Quaternion convolutional neural networks for heterogeneous image processing

Titouan Parcollet, Mohamed Morchid, Georges Linarès

QUATERNION CONVOLUTIONAL NEURAL NETWORKS FOR HETEROGENEOUS IMAGE PROCESSING

Titouan Parcollet[1,2]*, Mohamed Morchid*[1]*, Georges Linarès*[1]

[1]Université d'Avignon, LIA, France
[2]Orkis, Aix en provence, France

ABSTRACT

Convolutional neural networks (CNN) have recently achieved state-of-the-art results in various applications. In the case of image recognition, an ideal model has to learn independently of the training data, both local dependencies between the three components (R,G,B) of a pixel, and the global relations describing edges or shapes, making it efficient with small or heterogeneous datasets. Quaternion-valued convolutional neural networks (QCNN) solved this problematic by introducing multidimensional algebra to CNN. This paper proposes to explore the fundamental reason of the success of QCNN over CNN, by investigating the impact of the Hamilton product on a color image reconstruction task performed from a gray-scale only training. By learning independently both internal and external relations and with less parameters than real valued convolutional encoder-decoder (CAE), quaternion convolutional encoder-decoders (QCAE) perfectly reconstructed unseen color images while CAE produced worst and gray-scale versions.

Index Terms— Quaternion convolutional encoder-decoder, convolutional neural networks, heterogeneous image processing

1. INTRODUCTION

Neural network models are at the core of modern image recognition methods. Among these models, convolutional neural networks [1](CNN) have been developed to consider both basic and complex patterns in images, and achieved top of the line results in numerous challenges [2]. Nonetheless, in the specific case of image recognition, a good model has to efficiently encode local relations within the input features, such as between the Red, Green, and Blue (R,G,B) channels of a single pixel, as well as structural relations, such as those describing edges or shapes composed by groups of pixels. In particular, traditional real-valued CNNs consider pixels as three different and separated values (R, G, B), while a more natural representation is to process a pixel as a single multidimensional entity. More precisely, both internal and global hidden relations are considered at the same level during the training of CNNs.

Thereby, and strong of many applications [3, 4, 5], quaternion neural networks [6, 7, 8] (QNN) have been proposed to encapsulate multidimensional input features. Quaternions are hyper-complex numbers that contain a real and three separate imaginary components, fitting perfectly to three and four dimensional feature vectors, such as for image processing. Indeed, the three components (R,G,B) of a given pixel are embedded in a quaternion, to create and process pixels as entities. With the purpose to solve the above described problem of local and global dependencies, deep quaternion convolutional neural networks [9, 10, 11] (QCNN) have been proposed. In the previous works, better image classification results than real-valued CNN are obtained with smaller neural networks in term of number of parameters. The authors claim that such better performances are due to the specific quaternion algebra, alongside with the natural multidimensional representation of a pixel. Nonetheless, and despite promising results, no clear intuitions of QCNN performances in image recognition have been demonstrated yet. Moreover, these studies employ color images for training and validation sub-processes.

Therefore, the paper proposes: 1) to explore the impact of the *Hamilton product* (Section 2.1), which is at the heart of the better learning and representation abilities of QNN; 2) to show that quaternion-valued neural networks are able to perfectly learn color features dependencies (R,G,B). Quaternion and real-valued neural networks are therefore compared on a gray-scale to color image task that highlights the capability of a model to learn both internal (i.e. the relations that exist inside a pixel) and external relations of an image. In this extent, a quaternion convolutional encoder-decoder (QCAE) (Section 3) [1] and a real-valued convolutional encoder-decoder [12] (CAE) are trained to reconstruct a unique gray-scale image from the KODAK PhotoCD dataset (Section 4.1). During the validation process, an unseen color image is presented to both models, and reconstructed pictures are compared visually and with the peak signal to noise ratio (PSNR) as well as the structural similarity (SSIM) metrics (Section 4.3). To validate the learning of internal dependencies, these models must reconstruct the color image without prior information about

[1]Code is available at https://github.com/Orkis-Research/Pytorch-Quaternion-Neural-Networks

the color space given from the training phase. The experiments show that QCAE succeeds to produce an almost perfect copy of the testing image, while the CAE fails, by reconstructing a slightly worst and black and white version. Such behavior makes quaternion-valued models a better fit to image recognition in heterogeneous conditions. Indeed, quaternion-valued are less harmed by smaller and heterogeneous data, due to their ability to dissociate internal and global dependencies trough the *Hamilton product*, and convolutional process respectively. Finally, it is worth noticing that these performances are observed with a reduction of the number of neural parameters of four times for QCAE compared to CAE.

2. QUATERNION ALGEBRA

The quaternion algebra $\mathbb{H}$ defines operations between quaternion numbers. A quaternion Q is an extension of a complex number defined in a four dimensional space as:

$$Q = r1 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}, \tag{1}$$

where r, x, y, and z are real numbers, and 1, **i**, **j**, and **k** are the quaternion unit basis. In a quaternion, r is the real part, while $x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ with $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$ is the imaginary part, or the vector part. Such a definition can be used to describe spatial rotations.

2.1. Hamilton product

The *Hamilton product* ($\otimes$) is used in QNN to replace the standard real-valued dot product, and to perform transformations between two quaternions Q_1 and Q_2 following:

$$\begin{aligned} Q_1 \otimes Q_2 =& (r_1 r_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\ & (r_1 x_2 + x_1 r_2 + y_1 z_2 - z_1 y_2)\boldsymbol{i} + \\ & (r_1 y_2 - x_1 z_2 + y_1 r_2 + z_1 x_2)\boldsymbol{j} + \\ & (r_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 r_2)\boldsymbol{k}. \end{aligned} \tag{2}$$

The *Hamilton product* allows quaternion neural network to capture internal latent relations within the features of a quaternion (see Figure 1). In the case of a quaternion-valued neural network, the quaternion-weight components are shared through multiple quaternion-input parts during the *Hamilton product* , creating relations within the elements. Indeed, Figure 1 shows that, in a real-valued neural network, the multiple weights required to code latent relations within a feature are considered at the same level as for learning global relations between different features, while the quaternion weight w codes these internal relations within a unique quaternion Q_{out} during the *Hamilton product* (right).

3. QUATERNION CONVOLUTIONAL ENCODER-DECODER

The QCAE is an extension of the well-known real-valued convolutional networks (CNN) [2] and convolutional encoder-decoder [13] to quaternion numbers. Encoder-decoder models are simple unsupervised structures that aim to reconstruct the input feature at the output [12]. In a CAE or QCAE, encoding dense layers are simply replaced with convolutional ones, while decoding dense layers are either changed to transposed or upsampled convolutional layers [14]. In this extent, let us recall the basics of the quaternion-valued convolution process [10, 9]. The latter operation is performed with the real-number matrix representation of quaternions. Therefore, a traditional $1D$ convolutional layer, with a kernel that contains $K \times K$ feature maps, is split into 4 parts: the first part equal to r, the second one to $x\mathbf{i}$, the third one to $y\mathbf{j}$ and the last one to $z\mathbf{k}$ of a quaternion $Q = r1 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. The backpropagation is ensured by differentiable cost and activation functions that have already been investigated for quaternions in [15] and [16]. As a result, the so-called *split* approach [8, 6, 9, 17] is used as a quaternion equivalence of real-valued activation functions. Then, let γ^l_{ab} and S^l_{ab} , be the quaternion output and the pre-activation quaternion output at layer l and at the indexes (a, b) of the new feature map, and w the quaternion-valued weight filter map of size $K \times K$. A formal definition of the convolution process is:

$$\gamma^l_{ab} = \alpha(S^l_{ab}), \tag{3}$$

with

$$S^l_{ab} = \sum_{c=0}^{K-1} \sum_{d=0}^{K-1} w^l \otimes \gamma^{l-1}_{(a+c)(b+d)}, \tag{4}$$

where α is a *quaternion split activation* function [8, 6, 9, 17]. The output layer of a quaternion neural network is commonly either quaternion-valued such as for quaternion approximation [7], or real-valued to obtain a posterior distribution based on a softmax function following the split approach. Indeed, target classes are often expressed as real numbers.

4. EXPERIMENTS AND RESULTS

This section details the experiments (Section 4.1), the models architectures (Section 4.2), and the results (Section 4.3) obtained with both QCAE and CAE on a gray to color task with the KODAK PhotoCD dataset.

4.1. From gray-scale to the color space

We propose first to highlight the ability of a model to learn the internal relations that compose pixels (*i.e.* the color space), and ensures the robustness of the model in heterogeneous training/validation conditions. In this extent, models are trained to compress and reproduce a unique gray-scale image in an encoder-decoder fashion, and are then fed with two different color images at validation time. Models are expected to reproduce the exact same colors than the original test samples. Experiments are based on the KODAK

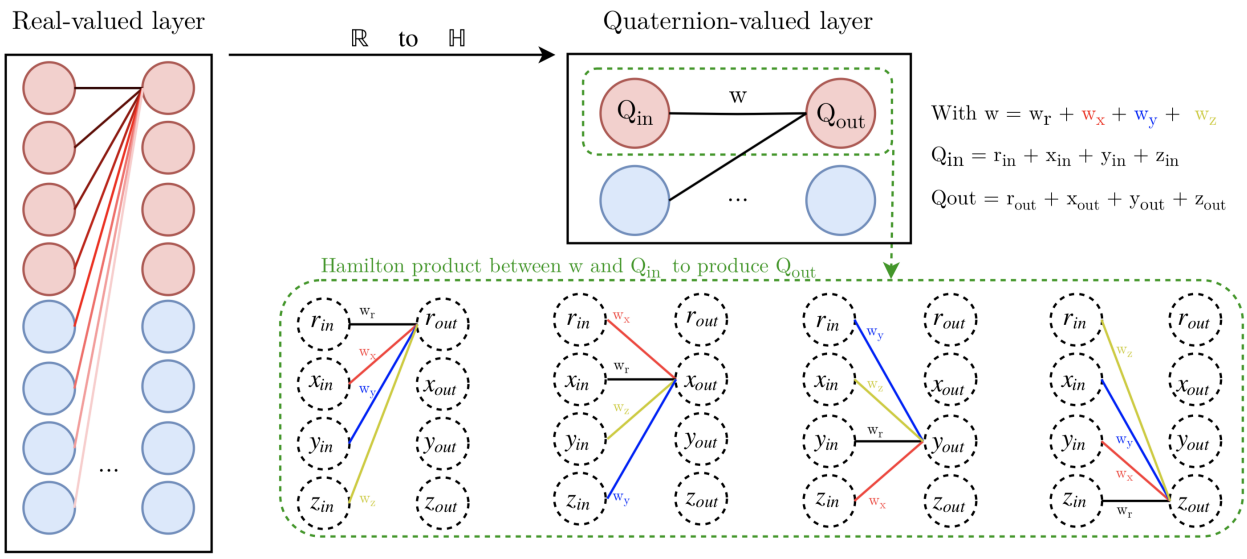

Fig. 1. Illustration of the impact of the *Hamilton product* in a quaternion-valued neural network layer, compared to a traditional real-valued neural network layer

PhotoCD data-set [2]. A random image (See Figure 2) is converted to gray-scale following the basic luma formula [18] and used as the training sample, while the others original color images are used as a validation subset. Therefore, training is performed on a single gray-scale image of 512×768 pixels with the gray value of a given pixel $p_{x,y}$ repeated three times to compose a quaternion $Q(p_{x,y}) = 0 + GS(p_{x,y})\,\mathbf{i} + GS(p_{x,y})\,\mathbf{j} + GS(p_{x,y})\,\mathbf{k}$. For a fair comparison, the gray value is also concatenated three times for each pixel in the real-valued CNN. Finally, the quaternion $Q(p_{x,y}) = 0 + R(p_{x,y})\,\mathbf{i} + G(p_{x,y})\,\mathbf{j} + B(p_{x,y})\,\mathbf{k}$ based on color images is composed and processed at validation time, while R, G, B components are concatenated for CNN. Reconstructed pictures are evaluated visually and with the peak signal to noise ratio (PSNR) [19] as well as structural similarity (SSIM)[20] metrics.

4.2. Models architectures

QCAE and CAE have the same topology. It is worth noticing that the number of output feature maps is four times larger in the QCAE due to the quaternion convolution, meaning 8 quaternion-valued feature maps correspond to 32 real-valued ones. Therefore, each model has two convolutional encoding layers and transposed convolutional decoding layers that deal with the same dimensions, but with different internal sizes. Indeed quaternion features maps are of size 8 and 16 to deal with an equivalent size of 32 and 64 for the CAE. Such dimensions ensure an encoding dimension slightly smaller than the original picture size. Kernel size and strides are set to 3 and 2 across all the layers respectively. Training is performed during $3,000$ epochs with the Adam optimizer [21], vanilla hyper-parameters and a learning rate of $5e^{-4}$. The hardtanh [22] activation function is used in both convolutional and transposed convolutional layers, and for both QCAE and CAE. Finally, quaternion parameters are initialized following the proposal of [23].

[2] http://r0k.us/graphics/kodak

4.3. Results and discussions

The results are reported in Figure 2. It is first important to notice that quaternion-valued CAE produced almost perfect color images *w.r.t.* to the test, while CAE completely failed to capture colors by outputting a black and white version. As motivated in Section 2, the quaternion representation alongside with the *Hamilton product* force the QCAE to consider and preserve the internal latent relations between the components of a quaternion (*i.e.* a pixel). Consequently, QCAE easily captures the color space from a gray-scale image since it learns to produce the exact same values from the input at the output, while real-valued CAE learns a gray-scale mapping by generating three identical components.

Other numerical measures are obtained based on the PSNR and SSIM of the reconstructed pictures. Due to the fact that CAE fails to learn colors, we propose to compare CAE results to the gray-scale equivalent of the test pictures. QCAE results are compared to the true color images. Consequently, we can measure how good each model is to reconstruct testing images, without being biased by the fact that CAE fails to learn colors. QCAE obtains a PSNR of

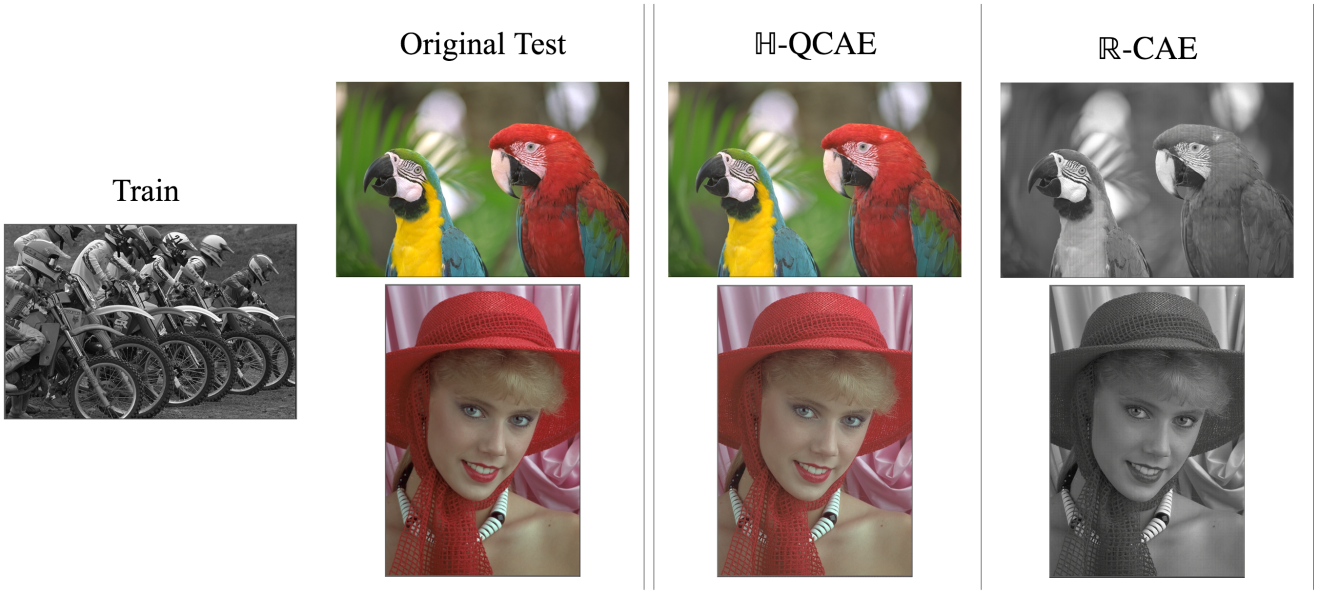

Fig. 2. Results on the gray-scale to color task with the KODAK data-set. A gray-scale training picture (*Train*) and two coloured original test images (*Original Test*) are randomly selected on the KODAK data-set and reproduced by both QCAE and CAE.

31.68dB and 28.06dB compared to 29.95dB and 27.01dB obtained with the CAE for the parrots and women images respectively. Nonetheless, while PSNR measures the amount of noise contained in an image, SSIM reports the structural and visual correlation of two pictures. SSIM of 0.96 and 0.93 are reported for QCAE compared to 0.87 and 0.86 for the CAE on the parrots and women images respectively. QCAE offered a better reconstructed image quality in both PSNR and SSIM metrics, even considering the inability of CAE to learn the color space. Moreover, the QCAE is composed of 6.4K parameters compared to 25K for the CAE. It is easily explained by the quaternion algebra. In the case of a dense layer with $1,024$ input values and $1,024$ hidden units, a real-valued model will have $1,024^2 \approx 1$M parameters, while to maintain equal input and output nodes $(1,024)$ the quaternion equivalent has 256 quaternions inputs and 256 quaternion-valued hidden units. Therefore the number of parameters for the quaternion model is $256^2 \times 4 \approx 0.26$M.

Discussions. In the one hand, the size reduction offered by QNN turns out to produce better results with an higher generalization capacity and may have other advantages such as a smallest memory footprint while saving models. Then, the natural internal relation representation induced by the *Hamilton product*, alongside with the convolution process provides an important step toward better performances of models that operate in heterogeneous contexts, or with very small data-sets. The small number of neurons allows the QCAE to obtain "robust" and "compact" memory that build a robust hidden representations of the image content in the latent space. Indeed, QCAE are not altered by heterogeneous color spaces (*e.g.* corpus of boats with a predominating blue spectrum), and are able to learn internal relations with very few examples trough the *Hamilton product*.

5. CONCLUSION

This paper proposes to clarify the recent better performances observed on image recognition with quaternion-valued neural networks, through a investigation of the impact of the *Hamilton product*. The conduced experiments demonstrate that quaternion convolutional encoder-decoders are able to perfectly learn the color-space with a training performed on a unique gray-scale image, while real-valued CAE fail, proving that the *Hamilton product* allows QNN to encode local dependencies such as the RGB relation of a pixel. Moreover, QCAE produce better quality reconstructions with respect to the PSNR and SSIM metrics than CAE, even with considering the unability of CAE to learn colors. Moreover, the quaternion representation offers more compact and expressive models. Thereby, the experiments have validated the initial intuition that the *Hamilton product*, alongside with the convolution process, allows QCAE to better separate both local and global dependencies of color images. These results are an important step forward for a more robust image recognition system in heterogeneous conditions. Future work will attempt to introduce the efficient quaternion representation to image compression and image recognition.

6. REFERENCES

[1] Yoon Kim, "Convolutional neural networks for sentence classification," *arXiv preprint 1408.5882*, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Stephen John Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex, numbers," *Electronics letters*, vol. 32, no. 21, pp. 1979–1980, 1996.

[4] Soo-Chang Pei and Ching-Min Cheng, "Color image processing by using binary quaternion-moment-preserving thresholding technique," *IEEE Transactions on Image Processing*, vol. 8, no. 5, pp. 614–628, 1999.

[5] Nicholas A Aspragathos and John K Dimitros, "A comparative study of three methods for robot kinematics," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 2, pp. 135–145, 1998.

[6] Paolo Arena, Luigi Fortuna, Luigi Occhipinti, and Maria Gabriella Xibilia, "Neural networks for quaternion-valued function approximation," in *Circuits and Systems, ISCAS'94., IEEE International Symposium on*. IEEE, 1994, vol. 6, pp. 307–310.

[7] Paolo Arena, Luigi Fortuna, Giovanni Muscato, and Maria Gabriella Xibilia, "Multilayer perceptrons to approximate quaternion valued functions," *Neural Networks*, vol. 10, no. 2, pp. 335–342, 1997.

[8] Teijiro Isokawa, Tomoaki Kusakabe, Nobuyuki Matsui, and Ferdinand Peper, "Quaternion neural network and its application," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2003, pp. 318–324.

[9] Titouan Parcollet, Ying Zhang, Mohamed Morchid, Chiheb Trabelsi, Georges Linarès, Renato de Mori, and Yoshua Bengio, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 22–26.

[10] Chase J Gaudet and Anthony S Maida, "Deep quaternion networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[11] Xuanyu Zhu, Yi Xu, Hongteng Xu, and Changjian Chen, "Quaternion convolutional neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–647.

[12] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[13] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

[14] Vincent Dumoulin and Francesco Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint 1603.07285*, 2016.

[15] D Xu, L Zhang, and H Zhang, "Learning alogrithms in quaternion neural networks using ghr calculus," *Neural Network World*, vol. 27, no. 3, pp. 271, 2017.

[16] Tohru Nitta, "A quaternary version of the back-propagation algorithm," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*. IEEE, 1995, vol. 5, pp. 2753–2756.

[17] Titouan Parcollet, Mohamed Morchid, Pierre-Michel Bousquet, Richard Dufour, Georges Linarès, and Renato De Mori, "Quaternion neural networks for spoken language understanding," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 362–368.

[18] Eric Hamilton, "Jpeg file interchange format," 2004.

[19] Deepak S Turaga, Yingwei Chen, and Jorge Caviedes, "No reference psnr estimation for compressed pictures," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 173–184, 2004.

[20] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[21] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] Ronan Collobert, "Large scale machine learning," 2004.

[23] Titouan Parcollet, Mirco Ravanelli, Mohamed Morchid, Georges Linarès, Chiheb Trabelsi, Renato De Mori, and Yoshua Bengio, "Quaternion recurrent neural networks", *arXiv preprint 1806.04418v2*, 2018.