Time Series Prediction by Quaternionic Qubit Neural Network

Takuya Teguri
Graduate School of Engineering
University of Hyogo
Himeji, Hyogo, Japan

Teijiro Isokawa
Graduate School of Engineering
University of Hyogo
Himeji, Hyogo, Japan
isokawa@eng.u-hyogo.ac.jp

Nobuyuki Matsui
University of Hyogo
Himeji, Hyogo, Japan

Haruhiko Nishimura
Graduate School of Applied Informatics
University of Hyogo
Kobe, Hyogo, Japan

Naotake Kamiura
Graduate School of Engineering
University of Hyogo
Himeji, Hyogo, Japan

Abstract—We propose a neural network model based on quantum information processing with quaternionic representation and operations, called Quaternionic Qubit Neural Network. The state of a neuron is represented by a point on the Bloch sphere with incorporating quaternionic representation. The operations for this neuron also follow the operations in quaternions. The proposed neural networks are evaluated through numerical experiments for predicting chaotic time series produced by a Lorentz system. They have better performances in predicting long-term series, as compared to conventional (real-valued) neural networks.

Index Terms—qubit, quaternion, Bloch sphere, Lorentz system

I. Introduction

Much attention has been paid recently to the quantum computational intelligence (QCI), which is a promising method for improving the information processing capabilities in conventional artificial intelligence methods [1]. Among various QCI methods being proposed and available, some researchers have developed the quantum-neuro computing in which the algorithm of quantum computation is used to improve the efficiency of neural computing system [2], [3]. The quantum state and the operator of quantum computation are both important to realize parallelisms and plasticity respectively in information processing systems. Complex valued representation of these quantum concepts allow neural computation system to advance in learning abilities and to enlarge its possibility of practical applications. We have proposed Qubit(quatntum bit) neural network models and investigated their properties and performances, such as the effects of quantum superposition and probabilistic interpretation in constructing quantum computing to neural networks [4]–[6].

This study was financially supported by Japan Society for the Promotion of Science (Grant-in-Aids for Scientific Research (C) 19K12141).

Many qubit neural network models, including our models, adopt the complex numbers to represent neuronal states and also use the operators in complex numbers for operating these states. This is useful for achieving efficient manipulation for the phase angle in neuron's state, but the degree of freedom in neuron's state is limited by using one phase angle. The pure state of the qubit system is represented as a point in Bloch sphere [7], i.e., two phase angles, resulting in more degree of freedoms in neuron's state. The state can be manipulated by incorpolating unitary transformations using Pauli matrices, and these operations are equivalent to the operations in quaternions, which is one of hypercomplex number systems [8], [9]. It is expected to extend the neuronal states by using Bloch representation and to incorporate operators by using quaternions, in order to attain rich representation ability for neuron's state and more efficient ability for manipulating neuron's state. Though several types of qubit-based neural network models have been proposed, and there are also a lot of neural networks based on quaternions, there are few neural network models incorpolating both of quantum information processing and quaternionic operators.

In this paper, we propose a neural network model based on quantum information processing based on quaternionic represetation and operations. Called Quaternionic Qubit Neural Network (QQNN), this network accepts three-dimensional signals using pure imaginary quaternions. Operations for neuronal signals in qubits, i.e. single-bit rotation gate and two-bit controlled NOT (CNOT) gates, are described by operations in quaternions. QQNN in this paper is a type of multilayer perceptron neural networks with back-propagation algorithm as its learning scheme, thus a QQNN equivalent of back-propagation algorithm has been formulated. The proposed QQNN is evaluated through numerical experiments as compared with conventional (real-valued) neural network. The task

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

for these network is to conduct long-term prediction for time series of a Lorentz system (three-dimensional chaotic system) [10]. The rest of the paper is organized as follows. Section 2 gives the basic definitions of quaternion number system. The proposed neuron model and its network, followed by quaternionic representation and operations for qubit and quantum gates, are described in section 3. Experimental setups and results are demonstrated in section 4. Finally this paper concludes with section 5.

II. Definition of Quaternion

Quaternions form a class of hypercomplex numbers that consist of a real number and three imaginary numbers– $\boldsymbol{i}, \boldsymbol{j}$, and $\boldsymbol{k}$. Formally, a quaternion number is defined as a vector $\boldsymbol{x}$ in a 4-dimensional vector space,

$$\boldsymbol{x} = x^{(e)} + x^{(i)}\boldsymbol{i} + x^{(j)}\boldsymbol{j} + x^{(k)}\boldsymbol{k} \tag{1}$$

where $x^{(e)}$, $x^{(i)}, x^{(j)}$, and $x^{(k)}$ are real numbers. $\boldsymbol{H}$, the division ring of quaternions, thus constitutes the four-dimensional vector space over the real numbers with the bases $1, \boldsymbol{i}, \boldsymbol{j}$, and $\boldsymbol{k}$. Eq.(1) can also be written using 4-tuple or 2-tuple notation as

$$\boldsymbol{x} \;=\; (x^{(e)}, x^{(i)}, x^{(j)}, x^{(k)}) = (x^{(e)}, \vec{x}), \tag{2}$$

where $\vec{x} = \{x^{(i)}, x^{(j)}, x^{(k)}\}$. In this representation, $x^{(e)}$ is the scalar part of $\boldsymbol{x}$, and $\vec{x}$ forms the vector part. Pure imaginary quaternion corresponds to a pure imaginary number in complex numbers, a quternion $\boldsymbol{x}$ without the scalar part ($x^{(e)} = 0$). The quaternion conjugate is defined as

$$\boldsymbol{x}^* = (x^{(e)}, -\vec{x}) = x^{(e)} - x^{(i)}\boldsymbol{i} - x^{(j)}\boldsymbol{j} - x^{(k)}\boldsymbol{k}. \tag{3}$$

Quaternion bases satisfy the following identities,

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1, \tag{4}$$

$$\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k},\;\; \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i},\;\; \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}, \tag{5}$$

which are known as the Hamilton rule. From these rules, it follows immediately that multiplication of quaternions is not commutative.

Now, we define the operations between quaternions $\boldsymbol{p} = (p^{(e)}, \vec{p}) = (p^{(e)}, p^{(i)}, p^{(j)}, p^{(k)})$ and $\boldsymbol{q} = (q^{(e)}, \vec{q}) = (q^{(e)}, q^{(i)}, q^{(j)}, q^{(k)})$. The addition and subtraction of quaternions are defined in the same manner as they are for complex-valued numbers or vectors, that is,

$$\boldsymbol{p} \pm \boldsymbol{q} = (p^{(e)} \pm q^{(e)},\; \vec{p} \pm \vec{q}) \tag{6}$$

$$= (p^{(e)} \pm q^{(e)}, p^{(i)} \pm q^{(i)}, p^{(j)} \pm q^{(j)}, p^{(k)} \pm q^{(k)}). \tag{7}$$

The product of $\boldsymbol{p}$ and $\boldsymbol{q}$ is determined by Eq. (5) as

$$\boldsymbol{pq} = (p^{(e)}q^{(e)} - \vec{p}\cdot\vec{q},\;\; p^{(e)}\vec{q} + q^{(e)}\vec{p} + \vec{p}\times\vec{q}), \tag{8}$$

where $\vec{p}\cdot\vec{q}$ and $\vec{p}\times\vec{q}$ denote the dot and cross products, respectively, between three-dimensional vectors $\vec{p}$ and $\vec{q}$. The conjugate of the product is given as

$$(\boldsymbol{pq})^* = \boldsymbol{q}^*\boldsymbol{p}^*. \tag{9}$$

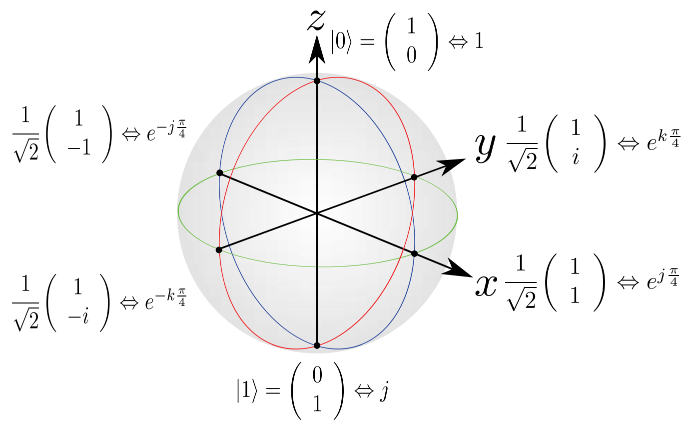

Fig. 1. The quantum state on Bloch sphere (from Fig.1 in [7])

The quaternion norm of $\boldsymbol{x}$, denoted by $|\boldsymbol{x}|$, is defined as

$$|\boldsymbol{x}| = \sqrt{\boldsymbol{x}\boldsymbol{x}^*} = \sqrt{{x^{(e)}}^2 + {x^{(i)}}^2 + {x^{(j)}}^2 + {x^{(k)}}^2}. \tag{10}$$

III. Quaternionic Qubit Neural Network

A. Qubit(Quantum bit) and its representation by quaternion

In quantum computing, the concept of 'qubit' has been introduced as the counterpart of the classical concept of 'bit' in conventional computers. The two qubit states labeled as $|0\rangle$ and $|1\rangle$ correspond to the classical bits 0 and 1 respectively. The arbitrary qubit state $|\phi\rangle$ maintains a coherent superposition of states $|0\rangle$ and $|1\rangle$:

$$|\phi\rangle = a\,|0\rangle + b\,|1\rangle\,, \tag{11}$$

where a and b are complex numbers called probability amplitudes. This means, that the qubit state $|\phi\rangle$ collapses into either $|0\rangle$ state with probability $|a|^2$, or $|1\rangle$ state with probability $|b|^2$ with satisfying $|a|^2 + |b|^2 = 1$.

The qubit state $|\phi\rangle$ is described as

$$|\phi\rangle = \cos\theta\,|0\rangle + e^{i\psi}\sin\theta\,|1\rangle\,, \tag{12}$$

which is called Bloch-sphere representation. In this representation, the state of the quantum state can be represented as a point on Bloch sphere (Fig. 1) [7]. The complex-valued probability amplitudes a and b in Eq.(11) are defined as

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} e^{-i\varphi/2}\cos\theta/2 \\ e^{i\varphi/2}\sin\theta/2 \end{pmatrix}, \tag{13}$$

on the Bloch sphere. Then, the state of a quibit is described by using the real-parts and imaginary-parts of a and b,

$$\vec{q} = \Re(a) + \Im(a)\boldsymbol{i} + \Re(b)\boldsymbol{j} + \Im(b)\boldsymbol{k}. \tag{14}$$

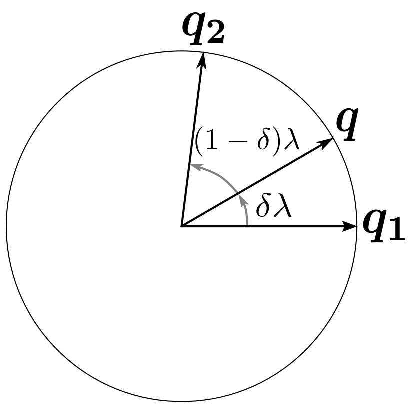

Fig. 2. Interpolation operation on Bloch sphere in CNOT gate

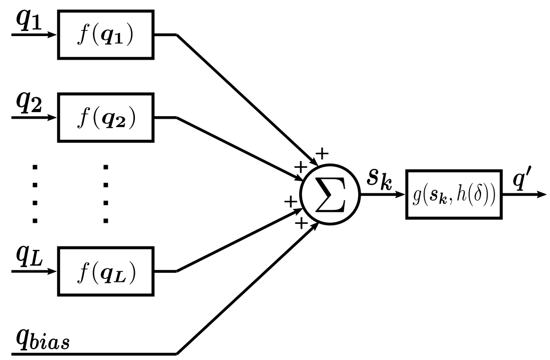

Fig. 3. Neuron model based on quaternionic rotation gates and CNOT gate

B. Operations by quantum gates

In quantum logic circuit, fundamental quantum gates are the single bit rotation gate and two-bit controlled NOT gate. These gates are universal in the sense that any quantum logic circuit can be constructed by their combinations.

A single bit rotation gate takes a quantum state as its input and outputs a rotated state in the complex plane. The function of this gate can be described using $\boldsymbol{q} = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$ as its input,

$$f(\boldsymbol{q}) = \boldsymbol{q}\boldsymbol{v} = \boldsymbol{q}\exp\left(-\boldsymbol{u}\omega/2\right), \tag{15}$$

$$\boldsymbol{u} = \boldsymbol{x}/|\boldsymbol{x}|, \tag{16}$$

where $\boldsymbol{x} = l\boldsymbol{i} + m\boldsymbol{j} + n\boldsymbol{k}$ is a pure imaginary quaternion. The rotation in this gate is represented by quaternion multiplication, i.e. the rotation is conducted by setting axis of rotation as $\boldsymbol{u}$ (which is a three-dimensional normalized vector $\boldsymbol{x}$) and rotation angle as ω.

A two-bit controlled NOT (CNOT) gate takes two quantum states as its input and produces two outputs: one of the input states and the exclusive OR-ed of two inputs. It is necessary to represent the inversion and non-inversion of the quantum state in order to describe this operation, thus a controlled input parameter δ is introduced:

$$g(\boldsymbol{q}, \delta) = \frac{\sin\left((1-\delta)\lambda\right)}{\sin\lambda}\boldsymbol{q} + \frac{\sin\left(\delta\lambda\right)}{\sin\lambda}\boldsymbol{q}_{\mathbf{not}}, \tag{17}$$

$$\boldsymbol{q}_{\mathbf{not}} = c + d\boldsymbol{i} + a\boldsymbol{j} + b\boldsymbol{k}, \tag{18}$$

$$\lambda = \arccos\left(\boldsymbol{q} \cdot \boldsymbol{q}_{\mathbf{not}}\right). \tag{19}$$

$\delta = 1$ and $\delta = 0$ corresponds to the inversion and non-inversion of the input quantum state, respectively. The output of this gate comes from $\boldsymbol{q}$ and $\boldsymbol{q}_{\mathbf{not}}$ linearly interpolated on the bloch sphere by δ (see Fig. 2).

C. Quaternionic qubit neuron and network models

The proposed neuron model is shown in Fig. 3, which is composed by rotation gates $f(\cdot)$ and a CNOT gate $g(\cdot, \cdot)$. Each of input signals q_is to the neuron is first rotated by each of rotation gate, and they and bias signal (q_{bias}) are accumulated. The accumulated signal s_k is transformed by a CNOT gate with the parameter $h(\delta)$, where the function h controls the degree of interpolation. ReLU function is used for the function h in this paper. The transformed signal is the output from the neuron.

A multilayered perceptron-type network is constructed by configuring neurons in each layer and connecting neurons between layers. In order to construct a learning scheme for this network, differential of the error E with respect to the connection weights $\boldsymbol{w}$s should be defined. We adopt the differential by quaternionic variable as quaternionic element-wise differential, such as

$$\frac{\partial E}{\partial \boldsymbol{q}} = \frac{\partial E}{\partial q_e} + \frac{\partial E}{\partial q_i}\boldsymbol{i} + \frac{\partial E}{\partial q_j}\boldsymbol{j} + \frac{\partial E}{\partial q_k}\boldsymbol{k}. \tag{20}$$

The partial differential $\partial E/\partial q_e$ is calculated as

$$\frac{\partial E}{\partial q_e} = \frac{\partial E}{\partial p_e}\frac{\partial p_e}{\partial q_e} + \frac{\partial E}{\partial p_i}\frac{\partial p_i}{\partial q_e} + \frac{\partial E}{\partial p_j}\frac{\partial p_j}{\partial q_e} + \frac{\partial E}{\partial p_k}\frac{\partial p_k}{\partial q_e}. \tag{21}$$

This rule of differential can be applied to calculate the differentials of rotation and CNOT gates.

IV. Experimental results

The performances of the proposed QQNN are explored in this section, with comparing with conventional (real-valued) neural network. Prediction of three-dimensional signals produced from a chaotic system, called Lorentz system, is used for evaluation of both networks.

A. Setup

The Lorentz system is a system that consists of three ordinary differential equations:

$$\frac{dx}{dt} = \sigma(y - x), \tag{22}$$

$$\frac{dy}{dt} = x(\rho - z) - y, \tag{23}$$

$$\frac{dz}{dt} = xy - \beta z, \tag{24}$$

where x, y, and z describes the state of the system with time t, and σ, ρ, and β are the parameters of the system.

This system exhibits chaotic behaviors for particular sets of parameters, for an example, when $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$ are set to the system, this system has two fixed attractors called Lorentz attractors. Signals for training and testing networks are generated by solving these differential equations by Euler method with the initial configuration $(x, y, z) = (1.0, 1.0, 1.0)$ and the time step $dt = 0.01$. A set of 3,000 three-dimensional time series is then obtained for $0 \leq t < 30$.

A three layered QQNN and real-valued NN are prepared for predicting the state of a system. The neurons for QQNN are 9, 16, and 3 for the input layer, hidden layer, and the output layer, respectively. The number of trainable parameters is 963. For real-valued NN, the neurons are 9, 75, and 3 for input, hidden, and output layers, respectively. The number of trainable parameters is 975. These networks are selected so the the number of trainable parameters for these networks become almost comparative.

For these networks, training and test data sets are prepared from the time series. For training networks, the time series for $0 \leq t < 20$ (2,000 samples) are used, and remaining series for $20 \leq t < 30$ (1,000 samples) are used for testing networks. Each component for the input is normalized in the range $[0, 1]$.

These networks are trained so that the prediction state at time $(t + \Delta t)$ is generated (predicted) from the system states at time t, $(t - 1)$, and $(t - 2)$ presented on the input of the network. The parameter Δt takes a integer value that determines how many time steps ahead should be predicted. The experiments are conducted for $\Delta t = 10, 20, \cdots, 100$. The number of iterations for training network is set to $1,000$ for both networks.

B. Results

After training a network with training data set, this network is to predict the states of Lorentz system by using test data set. From the predicted states, it is possible to compare this time-series states with the time-series data from the original Lorentz system. Figures 4 and 5 show the three-dimensional trajectories constructed from the output of the network (shown in red blobs) and the output of the Lorentz system (shown in black blobs) for real-valued network and the proposed network, respectively. In this case $\Delta t = 10$ is used, i.e., the networks should predict the states at 10 time step ahead from the latest three states. From the output by real-valued network (see Fig. 4), we see the outline of the trajectory from the real-valued network look like the original data, but several points of which difference between states can be found. On the other hand, the trajectory from the proposed QQNN is quite similar to the original one, meaning that the proposed network can acquire the functions of Lorentz system. In order to evaluate the reconstruction quality in a quantitative manner, we use the mean squared error (MSE) between output of Lorentz system and output of the network. The MSE values for the real-valued and the proposed QQNN are 1.64×10^{-5} and 9.51×10^{-7}, respectively. From these MSE values, it is shown that QQNN can simulate the Lorentz system better than real-valued network.

When the parameter Δt becomes larger, both networks degrade in predicting the states of Lorentz system, but the degree of degradation is different. Figures 6 and 7 show reconstructed trajectories from the test dataset by the real-valued network and the proposed network, respectively, in the case of $\Delta t = 30$. The real-valued network fails to acquire the functions of Lorentz system from the training dataset, but the proposed network maintain the capability of the reconstruction ability though the differences from the original data becomes large. It can be confirmed by calculating MSEs from the outputs of these networks, and MSEs for the real-valued and proposed QQNN are 3.69×10^{-4} and 2.64×10^{-5}, respectively.

We then evaluate the degree of reconstruction quantitatively for the real-valued and proposed networks. For the test dataset, the squared distances from the predicted states and original states (test dataset) can be calculated, and the minimum distance from these distances can also be identified. We have obtained the minimum distances for each of Δt for both networks. The changes of minimum distance with respect to Δt are shown in Fig. 8. It is observed that the minimum losses from the proposed network are always lower than those from the real-valued network. This means that the proposed networks can perform better in predicting time-series of a chaotic system.

V. Conclusion

In this paper, we have proposed a neural network model based on quantum information processing. The neuron in this model has four-dimensional state by using quaternionic representation, which is a hypercomplex number system; the state of a neuron is represented by a point on the Bloch sphere. The operation of neuron model adopts the operations in quaternion, which is equivalent to the operations by Pauli matrices. Multilayered perceptron-type network is composed from this neuron model and the learning algorithm is also formulated.

The performances of the proposed network are evaluated through a problem for predicting the time-series states from a chaotic system called Lorentz system. The results from numerical experiments show that the proposed network outperforms the conventional real-valued network.

There are several points to be considered for the proposed network. The activation functions in the proposed network are missing for neurons in the hidden layer, thus appropriate configuration for the activation function is necessary. In order to compose the networks for accepting image/test signals, it is important to extend the proposed network so that convolution and pooling functions are

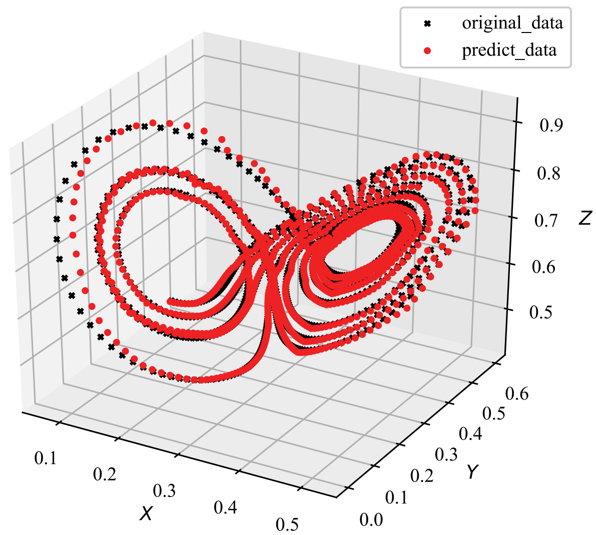

Fig. 4. Trajectory reconstructed from NN ($\Delta t = 10$)

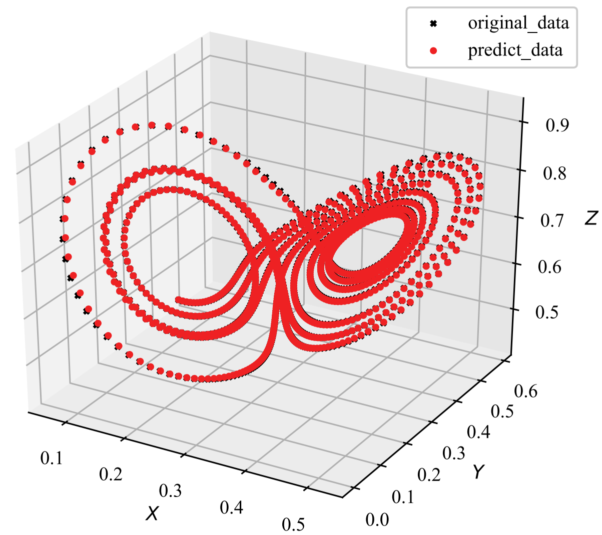

Fig. 5. Trajectory reconstructed from QQNN ($\Delta t = 10$)

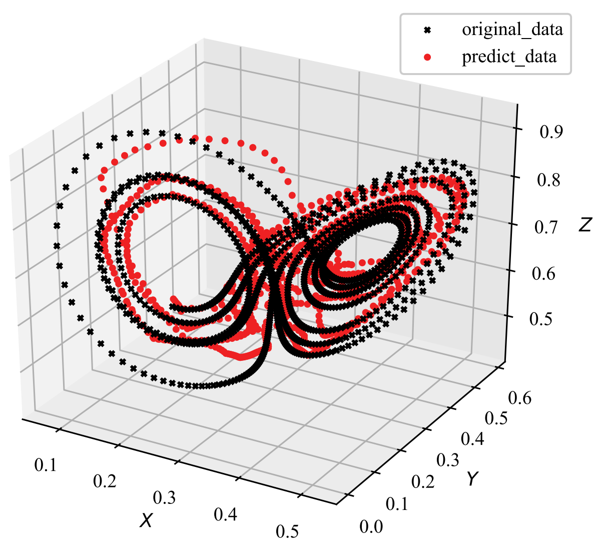

Fig. 6. Trajectory reconstructed from NN ($\Delta t = 30$)

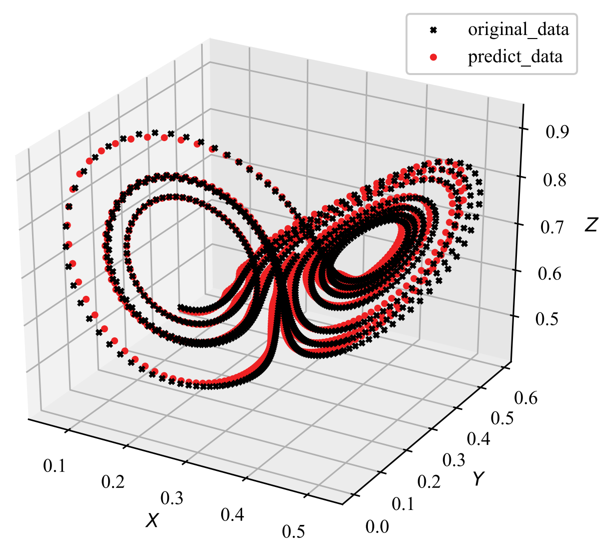

Fig. 7. Trajectory reconstructed from QQNN ($\Delta t = 30$)

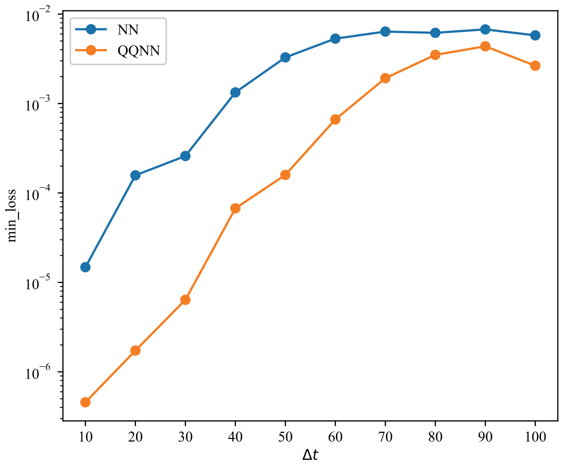

Fig. 8. Δt-dependence for minimum loss in learning networks

endowed. Also, the performance evaluations for various tasks are necessary. These remain for our future work.

References

[1] A. Manju and M. Nigam, "Applications of quantum inspired computational intelligence: a survey," Artificial Intelligence Review, vol. 42, no. 1, pp. 79—156, 2014.

[2] S. Kak, "On quantum neural computing," Information Sciences, vol. 83, no. 3, pp. 143–160, 1995.

[3] M. Peruš, "Neuro-quantum parallelism in brain-mind and computers," Informatica, vol. 20, no. 2, pp. 173–184, 1996.

[4] N. Matsui, M. Takai, and H. Nishimura, "A network model based on qubit-like neuron corresponding to quantum circuit," Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol. 83, no. 10, pp. 67–73, 2000.

[5] N. Kouda, N. Matsui, and H. Nishimura, "A multilayered feed-forward network based on qubit neuron model," Systems and Computers in Japan, vol. 35, no. 13, pp. 43–51, 2004.

[6] N. Kouda, N. Matsui, H. Nishimura, and F.Peper, "Qubit Neural Network and Its Learning Efficiency," Neural Computing and Applications, vol. 14, no. 2, pp. 114–121, 2005.

[7] K. B. Wharton and D. Koch, "Unit quaternions and the bloch sphere," Journal of Physics A: Mathematical and Theoretica, vol. 48, no. 23, p. 235302, 2015.

[8] W. R. Hamilton, Lectures on Quaternions. Dublin: Hodges and Smith, 1853.

[9] F. Zhang, "Quaternions and matrices of quaternions," Linear Algebra and its Applications, vol. 251, pp. 21–57, 1997.

[10] E. N. Lorenz, "Deterministic nonperiodic flow," Journal of the Atmospheric Sciences, vol. 20, no. 2, pp. 130–141, 1963.