) = 9.95 \times 10^{-5}J$.

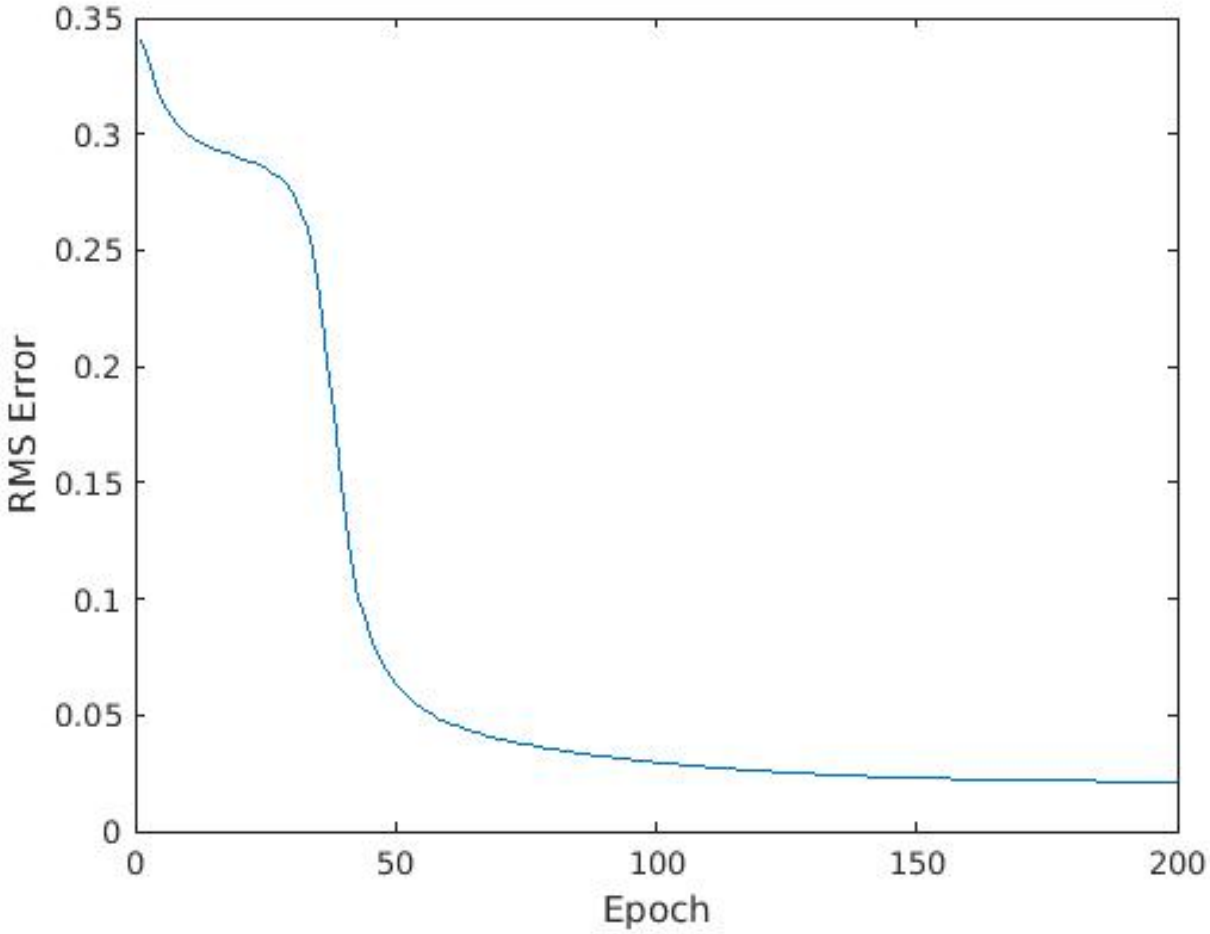

Figure 9: The mean square error of the training of the four qubit system as a function of epoch. The asymptotic error was 0.0209. The learning rate was $6.2 \times 10^{-6} J$.

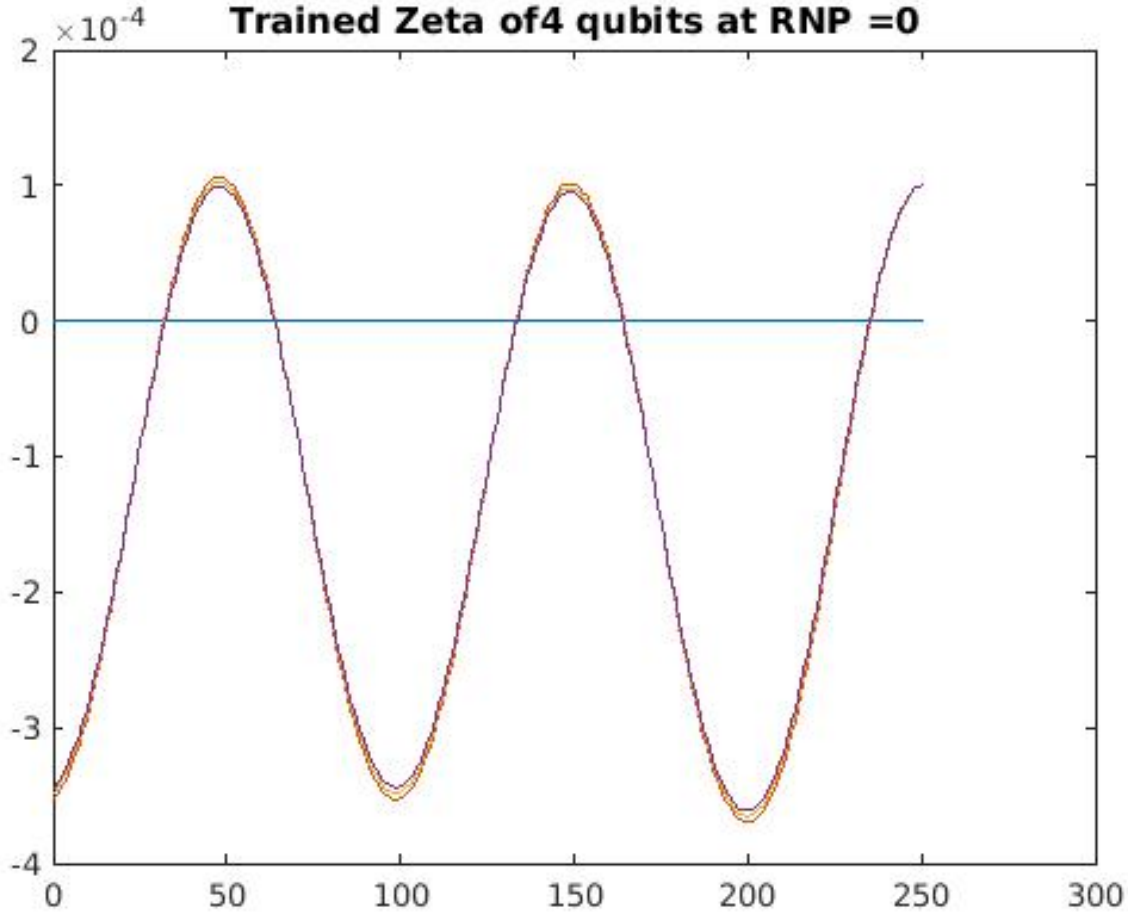

Figure 10: The trained parameter function ζ of four qubit system as a function of time. $\zeta = 10^{-4}J$. The total time was 251.4648 ns.

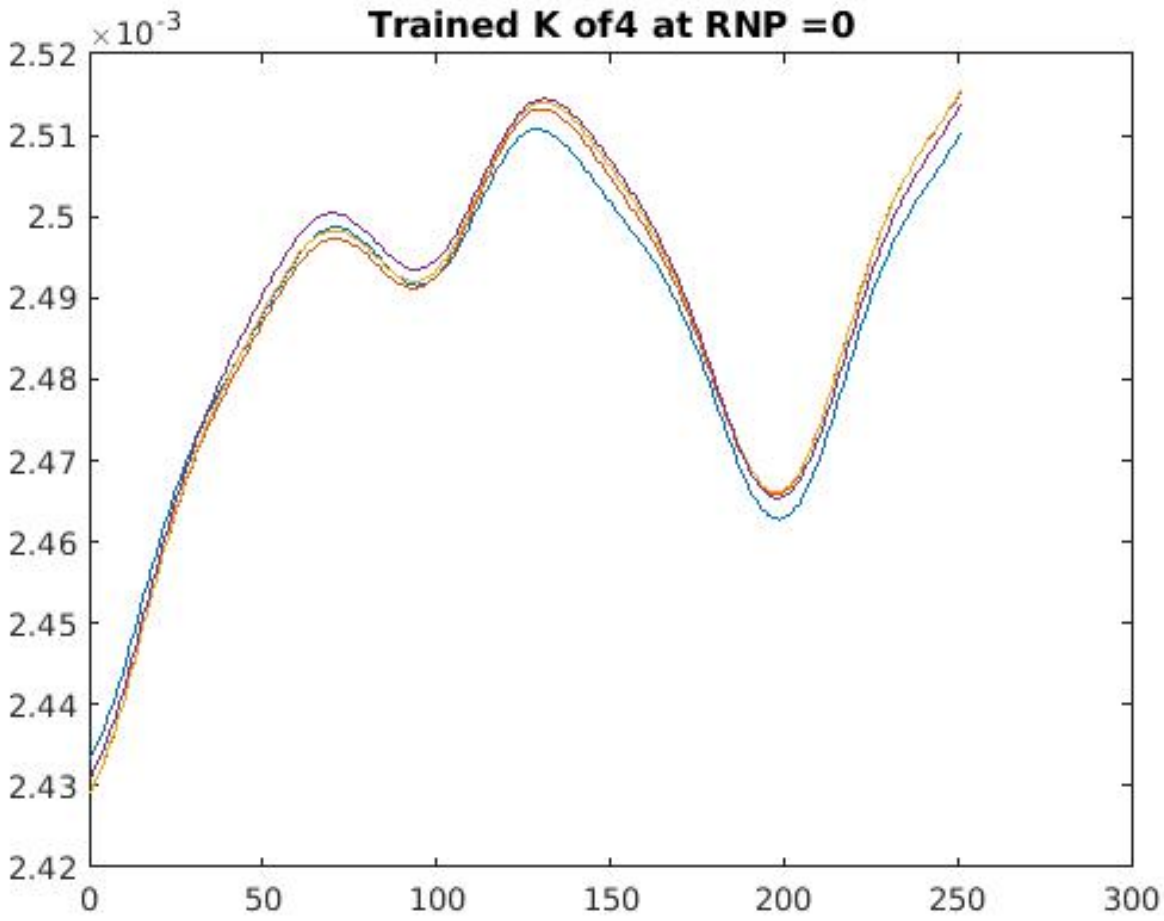

Figure 11: The trained parameter function K of four qubit system as a function of time. $K = 2.5 \times 10^{-3} J$. The total time was 251.4648 ns.

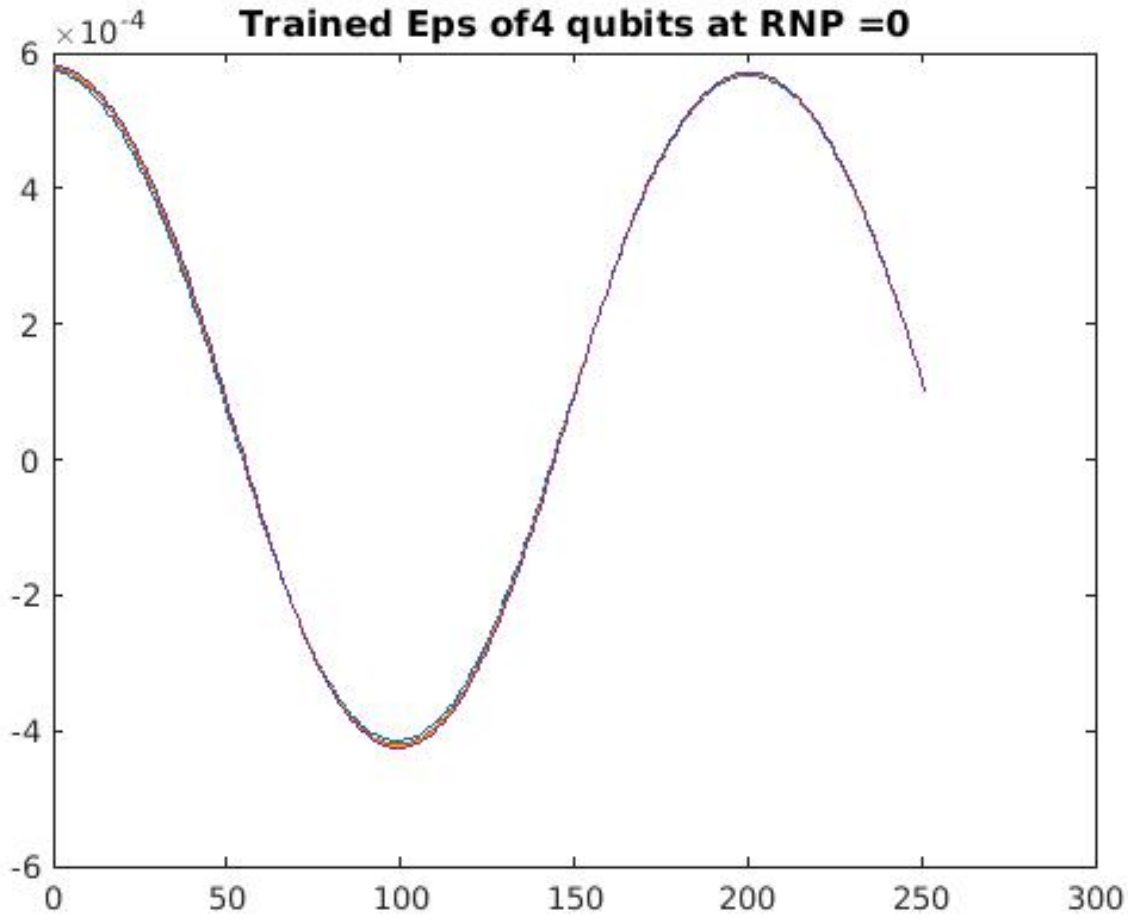

Figure 12: The trained parameter function of four qubit system as a function of time. $\varepsilon = 9.9556 \times 10^{-5} J$. The total time was 251.4648 ns.

CHAPTER 5

QUANTUM ANNEALING

In Quantum Annealing approach, we are storing patterns by reducing to the ground state of a system of interacting spins via using machine learning method as a solution for this computational problem. Quantum annealing is an effective technique of finding the global minimum of a given function. In Quantum Annealing, we start from "flat" state and allow system to anneal to desired state (pattern) via imaginary time evolution. Thus, the Hamiltonian of each pattern can be determined. Therefore, this is an inverse process of Adiabatic Quantum Computing (AQC). In previous chapter, we showed that the patterns can be stored using realtime evolution.

5.1 Superconducting Quantum Interference Device (SQUID)

To store patterns in a ground state we need a programmable quantum spin system which we could control individual spins and their couplings. The SQUID arrays are made by a superconducting wires, as an example, D-wave $2X^{TM}$, Niobium (Nb) conductors are used [49]. According to the Figure (13), there are two superconducting loops which are controlled by an external flux bias ϕ_{1x} and ϕ_{2x}. The lowest energy state (two minima) of the system are expressed by $|\uparrow\rangle$ and $|\downarrow\rangle$ and those are depending on the direction of the current in the loop. The energy gap between two minima is manipulated by ϕ_{1X} [48]. In this work we show that the patterns can be stored in the lowest energy configuration of a SQUID array.

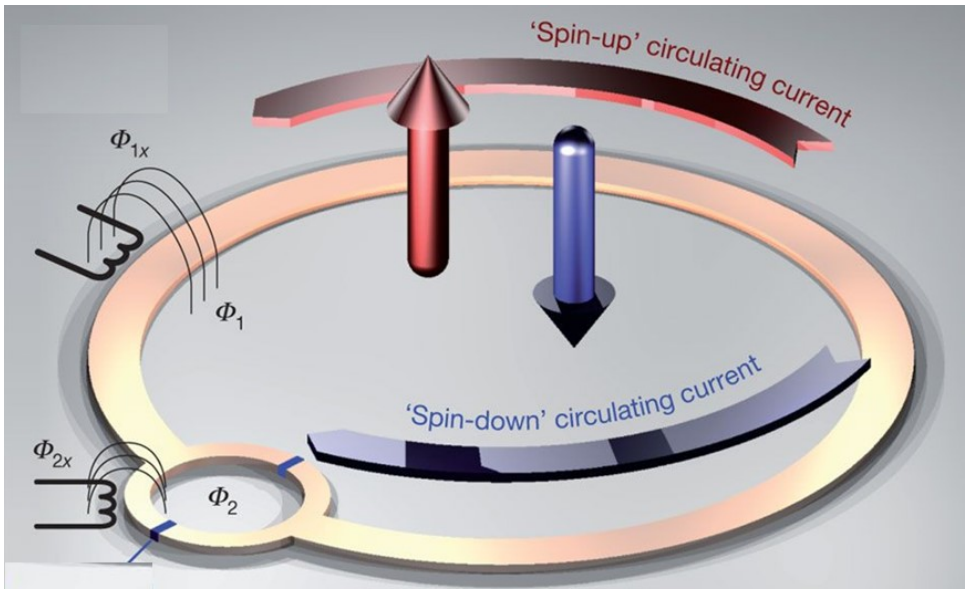

Figure 13: Superconducting Quantum Interference Device [48].

5.2 Adiabatic Quantum Computing

Adiabatic quantum computing is used to find the global minimum of a given function f, by slowly transforming the Hamiltonian at $t = 0$ to the problem Hamiltonian at $t = t_f$. The energy of the eigenstate of the Hamiltonian is minimized and therefore, the function f has domain $\{0,1\}^n$. Thus, in an adiabatic process, $f : \{0,1\}^n \to [0,\infty]$, where $min_x f(x) = f_{min}$ and $f(x) = f_{min}$ if and only if $x = x_0$. Lets' consider the Hamiltonian for a given below [28,30].

$$H(\lambda) = (1 - \lambda(t))H_0 + \lambda(t)H_f \tag{41}$$

According to the above transformation, the ground state of the initial Hamiltonian H_0 will slowly evolve to the ground state of H_f, by increasing λ (see the Figure 14).

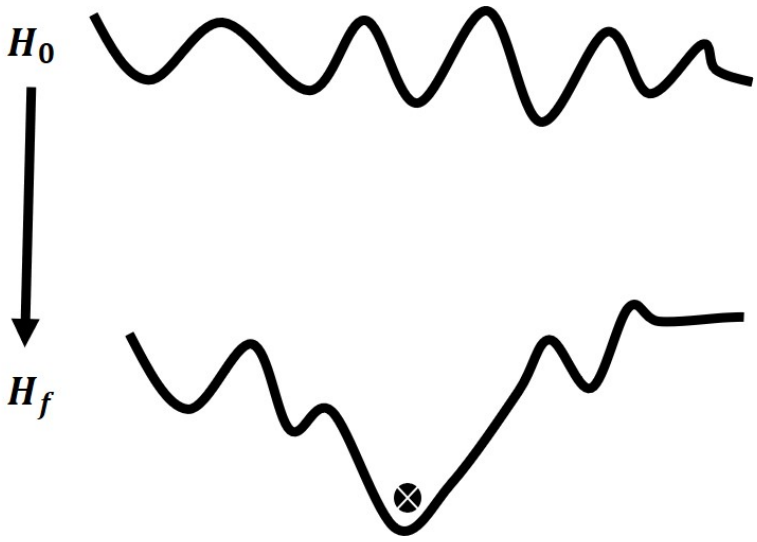

Figure 14: In Adiabatic Quantum Computing, the system starts in the ground state of an initial Hamiltonian and after the adiabatic changes, it will end up in the ground state of the final Hamiltonian (at $t = t_f$).

When solving global optimization problems with a machine, such as D-wave 2X, quantum annealing algorithms are implemented. We use a quantum annealing algorithm which is an inverse adiabatic quantum computational approach to store patterns on the ground state. Then we can compute the problem Hamiltonian at the ground state (see the Figure 15).

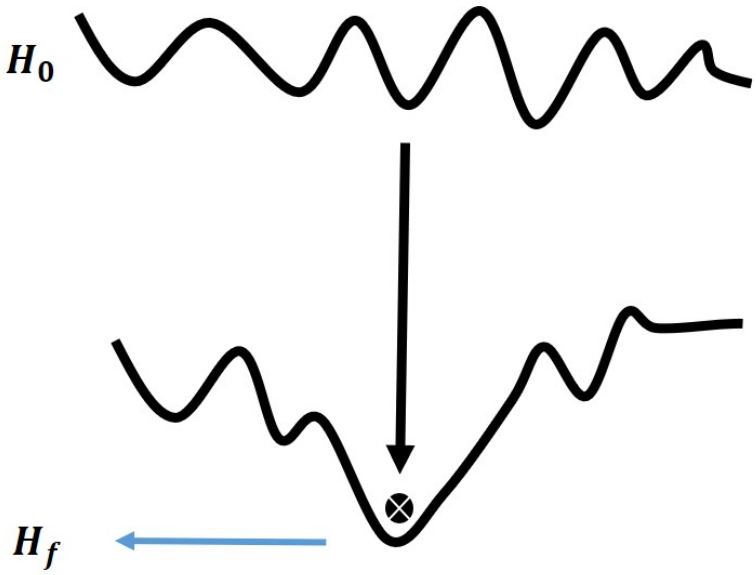

Figure 15: The initial state (flat state) is mapped to the desired state and Hamiltonian is determined at the ground state.

5.3 Training A Spin System with the Imaginary Time Code

In order to obtain the global minimum, we have to derive the learning rule as we discussed in chapter IV. Therefore we can start it from the formal solution we have obtained in for the Schrodinger Equation (see the equation 24).

The solution for the equation (24) can be written as for constant H

$$\rho(t) = exp(iHt/\hbar)\rho(0)exp(-iHt/\hbar) \tag{42}$$

Analytically, above solution can be changed by replacing imaginary time ($t \rightarrow i\beta\hbar$), so that this result becomes as the density matrix of a specific pattern (or letter) as a function of inverse temperature.

$$\rho(\beta) = exp(-\beta H)\rho(0)exp(\beta H) \tag{43}$$

where $\beta = \frac{1}{kT}$ is the inverse temperature in units of Boltzmann's constant. Now, using the interaction picture, the Hamiltonian can be splited in to two parts, one in real

time and one in imagianry time. The Schrodinger equation can be integrated numerically in real time to find the time evolution of $\rho_s(t)$ (s represents the Schrodinger picture). And the temperature dependence can be found with (I represents the interaction picture),

$$\rho_I(t,\beta) = exp(-\beta H)\rho_S(t)exp(\beta H). \tag{44}$$

Using above relationships, we want to develop a weight update rule based on "gradient descent" method by constructing the Lagrangian as in equation (45). Therefore the Lagrangan to be minimized,

$$L = \frac{1}{2}\left|\rho_{Ides} - \rho_I(t)\right|^2 + \int_0^{t_f} \lambda^\dagger(t)exp(-\beta H)\left(\frac{\partial \rho_s}{\partial t} - \frac{i}{\hbar}[\rho_s, H]\right)exp(\beta H)\gamma(t). \tag{45}$$

where ρ_{Ides} is the density matrix of the final state of the letter (pattern). As we explained in the chapter IV,we change the parameters (K, ε, ζ), with time and also the inverse temperature $\beta(t)$. We take the first variation of the Lagrangian L with respect to ρ, and take the integral by parts of the $\delta L_{\delta\rho} = 0$, in order to obtain the learning rule which has the boundary condition $-[\rho_{Ides} - \rho_I(t_f)]_{ji} + \lambda_i(t_f)\gamma_j(t_f)$ at final time. Therefore vector elements of the Lagrange multipliers can be calculated and we can have the learning rule as below.

$$\gamma_i\frac{\partial \gamma_j}{\partial t} + \frac{\partial \lambda_i}{\partial t}\gamma_j - \frac{i}{\hbar}\sum_k \lambda_k H_{ki}\gamma_j + \frac{i}{\hbar}\sum_k \lambda_i H_{jk}\gamma_k \tag{46}$$

Therefore we can write the gradient descent rule for each parameter (here η is the learning rate)

$$w_{New} = w_{Old} - \eta\frac{\partial L}{\partial w} \tag{47}$$

$$\frac{\partial L}{\partial w} = \frac{\partial}{\partial w}\left\{\int_0^{t_f} \lambda^\dagger(t)\left\{\frac{\partial \rho_s}{\partial t} - \frac{i}{\hbar}[\rho_s, H]\right\}\gamma(t)\right\}dt \tag{48}$$

The Hamiltonian, density matrix and derivative of the Lagrangian with respect to weights are real numbers because of the Hermiticity. So that the derivative of the Lagrangian can be written as,

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w}\bigg|_{\beta=0} + \int_0^{t_f} \beta(t)\lambda^\dagger(t)\left\{\frac{\partial}{\partial t}\left[\rho_I, \frac{\partial H}{\partial w}\right] - \frac{i}{\hbar}\left[\left[\rho_I, \frac{\partial H}{\partial w}\right], H\right]\right\}\gamma(t)dt + \frac{\beta_f}{t_f}\int \lambda_\dagger(t)\left[\rho_I, \frac{\partial H}{\partial w}\right]\gamma(t)dt \quad (49)$$

Note that the first correction term is of the same form as the original, but with the commutator playing the role of the density matrix.

5.3.1 Analysis and Results

Generally, the large SQUID arrays are initialized to a coherent equal superposition of all basis state (flat state). In charge basis, density matrix of flat state for N-qubit system is $\rho_{flat} = \frac{1}{2^N}\Pi_{i=1}^{N}[|0\rangle + |1\rangle]_i \otimes [\langle 0| + \langle 1|]_i$. In the optimization process, the annealing machine maps the flat state to the desired state (pattern) state by integrating the Schrodinger equation. In a four qubit system, we trained the spin system for Letter N, L, Z and X in the ground state with approximately 1.5% RMS error (see the figure 16). With the extension of this algorithm up to five and six qubit systems, 14 patterns had been successfully stored on the ground state (see the table 1). After this calculations, noise and coherence are added to the density matrix in order to check the robustness of the QNN output for the damged data.

Table 1: QNN output to encoded states of different letters in 4,5 and 6 qubit systems.

Number of Qubits	Pattern	Epoch	RMS
4	N	50	0.0171
4	Z	50	0.0107
4	X	50	0.0163
4	O	50	0.0153
4	L	50	0.0140
5	Pentagram	60	0.0240
5	K	60	0.0283
6	F	60	0.0279
6	H	60	0.0149
6	I	60	0.0149
6	S	60	0.0150
6	R	60	0.0149
6	Z	60	0.0158
6	J	60	0.0158

5.4 Robustness to Noise and Decoherence

Quantum systems are not isolated and can be affected by noise. That means each of the elements of the density matrix that we have defined for the each letter state contain some uncertainties. Since we are working with the simulations, we can isolate the different effects of noise and deoherence. Therefore, a perturbation (δA) is added to the density matrix (see the equation 50). Noise perturbation is referring to a random (uncorrelated) magnitude of a given size, and the phase perturbation is referring to a random phase. In general, both effects will be present.

$$\rho_I(t,\beta) = exp(-\beta H)(\rho_S(t) + \delta A)exp(\beta H). \tag{50}$$

After we add the noise to the density matrix of the letter state, the system is allowed to evolve and then the perturbed density matrix at the final time ($t = t_f$) is obtained in order to calculate the correlations of each pair. In this work, we are considering only a four qubit system. The correlation of each pair (for example AB), can be determined from $tr[(\rho(t_f)\sigma_{zA} \otimes \sigma_{zB}]^2$ and each pair will be illustrated on a 2-D plane. We trained 5 Letters

(patterns) with several noise levels. For very large noise levels , there are less possibility to recognize the character (see Figure 18). According to these results, quantum neural network recalls the stored damaged letters successfully from 0 to 0.01 noise levels. However, at large noise levels ($\gg 0.1$), character recognition is greatly reduced. Experimentally, we were able to show that the RMS error is bounded at large noise powers (see Figure 16).

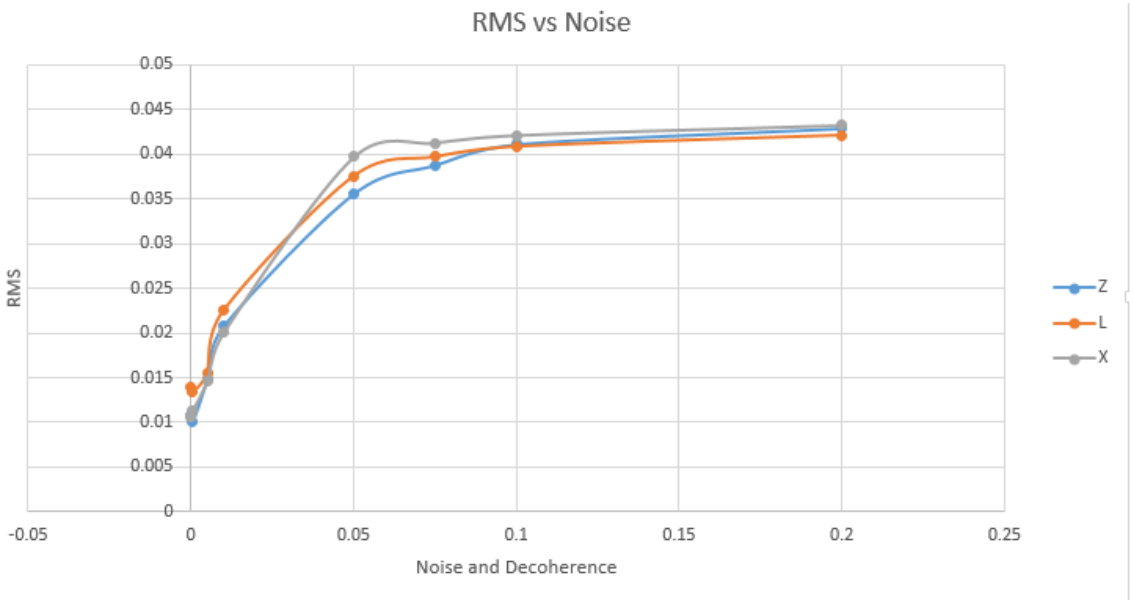

Figure 16: For the large noise levels, the RMS error reaches it's maximum value. This shows that, QNN output is robust to noise and decoherence.

.

5.4.1 Recalling Patterns Using Qubit-Qubit Correlations

To demonstrate the patterns which have been encoded by pairwise entanglement, we can consider the correlations between each pair of qubits in a N qubit system. We represent the correlation by a line segment drawing between the i-th qubit and the j-th qubit $(e_{ij}; i \neq j)$ such that the width of the line segment (w) proportional to e_{ij}.

$$w \propto e_{ij} = tr[(\rho(t_f)\sigma_{zi} \otimes \sigma_{zj}]^2 \tag{51}$$

For example, if we have the following combination according to equation 52,

$$e_{AB}, e_{BD}, e_{CD} \gg e_{AD}, e_{BC}, e_{AC} \tag{52}$$

The letter "N" can be implemented as shown in Figure 7. The correlations between AB, BC and CD should be significantly greater than the correlations between AD, BD and AC. According to Figure 18, letter X shows more robustness than others becuse it is very easy to train BD and AD. But, training AB and AD with a low a correlation between B and D qubits is harder than training letter X.

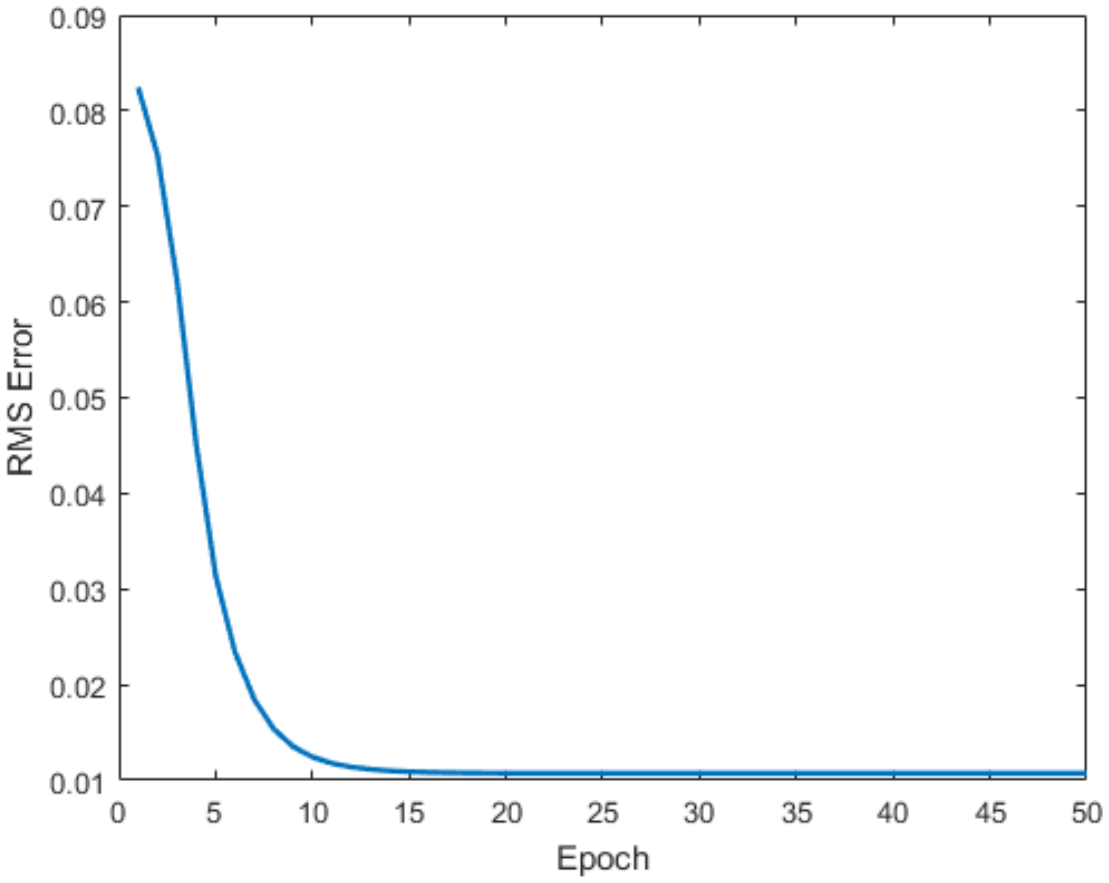

Figure 17: Training the annealing of the four-qubit system from flat state to letter N state. This graph shows the root mean squared error for training as a function of epoch. The asymptotic error was 0.0283 and the learning rate was $1.2 \times 10^{-6} J$.

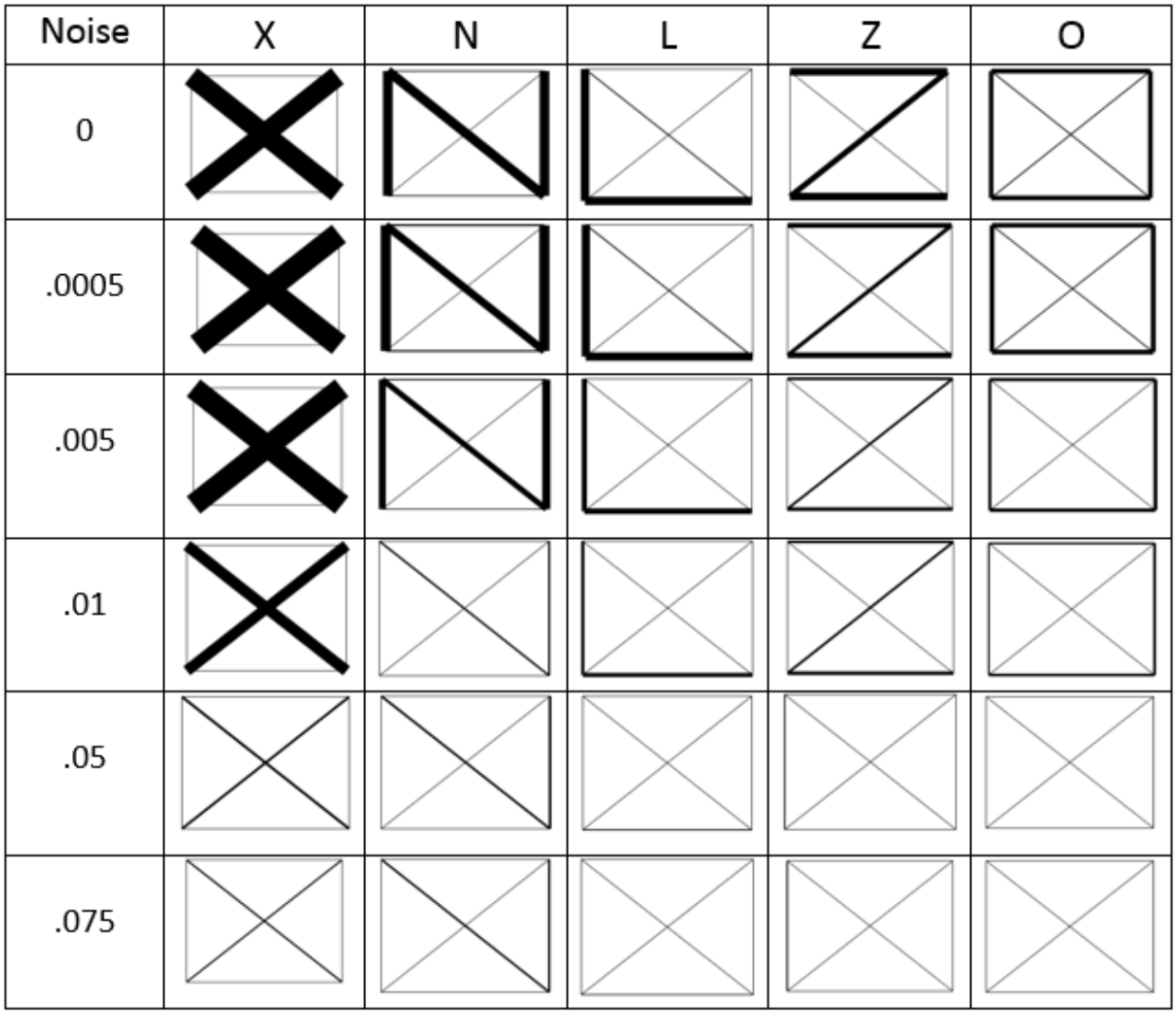

Figure 18: The illustrations of patterns (Letters) X, N, L Z and O in different noise levels. When the noise level is 0.05 and 0.075 letters L, Z and O are cannot be distinguished.

CHAPTER 6

CONCLUSIONS

We have presented a new approach for pattern storage via pairwise entanglement in a multiqubit system. There is a limit of the number of characters that can be stored with a 4 qubit system but, theoretically, the quantum neural network can be generalized to large qubit systems. Hence, entanglement can be used to read out as many different characters and shapes with real time evolution. Using real time evolution of a quantum system, we successfully trained four letters (X,Z,O,N and L) on the ground state of a four qubit system with approximately 2% RMS error.

We were also able to show that patterns can be stored in a SQUID array by annealing and determined the problem Hamiltonian to produce the desired state at the ground state. It turns out that, training an artificial spin system by using inverse quantum annealing is much easier that the real-time evolution. In this work, we trained 14 characters in 4,5 and 6 qubit systems, with around 1.5% RMS error. Those 14 patterns were stored on the ground state via mapping by starting from the flat state at zero noise level.

It is shown that, the quantum neural network calculations are robust to both noise and decoherence when storing damaged patterns up to 6 qubit system. Quantum processors such as D-wave 2X, contains 1000 qubits. According to this work, training images and the system is more robust to noise and decoherence as the size of the system increases. As the number of qubits increases the number of epochs required for training decreases, even though the total time of simulation goes up. Therefore, in a proper hardware, these simulations can be extended successfully in pattern storage at the lowest energy level even with noise and decoherence.

REFERENCES

REFERENCES

[1] Behrman EC, Chandrashekar V, Wang Z, Belur CK, Steck JE, Skinner SR. "A quantum neural network computes entanglement", (2002) arXiv:0202131.

[2] Behrman EC, Steck JE, Kumar P, Walsh KA. "Quantum algorithm design using dynamic learning", *Quantum Inf. Computatin*, vol. 8, pp. 12-29; 2008.

[3] Behrman EC, Bonde REF, Steck JE, Behrman JF, "On the correction of anomalous phase oscillation in entanglement witnesses using quantum neural networks". *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, pp 1696-1703.2013.

[4] Behrman EC, Steck JE, Moustafa MA. "Learning quantum annealing", *Quant. Inf. Comput.* vol. 17, pp. 0469-0487, 2017.

[5] Aizenberg I. *Complex-Valued Neural Networks with Multi-Valued Neurons*, Springer. 2011.

[6] Yann le Cun, Touretzky D, Hinton G, Sejnowski T, Morgan K, *A theoretical framework for back-propagation* in *Proc. 1998 Connectionist Models Summer School* , pp. 21-28, 1988.

[7] Paul W, Van NR. *Handbook of Intelligent Control*, pp. 79-80 and 339-344, 1992.

[8] Behrman EC, Steck JE. "Multiqubit entanglement of a general input state". *Quantum Inf. Comput.* vol. 13, pp. 36-53, 2013.

[9] Shapiro M, Brumer P. *Quantum Control of Molecular Processes.* Singapore: Wiley, (2012).

[10] Efron B, Tibshirani RJ. "An Introduction to the bootstrap". *Boca Raton, FL: Chapman and Hall/CRC.* 1994.

[11] Widrow B. "The rubber-mask" technique-II. pattern storage and recognition. Pattern recognition. 1973 Sep 1;5(3):199-211.

[12] Bibitchkov D, Herrmann JM, Geisel T. "Pattern storage and processing in attractor networks with short-time synaptic dynamics". *Network: Computation in neural systems.* 1;13(1):115-29, Jan 2002.

[13] Cohen MA, Grossberg S. "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks". *IEEE transactions on systems, man, and cybernetics.* 815-26. Sep 15, 1983.

[14] Singh M, Rajput B. "New Maximally Entangled States for Pattern-Association Through Evolutionary Processes in a Two-Qubit System". *International Journal of Theoretical Physics.* 2017 Apr : 56(4):1274-85.

[15] Rigui Z. "Quantum Competitive Neural Network". *International Journal of Theoretical Physics*. 2010 Jan:49(1):110-9.

[16] Singh M, Rajput B. "Processes of Quantum Associative Memory (QuAM) Through New Maximally Entangled States (Singh-Rajput MES)". *International Journal of Theoretical Physics*. 2016 Jul;55(7):3207- 19.

[17] Baldassi C, Zecchina R. "Efficiency of quantum vs. classical annealing in nonconvex learning problems". *Proceedings of the National Academy of Sciences of the United States of America*. 2018 Feb 13;115(7):1457- 62.

[18] Quantum machine learning-using quantum computation in artificial intelligence and deep neural networks: Quantum computation and machine learning in artificial intelligence. 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), Industrial Automation and Electromechanical Engineering Conference (IEMECON), 2017 8th Annual. 2017;268.

[19] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. "Quantum machine learning". *Nature*. 549(7671):195, Sep 2017.

[20] Online optimization of security-sensitive real-time storage applications for NAND flash memory storage systems. 2013 IEEE 19th International Conference on Embedded and Real-Time Computing Systems and Applications, Embedded and Real-Time Computing Systems and Applications (RTCSA), 2013 IEEE 19th International Conference on. 2013;121.

[21] Neigovzen R, Neves JL, Sollacher R, Glaser SJ. "Quantum pattern recognition with liquid-state nuclear magnetic resonance". *Physical Review A*. 16;79(4):042321. Apr 2009.

[22] Schuld M, Sinayskiy I, Petruccione F. Quantum computing for pattern classification. InPacific Rim International Conference on Artificial Intelligence 2014 Dec 1 (pp. 208-220). Springer, Cham.

[23] Schutzhold R. "Pattern recognition on a quantum computer". *Physical Review*. 26;67(6):062311. Jun 2003.

[24] Trugenberger CA. Quantum pattern recognition. *Quantum Information Processing*. 2002 Dec 1;1(6):471-93.

[25] Diamantini M, Trugenberger Carlo. "High-Capacity Quantum Associative Memories". *Journal of Applied Mathematics and Physics*. 2016 Jan(4)

[26] Dur W, Vidal G, Cirac J I. "Three qubits can be entangled in two inequivalent ways". (Feb 2008), arXiv: 0005115.

[27] Scott H, Wootters WK. "Entanglement of a pair of qubit bits" (Mar 1997), arXiv: 9703041.

[28] Sergio B, Vadim NS, Alireza S, Sergei VI, Mark D, Vasil SD, Mohammad A, Anatoly S, Masoud M, Hartmut N. "Computational Role of Multiqubit Tunneling in a Quantum Annealer" (Feb 2015), arXiv:1502.05754.

[29] Behrman EC, Nguyen NH, Steck JE, McCann M. "Quantum neural computation of entanglement is robust to noise and decoherence" (Nov 2015), arXiv:1510.09173.

[30] Wim VD, Michele M, Umesh V. "How Powerful is Adiabatic Quantum Computation?" (Jun 2002), arXiv:0206003.

[31] Christoper MB. *Neural Networks for Pattern Recognition*. Clarendon press Oxford, 1995.

[32] Kumar S. *Neural Networks*. Tata mcgraw hill education private limited, 2004.

[33] Breuckmann NP, Ni X. Scalable neural network decoders for higher dimensional quantum codes (Jun 2017), arXiv:1710.09489.

[34] Wotters WK. "Entanglement of formation of an arbitrary state of two qubits". *Phys. Rev. Lett 80,* 2245.1997.

[35] Nielsen MA, Isaac CL. *Quantum computation and quantum information*, Cambridge university press, 2010.

[36] Zhih AJ, Biao Y, Rui Z, Yu CW, Guang CG, Guo PG. "Quantum Neural Network States: A Brief Review of Methods and Applications", (Feb 2019), arXiv:1808.10601.

[37] Shankar R. *Principles of quantum mechanics*. Plenum publications, 1994.

[38] Tuckerman ME, *Statistical Mechanics: Theory and Molecular Simulation*. Oxford university press, 2010.

[39] Lewenstein M. "Quantum Perceptron", *Journal of modern optics*, vol. 41, pp. 2491-2501, 1994.

[40] Dan V, Tony M. "Quantum Associative Memory", *IEEE Transactions on Neural Networks*, June 16 1998.

[41] Siddhartha B, Ujjwal M, Paramartha D. *Quantum Inspired Computational Intelligence: Research and Applications*, Todd green, 2017.

[42] Wootters WK. "Entanglement of Formation of an Arbitrary State of Two Qubits" (Sep 1997), arXiv:9709029.

[43] Martin B, Plenio, Shashank V. "An introduction to entanglement measures". (Jun 2006), arXiv:0504163.

[44] Bell JS. "On the Einstine Podolsky Rosen paradox". *Physics publishing co.* 1994. Vol. 1, pp. 195-200.

[45] Einstein A, Podolsky B, Rosen N. "Can Quantum-Mechanical Description of Physical Reality be Considered Complete?". *Phys. Rev.* vol. 47, p 777. 1935.

[46] Barnett SM. *Introduction to quntum information*, Oxford university press, 2009.

[47] Sergio B, Vadim NS, Alireza S, Sergei VI, Mark D, Vasil SD, Mohammad A, Anatoly S, Masoud M, Hartmut N. "Computational Role of Multiqubit Tunneling in a Quantum Annealer", (Feb 2015), arXiv:1502.05754.

[48] Johnson MW. Amin MHS, Gildert S, Lanting T, Hamze F, Dickson N, Harris R, Berkley AJ, Johansson J, Bunyk P, Chapple EM, Enderum C, Hilton JP, Karimi K, Ladizinsky E, Ladizinsky N, Oh T, Perminov I, Rich C, Thom MC, Tolkacheva E, Truncik CJS, Uchaikin S, Wang J, Wilson B, Rose G. "Quantum annealing with manufactured spins", *Nature*. vol. 473, p 194, 2011.

[49] Johnson MW, Bunyk P,Maibaum F,Tolkacheva E, Berkley AJ, Chapple EM, Harris R, Johansson J, Lanting T, Perminov I, Ladizinsky E, Oh T, Rose G. "A scalable control system for a superconducting adiabatic quantum optimization processor", *Supercond. Sci. Technol.* 23, 065004 (2010).

APPENDICES

APPENDIX A

PAULI MATRICES

The Pauli matrices (Pauli spin matrices) are set of 2×2 complex matrices which are Hermitian and Unitary. Those are,

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

APPENDIX B

TENSOR PRODUCT

Suppose V and W are vector spaces of dimensions m and n respectively. And also V and W are Hilbert spaces. Then $V \otimes W$ is an mn dimensional vector space. The elements of $V \otimes W$ are liner combinations of "tensor products" $|v\rangle \otimes |w\rangle$ of V and W. If $|i\rangle$ and $|j\rangle$ are orthonormal bases for the spaces V and W then $|i\rangle \otimes |j\rangle$ is basis for $V \otimes W$. Tensor product satisfies the following basic properties.

(1). For any arbitrary scalar z and elements $|v\rangle$ of V and $|w\rangle$ of W,

$$z(|v\rangle \otimes |w\rangle) = (z\,|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z\,|w\rangle). \tag{53}$$

(2). For arbitrary $|v_1\rangle$ and $|v_2\rangle$ in V and $|w\rangle$ in W,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle \tag{54}$$

(3). For arbitrary $|v\rangle$ in V and $|w_1\rangle$ and $|w_2\rangle$ in W,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle\,. \tag{55}$$