QUATERNION CONVOLUTIONAL NEURAL NETWORKS FOR THEME IDENTIFICATION OF TELEPHONE CONVERSATIONS

Titouan Parcollet[1,2]*, Mohamed Morchid*[1]*, Georges Linarès*[1]*, and Renato De Mori*[1,3]

[1]LIA, Université d'Avignon (France)
[2]Orkis (France)
[3]McGill University, Montréal, (Canada)

ABSTRACT

Quaternion convolutional neural networks (QCNN) are powerful architectures to learn and model external dependencies that exist between neighbor features of an input vector, and internal latent dependencies within the feature. This paper proposes to evaluate the effectiveness of the QCNN on a realistic theme identification task of spoken telephone conversations between agents and customers from the call center of the Paris transportation system (RATP). We show that QCNNs are more suitable than real-valued CNN to process multidimensional data and to code internal dependencies. Indeed, real-valued CNNs deal with both internal and external relations at the same level since components of an entity are processed independently. Experimental evidence is provided that the proposed QCNN architecture always outperforms real-valued equivalent CNN models in the theme identification task of the DECODA corpus. It is also shown that QCNN accuracy results are the best achieved so far on this task, while reducing by a factor of 4 the number of model parameters.

Index Terms— Quaternions, Convolutional Neural Networks, Spoken Language Understanding

1. INTRODUCTION

Spoken language understanding (SLU) is an essential component of human-machine interaction. An SLU task, particularly important in customer care services, is the identification of themes discussed in spoken conversations. The quality of the classification results heavily depends on the selection of features, and the classifier architectures used for the task. As reviewed in [1], statistics of selected words characterizing mentions of semantic contents have been considered as sufficient features for statistical classifiers. Hidden topic features obtained with Latent Dirichelet Allocation (LDA) have also been proposed. These LDA features have been compared using statistical and deep neural network classifiers [2]. State-of-the-art methods are based on different neural networks (NN), such as deep and denses (DNN)[3], recurrents (RNN) [4, 5, 6, 7], or convolutionals (CNN) [8]. However, such models rely on unidimensional representations of the input information based on real numbers. Many realistic tasks require an adapted representation to fit the multidimensionality of the input features, such as pixels of an image, acoustic features, 3D models, or the different speech turns in a conversation. Therefore, traditional NNs process each component independently while a more natural way is to process each group of components as a single entity to learn both internal and contextual dependencies. Indeed, it is known that human-human conversations about specific items contain contextual relations between mentions of different speakers. In order to capture a part of these relations, it has been proposed to model a conversation with hyper-complex numbers [9, 10] that integrate specific features for each speaker.

Quaternions are hypercomplex numbers that contain a real and three separate imaginary components, fitting perfectly to 3 and 4 dimensional input feature vectors, such as for image processing and robot kinematics [11, 12, 13]. The idea of bundling groups of numbers into separate entities is also exploited by the recent capsule network [14]. Conversaly to traditional homogeneous representations, capsule and quaternion networks bundle sets of features together. Thereby, quaternion neural network based models are able to code latent inter-dependencies between groups of input features during the learning process with less parameters than traditional NNs, by taking advantage of the *Hamilton product* as the equivalent of the ordinary product, but between quaternions. Quaternion neural networks [15, 16, 17] have been proposed to solve different tasks that involve composed entities as input features [16, 17]. In particular, a deep quaternion network (QDNN)[18, 19], a quaternion convolutional network (QCNN)[20, 21], and a quaternion recurrent neural network (QRNN)[22] have been successfully employed for challenging tasks such as images and language processing.

More precisely, good results have been obtained in the past for theme identification of telephone conversations [9] using a quaternion-based multilayer perceptron (QMLP) with adapted features for each speaker. However, the QMLP used

as a solution to this task does not take into consideration the external and contextual informations that can exist between different turns of a dialogue. Consequently, the novelties introduced in this paper are:

- Merge the user-agent conversation segmentation of [9] with a quaternion convolutional neural network [1] to learn efficiently both internal and external dependencies (Section3).

- Evaluate the proposed method on a realistic task of theme identification of telephone conversations on the DECODA framework (Section 4).

The conducted experiments show that the proposed QCNN always outperforms equivalent real-valued CNN with a drastic reduction of the number of free parameters (up to 4 times less). Moreover, it obtains the best observed result so far with an accuracy of 87% compared to 85.2% with the previous method.

2. QUATERNION ALGEBRA

The quaternion algebra H defines operations between quaternion numbers. A quaternion Q is an extension of a complex number to the hyper-complex plane 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. A quaternion Q can also be summarized into the following matrix of real numbers, that turns out to be more suitable for computations:

$$Q_{mat} = \begin{bmatrix} r & -x & -y & -z \\ x & r & -z & y \\ y & z & r & -x \\ z & -y & x & r \end{bmatrix}. \tag{2}$$

The conjugate Q^* of Q is defined as:

$$Q^* = r1 - x\mathbf{i} - y\mathbf{j} - z\mathbf{k}. \tag{3}$$

Then, a normalized or unit quaternion $Q^\triangleleft$ is expressed as:

$$Q^\triangleleft = \frac{Q}{\sqrt{r^2 + x^2 + y^2 + z^2}}. \tag{4}$$

[1]The code is available at https://github.com/TParcollet/Quaternion-Convolutional-Neural-Networks-for-End-to-End-Automatic-Speech-Recognition

Finally, the Hamilton product $\otimes$ between two quaternions Q_1 and Q_2 is computed as follows:

$$\begin{aligned} Q_1 \otimes Q_2 =& (r_1 r_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\ & (r_1 x_2 + x_1 r_2 + y_1 z_2 - z_1 y_2)\boldsymbol{i} + \\ & (r_1 y_2 - x_1 z_2 + y_1 r_2 + z_1 x_2)\boldsymbol{j} + \\ & (r_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 r_2)\boldsymbol{k}. \end{aligned} \tag{5}$$

The Hamilton product is used in QCNNs to perform transformations of vectors representing quaternions, as well as scaling and interpolation between two rotations following a geodesic over a sphere in the R^3 space as shown in [23].

3. QUATERNION CONVOLUTIONAL NEURAL NETWORKS

This section defines the quaternion neural network convolution (Section 3.1) and an appropriate parameter initialization (Section 3.2).

3.1. Quaternion convolution

The QCNN is an extension of the well-known real-valued deep convolutional networks (CNN) [24] to quaternion numbers. Following recent propositions for convolution of complex [25], and quaternion numbers [21, 20], the quaternion convolution operation is performed with the real-number matrices representation of quaternions. Therefore, a traditional $1D$ convolutional layer, with a kernel that contains FM 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 [26] and [27]. As a result, the so-called "split" approach [16, 9] is used as a quaternion equivalence of real-valued activation functions:

$$\alpha(Q) = \alpha(r) + \alpha(x)\mathbf{i} + \alpha(y)\mathbf{j} + \alpha(z)\mathbf{k}, \tag{6}$$

with α corresponding to any standard activation function. Finally, the convolution of a quaternion filter matrix with a quaternion vector is performed. For this computation, the Hamilton product is computed using the real-valued matrices representation of quaternions. Let $W = R + X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}$ be a quaternion weight filter matrix, and $X_p = r + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ the quaternion input vector. The quaternion convolution w.r.t the Hamilton product $W \otimes X_p$ is defined as follows:

$$\begin{aligned} W \otimes X_p =& (Rr - Xx - Yy - Zz) + \\ & (Rx + Xr + Yz - Zy)\mathbf{i} + \\ & (Ry - Xz + Yr + Zx)\mathbf{j} + \\ & (Rz + Xy - Yx + Zr)\mathbf{k}, \end{aligned} \tag{7}$$

and can thus be expressed in a matrix form following eq. 2:

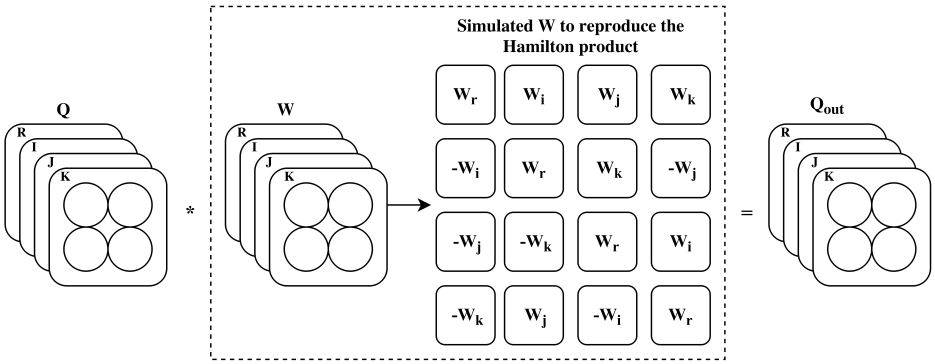

Fig. 1. Illustration of the quaternion convolution process

$$W \otimes X_p = \begin{bmatrix} R & -X & -Y & -Z \\ X & R & -Z & Y \\ Y & Z & R & -X \\ Z & -Y & X & R \end{bmatrix} * \begin{bmatrix} r \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r' \\ x'\mathbf{i} \\ y'\mathbf{j} \\ z'\mathbf{k} \end{bmatrix}. \quad (8)$$

An illustration of the quaternion convolution operation is depicted in Figure 1.

3.2. Quaternion parameters initialization

A suitable initialization scheme improves neural networks convergence and reduces the risk of exploding and vanishing gradient. However, quaternion numbers cannot be initialized component-wise as for traditional initialization criterions. The reason for this relies in the specific quaternion algebra and the interaction between components. In [21], this issue is addressed by introducing a well-designed algorithm for initializing quaternion parameters. Consequently, a weight component w of the weight matrice W can be sampled as follows:

$$\begin{aligned} w_{\mathbf{r}} &= \varphi\, cos(\theta), \\ w_{\mathbf{i}} &= \varphi\, q^{\triangleleft}_{imag\mathbf{i}}\, sin(\theta), \\ w_{\mathbf{j}} &= \varphi\, q^{\triangleleft}_{imag\mathbf{j}}\, sin(\theta), \\ w_{\mathbf{k}} &= \varphi\, q^{\triangleleft}_{imag\mathbf{k}}\, sin(\theta). \end{aligned} \quad (9)$$

The angle θ is randomly generated in the interval $[-\pi, \pi]$. Then, the quaternion $q^{\triangleleft}_{imag}$ is defined as purely normalized imaginary, and is expressed as $q^{\triangleleft}_{imag} = 0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. The imaginary components x**i**, y**j**, and z**k** are sampled from an uniform distribution in $[0, 1]$ to obtain q_{imag}, which is then normalized (following eq. 4) to obtain $q^{\triangleleft}_{imag}$. The parameter φ is a random number generated with respect to well-known initialization criterions, such as Glorot [28] or He [29], but extended to the quaternion space as:

$$\sigma = \frac{1}{\sqrt{2(n_{in} + n_{out})}}, \quad (10)$$

and,

$$\sigma = \frac{1}{\sqrt{2n_{in}}}. \quad (11)$$

for the Glorot and He criterions respectively, with n_{in} and n_{out} the number of neurons of the input and output layers. Finally, φ can be sampled from $[-\sigma, \sigma]$ to complete the weight initialization of Eq. 9.

4. EXPERIMENTS

4.1. Theme identification in spoken conversations

The experiments considered on this paper concern the automatic analysis of telephone conversations between one or more operators and a customer, in the call center of the Paris public transportation system (RATP). In order to plan improvements of customer satisfaction, a domain application ontology has been defined. For this purpose, 8 themes, described in Section 4.2, have been identified to classify customer concerns. More precisely, a conversation involves a customer calling from an unconstrained environment (typically from train station or street, by using a mobile phone) and one or more agents that are supposed to follow a conversation protocol to address customers requests or complains. Such conversations tend to follow the model described by the agents protocol, an example of which is shown in Figure 2. Based on this protocol, the paper considers agents and customer speech turns seperatly to better capture the structure and the theme of the conversation. However, the identification of conversation themes is perturbed mostly by acoustic environment noise, that affects the quality of the transcription provided by an automatic speech recognition (ASR) system. Most of them come from user turns, while useful information for the classification task is correctly encoded in agent turns. Moreover, themes are ambiguous due to the applicative context. Indeed, most of the conversations focus on traffic details or issues, station names or schedules, etc ... Finally, many dialogues contain secondary topics such as detailed traffic and

itinerary perturbations, increasing the risk of a wrong prediction of the dominant theme.

4.2. DECODA dataset

The DECODA corpus [30] contains real-life human-human telephone conversations collected in the customer care service of the Paris transportation system (RATP). It is composed of $1,242$ telephone conversations, corresponding to about 74 hours of signal, split into a train (train - 739 dialogues), a development (dev - 175 dialogues) and a test set (test - 327 dialogues). Each conversation is annotated with one of 8 themes. Themes correspond to customer problems or inquiries about itinerary, lost and found, time schedules, transportation cards, state of the traffic, fares, fines and special offers. The LIA-Speeral ASR system [31] is used to obtains the automatic transcription of each conversation. In this context, acoustic model parameters are estimated from 150 hours of telephone speech. The vocabulary contains $5,782$ words. A 3-gram language model (LM) is obtained by adapting a basic LM with the training set transcriptions. Finally, word error rates (WERs) of 33.8%, 45.2% and $49.\%$ are reported on the train, development and test sets respectively. These high WERs are mainly due to speech disfluencies in casual users and to adverse acoustic environments in metro stations and streets.

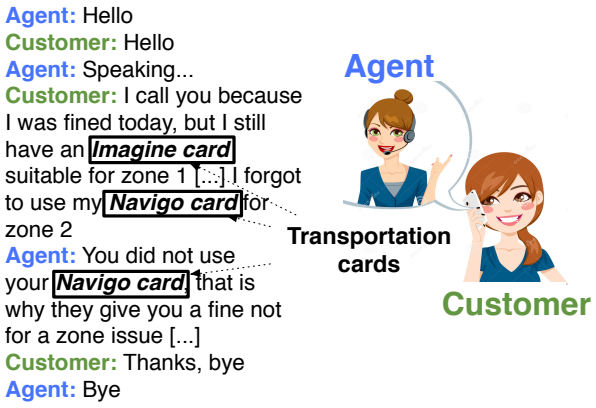

Fig. 2. Example of a *manualy transcripted* dialogue from the DECODA corpus for the SLU task of theme identification.

4.3. Quaternions of conversation features

A specific user-agent segmentation has been proposed in [9] based on a LDA [32] space of 25 topics, to take into account the structure of the dialogues, and to build a quaternion $Q = r1 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ with the user part of the dialogue in the first complex value x, the agent in y and the whole conversation on z. Moreover, 10 different folds of these features are generated and concatenated into one vector, to alleviate any variation due to the LDA spaces. The input vector contains $25 \times 10 = 250$ quaternions or $25 \times 10 \times 4 = 1,000$ real numbers.

4.4. Models architectures

The architectures of both CNN and QCNN are inspired by deep residual and convolutional neural networks [24]. Consequently, the proposed models contain two blocks of L 1D convolutional layers composed of FM feature maps, with FM doubled between each block. However, no residual connections are added due to the small number of blocks. Finally, 3 quaternion- or real-valued dense layers of size 64 and 256 respectively are stacked together with a last dense and real-valued layer of size 8 (corresponding to the 8 themes). Indeed, the output of a dense quaternion-valued layer has $64 \times 4 = 256$ nodes and is 4 times larger than the number of units. Each convolutional layer is based on a filter size of 3, and is padded to keep the sequence and signal sizes unaltered. Best models are investigated by varyating the number of layers from 4 to 12, and the number of feature maps from 16 to 64 and 64 to 256 for the real- and quaternion-valued models respectively. Indeed, the number of output feature maps is 4 times larger in the QCNN due to the quaternion convolution, meaning that 32 quaternion-valued feature maps correspond to 128 real-valued ones. The ReLU activation function is employed for both models [33]. A dropout of 0.3 is used across all the layers, except the input and output ones. CNNs and QCNNs are trained with the Adam learning rate optimizer and vanilla hyperparameters [34] during 50 epochs with an initial learning rate of $1e^{-4}$. Experiments are performed on Tesla P100 GPUs.

4.5. Results and discussion

This section provides the results observed for CNNs and QCNNs on the theme identification of telephone conversation task of the DECODA dataset. The best architectures are first investigated for both models and are then compared to previous work on this benchmark. All the results are from a 3 folds average.

QCNN vs CNN

Table 1 reports the results observed for different topologies of QCNNs and CNNs. It is worth underlying the important difference in term of number of parameters between quaternion- and real-valued models. Indeed, a 4 convolutional layered CNN with $FM = 128$ contains 16.8 millions parameters compared to only 4.2 millions for an equivalent QCNN. This is easily explained by the quaternion algebra. Indeed, a real-valued dense layer with 256 input values and 256 hidden units has $256^2 = 65.5K$ free parameters, while to maintain equal input and output nodes (256) the quaternion equivalent has 64 quaternions inputs and 64 quaternion-valued hidden units.

Consequently, the number of parameters is $64^2 \times 4 = 16,3K$. Such a complexity reduction turns out to produce better results due to a more compact representation of the information, and have other advantages such as a smaller memory footprint. Therefore, equally sized QCNN use always 4 times less parameters than real-valued CNNs. The best accuracy of 87.0% is obtained with a QCNN of 4 convolutional layers and 256 feature maps, compared to 85.4% for a real-valued CNN with 4 layers and with $FM = 128$. Due to the small size of the DECODA dataset both QCNNs and CNNs tend to overfit when increasing the number of layers. The accuracy drops from 85.0% with 4 layers to 82.7% with 12 layers, and 85.4% to 84.9% for CNNs and QCNNs respectively. However, a higher number of feature maps leads to the overfitting phenomenon only for real-valued CNNs. Indeed, QCNNs produce better accuracies with bigger feature maps. This is also explained by the quaternion algebra reduction property, since the actual feature map size is 4 times lower but produces the same dimension output. Finally, QCNNs always perform better than equivalent CNNs with less parameters, and tend to scale better with larger architectures due to a more compact representation.

Table 1. Theme identification results from a 3 folds average. 'L' stands for the number of layers, 'FM' for the number of feature maps, and 'Params' for the number of learning parameters. 'FM' is expressed in order to be equivalent for both models. Therefore, 64FM is equal to 64FM for real numbers and 16 quaternion-valued FM

Models	Dev. %	Test %	Params
R-CNN-4L-64FM	91.7	85.0	8.3M
H-QCNN-4L-64FM	92.6	85.4	2.1M
R-CNN-8L-64FM	91.7	85.0	8.5M
H-QCNN-8L-64FM	91.9	85.1	2.1M
R-CNN-12L-64FM	89.9	82.7	8.6M
H-QCNN-12L-64FM	91.8	84.9	2.2M
R-CNN-4L-128FM	**91.7**	**85.4**	16.8M
H-QCNN-4L-128FM	93.2	86.3	4.2M
R-CNN-8L-128FM	91.7	84.2	17.2M
H-QCNN-8L-128FM	93.2	86.1	4.3M
R-CNN-12L-128FM	91.5	84.2	17.8M
H-QCNN-12L-128FM	92.2	85.3	4.4M
R-CNN-4L-256FM	91.5	85.0	34.2M
H-QCNN-4L-256FM	**93.6**	**87**	8.6M
R-CNN-8L-256FM	91.6	84.9	36.1M
H-QCNN-8L-256FM	92.1	85.8	9.1M
R-CNN-12L-256FM	91.5	84.6	38.1M
H-QCNN-12L-256FM	90.9	85.1	9.5M

QCNN vs other models on the DECODA dataset

Many experiments have been conducted on the DECODA framework to provide a reliable solution to this problematic. Therefore, Table 2 sums up all the results obtained with different neural networks architectures. It is worth mentioning that the 87.0% accuracy of the QCNN is the best result observed so far on the DECODA framework. Moreover, quaternion-based models always show better results than the corresponding real-valued competitors.

Table 2.

Models	Type	Test %
MLP[18]	R	83.4
QMLP[18]	H	84.6
DSAE[2]	R	82.0
DAE[19]	R	83.0
QDAE[19]	H	85.2
DNN[18]	R	84.0
QDNN[18]	H	85.2
CNN	R	85.4
QCNN	H	87.0

5. CONCLUSION

Summary. This paper proposes to merge the effective quaternion representation of multiple speech turns of telephone conversations, with the well-known convolutional process to achieve a better internal and external representation of the relevant information. The spoken language understanding experiments on the DECODA dataset have shown that: 1) QCNN obtains the best results observed so far on this task; 2) Quaternion based models always outperform real-valued ones; 3) Improvements are observed with an important reduction of the number of learning parameters; Therefore, the initial intuition that a well adapted quaternion representation alongside with convolutional neural networks, allow the QCNN to better model the external and internal relations into a compact and efficient representation has been validated.

Limitations and Future Work. The DECODA dataset is not large enough to fully highlight the potential of quaternion-based models. Consequently, a future work will consist to apply QCNNs to larger text-corpora. Moreover, convolutional neural networks do not adequately model relevant sequential dependencies in text and speech documents. Quaternion-based recurrent neural networks (QRNNs) will be used in future works to take into account long and short term dependencies to obtain more accurate results.

6. ACKNOWLEDGEMENTS

The experiments were conducted using Keras [35]. The authors would like to acknowledge the computing support of Compute Canada and the founding support of Orkis. Thanks to Mirco Ravanelli for his helpfull comments.

7. REFERENCES

[1] Timothy J Hazen, Fred Richardson, and Anna Margolis, "Topic identification from audio recordings using word and phone recognition lattices," in *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*. IEEE, 2007, pp. 659–664.

[2] Killian Janod, Mohamed Morchid, Richard Dufour, Georges Linares, and Renato De Mori, "Deep stacked autoencoders for spoken language understanding," *ISCA INTERSPEECH*, vol. 1, pp. 2, 2016.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[4] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, 2010, vol. 2, p. 3.

[5] Ken-ichi Funahashi and Yuichi Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural networks*, vol. 6, no. 6, pp. 801–806, 1993.

[6] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[7] Alex Graves and Jürgen Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[8] Xiang Zhang, Junbo Zhao, and Yann LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[9] 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.

[10] Mohamed Morchid, Georges Linarès, Marc El-Beze, and Renato De Mori, "Theme identification in telephone service conversations using quaternions of speech features," in *Interspeech*. ISCA, 2013.

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

[12] 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.

[13] 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.

[14] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, "Dynamic routing between capsules," *arXiv preprint arXiv:1710.09829v2*, 2017.

[15] 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.

[16] 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.

[17] 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.

[18] Titouan Parcollet, Mohamed Morchid, and Georges Linares, "Deep quaternion neural networks for spoken language understanding," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 504–511.

[19] Parcollet Titouan, Mohamed Morchid, and Georges Linares, "Quaternion denoising encoder-decoder for theme identification of telephone conversations," *Proc. Interspeech 2017*, pp. 3325–3328, 2017.

[20] Anthony Maida Chase Gaudet, "Deep quaternion networks," *arXiv preprint arXiv:1712.04604v2*, 2017.

[21] Parcollet Titouan, Zhang Ying, Morchid Mohamed, Trabelsi Chiheb, Linarès Georges, De Mori Renato, and Bengio Yoshua, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," *arXiv preprint arXiv:1806.07789*, 2018.

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

[23] Toshifumi Minemoto, Teijiro Isokawa, Haruhiko Nishimura, and Nobuyuki Matsui, "Feed forward neural network with random quaternionic neurons," *Signal Processing*, vol. 136, pp. 59–68, 2017.

[24] 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.

[25] Chiheb Trabelsi, Olexa Bilaniuk, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal, "Deep complex networks," *arXiv preprint arXiv:1705.09792*, 2017.

[26] 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.

[27] 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.

[28] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[30] Frederic Bechet, Benjamin Maza, Nicolas Bigouroux, Thierry Bazillon, Marc El-Beze, Renato De Mori, and Eric Arbillot, "Decoda: a call-centre human-human spoken conversation corpus.," in *LREC*, 2012, pp. 1343–1347.

[31] Georges Linares, Pascal Nocéra, Dominique Massonie, and Driss Matrouf, "The lia speech recognition system: from 10xrt to 1xrt," in *Text, Speech and Dialogue*. Springer, 2007, pp. 302–308.

[32] David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[33] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

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

[35] François Chollet et al., "Keras," https://github.com/keras-team/keras, 2015.