Quantum Information Processing (2019) 18:230
https://doi.org/10.1007/s11128-019-2348-9

Variational learning the SDC quantum protocol with gradient-based optimization

Haozhen Situ[1,2] · Zhiming Huang[3] · Xiangfu Zou[2,5] · Shenggen Zheng[2,4]

Received: 22 February 2019 / Accepted: 28 May 2019 / Published online: 3 June 2019

Abstract

Recently, a variational learning approach is adopted to discover quantum communication protocols (Wan et al. in npj Quantum Inf 3:36, 2017). Because designing quantum protocols manually is a delicate and difficult work, this variational learning approach is well worth further study. In this paper, we use the same approach to learn the simultaneous dense coding (SDC) protocols with two or three receivers. The gradient-based optimization is used to learn the parameters of the locking operator of the SDC protocol. Two different designs of the loss function are considered. Numerical experiment results show the effectiveness of this variational learning approach.

Keywords Quantum machine learning · Quantum protocol · Simultaneous dense coding

✉ Zhiming Huang
465609785@qq.com

Haozhen Situ
situhaozhen@gmail.com

Xiangfu Zou
xf.zou@hotmail.com

Shenggen Zheng
zhengshg@pcl.ac.cn

1 College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China

2 Center for Quantum Computing, Peng Cheng Laboratory, Shenzhen 518055, China

3 School of Economics and Management, Wuyi University, Jiangmen 529020, China

4 Institute for Quantum Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

5 School of Mathematics and Computational Science, Wuyi University, Jiangmen 529020, China

1 Introduction

Quantum machine learning (QML) [1] is a new research subject which combines the studies of quantum information processing [2] and machine learning [3]. There are two main directions in the field of QML: (1) use quantum computing resource to improve the efficiency of machine learning and (2) use machine learning techniques to solve quantum problems. In this paper, we are interested in the second direction. Machine learning has achieved great success in various fields such as manufacturing, finance, retailing, medical treatment and transportation. Recently, some researches from the community of quantum information have tried to learn quantum protocols or algorithms using machine learning [4–6]. We think this approach will open a new way to the discovery of more and more new quantum protocols or algorithms.

The simultaneous dense coding (SDC) quantum communication protocol was first proposed in 2010 [7] and has been widely studied in the past few years [8–13]. The SDC protocol can distribute secret classical messages to two receivers and ensure that the receivers must collaborate to reveal the secrets simultaneously. The application of the SDC protocol in fair contract signing was given in Ref. [9]. The fundamental building block of the SDC protocol is the locking operator, which can entangle two different dense coding channels, so a single receiver cannot obtain information about the encoded bits. Finding locking operators is the main task of designing new SDC protocols. Several locking operators have been proposed in the literature. However, a general method for finding locking operators is still absent. Furthermore, it is not yet known how to construct a locking operator for the SDC protocol with three receivers. (We call it the SDC3R protocol in the following for convenience.)

In this paper, we try to use gradient-based optimization, one of the most important machine learning techniques, to find the locking operators for the SDC protocols with two or three receivers. The locking operators for SDC2R (abbreviation for SDC with two receivers) are parameterized by 16 real numbers, while the locking operators for SDC3R are parameterized by 64 real numbers. The optimization goal is to minimize a loss function, which is defined based on the trace distance or the fidelity [2], in order to measure the distance or similarity of the subsystems corresponding to different encoded bits. The gradient-based optimization algorithm we adopt here is the Adam algorithm [14], a representative member from the family of gradient-based optimization algorithms. By numerical experiments, we show the effectiveness of this variational approach for finding new quantum communication protocols.

The remainder of this paper is organized as follows. Section 2 briefly reviews the SDC protocol that we will try to learn in this work. Section 3 gives the parameterization, the design of the loss function and the optimization algorithm. Section 4 reports the results of the numerical experiments we perform in the MATLAB environment. A brief conclusion follows in Sect. 5.

2 Review of the SDC protocol

For the sake of completeness, we first review the standard SDC2R protocol [7]. Other variants of SDC protocols can be found in the literature [8–11]. In the standard SDC2R

protocol, there are one sender called Alice and two receivers called Bob and Charlie, respectively. Initially, Alice and Bob share a maximally entangled quantum state $\frac{1}{2}(|00\rangle + |11\rangle)_{A_1B}$, and Alice and Charlie also share a pair of maximally entangled qubits $\frac{1}{2}(|00\rangle + |11\rangle)_{A_2C}$. The protocol proceeds as follows:

1. *Encoding* Alice performs unitary transforms on qubits A_1 and A_2 to encode 4 classical bits $b_1b_2c_1c_2$.
2. *Locking* Alice performs the locking operator U on qubits A_1 and A_2 to entangle the four qubits A_1BA_2C. Then, A_1 is sent to Bob and A_2 is sent to Charlie, respectively.
3. *Unlocking* Bob and Charlie collaborate to perform $U^\dagger$ on qubits A_1 and A_2.
4. *Decoding* Bob and Charlie perform the Bell state measurement on qubits A_1B and A_2C, respectively, to achieve b_1b_2 and c_1c_2.

A carefully chosen locking operator can make the subsystem A_1B and the subsystem A_2C after the *Locking* phase independent of the encoded bits $b_1b_2c_1c_2$, so the receivers cannot know the secret bits unless they collaborate to finish the *Unlocking* phase.

Besides SDC2R, we also study SDC3R in this paper. The third receiver is named by Donald, who also shares a maximally entangled quantum state $\frac{1}{2}(|00\rangle + |11\rangle)_{A_3D}$ with Alice and intends to receive 2 bits d_1d_2. The locking operator involves 3 qubits $A_1A_2A_3$. It is required that subsystems A_1B, A_2C, A_3D after the *Locking* phase are independent of the encoded bits.

In the following sections, we design the machine learning scheme and perform numerical experiments for the SDC2R and SDC3R protocols.

3 Learning the SDC protocol

This section introduces the scheme for learning the locking operator of the SDC protocol, including the parameterization of the locking operator, the construction of the loss function and the optimization algorithm.

3.1 Parameterization

Given that any unitary matrix U can be expressed as $U = e^{iH}$, where H is a Hermitian matrix, and that such matrices can be written as linear combinations of tensor products of the Pauli matrices and the identity, it follows that a general $N-$qubit unitary can be expressed as

$$U_N = \exp\left[i \left(\sum_{j_1,\ldots,j_N=0,\ldots,0}^{3,\ldots,3} p_{j_1,\ldots,j_N} (\sigma_{j_1} \otimes \ldots \otimes \sigma_{j_N}) \right) \right], \tag{1}$$

where σ_i are the Pauli matrices for $i \in \{1, 2, 3\}$ and σ_0 is the 2×2 identity matrix [4].

The parameters $p_{j_1,\ldots,j_N}$ are initialized with random values and optimized through iterations of slight updates. A two-qubit locking operator for SDC2R has $4 \times 4 = 16$ real parameters, while a three-qubit locking operator for SDC3R has $4 \times 4 \times 4 = 64$ real parameters.

3.2 Loss function

We first review the concepts of trace distance and fidelity between quantum states, and then we can define the loss functions based on trace distance and fidelity.

The trace distance between quantum states ρ and σ is defined as

$$D(\rho, \sigma) = \frac{1}{2}\mathrm{tr}|\rho - \sigma|, \tag{2}$$

where $|A| = \sqrt{A^\dagger A}$ is the positive square root of $A^\dagger A$. Notice that the quantum trace distance generalizes the classical trace distance in the sense that if ρ and σ commute, then the trace distance between ρ and σ is equal to the classical trace distance between the eigenvalues of ρ and σ. The trace distance can be preserved under unitary transformations, i.e., $D(U\rho U^\dagger, U\sigma U^\dagger) = D(\rho, \sigma)$. $D(\rho, \sigma) = 0$ if and only if $\rho = \sigma$, and $D(\cdot, \cdot)$ is a symmetric function of its inputs, i.e., $D(\rho, \sigma) = D(\sigma, \rho)$ [2].

The fidelity of states ρ and σ is defined to be

$$F(\rho, \sigma) = \mathrm{tr}\sqrt{\rho^{\frac{1}{2}}\sigma\rho^{\frac{1}{2}}}. \tag{3}$$

The fidelity is symmetric in its inputs, i.e., $F(\rho, \sigma) = F(\sigma, \rho)$. When ρ and σ commute the quantum fidelity, $F(\rho, \sigma)$ reduces to the classical fidelity $F(r_i, s_i)$ between the eigenvalue distributions r_i and s_i of ρ and σ. The fidelity is invariant under unitary transformations, i.e., $F(U\rho U^\dagger, U\sigma U^\dagger) = F(\rho, \sigma)$. The fidelity is bounded between 0 and 1, i.e., $0 \leqslant F(\rho, \sigma) \leqslant 1$. If $\rho = \sigma$, then $F(\rho, \sigma) = 1$. If $\rho \neq \sigma$, then $F(\rho, \sigma) < 1$. $F(\rho, \sigma) = 0$ if and only if ρ and σ have support on orthogonal subspaces [2].

Now the loss function for SDC2R can be constructed using the trace distance or the fidelity. We use the notation $\rho_{\mathrm{A_1B}}(b_1b_2c_1c_2)$ to denote the subsystem $\mathrm{A_1B}$ after the *Locking* phase with the encoded bits being $b_1b_2c_1c_2$. $\rho_{\mathrm{A_2C}}(b_1b_2c_1c_2)$ is defined analogously. We hope the locking operator can make the states $\rho_{\mathrm{A_1B}}(b_1b_2c_1c_2)$ as close as possible for different $b_1b_2c_1c_2$. That is to minimize

$$\sum_{b_1b_2c_1c_2 \neq b_1'b_2'c_1'c_2'} D(\rho_{\mathrm{A_1B}}(b_1b_2c_1c_2), \rho_{\mathrm{A_1B}}(b_1'b_2'c_1'c_2')), \tag{4}$$

or to maximize

$$\sum_{b_1b_2c_1c_2 \neq b_1'b_2'c_1'c_2'} F(\rho_{\mathrm{A_1B}}(b_1b_2c_1c_2), \rho_{\mathrm{A_1B}}(b_1'b_2'c_1'c_2')). \tag{5}$$

The same criterion holds for the case of subsystem A_2C. By adding a minus sign before Eq. 5 and taking the symmetry property of the trace distance and the fidelity into consideration, we construct two loss functions:

$$L_D^{2R} = \frac{1}{2} \cdot \frac{1}{(16^2-16)/2} \left(\sum_{b_1b_2c_1c_2 \neq b_1'b_2'c_1'c_2'} D(\rho_{A_1B}(b_1b_2c_1c_2), \rho_{A_1B}(b_1'b_2'c_1'c_2')) \right.$$
$$\left. + \sum_{b_1b_2c_1c_2 \neq b_1'b_2'c_1'c_2'} D(\rho_{A_2C}(b_1b_2c_1c_2), \rho_{A_2C}(b_1'b_2'c_1'c_2')) \right), \tag{6}$$

$$L_F^{2R} = -\frac{1}{2} \cdot \frac{1}{(16^2-16)/2} \left(\sum_{b_1b_2c_1c_2 \neq b_1'b_2'c_1'c_2'} F(\rho_{A_1B}(b_1b_2c_1c_2), \rho_{A_1B}(b_1'b_2'c_1'c_2')) \right.$$
$$\left. + \sum_{b_1b_2c_1c_2 \neq b_1'b_2'c_1'c_2'} F(\rho_{A_2C}(b_1b_2c_1c_2), \rho_{A_2C}(b_1'b_2'c_1'c_2')) \right). \tag{7}$$

There are $2 \cdot (16^2 - 16)/2$ terms in the parentheses, so a divisor is added to get the average trace distance and fidelity.

The loss functions for SDC3R can be constructed analogously.

$$L_D^{3R} = \frac{1}{3} \cdot \frac{1}{(64^2-64)/2} \left(\sum D(\rho_{A_1B}(b_1b_2c_1c_2d_1d_2), \rho_{A_1B}(b_1'b_2'c_1'c_2'd_1'd_2')) \right.$$
$$+ \sum D(\rho_{A_2C}(b_1b_2c_1c_2d_1d_2), \rho_{A_2C}(b_1'b_2'c_1'c_2'd_1'd_2'))$$
$$\left. + \sum D(\rho_{A_3D}(b_1b_2c_1c_2d_1d_2), \rho_{A_3D}(b_1'b_2'c_1'c_2'd_1'd_2')) \right), \tag{8}$$

$$L_F^{3R} = -\frac{1}{3} \cdot \frac{1}{(64^2-64)/2} \left(\sum F(\rho_{A_1B}(b_1b_2c_1c_2d_1d_2), \ \rho_{A_1B}(b_1'b_2'c_1'c_2'd_1'd_2')) \right.$$
$$+ \sum F(\rho_{A_2C}(b_1b_2c_1c_2d_1d_2), \rho_{A_2C}(b_1'b_2'c_1'c_2'd_1'd_2'))$$
$$\left. + \sum F(\rho_{A_3D}(b_1b_2c_1c_2d_1d_2), \rho_{A_3D}(b_1'b_2'c_1'c_2'd_1'd_2')) \right), \tag{9}$$

where the summations are all done over condition $b_1b_2c_1c_2d_1d_2 \neq b_1'b_2'c_1'c_2'd_1'd_2'$. There are $3 \cdot (64^2 - 64)/2$ terms in the parenthesis, so a divisor is added to get the average trace distance and fidelity.

In the numerical experiment, the performance of L_D^{2R} and L_F^{2R} will be compared in the case of SDC2R, and the comparison between L_D^{3R} and L_F^{3R} will also be made in the case of SDC3R.

3.3 Optimization

The gradient descent method is the most important optimization tool in machine learning. Given an optimization problem, this method searches the solution space from a

Algorithm 1 Learning the SDC protocol using gradient-based optimization

Input: f: $D(\cdot,\cdot)$ or $F(\cdot,\cdot)$, specifying the trace distance or the fidelity as the distance of similarity measure of two quantum states
Input: num_epoch: Number of times of parameters' updates
Input: ε: Stepsize of forward difference
1: Initialize the parameters $p^{(0)}$ with uniformly random numbers in $(-1, 1)$
2: $t \leftarrow 0$
3: **repeat**
4: $t \leftarrow t + 1$
5: Calculate the gradient using Eq. 11
6: Update the parameters using Adam optimization
7: **until** $t =$ num_epoch
8: **return** L_f and p

random starting point $p^{(0)}$, where p stands for the vector of learnable parameters. In the tth epoch, the gradient of the loss function with respect to each adjustable parameter is estimated, and then the parameters are updated according to the formula

$$p^{(t)} = p^{(t-1)} - \textbf{learning\_rate} \cdot \frac{\partial L}{\partial p}, \tag{10}$$

where **learning_rate** is a small number specifying the degree of modification. Usually, this quantity needs to be tuned in the numerical experiment in order to make the training process converge fast without oscillation around the optimal solution. Some advanced gradient-based optimization algorithms have been proposed to vary the learning rate during the training process. At the same time, more heuristic statistics have been put into the update formula.

Here, we use a famous and frequently used gradient-based optimization algorithm called Adam [14]. The name Adam is derived from adaptive moment estimation. This method combines the advantages of two popular optimization methods: the ability of AdaGrad [15] to deal with sparse gradients and the ability of RMSProp [16] to deal with non-stationary objectives.

The partial derivatives of the loss function L with respect to each parameter p are approximated using the forward difference formula:

$$\frac{\partial L}{\partial p} \approx \frac{L(p+\varepsilon) - L(p)}{\varepsilon}, \tag{11}$$

which is the same as the approximating formula used for the numerical partial derivatives of the loss function in Ref. [4]. The limit of the right hand side of the above formula as ε approaches zero is the partial derivative of the loss function L with respect to the parameter p.

The whole training process is described in Algorithm 1.

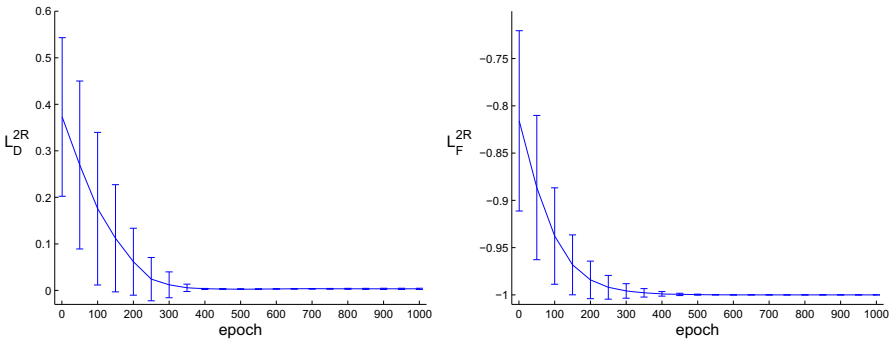

Fig. 1 The average and standard deviation of the loss function with respect to the number of epochs. (Left) Learning the SDC2R protocol, using trace distance as a measure. (Right) Learning the SDC2R protocol, using fidelity as a measure

4 Numerical experiment

This section reports the numerical experiment results of learning the SDC2R and SDC3R protocols. The experiment is done on MATLAB R2014a. For each protocol, we perform 10 independent trainings of 1000 epoches and then report the averages and the standard deviations of the loss function every 50 epoches. Both the trace distance and the fidelity are considered in our experiments. The stepsize of forward difference is $\varepsilon = 0.01$. A larger stepsize results in the oscillation around the optimal solution, while a smaller stepsize may cause the precision issue in numerical calculation software. The hyper-parameters of the Adam algorithm are set to the suggested values in Ref. [14] ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$). The learnable parameters in the first epoch are picked from uniformly random distribution in $(-1, 1)$ and then updated according to the Adam optimization.

The training for the SDC2R protocol is depicted in Fig. 1, from which we can see that the loss converges to 0 in the case of trace distance and converges to -1 in the case of fidelity. Because the loss represents the average trace distance or the minus average fidelity between pairs of quantum states corresponding to different encodings, these states are approximately identical, so the receivers have no information about the encoded bits before the *Unlocking* phase. We can conclude that the optimal solution has been found.

The training for the SDC3R protocol is depicted in Fig. 2. When the trace distance is used in the loss function, the loss decreases gradually in the first 200 epoches and then oscillates around 0.02. When the fidelity is used in the loss function, the loss decreases gradually and converges to -1, so the training is successful. Comparing these two figures, it is clear that fidelity is a better choice than the trace distance in the training of SDC3R protocol.

By comparing Eqs. 2 and 3, we find that the definition of the trace distance involves four product terms under the square root sign, i.e., $\rho^\dagger\rho - \rho^\dagger\sigma - \sigma^\dagger\rho + \sigma^\dagger\sigma$, while there is only one product term under the square root sign in the definition of the fidelity. In this respect, the loss function based on the fidelity is simpler to be optimized. This

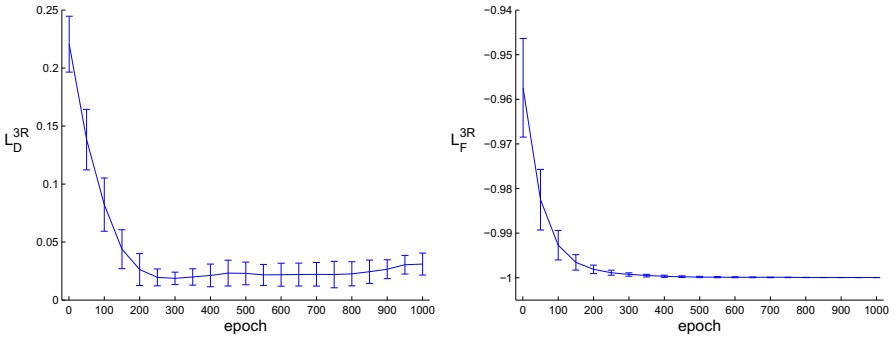

Fig. 2 The average and standard deviation of the loss function with respect to the number of epochs. (Left) Learning the SDC3R protocol, using trace distance as a measure. (Right) Learning the SDC3R protocol, using fidelity as a measure

explains the oscillatory behavior and the large standard deviation of L_D^{3R} in Fig. 2. In fact, L_D^{2R} also has larger standard deviation than L_F^{2R} in Fig. 1.

5 Conclusion

In this paper, we use the popular gradient descent method to optimize the parameters of a unitary operator, which acts as the locking operator in the SDC protocol. Numerical experiment results show that the training is successful. After training, the locking operator can make sure that each receiver's subsystem state is independent of the encoded bits, so they have to collaborate to perform the unlock operation to restore the states that encodes the secret bits. A comparative study demonstrates that the design of loss function has influence on the training result. In our scenario, using the fidelity to measure the distance between two quantum states leads to better training result than the trace distance. Together with previous research results [4–6], the hybrid approach combining machine learning and quantum computing has been verified to be a useful and promising method to the discovery of new quantum protocols and algorithms.

Acknowledgements We are very grateful to the reviewers and the editors for their invaluable comments and detailed suggestions that helped to improve the quality of the present paper. This work is supported by the National Natural Science Foundation of China (Nos. 61772565, 61602532, 61871205), the Innovation Project of Department of Education of Guangdong Province of China (No. 2017KTSCX180), the Jiangmen Science and Technology Plan Project for Basic and Theoretical Research (No. 2018JC01010) and the Science and Technology Project of Jiangmen City of China (No. 2018JC01019).

Appendix

This "Appendix" gives the solutions for SDC2R and SDC3R found by our methods. Because our method is inherently stochastic, different runs of the experiment give

different optimal solutions. Here, we list one optimal solution for SDC2R and one optimal solution for SDC3R, respectively.

$$U_2 = \begin{pmatrix} -0.0393 + 0.7878\mathrm{i} & -0.1875 + 0.4444\mathrm{i} & -0.1661 + 0.3331\mathrm{i} & -0.0470 + 0.0664\mathrm{i} \\ -0.3271 + 0.1792\mathrm{i} & -0.0690 + 0.0344\mathrm{i} & 0.8352 - 0.0342\mathrm{i} & 0.2516 - 0.3049\mathrm{i} \\ -0.4302 + 0.2283\mathrm{i} & 0.8346 - 0.1481\mathrm{i} & -0.0469 + 0.0035\mathrm{i} & -0.0995 + 0.1794\mathrm{i} \\ -0.0371 - 0.0115\mathrm{i} & -0.1982 + 0.0601\mathrm{i} & 0.3466 - 0.2011\mathrm{i} & -0.2396 + 0.8589\mathrm{i} \end{pmatrix}, \quad (12)$$

$$U_3 = \begin{pmatrix} 0.1225 & 0.0103 & -0.3390 & -0.1423 & -0.1179 & 0.1186 & -0.1449 & -0.1298 \\ -0.4212 & 0.0371 & -0.0202 & -0.5556 & -0.1380 & -0.2497 & 0.2013 & 0.0684 \\ -0.5623 & -0.4511 & 0.3432 & -0.0888 & -0.2240 & 0.1570 & -0.1438 & -0.2143 \\ 0.1207 & -0.0993 & 0.3640 & 0.3322 & -0.2098 & -0.3190 & -0.4001 & 0.0796 \\ 0.4173 & -0.1945 & 0.2179 & -0.3981 & 0.1161 & -0.2262 & -0.1019 & -0.3981 \\ 0.0677 & -0.1777 & -0.0841 & -0.1454 & -0.0134 & 0.2191 & 0.2885 & 0.4974 \\ 0.2113 & 0.0094 & 0.2078 & -0.4535 & -0.1338 & 0.1132 & -0.1770 & 0.4125 \\ -0.0302 & 0.3255 & 0.0905 & -0.0081 & -0.4659 & -0.1066 & -0.2799 & 0.3438 \end{pmatrix}$$
$$+ \mathrm{i} \begin{pmatrix} 0.0729 & -0.1310 & 0.0849 & 0.0888 & -0.1087 & -0.7016 & 0.4813 & 0.1032 \\ 0.0961 & 0.1796 & -0.3401 & -0.0640 & -0.4289 & -0.0927 & -0.0548 & -0.1693 \\ 0.0320 & -0.0355 & 0.0879 & 0.1977 & 0.2519 & 0.0032 & 0.1044 & 0.2996 \\ 0.1804 & -0.3197 & -0.2770 & 0.0217 & -0.3671 & 0.0686 & 0.2294 & -0.1283 \\ 0.2255 & -0.1263 & -0.1262 & -0.2689 & 0.2459 & -0.1643 & -0.3169 & 0.0752 \\ 0.0613 & -0.6424 & -0.1938 & 0.0704 & 0.0080 & -0.0004 & -0.1700 & 0.2626 \\ 0.2238 & 0.1253 & 0.4775 & -0.1429 & 0.0168 & 0.2567 & 0.2942 & -0.0852 \\ -0.3158 & 0.1119 & -0.1989 & -0.1350 & 0.4117 & -0.2747 & -0.2011 & 0.0677 \end{pmatrix}. \quad (13)$$

The known unlocking operators for SDC2R in the previous literature include quantum Fourier transform [7], double controlled-NOT [9] and SWAP [9] operators. The quantum circuits for these three unlocking operators are already known. In contrast, the optimal solutions found by our method need to be compiled into quantum circuits or implemented by special quantum devices. However, our method is quite useful for constructing unlocking operators for SDC protocols with more than two receivers.

References

1. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**, 195 (2017)
2. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
4. Wan, K.H., Dahlsten, O., Kristjánsson, H., Gardner, R., Kim, M.S.: Quantum generalisation of feed-forward neural networks. npj Quantum Inf. **3**, 36 (2017)
5. Cincio, L., Subasi, Y., Sornborger, A.T., Coles, R.J.: Learning the quantum algorithm for state overlap. New J. Phys. **20**, 113022 (2018)
6. Morales, M.E.S., Tlyachev, T., Biamonte, J.: Variational learning of Grovers quantum search algorithm. Phys. Rev. A **98**, 062333 (2018)
7. Situ, H.Z., Qiu, D.W.: Simultaneous dense coding. J. Phys. A Math. Theor. **43**, 055301 (2010)
8. Situ, H.Z.: Controlled simultaneous teleportation and dense coding. Int. J. Theor. Phys. **53**, 1003 (2014)
9. Situ, H.Z., Qiu, D.W., Mateus, P., Paunkovic, N.: Secure N-dimensional simultaneous dense coding and applications. Int. J. Quantum Inf. **13**, 1550051 (2015)
10. Zhang, C., Situ, H.Z., Li, Q., He, G.P.: Efficient simultaneous dense coding and teleportation with two-photon four-qubit cluster states. Int. J. Quantum Inf. **14**, 1650023 (2016)

11. Yang, X., Bai, M.Q., Zuo, Z.C., Mo, Z.W.: Secure simultaneous dense coding using χ-type entangled state. Quantum Inf. Process. **17**, 261 (2018)
12. Huang, Z.M., Zhang, C., Situ, H.Z.: Performance analysis of simultaneous dense coding protocol under decoherence. Quantum Inf. Process. **16**, 227 (2017)
13. Huang, Z.M., Ye, Y.Y., Luo, D.R.: Simultaneous dense coding affected by fluctuating massless scalar field. Quantum Inf. Process. **17**, 101 (2018)
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv: 1412.6980
15. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121 (2011)
16. Tieleman, T., Hinton, G.: Lecture 6.5—RMSProp, COURSERA: neural networks for machine learning. Technical report (2012)