
Réseaux de neurones convolutifs de quaternions pour l'identification de thèmes de conversations téléphoniques

Titouan Parcollet^{1,2}, Mohamed Morchid¹, Georges Linarès¹

1. Avignon Université, LIA, France

prenom.nom@univ-avignon.fr

2. Orkis, Aix-en-Provence, France

RÉSUMÉ. Les réseaux de neurones convolutifs de quaternions (QCNN) forment un ensemble d'algorithmes particulièrement efficaces pour l'apprentissage et la modélisation des relations internes et externes existantes entre les caractéristiques d'un vecteur de données. Ce travail propose d'évaluer l'efficacité des QCNNs à travers une tâche réaliste d'identification automatique de thèmes contenus dans des conversations téléphoniques, impliquant des agents et des usagers du centre d'appel du service de transport de la ville de Paris (RATP). Il est ainsi démontré que les QCNNs sont plus performants que leurs homologues basés sur l'algèbre des nombres réels (CNN) pour le traitement de données multidimensionnelles, et pour coder les relations internes. En effet, les CNNs traitent, sans distinction, et au même niveau, les relations internes et externes, dû à la nature propre des représentations dans des espaces composés de nombres réels. Les expérimentations montrent que le QCNN surpasse de manière constante et pour l'ensemble des configurations le CNN durant la tâche d'identification de thèmes basée sur le corpus de dialogues parlés DECODA. De plus, cette hausse de précision est accompagnée d'une réduction significative du nombre de paramètres neuronaux composant le modèle par un facteur de quatre.

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 four the number of model parameters.

MOTS-CLÉS : Quaternions, Réseaux de Neurones Convolutifs, Traitement du Langage

KEYWORDS: Quaternions, Convolutional Neural Networks, Spoken Language Understanding

DOI:10.3166/RIA..1-13 © 2019 Lavoisier

1. Introduction

Le traitement automatique du langage (TAL) est un processus essentiel pour la compréhension des interactions entre l'homme et la machine. Par exemple, une tâche de traitement automatique du langage particulièrement importante dans les services d'assistance à la clientèle, consiste à identifier les thèmes abordés lors de conversations. La qualité des résultats de la classification dépend fortement de la sélection des données et des architectures des classificateurs utilisés. Comme indiqué par (Hazen *et al.*, 2007), les statistiques de certains mots caractérisant des mentions de contenu sémantique ont été considérées comme suffisantes pour ces classificateurs. Des caractéristiques de *topics* cachées obtenues par une Allocation Latente de Dirichlet (LDA) ont également été proposées. Ces caractéristiques LDA ont d'ailleurs été comparées à l'aide de classificateurs statistiques et de classificateurs neuronaux profonds (Janod *et al.*, 2016). Les méthodes les plus récentes reposent sur différents réseaux de neurones (NN), tels que les réseaux profonds et denses (DNN) (Graves *et al.*, 2013), récurrents (RNN) (Mikolov *et al.*, 2010; Funahashi, Nakamura, 1993; Gers *et al.*, 1999; Graves, Schmidhuber, 2005) ou convolutifs (CNN) (Zhang *et al.*, 2015). Cependant, ces modèles utilisent une représentation unidimensionnelle de l'information d'entrée, basée sur l'algèbre des nombres réels. De nombreuses tâches du quotidien nécessitent toutefois une représentation adaptée à la multidimensionnalité des entités en entrée, tels que les pixels composant une image, les caractéristiques acoustiques d'un signal de parole, les modèles 3D, ou encore les différents tours de parole qui rythment une conversation. Aussi, les réseaux neuronaux traditionnels traitent chaque composant de manière indépendante, alors qu'il serait bien plus naturel de traiter chaque groupe de composants à l'aide d'une seule entité, afin d'apprendre les dépendances internes et contextuelles entre ces composants. En effet, il est connu que les conversations entre être humains sur des sujets spécifiques contiennent des relations contextuelles entre les mentions des différents protagonistes. Afin de capturer une partie de ces relations, il a été proposé de modéliser une conversation en s'appuyant sur les nombres hyper-complexes (Parcollet *et al.*, 2016; Morchid *et al.*, 2013) qui intègrent ces informations spécifiques à chaque locuteur.

Les quaternions sont des nombres hyper-complexes qui contiennent une partie réelle et trois composants imaginaires distincts, s'adaptant parfaitement aux vecteurs d'entités de dimension comprise entre trois et quatre, tels que pour le traitement d'image et la robotique inversée (Sangwine, 1996; Pei, Cheng, 1999; Aspragathos, Dimitros, 1998). L'idée de regrouper des ensembles de nombres en entités distinctes

est également exploitée par les récents réseaux de neurones à capsules (Sabour *et al.*, 2017). À l'inverse des représentations homogènes traditionnelles, les réseaux de capsules et de quaternions regroupent des ensembles de caractéristiques. Ainsi, les modèles basés sur les réseaux de neurones de quaternions sont capables de coder des inter-dépendances latentes entre des groupes de caractéristiques d'entrée pendant le processus d'apprentissage, avec moins de paramètres que les NNs traditionnels tout en tirant parti du produit de Hamilton comme équivalent au produit scalaire. Les réseaux de neurones de quaternions (Isokawa *et al.*, 2003 ; Arena *et al.*, 1994 ; 1997) ont été initialement proposés pour résoudre différentes tâches liées à l'approximation de fonctions hyper-complexes (Arena *et al.*, 1994 ; 1997). Ensuite, un réseau de quaternions profond (QDNN) (Parcollet *et al.*, 2017a ; 2017b), un réseau de quaternions convolutif (QCNN) (Chase Gaudet, 2017 ; Parcollet *et al.*, 2018) et un réseau de neurones de quaternions récurrent (QRNN) ont été employés avec succès pour des tâches difficiles telles que le traitement des images et du langage.

Plus précisément, des résultats prometteurs ont été précédemment obtenus pour l'identification thématique de conversations téléphoniques (Parcollet *et al.*, 2016) en employant un perceptron multicouche basé sur l'algèbre des quaternions (QMLP), et en considérant des caractéristiques adaptées à chaque locuteur. Cependant, le QMLP ainsi utilisé ne prend pas en compte les informations externes et contextuelles pouvant exister entre différents tours de parole composant le dialogue. En effet, a contrario des réseaux récurrents et convolutifs, l'architecture simple du QMLP ne lui permet pas de bénéficier du contexte global. Par conséquent, les nouveautés introduites dans cet article sont les suivantes:

- Fusionner la segmentation de conversation usager-agent (Parcollet *et al.*, 2016) avec un réseau de neurones de quaternions convolutif¹ pour apprendre efficacement les dépendances internes et externes (Section 3).
- Evaluer la méthode proposée sur une tâche réaliste d'identification de thème de conversations téléphoniques basé sur le corpus DECODA (Section 4).

Les expériences menées montrent que le QCNN proposé surpasse toujours le CNN avec une réduction drastique du nombre de paramètres neuronaux (jusqu'à quatre fois moins). De plus, le QCNN obtient le meilleur résultat observé jusqu'à présent sur la tâche DECODA avec une précision de 87% par rapport aux 85;2% obtenus avec les méthodes précédentes.

1. Le code source est disponible à l'adresse: <https://github.com/Orkis-Research/Pytorch-Quaternion-Neural-Networks>

2. Algèbre des Quaternions

L'algèbre des quaternions \mathbb{H} définit les opérations entre les nombres hyper-complexes. Ainsi, un quaternion Q est une extension d'un nombre complexe au plan hyper-complexe défini dans un espace à quatre dimensions tel que:

$$Q = r1 + xi + yj + zk: \quad (1)$$

où r, x, y et z sont des nombres réels et $1, \mathbf{i}, \mathbf{j}$ et \mathbf{k} sont la base de l'unité de Q . Dans un quaternion, r est la partie réelle, alors que $xi + yj + zk$ avec $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$ est la partie imaginaire ou la partie vectorielle. Une telle définition est utilisée pour décrire les rotations dans un espace à trois dimensions. Un quaternion Q peut également être résumé pas la matrice réelle suivante, qui s'avère plus appropriée pour optimiser les opérations nécessaires lors des différents calculs :

$$Q_{mat} = \begin{bmatrix} r & x & y & z \\ x & r & z & y \\ y & z & r & x \\ z & y & x & r \end{bmatrix} : \quad (2)$$

Le conjugué Q de Q est défini comme :

$$Q = r1 - xi - yj - zk: \quad (3)$$

Un quaternion normalisé ou unitaire Q' est exprimé par :

$$Q' = \frac{Q}{\sqrt{r^2 + x^2 + y^2 + z^2}}: \quad (4)$$

Enfin, le produit de Hamilton entre deux quaternions Q_1 et Q_2 est défini comme suit :

$$\begin{aligned} Q_1 Q_2 = & (r_1 r_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\ & (r_1 x_2 + x_1 r_2 + y_1 z_2 - z_1 y_2) \mathbf{i} + \\ & (r_1 y_2 - x_1 z_2 + y_1 r_2 + z_1 x_2) \mathbf{j} + \\ & (r_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 r_2) \mathbf{k}: \end{aligned} \quad (5)$$

Le produit de Hamilton est utilisé dans les QCNNs pour effectuer des transformations de vecteurs représentant des quaternions, ainsi que des opérations d'interpolation entre deux rotations suivant une géodésique sur une sphère de l'espace \mathbb{R}^3 (Minemoto *et al.*, 2017).

3. Réseaux de neurones de quaternions convolutifs

Cette section définit la convolution au sein des réseaux de neurones de quaternions (Section 3.1, et une initialisation des paramètres neuronaux adaptée (Section 3.2).

3.1. Convolution de quaternions

Le QCNN est une extension des réseaux convolutifs profonds (CNN) à valeurs réelles (He *et al.*, 2016) aux quaternion. Suivant les récentes propositions de convolution de nombres complexes (Trabelsi *et al.*, 2017), et des quaternions (Parcollet *et al.*, 2018; Chase Gaudet, 2017), l'opération de convolution au sein d'un réseau de neurones de quaternions est effectuée à l'aide des matrices de nombres réels représentant les quaternions. Par conséquent, une couche convolutionnelle 1D traditionnelle, avec un noyau contenant FM matrices de filtres, est divisée en 4 parties : la première partie est égale à r , la seconde à $x\mathbf{i}$, la troisième à $y\mathbf{j}$ et la dernière à $z\mathbf{k}$ d'un quaternion $Q = r\mathbf{1} + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. La propagation arrière est assurée par des fonctions de coût et d'activation différentiables, et qui ont déjà été étudiées pour les nombres hypercomplexes (Xu *et al.*, 2017; Nitta, 1995). Par conséquent, l'approche dite "scindée" (Arena *et al.*, 1994; Parcollet *et al.*, 2016) est utilisée comme équivalence quaternionique des fonctions d'activation à valeurs réelles:

$$(Q) = (r) + (x)\mathbf{i} + (y)\mathbf{j} + (z)\mathbf{k}; \quad (6)$$

avec σ correspondant à n'importe quelle fonction d'activation standard. Enfin, la convolution d'une matrice de filtres de quaternions avec un vecteur de quaternions est effectuée en se basant sur la représentation matricielle des quaternions et du produit de Hamilton. Soit $W = R + X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}$ une matrice de filtres de quaternions et $X_p = r + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ le vecteur d'entrée de quaternions. La convolution en rapport avec le produit de Hamilton $W \otimes X_p$ est définie comme :

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

et peut donc être exprimée sous forme matricielle (éq. 2) :

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

La Figure 1 illustre ce produit de convolution.

3.2. Initialisation des paramètres neuronaux

Un schéma d'initialisation approprié améliore la convergence des réseaux de neurones et réduit le risque d'explosion et/ou de forte réduction du gradient. Toutefois, les

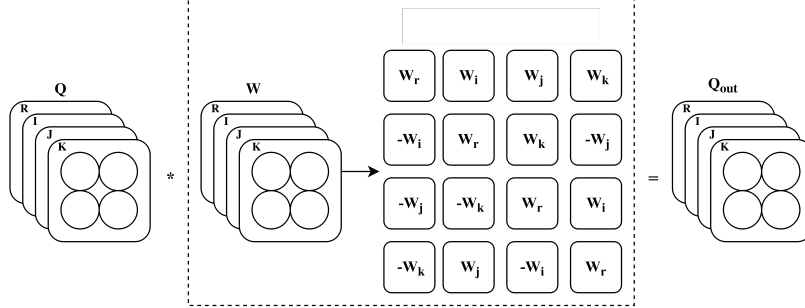


FIGURE 1. Illustration du produit de convolution dans un réseau de neurones de quaternions convolutif

éléments d'un paramètre quaternionique ne peuvent pas être initialisés indépendamment les uns des autres, contrairement aux critères d'initialisation classiques. En effet, l'algèbre des quaternions définit une interaction entre chaque composant. En nous basant sur (Parcollet *et al.*, 2018), ce problème peut être résolu par l'introduction d'un algorithme spécialement conçu pour l'initialisation des paramètres quaternioniques. Par conséquent, un poids w de la matrice W peut être échantillonné comme suit :

$$\begin{aligned}
 w_r &= ' \cos(\); \\
 w_i &= ' q'_{imagi} \sin(\); \\
 w_j &= ' q'_{imagj} \sin(\); \\
 w_k &= ' q'_{imagk} \sin(\);
 \end{aligned} \tag{9}$$

L'angle est généré aléatoirement dans l'intervalle $[\ ; \]$. Le quaternion purement unitaire et imaginaire q'_{imag} est exprimé comme suit : $q'_{imag} = 0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. Les composantes imaginaires $x\mathbf{i}$, $y\mathbf{j}$, et $z\mathbf{k}$ sont tirées d'une distribution uniforme entre $[0; 1]$ afin d'obtenir q_{imag} , qui est ensuite normalisé (éq. 4) pour trouver q'_{imag} . Le paramètre ' est un nombre aléatoire généré suivant des critères d'initialisation connus comme Glorot (Glorot, Bengio, 2010), ou He (He *et al.*, 2015), mais étendu au corps des quaternions :

$$= \frac{1}{\sqrt{2(n_{in} + n_{out})}}; \tag{10}$$

et,

$$= \rho \frac{1}{2n_{in}}; \tag{11}$$

pour les critères de Glorot et He respectivement, avec n_{in} et n_{out} les nombres de neurones des couches d'entrée et de sortie. Enfin, ϵ est généré entre $[\epsilon_{min}; \epsilon_{max}]$ pour terminer l'initialisation du poids w de l'éq. 9.

4. Experiences

4.1. Identification de thèmes dans des conversations parlées

Les expériences considérées dans ce travail concernent l'analyse automatique des conversations téléphoniques entre un ou plusieurs opérateurs et un client, dans le centre d'appel de la société du système de transport en commun parisien (RATP). Afin de planifier des améliorations de la satisfaction client, une ontologie a été définie. À cette fin, huit thèmes ont été identifiés (Section 4.2) pour permettre la classification des préoccupations des usagers. Plus précisément, une conversation implique un client appelant depuis un environnement non contraint (généralement depuis une gare ou une rue, en utilisant un téléphone mobile), et un ou plusieurs agents censés suivre un protocole de conversation pour répondre aux demandes ou aux plaintes des clients. De telles conversations tendent à suivre le modèle décrit par le protocole des agents, dont un exemple est présenté dans la Figure 2. Sur la base de ce protocole, il est important de traiter séparément le contenu parlé de l'agent et de l'utilisateur afin de mieux saisir la structure et le thème de la conversation. Toutefois, l'identification des thèmes est perturbée par le bruit de l'environnement acoustique, qui affecte fortement la qualité de la transcription fournie par le système de reconnaissance automatique de la parole (SRAP). De plus, les thèmes sont ambigus en raison du contexte applicatif. En effet, la plupart des conversations portent sur des détails ou des problèmes de trafic, des noms de station ou des horaires, etc. Enfin, de nombreux dialogues contiennent des sujets secondaires tels que des perturbations détaillées du trafic et des itinéraires, augmentant le risque de prédiction erronée du thème dominant.

4.2. Le corpus DECODA

Le corpus DECODA (Bechet *et al.*, 2012) regroupe des conversations téléphoniques provenant du service clientèle du système de transport parisien (RATP). Il est composé de 1;242 conversations, correspondant à environ 74 heures de signal, réparties en un corpus d'apprentissage de 739 dialogues, un développement de 175 dialogues et un ensemble de test de 327 documents. Chaque conversation est annotée avec l'un des huit thèmes. Les thèmes correspondent aux problèmes des clients ou aux demandes de renseignements concernant l'itinéraire, les objets perdus et trouvés, les horaires, les cartes de transport, l'état du trafic, les tarifs, les amendes et les offres spéciales. Le système SRAP LIA-Speeral (Linares *et al.*, 2007) est utilisé pour obtenir la transcription automatique de chaque conversation. Dans ce contexte, les paramètres du modèle acoustique sont estimés à partir de 150 heure de conversation téléphonique. Le vocabulaire contient 5;782 mots. Un modèle de langage (ML) 3-gram est obtenu en adaptant un ML de base avec les transcriptions des ensembles d'apprentissage. En-

fin, des taux d'erreur mots (WERs) de 33.8% , 45.2% et 49% sont obtenus pour les ensembles d'apprentissage, développement et test respectivement. Ces taux élevés de WER sont principalement dus à des troubles de la parole chez les utilisateurs et aux environnements acoustiques défavorables des stations de bus et métro.



FIGURE 2. Exemple d'une conversation transcrite manuellement provenant du corpus DECODA.

4.3. Les quaternions comme caractéristiques du dialogue

Une segmentation adaptée aux quaternions, et nommée usager-agent a été proposée par (Parcollet *et al.*, 2016) en fonction d'un espace LDA (Blei *et al.*, 2003) de 25 classes, afin de prendre en compte la structure des dialogues et de construire un quaternion $Q = r1 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ avec la partie usager du dialogue dans la première valeur complexe x , l'agent dans y et la conversation entière dans z . Ainsi, chaque partie est projetée dans cet espace LDA de dimension 25 pour obtenir les composants des quaternions. Il est attendu que la capacité des réseaux de neurones de quaternions à apprendre les relations internes, aide le modèle à découvrir des relations existantes à l'intérieur du dialogue entre un usager et un agent. La partie réelle r est initialisée à zéro, comme il a été proposé pour le traitement de l'image par (Isokawa *et al.*, 2003). Cette partie réelle changera pour une autre valeur après l'inférence de la première couche cachée. De plus, 10 tirages LDA différents sont concaténés en un seul vecteur, afin d'atténuer les variations dues à ces espaces latents comme proposé par (Parcollet *et al.*, 2017a). Le vecteur d'entrée contient donc $25 \times 10 = 250$ quaternions ou $25 \times 10 \times 4 = 1,000$ nombres réels.

4.4. Architectures des modèles

Les architectures des CNNs et des QCNNs s’inspirent de réseaux neuronaux convolutifs profonds (He *et al.*, 2016). Par conséquent, les modèles proposés contiennent deux blocs de couches convolutives unidimensionnelles L composées de FM matrices de filtres, avec FM éléments entre chaque bloc. Enfin, des couches denses de tailles respectives de 64 et de 256, pour les quaternions et les réels, sont agglomérées à une dernière couche dense de taille 8 (correspondant aux 8 thèmes). En effet, la sortie d’une couche dense contenant 64 quaternions est équivalente à $64 \cdot 4 = 256$ nombres réels. Chaque opération de convolution est basée sur un noyau de taille 3. Le signal contient un *padding* afin de conserver une même dimension des vecteurs en entrée tout au long du traitement. Les modèles sont étudiés en faisant varier le nombre de couches de 4 à 12. Le nombre de matrices de filtres varie de 16 à 64 et de 64 à 256 pour les modèles à valeur réelle et à quaternion, respectivement. En effet, de part la nature des quaternions, une matrice de filtre de quaternions de dimension 16 est équivalente à une matrice de filtres réelle de taille 64. La fonction d’activation ReLU est utilisée pour les deux modèles (Nair, Hinton, 2010). Un bruit de type *dropout* d’une probabilité égale à 0.3 est utilisé pour toutes les couches, à l’exception des couches d’entrée et de sortie. Les CNNs et les QCNNs sont entraînés en se basant sur l’optimiseur Adam (Kingma, Ba, 2014), durant 50 itérations. Les expériences sont effectuées sur des cartes graphiques Tesla P100.

4.5. Résultats et discussions

Cette section fournit les résultats observés pour les CNNs et les QCNNs sur l’identification de thèmes de conversation téléphonique du corpus DECODA. Les meilleures architectures sont d’abord explorées pour les deux modèles, puis comparées aux travaux antérieurs. Tous les résultats proviennent d’une moyenne de 3 expérimentations.

4.5.1. QCNN contre CNN

Le Tableau 1 rapporte les résultats observés pour différentes topologies de QCNNs et de CNNs. Il convient de souligner l’importante différence en terme de nombre de paramètres entre les modèles quaternioniques et les modèles réels. En effet, un réseau convolutif réel de 4 couches contenant chacune 128 FM est composé de 16.8 millions de paramètres, comparé à seulement 4.2 millions pour un QCNN équivalent. Cela est facilement explicable par l’algèbre des quaternions. En effet, une couche dense réelle avec un vecteur d’entrée de taille égale à 256 et une couche cachée de dimension 256 possède $256^2 = 65.5K$ paramètres réels, tandis que pour maintenir des dimensions d’entrée et de sortie égales (256), le modèle quaternionique équivalent possède un vecteur d’entrée de quaternions de taille 64 et 64 unités cachées. Par conséquent, le nombre de paramètres est de $64^2 \cdot 4 = 16.3K$. Une telle réduction de la complexité offre de meilleures performances grâce à une représentation plus compacte de l’information et présente d’autres avantages, tels qu’une empreinte mémoire réduite. Ainsi, un QCNN de taille égale à un CNN utilise toujours quatre fois moins de paramètres.

Une précision maximale de 87.0% est obtenue avec un QCNN de 4 couches convolutives et $FM = 256$, comparé à 85.4 % pour un CNN composé d'un même nombre de couches, mais $FM = 128$. En raison de la petite taille du jeu de données DECODA, les QCNNs et CNNs ont tendance à sur-apprendre avec l'augmentation du nombre de couches. La précision passe donc de 85.0% avec 4 couches à 82.7% avec 12, et de 85.4% à 84.9 % pour les CNNs et les QCNNs, respectivement. Cependant, un nombre plus élevé de matrices de filtres n'entraîne un sur-apprentissage que pour les CNNs. En effet, les QCNNs produisent de meilleurs résultats avec un plus grand nombre de matrices de filtres. Ceci est également expliqué par la propriété de réduction de l'algèbre des quaternions, puisque le nombre de matrices de filtres est en fait 4 fois inférieur, mais produit la même dimension en sortie. Enfin, les QCNNs obtiennent toujours de meilleurs résultats que les CNNs équivalents avec moins de paramètres, et ont tendance à mieux s'adapter aux architectures plus grandes grâce à une représentation plus compacte.

Tableau 1. Résultats de la tâche d'identification de thèmes sur le corpus DECODA, provenant d'une moyenne de 3 lancers. 'L' correspond au nombre de couches convolutives, 'FM' exprime le nombre de matrices de filtres et 'Params' indique le nombre total de paramètres du modèle. 'FM' est exprimé de façon à être équivalent entre les deux algèbres. Ainsi, 64FM correspondent à 64FM pour le CNN et 16 pour le QCNN.

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

4.5.2. QCNN et l'état de l'art sur la tâche de DECODA

De nombreuses expériences ont été menées sur le corpus DECODA afin de fournir une solution fiable à cette problématique. Par conséquent, le Tableau 2 résume tous

les résultats obtenus avec différentes architectures de réseaux de neurones. Il convient de noter que la précision de 87.0% obtenue par notre QCNN est le meilleur résultat observé jusqu'à présent sur cette tâche. De plus, les modèles à base de quaternions obtiennent toujours de meilleurs résultats que leurs concurrents réels.

Tableau 2. Résultats de plusieurs architectures sur la tâche DECODA

Modèles	Type	Test %
MLP (Parcollet <i>et al.</i> , 2017a)	R	83.4
QMLP (Parcollet <i>et al.</i> , 2017a)	H	84.6
DSAE (Janod <i>et al.</i> , 2016)	R	82.0
DAE (Parcollet <i>et al.</i> , 2017b)	R	83.0
QDAE (Parcollet <i>et al.</i> , 2017b)	H	85.2
DNN (Parcollet <i>et al.</i> , 2017a)	R	84.0
QDNN (Parcollet <i>et al.</i> , 2017a)	H	85.2
CNN	R	85.4
QCNN	H	87.0

5. Conclusion

En résumé. Cet article propose de fusionner la représentation de plusieurs tours de conversations téléphoniques par des quaternions, avec le processus bien connu de convolution pour obtenir une meilleure représentation interne et externe des informations pertinentes. Les expériences en compréhension du langage sur le corpus DECODA ont montrées que: 1) le QCNN obtient les meilleurs résultats observés jusqu'à présent durant cette tâche; 2) Les modèles basés sur l'algèbre des quaternions surpassent toujours les modèles réels; 3) Ces améliorations sont observées avec une réduction importante du nombre de paramètres des modèles; Par conséquent, l'intuition initiale selon laquelle la fusion d'une représentation à base de quaternions, avec les réseaux neuronaux convolutifs permettrait au QCNN de mieux modéliser les relations externes et internes en une représentation compacte et efficace a été validée.

Limitations et directions futures. Le corpus DECODA n'est pas assez important pour mettre pleinement en valeur le potentiel des modèles basés sur les quaternions. En conséquence, un futur travail consistera à appliquer les QCNNs à de plus grands corpus textuels. De plus, les réseaux de neurones convolutifs ne modélisent pas de manière adéquate les dépendances séquentielles pertinentes dans les documents textuels et vocaux. Les réseaux de neurones récurrents à base de quaternions (QRNN) seront utilisés dans les futurs travaux pour prendre en compte les dépendances à long et à court terme, afin d'obtenir des résultats encore plus précis.

Remerciements

Les expériences ont été menées avec Keras (Chollet et al., 2015). Les auteurs aimeraient souligner le soutien informatique de Calcul Canada et le soutien financier d'Orkis.

Bibliographie

- Arena P., Fortuna L., Muscato G., Xibilia M. G. (1997). Multilayer perceptrons to approximate quaternion valued functions. *Neural Networks*, vol. 10, n° 2, p. 335–342.
- Arena P., Fortuna L., Occhipinti L., Xibilia M. G. (1994). Neural networks for quaternion-valued function approximation. In *Circuits and systems, iscas'94., ieee international symposium on*, vol. 6, p. 307–310.
- Aspragathos N. A., Dimitros J. K. (1998). A comparative study of three methods for robot kinematics. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, n° 2, p. 135–145.
- Bechet F., Maza B., Bigouroux N., Bazillon T., El-Beze M., De Mori R. *et al.* (2012). Decoda: a call-centre human-human spoken conversation corpus. In *Lrec*, p. 1343–1347.
- Blei D. M., Ng A. Y., Jordan M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, vol. 3, p. 993–1022.
- Chase Gaudet A. M. (2017). Deep quaternion networks. *arXiv preprint arXiv:1712.04604v2*.
- Chollet F. *et al.* (2015). *Keras*. <https://github.com/keras-team/keras>. GitHub.
- Funahashi K.-i., Nakamura Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, vol. 6, n° 6, p. 801–806.
- Gers F. A., Schmidhuber J., Cummins F. (1999). Learning to forget: Continual prediction with lstm.
- Glorot X., Bengio Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, p. 249–256.
- Graves A., Mohamed A.-r., Hinton G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, p. 6645–6649.
- Graves A., Schmidhuber J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, vol. 18, n° 5, p. 602–610.
- Hazen T. J., Richardson F., Margolis A. (2007). Topic identification from audio recordings using word and phone recognition lattices. In *Automatic speech recognition & understanding, 2007. asru. ieee workshop on*, p. 659–664.
- He K., Zhang X., Ren S., Sun J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the ieee international conference on computer vision*, p. 1026–1034.
- He K., Zhang X., Ren S., Sun J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition*, p. 770–778.
- Isokawa T., Kusakabe T., Matsui N., Peper F. (2003). Quaternion neural network and its application. In *International conference on knowledge-based and intelligent information and engineering systems*, p. 318–324.
- Janod K., Morchid M., Dufour R., Linares G., De Mori R. (2016). Deep stacked autoencoders for spoken language understanding. *ISCA INTERSPEECH*, vol. 1, p. 2.

- Kingma D., Ba J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Linares G., Nocéra P., Massonie D., Matrouf D. (2007). The lia speech recognition system: from 10xrt to 1xrt. In *Text, speech and dialogue*, p. 302–308.
- Mikolov T., Karafiát M., Burget L., Cernocký J., Khudanpur S. (2010). Recurrent neural network based language model. In *Interspeech*, vol. 2, p. 3.
- Minemoto T., Isokawa T., Nishimura H., Matsui N. (2017). Feed forward neural network with random quaternionic neurons. *Signal Processing*, vol. 136, p. 59–68.
- Morchid M., Linares G., El-Beze M., De Mori R. (2013). Theme identification in telephone service conversations using quaternions of speech features. In *Interspeech*.
- Nair V., Hinton G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (icml-10)*, p. 807–814.
- Nitta T. (1995). A quaternary version of the back-propagation algorithm. In *Neural networks, 1995. proceedings., ieee international conference on*, vol. 5, p. 2753–2756.
- Parcollet T., Morchid M., Bousquet P.-M., Dufour R., Linares G., De Mori R. (2016). Quaternion neural networks for spoken language understanding. In *Spoken language technology workshop (slt), 2016 ieee*, p. 362–368.
- Parcollet T., Morchid M., Linares G. (2017a). Deep quaternion neural networks for spoken language understanding. In *Automatic speech recognition and understanding workshop (asru), 2017 ieee*, p. 504–511.
- Parcollet T., Morchid M., Linares G. (2017b). Quaternion denoising encoder-decoder for theme identification of telephone conversations. *Proc. Interspeech 2017*, p. 3325–3328.
- Parcollet T., Zhang Y., Morchid M., Trabelsi C., Linares G., Mori R. de *et al.* (2018). 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.*, p. 22–26. Consulté sur <https://doi.org/10.21437/Interspeech.2018-1898>
- Pei S.-C., Cheng C.-M. (1999). Color image processing by using binary quaternion-moment-preserving thresholding technique. *IEEE Transactions on Image Processing*, vol. 8, n° 5, p. 614–628.
- Sabour S., Frosst N., Hinton G. E. (2017). Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829v2*.
- Sangwine S. J. (1996). Fourier transforms of colour images using quaternion or hypercomplex numbers. *Electronics letters*, vol. 32, n° 21, p. 1979–1980.
- Trabelsi C., Bilaniuk O., Serdyuk D., Subramanian S., Santos J. F., Mehri S. *et al.* (2017). Deep complex networks. *arXiv preprint arXiv:1705.09792*.
- Xu D., Zhang L., Zhang H. (2017). Learning algorithms in quaternion neural networks using ghr calculus. *Neural Network World*, vol. 27, n° 3, p. 271.
- Zhang X., Zhao J., LeCun Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, p. 649–657.