Learning Rule for a Quantum Neural Network Inspired by Hebbian Learning

Yoshihiro OSAKABE†*,††, Member, Shigeo SATO†, Senior Member, Hisanaka AKIMA‡*,‡‡, Mitsunaga KINJO††, Members, and Masao SAKURABA†, Nonmember

SUMMARY Utilizing the enormous potential of quantum computers requires new and practical quantum algorithms. Motivated by the success of machine learning, we investigate the fusion of neural and quantum computing, and propose a learning method for a quantum neural network inspired by the Hebb rule. Based on an analogy between neuron–neuron interactions and qubit–qubit interactions, the proposed quantum learning rule successfully changes the coupling strengths between qubits according to training data. To evaluate the effectiveness and practical use of the method, we apply it to the memorization process of a neuro-inspired quantum associative memory model. Our numerical simulation results indicate that the proposed quantum versions of the Hebb and anti-Hebb rules improve the learning performance. Furthermore, we confirm that the probability of retrieving a target pattern from multiple learned patterns is sufficiently high.

key words: learning, quantum neural network, Hebb rule, Boltzmann machine, adiabatic quantum computation

1. Introduction

Quantum computing has attracted considerable academic interest owing to its inherent parallel computing characteristic achieved using superposition states. Quantum computers require reliable hardware and software. Implementing a quantum computer has become more realistic owing to the recent progress in nanotechnology, including gate models [1] and adiabatic quantum computation schemes [2], [3]. On the other hand, the development of new quantum algorithms is still necessary for practical applications because there exists only a small number of powerful quantum algorithms [4], [5], and many fields could benefit from quantum computing.

Machine learning is a rapidly growing field in computer engineering. In particular, deep learning, a recent machine learning technique, has motivated studies on broad range of applications [6], [7], such as classification, recognition, and artificial intelligence. These algorithms tend to be based on models with parameters that can be determined from a large amount of training data. This means that learning from data sets is identical to acquiring new algorithms automatically, though the success rate of training depends on the learning rule. Therefore, the possibility of using machine learning techniques for quantum computing has been considered theoretically for both gate models [8]–[10] and quantum annealing schemes [11]–[17]. In particular, Amin et al. recently updated this classical model to a quantum mechanical model [17], inspired by the Boltzmann machine (BM). Here, the authors proposed a new machine learning method based on the quantum Boltzmann distribution of a transverse-field Ising Hamiltonian $H_{\text{IS}}$, which is inspired by the stochastic gradient descent in training a classical BM. Their learning method minimizes the Kullback–Leibler (KL) divergence, which corresponds to the difference between the actual and the desired probability distributions, using quantum sampling. They emphasize that the quantum mechanical feature is not only used to facilitate the quantum sampling process, but is also used in the training process of their quantum Boltzmann machine (QBM).

We employ another simple approach, based on a neuro-inspired Hamiltonian $H_{\text{NI}}$ and qubit–qubit interactions, which differs from the $H_{\text{IS}}$. Using an analogy between a neural network and a quantum neural network (QNN), as suggested by Sato et al. [18], [19], we can determine the qubit–qubit interactions of a QNN from the neuron–neuron interactions of a neural network. In short, we convert a weight matrix for a neural network into an $H_{\text{NI}}$ for a QNN. Therefore, we assume that we can introduce neural functions into quantum computing using this conversion method. Recently, a new model of quantum associative memory (Q-AM) was proposed [20], [21] with a QNN. The model’s memorization is realized using an $H_{\text{NI}}$ converted from an autocorrelation matrix of target patterns by employing an adiabatic Hamiltonian evolution [22]. The results indicate that the memorization and retrieval processes of the Q-AM work properly, but that the retrieval performance degrades if the number of target patterns increases. To provide quantum computers with a practical neural function, an iterative learning method, such as that in a classical BM, must be developed. Thus, in this paper we propose learning rules for a QNN by extending the preliminary result on quantum Hebbian learning [23], and then apply these rules to the Q-AM. Inspired by the Hebb rule [24] for a neural network, we update the qubit–qubit interactions of a QNN according to the neuron–neuron interactions of a neural network. In
other words, the learning function is realized using an $H_{NI}$, iteratively converted from the weight matrix, and updated repeatedly according to the probability distributions of the QNN states. Because an $H_{NI}$ does not include diagonal elements, it differs from an $H_S$. Therefore, studying an $H_{NI}$ expands the range of materials and devices that can be used to design quantum computing architectures. For example, the Hamiltonian for a $1/2$-spin system (i.e., a Heisenberg Hamiltonian) has both diagonal and non-diagonal elements. Thus, the conversion method of the proposed model is unique, which is the significant difference between this and previous quantum learning models.

In the following, we first summarize the basic theory used in the quantum learning scheme. Next, we propose the quantum versions of the Hebb and anti-Hebb rules for a QNN, and discuss their basic learning properties. Then, we show the effectiveness of the proposed learning rules for the neuro-inspired Q-AM as a practical application.

2. Basic Theory: Neuromorphic Adiabatic Quantum Computation

In this section, we summarize the adiabatic Hamiltonian evolution, which is utilized in the dynamics of a QNN. Farhi et al. proposed an adiabatic quantum computation (AQC) algorithm [22] utilizing an adiabatic Hamiltonian evolution, defined as

$$H(\tau) = (1 - A(\tau)) H_I + A(\tau) H_F,$$

(1)

where $A(\tau)$ increases from 0 to 1 as $\tau$ goes from 0 to its maximum $\tau_{\text{Max}}$. Here, $H_I$ and $H_F$ are the initial and final Hamiltonians, respectively. To solve problems, $H_F$ is chosen according to a desired operation or a target problem. Sato et al. investigated a neuro-inspired approach to provide a proper $H_F$, which they call a neuro-morphic adiabatic quantum computation (NAC) [18, 19, 25]. Inspired by the analogy between neuron–neuron interactions and qubit–qubit interactions, we convert a weight matrix of a neural network into a Hamiltonian of a QNN. The conversion method is expressed as

$$H_{NI} = \lambda I^{(2^N)} + \sum_{i,j} \sum_{\alpha,\beta} |w_{i\alpha}|^2 (H_{ij}(w_{i\alpha}) + H_{ij}^*(w_{i\alpha})).$$

(2)

where $\lambda$ is a constant and $I^{(2^N)}$ is the $2^N \times 2^N$ identity matrix. Then, $H_{ij}(w_{i\alpha})$, which denotes the interaction between the $i$-th and $j$-th qubits, is given as

$$H_{ij}(w_{i\alpha}) = \left\{
\begin{array}{ll}
(I^{(2^i)} \otimes \cdots \otimes a_{\alpha} \otimes \cdots \otimes a_{\beta} \otimes \cdots \otimes I^{(2^j)}) & (w_{ij} > 0) \\
O^{(2^i)} & (w_{ij} = 0),
\end{array}
\right.$$

(3)

where

$$a = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad a^\dagger = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

(4)

respectively, and $O^{(2)}$ is the $2 \times 2$ zero matrix. Table 1 describes the example of Eqs. (2)–(4) in the case of $N = 2, 3$. Previous works have applied this method to optimization problems using the weight matrices of a Hopfield network [18, 19, 25]. The results indicate that both excitatory ($w_{ij} > 0$) and inhibitory ($w_{ij} < 0$) interactions can be realized effectively. At the same time, these previous results revealed that the arbitrary constant $\lambda$ doesn’t affect the performance, thus we set $\lambda = 0$ in the following simulations, which means $H_{NI}$ doesn’t have diagonal elements.

3. Quantum Learning for Quantum Associative Memory

To provide quantum computers with a practical learning function, an iterative learning method is required. In this section, we propose a novel iterative learning method for a QNN, and then confirm its learning effectiveness by applying it to neuro-inspired Q-AM model [21].

3.1 Quantum Learning Rule Inspired by the Hebb Rule

The Boltzmann machine (BM) is a classical neural network model utilizing stochastic neurons. In the learning process, the parameters of the model can be learned from a large amount of training data. The learning methods, called Hebb and anti-Hebb rules, employed in BM learning [24] specify how much the weight of the connection between the $i$-th and $j$-th neurons, denoted by $w_{ij}$, should be increased or decreased in proportion to the synchronous firing rates of the neurons. In this way, the network parameters ($w_{ij}$) are tuned

| $N$ | $W$ | $H_{NI}$ |
|-----|-----|-----------|
| (i) | 2   | $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ | $\begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{pmatrix}$ |
| (ii) | 2 | $\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$ | $\begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{pmatrix}$ |
| (iii) | 3 | $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}$ | $\begin{pmatrix} \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 1 & 0 & 0 & 0 \\ 0 & 1 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 1 \\ 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda \end{pmatrix}$ |
| (iv) | 3 | $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}$ | $\begin{pmatrix} \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 1 & 0 & 0 & 0 \\ 0 & 1 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 1 \\ 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda \end{pmatrix}$ |
Fourth, the probability distribution where ηW leads to avoid the divergence of W during a learning session. If the learning is successful, the obtained probability distribution P is for a given state v resembles the probability distribution of the training data P\text{data}.

By imitating the Hebb rule, we can tune the coupling strengths between qubits of a QNN to store multiple binary patterns. Let us assume that an N-qubit QNN memorizes a set S of M patterns. The pattern is given as an N-dimensional random vector, the elements of which take random values of ±1. The vector is denoted by ξμ = (xμ,1, xμ,2, · · · , xμ,N), where μ is a pattern index in S (1 ≤ μ ≤ M). The purpose of learning is to obtain an optimized weight matrix, producing a Hamiltonian with a ground state that realizes a superposition state of ξμs. The learning method of the QNN is given as follows (see Fig. 1).

First, an input data set S composed of M binary patterns is prepared. Second, a weight matrix W(t), where t is the number of learning iterations, is generated or updated. Third, a Hamiltonian for learning H_L(t) is converted from W(t). Fourth, the probability distribution P_L(t) of the qubit state v is calculated by employing H_L(t) as a final Hamiltonian for the NAQC [18], [19], [25]. Fifth, if t is less than T, which is the maximum of t, W(t) is updated according to the temporal P_L(t). After sufficient iterations, the target patterns are realized as a superposition state in the ground state of H_L(T).

In the following, the method is explained in detail in mathematical form. The weight matrix W(t) used to generate H_L(t) is defined as

\[ W(t) = ηW(t - 1) + ΔW(t), \]

(5)

\[ ΔW(t) = W_H + εW_{AH}(t), \]

(6)

where η is a decay factor. Usually, η less than 1 is used to avoid the divergence of W(t). However, we set η to 1, which leads to W(t - 1) - W(t) = ΔW(t), in all numerical calculations in order to study the influence of ΔW clearly. The second term ΔW is the main component used to update W(t), and is described in Eq. (6). Here, W_H and W_{AH} correspond to the Hebbian and anti-Hebbian learning in the clamped and unclamped phases of the BM learning, respectively. Then, ε is the ratio of the anti-Hebbian learning rate to the Hebbian learning rate. The elements in W_H and W_{AH} indicate the coupling strengths between the i-th and j-th qubits, respectively, and are defined as

\[ w_{ijH} = \begin{cases} + \sum_{v=0}^{2^{N-1}} |P^v_i(t)| & \text{if } x_i^v = x_j^v, \\ - \sum_{v=0}^{2^{N-1}} |P^v_i(t)| & \text{if } x_i^v \neq x_j^v, \end{cases} \]

(7)

\[ w_{ijAH}(t) = \begin{cases} - \sum_{v=0}^{2^{N-1}} |P^v_i(t)| & \text{if } x_i^v = x_j^v, \\ + \sum_{v=0}^{2^{N-1}} |P^v_i(t)| & \text{if } x_i^v \neq x_j^v, \end{cases} \]

(8)

where P^v is the desired and temporal probability distributions, respectively, of the v-th qubit state. More precisely, a coupling strength decreases when two qubits are positively correlated, and increases when they are negatively correlated. In conventional neural networks, the anti-Hebb rule allows a network to remove inappropriate correlations. Therefore, we expect that W_{AH}(t) will improve the learning performance of a QNN.

In the following experiments, we use an autocorrelation matrix W_{AC} of target patterns as W_{H}, defined as

\[ W_{AC} = \frac{1}{N} \left( \sum_{\mu=1}^{M} (\xi^\mu)^T (I^{N}) \right), \]

(9)

where I^{N} is an N × N identity matrix. To make the calculation easier, the value of P^v_i(t) is rounded off to six significant figures.

3.2 Basic Property of Quantum Learning

As a first step, to investigate how the coupling strengths are updated using the learning rules expressed as Eqs. (5) and (6), we visualize two-dimensional updates of w_{ij}. We train the QNN using a weight matrix with elements that are fixed to arbitrary constants, except for two selected elements. In other words, only two w_{ij}s are allowed to change during the learning process. In this simulation, the network is composed of nine qubits, there are three target patterns, and we use W_{AC} instead of W_H. The maximum number of iterations for the learning process is set to T = 500.

The numerical simulation results are shown in Fig. 2, where w_{51} and w_{94} are chosen for the variables, with initial values of zero. The background corresponds to the average probability of the target patterns calculated from H_L(t). The white arrows denote the actual changes of w_{51} and w_{94} from their initial values to final values through learning, along with t. In other words, w_{51} and w_{94} change from zero to their proper values via the learning session, until W_H and εW_{AH} counterbalance each other because η = 1.

In the case of ε = 0, shown in Fig. 2(a), the trace corresponds to the learning process caused only by the autocorrelation matrix W_{AC}. In fact, the target patterns ξ^μ in this case
According to Eq. (9), $w_{51}$ and $w_{94}$ should be positive and negative, respectively. Thus, the trace of $\varepsilon = 0$ goes toward in the right in an oblique downward direction, but the high probability region is not placed in this direction. In the cases of non-zero $\varepsilon$, shown in Fig. 2 (b), (c), and (d), the traces go toward the higher probability region, thanks to anti-Hebbian learning. Therefore, the effectiveness of the anti-Hebbian learning is confirmed. The final state of the trace with $\varepsilon = 9$ is slightly away from those with $\varepsilon = 3$ and 6. These results indicate that the final value of $w_{ij}$ depends on $\varepsilon$, and there exists an appropriate $\varepsilon$ for better learning results.

3.3 Learning Performance

In the previous subsection, we found that the proposed learning rule works properly and an inappropriate $\varepsilon$ degrades the learning performance. Next, we apply the rules to the Q-AM in order to confirm its possibility for practical applications. It has been shown that a classical associative memory (C-AM) fails to learn if the target patterns have strong correlations [26]. To investigate whether the same behavior is evident in the proposed quantum learning, we perform several numerical simulations.

In this experiment, three target pattern sets $\mathcal{S}$ are prepared, where each $\mathcal{S}$ is composed of three random target patterns with average hamming distances $\langle D_{H} \rangle$ of 6.00, 4.00, and 2.67. At this time, we do not fix any $w_{ij}$, and investigate the time development of the average probability of target patterns over $\mu$, given as

$$
\langle P_{L}(t) \rangle_{\mu} = \frac{1}{M} \sum_{\mu=1}^{M} P_{L}^{\mu}(t),
$$

where $P_{L}^{\mu}(t)$ is the probability of a target pattern $\mu$, calculated from the ground states of $H_{L}(t)$.

In the case of $\langle D_{H} \rangle = 6.00$, shown in Fig. 3 (a), the anti-Hebbian learning works correctly. On the other hand, in the cases of $\langle D_{H} \rangle = 4.00$ and 2.67, shown in Fig. 3 (b) and (c), respectively, $\langle P_{L}^{\mu}(t) \rangle_{\mu}$ oscillates when $\varepsilon$ becomes large. Comparing these figures, we find that the threshold value
Fig. 3 Time development of the average probability of learning a single target pattern over μ, \( \langle P^\mu L(t) \rangle _\mu \), with average hamming distances of (a) \( \langle D_H \rangle = 6.00 \), (b) \( \langle D_H \rangle = 4.00 \), and (c) \( \langle D_H \rangle = 2.67 \).

of \( \varepsilon \) that triggers the \( \langle P^\mu L(t) \rangle _\mu \) oscillation decreases as \( \langle D_H \rangle \) decreases. Such an oscillation is not observed in the preliminary results shown in Sect. 3.2. However, fortunately, even when \( \langle D_H \rangle \) is small, the small \( \varepsilon \) successfully avoids the oscillation of \( \langle P^\mu L(t) \rangle _\mu \). In successful learning cases, \( \langle P^\mu L(t) \rangle _\mu \) actually converges to a proper value immediately. Therefore, we can conclude that a large \( \varepsilon \) improves the learning performance, unless it induces an \( \langle P^\mu L(t) \rangle _\mu \) oscillation, while a small \( \varepsilon \) has the advantage of avoiding the oscillation.

Next, we evaluate how many patterns can be learned by the proposed method and study the total probability of all target patterns for each \( S \), denoted as \( P^S \) total \( (t) = \sum_{\mu=1}^{M} P^\mu L(t) \), calculated from the ground states of \( H_L(t) \). We prepare three random target pattern sets \( S \) for each \( M \), which have average hamming distances in the range of \( 3.00 \leq \langle D_H \rangle \leq 5.00 \), and train the QNN 500 times. Because \( \langle P^\mu L(t) \rangle _\mu \) oscillates in some cases, as shown in Fig. 3, we calculate the average time of \( P^S \) total \( (t) \) for \( 50 \leq t \leq 500 \), and obtain its statistical average over \( S \), which is expressed as

\[
\langle P_{\text{total}} \rangle _{t,S} = \frac{1}{450} \sum_{S} \sum_{t=50}^{500} P^S \text{total} (t).
\]  

The results shown in Fig. 4 indicate that a larger \( \varepsilon \) keeps the learning performance high, even when \( M \) increases. Thus, anti-Hebbian learning is more important for large \( M \) to remove inappropriate correlations. Overall, the probability differences among \( M \) for each \( \varepsilon \) are quite small. Thus, we conclude that the learning performance does not degrade when \( M \) increases in the scope of our experiments.

We should compare these results with previous findings using a Q-AM [21]. In the previous model, we used a Hamiltonian converted simply from \( W_{\text{AC}} \) for memorization, which means the result of \( \varepsilon = 0 \) corresponds to those of previous studies on Q-AM. With sufficiently large \( \varepsilon \), \( \langle P_{\text{total}} \rangle _{t,S} \) does not decrease when \( M \) increases. Therefore, anti-Hebbian learning certainly contributes to the performance improvement in memorization.

4. Retrieval Performance of Quantum Associative Memory

4.1 Summary of Retrieval Operations

A fully connected neural network has an association func-
tion [27], [28], which means the network state converges to a state corresponding to one of the memorized patterns, according to the initial state given as the key input. On the other hand, the information of an initial state vanishes during a Hamiltonian evolution, and the QNN does not converge to a specific state unless some additional treatment is applied.

In a previous study [21], the authors proposed applying a Hamiltonian \( H_{\text{fix}} \) in addition to a memorization Hamiltonian \( H_{\text{mem}} \). Here, \( H_{\text{fix}} \) is based on the idea of an external local magnetic field to a qubit, which partially fixes some qubit states. The Hamiltonian evolution for retrieval is expressed as

\[
H(t) = \left(1 - \frac{t}{T}\right)H_I + \frac{t}{T} (H_{\text{mem}} + A_{\text{fix}} H_{\text{fix}}),
\]

where \( H_{\text{mem}} \) is converted from \( W_{\text{AC}} \), according to Eq. (2). In this study, we follow the same retrieval operations and substitute a learned Hamiltonian \( H_{\text{L}}(T) \) for \( H_{\text{mem}} \). Here, \( A_{\text{fix}} \) is a scaling factor of \( H_{\text{fix}} \), where \( H_{\text{fix}} \) is given as

\[
H_{\text{fix}} = (\kappa_1 \otimes I^{(2)} \otimes \cdots \otimes I^{(2)}) + (I^{(2)} \otimes \kappa_2 \otimes \cdots \otimes I^{(2)}) + \cdots + (I^{(2)} \otimes I^{(2)} \otimes \cdots \otimes \kappa_N).
\]

Then, \( \kappa_i \) is defined as

\[
\kappa_i = \begin{cases} +\sigma^z & \text{(if } x_i \text{ is } +1), \\ I^{(2)} & \text{(if } x_i \text{ is unfixed)}, \\ -\sigma^z & \text{(if } x_i \text{ is } -1), \end{cases}
\]

where \( \sigma^z \) is the z-component of the Pauli spin matrices. When we add \( H_{\text{fix}} \) during the Hamiltonian evolution, we can retrieve a target pattern according to a key input \( \xi_{\text{key}} \). For further information on the operations and properties, please refer to the work cited in [21].

4.2 Numerical Results of the Retrieval Process

We examined the retrieval process to evaluate whether the Hamiltonian obtained by learning can be utilized for associative memory recall. In the simulations, the number of unfixed qubits in \( \xi_{\text{key}} \) is one, which means \( H_{\text{fix}} \) provides the Q-AM with a single unknown qubit. For ease of calculation, we obtain the probability distribution of the QNN state by calculating the ground states of \( (H_{\text{L}}(T) + A_{\text{fix}} H_{\text{fix}}) \) rather than by simulating the NAQC described in Eq. (15). If the ground states are degenerated, we average their probabilities of target patterns.

The averaged probability over \( S \) of retrieving a correct target pattern \( \mu \), denoted as \( \langle P_{\text{ret}} \rangle_\mu \), is shown as a function of \( \varepsilon \) in Fig. 5. Although \( \langle P_{\text{ret}} \rangle_\mu \) becomes high if \( \varepsilon \) becomes large, the difference of the retrieval performance is not significant. Overall, \( \langle P_{\text{ret}} \rangle_\mu \) worsens as \( M \) increases. This tendency is similar to the feature of a conventional neural network, and is understandable. However, a sufficiently large \( A_{\text{fix}} \) realizes high retrieval performance. Thus, a target pattern can be retrieved from multiple candidates realized in the ground state of \( H_{\text{L}}(T) \).

Next, we study how the improvement of \( H_{\text{L}}(t) \) influences \( \langle P_{\text{ret}} \rangle_\mu \). As shown in Fig. 6, we again plot \( \langle P_{\text{ret}} \rangle_\mu \) as a function of \( \langle P_{\text{single}} \rangle_\mu = \langle P_{\text{total}} \rangle_\mu / M \), which is the average probability of all single target patterns. This indicates that a larger \( \langle P_{\text{single}} \rangle_\mu \) yields a larger \( \langle P_{\text{ret}} \rangle_\mu \), that is, the \( H_{\text{L}}(t) \) that
Fig. 6  The retrieval probabilities of a target pattern averaged over μ, \( \langle P_{\text{ret}}^\mu \rangle \), as a function of the average probability of all single target patterns, \( \langle P_{\text{single}} \rangle \).

Fig. 7  The comparison of the retrieval probabilities \( \langle P_{\text{ret}}^\mu \rangle \) of the quantum associative memory (Q-AM) models with and without learning and a classical associative memory (C-AM) model.

4.3 Discussion

The results we have seen so far are for very primary problems, and the behavior of a QNN and the proposed learning method have not been prescribed via analytical procedures. It is difficult to judge whether the good performance shown here is based on the nature of the proposed method itself, or whether it is artificially inflated by the choice of example problems. Therefore, in this section, we first check the difference between the learning methods of classical neural network (CNN) and QNN. After that, in order to evaluate the performance of Q-AM more correctly, we compare the retrieval performance of both Q-AM and classical associative memory (C-AM), because it seems that such associative memory recall task is rather advantageous for C-AM.

Now, let us begin discussion from the comparison of learning methods of CNN and QNN. The learning process of the classical BM minimizes the difference between \( P^e \) and \( P^\text{data} \), called Kullback–Leibler (KL) divergence, using the steepest descent method. We use the BM as the basis for the model of quantum learning for a QNN, and imitate the concept of clamped and unclamped phases in the BM. Thus, the proposed quantum Hebb and anti-Hebb rules are expected to serve the same function. Because \( H_{\text{NI}} \) has only non-diagonal elements as explained in the Sect. 2, it is inevitable that a QNN will learn both desired and undesired patterns. Therefore, it is not always guaranteed that the proposed learning rules minimizes the KL divergence. In addition, the weight matrix reflects the probability distribution calculated from the ground states of \( H_{\text{NI}} \) during the learning process. It means that the quantum mechanical process interrupts the classical weight matrix updating process. We assume that this operation also introduces non-trivial dynamics into quantum learning. Therefore, our proposed learning method is not completely compatible with the classical BM learning method. However, the proposed learning method certainly works well. As seen in the previous experimental results, it shows better performance compared with the case of the Hebb rule only.

To evaluate its performance in detail, the Q-AM model with learning should be compared with the other two associative memory models; the original Q-AM model without learning [21], and a C-AM model with the conventional Hebb learning. Note that the Hamiltonian used in the original Q-AM model is just converted from the autocorrelation matrix of the training data, in short the model doesn’t have the iterative learning process. The retrieval method is the same as the Q-AM model with learning, which is explained in this paper. We also trained a 9-neuron fully connected network with the conventional Hebb rule to perform as C-AM. Because this network is understood as a Hopfield model, the synaptic weights can be calculated[27],[28]. For retrieval, we give the target pattern as an initial state of the Hopfield network, which indicates that this retrieving task is considerably advantageous for C-AM. The neuron state is updated 100 times for retrieving and the network state reaches a stationary state. The retrieval probability \( \langle P_{\text{ret}}^\mu \rangle \) of C-AM is obtained as the average over 100 trials. In Fig. 7, the solid line with circles indicates the performance of the Q-AM with the proposed learning method, and the dashed line with squares denotes the performance of the Q-AM without learning, where the horizontal axis is the load parameter \( M/N \). This result supports the efficiency of the proposed learning method. The dotted line with triangles is the retrieval performance of C-AM, and it declines and is lower than Q-AM performance when \( M/N > 0.5 \). These experimental results certainly demonstrate that Q-AM is su-
perior to C-AM in retrieval performance, regardless of the usage of the learning method.

5. Conclusion

Motivated by success of classic neural network learning functions, we investigated the possibility of training a quantum neural network (QNN). We have proposed quantum versions of the Hebb and anti-Hebb rules to update the interaction strengths between qubits. Because the Hamiltonian of a QNN is converted from a weight matrix, learning is realized with a weight matrix that changes along with the temporal probability distribution of the QNN states. More precisely, Hebbian learning reinforces the interactions given by the autocorrelations of the training patterns. On the other hand, anti-Hebbian learning weakens undesired interactions, following a certain probability distribution. To evaluate its learning performance, the proposed learning rule was applied to realize a neuro-inspired quantum associative memory (Q-AM). As a result, we confirmed that the proposed learning method works properly, and that desired patterns are memorized successfully. However, the performance depends on \( \epsilon \), which is the ratio of the anti-Hebbian learning rate to the Hebbian learning rate, and an appropriate value has to be assigned to this variable to achieve the best results. We also examined the retrieval performance of the obtained Hamiltonian \( H_{H_1(T)} \). The results indicate that the Q-AM retrieves a target pattern from multiple learned patterns if a sufficiently large \( H_{fix} \) is applied. Thus, the learning and associative retrieval functions are realized successfully for a QNN using neuro-inspired Hamiltonians operated by adiabatic Hamiltonian evolution. Compared with the previous Q-AM model [21], anti-Hebbian learning certainly improves the learning performance of the proposed model. In addition, the comparison with a classical associative memory (C-AM) model indicates that the Q-AM model with and without the learning method is definitely superior to the C-AM model though we have to pay attention to the difference of applying associative key inputs. In the numerical experiments performed here, we comprehensively calculate all possible combinations of training patterns and adopt their average values as the performance. However, a large-scale simulation would be required for further understanding since the sizes of the simulations in this study are limited.

Finally, we should mention the hardware implementation of the proposed quantum learning method. We need architectures that enable us to tune the coupling interactions between any two arbitrary qubits. We assume that such an architecture can be constructed using simple devices, and the quantum learning system can be modified according to the characteristics of the materials or devices. Thus, the use of a neuro-inspired Hamiltonian for QNN operation provides new opportunities for the development of quantum computing architectures.

Acknowledgments

This study was supported in part by the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University; JSPS KAKENHI (Grant Number 17K18864); and JST CREST (Grant Number JPMJCR19K3), Japan.

References

[1] IBM, “IBM Quantum Experience.” http://research.ibm.com/ibm-q/
[2] M. Johnson, M. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. Berkley, J. Johansson, P. Bunyk, E.M. Chapple, C. Enderud, J.P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M.C. Thom, E. Tolkacheva, C.J.S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, “Quantum annealing with manufactured spins,” Nature, vol.473, no.7346, pp.194–198, 2011.
[3] C.C. McGeoch and C. Wang, “Experimental evaluation of an adiabatic quantum system for combinatorial optimization,” Proc. ACM International Conference on Computing Frontiers, p.23, ACM, 2013.
[4] P.W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM Review, vol.41, no.2, pp.303–332, 1999.
[5] L.K. Grover, “A fast quantum mechanical algorithm for database search,” Proc. twenty-eighth annual ACM symposium on Theory of computing, pp.212–219, ACM, 1996.
[6] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Networks, vol.61, pp.85–117, 2015.
[7] M.I. Jordan and T.M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” Science, vol.349, no.6245, pp.255–260, 2015.
[8] S. Lloyd, M. Mohseni, and P. Rebentrost, “Quantum algorithms for supervised and unsupervised machine learning,” arXiv preprint arXiv:1307.0411, 2013.
[9] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” Physical review letters, vol.113, no.13, p.130503, 2014.
[10] N. Wiebe, A. Kapoor, and K.M. Svore, “Quantum deep learning,” arXiv preprint arXiv:1412.3489, 2014.
[11] H. Neven, V.S. Denchev, G. Rose, and W.G. Macready, “Training a binary classifier with the quantum adiabatic algorithm,” arXiv preprint arXiv:0811.0416, 2008.
[12] H. Neven, V.S. Denchev, G. Rose, and W.G. Macready, “Training a large scale classifier with the quantum adiabatic algorithm,” arXiv preprint arXiv:0912.0779, 2009.
[13] V.S. Denchev, N. Ding, S. Vishwanathan, and H. Neven, “Robust classification with adiabatic quantum optimization,” arXiv preprint arXiv:1205.1148, 2012.
[14] K.L. Pudenz and D.A. Lidar, “Quantum adiabatic machine learning,” Quantum information processing, vol.12, no.5, pp.2027–2070, 2013.
[15] R. Babbush, V. Denchev, N. Ding, S. Isakov, and H. Neven, “Construction of non-convex polynomial loss functions for training a binary classifier with quantum annealing,” arXiv preprint arXiv:1406.4203, 2014.
[16] S.H. Adachi and M.P. Henderson, “Application of quantum annealing to training of deep neural networks,” arXiv preprint arXiv:1510.06356, 2015.
[17] M.H. Amin, E. Andriyash, J. Rolfe, B. Kalchtytskyy, and R. Melko, “Quantum boltzmann machine,” arXiv preprint arXiv:1601.02036, 2016.
[18] S. Sato, M. Kinjo, and K. Nakajima, “An approach for quantum computing using adiabatic evolution algorithm,” Japanese journal of
applied physics, vol.42, no.11R, pp.7169–7173, 2003.

[19] S. Sato, M. Kinjo, O. Takahashi, Y. Nakamiya, and K. Nakajima, “A study on neuromorphic quantum computation,” Proc. 2004 IEEE International Joint Conference on Neural Networks, 2004, pp.3251–3256, IEEE, 2004.

[20] Y. Osakabe, S. Sato, H. Akima, M. Sakuraba, and M. Kinjo, “Neuro-inspired quantum associative memory using adiabatic hamiltonian evolution,” 2017 International Joint Conference on Neural Networks (IJCNN), pp.803–807, IEEE, 2017.

[21] Y. Osakabe, S. Sato, H. Akima, M. Kinjo, and M. Sakuraba, “Quantum associative memory with quantum neural network via adiabatic Hamiltonian evolution,” IEICE Trans. Inf. & Syst., vol.E100-D, no.11, pp.2683–2689, 2017.

[22] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, “A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem,” Science, vol.292, no.5516, pp.472–475, 2001.

[23] M. Kinjo, S. Sato, and K. Nakajima, “A study on learning with a quantum neural network,” The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp.203–206, 2006.

[24] D.O. Hebb, The organization of behavior: A neuropsychological theory, New York: Wiley, 1949.

[25] M. Kinjo, S. Sato, Y. Nakamiya, and K. Nakajima, “Neuromorphic quantum computation with energy dissipation,” Physical Review A, vol.72, no.5, p.052328, 2005.

[26] W. Kinzel, “Learning and pattern recognition in spin glass models,” Zeitschrift für Physik B Condensed Matter, vol.60, no.2, pp.205–213, June 1985.

[27] J.J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” Proc. national academy of sciences, vol.79, no.8, pp.2554–2558, 1982.

[28] S. Amari and K. Maginu, “Statistical neurodynamics of associative memory,” Neural Networks, vol.1, no.1, pp.63–73, 1988.

Yoshihiro Osakabe received the B.E., M.E. and Ph.D. degrees from Tohoku University, Sendai, Japan, in 2013, 2015, and 2018, respectively. From April 2018, he is with Hitachi Ltd. Research and Development Group, Tokyo, Japan, where he works on artificial intelligence and knowledge discovery, especially for materials informatics and geospatial informatics. His current research interests include emergence of intelligence and non-von Neumann computing.

Shigeo Sato received the B.S., M.S., and Ph.D. degrees from Tohoku University, Sendai, Japan, in 1989, 1991, and 1994, respectively. In 1996, he joined the Research Institute of Electrical Communication, Tohoku University. His research interests are in LSI implementation and application of neural networks.

Hisanao Akima received the B.E., M.S., and Ph.D. degrees from Tohoku University, Sendai, Japan, in 2000, 2002, and 2006, respectively. From April 2006 to March 2013, he was with Hitachi, Ltd., Central Research Laboratory, Tokyo, Japan, where he worked on spherical aberration corrector for electron microscopes. From April 2013 to March 2018, he was an assistant professor in the Research Institute of Electrical Communication, Tohoku University. Currently, he is a senior researcher in Fujitsu Laboratory Ltd. His current research interests include knowledge processing and non von Neumann computing.

Mitsunaga Kinjo received B.E. degree in Electrical and Information Engineering from University of the Ryukyus, Okinawa, Japan in 1996. And he received M.E. degree in Electrical Engineering and Ph.D. degree in Information Science from Tohoku University, Sendai, Japan in 1998 and 2001, respectively. From April 2001 to September 2003, he was a researcher in the PRESTO program of the Japan Science and Technology Corporation (JST) conducted by Dr. Shigeo SATO at Tohoku University. From October 2003 to September 2006, he was a research fellow of the Research Institute of Electrical Communication (RIEC), Tohoku University. He worked on the development of quantum computation algorithm and its hardware implementation. Currently he is an associate professor in the Faculty of Engineering, University of the Ryukyus, since October 2008. His research interests are in quantum computation and its hardware implementation.

Masao Sakuraba received the B.E., M.E. and Ph.D. degrees from Tohoku University in 1990, 1992, and 1995, respectively. There, he has been engaged in the development of atomic layer growth control of germanium and silicon and research on surface reaction mechanisms. He became a Research Associate and an Associate Professor of Tohoku University in 1995 and 2002, respectively. Now, in the Laboratory for Nanoelectronics and Spintronics of Research Institute of Electrical Communication, he is actively involved in the development of low-damage plasma chemical-vapor deposition processing for group IV quantum heterointegration.