Machine-learning quantum mechanics: Solving quantum mechanics problems using radial basis function networks

Peiyuan Teng[*]

Department of Physics The Ohio State University Columbus, Ohio, 43210, USA

(Received 10 October 2017; revised manuscript received 15 April 2018; published 14 September 2018)

In this article, machine-learning methods are used to solve quantum mechanics problems. The radial basis function network in a discrete basis is used as the variational wave function for the ground state of a quantum system. Variational Monte Carlo (VMC) calculations are carried out for some simple Hamiltonians. The results are in good agreement with theoretical values. The smallest eigenvalue of a Hermitian matrix can also be acquired using VMC calculations. Results are provided to demonstrate that machine-learning techniques are capable of solving quantum mechanical problems.

DOI: 10.1103/PhysRevE.98.033305

I. INTRODUCTION

Machine-learning theory has been developing rapidly in recent years. Machine-learning techniques have been successfully applied to solve a variety of problems, such as email filtering, optical character recognition (OCR), and natural language processing, and have become a part of everyday life. In the physical sciences, researchers are also applying machine-learning methods to explore new possibilities. For example, machine-learning methods are used in molecular dynamics [1,2], as a way to bypass the Kohn-Sham equation in density functional theory [3], to assist in materials discovery [4], or to identify phase transitions [5]. Considering the power of machine learning, it is interesting to consider solving quantum mechanics problems using machine-learning methods.

Artificial neural networks (ANNs) [6], which are inspired by biological neural networks, are one of the most important methods in machine-learning theory. An ANN consists of a network of artificial neurons, and examples of ANNs include feed-forward neural networks [7], radial basis function (RBF) networks [8], and restricted Boltzmann machines [9]. As a universal approximator [10,11], an ANN can be used to represent functions, and it is possible to use an ANN as a representation of the wave function in a quantum system.

Researchers have been trying to combine neural network theory and quantum mechanics, for example, using a neural network in the real space to solve differential equations, especially the Schrödinger equation with some specific potential [12]. Another example is the quantum neural network [13], where information in an ANN is processed quantum mechanically. One of the most promising works was the recent research by Carleo and Troyer in Ref. [14], where the restricted Boltzmann machine was used as the variational Monte Carlo (VMC) ground state wavefunction. In their work, the ground state of a many-body system could be efficiently represented by a neural network. Following their work, other possibilities were also explored. Most recently, in Ref. [15], a three-layer feed-forward neural network was used to calculate the ground state energy of the Bose-Hubbard model. Machine-learning methods were shown to be able to distinguish between different phases, even for systems with the sign problem [16]. VMC methods do not suffer from the fermion sign problem; therefore, using a neural network as a VMC ansatz is very promising and has the potential to tackle the calculations that are almost impossible in other Monte Carlo methods.

In this article, the possibility of using an RBF network to represent the wave function of a quantum mechanical system is discussed. Our work involves two major aspects. First, the representation power of the RBF network is illustrated, which has not been discussed in the physics literature. Second, instead of a lattice system, where the dimension of the Hilbert space of each site is finite, a general quantum mechanical system with infinite or continuous degrees of freedom is discussed. A binary restricted Boltzmann machine is not sufficient for the simulations of such a system; therefore, it is interesting to search for new ansatz. An RBF network is one of the candidates.

In our work, a VMC procedure is formulated, where an RBF network is used as the variational wave function. A harmonic oscillator in a linear potential and a particle in a box with a linear potential are then used as benchmarks. Furthermore, we discuss the possibility of using the VMC method to solve for the lowest eigenvalue of a matrix.

This article is organized as follows. In Sec. II, artificial neural network theory and variational Monte Carlo theory are reviewed. Section III contains major results, that is, quantum mechanical problems are solved using the radial basis neural network. In Sec. IV, we discuss some related questions.

II. ARTIFICIAL NEURAL NETWORK THEORY AND THE VARIATIONAL MONTE CARLO METHOD

In this section, two cornerstones of this work will be introduced, which are the artificial neural network theory and the variational Monte Carlo method.

[*]teng.73@osu.edu

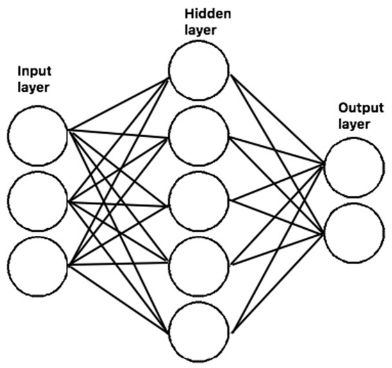

FIG. 1. An illustration of the artificial neural network. A typical neural network consists of three layers of neurons: the input layer, the hidden layer, and the output layer. Each neuron is represented by a circle. The lines between layers are associated with the parameters of the neural network.

A. Artificial neural network theory

Inspired by the biological neural network model, ANN theory was proposed by McCulloch and Pitts in 1943 [6], in an attempt to propose a mathematical description of the biological nervous system. Figure 1 illustrates a simple example of a neural network which consists of three layers of artificial neurons.

Neural networks are widely used tools in machine-learning theory, for example, as a function approximation tool in supervised learning. The goal is to find the optimal parameters by minimizing the cost function. This can be a highly nontrivial problem when there are a large number of parameters. For such algorithms such as the back-propagation, please see Ref. [7].

In a typical machine-learning problem using neural network methods, the input neuron can be a binary number. For example, in a handwritten digit recognition problem, each input neuron corresponds to a pixel in a figure and takes a value of 0 or 1. The input values are processed through the neural network using, for example, the rules mentioned above. The output values of the neural network are compared with the objective values, and the error is minimized by finding the optimal parameters.

In this article, the radial basis function (RBF) network is used as a variational wave-function ansatz. For example, for a three-layer RBF network with one single output neuron, the output function $z(\boldsymbol{x})$ of the neural network can be written as

$$z(\boldsymbol{x}) = \sum_{i=1}^{M} a_i \rho_i(||\boldsymbol{x} - \boldsymbol{c}_i||). \tag{1}$$

In this output function, a_i and $\boldsymbol{c}_i$ are parameters of the neural network. $\boldsymbol{x}$ is the input vector which has the same dimension as $\boldsymbol{c}_i$. M is the number of neurons in the hidden layer. $\rho(||\dots||)$ is the radial basis function which can be a Gaussian function with a Euclidean norm:

$$\rho_i(||\boldsymbol{x} - \boldsymbol{c}_i||) = e^{-|b_i||\boldsymbol{x}-\boldsymbol{c}_i|^2}, \tag{2}$$

or an exponential absolute value function

$$\rho_i(||\boldsymbol{x} - \boldsymbol{c}_i||) = e^{-|b_i||\boldsymbol{x}-\boldsymbol{c}_i|}. \tag{3}$$

Other activation functions, such as multiquadratics

$$\rho_i(||\boldsymbol{x} - \boldsymbol{c}_i||) = \sqrt{|\boldsymbol{x} - \boldsymbol{c}_i|^2 - |b_i|^2}, \tag{4}$$

or inverse multiquadratics

$$\rho_i(||\boldsymbol{x} - \boldsymbol{c}_i||) = (|\boldsymbol{x} - \boldsymbol{c}_i|^2 - |b_i|^2)^{-\frac{1}{2}}, \tag{5}$$

are also commonly used in the machine-learning community. These activation functions can also be understood as kernel functions. In the activation functions, $|b_i|$ are parameters that control the spread of the activation function. Other activation functions are also possible; discussions about the activation function can be found in Ref. [17].

In addition to the RBF network, many different types of neural networks can be constructed, such as the restricted Boltzmann machines or the autoencoders, which are widely used in deep learning technology. The universal approximation theorem establishes the mathematical foundation of neural network theory, which states that neural network functions are dense in the space of continuous functions defined on a compact subset of R^n, under some assumptions about the activation function and given enough hidden neurons [10,11].

In this paper, the RBF network is used as a variational wave function represented in a discrete eigenbasis. Note that we use $|b_i|$ as a variational parameter in our calculations instead of a constant number as in a regular RBF network. The absolute value of $|b_i|$ is for the stability of the optimization.

When neural network methods are applied to quantum physics, the inputs of the neural network can take discrete quantum numbers. After being processed through the neural network, the outputs of the neural network represent the amplitudes of the wave function on the basis vector labeled by the input quantum numbers. The neural network is then trained by minimizing the energy expectation value. For example, for a three-dimensional quantum harmonic oscillator in an orthogonal coordinate system, we can use a neural network with three input neurons, where each input can take integer values for 0 to ∞. The trained neural network should represent the ground state of this system, in which, after proper normalization, the output should be 1 given 000 as the input, and 0 for other inputs.

B. Variational Monte Carlo method (VMC)

The VMC method, first proposed by McMillan in 1965 [18], combines the variational method and the Monte Carlo method in order to evaluate the ground state of a quantum system.

Start from a Hamiltonian $\hat{H}$ and a variational wave function $|\psi(\lambda)\rangle$, where λ is a set of variational parameters, the energy expectation value can be written as

$$E(\lambda) = \frac{\langle\psi(\lambda)|\hat{H}|\psi(\lambda)\rangle}{\langle\psi(\lambda)|\psi(\lambda)\rangle}. \tag{6}$$

This energy expectation value can be computed using the widely known Metropolis algorithm [19], which is one of the most efficient algorithms in computational science. As a Markov chain Monte Carlo method, it may currently be the only efficient algorithm for evaluating a multidimensional integral.

The next step of the VMC method is to minimize the energy in the parameter space. This can be a difficult problem when there are many variational parameters. Two examples of such algorithms are the linear method [20] and the stochastic reconfiguration method [21]. The minimization algorithm gives the minimum of the energy in the parameter space, and it is reasonable to use this value as our approximation for the ground state energy. For a detailed review of the VMC method, please refer to Ref. [22].

Currently, physicists believe that the accuracy of the VMC method depends, to a great extent, on a proper choice of the variational wave function; therefore, it is important to choose a wave function based on physical intuition or a physical understanding of the system. This belief may not be true in the age of machine learning. Neural network functions are capable of approximating unknown functions by maximizing or minimizing the objective function. It would be interesting to further explore the possibility of using a neural network function as the variational wave function of a quantum system.

III. SOLVING QUANTUM MECHANICS PROBLEMS USING ARTIFICIAL NEURAL NETWORK

In the pioneering work of Carleo and Troyer [14], restricted Boltzmann machine (RBM) was used as a variational wave function for many-body systems. The transverse-field Ising model and antiferromagnetic Heisenberg model were benchmarked using the RBM wave function. Variational Monte Carlo calculations were carried out. Their results demonstrate that a neural network wave function is capable of capturing the quantum entanglement of the ground states and giving an accurate estimation of the ground state energy.

In this article, we continue developing this idea using artificial neural network functions as the ground state variational wave function. In Ref. [14], the restricted Boltzmann machine is only binary valued; we will demonstrate the representation power of a neural network wave function without this constraint. In addition, we discuss the possibility of using a neural network wave function to solve a generic quantum mechanics problem. This VMC method behaves at least as accurate as the perturbation theory.

A. Theoretical outline

Consider a quantum system which has countable number of basis vectors, an arbitrary state $|\psi\rangle$ in the Hilbert space can be represented by

$$|\psi\rangle = \sum_{n_1,n_2,\dots,n_p} \psi(n_1,n_2,\dots,n_p)|n_1,n_2,\dots,n_p\rangle, \tag{7}$$

where $|n_1,n_2,\dots,n_p\rangle$ is a basis vector labeled by quantum number n_i, $i=1,2\dots p$, and p is the number of sites in the system. For example, for the Heisenberg model, p represents the number of spins; for a three-dimensional harmonic oscillator in a Cartesian coordinate, we could use n_1,n_2,n_3 to label three quantum numbers. $\psi(n_1,n_2,\dots,n_p)$ is the amplitude of $|\psi\rangle$ on basis vector $|n_1,n_2,\dots,n_p\rangle$. We can interpret this amplitude as a function of $n_1,n_2,\dots,n_p$. A similar ansatz is also used in Ref. [15].

This function can be represented by a neural network with *one* output neuron. Using an RBF network, the amplitude function can be written as

$$\psi(n_1,n_2,\dots,n_p;\boldsymbol{a},\boldsymbol{c}) = \sum_i^M a_i\rho_i(||\boldsymbol{n}-\boldsymbol{c}_i||), \tag{8}$$

with $\boldsymbol{n}$ representing an array of quantum numbers and

$$\rho_i(||\boldsymbol{x}-\boldsymbol{c}_i||) = e^{-|b_i||\boldsymbol{x}-\boldsymbol{c}_i|^2}. \tag{9}$$

One reason to choose this neural network is that the Gaussian activation function guarantees that the amplitude does not diverge when $n\to\infty$.

Practically, it is useful to truncate the quantum number n_i if its range is countably infinite. This is not necessary for a spin-half lattice system since n_i can only take two values. For a harmonic oscillator, however, we may truncate the quantum number at some finite value. The universal approximation theorem is only valid for a closed space. This truncation will also facilitate numerical simulations.

Using this variational wave function, the energy expectation value is

$$E(\boldsymbol{\lambda}) = \frac{\langle\psi(\boldsymbol{\lambda})|H|\psi(\boldsymbol{\lambda})\rangle}{\langle\psi(\boldsymbol{\lambda})|\psi(\boldsymbol{\lambda})\rangle} = \frac{\int|\psi(\boldsymbol{n};\boldsymbol{\lambda})|^2E_{\text{local}}(\boldsymbol{n};\boldsymbol{\lambda})d\boldsymbol{n}}{\int|\psi(\boldsymbol{n};\boldsymbol{\lambda})|^2d\boldsymbol{n}}, \tag{10}$$

with

$$E_{\text{local}}(\boldsymbol{n};\boldsymbol{\lambda}) = \frac{\langle\boldsymbol{n}|H|\psi(\boldsymbol{\lambda})\rangle}{\langle\boldsymbol{n}|\psi(\boldsymbol{\lambda})\rangle} = \frac{\sum_{n'}\langle\boldsymbol{n}|H|\boldsymbol{n}'\rangle\langle\boldsymbol{n}'|\psi(\boldsymbol{\lambda})\rangle}{\langle\boldsymbol{n}|\psi(\boldsymbol{\lambda})\rangle}, \tag{11}$$

Here, $\boldsymbol{\lambda}$ represents all the variational parameters, for example, a_i, b_i, and $\boldsymbol{c}_i$.

The energy expectation can be evaluated using the Metropolis algorithm. After initialization and thermalization, repeat these two steps until equilibrium: (1) generate a move from configuration $\boldsymbol{n}$ to $\boldsymbol{n}''$. (2) Using proper transition probability, accept or reject the move with probability $\min(1,|\frac{\langle\boldsymbol{n}''|\psi(\boldsymbol{\lambda})\rangle}{\langle\boldsymbol{n}|\psi(\boldsymbol{\lambda})\rangle}|^2)$. The expectation value of other operators can be evaluated similarly.

Compared with exact diagonalization, one advantage of this formalism is that the matrix element $\langle\boldsymbol{n}|H|\boldsymbol{n}'\rangle$ is never stored explicitly. Only the nonzero matrix elements are needed to be valued and summed during the sampling process.

The energy as a function of parameters $\boldsymbol{\lambda}$ can be, for example, minimized using the stochastic reconfiguration method [21]. In the stochastic reconfiguration method, an operator

$$O_i(\boldsymbol{n}) = \frac{\partial_{\lambda_i}\psi_{\boldsymbol{\lambda}}(\boldsymbol{n})}{\psi_{\boldsymbol{\lambda}}(\boldsymbol{n})} \tag{12}$$

can be defined for each parameter in the variational wave function.

For a radial basis neural network with the Gaussian basis function

$$O_{a_i}(\boldsymbol{n}) = \frac{\rho_i}{\psi}, \tag{13}$$

$$O_{b_i}(\boldsymbol{n}) = -\frac{a_ib_i|\boldsymbol{n}-\boldsymbol{c}_i|^2\rho_i}{|b_i|\psi}, \tag{14}$$

$$O_{c_{ij}}(\boldsymbol{n}) = \frac{2a_i|b_i|(n_j-c_{ij})\rho_i}{\psi}, \tag{15}$$

where c_{ij} is the jth component of of $\boldsymbol{c}_i$. The covariance matrix and forces are defined as

$$S_{ij} = \langle O_i^* O_j \rangle - \langle O_i^* \rangle \langle O_j \rangle, \tag{16}$$

$$F_i = \langle E_{\text{local}} O_i^* \rangle - \langle E_{\text{local}} \rangle \langle O_i^* \rangle. \tag{17}$$

The parameters can be updated by

$$\lambda_j' = \lambda_j + \alpha S_{ij}^{-1} F_i. \tag{18}$$

Here, $\langle \dots \rangle$ is the expectation value of an operator. α can be understood as the learning rate of the optimization algorithm. A regularization $S_{ii}' = S_{ii} + r(k) S_{ii}$ is applied to the diagonal elements of matrix S in all our calculation, where $r(k) = \max(100 \times 0.9^k, 10^{-4})$ [14]. This process iterates until the optimization converges, and we treat the converged energy as our best approximation of the ground state energy.

In this article, the method mentioned above is used for the optimization. We notice that the recent work of Saito [15], in which feed-forward neural network was successfully used to represent the ground state of the Bose-Hubbard model. In their work, an exponential function was written based on the output of the feed-forward neural network. It is an interesting question whether an exponential of feed-forward neural network output function can be used to represent a quantum mechanical wave function.

B. One-dimensional quantum harmonic oscillator in electric field

To start with, we would like to benchmark the quantum harmonic oscillator. Since we use a set of discrete quantum numbers to describe the variational wave function, it is natural to use the energy eigenbasis of an unperturbed harmonic oscillator to calculate the matrix element.

Consider the one-dimensional (1D) Hamiltonian

$$H = \frac{\hat{p}^2}{2} + \frac{\hat{x}^2}{2} + E\hat{x} = H_0 + E\hat{x}, \tag{19}$$

where E is a parameter that can be understood as the electric field.

Using natural units, it is easy to see that the ground state energy of H_0 is 0.5. Assuming the eigenstates of H_0 are labeled by $|n\rangle$, the variational ansatz for the ground state of H can be approximated by

$$|\psi\rangle = \sum_{n=0}^{n_{\max}-1} \psi(n)|n\rangle, \tag{20}$$

with $\psi(n)$ represented by an RBF network with one input neuron, and we truncate the quantum number to $n_{\max} - 1$. In this notation, the RBF network represents the function $\psi(n)$. The variable n can take different values, for example, if $n = 1$, the output of the neural network is the coefficient on the basis vector $|1\rangle$, which is $\psi(1)$. The neural network represents the function ψ, and the coefficient on basis vector $|n\rangle$ is represented by $\psi(n)$.

We use the VMC procedure described in Sec. III A to conduct the calculation. The parameters are initialized randomly. Our codes are written in C++, where the matrix solving library Eigen [23] is used for the stochastic reconfiguration. Sample codes will be available at https://github.com/peiyuanteng.

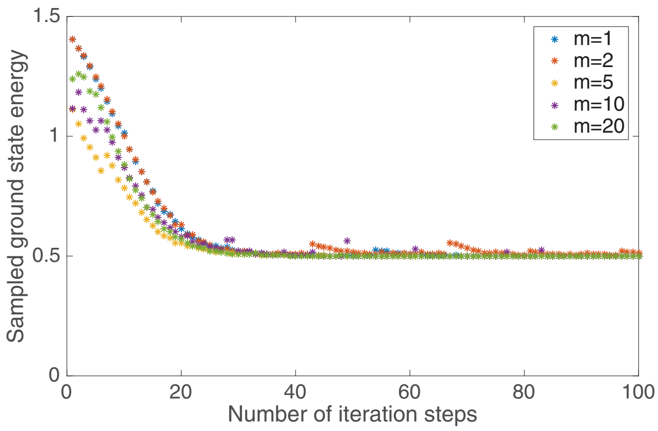

FIG. 2. Minimization of the ground state energy of H at $E = 0$, using Gaussian radial basis network. m is the number of hidden layers in the neural network.

A neural network with random parameters is first created. Then, the ground state energy under one set of parameters are calculated using the Monte Carlo method. The state space of the Monte Carlo sampling is a truncated discrete space denoted by $\boldsymbol{n}$. Specifically, our quantum number is the quantum number of the unperturbed Hamiltonian H_0, and the basis is the eigenbasis of H_0. We are trying to solve for the ground state of the perturbed one. A random plus or minus move is generated for each sample and accepted using the Metropolis algorithm. In this work, when a random move yields a quantum number that is below zero or above $n_{\max} - 1$ at the boundary of state space, the quantum number is reflected back in order to satisfy the detailed balance condition. For each specific $\boldsymbol{n}$, we can plug it into the neural network and get its amplitude. During the Monte Carlo process, 50000 samples are used. Being able to calculate the energy, we can then use the stochastic reconfiguration method to find the minimal energy, and we treat this energy as our best approximation of the ground state energy.

In Fig. 2, we illustrate the minimization of ground state energy during the iteration process using the Gaussian basis function [see Eq. (2)]. The learning rate is set at 0.1. m denotes the number of neurons in the hidden layer.

Alternatively, we can use the exponential absolute value function as the RBF [see Eq. (3)]. Under the same learning rates, this RBF network also converges to the correct eigenvalue (see Fig. 3). It is easy to see that the Gaussian RBF network behaves better than the others. Based on our experience, the Gaussian network also performs better in other cases, therefore, we use the Gaussian network in later examples.

Remarks. We use n as our variable for the variational wave function. The output of $\psi(n)$ is discrete. It should not be confused with the method that uses a Gaussian function in the coordinate representation as the variational wave function, which is trivial. One reason that we compare Eqs. (2) and (3) is to demonstrate that this method is capable of giving the correct coefficients regardless of the radial basis function.

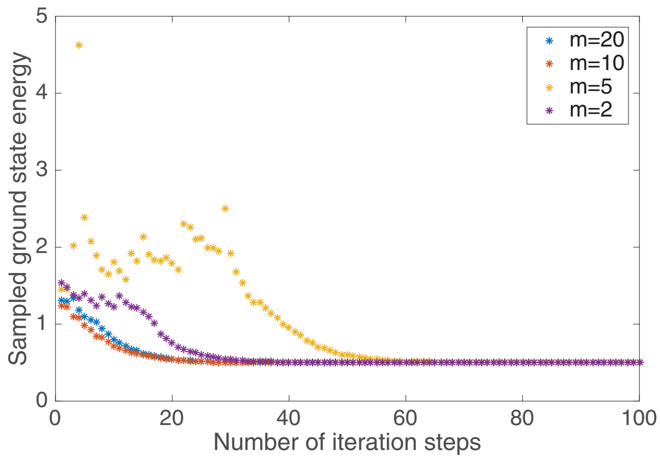

FIG. 3. Minimization of the ground state energy of H at $E = 0$, using Eq. (3) as the radial basis function. m is the number of hidden layers in the neural network.

Figure 4 illustrates the behavior of VMC under different electric field. In our simulation, a separate neural network is trained for each E. The theoretical value of the ground state energy e_g is $e_g = 0.5(1 - E^2)$. The VMC results converge at 0.375 ± 0.000, 0.000 ± 0.000, -1.446 ± 0.003 when $E = 0.5, 1.0, 2.0$, while the exact value is at 0.375, 0.000, −1.5, respectively. Notice that the error increase with E under certain nmax. In this section nmax = 20. Expectation value and errors in this article are calculated when optimization is saturated.

Notice that during the optimization process, the sampled ground state energy may have some spikes. The author believes that this phenomenon is a result of the stochastic nature of the optimization algorithm. Random fluctuations of the expectation value of the operator and the complicated structure of the energy function may lead to drastic changes in the ground state energy during the optimization process.

Figure 5 shows $\psi(n)$ as a function of n under different E. $\psi(n)$ is normalized and its value means the overlap between new ground state of H and the energy eigenstate $|n\rangle$ of H_o.

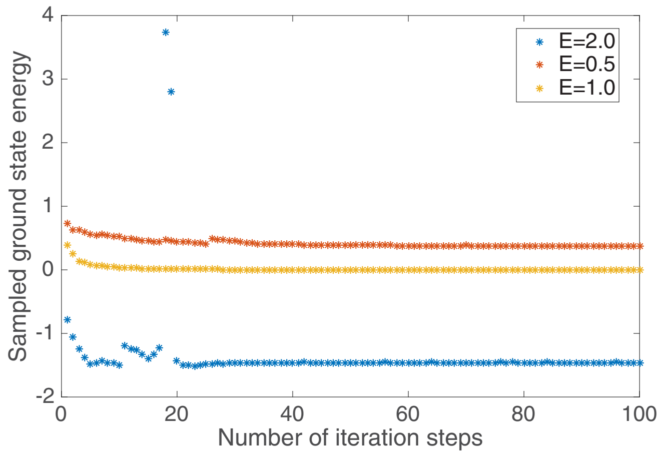

FIG. 4. Minimization of the ground state energy of H at $E = 0.5, 1.0, 2.0$, using Gaussian radial basis function.

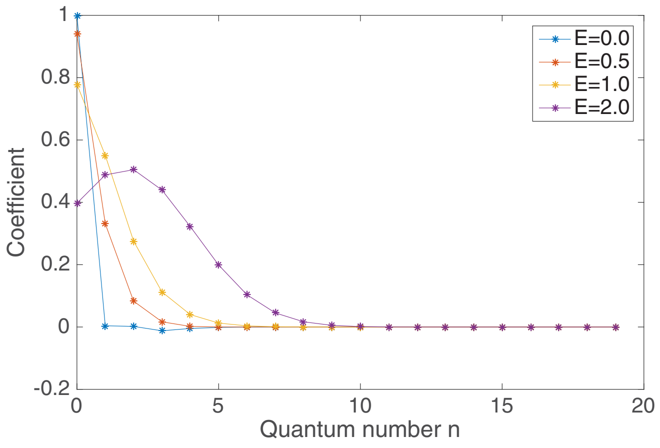

FIG. 5. $\psi(n)$ as a function of n at $E = 0.0, 0.5, 1.0, 2.0$, using Gaussian radial basis function. Circles represent theoretical values and asterisk represents the values with RBF network.

Theoretically, one can calculate that

$$\psi(n) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2^n n!}} \left(\frac{1}{\pi}\right)^{\frac{1}{2}} e^{-(x-E)^2/2} e^{-x^2/2} H_n(x) dx, \quad (21)$$

where $H_n(x)$ are the Hermite polynomials. Simplifying this expression, we will get

$$\psi(n) = \frac{1}{\sqrt{2^n n!}} E^n e^{-E^2/4}. \quad (22)$$

It can be seen that VMC values agree very well with the exact value when E is small. Errors begin to increase when E gets larger.

Based on these results, we claim that the radial basis neural network clearly captures the behavior of the 1D quantum harmonic oscillator.

C. Two-dimensional quantum harmonic oscillator in electric field

Similarly, we can consider a radial basis neural network with many input neurons. For example, with two input neurons, we can consider a two-dimensional (2D) quantum harmonic oscillator in an electric field.

Consider a Hamiltonian

$$\begin{aligned} H &= \frac{\hat{p}_x{}^2}{2} + \frac{\hat{x}^2}{2} + \frac{\hat{p}_y{}^2}{2} + \frac{\hat{y}^2}{2} + E_x\hat{x} + E_y\hat{y} \\ &= H_0 + E_x\hat{x} + E_y\hat{y}. \end{aligned} \quad (23)$$

It is easy to see that the eigenvalue of H_0 is 1.0. We will treat E_x and E_y as our parameters.

Our neural network wave function can be written as

$$|\psi\rangle = \sum_{n_x, n_y = 0}^{n_{\max}-1} \psi(n_x, n_y) |n_x, n_y\rangle. \quad (24)$$

We can use the same VMC procedure as the previous part to perform the calculation. The learning rate, in this case, is set at 0.2; our neural network has 10 hidden neurons and 2 input neurons. The algorithm used for this 2D example is similar to the 1D harmonic oscillator.

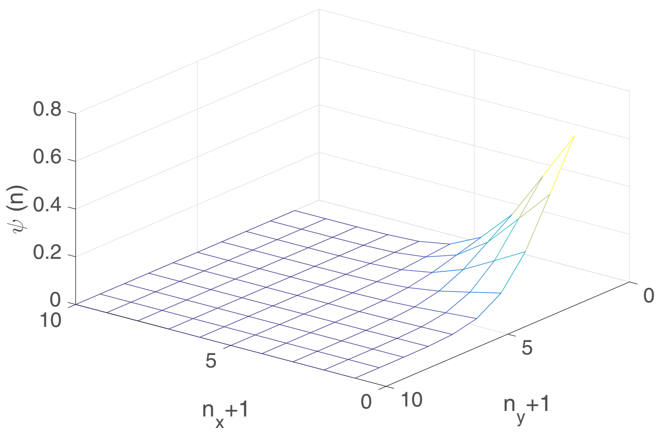

FIG. 6. $\psi(n_x, n_y)$ as a function of $n_x + 1$, $n_y + 1$ at $E_x = 1.0$, $E_y = 1.0$, using Gaussian radial basis function. In this figure, $\psi(n_x, n_y)$ is not normalized.

Figures 6 and 7 illustrate the behavior of the trained neural network at different electric fields. From the shape of the surface, we can see that a proper choice of nmax is important to the accuracy of this method. The reason is that, in this example, when E_x and E_y get larger, the bump in the function $\psi(n)$ will shift away from the origin. The states out of nmax are not considered, therefore, the accuracy will be affected if the overlaps out of nmax are large. In these figures, we choose nmax = 10 to illustrate the influence of nmax on the accuracy.

The exact value of $\psi(n_x, n_y)$ can be solved as

$$\psi(n_x, n_y) = \frac{1}{\sqrt{2_x^n n_x!}}\frac{1}{\sqrt{2_y^n n_y!}} E_x^{n_x} E_y^{n_y} e^{-E_x^2/4} e^{-E_y^2/4}. \quad (25)$$

Table I lists a sample of the relation between nmax and the VMC energy at $E_x = 4.0$, $E_y = 2.0$. We can see that in this example the accuracy of the results improve with nmax. Figure 8 shows $\psi(n_x, n_y)$ as a function of n_x and n_x under different $E = (1.0, 1.0)$. We can see that numerical results agree well with exact results.

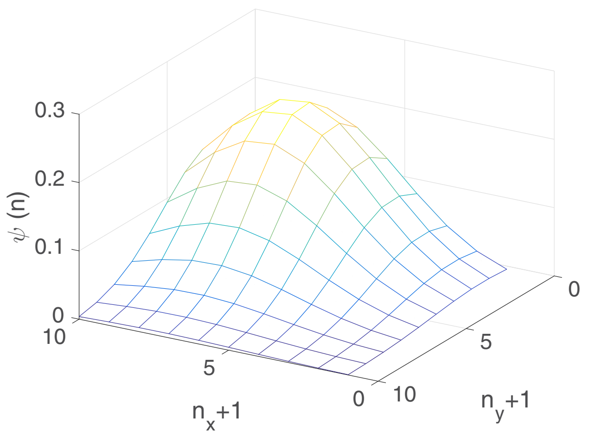

FIG. 7. $\psi(n_x, n_y)$ as a function of $n_x + 1$, $n_y + 1$ at $E_x = 4.0$, $E_y = 2.0$, using Gaussian radial basis function. In this figure, $\psi(n_x, n_y)$ is not normalized.

TABLE I. The relation between nmax and the VMC energy at $E_x = 4.0$, $E_y = 2.0$. VMC energy converges at -8.99571 ± 0.00627. Exact value is 9.

nmax	VMC energy
3	–6.28397
4	–7.80747
5	–8.02855
10	–8.71073
20	–8.90894
40	–8.99571

D. Particle in a box

Another example that is benchmarked is a particle in a box with perturbation. Consider the Hamiltonian

$$H = \frac{\hat{p}^2}{2} + V(x) + a\hat{x} = H_0 + a\hat{x}, \quad (26)$$

with $V(x) = 0$ when $0 < x < 1$ and $V(x) = \infty$ when x takes other values. $a\hat{x}$ is a linear potential defined on $0 < x < 1$ with a as a parameter.

In natural units, the ground state energy of H_0 is $\frac{\pi^2}{2} = 4.9348$. The first order perturbation theory correction for the ground state energy is $a/2$. The second order perturbation will give a correction of $-0.002194a^2$. A radial basis neural network VMC simulation can be similarly carried out. As always, we choose the basis to be the eigenbasis of H_0. 50 000 samples are used. Ten hidden neurons ($m = 10$) are chosen in our calculation. nmax is set at 20. The learning rates are set at 0.01. The matrix element in the local energy can be calculated as

$$\langle n_1|ax|n_2\rangle = a\frac{4[(-1)^{n_1+n_2} - 1]n_1 n_2}{(n_1 - n_2)^2(n_1 + n_2)^2\pi^2}, \quad (27)$$

when $n_1 \neq n_2$. And,

$$\langle n_1|ax|n_2\rangle = 0.5a, \quad (28)$$

when $n_1 = n_2$.

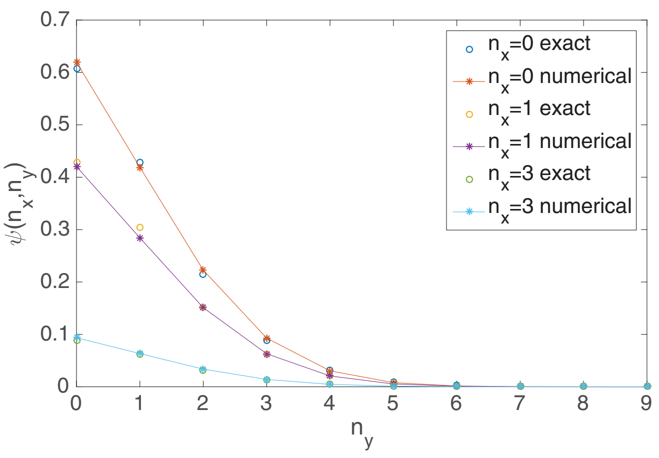

FIG. 8. $\psi(n_x, n_y)$ as a function of n_y at different n_x with $E_x = 1.0$, $E_y = 1.0$. Circles represent theoretical values and asterisk represents the values with RBF network. In this figure, $\psi(n_x, n_y)$ is normalized.

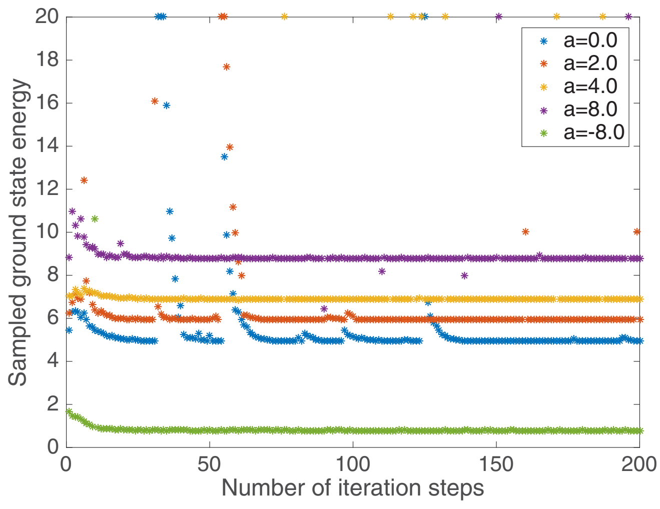

FIG. 9. Minimization of the ground state energy of H at $a = 0.0, 2.0, 4.0, 8.0, -8.0$.

In Fig. 9, the convergence VMC ground state at different parameters is illustrated. Intermediate points that have a value which is larger than 20 are set at 20 to maintain the scale of this graph. Notice that we get more spikes during the iteration when a is small. The heights of the spikes decrease if smaller learning rates are used.

Table II compares the result using an RBF network VMC, theoretical results up to second-order perturbation theory, and exact results. The exact ground state energy values are calculated using *Mathematica*. We can see that VMC performs much better than first-order perturbation theory and converge to the ground state energy that is very close to the theoretical ground state energy.

E. Neural network as a Hermitian matrix lowest eigenvalue solver

So far, the examples that are benchmarked can all be solved by perturbation theory. Can neural network VMC method have a wider application than the perturbation theory? In this part, we will illustrate the possibility of using an RBF network VMC method to solve for the smallest eigenvalue of a Hermitian matrix. This problem is nonperturbative and purely mathematical, and our result implies that neural network VMC can have much broader scope than perturbation method.

Consider an $n \times n$ Hermitian matrix H. The eigenvector that corresponds to the lowest energy is an n-dimensional vector. We can write this eigenvector as

$$\vec{x} = \sum_{i=1}^{n} \psi(i)\hat{i}, \tag{29}$$

and any vector in this finite vector space can be written in this form.

Define the objective function to be

$$E = \vec{x}^{*} H \vec{x}. \tag{30}$$

Then, the smallest value of E corresponds to the lowest eigenvalue of H. Our goal is to find a set of parameters in neural network ψ that minimize E.

We can convert the matrix multiplication in E into a discrete sum, which can be evaluated using the Metropolis algorithm. Instead of the energy eigenbasis, in this situation, we can choose our configuration space to be n points, where n is the dimension of vector $\vec{x}$, and the trial move would be from basis $\hat{i}$ to $\hat{i}'$. Therefore, we can use the same VMC technique to minimize E. Our previous examples can be essentially understood in this way since our Hamiltonians are truncated to a finite-dimensional matrix.

To give a concrete implementation of this idea, we consider a matrix

$$H(d)_{pq} = 1/p + 1/q. \tag{31}$$

Here, $H(d)$ is a $(d \times d)$-dimensional matrix. p, q are the labels for $H(d)$. The matrix element on the pth row and qth column equals $1/p + 1/q$.

We use the RBF network ansatz to calculate the lowest eigenvalue of $H(d)$. Hidden neuron numbers are set at 20. 50 000 samples are chosen. Iteration undergoes 300 steps and learning rate is 0.01. Table III shows the result of our VMC simulation.

Our optimized neural network also yields the eigenvector that corresponds to the lowest eigenvalue. The components can be acquired by plugging in i into $\psi(i)$. For example, when $n = 10$, VMC gives a eigenvector $\overrightarrow{V}$, which is (0.6851, 0.1174, −0.0711, −0.1646, −0.2200, −0.2562,−0.2813, −0.2994, −0.3127, −0.3226), while the exact vector $\overrightarrow{V_0}$ is (0.6807, 0.1194, −0.0677, −0.1613, −0.2174, −0.2548, −0.2816, −0.3016, −0.3172, −0.3297). The Euclidean norm of the error $d = |\overrightarrow{V} - \overrightarrow{V_0}| = 1.1 \times 10^{-2}$.

We also calculate the relation between the accuracy and m (the number of neurons in the hidden layer). For $d = 10$, the variational energy is −0.0811, −0.0811, −0.9943, −1.0002 for $m = 5, 10, 15, 20$, respectively.

Caveat. The learning rate depends on the number of hidden neurons, and it has to be set by trial and error. We also have to

TABLE II. Comparison between exact values, perturbation results, and numerical VMC energy at different a.

a	First order	Second order	VMC energy	Exact value
0.0	4.9348	4.9348	4.9348 ± 0.0001	4.93481
2.0	5.9348	5.9260	5.9260 ± 0.0001	5.92603
4.0	6.9348	6.8997	6.8998 ± 0.0001	6.89974
8.0	8.9348	8.7944	8.7960 ± 0.0003	8.79508
−8.0	0.9348	0.7944	0.7950 ± 0.0003	0.795078

TABLE III. VMC results of the lowest eigenvalue of $H(d)$.

d	Exact value	VMC result
2	−0.0811	−0.0811 ± 0.0000
3	−0.1874	−0.1873 ± 0.0002
5	−0.4219	−0.4220 ± 0.0008
10	−1.008	−1.008 ± 0.0008

point out that when $d > 10$, the VMC optimization procedure may converge slowly or fail to converge. The stability also depends on forms of H. For some large ill-conditioned matrices, it is expected that the random sampling process will not capture all the matrix elements and lead to inaccurate results.

IV. DISCUSSION

Is it possible to use an RBF network with continuous variables as the variational wave function? This is possible for some certain Hamiltonians. For example, we can use an RBF network with a Gaussian basis as the variation wave function for the ground state of a harmonic oscillator. Based on our test, although this ansatz works perfectly for the harmonic oscillator, the iteration may not converge to the correct ground state when applied to other models. This test is trivial for the harmonic oscillator since its ground state is intrinsically a Gaussian function. For wave functions with continuous variables, the Kato's cusp condition [24] poses strong constraints on the mathematical form of the wave function. A wave function that does not satisfy this condition will result in strong numerical instability in the VMC calculation.

How is this approach useful? This approach provides a way to find the ground state energy of a quantum system. Compared with traditional variational Monte Carlo simulation, this method does not require choosing a specific wave function from our intuition. Does this method depend on a specific basis? The example on the diagonalization of a Hermitian matrix illustrates that it does not depend on it as well, although a good basis may improve the accuracy and stability.

One advantage of ANN-based VMC is that the code is easy to modularize. When programming, we can write the modules for a neural network, Hamiltonian, and optimization separately. For the same Hamiltonian, we can also compare the representation power of different neural networks and different optimization methods. This greatly reduces programming difficulties and improves accuracy.

A potential issue with the neural network VMC method is that the optimization algorithm may fail to find the global minimum of the objective function. This is a common issue in machine-learning methods. We see that the stochastic reconfiguration may not work well enough that it could find the smallest eigenvalue of a matrix of arbitrarily large dimension. Therefore, finding a stable algorithm or stable neural network mathematical form for the VMC optimization should be a crucial task. If successful, the neural network VMC method may give numerical conclusions to many unsolved problems in quantum physics.

Based on the above points, one important research direction is to develop more efficient VMC optimization algorithms. Another interesting direction is to discuss the representation power of different neural networks since there are a variety of neural networks developed by the machine-learning community. For example, one interesting problem is the representation power of a continuous restricted Boltzmann machine [25]. With a Gaussian activation function, a continuous restricted Boltzmann machine has some similarities with the RBF network ansatz discussed in this paper. It is promising to provide more accurate results due to the elegant mathematical structure of the restricted Boltzmann machine.

V. CONCLUSION

In this article, RBF networks are used as the variational wave function for quantum systems, and VMC calculations are carried out. For the examples that are examined, the VMC results agree well with theoretical predictions. Furthermore, it is possible to use the VMC method to calculate the lowest eigenvalue of a Hermitian matrix.

ACKNOWLEDGMENTS

I acknowledge Dr. Y.-M. Lu for his helpful discussions and comments. I also want to thank the Ohio State University Physics Department for supporting my study. This work is supported by the startup funds of Dr. Y.-M. Lu at the Ohio State University.

[1] F. Hase, S. Valleau, E. Pyzer-Knapp, and A. Aspuru-Guzik, Machine learning exciton dynamics, Chem. Sci. **7**, 5139 (2016).
[2] M. Gastegger, J. Behler, and P. Marquetand, Machine learning molecular dynamics for the simulation of infrared spectra, Chem. Sci. **8**, 6924 (2017).
[3] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, Bypassing the kohn-sham equations with machine learning, Nat. Commun. **8**, 872 (2017).
[4] P. Raccuglia, K. C. Elbert, P. D. F. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist, Machine-learning-assisted materials discovery using failed experiments, Nature (London) **533**, 73 (2016).
[5] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).
[6] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. **5**, 115 (1943).
[7] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, 2006).
[8] F. Schwenker, H. A. Kestler, and G. Palm, Three learning phases for radial-basis-function networks, Neural Networks **14**, 439 (2001).
[9] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science **313**, 504 (2006).
[10] F. Scarselli and A. C. Tsoi, Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results, Neural Networks **11**, 15 (1998).
[11] J. Park and I. W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. **3**, 246 (1991).
[12] I.E. Lagaris, A. Likas, and D.I. Fotiadis, Artificial neural network methods in quantum mechanics, Comput. Phys. Commun. **104**, 1 (1997).

[13] A. J. da Silva, T. B. Ludermir, and W. R. de Oliveira, Quantum perceptron over a field and neural network architecture selection in a quantum computer, Neural Networks **76**, 55 (2016).

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

[15] H. Saito, Solving the bose-hubbard model with machine learning, J. Phys. Soc. Jpn. **86**, 093001 (2017).

[16] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, Sci. Rep. **7**, 8823 (2017).

[17] T. Poggio and F. Girosi, Networks for approximation and learning, Proc. IEEE **78**, 1481 (1990).

[18] W. L. McMillan, Ground state of liquid He^4, Phys. Rev. **138**, A442 (1965).

[19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087 (1953).

[20] C. J. Umrigar, J. Toulouse, C. Filippi, S. Sorella, and R. G. Hennig, Alleviation of the Fermion-Sign Problem by Optimization of Many-Body Wave Functions, Phys. Rev. Lett. **98**, 110201 (2007).

[21] S. Sorella, Wave function optimization in the variational monte carlo method, Phys. Rev. B **71**, 241103 (2005).

[22] B. Rubenstein, Introduction to the variational Monte Carlo method in quantum chemistry and physics, in *Variational Methods in Molecular Modeling*, edited by Jianzhong Wu (Springer, Singapore, 2017), pp. 285–313.

[23] G. Guennebaud, B. Jacob *et al.*, Eigen v3, http://eigen.tuxfamily.org

[24] T. Kato, On the eigenfunctions of many-particle systems in quantum mechanics, Commun. Pure Appl. Math. **10**, 151 (1957).

[25] H. Chen and A. Murray, A continuous restricted boltzmann machine with a hardware- amenable learning algorithm, in *Artificial Neural Networks: ICANN 2002*, edited by José R. Dorronsoro (Springer, Berlin, 2002), pp. 358–363.