



Master's Degree Thesis

# Study on Correlation Properties of the Bose-Hubbard model based on Neural Networks

Department of Engineering Science  
The University of Electro-Communications

HUANG DONGLAN  
Student Number: 2133040

Supervisor: Prof. Hiroki Saito  
Co-Supervisor: Prof. Kishimoto  
Department of Engineering Science  
The University of Electro-Communications  
Tokyo  
January, 2023

# Abstract

The most challenge posed by the many-body problem in quantum physics arises from the difficulty in describing the nontrivial dependencies encoded in the exponential complexity of the many-body wave function. Systematic machine learning of the wave function can reduce this complexity to a tractable computational form, for some notable cases of physical interest. We developed a variational representation to obtain quantum many-body ground state states of Bose-Hubbard model and based on artificial neural networks with variable number of hidden neurons. We show that convolutional network works better than fully connected network, and a multi-layers convolutional network is more efficient than single layer convolutional network . Adam is a better optimization method and Relu is a better efficiency lost function than other normal lost function. Moreover, we also got some correlation properties of the one dimensional extended Bose-Hubbard model based on above network model and method. It showed that correlation becomes decrease by interaction range increases in Superfluid(SF) phase, correlation becomes exponential suppression in Mott insulator(MI) phase. Those results offering a new powerful method to solve the quantum many-body problem.

# Keywords

Many-Body problem, Bose Hubbard model, Neural Network, Correlation.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Problem Description . . . . .	2
<b>2</b>	<b>Model</b>	<b>5</b>
2.1	Bose-Hubbard Model . . . . .	5
2.2	1D Extended Bose-Hubbard Model . . . . .	6
2.3	Ground-state Phase Diagrams . . . . .	7
2.4	Correlation Functions . . . . .	8
<b>3</b>	<b>Method</b>	<b>10</b>
3.1	Exact-Diagonalization . . . . .	10
3.2	Neural Networks . . . . .	14
3.3	Model Architecture . . . . .	16
<b>4</b>	<b>Results</b>	<b>21</b>
4.1	Network Model Parameters . . . . .	21
4.2	Correlation Properties . . . . .	23
<b>5</b>	<b>Conclusion</b>	<b>31</b>
5.1	Theoretical and Practical Implications . . . . .	31
5.2	Future Work . . . . .	32
5.3	Acknowledgments . . . . .	32

# Chapter 1

## Introduction

### 1.1 Background

Recently, the potential role that machine learning might play in the scientific discovery process has received increasing attention. Neural networks are among the most versatile and successful tools in machine learning and have been applied to a wide variety of problems in physics. Many of the earlier applications have focused on solving specific problems that are intractable analytically and for which conventional numerical methods deliver only unsatisfactory results. Conversely, neural networks may also lead to new insights into how the human brain develops physical intuition from observations[?].

Seeking ways to establish artificial intelligence methodologies as general-purpose tools for scientific research. This is motivated from various directions: from an artificial intelligence perspective, having machines autonomously discover scientific concepts about the world is often seen as an important step towards artificial general intelligence[?]; from the perspective of science, machine learning might complement human scientific research to both speed up scientific discovery and make it less susceptible to human biases.

An important step in the scientific process is to convert experimental data to a more succinct representation that is amenable to a theoretical treatment. A central challenge in quantum many-body physics is developing efficient numerical tools for strongly correlated systems, whose Hilbert space dimensionality grows exponentially with the system size, so does the information required for characterizing a generic state of the system. However, for many physical systems of practical interest, the ground states may have a simplified structure, thus, it can be appropriately approximated using an exponentially smaller number of parameters than that required for characterizing generic states. Typical examples include low dimensional strongly correlated systems, whose ground states can be represented in terms of matrix product states by taking advantage of limited entanglement entropy in the ground states.

On the other hand, in the last decade, ultracold atomic gases loaded in optical lattices successfully established as an excellent setup to probe the equilibrium, as well as the out-of-equilibrium physics of strongly correlated quantum systems. The great advantages of these experimental setups are essentially related to two aspects. On one hand, they have an extremely high degree of flexibility: besides the ability to address different geometries, and to deal with Bosonic and with Fermionic species, they admit the possibility to manipulate the underlying Hamiltonian system parameters to a large extent. Moreover, the remarkably high degree of isolation from any environmental source of decoherence opened up entirely new scenarios in the observation of genuinely many-body quantum phenomena[?].

## 1.2 Problem Description

The Axioms of Quantum Mechanics[?] as developed in the 1930s describe the quantum state of a physical system as a ray in a Hilbert Space. This

state vector contains all relevant information about the physical properties of the system under study, with each coefficient of the state providing a probability distribution over the set of possible observation outcomes for any measurement of the system. While this physical model is generally accepted as the most accurate description of reality so far, when dealing with systems of many particles the methods of Quantum Mechanics become intractable; specifically, for a system of  $N$  particles each with  $d$  degrees of freedom, the Hilbert Space of the state vector is  $d^N$ -dimensional, exponentially large in the number of particles and thus far too large for computations involving systems on practical scales. Computational physicists have worked extensively on methods for approximating properties of many-body systems, using either stochastic approaches that compute observable quantities as expectation values over samples drawn from the target distribution or by finding compressed representations of the quantum state within the Hilbert Space that efficiently carry the state's information with sub-exponential scaling of parameters with system size.

Because the Many-Body Problem ultimately involves approximating a function of exponentially many parameters, Machine Learning provides ample tools for finding efficient models that approximate the coefficients of the state vector. In this paper we adopt the Neural Quantum State Ansatz of Carleo and Troyer[?] for analysis of the representational power of Neural Networks in describing a quantum state, where the network is trained on data sampled from the probability distribution of the system under study and iteratively optimized in a feedback scheme until convergence to the desired state. After describing the project architecture and the training procedure, we continue to demonstrate the effectiveness of this method in capturing particle-particle correlations for different physical environments.

The paradigm model to describe cold Bosonic atoms trapped in an optical lattice is obtained by combining the kinetic energy in the lowest band with the on-site repulsion arising for sufficiently deep lattices. This leads to the celebrated Bose-Hubbard model (BHM)[?]. The rich physics of the BHM stems from the competition between the kinetic energy  $J$ , which is gained by delocalizing particles across the lattice in an extended Bloch state, and the repulsive on-site interaction  $U$ , which disfavors having more than one particle per site. When the kinetic energy term dominates, the system is in a coherent Superfluid (SF) phase; on the other hand, repulsive interactions tend to favour a Mott insulating (MI) phase[?].

The recent advances in manipulating magnetic atoms and molecules with large dipole momentum make it possible to achieve longer-range interactions, which can be accurately tuned as well, thus permitting to probe the interplay between strong correlations and charge-ordering effects[?]. Dipolar bosons confined in optical lattices are typically described by an extended version of the BHM, the so called Extended Bose Hubbard Model (EBHM), which also includes a two-body non local repulsive term typically decaying as  $r^3$  with the distance  $r$ .

# Chapter 2

## Model

### 2.1 Bose-Hubbard Model

The basic physics of interacting bosons on a lattice is contained in the Bose-Hubbard model. The Hamiltonian describes the interactions between bosons that can occupy sites in a  $d$ -dimensional lattice. The model is characterized by a hopping energy  $J$ , an on-site interaction  $U$ .

$$\hat{H} = -J \sum_{\langle i,j \rangle} \hat{a}_i^\dagger \hat{a}_j + \frac{U}{2} \sum_i \hat{n}_i (\hat{n}_i - 1), \quad (2.1)$$

where  $\hat{a}_i^\dagger$  is the annihilation operator at site  $i$ , and  $\hat{n}_i = \hat{a}_i^\dagger \hat{a}_i$  is the number operator, where the Bose commutation relation  $[\hat{a}_i, \hat{a}_j^\dagger] = \delta_{ij}$  is satisfied. The notation  $\langle i, j \rangle$  indicates that the sum runs over pairs of neighbor sites in the lattice. The Bose-Hubbard model can be used to describe physical systems

such as bosonic atoms in an optical lattice(Fig.2.1.a). The tunneling rate for cold atoms in an optical lattice(Fig.2.1.b) can be made to depend on whether a neighboring site is occupied—a behavior that may reflect the tunneling in complex materials. Cold atoms in an optical lattice are often used to

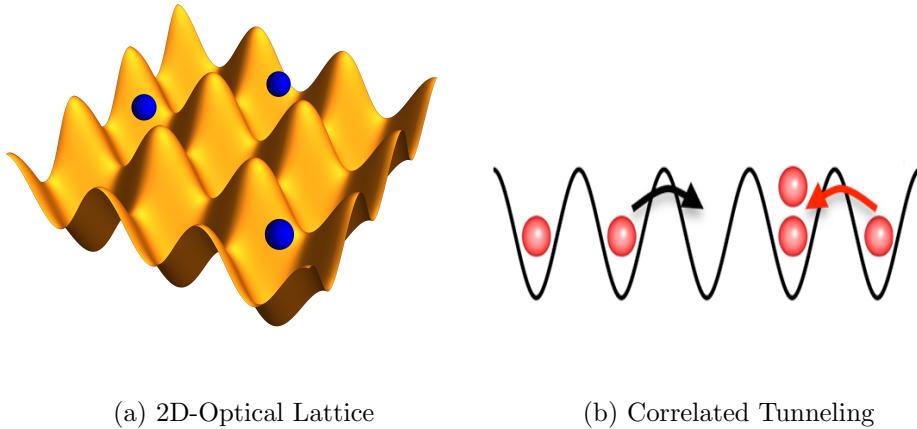


Figure 2.1: The description of Bose-Hubbard model. [?][?]

simulate other systems, with the advantage that researchers can create a potential-energy landscape of their choosing.

## 2.2 1D Extended Bose-Hubbard Model

In the following we will focus on the 1D extended Bose Hubbard model (EBHM) with on-site and nearest-neighbor repulsive interactions. The sharp cutoff of the  $r^{-3}$  interaction range, although presenting quantitative differences with the standard long range EHBM, does not qualitatively alter its physics:

$$\hat{H} = -J \sum_{\langle i,j \rangle} \hat{a}_i^\dagger \hat{a}_j + \frac{U}{2} \sum_i \hat{n}_i (\hat{n}_i - 1) + V \sum_i \hat{n}_i \hat{n}_{i+1}, \quad (2.2)$$

compare to BHM, EBHM be added one item  $V$  - the nearest-neighbor interaction strengths. And then, here  $\hat{a}_i^\dagger, \hat{a}_i$  are creation and annihilation operators of bosons on site  $i$ ,  $J$  denotes the hopping strength, while  $U$  is the on-site interaction strengths. Hereafter we set the energy scale by taking

$J = 1.$

### 2.3 Ground-state Phase Diagrams

We will concentrate on the model in Eq.(2.2) at zero temperature, which is known to exhibit a quite rich phase diagram with various quantum phases, ranging from the superfluid (SF) for low  $U$ ,  $V$  interaction strengths, to insulating phases in the opposite regime. Depending on the relative strength of  $U$  and  $V$ , a Mott insulator (MI) or a density-wave state (DW) can form, the first one establishing with large values of  $U$ , while the second one with large  $V$ . In between the two conventional insulating phases, a peculiar gapped phase emerges: the Haldane insulator (HI).

The phase diagram and the phase transitions of such kind of model, describing bosons loaded in optical lattices with short range interactions, have been addressed since long time[?]. Emphasis was put on the experimentally accessible SF-MI transition[?]. In the 1D case and with a constant integer filling, the transition is of the BerezinskiiKosterlitz-Thouless type and was extensively analyzed with Density Matrix Renormalization Group(DMRG) in Refs.[?][?]. In this thesis, we just focus on discussing SF and MI phases, SF-MI transition. Before that, we need to work out the Ground-state phase diagram of the pure one-dimensional (1D) EBHM, by means of the numerical DMRG technique. Refs.[?] has been already provided very accurate results for these systems(Fig 2.3). In the following context we based on this complete phase diagram to discuss the BHM correlation properties.

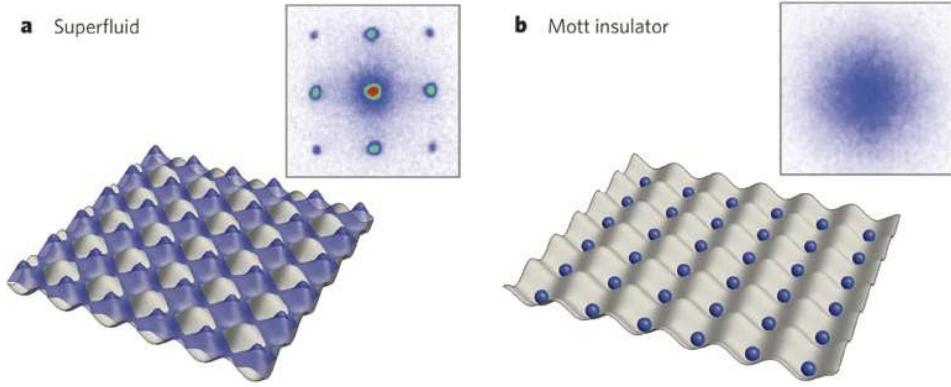


Figure 2.2: An illustration that helps visualize the behavior of the Superfluid and Mott Insulating phases. In (a) we see that in the superfluid phase the particles are delocalized across the lattice and that multiple particles can occupy a single site. In (b), which shows the Mott Insulating phase, we have a single particle in a single site. This phase has strong localization and is incompressible. The diagram is taken from. [?]

## 2.4 Correlation Functions

The nature of the different quantum phases emerging in the EBHM, as well as all the Quantum Phase Transition (QPT) points determining the various phase boundaries, are revealed by the behavior of certain two-point correlation functions. The following correlators are able to univocally distinguish between all the emerging quantum phases:

$$C_{SF}(r) = \langle \hat{a}_j^\dagger \hat{a}_{j+r} \rangle, \quad (2.3)$$

in this thesis, we just focus on discussing SF and SF-MI transition. The long range off-diagonal order, typical of superfluid states, manifests itself into a

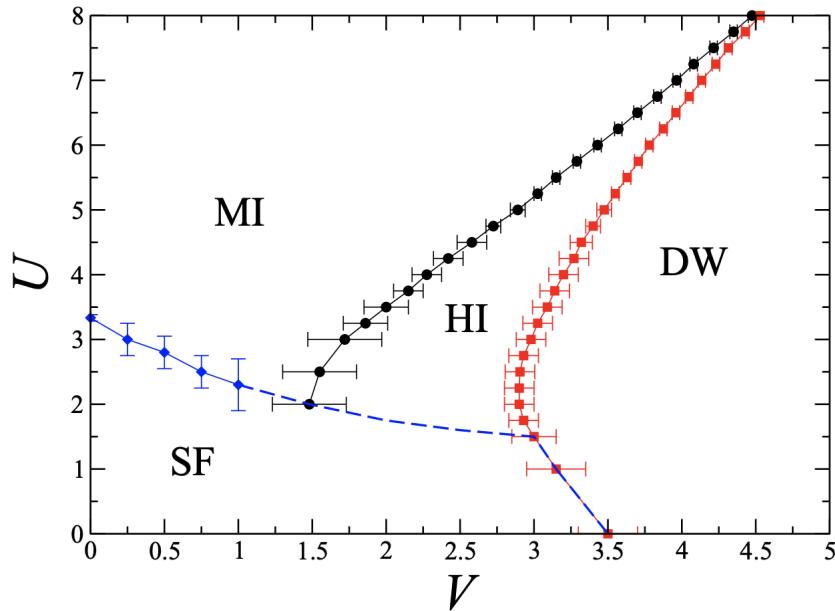


Figure 2.3: Ground-state phase diagram of the 1D EBHM in the  $(U, V)$  plane, with a maximum admissible number of 3 bosons per site. The boundaries between the different phases have been calculated by analyzing the string order parameter (MI-HI — black circles), the density wave order parameter (HI-DW — red squares) and the charge energy gaps of the system (SF-MI — blue diamonds). The dashed blue line is an interpolation of the continuous blue curve and the red points at large  $V$  values. [?]

power-law decay to zero of  $C_{SF}(r)$ ; on the other hand, in the insulating phases the absence of such ordering is characterized by an exponential suppression of  $C_{SF}(r)$  with distance  $r$ .

# Chapter 3

## Method

### 3.1 Exact-Diagonalization

Before we proceed further to discuss specific examples, let us describe the details of the exact-diagonalization methods. That's the basic method be used in physics to determine the eigenstates and energy eigenvalues of a quantum Hamiltonian. In this technique, a Hamiltonian for a discrete, finite system is expressed in matrix form and diagonalized using a computer. Exact diagonalization is only feasible for systems with a few tens of particles, due to the exponential growth of the Hilbert space dimension with the size of the quantum system. It is frequently employed to study lattice models, including the Hubbard model, Ising model, etc [?].

We will study the problem using a technique called exact diagonalization. This method takes the extended Bose-Hubbard Hamiltonian and represents it as a matrix. This extremely sparse matrix is then diagonalized using the ARPACK package available for FORTRAN. This package uses an iterative process called the Lanczos algorithm in order to find extremal eigenvalues. We will use these tools to find several of the lowest lying eigenvalues and

eigenvectors which will in turn be used to calculate some useful observables. The results of this method give precise (i.e. machine precision) answers for the eigenvalues and eigenvectors in a reasonable amount of time, unfortunately it is still extremely limited for large systems. The dimensionality of the Hilbert space is given by

$$D = \frac{(N + M - 1)!}{N!(M - 1)!}, \quad (3.1)$$

where  $N$  is the number of bosons and  $M$  is the number of sites. For our 8 site ring with 8 bosons the Hilbert space is  $D = 64,523$ . To get a good sense of how quickly this figure grows we look at a few more examples. For fixed filling  $N/M = 1$ , we get  $D = 6,435$  when  $M = 8$ ,  $D = 92,378$  when  $M = 10$ , and finally  $D = 1,352,078$  when  $M = 12$ . Now we take a more in depth analysis into the workings of exact diagonalization.

As was mentioned earlier we only need the first few lowest eigenvalues and eigen-vectors in order to fully understand the behavior of the system. Therefore we use the Lanczos methods to significantly reduce the time needed for performing the calculations. In order to solve the full matrix (i.e. get all eigenvalues and eigenvectors), the computational time required would be on the order of  $O(D^3)$  while the space needed to store the entire matrix would take up  $O(D^2)$  space. By using the Lanczos method we are able to get the same machine precision convergence much faster. Without going into any details, this method allows us to find extremal eigenvalues and their corresponding eigenvectors using an iterative process that reduces the computation time by a significant amount.

The solutions of this algorithm give eigenvalues  $E^{(v)}$  and eigenvectors  $C^{(v)}$  where each  $v$  corresponds to a specific energy level of the system. The

ground state can be written as a superposition of all the Fock states with different weights given by elements of  $C^{(r)}$ . In the number basis we can write this as

$$|\Psi_v\rangle = \sum_{\alpha=1}^D C_{\alpha}^v |\{n_1, \dots, n_M\}_{\alpha}\rangle. \quad (3.2)$$

Now there are a few tricks that can be utilized in order to make the program more efficient. First of all we need to identify all of our basis vectors. Since we are working in the occupation number representation  $\{|n_1, n_2, \dots, n_M\rangle\}$  with a fixed number of particles  $\sum_{i=1}^M n_i = N$ , we need to have an effective way of calculating every Fock state that makes up the basis. In the occupation number basis we of course have

$$\hat{n}_i |n_1, n_2, \dots, n_M\rangle = n_i |n_1, n_2, \dots, n_M\rangle, \quad (3.3)$$

where  $n_i \geq 0$ . We use a combinatorial algorithm that efficiently enumerates all the possible states.

Another trick that is used in order to reduce computation time is a technique called tagging. The process of creating tags for all the basis vectors of the Hamiltonian matrix will greatly reduce the space and time needed for evaluating our lowest energies. Let's take the Hamiltonian matrix

$$H_{uv} = \langle u | H | v \rangle, \quad (3.4)$$

where  $u$  and  $v$  are all the basis vectors. Typically the entire basis vector would

be stored in a matrix,  $A = D \times M$ , where the  $v^{th}$  row would correspond to the  $v^{th}$  vector. So we could define each vector as,

$$|v\rangle = |A_{v1}, A_{v2}, \dots, A_{vM}\rangle. \quad (3.5)$$

From this matrix let us pick out an arbitrary basis vector  $|v\rangle$  which is just one of the  $D$  possible Fock states and change it into a unique number. This unique number will be referred to as the tag of the vector and will be defined as,

$$T(v) = (A_{v1}, A_{v2}, \dots, A_{vM}). \quad (3.6)$$

This is accomplished by letting  $T(v) = \sum_{i=1}^M \sqrt{P_i A_{vi}}$  where  $p_i = 100 * i + 3$ . This process guarantee that none of the tags will repeat.

Now in order to be able to find any arbitrary vector in our new list of tags it would take an average of  $D/2$  tries. To make this more efficient let's sort them in an ascending or descending order. Once the tags are all sorted we we can use Newton's binary method in order to locate the proper tag that we are looking for. This now reduces the search time from  $D/2$  to  $\log_2 D$ . Just to give an idea of how much better this method is let's compare how many tries it would take to find the right tag for our small system of 8 sites with 8 bosons. We will recall that the Hilbert space has the dimensions of  $D = 65,482$  so using the standard method of just scanning down the unsorted list will require an average of 32,471 tries while the binary method takes only about 16 tries.

Still the most effective way that we can reduce the Hilbert space and

make the computations even faster is by considering the symmetries found in the Bose- Hubbard model.

### 3.2 Neural Networks

Before we proceed further to discuss specific examples, let us describe the details of the Artificial Neural Networks(ANNs) and the optimization methods we will use. We consider a fully connected feed-forward neural network consisting of interconnected group of nodes (neurons) with a stacked layered structure, and it's expressibility is encoded in sets of adaptive weights of connections between neurons in adjacent layers.

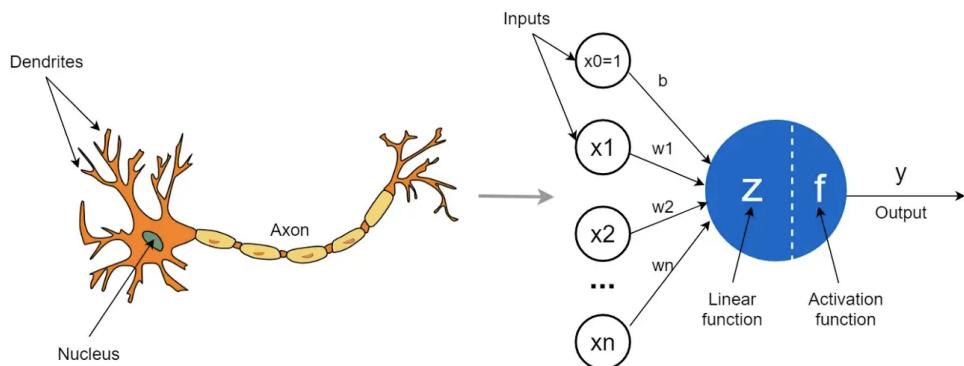


Figure 3.1: The Concept of Artificial Neurons. A perceptron takes the inputs,  $x_1, x_2, \dots, x_n$ , multiplies them by weights,  $w_1, w_2, \dots, w_n$  and adds the bias term,  $b$ , then computes the linear function,  $Z$  on which an activation function,  $f$  is applied to get the output,  $y$ . [?]

Artificial neurons (also called Perceptrons, Units or Nodes) are the simplest elements or building blocks in a neural network. They are inspired by biological neurons that are found in the human brain. It is worth discussing how artificial neurons (perceptrons) are inspired by biological neurons. You can consider an artificial neuron as a mathematical model inspired by a

biological neuron.

- A biological neuron receives its input signals from other neurons through dendrites (small fibers). Likewise, a perceptron receives its data from other perceptrons through input neurons that take numbers.
- The connection points between dendrites and biological neurons are called synapses. Likewise, the connections between inputs and perceptrons are called weights. They measure the importance level of each input.
- In a biological neuron, the nucleus produces an output signal based on the signals provided by dendrites. Likewise, the nucleus (colored in blue) in a perceptron performs some calculations based on the input values and produces an output.
- In a biological neuron, the output signal is carried away by the axon. Likewise, the axon in a perceptron is the output value which will be the input for the next perceptrons.

Fig.3.1 shows a detailed structure of a perceptron. This perceptron more like a mathematical model in which linear function and non-linear activation function work together to calculate an output that can be an input for the next perceptrons or just the final output. Activation function is the most important items among of these units. The type of activation function determines how a neuron should fire or activate. In fact, neural networks without activation functions are just linear regression models that can only model linear relationships in the data. The parameters of a perceptron are weights and bias. The weights control the level of importance of each input. It determines how easily a neuron fires or activates as described above. There

may be thousands of parameters (weights and biases) in a neural network as the number of perceptrons increases. As mentioned earlier, the optimal values for those parameters are found during the learning (training) process of the neural network. Therefore, parameters (weights and biases) learn their optimal values from the data we provided. Artificial Neural Networks(ANNs) combined by many artificial neurons.

### 3.3 Model Architecture

The quantum state is expanded by the Fock states  $|\Psi\rangle = \sum \psi(n_1, n_2, \dots) |n_1, n_2, \dots\rangle = \sum \psi(\mathbf{n}) |\mathbf{n}\rangle$ , where  $n_i$  is the number of particles at the  $i$ th site. When a set of integers  $\mathbf{n}$  is input to the network, the value of the wave function  $\psi(\mathbf{n})$  is obtained from the output layer. We attempt to optimize the parameters of the network so that the output  $\psi(\mathbf{n})$  is close to the ground-state wave function. According Eq.(2.1) and Fig.3.1, The feedforward network operates as follows. The integers  $\mathbf{n}$  are set to the input units as  $u_j^{(0)} = n_j$ , where the number of units in the input layer is  $M$ . The values of the hidden units are calculated as

$$U_k^{(1)}(\mathbf{n}) = \sum_{j=1}^M W_{kj}^{(1)} n_j + h_k^{(1)}. \quad (3.7)$$

We adopt the rectified linear(Relu) as an activation function(The reason please refer to Results.4.1), and the output units become

$$U_m^{(2)}(\mathbf{n}) = \sum_{k=1}^{N_H} W_{mk}^{(2)} \text{relu} u_k^{(1)} \mathbf{n} + h_m^{(2)}. \quad (3.8)$$

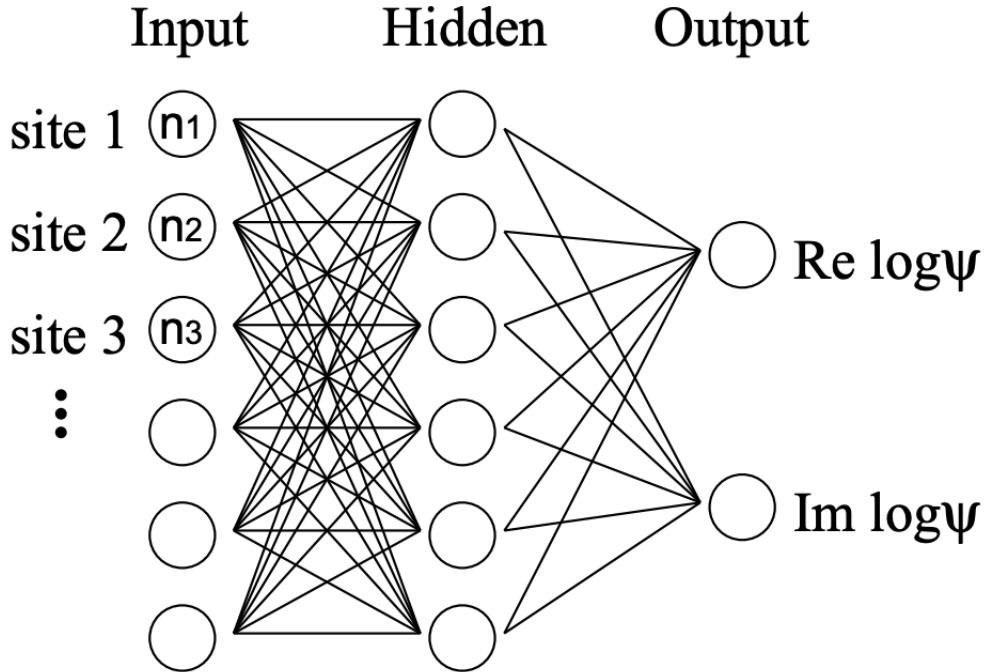


Figure 3.2: Schematic diagram of the ANNs used to solve the Bose-Hubbard model. The number of particles at each site is assigned to the input layer, and the corresponding value of the wave function is obtained from the output layer. The units in the adjacent layers are fully connected. [?]

where  $N_H$  is the number of units in the hidden layer, and  $m = 1, 2$ . The weights  $W_{kj}^{(1)}$  and  $W_{mk}^{(2)}$  and the biases  $h_k^{(1)}$  and  $h_m^{(2)}$  are real. The wave function is thus given by

$$\psi(\mathbf{n}) = \exp[u_1^{(2)}(\mathbf{n}) + iu_1^{(2)}(\mathbf{n})]. \quad (3.9)$$

An expectation value of a quantity  $\hat{A}$ ,

$$\langle \hat{A} \rangle = \frac{\sum_{n,n'} \psi^*(\mathbf{n}) \langle \mathbf{n} | \hat{A} | \mathbf{n}' \rangle \psi(\mathbf{n}')}{\sum_n |\psi(\mathbf{n})|^2}, \quad (3.10)$$

is calculated by the Monte Carlo method with Metropolis sampling.

Given  $\mathbf{n}_1$  and  $\mathbf{n}_2$ , the probability that  $\mathbf{n}_1 \rightarrow \mathbf{n}_2$  is adopted,  $|\psi(\mathbf{n}_2)/\psi(\mathbf{n}_1)|^2$ , can be calculated from the network, and we can then sample  $\mathbf{n}$  with probability  $|\psi(\mathbf{n})|^2 / \sum_{n'} |\psi(\mathbf{n}')|^2$ . The expectation value in Eq.(3.10) is therefore stochastically calculated as

$$\left\langle \sum_{n'} \langle \mathbf{n} | \hat{A} | \mathbf{n}' \rangle \frac{\psi(\mathbf{n}')}{\psi(\mathbf{n})} \right\rangle_M \equiv \langle \hat{A} \rangle_M, \quad (3.11)$$

where  $\langle \dots \rangle_M$  denotes the average over the Metropolis sampling of  $\mathbf{n}$ . When the matrix  $\langle \mathbf{n} | \hat{A} | \mathbf{n}' \rangle$  is sparse, the sum over  $\mathbf{n}'$  in Eq.(3.12) can easily be calculated.

The network parameters  $\mathbf{W}$  and  $\mathbf{h}$  in Eqs.(3.7) and (3.8) are optimized so that the expectation value of the Hamiltonian  $\langle \hat{H} \rangle$  becomes minimum. Although the stochastic reconfiguration method is more stable, for simplicity, we use the steepest descent method for the optimization. The derivative of the energy with respect to the network parameter is given by

$$\begin{aligned} \frac{\partial \langle \hat{H} \rangle}{\partial w} &= \frac{\sum_{n,n'} O_w^*(\mathbf{n}) \langle \mathbf{n} | \hat{H} | \mathbf{n}' \rangle \frac{\psi(\mathbf{n}')}{\psi(\mathbf{n})} + O_w(n') \langle \mathbf{n} | \hat{H} | \mathbf{n}' \rangle}{\sum_n |\psi(\mathbf{n})|^2} \\ &\quad - \langle \hat{H} \rangle \frac{\sum_n \{O_w^*(n) + O_w n\} |\psi(\mathbf{n})|^2}{\sum_n |\psi(\mathbf{n})|^2} \quad (3.12) \\ &\simeq 2Re(\langle O_w^* \tilde{H} \rangle_M - \langle O_w^* \rangle_M \langle \tilde{H} \rangle_M), \end{aligned}$$

where  $w$  is one of the network parameters  $\mathbf{W}$  or  $\mathbf{h}$ , and

$$O_w(\mathbf{n}) = \frac{1}{\psi(\mathbf{n})} \frac{\partial \psi(\mathbf{n})}{\partial w}. \quad (3.13)$$

There are various ways to update the network parameters to reduce the expectation value of the Hamiltonian. In the steepest-descent method, the *i*th network parameter  $w_i$  is updated as

$$w_i \rightarrow w_i - \alpha \frac{\partial \langle \hat{H} \rangle}{\partial w_i}, \quad (3.14)$$

where  $\alpha < 1$  controls the magnitude of the change in each update. More efficient methods to update the network parameters have been developed in the field of machine learning. In the following we concentrated on three method - Adam, AdaDelta, AdaGrad. According several times experiments, we confirmed The Adam method is the best optimize method among of them.

At first, in the Adam method, the network parameters are updated as

$$v_i \rightarrow v_i + \left( \frac{\partial \langle \hat{H} \rangle}{\partial w_i} \right)^2 \\ (3.15)$$

$$w_i \rightarrow w_i + \frac{\gamma}{\sqrt{v_i} + \epsilon} \frac{\partial \langle \hat{H} \rangle}{\partial w_i},$$

where  $\epsilon \ll 1$  avoids division by zero and  $\gamma < 1$ . The Adam method is given by

$$m_i \rightarrow \beta_1 m_i + (1 - \beta_1) \frac{\partial \langle \hat{H} \rangle}{\partial w_i},$$

$$v_i \rightarrow \beta_2 v_i + (1 - \beta_2) \left( \frac{\partial \langle \hat{H} \rangle}{\partial w_i} \right)^2, \quad (3.16)$$

$$w_i \rightarrow w_i - \delta \frac{m^i}{1 - \beta_1^\varrho} \frac{1}{\sqrt{\frac{v_i}{1 - \beta_1^\varrho}} + \epsilon},$$

for the  $\varrho$ th update, where  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  are usually used, and  $\delta < 1$ . The initial values of  $v_i$  and  $m_i$  in Eqs.(3.15) and (3.16) are zero, and the parameters  $\beta$ ,  $\gamma$ , and  $\delta$  are chosen so that the optimization works efficiently. It is known that if a valley exists in the energy landscape, the steepestdescent method may cause oscillation between the two sides of the valley, while AdaGrad and Adam can avoid oscillation and efficiently decrease the energy[?].

# Chapter 4

## Results

Here you present the results of your study (if you carried out one), and any data analysis you may have performed to answer your question.

### 4.1 Network Model Parameters

In this section, we shows some results that compare of different network model parameters. We consider a one-dimensional (1D) system of  $N = 64$  bosons on  $M = 64$  sites with the periodic boundary condition based on Eq.(2.1).

We first compare different neural network models: Fully Connect Neural Networks(FCNNs) and Convolution Neural Networks(CNNs). FCNN are the oldest and most common type of neural networks. Basically, the first mathematical model of a multilayer neural network, called multilayer perceptron (MLP), was a fully connected neural network. Artificial neurons are the base of all neural networks. They are units inspired by biological neurons. Each artificial neuron receives inputs and generates a single output that we transmit to multiple other neurons. Inputs are typically numeric values from a sample of external data, but they can also be the outputs of other neurons.

The outputs of the final output neurons of the neural network represent the value that defines prediction. CNN is a type of neural network that has at least one convolution layer. Besides convolutional and pooling layers typical for CNN, this type of network usually includes fully-connected layers. But in this thesis, we didn't use pooling layer. Unlike an artificial neuron in a fully-connected layer, a neuron in a convolutional layer is not connected to the entire input but just some section of the input data.

Fig.4.1 shows the energy  $\langle \tilde{H} \rangle_M$  as a function of the number of updates using two different neural network models. We can see that the energy decreases quickly for the initial updates and then gradually converges to the final value. When these update rates are smaller, the convergence becomes slower, and when the update rates are too large, the calculation becomes unstable. The convergence is faster CNNs than for FCNNs.

And then, we compared different optimal methods. Fig.4.2 shows the energy  $\langle \tilde{H} \rangle_M$  as a function of the number of updates using different optimal methods. The energy also decreases quickly for the initial updates and then gradually converges to the final value. The convergence is faster Adam than for AdaDelta and AdaGrad. Adam is an adaptive learning rate optimization algorithm that utilises both momentum and scaling, combining the benefits of RMSProp and SGD with Momentum. The optimizer is designed to be appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients. The weight updates are performed as Eqs.(3.15) and (3.16).

Finally, we compared different activation functions. Fig.4.3 shows the energy  $\langle \tilde{H} \rangle_M$  as a function of the number of updates using different optimal methods. We select seven activation functions as the samples. Obviously, All the ground energy decreases quickly for the initial updates and then

gradually converges to the final value and stable. The convergence is faster Rectified Linear Activation Function(ReLU) than for others. In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input. ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance. The sigmoid and hyperbolic tangent activation functions(tanh) cannot be used in networks with many layers due to the vanishing gradient problem. ReLU can overcomes the vanishing gradient problem, allowing models to learn faster and perform better. It's also the default activation when developing multilayer Perceptron and convolutional neural networks.

Description	Values
Hopping strength(J)	-1
On-site interaction energy(U)	1
Bosons(N)	64
Sites(M)	64
Samples	1000
Training steps	1024

Table 4.1: Model Parameters.

## 4.2 Correlation Properties

In this section, we based on section.4.1 network model parameters constructed an efficiency network model, this model combine by , ***CNNs, Adam, Relu***. We consider a 1D Extended Bose-Hubbard Model system of ( $N = 8/M = 32$ ) bosons on ( $M = 8/M = 32$ ) sites with the periodic boundary condition

based on Eqs.(2.2) and (2.3). And then, we discuss correlation properties of eBHM by these results.

At first, We discuss the ground state energy change of 1D eBHM on two phase-SuperFluid(SF), Mott Insulator(MI). Based on Table.4.2 model parameters we did several training by network model and drew out some conclusions to prove the properties of 1D eBHM. Firstly, we selected a small system ( $N = 8, M = 8$ ), according Fig.(4.4) and Fig.(4.5), we can draw the conclusion: All the energy decreases quickly for the initial updates and then gradually converges to the final value and stable. The convergence is faster in MI phase than for SF phase.

Description	SF	MI
Hopping strength( $J$ )	-1	-1
On-site interaction energy( $U$ )	1	6
Nearest-neighbor interaction( $V$ )	1	1
Bosons( $N$ )	8	8/32
Sites( $M$ )	8	8/32
Samples	1000	1000
Training steps	1024	1024

Table 4.2: Model Parameters.

And then, we discuss 1D eBHM correlation properties in *SF&MI* phase. Since open boundary conditions are used, special care has to be taken to reduce boundary effects. The most obvious form of these are local density oscillations. In the SF phase they show the power-law decay away from the edge of the system characteristic for the Luttinger liquid. In the MI phase the boundary effects decay exponentially. Still based on a small system ( $N = 8, M = 8$ ), according Fig.(4.6) and Fig.(4.7), the 1D eBHM correlation value all keep stable equal to 1 in both SF phase and MI phase at distance ( $r = 0$ ). As the  $r$  increases, the correlation converges and keep stable.

Refer to (*PCBs*) periodic boundary conditions(Chosen for approximating a large (infinite) system by using a small part called a unit cell). In quantum mechanics it is often useful to imagine that the system under consideration is contained in a box represented by walls of infinitely large potential energy. The PCBs imply that the value of the wave function at a point situated on one wall is the same as the value at the corresponding point on the opposite wall. Fig.(4.6) and Fig.(4.7) shows us: no matter in SF phase or MI phase, when ( $r = 3$ ) and ( $r = 5$ ), the correlation finally converges to the same value and keep stable. As we know, position ( $r = 3$ ) and position ( $r = 5$ ) is in opposite position. And then, Fig.(4.8) shows us, even in a bigger system ( $N = 32, M = 32$ ), our network model also can perfectly intuitive this properties. When ( $r = 3$ ) and ( $r = 29$ ) the correlation finally converges to the same value and keep stable. Fig.(4.9) shows us the correlation properties of  $r$  dependence. In this situation, the system from SF phase transits to MI phase by increasing  $U$ ) (According to Fig.(2.3)). This result shows that all the correlation value decreases for the distance increases. In the SF phase correlation value show the power-law decay. In the MI phase it effects decay exponentially.

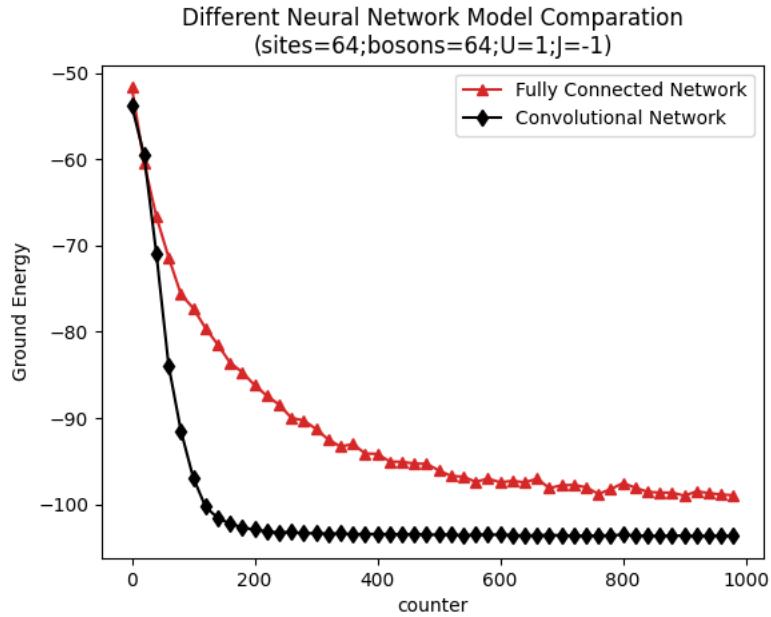


Figure 4.1: Description of different Neural Network Model compare results based on Table.4.1. Obviously, the ground energy decreases quickly for the initial updates and then gradually converges to the final value and stable. The convergence is faster CNNs than for FCNNs.

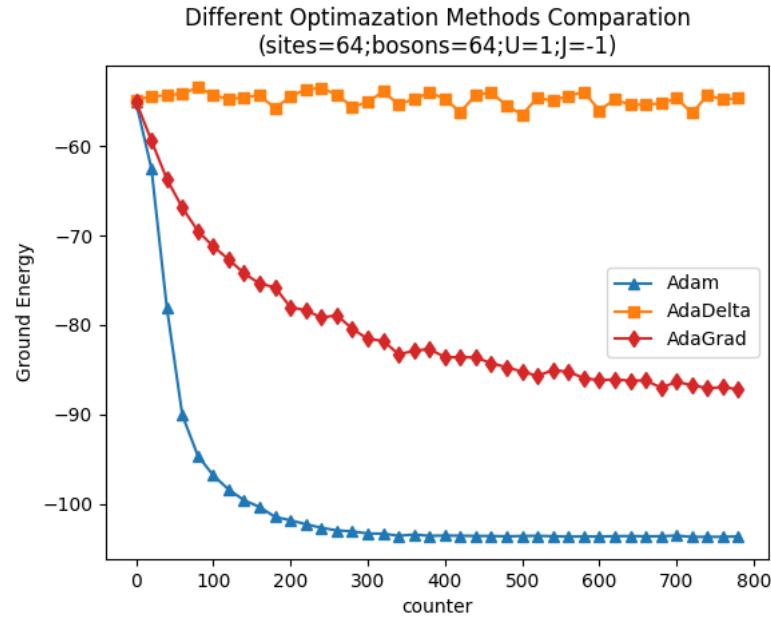


Figure 4.2: Description of different Optimazation methods compare results based on Table.4.1. The convergence is faster Adam than for AdaGrad and AdaDelta.

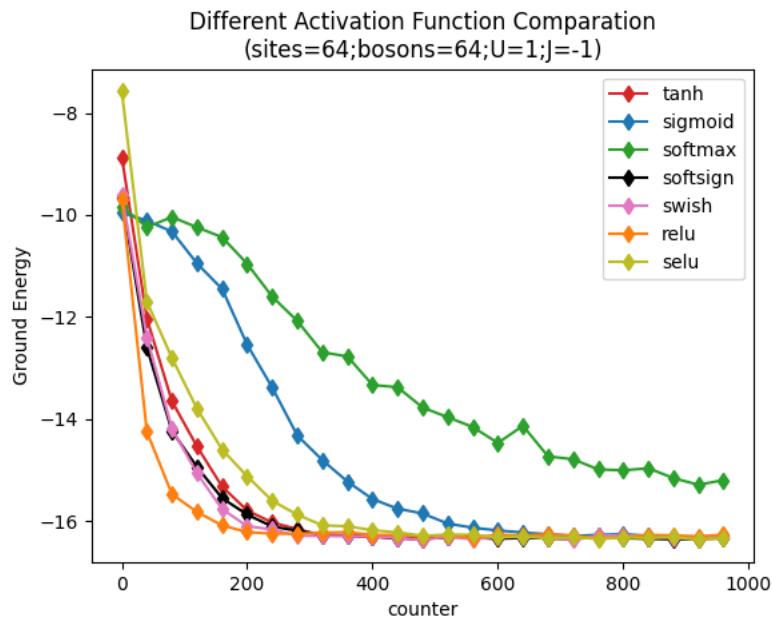


Figure 4.3: Description of different Activation functions compare results based on Table.4.1. Obviously, all the ground energy decreases quickly for the initial updates and then gradually converges to the final value and stable. When these update rates are smaller, the convergence becomes slower, and when the update rates are too large, the calculation becomes unstable. The convergence is faster Relu than for others

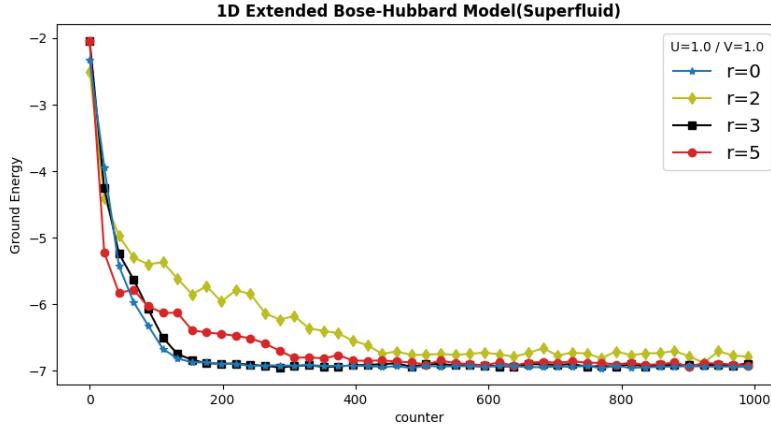


Figure 4.4: Description of 1D eBHM ground state changing situation in SF based on Table.4.2.  $(U, V) = (1.0, 1.0)$  is mean the System stay in the SF phase(According Fig.(2.3)). It's showed all the ground energy decreases quickly for the initial updates and then gradually converges to the final value and stable. And as the  $r = (0, 2, 3, 5)$  increases, the ground state energy converges faster.

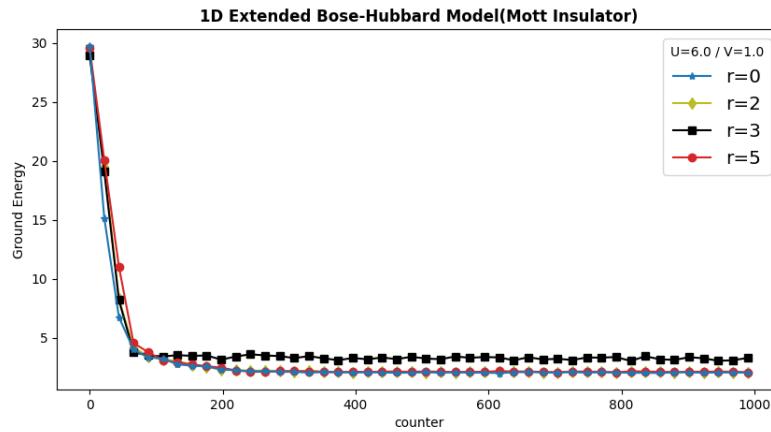


Figure 4.5: Description of 1D eBHM ground state changing situation in SF based on Table.4.2.  $(U, V) = (6.0, 1.0)$  is mean the System stay in the MI phase(According Fig.(2.3)). It's also showed all the ground energy decreases quickly for the initial updates and then gradually converges to the final value and stable. And as the  $r = (0, 2, 3, 5)$  increases, the ground state energy converges faster.

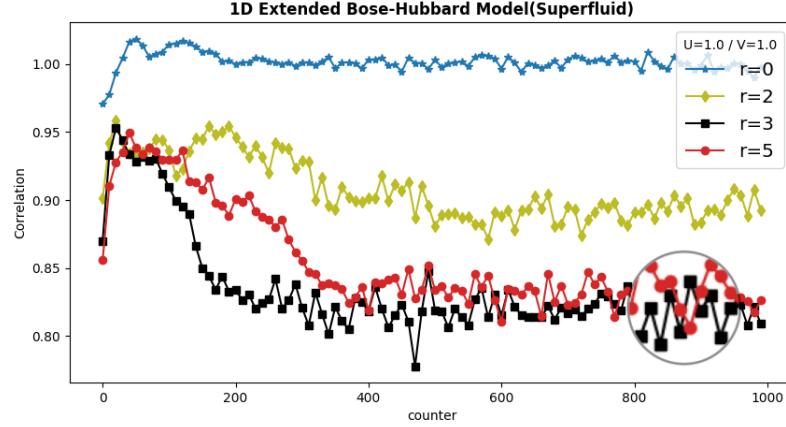


Figure 4.6: Description of 1D eBHM( $N = 8, M = 8$ ) correlation properties of distance-dependence in SF based on Table.4.2. ( $U, V = (1.0, 1.0)$ ) is mean the System stay in the SF phase(According Fig.(2.3)). Correlation almost equal to 1, when  $r = 0$ . And then, it deceases by the  $r$  increases. According PCBs, correlation finally almost equal in the opposite position ( $r = 3, r = 5$ ).

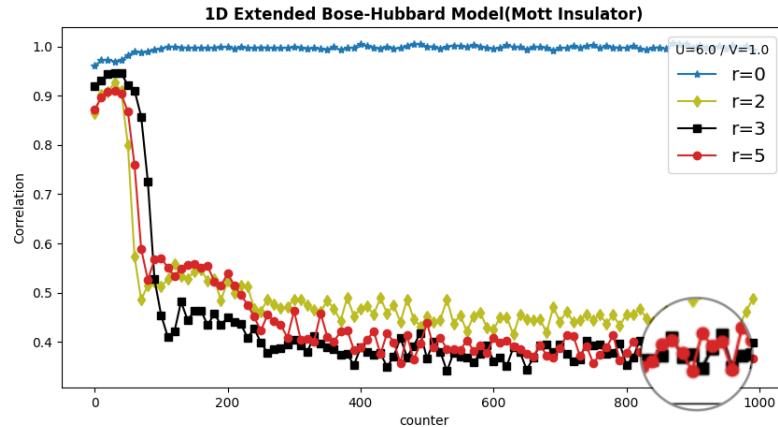


Figure 4.7: Description of 1D eBHM( $N = 8, M = 8$ ) correlation properties of distance-dependence in SF based on Table.4.2. ( $U, V = (6.0, 1.0)$ ) is mean the System stay in the MI phase(According Fig.(2.3)). Correlation almost equal to 1, when  $r = 0$ . And then, it deceases by the  $r$  increases. According PCBs, correlation finally almost equal in the opposite position ( $r = 3, r = 5$ ).

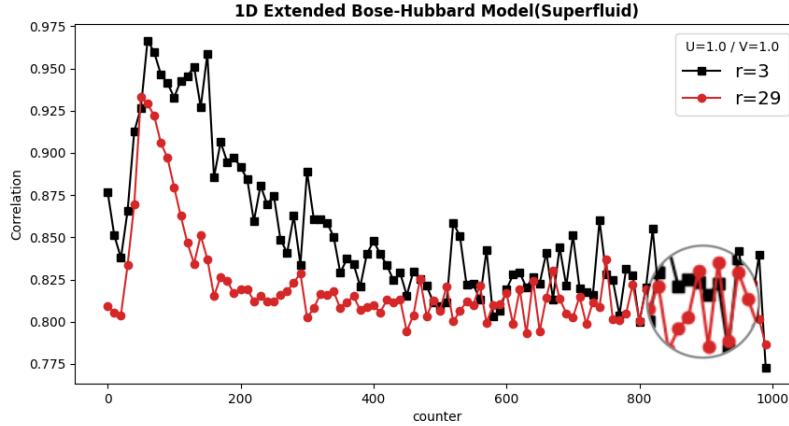


Figure 4.8: Description of 1D eBHM( $N = 32, M = 32$ ) correlation properties of distance-dependence in SF based on Table.4.2. According PCBs, correlation finally almost equal in the opposite position ( $r = 3, r = 29$ ).

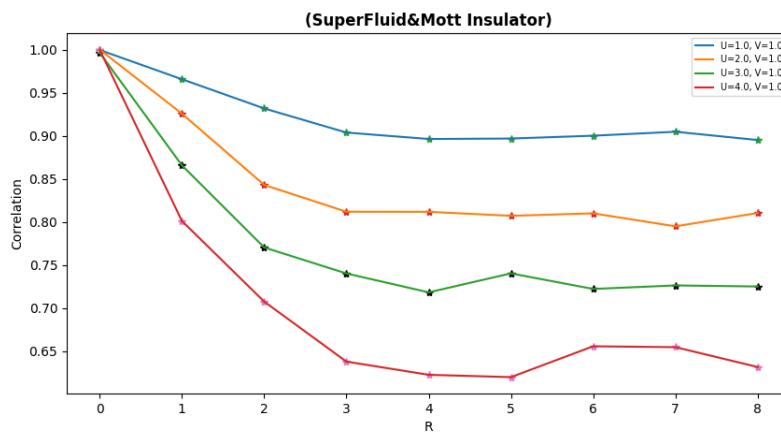


Figure 4.9: Description of 1D eBHM( $N = 32, M = 32$ ) correlation properties of distance-dependence in SF phase and MI phase. According Fig.(2.3), When  $U = 1, U = 2$ , system stay in SF phase,  $U = 3, U = 4$ , system stay in MI phase. All the correlation value decreases for the distance increases. In the SF phase correlation value show the power-law decay. In the MI phase it effects decay exponentially.

# Chapter 5

## Conclusion

### 5.1 Theoretical and Practical Implications

In conclusion, at first, a method to obtain the ground state of the Bose-Hubbard model using an artificial neural network was proposed. It was demonstrated that the approximate ground state can be obtained by a simple optimization scheme of the network parameters. We adjusted different network model parameters to supervise the ground state coverage performance. And we finally drew a conclusion to find out the most suitable network model configuration-*CNNs*, *Adam* optimal method, *Relu* activation function.

And then based on the above network model configurations, we continue to research 1D extend Bose-Hubbard model Correlation properties. Firstly, we use a small system( $N = 8, M = 8$ ) to proved that no matter in SF phase or Mi phase, all ground energy decreases by distance  $r$ ) increases. Correlation value alomost equal to 1, when distance  $r = 0$ . No matter in SF phase or Mi phase, all correlation value decreases by distance  $r$ ) increases. But according PCBs, correlation value finally converges to the same value opposite position

point. Even if in a bigger system, our network model also performed nicely. In the SF phase correlation value show the power-law decay. In the MI phase it effects decay exponentially.

There may be a variety of extensions of the present study. The present method can easily be extended to multiple layers, which is interesting from the viewpoint of deep learning. Much larger systems may be explored using an existing neural-network framework optimized for GPU computing, which enables us to obtain phase structures in the thermodynamic limit.

## 5.2 Future Work

For the machine learning, there are many different solution models of machine learning due to the difference problems. We can find out more effective network model to deal with quantum many-body problem by supervising the model performance. And about the correlation properties of 1D extend Bose-Hubbard model, this thesis just researched the correlation properties in Superfluid (SF) phase and Mott Insulator (MI) phase situation. We can try to discuss more complex phase - Haldane insulator (HI) and density-wave state (DW).

## 5.3 Acknowledgments

I sincerely thank my professor Hiroki Saito and all my classmates fruitful help, and financial support from UEC and my parents.