Constructing Convolutional Neural Networks Based on Quaternion

Shuto Hongo
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—A convolutional neural network based on quaternion, a four-dimensional hypercomplex number system, is proposed and evaluated in this paper. Called Quaternionic Convolutional Neural Networks (QCNNs), these networks can accept and operate three-dimensional signals by neurons in the networks. The performances of the proposed networks are investigated through classification of CIFAR-10 color images, and it is shown that the proposed QCNN outperforms a conventional (real-valued) CNN.

Index Terms—quaternion, convolutional neural network, complex-valued neural network

I. Introduction

Hypercomplex number systems would be useful tools to cope with multi-dimensional data in neural networks. Typically a single neuron can take only one real value as its input, thus a network should be configured so that several neurons are assigned for accepting multi-dimensional data. This traditional way to treat multidimensional data can be inadequate for the data of which each element itself has less meaning than the data with unified elements, such as radio frequency/acoustic signals and coordinates in the space.

The complex number system is a system where a number consists of one real number and one imaginary number, or a phase and an amplitude. Complex number systems have thus been utilized to represent two-dimensional data elements as a single entity. Application of complex numbers to neural networks have been extensively investigated, and recent researches are summarized in the references [1]–[4].

Extensions for higher dimensionality (dimensions more than two), which is similar to the extension from the real numbers to complex numbers, can be considered. Several hypercomplex systems have been introduced, such as quaternion (four dimensions), octonion (eight dimensions), and sedenion (16 dimensions). Also, Clifford algebras are introduced as a generalization of these hypercomplex systems [5], [6].

Quaternion is a four-dimensional hypercomplex number system introduced by Hamilton [7], [8]. This number system has been extensively employed in several fields, such as modern mathematics, physics, control of satellites, computer graphics, and so on [9]–[11]. One of the benefits provided by quaternions is that affine transformations of geometric figures in three-dimensional spaces, especially spatial rotations, can be represented compactly and efficiently. Applying quaternions to the field of neural networks has been recently explored in an effort to naturally represent high-dimensional information, such as color and three-dimensional body coordinates, by a quaternionic neuron, rather than complex-valued or real-valued neurons.

Thus, there has been a growing number of studies concerning the use of quaternions in neural networks. Multilayer perceptron (MLP) models have been developed in [12]–[18]. The use of quaternion in MLP models has been applied to several engineering problems such as control problems [13], color image compression [15], color night vision [19], [20], and predictions for the output of chaos circuits and winds in three-dimensional space [16], [17].

Though many quaternionic neural network models have been proposed, only a small number of neural networks with much more complexities, such as deep neural networks and convolutional neural networks, have been considered [21]–[24]. Thus, scalability for quaternionic (also complex-valued or hypercomplex-valued) neural networks has not been clarified for large-scaled networks. It would be important to investigate the properties and performances of such types of neural networks based on quaternion, for the applications of image classifica-

This study was financially supported by Japan Society for the Promotion of Science (Grant-in-Aids for Scientific Research (C) 16K00248 and 19K12141).

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

tion/generation tasks by neural networks.

In this paper, we propose a convolutional neural network based on quaternion, called Quaternionic Convolutional Neural Network (QCNN) and investigate its performances through the classification of color images in the CIFAR-10 dataset. The neuron model in the proposed network follows the one from [15] with split-type activation function. Operations in the convolution and pooling layers in the proposed network are also defined. The performances in a simple QCNN with three pairs of convolution and pooling layers are explored through the classification of images, with compared to conventional (real-valued) convolutional neural network. The rest of the paper is organized as follows. Section 2 gives the basic definitions of quaternion number system. The proposed convolutional neural network is described in section 3. Section 4 demonstrates experimental results. This paper finishes with conclusion in 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 quaternion $\boldsymbol{x}$ 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,

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

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

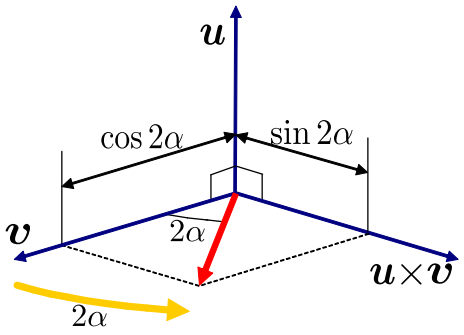

Fig. 1. A rotation in three-dimensional space using quaternions

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

$$\begin{aligned}\boldsymbol{qp} = &(q^{(e)}p^{(e)} - q^{(i)}p^{(i)} - q^{(j)}p^{(j)} - q^{(k)}p^{(k)}) \\ &+ (q^{(i)}p^{(e)} + q^{(e)}p^{(i)} - q^{(k)}p^{(j)} + q^{(j)}p^{(k)})\boldsymbol{i} \\ &+ (q^{(j)}p^{(e)} + q^{(k)}p^{(i)} + q^{(e)}p^{(j)} - q^{(i)}p^{(k)})\boldsymbol{j} \\ &+ (q^{(k)}p^{(e)} - q^{(j)}p^{(i)} + q^{(i)}p^{(j)} + q^{(e)}p^{(k)})\boldsymbol{k}.\end{aligned} \tag{8}$$

The conjugate of the product is given as

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

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

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

A rotation in three-dimensional space can be described by using a quatenrion and its conjugate, as

$$\boldsymbol{y} = \boldsymbol{pvp}^*, \tag{11}$$

where $\boldsymbol{p}$ is a quatenrion and $\boldsymbol{v}$ is a pure imaginary quaternion. $\boldsymbol{p}$ is composed from a pure imaginary quaterion $\boldsymbol{u}$ with $|\boldsymbol{u}| = 1$ and a phase parameter α with $|\alpha| < \pi$, defined as

$$\boldsymbol{p} = \cos\alpha + (\sin\alpha) \cdot \boldsymbol{u}. \tag{12}$$

Thus, we obtain (11) as

$$\boldsymbol{y} = (\cos 2\alpha) \cdot \boldsymbol{v} + (\sin 2\alpha) \cdot (\boldsymbol{u} \times \boldsymbol{v}). \tag{13}$$

This represents that, $\boldsymbol{y}$ is the three-dimensional vector (pure imaginary quaternion) $\boldsymbol{v}$ rotated with the axis $\boldsymbol{u}$ around the angle 2α (see Fig. 1).

III. Quaternoinic Neural Network Model

A. Neuron model

The proposed quatenrionic neural network is an extension of the one in [15], where each neuron in this network accepts three-dimensional signals as its input and output. The output of the neuron j, denoted by $\boldsymbol{y}_j$, is defined as

$$\boldsymbol{s}_j = \sum_{i=1}^{N} \frac{\boldsymbol{w}_{ji}\boldsymbol{x}_i\boldsymbol{w}_{ji}^*}{|\boldsymbol{w}_{ji}|} + \boldsymbol{\theta}_j, \tag{14}$$

$$\boldsymbol{y}_j = h(s_j^{(i)})\boldsymbol{i} + h(s_j^{(j)})\boldsymbol{j} + h(s_j^{(k)})\boldsymbol{k}, \tag{15}$$

where N is the number of neurons connected to the neuron j and i denotes the index for these neurons. The variables $\boldsymbol{x} \in \boldsymbol{I}$, $\boldsymbol{w} \in \boldsymbol{H}$, $\boldsymbol{\theta} \in \boldsymbol{I}$, $\boldsymbol{s} \in \boldsymbol{I}$ are the input, connection weight, threshold, and action potential, respectively. The activation function for this neuron adopts a split-type function, i.e. real-valued function $h(\cdot)$ is applied to each of quatenrionic components. We use the ReLU function as $h(\cdot)$, defined as

$$h(x) = \max(0, x). \tag{16}$$

B. Convolution and pooling elements

Convolutional operation for quaternionic neuron is presented. As in the real-valued convolutional neural networks, a quatenrionic convolutional neuron accept the signals from local receptive fields. The operation for this type of neuron is similar to neuron model in Eqs.(14) and (15):

$$\boldsymbol{s}_{ijk} = \sum_{s=1}^{m}\sum_{t=1}^{m} \frac{\boldsymbol{w}_{stk}\boldsymbol{x}_{i+s,j+t}\boldsymbol{w}^*_{stk}}{|\boldsymbol{w}_{stk}|} + \boldsymbol{\theta}_k, \tag{17}$$

$$\boldsymbol{y}_{ijk} = h(s^{(i)}_{ijk})\boldsymbol{i} + h(s^{(j)}_{ijk})\boldsymbol{j} + h(s^{(k)}_{ijk})\boldsymbol{k}, \tag{18}$$

where m denotes the size of filter for local receptive field.

Pooling operation act as a resolution reduction (down sampling) from the input signals. In the proposed network, we adopt so-called max pooling, where the maximum value for each of quaternionic components in a set of neurons is extracted:

$$\begin{aligned}\boldsymbol{p}_{ij} = &\max_{s\in[0,m]}(y^{(i)}_{mi+s,mj+s})\boldsymbol{i} \\ &+ \max_{s\in[0,m]}(y^{(j)}_{mi+s,mj+s})\boldsymbol{j} \\ &+ \max_{s\in[0,m]}(y^{(k)}_{mi+s,mj+s})\boldsymbol{k}.\end{aligned} \tag{19}$$

C. Batch Normalization

Batch normalization [25] is a process that is used for making normalization for the distribution of training data, in order to accelerate the speed for training. We incorporate a scheme for batch normalization as used in [24] for quaternionic neural networks.

An extension for the mean ($QE(\boldsymbol{x})$) and variance ($QV(\boldsymbol{x})$) of quaternions $\boldsymbol{x}$ are first defined as follows [26]:

$$QE(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}_i\boldsymbol{i} + x^{(j)}_i\boldsymbol{j} + x^{(k)}_i\boldsymbol{k}, \tag{20}$$

$$QV(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i - QE(\boldsymbol{x}))(\boldsymbol{x}_i - QE(\boldsymbol{x}))^*. \tag{21}$$

Using these quantities, a batch normalization $QBN(\boldsymbol{x}_i)$ is defined as

$$QBN(\boldsymbol{x}_i) = \gamma\left(\frac{\boldsymbol{x}_i - QE(\boldsymbol{x})}{\sqrt{QV(\boldsymbol{x}) + \varepsilon}}\right) + \boldsymbol{\beta}, \tag{22}$$

where γ is a scalar called stretch scale, $\boldsymbol{\beta}$ is a pure imaginary quaternion called shift scale, respectively, and ε is a non-zero small constant. The parameters γ and $\boldsymbol{\beta}$ will change through training a network.

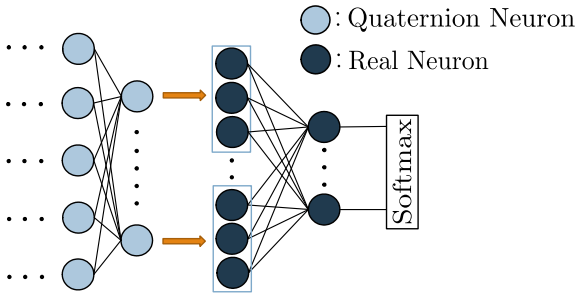

Fig. 2. Real-valued transformation for softmax operation in quaternionic neural network

D. Softmax operation

In classification tasks by neural networks, softmax operation is useful to obtain the classification results, because the outputs of neurons through softmax operation can be interpret as the probabilities or confidences for the classification. It is not applicable to introduce this softmax operation into quatenrionic neurons in a straightforward way, due to three-dimensional output being available in these neurons. Thus, it is necessary to introduce some procedures to implement softmax operation in quaternionic neural network.

In our model, we adopt a scheme for replacing quaternionic neurons to real-valued neurons at the output layer, in order to obtain real-valued signal (see Fig. 2). By using these real-valued outputs, the softmax operation can be applied.

IV. Experimental results

Two types of convolutional neural networks are evaluated for conventional (real-valued) and quaternionic extension networks. The first one is a simple convolutional neural network with two pairs of convolution and pooling layers. The structures of networks are shown in Table.I. These structures are chosen so that the trainable parameters for these networks are almost same. We adopt classification of images in the CIFAR-10 dataset the classification task for these networks. The images (RGB color images with 32 $\times$ 32 pixels) in this dataset are categorized as 10 classes. The numbers of training and testing images are 50,000 and 10,000, respectively. Adam optimizer is used for training networks with 128 of mini-batch size. In training networks, the training dataset is augmented by applying horizontal/vertical shift of 20%, horizontal flips, and $\pm 20^\circ$ rotations.

The transitions of the losses for test images are shown in Fig. 3, and the transitions of classification accuracies are shown in Fig. 4. From these results, we see that the proposed QCNN has better performance than the real-valued CNN.

The second convolutional neural network is so-called ResNet (residual network) [27], where residual blocks and shortcut connections for the block are incorporated. This network is shown as effective for constructing deep neural

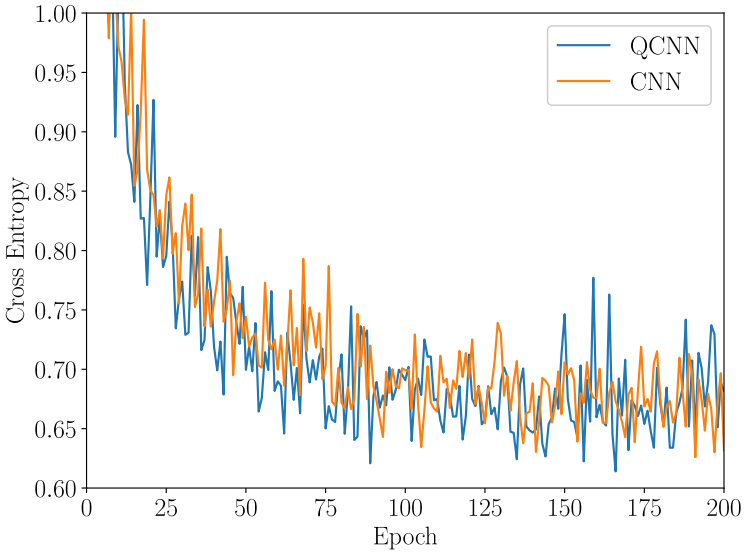

Fig. 3. Transitions of loss for QCNN and CNN

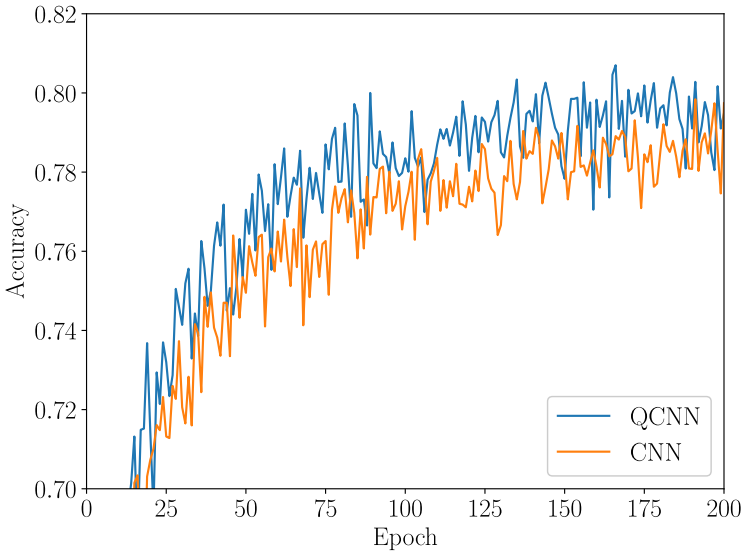

Fig. 4. Transitions of classification accuracies for QCNN and CNN

network with with maintaining the gradient information. Table II shows the structures of real-valued and quaternionic equivalent of ResNet networks, where residual block is represented with brackets. The residual block used in this paper is shown in Fig.5. Other configurations for training networks are the same as those in the experiments for convolutional neural networks, except for the setting of learning rates. The learning rates for this experiment is changed in a phased manner; the learning rate is reduced by 1/10 times at the epochs 150th and 225th.

The transitions of the losses for test images are shown in Fig. 6, and the transitions of classification accuracies are shown in Fig. 7. Sharp changes are observed at the epoch 150th, this is due to the change of learning rate. Also in this experiment, the proposed QCNN performs in the classification task, better than conventional network.

V. Conclusion

We have presented a convolutional neural network model based on quaternionic algebra, a hypercomplex number system. The proposed network deal with three-dimensional signals by using pure imaginary quaternions,

TABLE I
Network architectures on QCNN and CNN.

Layer	Output size		Filter size	Activation
	QCNN	CNN		
Conv1	$3 \times 32 \times 32 \times 16$	$32 \times 32 \times 32$	3×3	ReLU
Maxpool1	$3 \times 16 \times 16 \times 16$	$16 \times 16 \times 32$	2×2	
Conv2	$3 \times 16 \times 16 \times 16$	$16 \times 16 \times 32$	3×3	ReLU
Maxpool2	$3 \times 8 \times 8 \times 16$	$8 \times 8 \times 32$	2×2	
Flatten	3×1024	2048		
FC1	3×256	512		ReLU
FC2	10			Softmax

TABLE II
Network architectures for ResNet34 based on QCNN and CNN. Building blocks are shown in brackets, with the number of blocks stacked.

Layer	Output size		Filter	
	QCNN	CNN	QCNN	CNN
Conv1	$3 \times 32 \times 32 \times 16$	$32 \times 32 \times 32$	$3 \times 3, 16$	$3 \times 3, 32$
Conv2_x	$3 \times 32 \times 32 \times 16$	$32 \times 32 \times 32$	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$
Conv3_x	$3 \times 16 \times 16 \times 32$	$16 \times 16 \times 64$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$
Conv4_x	$3 \times 8 \times 8 \times 64$	$8 \times 8 \times 128$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 6$
Conv5_x	$3 \times 4 \times 4 \times 128$	$4 \times 4 \times 256$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$
GAP	3×128	256		
Dropout	3×128	256		
FC	10			

and affine transformations in three-dimensional space are adopted as operations in neurons. By introducing convolution and pooling layers in quatenrionic neural networks, it would be possible to construct many types of deep neural networks. Thus, two types of convolutional neural networks are constructed in this paper, such as simple convolutional network and Residual network. The performances of the proposed network are evaluated through the classification of color images in CIFAR-10 dataset. Experimental results show that the proposed network works better than the conventional network.

More elaborate investigations on the performances of the proposed network should be conducted for the image datasets larger than CIFAR-10, in order to show the effectiveness of the proposed network. Quaternionic neural networks have a wide variety of configurations such as accumulating input signal due to non-commutable property, definitions of gradients, selection of activation functions. Better performances can be obtained by appropriate choice for these configurations. It is also important to find applications for this type of networks, such as multidimensional signal classification and generation. These also remain for our future work.

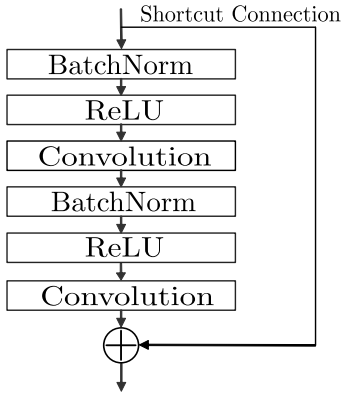

Fig. 5. Residual block used in ResNet-type networks

References

[1] A. Hirose, Ed., Complex-Valued Neural Networks: Theories and Application, ser. Innovative Intelligence. Singapore: World Scientific Publishing, 2003, vol. 5.

[2] A. Hirose, Complex-Valued Neural Networks, ser. Studies in Computational Intelligence. Berlin, Heidelberg: Springer-Verlag, 2006, vol. 32.

[3] T. Nitta, Ed., Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters. Hershey, New York: Information Science Reference, 2009.

[4] A. Hirose, Ed., Complex-Valued Neural Networks: Advances and Applications. Wiley-IEEE Press, 2013.

[5] lan R. Porteous, Clifford algebras and the classical groups. Cambridge University Press, 1995.

[6] P. Lounesto, Clifford algebras and spinors. Cambridge: Cambridge University Press, 2001.

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

[8] T. L. Hankins, Sir William Rowan Hamilton. Baltimore and London: Johns Hopkins University Press, 1980.

[9] R. Mukundan, "Quaternions: From Classical Mechanics to Computer Graphics, and Beyond," in Proceedings of the 7th Asian Technology Conference in Mathematics, 2002, pp. 97–105.

[10] J. B. Kuipers, Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aerospace and Virtual Reality. Princeton: Princeton Univ Press, 1998.

[11] S. G. Hoggar, Mathematics for Computer Graphics. Cambridge: Cambridge University Press, 1992.

[12] T. Nitta, "An Extension of the Back-propagation Algorithm to Quaternions," in Proceedings of International Conference on Neural Information Processing (ICONIP'96), vol. 1, 1996, pp. 247–250.

[13] P. Arena, L. Fortuna, G. Muscato, and M. Xibilia, "Multilayer Perceptrons to Approximate Quaternion Valued Functions," Neural Networks, vol. 10, no. 2, pp. 335–342, 1997.

[14] S. Buchholz and G. Sommer, "Quaternionic spinor MLP," in 8th European Symposium on Artificial Neural Networks (ESANN 2000), 2000, pp. 377–382.

[15] N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura, "Quaternion Neural Network with Geometrical Operators," Journal of Intelligent & Fuzzy Systems, vol. 15, no. 3–4, pp. 149–164, 2004.

[16] D. P. Mandic, C. Jahanchahi, and C. C. Took, "A Quaternion Gradient Operator and Its Applications," IEEE Signal Processing Letters, vol. 18, no. 1, pp. 47–50, 2011.

[17] B. C. Ujang, C. C. Took, and D. P. Mandic, "Quaternion-Valued Nonlinear Adaptive Filtering," IEEE Transactions on Neural Networks, vol. 22, no. 8, pp. 1193–1206, 2011.

[18] T. Isokawa, H. Nishimura, and N.Matsui, "Quaternionic Multilayer Perceptron with Local Analyticity," Information, vol. 3, no. 4, pp. 756–770, 2012.

[19] H. Kusamichi, T. Isokawa, N. Matsui, Y. Ogawa, and K. Maeda, "A New Scheme for Color Night Vision by Quaternion Neural Network," in Proceedings of the 2nd International Conference on Autonomous Robots and Agents (ICARA2004), 2004, pp. 101–106.

[20] T. Isokawa, N. Matsui, and H. Nishimura, "Quaternionic Neural Networks: Fundamental Properties and Applications," in Complex-Valued Neural Networks: Utilizing High-Dimensional

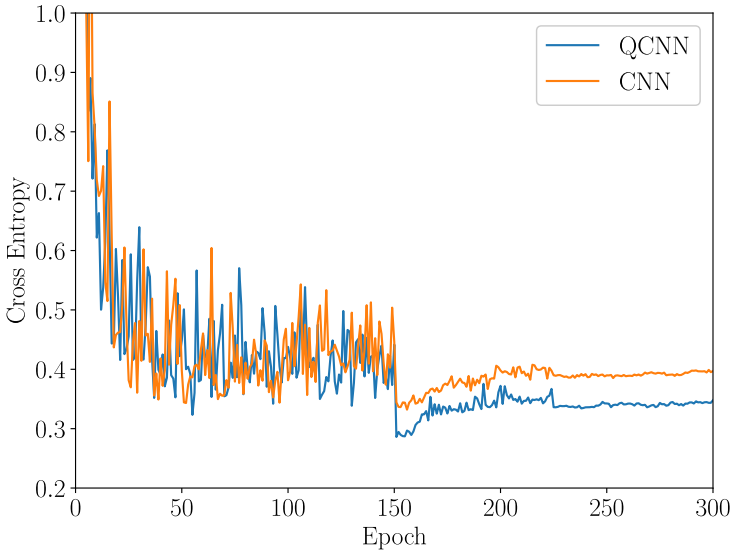

Fig. 6. Transitions of loss for ResNet-type QCNN and CNN

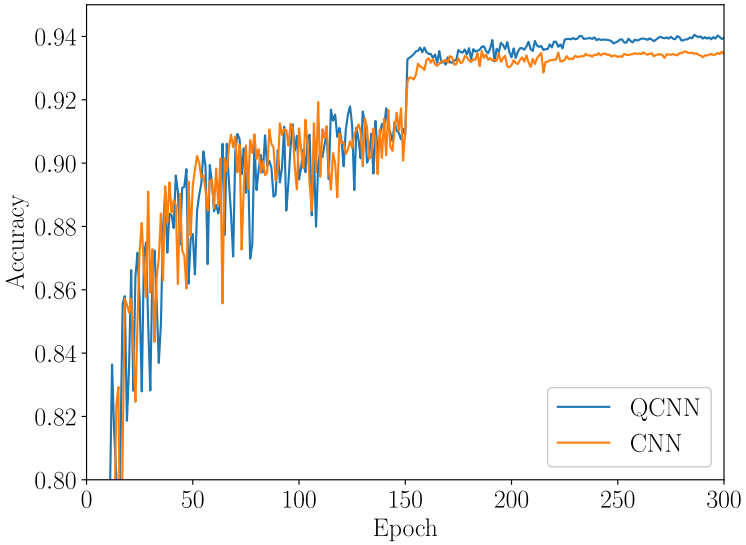

Fig. 7. Transitions of classification accuracies for ResNet-type QCNN and CNN

Parameters, T. Nitta, Ed. Hershey, New York: Information Science Reference, 2009, ch. XVI, pp. 411–439.

[21] C. J. Gaudet and A. S. Maida, "Deep quaternion networks," 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, 2017.

[22] X. Zhu, Y. Xu, H. Xu, and C. Chen, "Quaternion convolutional neural networks," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 631–647.

[23] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linarès, R. D. Mori, and Y. Bengio, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," in INTERSPEECH, 2018.

[24] Q. Yin, J. Wang, X. Luo, J. Zhai, S. K. Jha, and Y. Shi, "Quaternion convolutional neural network for color image classification and forensics," IEEE Access, vol. 7, pp. 20 293–20 301, 2019.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the 32nd International Conference on Machine Learning, vol. 37, 2015, pp. 448–456.

[26] J. Wang, T. Li, X. Luo, Y. Shi, and S. K. Jha, "Identifying computer generated images based on quaternion central moments in color quaternion wavelet domain," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2775–2785, 2019.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.