Pattern Recognition Letters 131 (2020) 56–62

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A bio-inspired quaternion local phase CNN layer with contrast invariance and linear sensitivity to rotation angles

E. Ulises Moya-Sánchez [a,*], Sebastiá Xambó-Descamps [b], Abraham Sánchez Pérez [c], Sebastián Salazar-Colores [d], Jorge Martínez-Ortega [c], Ulises Cortés [b]

[a] Gobierno del Estado de Jalisco y, Universidad Autónoma de Guadalajara, Independencia 55, Guadalajara 44100, Jalisco, México
[b] Universitat Politècnica de Catalunya and Barcelona Supercomputer Centre, Jordi Girona 1–3, Omega, 08034, Barcelona, Spain
[c] Gobierno del Estado de Jalisco, Independencia 55, Guadalajara, 44100 Jalisco, México
[d] Facultad de informática, Universidad Autónoma de Querétaro, Querétaro, México

ARTICLE INFO

Article history:
Received 31 July 2019
Revised 18 November 2019
Accepted 3 December 2019
Available online 4 December 2019

MSC:
41A05
41A10
65D05
65D17

ABSTRACT

Deep learning models have been particularly successful with image recognition using Convolutional Neural Networks (CNN). However, the learning of a contrast invariance and rotation equivariance response may fail even with very deep CNNs or by large data augmentations in training.

We were inspired by the V1 visual features of the mammalian visual system to emulate as much as possible the early visual system and add more invariant capacities to the CNN. We present a new quaternion local phase convolutional neural network layer encoding three local phases. We present two experimental setups: An image classification task with four contrast levels, and a linear regression task that predicts the rotation angle of an image. In sum, we obtain new patterns and feature representations for deep learning, which capture illumination invariance and a linear response to rotation angles.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Previous works indicate that pattern recognition in a deep learning classification process does not guarantee the required equivariance and invariance properties [13,14,26,27].

Convolutional Neural Networks (CNN), which were inspired by neuroscience principles, capture three essential properties of the primary visual cortex *V*1: extraction of features from two-dimensional data; spatial localization of the receptive field; and shift equivariance in the position of the feature [18].

It is to be stressed, however, that visual neurons of the mammalian visual system are also resilient in front of equivariant transformations such as local invariant response to some changes in lighting and rotations [5,18,19].

To emulate as much as possible the early visual system and add more equivariant capacities to the CNN, we have been inspired by some physiological experimental results like those reported in [22], including the stronger response to oriented lines and edges (local even and odd signals, respectively) [5,19].

To move forward toward such functionality, namely to be able to extract contrast invariant local features and to predict rotation angles of images, one idea is to embrace suitable hypercomplex geometric methods, in the sense of [16,20,24,35].

More specifically, in this work we propose what we call a *Quaternion Local Phase* CNN Layer (Q9 in this paper) and explore the performance boost it provides when it is stacked in front of a very simple CNN. What we find is that the compound net features contrast invariance and the capability of ascertaining rotation angles.

The remainder of the paper is structured as follows. In Section 2 we report on recent related work and in Section 3 we recall background notions and notations needed later on. Of particular relevance is Section 3.4, in which we specify our approach to the application of quaternionic local phases to image processing. Our main theoretical contribution, namely the quaternionic local phase layer Q9, is the subject of Section 4. The behaviour of Q9 is also illustrated with images. The experimental setup, including the data used, is described in Section 4 and a summary of the results, conclusions and future outlook can be found in Sections 6 and 7. The material on quaternionic phases used in the paper, particularly in Section 3 and 4, is recalled in the Appendix A.

2. Related work

The use of hypercomplex NN has been mostly in shallow cases: [4,8,9,32].

* Corresponding author.
E-mail address: dr.ulisesmoya@gmail.com (E.U. Moya-Sánchez).

https://doi.org/10.1016/j.patrec.2019.12.001
0167-8655/© 2019 Elsevier B.V. All rights reserved.

On the other hand, [17,36] formulate and implement quaternion convolution, batch normalization, weight initialization, and backpropagation for a deep quaternion CNN. The main difference between these references and our work is that we use the local quaternion phases. In addition, we propose only one CNN layer, not a full quaternion CNN. In fact, our experimental setup and CNN architectures have different objectives. Their CNNs tests aim at classification tasks and do not consider contrast invariance or linear response to rotations. Our CNNs architecture is designed to favor a systematic comparison with the performance of a traditional convolutional layer. To note that we care for maintaining the accuracy across all contrast degradation levels and not merely for getting the best accuracy for one degradation.

3. Background

3.1. Equivariance and invariance

The term equivariance tends to be used to refer to the predictable way in which features of a signal change under certain transformations: [14]. More formally, a function f: $X \to X$ is equivariant with respect to a group of transformations G of X if

$$f(g(x)) = g(f(x)), \tag{1}$$

for all $x \in X$ and $g \in G$.

For instance, one of the most important equivariant properties of the mammalian visual system (measured by [22]) is its equivariance under rotations. Subsequently, many authors have been interested in extending this property to NNs and CNNs: [7,13,14,21].

On the other hand, by invariance of a feature map f: $X \to Y$ we understand (see [16]) that

$$f(g(x)) = f(x) \tag{2}$$

for all $x \in X$ and $g \in G$.

3.2. Bio-inspired CNN

For convenience, in this section we borrow from [31, Section II.B; see also the references therein] the main properties of the V1 simple cells:

1. The V1 cells form the first layer of the hierarchical cortical processing.
2. They are insensitive to the color of the light falling on their receptive fields.
3. These neurons respond vigorously only to edges (odd-signal) and lines (even-signal) at a particular spatial direction through the orientation columns.

In this work we use four main bio-inspired tools: the local phase, (Quaternion) Gabor functions, the HSV color space and the artificial neural networks. Altogether, the proposed layer emulates some properties of V1 cells, as detecting edges (odd-signal) and lines (even-signal) at a particular direction. In addition, our approach exhibits translation equivariance and contrast invariance in object recognition.

3.3. Notations

The notations, structure and terminology we are going to use are as in [31, Sections II.C and IV.C]. We define 1D (resp. 2D) *multivectorial signals* as C^1 maps $U \to \mathcal{G}$ from an interval $U \subset \mathbf{R}$ (a region $U \subset \mathbf{R}^2$) into a *geometric algebra* $\mathcal{G}$ (see [35]). For $\mathcal{G} = \mathbf{R}$ ($\mathcal{G} = \mathbf{C}$, $\mathcal{G} = \mathbf{H}$) we say that the signal is *scalar* (*complex*, *quaternionic*). For technical reasons, we also assume that signals are in L^2 (that is, the modulus is square-integrable).

Table 1
Synopsis of notations and conventions.

Symbol	Structure / Meaning
$\mathbf{x}^0$	Array $\mathbf{R}(M, N, P)$ Input signal
$\mathbf{x} = \mathbf{x}^l$	Array $\mathbf{R}(M^l, N^l, P^l)$ l-th layer output
$\mathbf{w} = \mathbf{w}^l$	Array $\mathbf{R}(M^l, N^l, P^l, Q^l)$ l-th layer weights
$\mathbf{b} = \mathbf{b}^l$	Array $\mathbf{R}(M^l, N^l, P^l)$ l-th layer bias
σ	$\sigma: \mathbf{R} \to \mathbf{R}$ Activation function
C	C: $\mathbf{R}(M, N, P) \to \mathbf{R}(M', N', P')$ Convolutional layer
F	F: $\mathbf{R}(M, N) \to \mathbf{R}(M', N')$ Fully Connected layer
Q	Q: $\mathbf{R}(M, N, P) \to \mathbf{H}(M, N, P)$ $\mathbf{x} \to \mathbf{x}_Q$ (see Eqs. (5) and (6))
$Q9$	Q9: $\mathbf{R}(M, N, P) \to \mathbf{R}(M, N, 9 \cdot P)$ $\mathbf{x} \to [\phi_{RGB}, \theta_{RGB}, \psi_{RGB}]$ (see Eqs. (10)–(12))

Table 1 presents a synopsis of the notation, structure and meaning of the most important variables and functions concurring in a deep CNN. See also [2,18,33].

3.4. Quaternion local phase

A quaternion local phase array $\mathbf{x}_Q(x, y) \in \mathbf{H}$ is associated to a 2D signal $\mathbf{x} = \mathbf{x}(x, y) \in \mathbf{R}$ (where $x, y \in U$, U a region of $\mathbf{R}^2$). The computation of $\mathbf{x}_Q$ is inspired on the version of the Quaternion Fourier Transform (QFT) proposed in [11].[1] To explain how it works, it is convenient to introduce a few notations.

A Gaussian filter $g(u_1, u_2)$ in the frequency domain, rotated by an angle α and with standard deviations σ_1 and σ_2 in the u_1 and u_2 directions, respectively, is defined by the formula

$$g(u_1, u_2) = \exp\left(-\frac{u_1'^2}{2\sigma_1^2} - \frac{u_2'^2}{2\sigma_2^2}\right) \tag{3}$$

where $u_1' = u_1\cos(\alpha) + u_2\sin(\alpha)$ and $u_2' = -u_1\sin(\alpha) + u_2\cos(\alpha)$.

Given a function $f(u_1, u_2)$, which we regard as a filter in the frequency domain, we will also use the expression

$$\mathbf{x} \star f(u_1, u_2) = \mathcal{F}^{-1}((\mathcal{F}\mathbf{x}) \cdot f(u_1, u_2)), \tag{4}$$

where $\mathcal{F}$ denotes the Fourier transform from the *space* variables (x, y) to the *frequency* variables (u_1, u_2).

Now letting ω_1 and ω_2 denote positive constant frequencies, $\mathbf{x}_Q$ is defined as follows:

$$\mathbf{x}_Q = \mathbf{x}_{Qcc} + i\mathbf{x}_{Qsc} + j\mathbf{x}_{Qcs} + k\mathbf{x}_{Qss} \tag{5}$$

where

$$\mathbf{x}_{Qcc} = \mathbf{x} \star g(u_1, u_2)\cos(\sigma_1\omega_1 u_1)\cos(\sigma_2\omega_2 u_2)$$
$$\mathbf{x}_{Qsc} = \mathbf{x} \star g(u_1, u_2)\sin(\sigma_1\omega_1 u_1)\cos(\sigma_2\omega_2 u_2)$$
$$\mathbf{x}_{Qcs} = \mathbf{x} \star g(u_1, u_2)\cos(\sigma_1\omega_1 u_1)\sin(\sigma_2\omega_2 u_2)$$
$$\mathbf{x}_{Qss} = \mathbf{x} \star g(u_1, u_2)\sin(\sigma_1\omega_1 u_1)\sin(\sigma_2\omega_2 u_2)$$

As a result, we have rotated Quaternion Gabor filters (see [11]) which also depend on the parameters α, σ_1, σ_2, ω_1, ω_2. The Gabor filters are well known bio-inspired feature extractors: [11,28–30,32]. See Fig. 1 for an illustration.

Now based on the Eqs. (A.7)–(A.10) of Appendix A, we can rewrite the equation 5 as follows:

$$\mathbf{x}_Q = |\mathbf{x}_Q| e^{i\phi_{\mathbf{x}}} e^{k\psi_{\mathbf{x}}} e^{j\theta_{\mathbf{x}}}, \tag{6}$$

[1] See [3] for a Clifford Fourier Transform generalization.

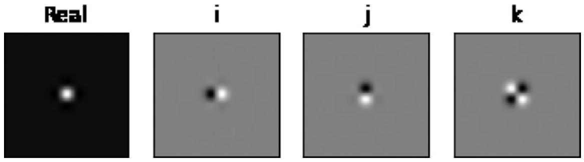

Fig. 1. Quaternion filters from Eq. (5). From left to right $g(u_1, u_2)\cos(\sigma_1\omega_1 u_1)\cos(\sigma_2\omega_2 u_2)$, $ig(u_1, u_2)\sin(\sigma_1\omega_1 x)\cos(\sigma_2\omega_2 y)$, $jg(u_1, u_2)\cos(\sigma_1\omega_1 x)\sin(\sigma_2\omega_2 y)$, $kg(u_1, u_2)\sin(\sigma_1\omega_1 x)\sin(\sigma_2\omega_2 y)$.

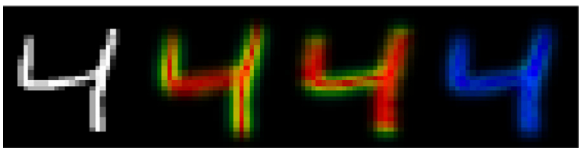

Fig. 2. From left to right: Original image, ϕ_{RGB}, θ_{RGB}, ψ_{RGB}.

where $(\phi_{\mathbf{x}}, \psi_{\mathbf{x}}, \theta_{\mathbf{x}})$ are the phases of the unit quaternion $\mathbf{x}_Q/|\mathbf{x}_Q|$.

It is important to remark that the frequency domain convolution is scarcely used on CNN. There are two main reasons for this: i) for N training weights, the complexity is of order N^2; ii) the Fourier transform is a global transformation, and it is not possible to correctly localize the features [6]. Previous works such as [34] try to use the cosine transform in order to avoid the local problem and [15] propose a *Spectrum pooling* and return to the spatial domain to avoid both drawbacks for 3D data.

4. Quaternion local phase layer *Q9*

The proposed convolution layer, *Q9*, a quaternion local phase layer, is described as follows:

1. Create a Hue Saturation Value (HSV) array using the quaternion phases, magnitude and constant **1**:

$$\phi_{HSV} = (\phi_{\mathbf{x}}, |\mathbf{x}_Q|, \mathbf{1}), \tag{7}$$

$$\theta_{HSV} = (\theta_{\mathbf{x}}, |\mathbf{x}_Q|, \mathbf{1}), \tag{8}$$

$$\psi_{HSV} = (\psi_{\mathbf{x}}, |\mathbf{x}_Q|, \mathbf{1}). \tag{9}$$

   where **1** is an $[m, n]$ array with all entries equal to 1.
2. Transform the HSV images into RGB images:

$$\phi_{HSV} \to \phi_{RGB}, \tag{10}$$

$$\theta_{HSV} \to \theta_{RGB} \tag{11}$$

$$\psi_{HSV} \to \psi_{RGB} \tag{12}$$

   according to the standard conventions (cf. [1], p. 304). See Fig. 2 for an illustration. Remark that ϕ_{RGB} enhances vertical lines (yellow), that θ_{RGB} enhances horizontal lines (yellow), and that ψ_{RGB} features a dark blue all over the image. Fig. 3 illustrates the nine feature maps of ϕ_{RGB}, θ_{RGB} and ψ_{RGB}.
   In our experimental setup, we notice that the phase ψ_{RGB} is sensitive to rotation. See Fig. 4 for an illustration of the three channels of ψ_{RGB} and their response to rotations.

In the Table 2 we can compare a simple convolution layer with the proposed layer *Q9*

5. Data and experimental setup

We have used the MNIST dataset, see [25] and CIFAR10 [23], in two main experimental setups: contrast invariance classification and regression for equivariance response measurement.

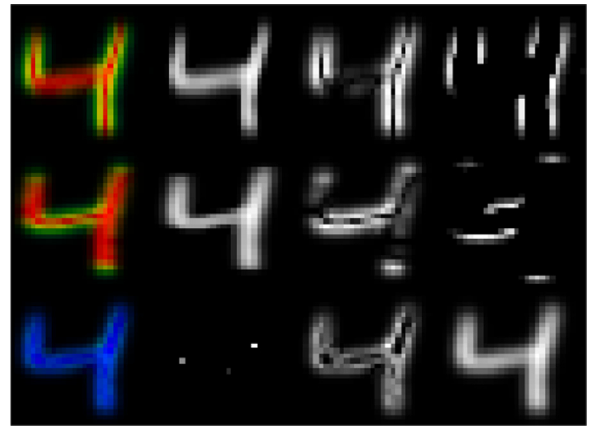

Fig. 3. In the first column ϕ_{RGB}, θ_{RGB} and ψ_{RGB}. Each row presents the three components of each *RGB* phase representation.

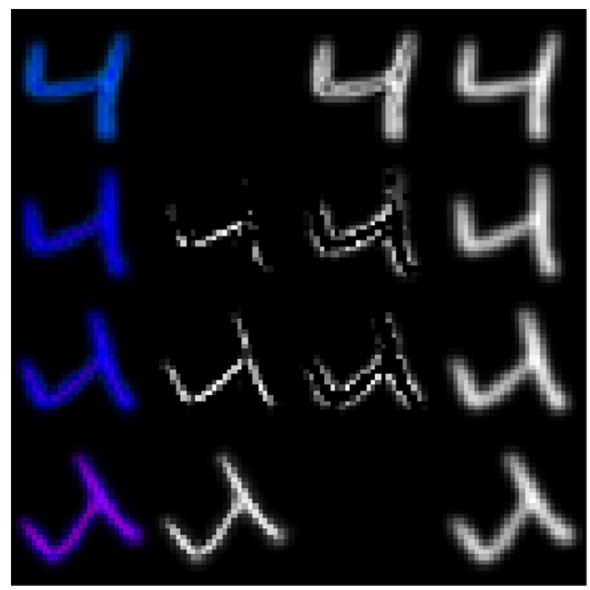

Fig. 4. In the first column ψ_{RGB} with 0, 15, 30 and 45 degrees. Each row depicts the components of each ψ_{RGB}.

Table 2
Comparison of the main characteristics of a standard convolutional layer *C* and our layer *Q9*. Although *Q9* is not itself trainable, the learning is carried out, in any particular application, by the net to which it is coupled.

Convolution Layer Comparison		
Characteristics	*C*	*Q9*
Hyperparameters	3	5
	Kernel shape	α, σ_1, σ_2, ω_1, ω_2
Convolution domain	space	frequency
Output domain	space	space
Padding	zero	–
Nonlinear function	ReLU	arctan, arcsin
Layer position	Any	First hidden
Learning	Yes	No

5.1. Contrast invariance classification

Table 3 shows the main characteristics of the degradation labels and the contrast values. The contrast reduction process was done by first normalizing the pixel values from 0 to 1, and then rescaling the pixel values to an interval $[d, 1] \subset [0, 1]$, which amounts to a contrast of $(1 - d)$. For instance, in the case d_1 the pixel values are rescaled to the interval [0.3,1], which amounts to a 70% contrast.

Table 3
Characteristics of datasets (*top*). Degradation labels and the corresponding contrast values used in our experiments (*bottom*).

Characteristics	MNIST Values	CIFAR10 Values
Training set	50,000	40,000
Validation set	10,000	10,000
Test set	10,000	10,000
Total of images	70,000	60,000
Image shape	[28,28,1]	[32,32,3]
Degradation	Contrast (%)	
d_0	100	
d_1	70	
d_2	30	
d_3	10	

Table 4
CNN Architectures for MNIST and CIFAR10 classification task. The C layer and all the F layers have a ReLU as activation function.

CNN classification architecture		
Layer	Characteristics	
(0) Input Layer	Input Image	
Shape	[28,28,1]or [32,32,3]	
(1) First layer	C	Q9
Parameters	[3, 3, 9]	[28,28,4]
Output	[28,28,9]	[28,28,9]
	Parameters Shape	
(2) FL	[7056]	
(3) F 1	[256]	
(4) F 2	[128]	
(5) SMAX	[10]	

Fig. 5. Example of degradation levels. From left to right: 100% contrast (d_0); 70% (d_1); 30% (d_2); and 10% (d_3).

Fig. 5 provides an illustration of the degradation levels d_0, d_1, d_2, d_3 for MNIST and CIFAR10 images.

We trained and tested two very simple CNNs, one with a regular convolutional layer (C) and another with the Q9 layer. Both CNNs have the same architecture, namely one convolutional layer (C), a flatten layer (FL), three fully connected layers (F) and a softmax function (SMAX). See Table 4 for more details.

Both CNNs were trained with the same hyperparameters: learning rate, 0.001; number of epochs, 100; loss function, categorical cross-entropy; and optimizer, ADAM; for Q9, $\alpha = 0, \sigma_1 = 4, \sigma_2 = 4, \omega_1 = 0.5, \omega_2 = 0.5$. All source code is available at https://github.com/asp1420/A-Bio-inspired-Quaternion-Local-Phase-CNN. It is written in the Tensorflow framework with Keras as wrapper (see [12]).

The experimental setup is organized according to the scheme summarized in Table 5, where d_i $(i = 0, 1, 2, 3)$ means a degradation degree. It is important to note that we tested the generalization capacity of each of the trained models by running them not only on the corresponding degradation level set, but also on the three modified versions of it.

Table 5
Experimental scheme for the classification task.

Experimental set up	
Trained and Contrast	Tested
d_0	d_0, d_1, d_2, d_3
d_1	d_0, d_1, d_2, d_3
d_2	d_0, d_1, d_2, d_3
d_3	d_0, d_1, d_2, d_3

Table 6
Experimental scheme for rotation for each image.

Experimental rotation set up	
Set	Angles (Degree)
Train	[0, 46] steps=3
Validation	[1, 46] steps=3
Test	[2, 46] steps=3

Table 7
CNN Architectures for a regression task to get the rotation angle. The *C* and all the *F* layers have a sigmoid as activation function.

CNN regression architecture		
Layer	Characteristics	
(0) Input Layer	Image	
Shape	[28,28,1] or [32,32,3]	
(1) First layer	*C*	Q9
Parameters	[3, 3, 9]	[28,28,4]
Output	[28,28,9] or [32,32,9]	[28,28,9] or [32,32,9]
	Parameters Shape	
(2) FL	[7056]	
(3) F 1	[1000]	
(4) F 2	[64]	
(5) F 3	[32]	
(6) F 4	[8]	
(7) SG	[1]	

5.2. Regression for rotational response

We created 100 data-sets (one for each image), by rotating according to Table 6. We trained, validated and tested with the first 100 (nonzero digits) from the MNIST and the first 100 images from CIFAR10. Fig. 6 presents an example of rotation data.

We have used the CNN architecture presented in Table 7. We trained both CNNs for regression with the same hyperparameters: learning rate, 0.001; number of epochs, 1000; loss function, mean

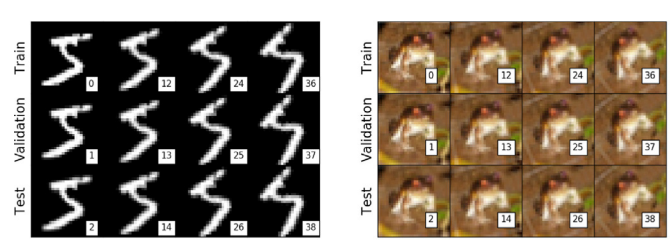

Fig. 6. Rotation examples for regression experimental setup from MNIST and CIFAR10. The value of the angle in degrees.

Table 8
Experimental results on the classification task for all degradation labels on MNIST images. The last row displays the average accuracy.

Train	Test							
	d_0		d_1		d_2		d_3	
	C	Q9	C	Q9	C	Q9	C	Q9
d_0	**98.7**	98.4	92.8	**98.4**	10.8	**98.3**	8.9	**98.3**
d_1	87.0	**98.5**	98.4	**98.5**	11.4	**98.4**	11.4	**98.5**
d_2	86.1	**98.5**	91.5	**98.5**	98.3	**98.5**	21.0	**98.5**
d_3	81.6	**98.4**	82.1	**98.4**	91.7	**98.4**	97.8	**98.4**
Avg	88.3	**98.4**	91.2	**98.4**	53.0	**98.4**	34.8	**98.4**

Table 9
Experimental results on the classification task for all degradation labels with CIFAR10. The last row displays the average accuracy.

Train	Test							
	d_0		d_1		d_2		d_3	
	C	Q9	C	Q9	C	Q9	C	Q9
d_0	**57.1**	47.2	36.5	**46.6**	15.7	**45.7**	10.0	**45.3**
d_1	41.5	**46.9**	**56.9**	47.3	25.6	**47.3**	10.0	**46.9**
d_2	23.0	**43.2**	30.2	**45.9**	**54.5**	47.1	23.6	**47.2**
d_3	13.6	**38.2**	16.3	**42.5**	27.0	**46.8**	43.8	**47.5**
Avg	33.8	**43.9**	34.9	**45.6**	30.7	**46.7**	21.9	**46.7**

squared error; optimizer, *rmsprop*; activation function, sigmoid; and for Q9 we chose $\alpha = 0, \sigma_1 = 4, \sigma_2 = 4, \omega_1 = 0.5, \omega_2 = 0.5$.

6. Results and analysis

6.1. Contrast invariance (classification)

Tables 8 and 9 show the best accuracy reached in the classification tasks by the C and Q9 nets for MNIST and CIFAR10, respectively. Each row shows the performance of both CNNs with different testing samples for a given training sample. As mentioned previously, samples are grouped according to the different contrast levels. The values in bold indicate the best result. As seen in Table 8, the superiority of Q9 over C is conspicuous on MNIST images for its high and virtually uniform accuracy. The gains for CIFAR10 images are also quite uniform and, except for the diagonal slots $d_0 - d_0$, $d_1 - d_1$ and $d_2 - d_2$, very appreciable (Table 9). In sum, Q9 exhibits invariance under contrast degradation while C does not.

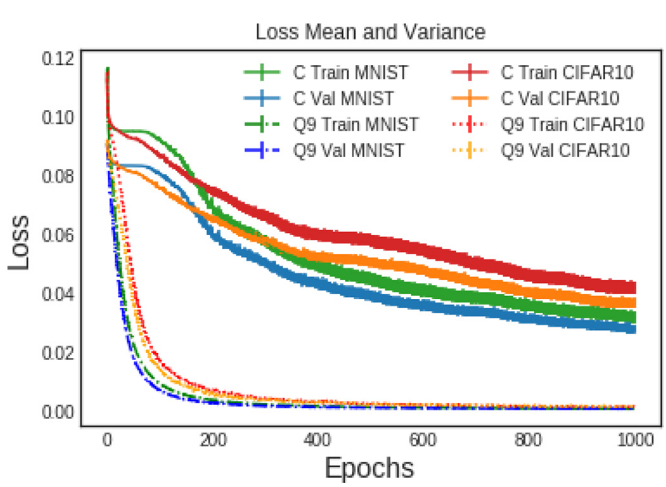

Fig. 7. Error plot with the mean value per epoch of the loss function in the training process for each layer. We again see the superior performance of Q9 versus C. The narrow curves show that variance for Q9 is much smaller than for C.

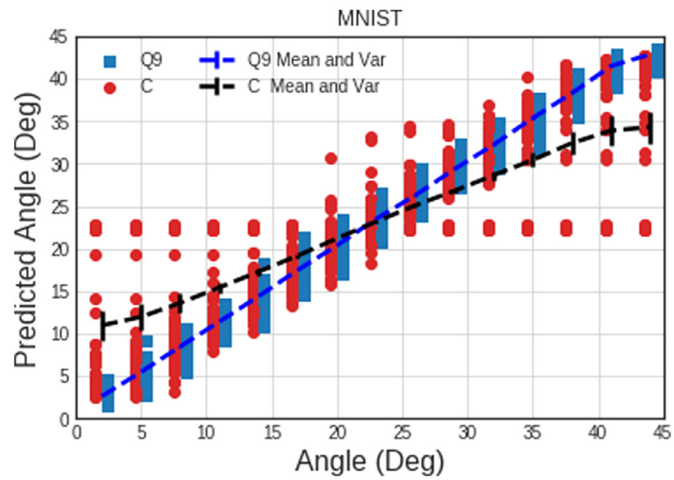

Fig. 8. Prediction of the rotation angle in steps of 3 degrees, from 0 to 45, over 100 MNIST images at epoch 1000. Note the low variance for Q9.

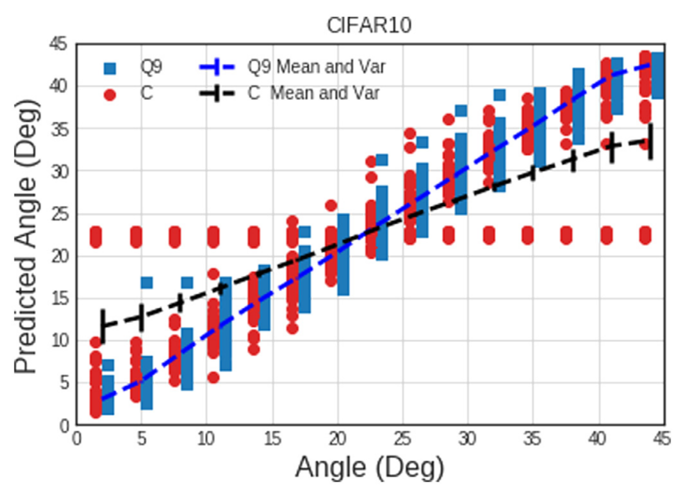

Fig. 9. Prediction of the rotation angle in steps of 3 degrees, from 0 to 45, over 100 CIFAR10 images at epoch 1000.

6.2. Rotational response (regression)

The first comparison we can try on the performance of C and Q9 is the behaviour of the loss function. Fig. 7 is an error plot to show the evolution of the mean value and the variance (as vertical line of each point) of the loss function, respectively, for both CNNs across epochs. These figures show that the loss function converges faster, and that it is more stable, for Q9 than for C. Faster convergence and higher stability are desirable properties that have a bearing, in particular, on a more efficient use of computing resources.

Figs. 8 and 9 show the rotation predictions over the 100 images of each CNN at epoch 1000. In addition a mean value of the prediction and its variance are represented by an error curve. We see that the CNN with the normal convolution layer C has more outliers in the ranges (0–20) and (25–45) degrees than Q9. In other words, the dispersion of the predictions is substantially lower for Q9 than for C. This result motivates us to explore in a future work whether Q9, or some variation of it, achieves an equivariant response with respect to plane rotations.

7. Conclusions and outlook

The main objective of the article has been to propose a new bio-inspired quaternionic layer, Q9, based on the kernel of the quaternionic Fourier transform proposed by Thomas Bülow [11] and to compare its performance to that of a regular convolutional layer in two types of computational experiments, one focused on classification tasks and another on rotation prediction by regression. Our layer has 9 channels and insures an invariant response to high contrast changes with almost constant performance in classification tasks even when the CNN is trained with quite different contrasts. The Q9 layer also features a faster learning of image rotation angles than a regular convolution layer. We believe that the proposed layer could be useful to recognize or classify images in outdoor scenarios with haze with no data augmentation. One current drawback is that the *C* layer has a better performance for CIFAR10 images in the case when the test set has the same degradation level (with the exception of d_3) than the training set.

Let us mention two lines of future inquiry: To seek how to insure that the hyperparameters of the layer are learned automatically, and to explore how to obtain an equivariant response to image rotations.

Declaration of Competing Interest

I recognizing and disclosing that we don't have any financial and other conflicts of interest that might bias this work. We acknowledge all financial support for the work. As a result, I declare that we don't have any conflict of interest related to this work.

Acknowledgment

The authors would like to thank to CONACYT and the Barcelona supercomputing Center. Sebastián Salazar-Colores (CVU 477758) would like to thank CONACYT (Consejo Nacional de Ciencia y Tecnología) for the financial support of his PhD studies under Scholarship 285651. Eduardo Ulises Moya-Sánchez, Jorge Martínez-Ortega and Ulises Cortés are members of the Sistema Nacional de Investigadores (SNI) de México.

Appendix A. Quaternions

The quaternion algebra $\mathbf{H}$ is a four dimensional real vector space with basis 1, *i, j, k*,

$$\mathbf{H} = \mathbf{R}1 \oplus \mathbf{R}i \oplus \mathbf{R}j \oplus \mathbf{R}k \tag{A.1}$$

endowed with the bilinear product (multiplication) defined by Hamilton's relations, namely

$$i^2 = j^2 = k^2 = ijk = -1. \tag{A.2}$$

As it is easily seen, these relations imply that

$$ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j. \tag{A.3}$$

The elements of $\mathbf{H}$ are called *quaternions*, and *i, j, k*, *quaternionic units*. By definition, a quaternion q can be written in a unique way in the form

$$q = a + bi + cj + dk, \quad a, b, c, d \in \mathbf{R}. \tag{A.4}$$

Its *conjugate*, $\bar{q}$, is defined as

$$\bar{q} = a - (bi + cj + dk), \tag{A.5}$$

and its *modulus*, $|q|$, by $|q| = \sqrt{q\bar{q}}$.

A polar representation of q is defined by

$$q = |q| e^{i\phi} e^{k\psi} e^{j\theta}, \tag{A.6}$$

where

$$(\phi, \theta, \psi) \in [-\pi, \pi) \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right) \times \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$$

are the phases of q as defined in [10] and [11]. For a unit q, the phase ψ is found to be

$$\psi = -\frac{\arcsin(2(bc - ad))}{2}. \tag{A.7}$$

If $\psi = \pm\frac{\pi}{4}$, set $\theta = 0$ and

$$\phi' = \frac{1}{2} \arctan 2\left(\frac{2(-cd + ab)}{a^2 - b^2 - c^2 + d^2}\right). \tag{A.8}$$

Else,

$$\phi' = \frac{1}{2} \arctan 2\left(\frac{2(cd + ab)}{a^2 - b^2 + c^2 - d^2}\right), \tag{A.9}$$

$$\theta = \frac{1}{2} \arctan 2\left(\frac{2(bd + ab)}{a^2 + b^2 - c^2 - d^2}\right). \tag{A.10}$$

With this, $e^{i\phi'} e^{k\psi} e^{j\theta} = \pm q$ and if it is $-q$, set $\phi = \phi' + \pi \mod 2\pi$. Here, arctan 2 is the four quadrant arctan (see [11]).

References

[1] M.K. Agoston, Computer Graphics and Geometric Modeling, 1, Springer, 2005.
[2] S. Arora, Mathematics of machine learning: An introduction, in: Proceedings of the International Congress of Mathematicians 2018 (Rio de Janeiro), Vol. 1, World Scientific, 2019, pp. 377–390.
[3] T. Batard, M. Berthier, Spinor fourier transform for image processing, IEEE J. Sel. Top. Signal Process. 7 (2013) 605–613.
[4] E. Bayro-Corrochano, E. Vazquez-Santacruz, E. Moya-Sanchez, E. Castillo-Muñis, Geometric bioinspired networks for recognition of 2D and 3D low-level structures and transformations, IEEE Trans. Neural Netw. Learn. Syst. 27 (2016) 2020–2034.
[5] J. Bigun, Vision with Direction, Springer, 2006.
[6] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond euclidean data, IEEE Signal Process. Mag. 34 (2017) 18–42.
[7] J. Bruna, S. Mallat, Invariant scattering convolution networks, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1872–1886.
[8] S. Buchholz, A theory of neural computation with Clifford algebras, Christian-Albrechts Universität Kiel, 2005.
[9] S. Buchholz, G. Sommer, Quaternionic spinor MLP, in: Proceeding of 8th European Symposium on Artificial Neural Networks (ESANN 2000: Bruges, Belgium, 26–28 April), Citeseer, 2000, pp. 377–382.
[10] T. Bülow, Hypercomplex spectral signal representations for the processing and analysis of images, Christian-Albrechts-Universitat zu Kiel, 1999.
[11] T. Bülow, G. Sommer, Hypercomplex signals – a novel extension of the analytic signal to the multidimensional case, IEEE Trans. Signal Process. 49 (2001) 2844–2852.
[12] F. Chollet, et al., Keras, 2015, (https://github.com/keras-team/keras).
[13] T. Cohen, M. Welling, Group equivariant convolutional networks, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 2990–2999.
[14] S. Dieleman, J. De Fauw, K. Kavukcuoglu, Exploiting cyclic symmetry in convolutional neural networks, in: Proceedings of the 33rd International Conference on Machine Learning, PMLR 48, 2016, pp. 1889–1898.
[15] C. Esteves, C. Allen-Blanchette, A. Makadia, K. Daniilidis, Learning so (3) equivariant representations with spherical CNNS, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 52–68.
[16] M. Felsberg, Low-level image processing with the structure multivector, 203, Institut für Informatik und Praktische Mathematik, 2002.
[17] C. Gaudet, A. Maida, Deep quaternion networks, in: Proceeding of the 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.
[18] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org
[19] G.H. Granlund, H. Knutsson, Signal Processing for Computer Vision, Springer, 2013.
[20] D. Hestenes, G. Sobczyk, Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics, 5, Springer, 2012.
[21] G. Hinton, A. Krizhevsky, N. Jaitly, T. Tieleman, Y. Tang, Does the brain do inverse graphics? Brain and Cognitive Sciences Fall Colloquium, 2012.
[22] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. Physiol. 160 (1962) 106–154.
[23] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images.Technical Report, Citeseer, 2009.
[24] C. Lavor, S. Xambó-Descamps, I. Zaplana, A Geometric Algebra Invitation to Space-Time Physics, Robotics and Molecular Geometry, SBMA/Springerbrief, Springer, 2018.

[25] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.
[26] K. Lenc, A. Vedaldi, Understanding image representations by measuring their equivariance and equivalence, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 991–999.
[27] S. Mallat, Understanding deep convolutional networks, Philos. Trans. R. Soc. A 374 (2016) 20150203.
[28] E.U. Moya-Sánchez, E. Bayro-Corrochano, Quaternion atomic function wavelet for applications in image processing, in: Proceedings of the Iberoamerican Congress on Pattern Recognition, Springer, 2010, pp. 346–353.
[29] E.U. Moya-Sánchez, E. Bayro-Corrochano, Symmetry feature extraction based on quaternionic local phase, Adv. Appl. Clifford Algeb. 24 (2014) 333–354.
[30] E.U. Moya-Sánchez, S. Salazar-Colores, A. Sánchez, U. Cortés, Quaternion phase CNN, in: Proceedings of the Artificial Intelligence Research and Development - Current Challenges, New Trends and Applications, CCIA 2018, 21st International Conference of the Catalan Association for Artificial Intelligence, Alt Empordá, Catalonia, Spain, 8-10th October 2018, 2018, pp. 89–95. Doi: 10.3233/978-1-61499-918-8-89
[31] E.U. Moya-Sánchez, S. Salazar Colores, S. Xambó-Descamps, U. Cortés, A bio-inspired monogenic CNN layer for illumination-contrast invariance, 2019. https://mat-web.upc.edu/people/sebastia.xambo/Papers/M6.pdf.
[32] E.U. Moya-Sánchez, E. Vázquez-Santacruz, A geometric bio-inspired model for recognition of low-level structures, in: Proceedings of the International Conference on Artificial Neural Networks, Springer, 2011, pp. 429–436.
[33] G. Strang, Linear Algebra and Learning From Data, Wellesley-Cambridge Press, 2019.
[34] Y. Wang, C. Xu, S. You, D. Tao, C. Xu, Cnnpack: Packing convolutional neural networks in the frequency domain, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 253–261.
[35] S. Xambó-Descamps, Real spinorial groups–a short mathematical introduction, SBMA/Springerbrief, Springer, 2018.
[36] X. Zhu, Y. Xu, H. Xu, C. Chen, Quaternion convolutional neural networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 631–647.