CHINESE PHYSICAL SOCIETY | IOP Publishing
CPB
Chinese Physics B
http://cpb.iphy.ac.cn https://iopscience.iop.org/journal/1674-1056

Digraph states and their neural network representations

Ying Yang(杨莹) and Huaixin Cao(曹怀信)

Citation: Chin. Phys. B, 2022, 31 (**6**): 060303. DOI: 10.1088/1674-1056/ac401d
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

Digraph states and their neural network representations

Ying Yang(杨莹)[1] and Huaixin Cao(曹怀信)[2,†]

[1] *School of Mathematics and Information Technology, Yuncheng University, Yuncheng* 044000*, China*
[2] *School of Mathematics and Statistics, Shaanxi Normal University, Xi'an* 710119*, China*

(Received 11 August 2021; revised manuscript received 1 December 2021; accepted manuscript online 5 December 2021)

With the rapid development of machine learning, artificial neural networks provide a powerful tool to represent or approximate many-body quantum states. It was proved that every graph state can be generated by a neural network. Here, we introduce digraph states and explore their neural network representations (NNRs). Based on some discussions about digraph states and neural network quantum states (NNQSs), we construct explicitly an NNR for any digraph state, implying every digraph state is an NNQS. The obtained results will provide a theoretical foundation for solving the quantum many-body problem with machine learning method whenever the wave-function is known as an unknown digraph state or it can be approximated by digraph states.

Keywords: digraph state, neural network, quantum state, representation

PACS: 03.67.–a, 03.65.–w, 03.65.Aa, 03.65.Wj **DOI:** 10.1088/1674-1056/ac401d

1. Introduction

In quantum physics, fully understanding and characterizing a complex system with a large number of interacting particles[1] is an extremely challenging problem. Solutions within the standard framework of quantum mechanics generally require the knowledge of the full quantum many-body wave function. Thus, the problem becomes how to solve the many-body Schrödinger equation[2–4] of the system with a large dimension. This is just the so-called quantum many-body problem (QMBP)[5–7] in quantum physics, which becomes a hot topic in high energy physics and condensed matter physics. When the dimension of the Hilbert space describing the system is exponentially large, it becomes a big challenge to solve the QMBP even with the most powerful computers.

To overcome this exponential difficulty and solve the QMBP, many methods have been used, including tensor network method (TNM)[8–10] and quantum Monte Carlo simulation (QMCS).[11] However, the TNM has difficulty to deal with high dimensional systems[12] or systems with massive entanglement.[13] The QMCS suffers from the sign problem.[14] Thus, some new methods are necessary for finding QMBPs.

The approximation capabilities of artificial neural networks (ANNWs) have been investigated by many authors, including Cybenko,[15] Funahashi,[16] Hornik,[17,18] Kolmogorov,[19] Roux.[20] It is known that ANNWs can be used in many fields, including representing complex correlations in multiple-variable functions or probability distributions,[20] studying artificial intelligence through the popularity of deep learning methods,[21] developing an artificial neural network potential for Au clusters,[22] and so on.[23–27]

Undoubtedly, the interaction between machine learning and quantum physics will benefit both fields.[28,29] For instance, in light of the idea of machine learning, Carleo and Troyer[30] found an interesting connection between the variational approach in the QMBP and learning methods based on neural network representations. They used a restricted Boltzmann machine (RBM) to describe the many-body wave-function and obtained an efficient variational representation by optimizing those variational parameters with powerful learning methods. Chen *et al.*[31] discussed the general and constructive connection between the RBM and tensor network states (TNS). This equivalence sets up a bridge between the field of deep learning and quantum physics, allowing one to use the well-established entanglement theory of TNS to quantify the expressive power of RBM. Robeva *et al.*[32] showed the duality between tensor networks and undirected graphical models with discrete variables. Clark[33] used the framework of tensor networks to unify neural-network quantum states with the broader class of correlator product states. Huang *et al.*[34] proved that any (local) tensor network state has a (local) neural network representation. Lei *et al.*[35] proposed to utilize artificial neural network to determine the PT-phase-transition points for non-Hermitian PT-symmetric systems with short-range potentials. Yin *et al.*[36] improved accuracy of estimating two-qubit states with hedged maximum likelihood. Yang *et al.*[37] researched approximation of unknown ground state of a given Hamiltonian with neural network quantum states. Numerical evidences suggest that an RBM optimized by the reinforcement learning method can provide a good solution to several QMBPs.[38–46] However, the obtained solutions are approximate, instead of exact ones. To find exact solution of QMBP with an ANNW, the authors of Ref. [47] introduced

†Corresponding author. E-mail: caohx@snnu.edu.cn

neural network quantum states (NNQSs) with general input observables from the mathematical point of view, and found some N-qubit states that can be represented by a normalized NNQS, such as all separable pure states, Bell states and GHZ states.

Graph states are a special class of pure multi-party quantum states, and they have extensive applications. One-way quantum computation takes graph states as resources[48] and all code words in the standard quantum error correcting codes correspond to graph states.[49] Graph states have been produced in optical lattices[50] and the basic elements of one-way quantum computing have been demonstrated experimentally.[51] In Ref. [47], we determined the necessary and sufficient conditions for the representability of a general graph state using normalized NNQS for a given number of hidden neurons. Gao *et al.*[52] proved theoretically that every graph state can be represented by an RBM with $\{0,1\}$-input and obtained the RBMRs of every graph state.

Spectra of quantum graphs display in general universal statistics characteristic for ensembles of random unitary matrices observed by Kottos and Smilansky in Refs. [53,54]. The quantization scheme of Kottos and Smilansky has been generalized to directed graphs (digraphs).[55–57] A digraph provide an intermediate step that gives explicit relationships between the process variables, human errors, and equipment failure events, from which the fault tree can be constructed.[58] It has many applications, e.g., fault-tree synthesis,[58] fault propagation model[59] and design of sensor network.[60]

In this paper, we aim to define digraph states (directed graph) and construct explicitly the neural network representations (NNRs) of digraph states. In Section 2, some notations and conclusions on NNQS with general input observables are recalled and some related properties are proved. In Section 3, digraph states are proposed, and some properties are explored. In Section 4, the NNRs of digraph states are constructed.

2. Neural network quantum states

To start with, let us first briefly introduce some notations in the neural network architecture oriented from Ref. [30] and mathematically formulated in Ref. [47].

Let $Q_1, Q_2, \ldots, Q_N$ be N quantum systems with state spaces $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_N$ of dimensions $d_1, d_2, \ldots, d_N$, respectively. We consider the composite system Q of $Q_1, Q_2, \ldots, Q_N$ with state space $\mathcal{H} := \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_N$.

Let $S_1, S_2, \ldots, S_N$ be non-degenerate observables of systems $Q_1, Q_2, \ldots, Q_N$, respectively. Then $S = S_1 \otimes S_2 \otimes \cdots \otimes S_N$ is an observable of the composite system Q. Use $\{|\psi_{k_j}\rangle\}_{k_j=0}^{d_j-1}$ to denote the eigenbasis of S_j corresponding to eigenvalues $\{\lambda_{k_j}\}_{k_j=0}^{d_j-1}$. Thus,

$$S_j|\psi_{k_j}\rangle = \lambda_{k_j}|\psi_{k_j}\rangle (k_j = 0, 1, \ldots, d_j - 1). \tag{1}$$

It is easy to check that the eigenvalues and corresponding eigenbases of $S = S_1 \otimes S_2 \otimes \cdots \otimes S_N$ are

$$\begin{aligned} &\lambda_{k_1}\lambda_{k_2}\ldots\lambda_{k_N}, \\ &|\psi_{k_1}\rangle \otimes |\psi_{k_2}\rangle \otimes \cdots \otimes |\psi_{k_N}\rangle \ \ (k_j = 0, 1, \ldots, d_j - 1), \end{aligned} \tag{2}$$

respectively. Put

$$V(S) = \left\{ \Lambda_{k_1k_2\ldots k_N} \equiv \left(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}\right)^{\mathrm{T}} : k_j = 0, 1, \ldots, d_j - 1 \right\},$$

called an input space. For parameters

$$\begin{aligned} &a = (a_1, a_2, \ldots, a_N)^{\mathrm{T}} \in \mathbb{C}^N, \\ &b = (b_1, b_2, \ldots, b_M)^{\mathrm{T}} \in \mathbb{C}^M, \quad W = [W_{ij}] \in \mathbb{C}^{M\times N}, \end{aligned}$$

write $\Omega = (a, b, W)$ and put

$$\begin{aligned} &\Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}) \\ &= \sum_{h_i=\pm1} \exp\left(\sum_{j=1}^{N} a_j\lambda_{k_j} + \sum_{i=1}^{M} b_ih_i + \sum_{i=1}^{M}\sum_{j=1}^{N} W_{ij}h_i\lambda_{k_j} \right). \end{aligned} \tag{3}$$

Then we obtain a complex-valued function $\Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N})$ of the input variable $\Lambda_{k_1k_2\ldots k_N}$. We call it a neural network quantum wave-function (NNQWF).[47] It may be identically zero. For example, when $b_i = \frac{\pi \mathrm{i}}{2}$, $W_{ij} = 0$ for $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, we have $\Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}) \equiv 0$ for all $\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}$. In what follows, we assume that this is not the case, i.e., assume that $\Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}) \neq 0$ for some input variable $\Lambda_{k_1k_2\ldots k_N}$. Then we define

$$\begin{aligned} |\Psi_{S,\Omega}\rangle = &\sum_{\Lambda_{k_1k_2\ldots k_N}\in V(S)} \Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N})|\psi_{k_1}\rangle \\ &\otimes|\psi_{k_2}\rangle \otimes \cdots \otimes |\psi_{k_N}\rangle, \end{aligned} \tag{4}$$

which is a nonzero vector (not necessarily normalized) of the Hilbert space $\mathcal{H}$. We call it a neural network quantum state (NNQS) induced by the parameter $\Omega = (a, b, W)$ and the input observable $S = S_1 \otimes S_2 \otimes \cdots \otimes S_N$ (Fig. 1).[47]

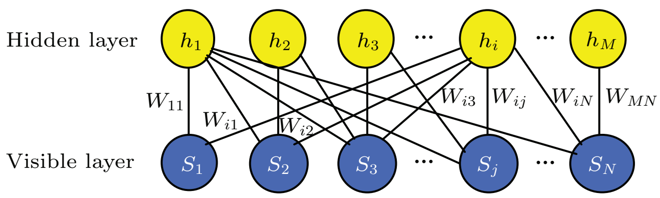


Fig. 1. Artificial neural network encoding an NNQS. It is a restricted Boltzmann machine architecture that features a set of N visible artificial neurons (blue disks) and a set of M hidden neurons (yellow disks). For each value $\Lambda_{k_1k_2\ldots k_N}$ of the input observable S, the neural network computes the value of the $\Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N})$.

The NNQWF can be reduced to

$$\begin{aligned} &\Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}) \\ &= \prod_{j=1}^{N} \mathrm{e}^{a_j\lambda_{k_j}} \cdot \prod_{i=1}^{M} 2\cosh\left(b_i + \sum_{j=1}^{N} W_{ij}\lambda_{k_j} \right). \end{aligned} \tag{5}$$

There is a special class of NNQSs:

When $S=\sigma_1^z\otimes\sigma_2^z\otimes\cdots\otimes\sigma_N^z$, we have

$$\lambda_{k_j}=\begin{cases}1, & k_j=0,\\ -1, & k_j=1,\end{cases}$$
$$|\psi_{k_j}\rangle=\begin{cases}|0\rangle, & k_j=0,\\ |1\rangle, & k_j=1,\end{cases}\quad (1\le j\le N) \tag{6}$$

and $V(S)=\{1,-1\}^N$.

In this case, the NNQS (4) becomes

$$|\Psi_{S,\Omega}\rangle = \sum_{\Lambda_{k_1k_2\ldots k_N}\in\{1,-1\}^N}\Psi_{S,\Omega}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})|\psi_{k_1}\rangle \otimes|\psi_{k_2}\rangle\otimes\cdots\otimes|\psi_{k_N}\rangle. \tag{7}$$

This leads to the NNQS induced in Ref. [30] and discussed in Refs. [47,61]. We call such an NNQS a spin-z NNQS.[47]

From the definition of NNQWF, we can easily obtain the following results.

Proposition 1 If a hidden layer neuron h_{M+1} is added into an RBM with NNQWF $\Psi_{S,\Omega}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})$, then the NNQWF $\Psi_{S,\Omega'}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})$ of the resulted network reads

$$\begin{aligned}&\Psi_{S,\Omega'}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})\\ &=\Psi_{S,\Omega}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})\cdot\Psi_{S,\tilde{\Omega}}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N}),\end{aligned}$$

where

$$\begin{aligned}&\Psi_{S,\tilde{\Omega}}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})\\ &=\sum_{h_{M+1}=\pm1}\exp\left(\sum_{j=1}^N\tilde{a}_j\lambda_{k_j}+\tilde{b}h_{M+1}+\sum_{j=1}^N h_{M+1}W_{(M+1)j}\lambda_{k_j}\right),\end{aligned}$$
$$\begin{aligned}\Omega &= (a,b,W),\quad \Omega'=(a',b',W'),\quad \tilde{\Omega}=(\tilde{a},\tilde{b},\tilde{W}),\\ a' &= a+\tilde{a},\ \tilde{b}=b_{M+1},\\ \tilde{W} &= (W_{(M+1)1},W_{(M+1)2},\ldots,W_{(M+1)N}),\\ b' &= \begin{pmatrix} b\\ \tilde{b}\end{pmatrix}\in\mathbb{C}^{M+1},\quad W'=\begin{pmatrix} W\\ \tilde{W}\end{pmatrix}\in\mathbb{C}^{(M+1)\times N}.\end{aligned}$$

This result can be illustrated by Fig. 2.

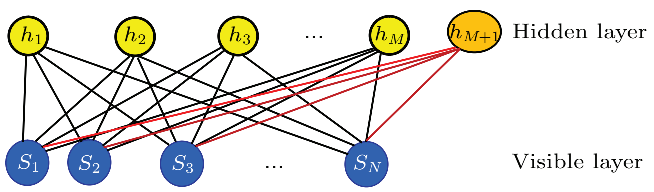


Fig. 2. The resulted network by adding a hidden layer neuron h_{M+1} into an network with visible layer $S_1,S_2,\ldots,S_N$ and hidden layer $h_1,h_2,\ldots,h_M$.

Proposition 2 Suppose that $\Psi_{S,\Omega'}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})$ and $\Psi_{S,\Omega''}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})$ are two spin-z NNQWFs with the same input observable $S=\sigma_1^z\otimes\sigma_2^z\otimes\cdots\otimes\sigma_N^z$, and individual parameters $\Omega'=(a',b',W')$, $\Omega''=(a'',b'',W'')$, respectively. Then

$$\begin{aligned}&\Psi_{S,\Omega'}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})\cdot\Psi_{S,\Omega''}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})\\ &=\Psi_{S,\Omega}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N}),\end{aligned}$$

where

$$\begin{aligned}&\Omega=(a,b,W),\quad a=a'+a'',\\ &b=\begin{pmatrix} b'\\ b''\end{pmatrix}\in\mathbb{C}^{M'+M''},\\ &W=\begin{pmatrix} W'\\ W''\end{pmatrix}\in\mathbb{C}^{(M'+M'')\times N}.\end{aligned}$$

3. Digraph states

In this section, we aim to introduce digraph states. To do this, let us start by introducing the definition of digraph. A digraph (or a directed graph)[57,62] is a pair $\overrightarrow{G}=(V,\overrightarrow{E})$ consisting of a set $V=\{1,2,\ldots,N\}$ and a nonempty subset $\overrightarrow{E}$ of $V\times V$. The elements of V and $\overrightarrow{E}$ are called vertices and edges of $\overrightarrow{G}$, respectively. When $e=(i_1,i_2)\in\overrightarrow{E}$, we say that e is an edge of $\overrightarrow{G}$ from the vertex i_1 to the vertex i_2.

Given a digraph $\overrightarrow{G}=(V,\overrightarrow{E})$, we call $\overleftarrow{G}=(V,\overleftarrow{E})$ the inverse graph of $\overrightarrow{G}=(V,\overrightarrow{E})$, where

$$\overleftarrow{E}=\{(j,i)|(i,j)\in\overrightarrow{E}\}.$$

For example, when

$$\overrightarrow{E}=\{(1,2),(2,1),(1,3),(4,3),(3,5),(5,3),(4,5)\},$$

we have

$$\overleftarrow{E}=\{(2,1),(1,2),(3,1),(3,4),(5,3),(3,5),(5,4)\},$$

see Figs. 3 and 4.

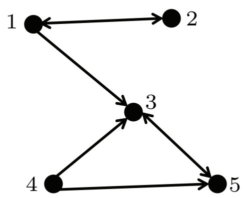


Fig. 3. A digraph $\overrightarrow{G}$.

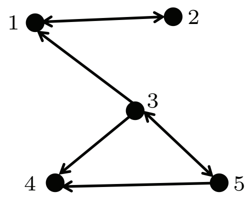


Fig. 4. The inverse graph of a digraph $\overrightarrow{G}$.

Given a digraph $\overrightarrow{G}=(V,\overrightarrow{E})$, for each edge $(i,j)\in\overrightarrow{E}$ define an operator on the N-qubit system $(\mathbb{C}^2)^{\otimes N}$:

$$U^{(i,j)}=\begin{cases}P_{Z,+}^{(i)}+P_{Z,-}^{(i)}Z^{(j)}, & i\le j,\\ P_{Z,-}^{(i)}+P_{Z,+}^{(i)}Z^{(j)}, & i>j,\end{cases}$$

where

$$P_{Z,\pm}^{(i)}=\frac{I\pm Z^{(i)}}{2},\quad |+\rangle=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle),$$

I is the $2^N \times 2^N$ unit matrix, and $Z^{(i)}$ denotes the Pauli σ_z gate acting on the i-subsystem, that is, $Z^{(i)} = \bigotimes_{k=1}^{N} T_k$ with $T_i = Z = \sigma_z$ and $T_k = I_2 (k \neq i)$ in which I_2 is the 2×2 unit matrix.

It is easy to check that $U^{(i,j)}$ is a Hermitian operator for every $(i,j) \in \overrightarrow{E}$ and has the following properties.

(1) When $i = j$, it holds that

$$U^{(i,i)} = P_{Z,+}^{(i)} + P_{Z,-}^{(i)} Z^{(i)} = Z^{(i)},$$

thus

$$U^{(i,i)}|k_1k_2\ldots k_N\rangle = (-1)^{k_i}|k_1k_2\ldots k_N\rangle, \tag{8}$$

for all $k_1, k_2, \ldots, k_N = 0, 1$.

(2) When $i < j$, it holds that

$$U^{(i,j)}|k_1k_2\ldots k_N\rangle = \begin{cases} -|k_1k_2\ldots k_N\rangle, & k_i = k_j = 1, \\ |k_1k_2\ldots k_N\rangle, & \text{otherwise}, \end{cases}$$

for all $k_1, k_2, \ldots, k_N = 0, 1$. Thus,

$$U^{(i,j)}|k_1k_2\ldots k_N\rangle = (-1)^{k_ik_j}|k_1k_2\ldots k_N\rangle, \tag{9}$$

for all $k_1, k_2, \ldots, k_N = 0, 1$.

(3) When $i > j$, it holds that

$$U^{(i,j)}|k_1k_2\ldots k_N\rangle = \begin{cases} -|k_1k_2\ldots k_N\rangle, & k_j = 1, k_i = 0; \\ |k_1k_2\ldots k_N\rangle, & \text{otherwise}, \end{cases}$$

for all $k_1, k_2, \ldots, k_N = 0, 1$. Thus,

$$U^{(i,j)}|k_1k_2\ldots k_N\rangle = (-1)^{(k_i+1)k_j}|k_1k_2\ldots k_N\rangle, \tag{10}$$

for all $k_1, k_2, \ldots, k_N = 0, 1$.

With these properties, we have the following proposition.

Proposition 3 If $i \neq j$, then

$$U^{(j,i)} = Z^{(\min\{i,j\})}U^{(i,j)} = U^{(i,j)}Z^{(\min\{i,j\})}, \tag{11}$$

$$U^{(i,j)}U^{(i,j)} = U^{(j,i)}U^{(j,i)} = I, \tag{12}$$

$$U^{(j,i)}U^{(i,j)} = U^{(i,j)}U^{(j,i)} = Z^{(\min\{i,j\})}. \tag{13}$$

Proof When $i \neq j$, without loss of generality, we assume $i < j$. From Eqs. (9)–(10) we obtain

$$U^{(i,j)}|k_1k_2\ldots k_N\rangle = (-1)^{k_ik_j}|k_1k_2\ldots k_N\rangle,$$

$$U^{(j,i)}|k_1k_2\ldots k_N\rangle = (-1)^{(k_j+1)k_i}|k_1k_2\ldots k_N\rangle$$

for all $k_1, k_2, \ldots, k_N = 0, 1$. Hence,

$$\begin{aligned} U^{(j,i)}|k_1k_2\ldots k_N\rangle &= (-1)^{k_i}U^{(i,j)}|k_1k_2\ldots k_N\rangle \\ &= Z^{(i)}U^{(i,j)}|k_1k_2\ldots k_N\rangle \end{aligned}$$

or

$$\begin{aligned} U^{(j,i)}|k_1k_2\ldots k_N\rangle &= (-1)^{k_i}U^{(i,j)}|k_1k_2\ldots k_N\rangle \\ &= U^{(i,j)}Z^{(i)}|k_1k_2\ldots k_N\rangle \end{aligned}$$

for all $k_1, k_2, \ldots, k_N = 0, 1$. Therefore,

$$U^{(j,i)} = Z^{(i)}U^{(i,j)} = U^{(i,j)}Z^{(i)}.$$

Thus, when $i \neq j$, we have

$$U^{(j,i)} = Z^{(\min\{i,j\})}U^{(i,j)} = U^{(i,j)}Z^{(\min\{i,j\})}.$$

Equations (9)–(10) easily yield

$$U^{(i,j)}U^{(i,j)} = U^{(j,i)}U^{(j,i)} = I.$$

Multiplying by $U^{(i,j)}$ on both sides of Eq. (11), we obtain

$$U^{(j,i)}U^{(i,j)} = U^{(i,j)}U^{(j,i)} = Z^{(\min\{i,j\})}.$$

We can see from Eq. (12) that $U^{(i,j)}$ is a unitary operator for every $(i,j) \in \overrightarrow{E}$. This enables us to define an N-qubit pure state

$$|\overrightarrow{G}\rangle = \left(\prod_{(i,j)\in\overrightarrow{E}} U^{(i,j)}\right)\underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N}, \tag{14}$$

called the digraph state corresponding to the digraph $\overrightarrow{G} = (V, \overrightarrow{E})$.

Proposition 4 The relationship between $|\overleftarrow{G}\rangle$ and $|\overrightarrow{G}\rangle$ is written as

$$|\overleftarrow{G}\rangle = \left(\prod_{(j,i)\in\overrightarrow{E},i\neq j} Z^{(\min\{i,j\})}\right)|\overrightarrow{G}\rangle.$$

Proof Using Eqs. (11) and (14), we have

$$\begin{aligned} |\overleftarrow{G}\rangle &= \prod_{(i,j)\in\overleftarrow{E}} U^{(i,j)}\underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N} = \left(\prod_{(i,i)\in\overleftarrow{E}} U^{(i,i)}\right)\left(\prod_{(i,j)\in\overleftarrow{E},i\neq j} U^{(i,j)}\right)\underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N} \\ &= \left(\prod_{(i,i)\in\overrightarrow{E}} U^{(i,i)}\right)\left(\prod_{(j,i)\in\overrightarrow{E},i\neq j} U^{(j,i)}Z^{(\min\{i,j\})}\right)\underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N} \\ &= \left(\prod_{(j,i)\in\overrightarrow{E},i\neq j} Z^{(\min\{i,j\})}\right)\left(\prod_{(i,i)\in\overrightarrow{E}} U^{(i,i)}\right)\left(\prod_{(j,i)\in\overrightarrow{E},i\neq j} U^{(j,i)}\right)\underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N} \end{aligned}$$

$$= \left(\prod_{(j,i)\in\overrightarrow{E},i\neq j} Z^{(\min\{i,j\})}\right)\left(\prod_{(i,j)\in\overrightarrow{E}} U^{(i,j)}\right)\underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N} = \left(\prod_{(j,i)\in\overrightarrow{E},i\neq j} Z^{(\min\{i,j\})}\right)|\overrightarrow{G}\rangle.$$

Next, we reduce the expression (14) of digraph state by the next procedure.

Let

$$\begin{aligned}
E_0 &= \{(i,j)|(i,j)\in\overrightarrow{E}, i=j\},\\
E_1 &= \{(i,j)|(i,j)\in\overrightarrow{E}, i<j\},\\
E_2 &= \{(i,j)|(i,j)\in\overrightarrow{E}, i>j\},\\
E_3 &= \{(i,j)|(i,j)\in\overrightarrow{E}, (j,i)\in\overrightarrow{E}, i\neq j\}.
\end{aligned}$$

Since

$$\begin{aligned}
|+\rangle^{\otimes N} &= \underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N}\\
&= \frac{1}{(\sqrt{2})^N}\sum_{k_1,k_2,\dots,k_N=0,1}|k_1k_2\dots k_N\rangle,
\end{aligned}$$

from Eqs. (8)–(10) and Eq. (13) we can see

$$\begin{aligned}
|\overrightarrow{G}\rangle &= \prod_{(i,j)\in\overrightarrow{E}} U^{(i,j)}\underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N}\\
&= \sum_{k_1,\dots,k_N=0,1}\frac{1}{(\sqrt{2})^N}\prod_{(i,j)\in\overrightarrow{E}} U^{(i,j)}|k_1k_2\dots k_N\rangle\\
&= \sum_{k_1,\dots,k_N=0,1}\frac{1}{(\sqrt{2})^N}\left(\prod_{(i,j)\in E_2} U^{(i,j)}\right)\left(\prod_{(i,j)\in E_1} U^{(i,j)}\right)\left(\prod_{(i,j)\in E_0} U^{(i,j)}\right)|k_1k_2\dots k_N\rangle\\
&= \sum_{k_1,\dots,k_N=0,1}\frac{1}{(\sqrt{2})^N}\left(\prod_{(i,j)\in E_2\setminus E_3} U^{(i,j)}\right)\left(\prod_{(i,j)\in E_2\cap E_3} U^{(i,j)}\right)\left(\prod_{(i,j)\in E_1\setminus E_3} U^{(i,j)}\right)\\
&\quad\times\left(\prod_{(i,j)\in E_1\cap E_3} U^{(i,j)}\right)\left(\prod_{(i,i)\in E_0} U^{(i,i)}\right)|k_1k_2\dots k_N\rangle\\
&= \sum_{k_1,\dots,k_N=0,1}\frac{1}{(\sqrt{2})^N}\left(\prod_{(i,j)\in E_2\setminus E_3} U^{(i,j)}\right)\left(\prod_{(i,j)\in E_1\setminus E_3} U^{(i,j)}\right)\left(\prod_{(i,i)\in E_0} U^{(i,i)}\right)\left(\prod_{(i,j)\in E_2\cap E_3} Z^{(j)}\right)|k_1k_2\dots k_N\rangle\\
&= \sum_{k_1,\dots,k_N=0,1}\frac{1}{(\sqrt{2})^N}\left(\prod_{(i,j)\in E_2\setminus E_3}(-1)^{(k_i+1)k_j}\right)\left(\prod_{(i,j)\in E_1\setminus E_3}(-1)^{k_ik_j}\right)\left(\prod_{(i,i)\in E_0}(-1)^{k_i}\right)\left(\prod_{(i,j)\in E_2\cap E_3}(-1)^{k_j}\right)|k_1k_2\dots k_N\rangle\\
&= \sum_{k_1,\dots,k_N=0,1}\frac{1}{(\sqrt{2})^N}\left(\prod_{(i,j)\in E_2\setminus E_3}(-1)^{k_ik_j}\right)\left(\prod_{(i,j)\in E_1\setminus E_3}(-1)^{k_ik_j}\right)\left(\prod_{(i,i)\in E_0}(-1)^{k_i}\right)\left(\prod_{(i,j)\in E_2}(-1)^{k_j}\right)|k_1k_2\dots k_N\rangle.
\end{aligned}$$

Note that

$$\begin{aligned}
&(-1)^{k_ik_j} = (-1)^{\frac{\left(1-\lambda_{k_i}\right)\left(1-\lambda_{k_j}\right)}{4}},\\
&\forall(i,j)\in(E_2\setminus E_3)\cup(E_1\setminus E_3);\\
&(-1)^{k_i} = (-1)^{\frac{\left(1-\lambda_{k_i}\right)}{2}},\ \forall(i,i)\in E_0;\\
&(-1)^{k_j} = (-1)^{\frac{\left(1-\lambda_{k_j}\right)}{2}},\ \forall(i,j)\in E_2,
\end{aligned}$$

we obtain

$$\begin{aligned}
|\overrightarrow{G}\rangle = &\sum_{\Lambda_{k_1k_2\dots k_N}\in\{1,-1\}^N}\Psi_{\overrightarrow{G}}(\lambda_{k_1},\lambda_{k_2},\dots,\lambda_{k_N})\\
&\cdot|\psi_{k_1}\rangle\otimes|\psi_{k_2}\rangle\otimes\cdots\otimes|\psi_{k_N}\rangle,
\end{aligned}\tag{15}$$

where

$$\begin{aligned}
&\Psi_{\overrightarrow{G}}(\lambda_{k_1},\dots,\lambda_{k_N})\\
&= \frac{1}{(\sqrt{2})^N}\left(\prod_{(i,j)\in E_2\setminus E_3}(-1)^{\frac{\left(1-\lambda_{k_i}\right)\left(1-\lambda_{k_j}\right)}{4}}\right)\\
&\quad\times\left(\prod_{(i,j)\in E_1\setminus E_3}(-1)^{\frac{\left(1-\lambda_{k_i}\right)\left(1-\lambda_{k_j}\right)}{4}}\right)\\
&\quad\times\left(\prod_{(i,i)\in E_0}(-1)^{\frac{1-\lambda_{k_i}}{2}}\right)\left(\prod_{(i,j)\in E_2}(-1)^{\frac{1-\lambda_{k_j}}{2}}\right),
\end{aligned}\tag{16}$$

and $\lambda_{k_1},\dots,\lambda_{k_N},|\psi_{k_1}\rangle,\dots,|\psi_{k_N}\rangle$ are shown in Eq. (6). We see that the simplified expression (15) is simpler and easier to use. Given a digraph, we can use this expression to obtain a digraph state associated to it very quickly.

For example, the digraph state $|\overrightarrow{C}_3\rangle$ given by the digraph

$\overrightarrow{C}_3$ (Fig. 5) is

$$
\begin{aligned}
|\overrightarrow{C}_3\rangle &= \frac{1}{(\sqrt{2})^3} \sum_{\Lambda_{k_1 k_2 k_3}\in\{1,-1\}^3} \\
&\quad \times \prod_{(i,j)\in E_1} (-1)^{\frac{\left(1-\lambda_{k_i}\right)\left(1-\lambda_{k_j}\right)}{4}} \cdot |\psi_{k_1}\psi_{k_2}\psi_{k_3}\rangle \\
&= \frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle-|011\rangle \\
&\quad +|100\rangle+|101\rangle-|110\rangle+|111\rangle),
\end{aligned}
$$

and $|\overleftarrow{C}_3\rangle$ corresponding to Fig. 6 is

$$
\begin{aligned}
|\overleftarrow{C}_3\rangle &= \frac{1}{(\sqrt{2})^3} \sum_{\Lambda_{k_1 k_2 k_3}\in\{1,-1\}^3} \prod_{(i,j)\in E_2} (-1)^{\frac{\left(1-\lambda_{k_i}\right)\left(1-\lambda_{k_j}\right)}{4}} \\
&\quad \cdot \left(\prod_{(i,j)\in E_2} (-1)^{\frac{1-\lambda_{k_j}}{2}} \right) \cdot |\psi_{k_1}\psi_{k_2}\psi_{k_3}\rangle \\
&= \frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle-|010\rangle+|011\rangle \\
&\quad -|100\rangle-|101\rangle-|110\rangle+|111\rangle).
\end{aligned}
$$

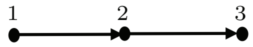


Fig. 5. Digraph $\overrightarrow{C}_3$ with $E_0 = E_2 = E_3 = \emptyset$, $E_1 = \{(1,2),(2,3)\}$.

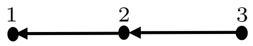


Fig. 6. Inverse graph of digraph $\overrightarrow{C}_3$ with $E_0 = E_1 = E_3 = \emptyset$, $E_2 = \{(3,2),(2,1)\}$.

Generally, digraph state can be implemented by quantum circuit. Specifically, given a digraph $\overrightarrow{G} = (V, \overrightarrow{E})$, one can implement the corresponding digraph state $|\overrightarrow{G}\rangle$ for Eq. (14) using quantum circuit, the procedures are as follows: First, assign to each vertex a qubit initialized as the state $|+\rangle$ so that the total initial state is an N-qubit $|0\rangle^{\otimes N} = \underbrace{|+\rangle|+\rangle\cdots|+\rangle}_{N}$. Then, for every $(i,j)\in\overrightarrow{E}$, make the following operations:

(a) When $i = j$, perform Z operation on the qubit i.

(b) When $i < j$, perform controlled-Z operation on j controlled by qubit i (see Fig. 7).

(c) When $i > j$, we first perform controlled-Z operation on qubit i controlled by qubit j following a Z operation on qubit j (see Fig. 8).

This procedure are demonstrated by the following two examples in Figs. 7 and 8, where

$$
\begin{aligned}
|\overleftarrow{C}_3\rangle &= \frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle-|010\rangle+|011\rangle \\
&\quad -|100\rangle-|101\rangle-|110\rangle+|111\rangle), \\
|\overrightarrow{C}_3\rangle &= \frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle-|011\rangle \\
&\quad +|100\rangle+|101\rangle-|110\rangle+|111\rangle).
\end{aligned}
$$

These figures also show the correspondence between a digraph and the circuit implementation of the corresponding digraph state.

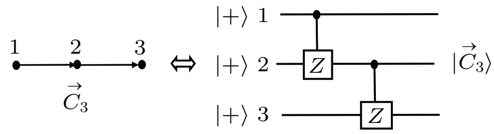


Fig. 7. The digraph $\overrightarrow{C}_3 = (\{1,2,3\},\{(1,2),(2,3)\})$ and the corresponding quantum circuit.

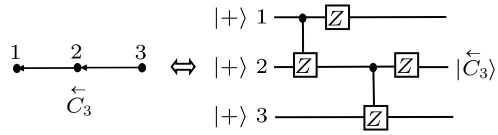


Fig. 8. The digraph $\overleftarrow{C}_3 = (\{1,2,3\},\{(2,1),(3,2)\})$ and the corresponding quantum circuit.

The above procedure implies that it is physically easy to prepare a digraph state. In addition, if there exists an edge $(i,j)\in\overrightarrow{E}$ with $i\neq j$ in a digraph G, i.e., there exist two different vertices that are connected by edge, then the corresponding digraph state must be entangled and then becomes a new kind of multipartite entangled states. Thus, digraph states form a valuable resource for various tasks, including quantum key distribution, randomness extraction, and quantum communication, and so on.

Moreover, we can clearly see $|\overrightarrow{C}_3\rangle \neq |\overleftarrow{C}_3\rangle$ since

$$
|\overrightarrow{C}_3\rangle - |\overleftarrow{C}_3\rangle = \frac{1}{\sqrt{2}}(|010\rangle - |011\rangle + |100\rangle + |101\rangle).
$$

The undirected graph C_3 obtained by deleting arrows in Fig. 5 is given by Fig. 9 and the corresponding graph state $|C_3\rangle$ reads

$$
\begin{aligned}
|C_3\rangle &= \frac{1}{2\sqrt{2}}|000\rangle+|001\rangle+|010\rangle-|011\rangle \\
&\quad +|100\rangle+|101\rangle-|110\rangle+|111\rangle),
\end{aligned}
$$

which is equal to the digraph state $|\overrightarrow{C}_3\rangle$, but not equal to the digraph state $|\overleftarrow{C}_3\rangle$.

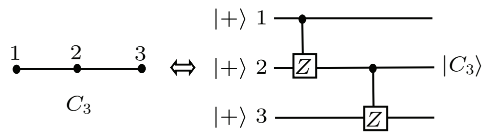
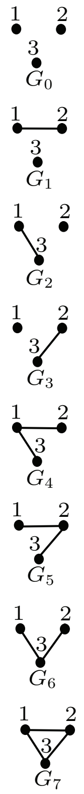


Fig. 9. Undirected graph C_3 and the corresponding quantum circuit.

Indeed, the digraph state $|\overleftarrow{C}_3\rangle$ is not any graph state, referring to Fig. 10 in which all of the 8 graph states of three qubits are list, including $|C_3\rangle = |G_5\rangle$.

Generally, every undirected graph $G = (V, E)$ can be regarded as a digraph $\overrightarrow{G} = (V, \overrightarrow{E})$, where $\overrightarrow{E} = \{(i,j) : (i,j)\in E(i<j)\}$. It easy to see that the digraph state $|\overrightarrow{G}\rangle$ is exactly equal to graph state $|G\rangle$. Thus, digraph states can be regarded as a type of generalizations of graph states. However, they are not the same, e.g., $|\overleftarrow{C}_3\rangle \neq |G_k\rangle$ referring to Fig. 10 for all $k = 0,1,\ldots,7$.



$$|G_0\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle+|110\rangle+|111\rangle)$$

$$|G_1\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle-|110\rangle-|111\rangle)$$

$$|G_2\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle-|101\rangle+|110\rangle-|111\rangle)$$

$$|G_3\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle-|011\rangle+|100\rangle+|101\rangle+|110\rangle-|111\rangle)$$

$$|G_4\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle-|101\rangle-|110\rangle+|111\rangle)$$

$$|G_5\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle-|011\rangle+|100\rangle+|101\rangle-|110\rangle+|111\rangle)$$

$$|G_6\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle-|011\rangle+|100\rangle-|101\rangle+|110\rangle+|111\rangle)$$

$$|G_7\rangle=\frac{1}{2\sqrt{2}}(|000\rangle+|001\rangle+|010\rangle-|011\rangle+|100\rangle-|101\rangle-|110\rangle-|111\rangle)$$

Fig. 10. All possible graph states of three qubits.

4. Representing a digraph state as an NNQS

In this section, we construct a neural network representation of a digraph state $|\overrightarrow{G}\rangle$ using $\{1,-1\}$-input NNQS, that is, to find an NNQS $|\Psi_{S,\Omega}\rangle$ such that $|\overrightarrow{G}\rangle=z|\Psi_{S,\Omega}\rangle$, i.e.,

$$\begin{aligned}&\Psi_{\overrightarrow{G}}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})=z\Psi_{S,\Omega}(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N}),\\&\forall(\lambda_{k_1},\lambda_{k_2},\ldots,\lambda_{k_N})\in\{-1,1\}^N\end{aligned}\tag{17}$$

for some normalized constant z. It is enough to represent the four factors in Eq. (16) as NNQWFs.

For each $(i,j)\in E_2\setminus E_3$ or $(i,j)\in E_1\setminus E_3$, put

$$\begin{aligned}&\Omega_{(i,j)}=(a_{(i,j)},b_{(i,j)},W_{(i,j)}),\\&a_{(i,j)}=\mathbf{0}\in\mathbb{C}^N,\quad b_{(i,j)}=\frac{\pi\mathrm{i}}{4},\\&W_{(i,j)}=[W_{(i,j)s}]\in\mathbb{C}^{1\times N},\\&W_{(i,j)s}=\begin{cases}-\frac{\pi\mathrm{i}}{4}, & s=i\ \text{or}\ s=j;\\0, & \text{otherwise},\end{cases}\end{aligned}$$

then NNQWF $\Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1},\ldots,\lambda_{k_N})$ generated by these parameters is

$$\begin{aligned}&\Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1},\ldots,\lambda_{k_N})\\&=\sum_{h_{(i,j)}=\pm1}\exp\left(\frac{\pi\mathrm{i}}{4}h_{(i,j)}-\frac{\pi\mathrm{i}}{4}h_{(i,j)}\lambda_{k_i}-\frac{\pi\mathrm{i}}{4}h_{(i,j)}\lambda_{k_j}\right)\\&=\sqrt{2}\cdot(-1)^{\frac{(1-\lambda_{k_i})(1-\lambda_{k_j})}{4}}.\end{aligned}$$

This implies that the function $\sqrt{2}\cdot(-1)^{\frac{(1-\lambda_{k_i})(1-\lambda_{k_j})}{4}}$ can be implemented by NNQWF $\Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1},\ldots,\lambda_{k_N})$, which is generated by the neural network with one hidden neuron $h_{(i,j)}$. This process can be illustrated in Fig. 11.

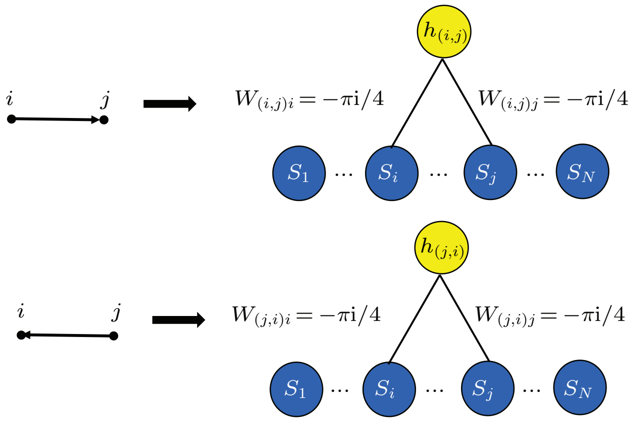


Fig. 11. Neural networks representing the functions $\sqrt{2}\cdot(-1)^{\frac{(1-\lambda_{k_i})(1-\lambda_{k_j})}{4}}$ for $(i,j)\in E_1\setminus E_3$ and $(j,i)\in E_2\setminus E_3$, respectively.

For each $(i,i)\in E_0$, put

$$\begin{aligned}&\Omega_{(i,i)}=(a_{(i,i)},b_{(i,i)},W_{(i,i)}),\ a_{(i,i)}=\mathbf{0}\in\mathbb{C}^N,\ b_{(i,i)}=\frac{\pi\mathrm{i}}{2},\\&W_{(i,i)}=[W_{(i,i)s}]\in\mathbb{C}^{1\times N},\ W_{(i,i)s}=\begin{cases}-\frac{\pi\mathrm{i}}{4}, & s=i;\\0, & s\neq i,\end{cases}\end{aligned}$$

then the resulted NNQWF $\Psi_{S,\Omega_{(i,i)}}(\lambda_{k_1},\ldots,\lambda_{k_N})$ is

$$\begin{aligned}&\Psi_{S,\Omega_{(i,i)}}(\lambda_{k_1},\ldots,\lambda_{k_N})\\&=\sum_{h_{(i,i)}=\pm1}\exp\left(\frac{\pi\mathrm{i}}{2}h_{(i,i)}-\frac{\pi\mathrm{i}}{4}h_{(i,i)}\lambda_{k_i}\right)=\sqrt{2}\cdot(-1)^{\frac{1-\lambda_{k_i}}{2}}.\end{aligned}$$

This implies that the function $\sqrt{2}\cdot(-1)^{\frac{1-\lambda_{k_i}}{2}}$ can be implemented by NNQWF $\Psi_{S,\Omega_{(i,i)}}(\lambda_{k_1},\ldots,\lambda_{k_N})$ for any $(i,i)\in E_0$ given by the neural network with one hidden neuron $h_{(i,i)}$, see Fig. 12.

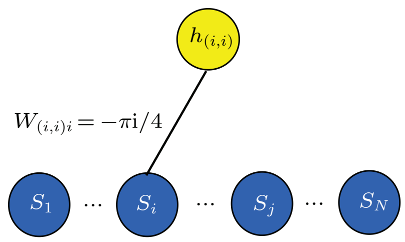


Fig. 12. Neural network generating the function $\sqrt{2}\cdot(-1)^{\frac{1-\lambda_{k_i}}{2}}$ for any $(i,i)\in E_0$.

For each $(i,j)\in E_2$, put

$$\begin{aligned}&\Omega_{(i,j)}=(a_{(i,j)},b_{(i,j)},W_{(i,j)}),\ \ a_{(i,j)}=\mathbf{0}\in\mathbb{C}^N,\ \ b_{(i,j)}=\frac{\pi\mathrm{i}}{2},\\&W_{(i,j)}=[W_{(i,j)s}]\in\mathbb{C}^{1\times N},\ W_{(i,j)s}=\begin{cases}-\frac{\pi\mathrm{i}}{4}, & s=j;\\0, & s\neq j,\end{cases}\end{aligned}$$

then NNQWF $\Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1},\ldots,\lambda_{k_N})$ generated by these parameters is

$$\Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1},\ldots,\lambda_{k_N})$$

$$= \sum_{h_{(i,j)}=\pm 1} \exp\left(\frac{\pi \mathrm{i}}{2} h_{(i,j)} - \frac{\pi \mathrm{i}}{4} h_{(i,j)} \lambda_{k_j}\right) = \sqrt{2} \cdot (-1)^{\frac{1-\lambda_{k_j}}{2}}.$$

This implies that the function $\sqrt{2} \cdot (-1)^{\frac{1-\lambda_{k_j}}{2}}$ can be implemented by NNQWF $\Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1}, \ldots, \lambda_{k_N})$ generated by the neural network with one hidden neuron $h_{(i,j)}$, see Fig. 13.

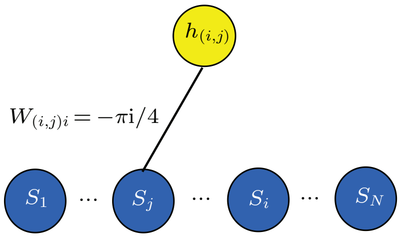


Fig. 13. Neural network representing the function $\sqrt{2} \cdot (-1)^{\frac{1-\lambda_{k_j}}{2}}$ for each $(i,j) \in E_2$.

It follows from Eq. (16) and proposition 2 that

$$\begin{aligned}
&\Psi_{\vec{G}}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}) \\
&= \frac{1}{(\sqrt{2})^{N+|E|}} \left(\prod_{(i,j)\in E_2 \setminus E_3} \Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1}, \ldots, \lambda_{k_N}) \right) \\
&\times \left(\prod_{(i,j)\in E_1 \setminus E_3} \Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1}, \ldots, \lambda_{k_N}) \right) \\
&\times \left(\prod_{(i,i)\in E_0} \Psi_{S,\Omega_{(i,i)}}(\lambda_{k_1}, \ldots, \lambda_{k_N}) \right) \\
&\times \left(\prod_{(i,j)\in E_2} \Psi_{S,\Omega_{(i,j)}}(\lambda_{k_1}, \ldots, \lambda_{k_N}) \right) \\
&= \frac{1}{(\sqrt{2})^{N+|E|}} \Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}),
\end{aligned}$$

for all $(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N}) \in \{-1,1\}^N$.

Now, we have constructed an NNQWF $\Psi_{S,\Omega}(\lambda_{k_1}, \lambda_{k_2}, \ldots, \lambda_{k_N})$ satisfying Eq. (17). This leads to the following conclusion.

Theorem 1 Any digraph state $|\vec{G}\rangle$ can be represented as a spin-z NNQS (7) generated by a neuron network with $|E| + |E_2 \setminus E_3|$ hidden neurons.

If we identity an undigraph $G = (V, E)$ with the digraph $\vec{G} = (V, \vec{E})$ in such a way that $\vec{E} = \{(i,j) : (i,j) \in E\}$, then the states $|G\rangle$ and $|\vec{G}\rangle$ are equal and $|E_2 \setminus E_3| = 0$. With this observation, we have the following corollary.

Corollary 1 Any (undirected) graph state $|G\rangle$ can be represented as a spin-z NNQS (7) generated by a neuron network with $|E|$ hidden neurons.

Example 1 Consider a digraph $\vec{G} = (V, \vec{E})$ with $V = \{1, 2, \ldots, 8\}$ and $\vec{E} = \{(1,2), (1,3), (3,1), (3,4), (4,6), (7,4), (7,5), (6,8), (8,6)\}$, which is represented on the left side of Fig. 14. In this case, the wave function of the digraph state $|\vec{G}\rangle$ reads

$$\begin{aligned}
&\Psi_{\vec{G}}(\lambda_{k_1}, \ldots, \lambda_{k_8}) \\
&= \frac{1}{(\sqrt{2})^8} \left(\prod_{(i,j)\in E_2 \setminus E_3} (-1)^{\frac{(1-\lambda_{k_i})(1-\lambda_{k_j})}{4}} \right) \\
&\times \left(\prod_{(i,j)\in E_1 \setminus E_3} (-1)^{\frac{(1-\lambda_{k_i})(1-\lambda_{k_j})}{4}} \right) \\
&\times \left(\prod_{(i,j)\in E_2} (-1)^{\frac{1-\lambda_{k_j}}{2}} \right),
\end{aligned}$$

where

$$\begin{aligned}
E_1 &= \{(1,2), (1,3), (3,4), (4,6), (6,8)\}, \\
E_2 &= \{(3,1), (7,4), (7,5), (8,6)\}, \\
E_3 &= \{((1,3), (3,1), (6,8), (8,6)\}, \\
E_1 \setminus E_3 &= \{(1,2), (3,4), (4,6)\}, \\
E_2 \setminus E_3 &= \{(7,4), (7,5)\}.
\end{aligned}$$

In the middle of Fig. 14, we demonstrate the idea of constructing a neural network representation of digraph state $|G\rangle$. The neural network that generates $\Psi_G(\lambda_{k_1}, \ldots, \lambda_{k_8})$ is given on the right side of Fig. 14.

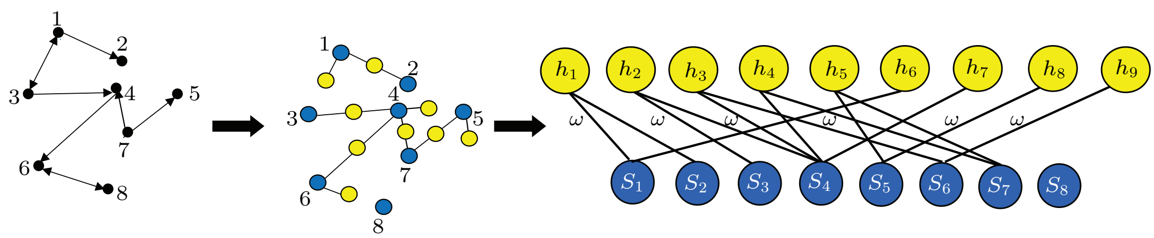


Fig. 14. Neural network representation of digraph states. The first figure is graph representation of a digraph state. The second one is an idea of the process. The third one is neural network representation of the digraph state, where $\omega = -\pi \mathrm{i}/4$, $S_i = \sigma_i^z$, $i = 1, \ldots, 8$.

In this case, the parameters are

$$a = \mathbf{0} \in \mathbb{C}^8, \quad b = \left(\frac{\pi \mathrm{i}}{4}, \frac{\pi \mathrm{i}}{4}, \frac{\pi \mathrm{i}}{4}, \frac{\pi \mathrm{i}}{4}, \frac{\pi \mathrm{i}}{4}, \frac{\pi \mathrm{i}}{2}, \frac{\pi \mathrm{i}}{2}, \frac{\pi \mathrm{i}}{2}, \frac{\pi \mathrm{i}}{2}\right)^{\mathrm{T}} \in \mathbb{C}^9,$$

$$W=\begin{pmatrix} -\pi_1/4 & -\pi_1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\pi_1/4 & -\pi_1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\pi_1/4 & 0 & -\pi_1/4 & 0 & 0 \\ 0 & 0 & 0 & -\pi_1/4 & 0 & 0 & -\pi_1/4 & 0 \\ 0 & 0 & 0 & 0 & -\pi_1/4 & 0 & -\pi_1/4 & 0 \\ -\pi_1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\pi_1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\pi_1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\pi_1/4 & 0 & 0 \end{pmatrix}.$$

5. Conclusion

In summary, we have introduced digraph states and constructed explicitly neural network representations for any digraph state. This means that we have found a new class of entangled multipartite quantum states that can be learned with neural network. Our method shows constructively that all digraph states can be represented precisely by proper neural networks proposed in Ref. [30] and mathematically formulated in Ref. [47]. The obtained results will provide a theoretical foundation for solving the quantum many-body problem with machine learning method whenever the wave-function is known as an unknown digraph state or it can be approximated by digraph states.

Acknowledgements

The authors would like to thank the anonymous referees for their kind comments and suggestions.

This work was supported by the National Natural Science Foundation of China (Grant Nos. 12001480 and 11871318), the Applied Basic Research Program of Shanxi Province (Grant Nos. 201901D211461 and 201901D211462), the Scientific and Technologial Innovation Programs of Higher Education Institutions in Shanxi (Grant No. 2020L0554), the Excellent Doctoral Research Project of Shanxi Province (Grant No. QZX-2020001), and the PhD Start-up Project of Yuncheng University (Grant No. YQ-2019021).

References

[1] Li Y, Ye X J and Chen J L 2016 *Chin. Phys. Lett.* **33** 080301
[2] Gao J and Zhang M C 2016 *Chin. Phys. Lett.* **33** 010303
[3] Du L, Wang Y L, Liang G H, Kang G Z and Zong H S 2016 *Chin. Phys. Lett.* **33** 030301
[4] Sun J and Lu S F 2020 *Chin. Phys. B* **29** 100303
[5] Klein A and Zemach C 1957 *Phys. Rev.* **108** 126
[6] Schweigler T, Kasper V, Erne S, Mazets I, Rauer B, Cataldini F, Langen T, Gasenzer T, Berges J and Schmiedmayer J 2017 *Nature* **545** 323
[7] Hodgman S S, Khakimov R I, Lewis-Swan R J, Truscott A G and Kheruntsyan K V 2017 *Phys. Rev. Lett.* **118** 240402
[8] Schollwöck U 2011 *Ann. Phys.* **326** 96
[9] Kolmogorov A N 1957 *Dokl. Akad. Nauk SSSR* **114** 953
[10] Schuch N, Wolf M M, Verstraete F and Cirac J I 2008 *Phys. Rev. Lett.* **100** 040501
[11] Ceperley D and Alder B 1986 *Science* **231** 555
[12] Schuch N, Wolf M M, Verstraete F and Cirac J I 2007 *Phys. Rev. Lett.* **98** 140506
[13] Verstraete F, Wolf M M, Garcia D P and Cirac J I 2006 *Phys. Rev. Lett.* **96** 220601
[14] Loh E Y, Gubernatis J E, Scalettar R T, White S R, Scalapino D J and Sugar R L 1990 *Phys. Rev. B* **41** 9301
[15] Cybenko G 1989 *Math. Control Signal Syst.* **2** 303
[16] Funahashi K 1989 *Neural Networks* **2** 183
[17] Hornik K, Stinchcombe M and White H 1989 *Neural Networks* **2** 359
[18] Hornik K 1991 *Neural Networks* **4** 251
[19] Kolmogorov A N 1957 *Dokl. Akad. Nauk SSSR* **114** 953
[20] Roux N L and Bengio Y 2008 *Neural Comput.* **20** 1631
[21] LeCun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436
[22] Cao L Z, Wang P J, Sai L W, Fu J and Duan X M 2020 *Chin. Phys. B* **29** 117304
[23] Hinton G E and Salakhutdinov R R 2006 *Science* **313** 504
[24] Salakhutdinov R R, Mnih A and Hinton G 2007 *ICML* **24** 791
[25] Larochelle H and Bengio Y 2008 *ICML* **25** 536
[26] Amin M H, Andriyash E, Rolfe J, Kulchytskyy B and Melko R 2016 *Phys. Rev. X* **8** 021050
[27] Lu S, Gao X and Duan L M 2019 *Phys. Rev. B* **99** 155136
[28] Sarma S D, Deng D L and Duan L M 2019 *Phys. Today* **72** 48
[29] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Maranto L V and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002
[30] Carleo G and Troyer M 2017 *Science* **355** 602
[31] Chen J, Cheng S, Xie H D, Wang L and Xiang T 2018 *Phys. Rev. B* **97** 085104
[32] Robeva E and Seigal A 2019 *Inf. Inference* **8** 273
[33] Clark S R 2018 *J. Phys. A: Math. Theor.* **51** 135301
[34] Huang Y C and Moore J E 2021 *Phys. Rev. Lett.* **127** 170601
[35] Lei S J, Bai D, Ren Z Z and Lyu M J 2021 *Chin. Phys. Lett.* **38** 051101
[36] Yin Q, Xiang G Y, Li C F and Guo G C 2017 *Chin. Phys. Lett.* **34** 030301
[37] Yang Y, Zhang C Y and Cao H X 2019 *Entropy* **21** 82
[38] Nomura Y, Darmawan A S, Yamaji Y and Imada M 2017 *Phys. Rev. B* **96** 205152
[39] Deng D L, Li X P and Sarma S D 2017 *Phys. Rev. B* **96** 195145
[40] Rocchetto A, Grant E, Strelchuk S, Carleo G and Severini S 2018 *npj Quantum Inf.* **4** 28
[41] Glasser I, Pancotti N, August M, Rodriguez I D and Cirac J I 2017 *Phys. Rev. X* **8** 011006
[42] Kaubruegger R, Pastori L and Budich J C 2018 *Phys. Rev. B* **97** 195136
[43] Cai Z 2018 *Phys. Rev. B* **97** 035116
[44] Saito H and Kato M 2017 *J. Phys. Soc. Jpn.* **87** 014001
[45] Jia Z A, Yi B, Zhai R, Wu Y C, Guo G C and Guo G P 2019 *Adv. Quantum Technol.* **2** 1800077
[46] Rao W J 2020 *Chin. Phys. Lett.* **37** 080501
[47] Yang Y, Cao H X and Zhang Z J 2020 *Sci. China-Phys. Mech. Astron.* **63** 210312
[48] Raussendorf R and Briegel H J 2001 *Phys. Rev. Lett.* **86** 5188
[49] Schlingemann D and Werner R F 2002 *Phys. Rev. A* **65** 012308
[50] Mandel O, Greiner M, Widera A, Rom T, Haensch T W and Bloch I 2003 *Nature* **425** 937
[51] Walther P, Resch J K, Rudolph T, Schenck E and Weinfurter H 2005 *Nature* **434** 169
[52] Gao X and Duan L M 2017 *Nat. Commun.* **8** 662
[53] Kottos T and Smilansky U 1997 *Phys. Rev. Lett.* **79** 4794
[54] Kottos T and Smilansky U 1999 *Ann. Phys. N.Y.* **274** 76
[55] Tanner G 2000 *J. Phys. A: Math. Gen.* **33** 3567
[56] Tanner G 2001 *J. Phys. A: Math. Gen.* **34** 8485
[57] Pakoński P, Życzkowski K and Kuś M 2001 *J. Phys. A: Math. Gen.* **34** 9303
[58] Chang C T and Hwang H C 1992 *Ind. Eng. Chem. Res.* **31** 1490
[59] Palmer C and Chung P W H 2000 *Ind. Eng. Chem. Res.* **39** 2548
[60] Bhushan M and Rengaswamy R 2000 *Ind. Eng. Chem. Res.* **39** 999
[61] Deng D L, Li X P and Sarma S D 2017 *Phys. Rev. B* **96** 195145
[62] Kay E 1977 *J. Oper. Res. Soc.* **28** 237