Gradient Descent Learning for Hyperbolic Hopfield Associative Memory*

Masayuki TSUJI†‡, Teijiro ISOKAWA†, Masaki KOBAYASHI§, Nobuyuki MATSUI† and Naotake KAMIURA†

This paper proposes a scheme for embedding patterns onto the Hyperbolic-valued Hopfield Neural Networks (HHNNs). This scheme is based on gradient descent learning (GDL), in which the connection weights among neurons are gradually modified by iterative applications of patterns to be embedded. The performances of the proposed scheme are evaluated through several types of numerical experiments, as compared to projection rule (PR) for HHNNs. Experimental results show that pattern embedding by the proposed GDL is still possible for large number of patterns, in which the embedding by PR often fails. It is also shown that the proposed GDL can be improved, in terms both of stability of embedded patterns and of computational costs, by configuring the initial connection weights by PR and then by modifying the connection weights by GDL.

1. Introduction

Researches on associative memories based on Hopfield-type neural networks using complex values and hypercomplex values have been extensively investigated[1], such as theoretical stability for the models[2–4], schemes for embedding patterns to the network[5,6], and applications for embedding and retrieving patterns[7,8]. Signals with multiple levels, such as intensities of pixels in the image, can be naturally represented and operated by using (hyper)complex values[9,10].

Associative memory based on complex-valued neural networks, called CHAM (Complex-valued Hopfield Associative Memory) with polar representation has been presented in [9]. By using polar representation in complex-values, neurons in this network can handle multilevel signals, and it has been demonstrated that this network can embed and retrieve gray-scaled images. It is necessary for associative memories to embed patterns that are correlated to each other. Thus, projection rule from the real-valued counterpart[11]

\[ h^2 = +1 \]

has been constructed and evaluated in [5]. Another type of learning algorithm, which manipulates the energy landscape from the patterns that are to be embedded, has also been presented and analyzed in [6].

A lot of associative memories have been presented for extensions of CHAMs. One extension is to introduce hypercomplex-valued number system, such as quaternion, for the neurons' states. Quaternion is a four-dimensional hypercomplex number system[12], and a quaternion can be represented by an amplitude and three kinds of phases, like complex-values being represented by an amplitude and a phase[13]. Thus, three-dimensional multilevel signals such as color-valued pixel intensities (red, green, and blue) for a pixel can be represented by three phases in a quaternion. A quaternionic Hopfield associative memory (QHAM) with this type of representation has been presented and theoretically analyzed in [10], and its performances have been investigated in [14] through numerical experiments. Also, several types of quaternionic associative memories have also been presented and analyzed in [15–18].

Another extension for CHAMs adopts other number system with two dimensions. Hyperbolic number is a two-dimensional number with one real-part and one imaginary-part, like a complex value, but the imaginary unit $h$ satisfies $h^2 = +1$ [19]. Hyperbolic Hopfield Neural Network (HHNN) has been proposed in [20–22], and a projection rule for embedding patterns has also been introduced. The performances of HHNNs have been explored through the networks configured by the projection rule, and it is shown that the proposed projection rule has superior performances from the viewpoint of noise tolerance, as compared to CHAMs with the same storage capacity.

---

Paper

---

* Manuscript Received Date: April 17, 2020

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

§ Graduate School of Engineering, University of Hyogo; 2167 Shosha, Himeji, Hyogo 671-2280, JAPAN

‡ Cadence Design Systems, JAPAN; 2-100-45 Shin-Yokohama, Kohoku-ku, Yokohama, Kanagawa 222-0033, JAPAN

§ Mathematical Science Center, University of Yamanashi; 4-3-11, Takeda, Kofu, Yamanashi 400-8511, JAPAN

Key Words: hopfield network, associative memory, hyperbolic number, gradient descent.
Projection rule has a constraint that the neurons in the network must be fully connected[23], thus it is not applicable for associative memories with partial connections of neurons, such as bidirectional associative memories (BAMs)[24,25]. One possible solution for embedding patterns onto the network with partial connections is to adopt gradient descent learning scheme[26,27]. This learning algorithm is based on gradient descent method, where the connection weights among neurons are gradually modified by imposing the patterns to be embedded to the network. This learning algorithm has been formulated for several types of associative memories, such as CHAM[5,28,29] and QHAM[30], but this is not available for HHNNs.

In this paper, we propose a gradient descent learning for HHNNs and investigate the performances of the networks with the proposed learning algorithm. We also investigate this learning scheme with the initial connection weights being set by projection rule. This configuration is expected to embed patterns more stably and to improve the speed for embedding patterns.

The rest of the paper is organized as follows. Section 2 provides preliminaries for associative memory based on hyperbolic neural network and projection rule for embedding patterns. A gradient descent learning scheme for HHNNs are presented in section 3. The performances for this learning scheme are shown in section 4 through the results by numerical experiments, and the experimental results are discussed in section 5. This paper concludes with section 6.

2. Associative Memory Based on Hyperbolic Hopfield Neural Network

2.1 Hyperbolic Algebra and Hyperbolic Hopfield Neural Network

Hyperbolic numbers, also called as split-complex numbers, are categorized as a Clifford algebra of degree 2. A hyperbolic number $z$ is composed of one real-part and one imaginary-part like complex number, denoted by

$$z = x + hy,$$

where $x$ and $y$ are real numbers and $h$ is an imaginary unit satisfying $h^2 = -1$ where $h \neq \pm 1$. The operations for addition and multiplication follow those of complex numbers with $h^2 = -1$.

Hyperbolic Hopfield Neural Network, called HHNN, has been proposed in [20–22], where the representation and operations for these networks follow those in hyperbolic number system. We adopt the neuron model proposed in [22] where the neurons’ states are on a circle in the hyperbolic plane.

The state of a hyperbolic neuron is represented as a point on a circle with radius 1 in the hyperbolic plane, denoted by

$$c_k = \cos 2k\theta_K + h\sin 2k\theta_K,$$

where $\theta_K = \pi/K$ and the parameter $K$ is an integer constant, called resolution factor. By using this representation, a multilevel signal $k (0 \leq k \leq (K - 1))$ can be represented by a discrete phase $c_k$.

The neuron $j$ accepts the output signal $z_k$ from the other neuron $k$ weighted by the connection weight $w_{jk}$, and a weighted sum for the neuron $j$, denoted by $S_j$, is calculated by

$$S_j = \sum_{k=1}^{N} w_{jk}z_k,$$  \hspace{1cm} (2)

where the network contains $N$ neurons that are connected to each other. The output of the neuron $j$, denoted by $z_j$, is determined by an activation function defined as:

$$z_j = f(S_j) = \arg \max_{c_k} \Re(c_k S_j),$$  \hspace{1cm} (3)

where $\Re(S_j)$ denotes the real part of $S_j$. This operation shifts the phase in a neuron to one of discrete phase in a circle where the phases are nearest to each other.

The stability of the network with hyperbolic neurons can be proven by showing that the energy of the network monotonically decreases with respect to the change of neuron’s state[22]. The energy of the network $E$ is defined as

$$E = -\frac{1}{2} \sum_{j}^{N} \sum_{k \neq j}^{N} \Re(z_j w_{jk} z_k).$$

The conditions for the decrease of the energy are

$$w_{jk} = w_{kj} \text{ (connection weights should be symmetric)}$$

and $w_{jj} = 0 \text{ (self-connection should be eliminated)}$.

The details of the proof are shown in [22].

2.2 Projection Rule

We also present a projection rule for HHNNs as a method for embedding patterns to the network[22]. Assume that $P$ patterns with $N$ elements and $K$ resolution factor are to be embedded to the network. Let $z_p = (z_{p}^1,z_{p}^2,\ldots,z_{p}^N)$ be the $p$-th pattern with $K$ resolution where $z_i \in \{0,\ldots,(K - 1)\}$ ($i = 1,\ldots,N$). Each element of this pattern is encoded to

$$\xi_i = c_{z_i},$$

by using Eq.(1). Thus $\xi_p = (\xi_{p}^1,\xi_{p}^2,\ldots,\xi_{p}^N)$ is the $p$-th encoded pattern for embedding to the network. Then, the matrix $\Xi$ containing all embedding patterns is given by

$$\Xi = \left( (\xi_1^1)^T (\xi_1^2)^T \cdots (\xi_1^P)^T \right).$$

In projection rule, the patterns in this matrix is first projected to the patterns that are orthogonal to each other, then these projected patterns are embedded by Hebbian rule. The weight matrix $W = \{w_{ij}\}$ is obtained from the embedding pattern matrix $\Xi$: 
\[ B = \Xi \left( \Xi^T \Xi \right)^{-1} \Xi^T, \]
\[ W = B. \]

Connection weights in \( W \) have the following properties: they are symmetric (\( w_{jk} = w_{kj} \)), and the diagonal elements (i.e., self-connections \( w_{jj} \)) are not 0 but take hyperbolic numbers (not real numbers). Although the self-connections are present, pattern retrieval is still possible when smaller numbers of patterns are embedded, as simulation results show [22].

We also describe the projection rule proposed in [31], where the noise tolerance for embedded patterns is improved. This is used for comparing with the performances of our proposed GDL. Connection weights \( W = \{ w_{jk} \} \) in this projection rule are defined as:
\[ w_{jk} = b_{kj} \quad \text{(for } j \neq k), \]
\[ w_{jj} = |\text{Im}(b_{jj})| + h|\text{Im}(b_{jj})|, \]
where \( b_{ij} \) is the \((i,j)\) component of \( B \).

3. Gradient Descent Learning for Hyperbolic Hopfield Neural Network

In this section, we introduce a gradient descent learning (GDL) for HHNNs, based on the derivation for CHAMs [5,28].

Let \( w_{ji} \) be the connection weight matrix from the neuron \( i \) to neuron \( j \), and \( \xi^j_p \) be the \( j \)-th element in the \( p \)-th pattern vector. They are defined as
\[ w_{ji} = u_{ji} + hR_{ji}, \]
\[ \xi^j_p = x^p_j + hI^p_j. \]
From \( w_{ji} = w_{ji} \) (a condition for the connection weight) and Eq.(5), we get
\[ u_{ji} = u_{ji}, \]
\[ v_{ji} = v_{ji}, \]
thus we only consider the parameters satisfying \( j > i \).

We define the error function in GDL for HHNNs in the same manner in CHAM. The error for the neuron \( j \) with respect to the \( j \)-th element in the \( p \)-th pattern, denoted by \( E^j_p \), is given as
\[ E^j_p = \frac{1}{2} |\xi^j_p - S^j_p|^2 \]
\[ = \frac{1}{2} \{(x^p_j - R^p_j)^2 + (y^p_j - I^p_j)^2\}, \]
where \( S^j_p = R^p_j + hI^p_j \) is the weighted sum for the neuron \( j \), which is calculated by Eq.(2), when \( \xi^p_p \) is input to the network. This function requires that both of the amplitudes and phases for the training pattern should be respectively same as those in the weighted sum of the neuron. Since \( f(S^j_p) \) is not a differentiable function and \( f(S^j_p) = \xi^j_p \) holds if \( S^j_p = \xi^j_p \), we adopt the error function as the difference between the pattern and the internal state of a neuron.

By using Eq.(9), the total error \( E \) is defined as an accumulation of the errors for all \( j \) and \( p \):
\[ E = \sum_{j,p} E^j_p, \]
If \( \xi^j_p = S^j_p \) is satisfied, which implies not only \( \xi^j_p = f(S^j_p) \) but also \( \xi^j_p = |S^j_p| \), then both phase and amplitude of \( \xi^j_p \) and \( S^j_p \) are equal, for all \( p \) and \( j \); \( E \) vanishes and it is regarded that all training patterns are completely embedded.

In GDL, the weight parameters are updated by the following equations:
\[ w_{ji} \leftarrow w_{ji} + \Delta w_{ji}, \]
\[ \Delta w_{ji} = \Delta u_{ji} + h \Delta v_{ji}, \]
\[ \Delta u_{ji} = -\eta \frac{\partial E}{\partial u_{ji}}, \]
\[ \Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}}, \]
where \( \eta \) is a constant parameter called learning rate. By using Eq.(10) with the conditions \( u_{ji} = v_{ji} \) and \( v_{ji} = v_{ji} \), the partial derivatives for real-part of the connection weight are obtained as follows:
\[ \frac{\partial E^p_i}{\partial u_{ji}} = x^p_i \frac{\partial E^p_i}{\partial R^p_i} + y^p_i \frac{\partial E^p_i}{\partial I^p_i}, \]
\[ \frac{\partial E^p_i}{\partial u_{ji}} = x^p_j \frac{\partial E^p_i}{\partial R^p_i} + y^p_j \frac{\partial E^p_i}{\partial I^p_i}, \]
\[ \frac{\partial E^p_i}{\partial R^p_i} = -x^p_j + R^p_j, \]
\[ \frac{\partial E^p_i}{\partial I^p_i} = -y^p_j + I^p_j. \]
Similarly, the partial derivatives for imaginary part are obtained as follows:
\[ \frac{\partial E^p_i}{\partial v_{ji}} = y^p_i \frac{\partial E^p_i}{\partial R^p_i} + x^p_i \frac{\partial E^p_i}{\partial I^p_i}, \]
\[ \frac{\partial E^p_i}{\partial v_{ji}} = y^p_j \frac{\partial E^p_i}{\partial R^p_i} + x^p_j \frac{\partial E^p_i}{\partial I^p_i}. \]
Thus, Eqs.(13) and (14) are respectively expanded as follows:
\[ \Delta u_{ji} = -\eta \frac{\partial E}{\partial u_{ji}}, \]
\[ \Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}}, \]
\[ \Delta u_{ji} = -\eta \sum_p \left( \frac{\partial E^p_i}{\partial u_{ji}} + \frac{\partial E^p_j}{\partial u_{ji}} \right), \]
\[ \Delta v_{ji} = -\eta \sum_p \left( \frac{\partial E^p_i}{\partial v_{ji}} + \frac{\partial E^p_j}{\partial v_{ji}} \right), \]
\[ = -\eta \sum_p \left( 2x^p_i x^p_j + 2y^p_i y^p_j - x^p_j R^p_j - y^p_j I^p_j \right) \]
\[ -x^p_j R^p_j - y^p_j I^p_j. \]
The update equation for the connection weights for neurons $w_{ji}$ can be written as

$$
\Delta w_{ji} = \Delta u_{ji} + h \Delta v_{ji} = \eta \sum_p (2\xi_i^p \xi_j^p - \xi_i^p S_i^p - \xi_j^p S_j^p).
$$

(23)

Iterative applications of the above update equation make the connection weight $w_{ji}$ gradually modified so that the training patterns can be retrieved. The self-connections $w_{ij}$ are set to 0 and not updated, due to the stability condition of the network. Since it often takes a long time to get the stable connections, actually, we introduce a condition for terminating the iteration, i.e., the updates for connection weights terminate when all embedded patterns are regarded as stable. Learning by GDL is terminated when

$$
\text{Re}(S^p \xi^p) > |S^p| |\xi^p| \cos \theta_K,
$$

(24)

$$
\text{Re}(S^p \xi^p) > |S^p| |\xi^p| \cos \theta_K,
$$

(25)

are obtained for all combinations of $i$ and $j$ ($i \neq j$) where $i = 1, \ldots, N$ and $j = 1, \ldots, N$. These conditions satisfy

$$
f(S^p) = \xi^p 
$$

and

$$
f(S^p) = \xi^p
$$

(26)

for all combination of $i$ and $j$ ($i \neq j$).

4. Experimental Results

In this section, we investigate the performances of the proposed GDL though numerical experiments for embedding and retrieving patterns to the networks. We use two types for investigating the proposed GDL, first type is GDL where all connection weights are initially set to zero (called GDL), and the second type is GDL where initial connection weights are determined by PR with the diagonal elements being set to zero (called PR-GDL). We also use the HHNN equivalent of the improved projection rule (called ‘Improved PR’) [31], which is mentioned in section 2.2, as a base line for the performances.

We use the following parameters: the number of neurons in the network is denoted by $N$, the number of patterns to be embedded to the network is denoted by $P$, the resolution factor for neurons’ state is denoted by $K$.

4.1 Experimental Setup

First, we describe the basic procedures and conditions for the experiments in sections 4.2, 4.3, and 4.4. Experimental conditions in the section 4.5 are rather different, thus they are described independently.

Experiments are performed by the following procedures.

1. An HHNN is configured, where the number of neurons in the network is set to $N = 200$ and the resolution factor in each neuron is set to $K$.

2. $P$ random patterns with $N$ elements, $\xi^p$ ($p = 1, \ldots, P$), are generated. Each element $\xi_i^p$ is set by $\xi_i^p = c_k$ with a uniformly distributed random number $k(=0, \ldots, (K-1))$. These $P$ patterns are embedded to the network by GDL, PR-GDL, or Improved PR.

3. Test patterns are generated. The ways for generating test patterns depend on the experiment, such as they are chosen from the embedded patterns with noise being affected, or randomly generated.

4. Each of the test patterns is set to the initial configuration of the network, and the states of the neurons in the network are updated until they do not change by updates.

5. Retrieval performances are evaluated from these final states for the test input patterns.

The experiments are conducted in the following computer:

- Processor: Intel(R) Xeon(R) CPU E5-1650 v4, operated in clock frequency of 3.60GHz
- Memory: 64GB
- OS: Ubuntu 16.04
- Compiler: gcc 5.4.0 (learning methods are implemented in C++)

4.2 Retrieval from Random Patterns

Second, we investigate the number of local minima emerged in the energy landscape in the network when $P$ patterns are embedded to the network. This can be estimated by the following experiment: A network with $P$ patterns being embedded is prepared, and the retrievals for patterns are conducted from the randomly generated test patterns. The retrieval is regarded as successful if the final state of the network is one of the embedded patterns.

The parameters for the experiments are $K = 4, 8, \ldots, 40$, and $P = 1, 5, 10$. For each $K$ and $P$, 100 networks are generated by embedding patterns and 100 patterns are also randomly generated as test patterns.

The success rates for $P = 1, 5, 10$ are shown in Figs. 1(a), (b), and (c), respectively. We see similar tendencies for the performances of GDL, PR-GDL and Improved PR. The success rates for the case $P = 1$ are almost 25% by Improved PR and GDL, whereas those by GDL are slightly low (around 20%). For $P = 5$ and $P = 10$, the success rates become around 13% and 10%, respectively.

We also calculate the standard deviations for the success rates in the experiments. Fig. 1(d) shows the standard deviations calculated from the trials including all $K$ for each of $P$ values. These values are almost same (around 4%), except for the case $P = 1$ by GDL (6.8%).

4.3 Computational Costs for Embedding Patterns

We measure the computation costs for embedding patterns as execution periods by the program in a computer. Mean execution periods with their standard deviations for $P = 1, 5, 10, 30, 60, 90$ for embedding
Fig. 1 Success rates for random pattern embedding and retrieval. (a) $P=1$ (the number of patterns embedded to networks is 1), (b) $P=5$, (c) $P=10$, and (d) Standard deviation comparison per $P$.

Fig. 2 Execution time (seconds) with standard deviation for embedding patterns by Improved PR, GDL, and PR-GDL. (a) $P=1$, (b) $P=5$, (c) $P=10$, (d) $P=30$, (e) $P=60$ and (f) $P=90$. 
patterns in the experiments in section 4.2 are shown in Figs. 2(a), (b), (c), (d), (e), and (f), respectively.

The computational costs by Improved PR are governed by calculating pseudo-inverse matrices, and these are almost constant regardless of $P$ and $K$ configurations. For GDL learning, the computational costs are linearly proportional to $K$ for all $P$s. Large $P$ and/or $K$ values mean large amount of information for patterns being loaded to the network, thus many iterations are necessary to embed patterns in the stable points. For PR-GDL learning, the computational costs for $P = 1, 5, 10$ are almost constant, but for larger $P$s, they show similar tendencies to those by GDL.

4.4 Retrieval from Noisy Inputs

Furthermore, we investigate the retrieval performances when the patterns affected by noises are used as initial state of the network. The following experiment is conducted; the network is configured with the parameters $P$ and $K$, the initial state for the network is selected from one of the embedded patterns (denoted by the original pattern) with being affected by the noise, and the retrieval is regarded as successful if the final state of the network is the same as the original pattern. The noise affection is conducted by the following scheme; the value of each element in the pattern is replaced to other value which is randomly selected with the probability $r$.

10 different connection weights are produced in this experiment, and 100 initial patterns are prepared for each $r = 0.05, 0.10, \ldots, 0.75, 0.80$. In the case $r = 0.0$ (the case without noise), the number of initial pattern is set to 1. The parameters in the experiment are $K = 4, 8, 16, 32, 64$.

Figs. 3, 4, 5, 6, 7, and 8 show the retrieval successes rates for Improved PR, GDL and PR-GDL, with $P = 10, 20, 30, 40, 50, 60$, respectively. The standard deviations for success rates are also calculated for each of $K$s and for each of learning schemes. When small numbers of patterns are embedded, such as $P = 10$ and $P = 20$, all retrieval success rates show almost the same tendencies with respect to the noise probability, although the success rates obtained by GDL learning have larger standard deviations (see Figs. 3 and 4). When $P$ and $K$ become larger, the networks trained by GDL or PR-GDL have better robustness against noisy input, as compared with those by Improved PR.

4.5 Image Retrieval with PR-GDL

Finally, we demonstrate that PR-GDL can be used for storing and retrieving natural images, as performed in [31]. This task is useful in the sense that GDL for HHNN can embed the patterns with high resolution $K$, and that pattern retrievals are possible from input patterns affected by the noise other than impulsive noise (as in section 4.4). The embedded patterns come from CIFAR-10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html), and color images in this dataset are converted to gray-scaled ones with 256 levels. Each image in this dataset has $32 \times 32$ pixels. Examples of the embedded patterns are shown in Fig. 9. 30 embedded patterns are randomly selected from the dataset. The intensity of a pixel located at $(x, y)$ in the embedded image $p$, denoted by $g_{xy}^p \in \{0, \ldots, 255\}$, is converted to the embedded pattern $\xi_p = \{\xi_i^p\}$ ($i = 1, \ldots, 1024$) by

$$\xi_{x+32y+1}^p = c g_{xy}^p,$$

where the pixels at coordinates $(0, 0)$, $(0, 31)$, $(31, 0)$, $(31, 31)$ correspond to the pixels at the corner of upper left, upper right, lower left, and lower right, respectively. For embedding patterns, a network with $N = 1024 (= 32 \times 32)$ neurons is prepared and the resolution faction $K$ is set to 256 that can represent the gray-scaled intensities of pixels in the images. The converted patterns are embedded to the network by using PR-GDL.

From the embedded patterns, one pattern is selected and this pattern is affected by Gaussian noise with $\sigma = 30$. This pattern is used for the input pattern to the network. Examples of noise-affected patterns are shown in Fig. 10(b), which are created from the image in Fig. 10(a). By using these patterns as the initial state of the network, the retrieved patterns from the network are shown in Fig. 10(c).

For evaluating the retrieved patterns from the network, the PSNR (peak signal-to-noise ratio) is calculated between the retrieved pattern and the original input pattern. From the definition of PSNR, higher PSNR is obtained if the images are more similar, and if the images are identical, their PSNR becomes infinite. PSNR value between Figs. 10(a) and 10(c) is $53.61$, which means that the retrieval image is not completely same as but quite similar to the embedded image.

Other two examples for image retrieval are shown in Fig. 11. In these cases, the retrieval images are respectively identical to the original images.

5. Discussion

In this section, we discuss the properties of networks embedded the patterns by GDL, PR-GDL, and Improved PR based on the results shown in the section 4.

First, we investigate the results obtained in section 4.2 (Fig. 1). In HHNNs, when one pattern $\xi^\ast$ is embedded to the network, three reflected patterns $-\xi^\ast$, $\pm h \xi^\ast$ are also to be embedded, because they have the same energy with respect to the network states. Thus, in the case of $P = 1$, the success rate is expected to take $1/4 = 0.25$. As shown in Fig.1(a), the success rates for Improved PR (and also GDL with lower $K$) take around $0.25$, which corresponds to this expected rate.

In the case $P = 1$ with large $K$s, the success rates obtained by GDL are lower than those by Improved PR and PR-GDL. We suppose that there are many
Fig. 3  Success rates for pattern retrieval from noisy patterns ($P = 10$)

Fig. 4  Success rates for pattern retrieval from noisy patterns ($P = 20$)
Fig. 5 Success rates for pattern retrieval from noisy patterns \((P = 30)\)

Fig. 6 Success rates for pattern retrieval from noisy patterns \((P = 40)\)
Fig. 7 Success rates for pattern retrieval from noisy patterns ($P = 50$)

Fig. 8 Success rates for pattern retrieval from noisy patterns ($P = 60$)
cases where the learning iterations by GDL are not enough, and this can be observed from larger standard deviation for this case (see Fig. 1(d)). This may be due to relatively soft conditions on which it is terminated to update connection weights (Eqs.(24) and (25)).

In the case $P=10$, the success rates obtained by GDL and PR-GDL are better than those by Improved PR (see Fig.1(c)). In this case, iterative updates by GDL in PR-GDL do not conduct, because the terminating conditions (Eqs.(24) and (25)) are satisfied before GDL is applied. Thus the connection weights are set by PR with eliminating the diagonal elements. On the other hand, in Improved PR the connection weights are configured according to Eq.(4). The difference for connection weights in this case is whether the diagonal elements in the connection weights are zero or not. Self connections by Improved PR cannot be removed and tend to stabilize the states of neurons in the network if they are not the states of embedded patterns[31], thus pseudomemories can emerge in the network.

The success rates obtained by PR-GDL for the case $P=10$ are always higher than those by GDL for many $K$s, and also this can be found for the cases $P=1$ and $P=5$. Thus, PR-GDL learning can be the best in the three types of embedding methods. It is supposed that PR-GDL learning can avoid local minima in the error landscape ($E$) in learning, by the initial configuration (connection weights) being set by PR.

The success rates for $P=5, 10$ and $K=4$ (and $K=8$) are degraded for all learning schemes, we suppose that this is simply because the pattern space spanned by $N$ and $K$ is not enough wide against the number of patterns to be embedded to the network.

Then, we proceed to the computational costs by the learning schemes (section 4.3, Fig. 2). Since the projection rule is a one-shot learning algorithm and almost computational time is governed by calculating inverse matrix, we see that the standard deviations of Improved PR is quite small.

Constant and relatively small computational costs can be achieved by PR-GDL, because it is not necessary to update the states of the neurons in the network by GDL, i.e., the initial connection weights set by PR can satisfy the conditions in Eqs.(24) and (25), for $P=1$, $P=5$ and $P=10$.

The execution time increases at $K=40$ in $P=30$, $K \geq 24$ in $P=60$, and $K \geq 16$ in $P=90$, where the connection weights in the network are updated by GDL in these cases. The execution time by PR-GDL does not exceed to the execution time by GDL for all $P$ and $K$ cases, smaller number of iterations by GDL learning is enough for embedding patterns, due to the initial connection weights being set by PR.
Furthermore, we discuss the results for noisy initial configuration (section 4.4, Figs. 3, 4, 5, 6, 7, 8). As we have described, when $P$ and $K$ become larger, the retrieval success rates by Improved PR deteriorate, while the success rates by GDL and PR-GDL maintain high values (for examples, see Figs. 5(f) ($P=30, K=64$) and 6(f) ($P=40, K=64$)). We suppose that the deterioration by Improved PR is due to the diagonal elements in the connection weights $(w_j,s)$; these are zero by GDL and PR-GDL, but these by Improved PR are not zero.

The retrieval success rates by PR-GDL are higher than those by GDL in most cases, particularly the differences between them are high values when $P$ and $K$ takes higher values. We suppose that this can be due to the stable attractors in the network made by the combination of PR and GDL; the attractors are initially created by PR and these are to become more stable by the iterative updates of connection weights by GDL.

6. Conclusion

This paper presents a scheme for embedding patterns to the Hopfield-type associative memories based on hyperbolic numbers called HHNNs, and its performances have been investigated. Connection weights in the network are gradually updated according to the states of embedded patterns by using gradient descent learning method.

The performances are evaluated by embedding and retrieving patterns by the proposed scheme and a projection rule with an improvement in terms of noise tolerance as a conventional method. It is shown that similar tendencies to the projection rule are obtained by the proposed schemes, and that for embedding many patterns with high resolution the proposed scheme can embed the patterns better than the projection rule. It is also shown that learning by gradient descent method can be improved, in terms of robustness of the embedded patterns and computational costs in learning, by configuring the initial connection weights by the projection rule.

The networks with the pattern being embedded by PR-GDL have better performances than those by Improved PR or GDL. This can be achieved by the combination of PR and GDL; at first PR can make stable attractors in the network, and then iterative updates for connection weights by GDL make these attractors more stable. We also demonstrate that gray-scaled images can be embedded by PR-GDL and can be successfully retrieved from noise affected images.

It is expected that our proposed scheme can be improved by incorporating various optimization methods used for training multilayer perceptron-type networks. It is important to explore the applicability of this scheme for bidirectional associative memories, where partial connections among neurons are present and learning by GDL is necessary. These remain for our future work.

References

[1] F. Z. de Castro and M. E. Valle: A broad class of discrete-time hypercomplex-valued hopfield neural networks; Neural Networks, Vol. 122, pp. 54-67 (2020)
[2] Y. Kuroe, S. Tanigawa and H. Imai: Models of hopfield-type clifford neural networks and their energy functions — hyperbolic and dual valued networks —; Proceedings of the 18th International Conference on Neural Information Processing (ICONIP), Vol. 1, pp. 560-569 (2011)
[3] W. Zhou and J. M. Zurada: Stability condition for discrete time multi-valued recurrent neural networks in asynchronous update mode; 2014 International Joint Conference on Neural Networks (IJCNN), pp. 3402-3405 (2014)
[4] E. F. Valle and F. Z. de Castro: On the dynamics of hopfield neural networks on unit quaternions; IEEE Transactions on Neural Networks and Learning Systems, Vol. 29, No. 6, pp. 2464-2471 (2018)
[5] D. L. Lee: Improvements of complex-valued hopfield associative memory by using generalized projection rules; IEEE Transaction on Neural Networks, Vol. 17, No. 5, pp. 1341-1347 (2006)
[6] T. M. Mieczkońgu, C. Güzelş and J. M. Zurada: A new design method for the complex-valued multistate hopfield associative memory; IEEE Transactions on Neural Networks, Vol. 14, No. 4, pp. 891-899 (2003)
[7] G. Tanaka and K. Aihara: Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction; IEEE Transactions on Neural Networks, Vol. 20, No. 9, pp. 1463-1473 (2009)
[8] T. Minemoto, T. Isokawa, H. Nishimura and N. Matsui: Pseudo-orthogonalization of memory patterns for complex-valued and quaternionic associative memories; Journal of Artificial Intelligence and Soft Computing Research, Vol. 17, No. 4, pp. 257-264 (2017)
[9] S. Jankowski, A. Lozowski and J. M. Zurada: Complex-valued multistate neural associative memory; IEEE Transactions on Neural Networks, Vol. 7, No. 6, pp. 1491-1496 (1996)
[10] T. Isokawa, H. Nishimura, A. Saitoh, N. Kamiura and N. Matsui: On the scheme of quaternionic multistate hopfield neural network; Proceedings of Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on advanced Intelligent Systems (SCIS & ISIS 2008), pp. 809-813 (2008)
[11] L. Personnaz, I. Guyon and G. Dreyfus: Information storage and retrieval in spin glass like neural networks; Journal of Physics Letter, Vol. 46, No. 8, pp. 359-365 (1985))
[12] W. R. Hamilton: Lectures on Quaternions, Hodges and Smith (1853)
[13] T. Bülow: Hypercomplex Spectral Signal Representations for the Processing and Analysis of Images, PhD thesis, Christian-Albrechts-Universit¨at zu Kiel (1999)
[14] T. Minemoto, T. Isokawa, H. Nishimura and N. Matsui: Quaternionic multistate hopfield neural network with extended projection rule; Artificial Life and Robotics, Vol. 21, No. 1, pp. 106-111 (2016)
[15] T. Isokawa, H. Nishimura, N. Kamiura and N. Matsui: Quaternionic multistate neural associative memories based on hyperbolic numbers called HHNNs; Journal of Artificial Intelligence and Soft Computing Research, Vol. 7, No. 4, pp. 257-264 (2017).
sui: Associative memory in quaternionic hopfield neural network; International Journal of Neural Systems, Vol. 18, No. 2, pp. 135–145 (2008)

[16] M. Kobayashi: Rotational invariance of quaternionic hopfield neural networks; Electrical Engineering in Japan, Electronics and Communications in Japan, Vol. 11, No. 4, pp. 516–520 (2016)

[17] M. Kobayashi: Symmetric quaternionic hopfield neural networks; Neurocomputing, Vol. 240, pp. 110–114 (2017)

[18] M. Kobayashi: Quaternionic hopfield neural networks with twin-multistate activation function; Neurocomputing, Vol. 267, pp. 304–310 (2017)

[19] J. Cockle: On a new imaginary in algebra; The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, Vol. 34, No. 226, pp. 37–47 (1849)

[20] M. Kobayashi: Hyperbolic hopfield neural networks; IEEE Transactions on Neural Networks and Learning Systems, Vol. 24, No. 2, pp. 335–341 (2012)

[21] M. Kobayashi: Global hyperbolic hopfield neural networks; IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E99-A, No. 12, pp. 2511–2516 (2016)

[22] M. Kobayashi: Hyperbolic hopfield neural networks with directional multistate activation function; Neurocomputing, Vol. 275, pp. 2217–2226 (2018)

[23] M. Kitahara and M. Kobayashi: Projection rule for complex-valued associative memory with large constant terms; Nonlinear Theory and Its Applications, Vol. 3, No. 3, pp. 426–435 (2012)

[24] B. Kosko: Bidirectional associative memories; IEEE Transactions on System, Man, and Cybernetics, Vol. 18, No. 1, pp. 49–60 (1988)

[25] D. L. Lee: A multivalued bidirectional associative memory operating on a complex domain; Neural Networks, Vol. 11, No. 9, pp. 1623–1635 (1998)

[26] D. L. Lee: Improving the capacity of complex-valued neural networks with a modified gradient descent learning rule; IEEE Trans. Neural Netw., Vol. 12, No. 2, pp. 439–443 (2001)

[27] D. L. Lee: Complex-valued neural associative memories: Network stability and learning algorithm; Complex-Valued Neural Networks—Theories and Applications ed. A. Hirose, Series on Innovative Intelligence, Vol. 5, pp. 29–55 (2003)

[28] M. Kobayashi, H. Yamada and M. Kitahara: Noise robust gradient descent learning for complex-valued associative memory; IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E94-A, No. 8, pp. 1756–1759 (2011)

[29] M. Kobayashi: Gradient descent learning rule for complex-valued associative memories with large constant terms; IEEE Transactions on Electrical and Electronic Engineering, Vol. 11, No. 3, pp. 357–363 (2016)

[30] M. Kobayashi: Gradient descent learning for quaternionic hopfield neural networks; Neurocomputing, Vol. 260, pp. 174–179 (2017)

[31] M. Kobayashi: Noise robust projection rule for hyperbolic hopfield neural networks; IEEE Transactions on Neural Networks and Learning Systems, Vol. 31, No. 1, pp. 352–356 (2020)

Authors

Masayuki TSUJII

Masayuki Tsuji has been belonging to Cadence Design Systems Japan, and has been pursuing his Ph.D degree at the Graduate School of Engineering, University of Hyogo.

Tejiro ISOKAWA (Member)

Tejiro Isokawa received his B.E. degree, M.E. degree, and D.E. degree in 1996, 1999, and 2004, respectively, from Himeji Institute of Technology. He is currently an associate professor in the Graduate School of Engineering, University of Hyogo. His research interests include nanocomputing, molecular robotics, hypercomplex-valued neural networks, and cognitive models.

Masaki KOBAYASHI

Masaki Kobayashi is a professor at University of Yamanashi. He received the B.S., M.S., and D.S. degrees in mathematics from Nagoya University in 1989, 1991, and 1996, respectively. He became a research associate and an associate professor at the University of Yamanashi in 1993 and 2006, respectively. He has been a professor since 2014.

Nobuyuki MATSUI

Nobuyuki Matsui received his B.S. degree in physics from the Faculty of Science, Kyoto University, Japan, in 1975, and M.E. and Dr. Eng. degrees in nuclear engineering from Kyoto University in 1977 and 1980, respectively. He is currently a professor emeritus at University of Hyogo. His research interests include machine learning, quantum computing, and their applications.

Naotake KAMIURA

Naotake Kamiura received his M.E. degree (Electronic Engineering) in 1992 and the D.E. degree (Doctor of Engineering) in 1995 from Himeji Institute of Technology, Japan. He is currently a professor in the Department of Electronics and Computer Science, Graduate School of Engineering, University of Hyogo. His research interests include the application of soft computing to medical engineering. He is members of the Institute of Electronics, Information and Communication Engineers, the the Japanese Society of Medical Imaging Technology and the IEEE.