Extreme Learning Machines on Cayley-Dickson Algebra Applied for Color Image Auto-Encoding

1st Guilherme Vieira
Department of Applied Mathematics
University of Campinas
Campinas, Brazil
emails: gvieira.mat@gmail.com

2nd Marcos Eduardo Valle
Department of Applied Mathematics
University of Campinas
Campinas, Brazil
emails: valle@ime.unicamp.br

***Abstract*—This paper aims to provide a useful framework for extreme learning machines (ELMs) on Cayley-Dickson algebras. Cayley-Dickson algebras, which include complex numbers, quaternions, and octonions as particular instances, are hypercomplex algebras defined using a recursive procedure. Firstly, we review some basic concepts on Cayley-Dickson algebras and formulate Cayley-Dickson matrix product using real-valued linear algebra. Then, we propose the Cayley-Dickson ELMs and derive their learning using Cayley-Dickson least squares problem. Lastly, we compare the performance of real-valued and four-dimensional Cayley-Dickson ELM models, including quaternion-valued ELM, in an experiment on color image auto-encoding using the well-known CIFAR dataset.**

***Index Terms*—Feedforward neural network, extreme learning machine, hypercomplex number system, Cayley-Dickson algebra.**

I. Introduction

Hypercomplex algebras over the real field are known to be of crucial importance for modern mathematics and applications. For example, complex numbers are vital for digital signal processing and information theory [1]. Quaternions are of unparalleled value for modeling 3-dimensional movement, such as in graphic design and automated control [2]. Octonions, also known as *Cayley's numbers*, are most commonly used in physics, mainly in string theory, special relativity, and quantum logic [3]. It turns out that the multiplication on complex numbers, quaternions, and octonions can be defined using the Cayley-Dickson construction [4].

The Cayley-Dickson construction is a recursive process which yields an infinite sequence of algebras known as Cayley-Dickson algebras [5], [6]. Unfortunately, the Cayley-Dickson construction results weaker algebras. Namely, we know that complex numbers are commutative, associative, and alternative. Quaternions, which can be derived from complex numbers using the Cayley-Dickson construction, are not commutative but associative and alternative. From quaternions, the Cayley-Dickson construction yields octonions. Although alternative, octonions are neither commutative nor associative. The next algebra, called sedenions, are neither commutative, associative, nor alternative. Notwithstanding, in this paper we show that the Cayley-Dickson algebras can be used to define effective hypercomplex-valued neural networks.

Hypercomplex-valued neural networks (HvNNs) extend traditional real-valued neural networks using hypercomplex number systems. Examples of HvNNs include complex-valued [7], [8], hyperbolic-valued [9]–[11], quaternion-valued neural networks [12], [13]. Besides treating high-dimensional data as a single entity, these neural networks can cope appropriately with phase information and rotations, for instance. Applications of HvNNs include signal and image processing [14]–[23], classification and prediction [24]–[29]. In this paper, we focus on hypercomplex-valued extreme learning machines defined on Cayley-Dickson algebras.

Extreme learning machines (ELM) are a type of feedforward neural networks introduced by Huang [30] in the early 2000s. This architecture consists of a multilayer perceptron (MLP) in which all but one layer have fixated randomly initialized parameters, and training on the last layer is performed by least squares optimization. This has been proven to maintain the universal approximation capabilities of MLP while also drastically decreasing computational complexity and, thus, training time [31]–[33]. Complex-valued and quaternion-valued ELMs have been developed respectively by Li et al. [34] and Minemoto et al [28] as well as Lv et al. [35]. Furthermore, complex-valued and quaternion-valued ELMs have outperformed real-valued ELMs with compatible total number of trainable parameters for certain high-dimensional data related tasks. This motivated us to investigate ELMs on Cayley-Dickson algebras as tools for high-dimensional data processing, common to image processing, time series forecasting and general classification and regression tasks. Precisely, in this paper we focus on ELMs defined on four-dimensional Cayley-Dickson algebras with application as color image auto-encoders.

The paper is organized as follows: Section II reviews basic concepts of Cayley-Dickson algebras. Matrix operations in Cayley-Dickson algebras and their equivalence to real linear algebra are detailed in Section III. Section IV introduces the Cayley-Dickson ELMs as well as their training algorithm. Computational experiments concerning color image auto-encoding are decribed in Section V. The paper finishes with the concluding remarks in Section VI.

This work was supported in part by CNPq under grant no. 310118/2017-4, FAPESP under grant no. 2019/02278-2, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

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

II. A Brief Review on Cayley-Dickson Algebras

Let us begin by recalling the origins of Cayley-Dickson algebras. Basically, the Cayley-Dickson algebras generalize the complex-numbers, quaternions, and octonions. Although octonions have been discovered by Graves [3], they are also called Cayley numbers due to the works of Cayley in mid 19th century. In 1919, Dickson formulated a recursive process from which complex numbers are obtained from real numbers, quaternions are obtained from complex numbers, and octonions are obtained from quaternions [36]. Later, in 1922, Albert formalized this iterative process which is known as the *Cayley-Dickson construction* [37]. Albert's work allows for the construction of an infinitely long sequence of algebras with doubling dimension, as observed in complex-numbers $\mathbb{C}$, quaternions $\mathbb{Q}$, and octonions $\mathbb{O}$ of dimensions respectively 2, 4, and 8.

In this section we review the basic definitions regarding the Cayley-Dickson construction. Although Cayley-Dickson algebras can be defined on an arbitrary field $\mathbb{F}$, in this paper we only consider real numbers as the ground field. Furthermore, we shall adopt the generalized construction presented in [6].

Definition 1 (Generalized Cayley-Dickson Algebras)**.** *The first Cayley-Dickson algebra is $\mathbb{A}_0 = \mathbb{R}$, the real numbers, with the identity mapping as the conjugate. Given a non-zero scalar $\gamma_k \in \mathbb{R}$ called generator, the kth Cayley-Dickson algebra is defined recursively as the Cartesian product $\mathbb{A}_{k-1} \times \mathbb{A}_{k-1}$, that is,*

$$\mathbb{A}_k = \{(x,y) : x, y \in \mathbb{A}_{k-1}\}, \quad k \geq 1, \tag{1}$$

where addition, scalar product, multiplication, and conjugate are defined, using the operations from A_{k-1}, as follows for all $(x,y),(z,w) \in \mathbb{A}_k$ and $\alpha \in \mathbb{R}$:

- *Addition:*
$$(x,y) + (z,w) = (x+z, y+w). \tag{2}$$
- *Scalar product:*
$$\alpha(x,y) = (\alpha x, \alpha y). \tag{3}$$
- *Multiplication:*
$$(x,y)(z,w) = (xz + \gamma_k w^* y, wx + yz^*). \tag{4}$$
- *Conjugation:*
$$(x,y)^* = (x^*, -y). \tag{5}$$

It can be easily verified that algebras derived from the Cayley-Dickson construction satisfy

$$\dim \mathbb{A}_k = 2 \dim \mathbb{A}_{k-1} = 2^k \dim \mathbb{R} = 2^k.$$

Therefore, the dimension of the algebras obtained by Definition 1 is a power of two.

Remark 1. The Cayley-Dickson algebra depends on the previous algebra $\mathbb{A}_{k-1}$ and the generator γ_k. Therefore, we can write

$$\mathbb{A}_k \equiv \mathbb{A}_{k-1}[\gamma_k]. \tag{6}$$

TABLE I
Multiplication table of quaternions.

$\mathbb{R}[-1,-1]$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_1$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2$	$-\boldsymbol{e}_1$	$\boldsymbol{e}_4$	$-\boldsymbol{e}_3$
$\boldsymbol{e}_3$	$\boldsymbol{e}_3$	$-\boldsymbol{e}_4$	$-\boldsymbol{e}_1$	$\boldsymbol{e}_2$
$\boldsymbol{e}_4$	$\boldsymbol{e}_4$	$\boldsymbol{e}_3$	$-\boldsymbol{e}_2$	$-\boldsymbol{e}_1$

More generally, we can derive the Cayley-Dickson algebra $\mathbb{A}_k$ from $\mathbb{R}$ using generators $\gamma_1, \gamma_2, \ldots, \gamma_k$. In other words, we have

$$\mathbb{A}_k \equiv \mathbb{R}[\gamma_1, \gamma_2, \ldots, \gamma_k]. \tag{7}$$

As pointed out previously, Definition 1 is obtained using the generalized construction presented in [5], [6]. Many researchers, however, consider generators $\gamma_k = -1$ for all $k \geq 1$. Aiming to keep the formulation as general as possible, we consider γ_k as a non-zero scalar.

In analogy to complex numbers, an element $(x,y) \in \mathbb{A}_k$ can be written as $x + y\boldsymbol{u}_k$, where $\boldsymbol{u}_k \in \mathbb{A}_k$ is a hypercomplex unit in $\mathbb{A}_k$ [6]. This hypercomplex unit is defined by $\boldsymbol{u}_k = (0,1)$, where 0 and 1 denote respectively the zero and the identity element of $\mathbb{A}_{k-1}$. Furthermore, it satisfies the equation $\boldsymbol{u}_k^2 = \gamma_k$, where γ_k is the generator of $\mathbb{A}_k$. The element $(1,0)$, denoted by $1 \in \mathbb{A}_k$, is the identity of $\mathbb{A}_k$.

Alternatively, we can represent an element $x \in \mathbb{A}_k$ using an appropriate hypercomplex basis [5]. Precisely, the addition and the scalar product of $\mathbb{A}_k$ yields a vector space of dimension $n = 2^k$. Using the canonical basis, we can write

$$x = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + \ldots + x_n\boldsymbol{e}_n, \tag{8}$$

where $\boldsymbol{e}_1 = (1, 0, \ldots, 0)$ corresponds to the identity and $\boldsymbol{e}_2 = (0, 1, \ldots, 0), \ldots, \boldsymbol{e}_n = (0, \ldots, 1)$ are hypercomplex units. The product of any two hypercomplex units define a multiplication table. The multiplication table also characterizes the hypercomplex algebra [38], [39].

Example 1 (Complex Numbers, Quaternions, and Octonions)**.** By taking $\gamma_1 = -1$, we obtain

$$\mathbb{A}_1 = \mathbb{R}[-1] = \{x + y\boldsymbol{u}_1 : \boldsymbol{u}_1^2 = -1, x, y \in \mathbb{R}\} \simeq \mathbb{C}$$

where $\simeq$ means that there exists an isomorphism between $\mathbb{A}_1$ and $\mathbb{C}$. In other words, $\mathbb{R}[-1]$ and $\mathbb{C}$ are equivalent. In a similar fashion, by taking $\gamma_2 = -1$, we obtain

$$\mathbb{A}_2 = \mathbb{A}_1[-1] = \{x + y\boldsymbol{u}_2 : \boldsymbol{u}_2^2 = -1, x, y \in \mathbb{A}_1\}.$$

It is not hard to show that the algebra $\mathbb{A}_2 = \mathbb{R}[-1,-1]$ is equivalent to the quaternions, denoted by $\mathbb{Q}$. The multiplication table of the quaternions is given in Table I. The next Cayley-Dickson algebra $\mathbb{A}_3 = \mathbb{R}[-1,-1,-1]$, obtained from the generator $\gamma_3 = -1$, is equivalent to the octonions, denoted by $\mathbb{O}$ and also known as Cayley numbers [3].

Example 1 illustrates an important fact about the Cayley-Dickson construction: The next algebra added in the sequence suffers a loss of properties. For instance, complex numbers are not an ordered set as are real numbers. The product in the quaternions is not commutative, even though complex product is. The product in the octonions is not associative, a property possessed by $\mathbb{R}$, $\mathbb{C}$ and $\mathbb{Q}$. Despite the loss of properties, which may discourage the use of higher Cayley-Dickson algebras, the next sections show that this is not a problem for the development of effective feedforward neural networks. Before addressing neural networks, however, we would like to address the four dimensional Cayley-Dickson algebras.

Example 2 (Four Dimensional Cayley-Dickson Algebras)**.** In analogy to the quaternions, Table II shows the multiplication tables of the Cayley-Dickson algebras $\mathbb{R}[+1,+1]$, $\mathbb{R}[-1,+1]$, and $\mathbb{R}[+1,-1]$, which have dimension 4. The multiplication table of the algebra $\mathbb{R}[-1,+1]$ reveals that this algebra is equivalent to the algebra of coquaternions, also called split-quaternions, introduced by Cockle in 1849 [40]. Note, at the diagonal of the multiplication tables, that $\boldsymbol{e}_1^2 = \boldsymbol{e}_1$, $\boldsymbol{e}_2^2 = \gamma_1 \boldsymbol{e}_1$, $\boldsymbol{e}_3^2 = \gamma_2 \boldsymbol{e}_1$, and $\boldsymbol{e}_4^2 = -(\gamma_1\gamma_2)\boldsymbol{e}_1$.

Definition 2 (Cayley-Dickson Norm [6])**.** *The Cayley-Dickson norm in $\mathbb{A}_k$ is defined by*

$$n(x) = xx^*, \quad \forall x \in \mathbb{A}_k. \tag{9}$$

It is not hard to show that $n(x)$ is a multiple of the identity $\boldsymbol{e}_1$ of $\mathbb{A}_k$ for all $x \in \mathbb{A}[\gamma]$. As a consequence, we can interpret $n(x)$ as a scalar. Also, we can alternatively define $n(x) = x^*x$. Moreover, an element $x \in \mathbb{A}_k$ has an inverse if $n(x) \neq 0$. In this case, the inverse of x is $x^{-1} = n(x)^{-1}x^*$ and its norm is $n(x^{-1}) = n(x)^{-1}$, where $n(x)$ is interpreted as a real number.

Example 3. Consider the algebra $\mathbb{A}_1 = \mathbb{R}[-1]$, which is equivalent to the complex numbers. The norm and the inverse of an element $z = (x,y) \in \mathbb{A}_1$ are respectively

$$n(z) = (x^2+y^2, 0) \equiv x^2+y^2 \quad \text{and} \quad z^{-1} = \frac{1}{x^2+y^2}(x,-y).$$

Note that $n(z)$ corresponds to the square of the absolute value of the complex number $z = x + y\boldsymbol{i}$.

Remark 2. We would like to emphasize that the Cayley-Dickson norm is *not* a norm in the linear algebra and functional analysis sense. In fact, we may have $n(x) = 0$ for some $x \neq 0$ in some Cayley-Dickson algebras. The following example illustrate this remark.

Example 4 (Hyperbolic Numbers)**.** Hyperbolic numbers, also called double numbers or split-complex numbers, have many applications in geometry and interval algebra [41], [42]. They are equivalent to the algebra $\mathbb{R}[+1]$, obtained from $\mathbb{R}$ by choosing the generator $\gamma_1 = 1$. A simple calculation in $\mathbb{R}[+1]$ yields

$$n(z) = x^2 - y^2, \quad \forall z = (x,y).$$

Clearly $n(z) = 0$ for all $z = (\pm x, \pm x)$. This shows that (9) does *not* satisfy the definition of norm in linear algebra, which requires $n(z) = 0$ if and only if $z = (0,0)$.

Apart from the Cayley-Dickson norm defined by (9), let us define the absolute value of $x = x_1\boldsymbol{e}_1 + \ldots + x_n\boldsymbol{e}_n \in \mathbb{A}_k$ by means of the following equation:

$$|x| = \sqrt{\sum_{i=1}^{n} |x_i|^2}. \tag{10}$$

In contrast to the Cayley-Dickson norm, the absolute value satisfies $|x| = 0$ if and only if $x = 0$.

III. Cayley-Dickson Matrix Algebra

Let us turn our attention to operations on matrices whose entries are elements of a Cayley-Dickson algebra $\mathbb{A}_k$. Such as real-valued linear algebra, the product of two matrices $\boldsymbol{A} \in \mathbb{A}_k^{M\times L}$ and $\boldsymbol{B} \in \mathbb{A}_k^{L\times N}$ is a new matrix $\boldsymbol{C} \in \mathbb{A}_k^{M\times N}$ with entries defined by

$$c_{ij} = \sum_{\ell=1}^{L} a_{i\ell} b_{\ell j}, \tag{11}$$

where $i = 1,\ldots,M$ and $j = 1,\ldots,N$. Here, $a_{i\ell}$ and $b_{\ell j}$ are entries of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively.

In order to take advantage of fast scientific computing softwares, in practice, we compute the Cayley-Dickson matrix product using real-valued linear algebra as follows. First, (8) allow us to establish an isomorphism between $\mathbb{A}_k$ and $\mathbb{R}^n$, where $n = 2^k$, by means of the following equation

$$\varphi(x) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \forall x \in \mathbb{A}_k. \tag{12}$$

Since the multiplication on $\mathbb{A}_k$ is a linear operator, it admits matrix representation in $\mathbb{R}^n$. Specifically, let $a \in \mathbb{A}_k$ be a fixed element. The multiplication to the left by a yields a linear operator $\mathcal{A}_L : \mathbb{A}_k \to \mathbb{A}_k$ defined by $\mathcal{A}_L(x) = ax$ for all $x \in \mathbb{A}_k$. From the isomorphism φ, we conclude that

$$\varphi\left(\mathcal{A}_L(x)\right) = \Phi_L(a)\varphi(x), \tag{13}$$

where $\Phi_L : \mathbb{A}_k \to \mathbb{R}^{n\times n}$ is the matrix representation of $\mathcal{A}_L$ with respect to the canonical basis [43], that is,

$$\Phi_L(a) = \begin{bmatrix} | & | & & | \\ \varphi(a\boldsymbol{e}_1) & \varphi(a\boldsymbol{e}_2) & \ldots & \varphi(a\boldsymbol{e}_n) \\ | & | & & | \end{bmatrix}. \tag{14}$$

Now, from (12), an entry c_{ij} of the Cayley-Dickson valued matrix $\boldsymbol{C}$ given by (11) satisfies

$$\begin{aligned} \varphi(c_{ij}) &= \sum_{\ell=1} \Phi_L(a_{i\ell})\varphi(b_{\ell j}) \\ &= \begin{bmatrix} \Phi_L(a_{i1} & \Phi_L(a_{i2}) & \ldots & \Phi_L(a_{iL}) \end{bmatrix} \begin{bmatrix} \varphi(b_{1j}) \\ \varphi(b_{2j}) \\ \vdots \\ \varphi(b_{Lj}) \end{bmatrix}. \end{aligned}$$

Equivalently, using real-valued matrix operations, we have

$$\varphi(\boldsymbol{C}) = \Phi_L(\boldsymbol{A})\varphi(\boldsymbol{B}), \tag{15}$$

TABLE II
MULTIPLICATION TABLES OF THREE FOUR-DIMENSIONAL CAYLEY-DICKSON ALGEBRAS.

$\mathbb{R}[+1,+1]$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_1$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2$	$\boldsymbol{e}_1$	$\boldsymbol{e}_4$	$\boldsymbol{e}_3$
$\boldsymbol{e}_3$	$\boldsymbol{e}_3$	$-\boldsymbol{e}_4$	$\boldsymbol{e}_1$	$-\boldsymbol{e}_2$
$\boldsymbol{e}_4$	$\boldsymbol{e}_4$	$-\boldsymbol{e}_3$	$\boldsymbol{e}_2$	$-\boldsymbol{e}_1$

$\mathbb{R}[-1,+1]$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_1$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2$	$-\boldsymbol{e}_1$	$\boldsymbol{e}_4$	$-\boldsymbol{e}_3$
$\boldsymbol{e}_3$	$\boldsymbol{e}_3$	$-\boldsymbol{e}_4$	$\boldsymbol{e}_1$	$-\boldsymbol{e}_2$
$\boldsymbol{e}_4$	$\boldsymbol{e}_4$	$\boldsymbol{e}_3$	$\boldsymbol{e}_2$	$\boldsymbol{e}_1$

$\mathbb{R}[+1,-1]$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_1$	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2$	$\boldsymbol{e}_1$	$\boldsymbol{e}_4$	$\boldsymbol{e}_3$
$\boldsymbol{e}_3$	$\boldsymbol{e}_3$	$-\boldsymbol{e}_4$	$-\boldsymbol{e}_1$	$\boldsymbol{e}_2$
$\boldsymbol{e}_4$	$\boldsymbol{e}_4$	$-\boldsymbol{e}_3$	$-\boldsymbol{e}_2$	$\boldsymbol{e}_1$

where Φ_L and ϕ are defined as follows for Cayley-Dickson matrix arguments:

$$\Phi_L(\boldsymbol{A}) = \begin{bmatrix} \Phi_L(a_{11}) & \Phi_L(a_{12}) & \dots & \Phi_L(a_{1L}) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_L(a_{M1}) & \Phi_L(a_{M2}) & \dots & \Phi_L(a_{ML}) \end{bmatrix}, \quad (16)$$

and

$$\varphi(\boldsymbol{B}) = \begin{bmatrix} \varphi(b_{11}) & \dots & \varphi(b_{1N}) \\ \varphi(b_{21}) & \dots & \varphi(b_{2N}) \\ \vdots & \ddots & \vdots \\ \varphi(b_{L1}) & \dots & \varphi(b_{LN}) \end{bmatrix}. \quad (17)$$

Note that $\Phi_L(\boldsymbol{A}) \in \mathbb{R}^{(nM)\times(nL)}$ and $\varphi(\boldsymbol{B}) \in \mathbb{R}^{(nL)\times N}$. The real-valued matrix $\varphi(\boldsymbol{C}) \in \mathbb{R}^{(nM)\times N}$ is defined analogously to (17). Furthermore, the Cayley-Dickson matrix $\boldsymbol{C} \in \mathbb{A}_k^{M\times N}$ is obtained reorganizing the elements of $\varphi(\boldsymbol{C})$. Formally, reorganizing the elements of $\varphi(\boldsymbol{C})$ defines the inverse mapping $\varphi^{-1} : \mathbb{R}^{(nM)\times N} \to \mathbb{A}_k^{M\times N}$. More importantly, we have

$$\boldsymbol{C} = \varphi^{-1}\left(\Phi_L(\boldsymbol{A})\varphi(\boldsymbol{B})\right), \quad (18)$$

which provides an effective formula for the computation of Cayley-Dickson matrix product using the real-valued linear algebra often available in scientific computing softwares.

We would like to point out that, in a similar fashion, we can compute the Cayley-Dickson matrix product $\boldsymbol{C} = \boldsymbol{AB}$ by considering in (11) the multiplication from the right by $b_{\ell j}$ instead of the multiplication from the left by $a_{i\ell}$. Formally, the multiplication from the right by $b \in \mathbb{A}_k$ yields a mapping $\mathcal{B}_R(x) = xb$ for all $x \in \mathbb{A}_k$. In analogy to (18) but using the matrix representation Φ_R of $\mathcal{B}_R$, we obtain the identity

$$\boldsymbol{C} = \phi^{-1}\left(\Phi_R(\boldsymbol{B})\phi(\boldsymbol{A})\right). \quad (19)$$

From the computational point of view, the difference between (18) and (19) is the construction of the real-valued matrices $\Phi_L(\boldsymbol{A})$ and $\Phi_R(\boldsymbol{B})$. Hence, (18) is faster than (19) if the matrix $\boldsymbol{A}$ has less entries than $\boldsymbol{B}$. We implemented both (18) and (19) in a single code that computes the Cayley-Dickson product using the fastest formula.

Let us now address the linear least square problem on Cayley-Dickson algebra, a key concept for training extreme learning machines (ELMs). First, in analogy to the real-valued case [44], the Frobenius norm of a matrix $\boldsymbol{A} \in \mathbb{A}_k^{M\times N}$ is given by

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} |a_{ij}|^2}, \quad (20)$$

where $|a_{ij}|$ denotes the absolute value of $a_{ij} \in \mathbb{A}_k$ given by (10). From (10) and (12), we have $|x| = \|\varphi(x)\|_2$, where $\|\cdot\|_2$ denotes the usual Euclidean norm. Therefore, we have

$$\|\boldsymbol{A}\|_F = \|\varphi(\boldsymbol{A})\|_F. \quad (21)$$

Using the Frobenius norm, we define the linear least saquare problem as follows:

Definition 3 (Cayley-Dickson Least Square Problem)**.** *Given matrices $\boldsymbol{A} \in \mathbb{A}_k^{M\times L}$ and $\boldsymbol{B} \in \mathbb{A}_k^{M\times N}$, the Cayley-Dickson least squares problem consists of finding the minimal Frobenius norm solution to the problem*

$$\min\left\{\|\boldsymbol{AX} - \boldsymbol{B}\|_F : \boldsymbol{X} \in \mathbb{A}_k^{L\times N}\right\}. \quad (22)$$

Like the matrix product, in practice we solve a Cayley-Dickson linear least squares problem using real-valued linear algebra. Precisely, from (21) and (15), we obtain

$$\begin{aligned} \|\boldsymbol{AX} - \boldsymbol{B}\|_F &= \|\varphi(\boldsymbol{AX} - \boldsymbol{B})\|_F = \|\varphi(\boldsymbol{AX}) - \varphi(\boldsymbol{B})\|_F \\ &= \|\Phi_L(\boldsymbol{A})\varphi(\boldsymbol{X}) - \varphi(\boldsymbol{B})\|_F. \end{aligned}$$

Therefore, the solution of the Cayley-Dickson least squares problem is equivalent to the unique minimal solution to the real-valued problem:

$$\min\{\|\Phi_L(\boldsymbol{A})\boldsymbol{X}^{(r)} - \varphi(\boldsymbol{B})\|_F : \boldsymbol{X}^{(r)} \in \mathbb{R}^{(nL)\times N}\}, \quad (23)$$

where $n = 2^k$. It turns out, however, that the solution of (23) is

$$\boldsymbol{X}^{(r)} = \Phi_L(\boldsymbol{A})^\dagger\varphi(\boldsymbol{B}), \quad (24)$$

where $\Phi_L(\boldsymbol{A})^\dagger$ denotes the pseudoinverse of the real-valued matrix $\Phi_L(\boldsymbol{A})$ [44]. Concluding, analogously to (18), the solution of the Cayley-Dickson least squares problem (22) is given by the following equation using real-valued matrix computations:

$$\boldsymbol{X} = \varphi^{-1}\left(\Phi_L(\boldsymbol{A})^\dagger\varphi(\boldsymbol{B})\right). \quad (25)$$

From the computational point of view, the computation of the matrix $\boldsymbol{X} \in \mathbb{A}_k^{L\times N}$ given by (25) is dominated by the construction of the matrix $\Phi_L(\boldsymbol{A})$ and the computation of its pseudo-inverse. In general terms, the computation of the pseudo-inverse of the real-valued matrix $\Phi(\boldsymbol{A}) \in \mathbb{R}^{(nM)\times(nL)}$ requires $\mathcal{O}(2^n M^2 L)$ arithmetic operations, where $n = 2^k$, when $M > L$ [44].

IV. Extreme Learning Machines on Cayley-Dickson Algebras

Extreme learning machines (ELMs) are feedforward neural network models well known for their extremely low computational cost [30]–[32], [45]. This characteristic is due to the parameters of all hidden layers being randomly generated and fixated. The free trainable parameters of the network are all present in the output layer. Moreover, training the output layer is often formulated as a least squares problem and, thus, it is achieved in a finite number of operations.

First of all, let us clarify how a single-layer feed-forward neural network is defined on a Cayley-Dickson algebra $\mathbb{A}_k$. The parameters of a hidden layer with L neurons are represented by a Cayley-Dickson matrix $\boldsymbol{W} \in \mathbb{A}_k^{D \times L}$. Given a Cayley-Dickson row vector $\boldsymbol{x} = [x_1, \ldots, x_D] \in \mathbb{A}_k^D$ as input, the feed-forward step through the hidden layer yields

$$\boldsymbol{h} = f(\boldsymbol{x}\boldsymbol{W}) \in \mathbb{A}_k^L, \tag{26}$$

where $f : \mathbb{A}_k \to \mathbb{A}_k$ is a non-linear activation function defined in an entry-wise manner for matrices. The activation function f is generally taken as a split function, i.e., a real-valued non-linear function applied separately to each component of the multi-dimensional element. For example, the split hyperbolic tangent function $\tanh : \mathbb{A}_k \to \mathbb{A}_k$ is defined as follows for any $x = x_1\boldsymbol{e}_1 + \ldots + x_n\boldsymbol{e}_n \in \mathbb{A}_k$, $n = 2^k$:

$$\tanh(x) = \tanh(x_1)\boldsymbol{e}_1 + \ldots + \tanh(x_n)\boldsymbol{e}_n. \tag{27}$$

Similarly, the output layer parameters are represented by a matrix $\boldsymbol{M} \in \mathbb{A}_k^{L \times N}$. As usual, there is no activation function in the output layer of a Cayley-Dickson ELM. Thus, the output of the CD-ELM is given by the vector-matrix product

$$\boldsymbol{y} = \boldsymbol{h}\boldsymbol{M} \in \mathbb{A}_k^N. \tag{28}$$

Analogously to real-valued models, ELMs on Cayley-Dickson algebras (CD-ELMs) have randomly generated Cayley-Dickson parameters in hidden layers while output layer parameters are trained using the Cayley-Dickson least squares problem (see Definition 3). Specifically, given a training set $\mathcal{T} = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) : i = 1, \ldots, M\} \subset \mathbb{A}_k^D \times \mathbb{A}_k^N$, let us organize the M input and target samples as rows of Cayley-Dickson matrices $\boldsymbol{X} \in \mathbb{A}_k^{M \times D}$ and $\boldsymbol{T} \in \mathbb{A}_k^{M \times N}$, respectively. The parameters of the hidden layer are randomly generated in $\mathbb{A}_k^{D \times L}$. For example, we can consider

$$w_{ij} = \alpha(\texttt{randn}\boldsymbol{e}_1 + \ldots + \texttt{randn}\boldsymbol{e}_n), \tag{29}$$

where α is a scaling factor and randn yields a normally distributed random number with mean 0 and variance 1. In our implementations, knowing *a priori* that the entries $x_i = x_{i1}\boldsymbol{e}_1 + \ldots + x_{in}\boldsymbol{e}_n$, $i = 1, \ldots, D$, of an input $\boldsymbol{x}$ satisfy $-1 \leq x_{ij} \leq +1$ for all i and $j = 1, \ldots, n$, we used $\alpha = 10/D$. Hence, the output $\boldsymbol{h}$ of the hidden layer does not have many saturated values. Finally, the parameters of the output layer are determined by solving the Cayley-Dickson least squares problem

$$\min\{\|\boldsymbol{H}\boldsymbol{M} - \boldsymbol{T}\|_F : \boldsymbol{M} \in \mathbb{A}_k^{L \times N}\}, \tag{30}$$

where $\boldsymbol{H} = f(\boldsymbol{X}\boldsymbol{W})$ is the hidden layer output matrix of the Cayley-Dickson neural network. From (25), we have

$$\boldsymbol{M} = \varphi^{-1}\left(\Phi_L(\boldsymbol{H})^\dagger \varphi(\boldsymbol{T})\right), \tag{31}$$

where Φ_L and φ are the operators defined respectively by (16) and (17) and $\Phi_L(\boldsymbol{H})^\dagger$ is the pseudoinverse of $\Phi_L(\boldsymbol{H})$. Recall that, apart from the transformations Φ_L, φ, and φ^{-1}, the computation of $\Phi_L(\boldsymbol{H})^\dagger$ requires $\mathcal{O}(2^{3k}M^2L)$ operations when $M > L$.

V. Computational Experiments - Color Image Auto-Encoder

An experiment on color image auto-encoding was conducted in order to evaluate the Cayley-Dickson ELMs capabilities. We also compared the performance of the hypercomplex-valued models with a real-valued ELM. To this end, we considered the CIFAR-10 as it is an open common benchmark dataset [46]. This database consists of 60,000 natural 32×32 8-bit RGB-encoded images originally designed for classification tasks. Elements in the CIFAR-10 fall into 10 different classes. The original training data has 50,000 elements split into 5 batches, the remaining 10,000 images are a test batch.

Despite the CIFAR-10 being conceived as a classification benchmark dataset, several works have performed encoding tasks using this set [28], [47], [48]. Auto-encoding is one such task of particular interest. It consists of the task of training the network by presenting it pairs where the input and desired output images are the same, while the dimension of the hidden layer is lesser than the dimension of the input. This forces the model to select a subset of important features of the input and later reconstruct a higher dimensional object with fewer signals. There is a clear interest in this task from the standpoint of information theory because it allows the exchanging of compressed minimal-loss information. Moreover, auto-encoders are powerful feature detectors and, thus, they can be used for unsupervised pre-training on large datasets. Finally, some auto-encoders can also be used as generative models [49].

In this experiment, one real-valued ELM and four four-dimensional Cayley-Dickson algebra-valued ELMs were trained with a single batch of the CIFAR-10 dataset, i.e., using a total of 10,000 color images. The four Cayley-Dickson algebras used are $\mathbb{R}[\pm 1, \pm 1]$, that is, the quaternions and the three algebras whose multiplication tables are given by Table II.

A 32×32 8bit RGB-encoded image $\mathbf{I}$ has been converted into a real-valued vector $\boldsymbol{x}^{(r)}$ of length 3072 by concatenating the red, green, and blue channel and re-scaling the pixel values to the interval $[-1, +1]$. The color image $\mathbf{I}$ has also been converted into a four-dimensional Cayley-Dickson vector $\boldsymbol{x}^{(CD)}$ of length 1024 whose components are

$$x_i^{(CD)} = \left(\frac{2\mathbf{I}_i^R}{255} - 1\right)\boldsymbol{e}_2 + \left(\frac{2\mathbf{I}_i^G}{255} - 1\right)\boldsymbol{e}_3 + \left(\frac{2\mathbf{I}_i^B}{255} - 1\right)\boldsymbol{e}_4,$$

where $\mathbf{I}_i^R$, $\mathbf{I}_i^G$, and $\mathbf{I}_i^B$ denote respectively the red, green, and blue values of the ith pixel of the image, for $i = 1, \ldots, 1024$.

Both real-valued and Cayley-Dickson ELMs have a single hidden layer with L neurons. The size of the hidden layer for each network has been determined using the total number of parameters (TNP). The TNP is calculated as the sum of the number of free trainable parameters and randomly generated fixed parameters. Therefore, a real-valued network with $L^{(r)}$ neurons in the hidden layer has

$$\text{TNP}^{(r)} = 2D^{(r)}L^{(r)}$$

parameters, where $D^{(r)} = 3072$ corresponds to the dimension of the real-valued input vectors. The multiplication factor 2 appears because in an auto-encoding task the desired output is equal to the input, that is, $\boldsymbol{t}_i = \boldsymbol{x}_i$ for all $i = 1, \ldots, 10000$. Thus, each network has D neurons in the output layer. Analogously, a Cayley-Dickson ELM has

$$\text{TNP}^{(CD)} = 4(2D^{(CD)}L^{(CD)})$$

parameters, where $D^{(CD)} = 1024$ is the dimension of the Cayley-Dickson input vectors. Based on [28], which proposed real and quaternion-valued ELMs for auto-encoding CIFAR color images, the number of hidden layer neurons for the real-valued and Cayley-Dickson ELMs have been set respectively as $L^{(r)} = 600$ and $L^{(CD)} = 450$. As a consequence, both real and hypercomplex-valued networks have $\text{TNP}^{(r)} = 3,686,400$.

The (split) hyperbolic tangent has been used as the activation function on the hidden layer of the ELM models. Furthermore, the parameters of the hidden layer were randomly generated according to a standard normal distribution and re-scaled by a constant α relative to the dimension of the images. We used $\alpha^{(r)} = 30/3072$ and $\alpha^{(CS)} = 10/1024$ for the real-valued and Cayley-Dickson valued models, respectively. This was done to ensure the output values of hyperbolic tangent would not be saturated.

One test batch of the CIFAR-10, also containing 10,000 images, was used as test set for the experiment. For illustrative purposes, Figs 1 shows original color images from the CIFAR dataset and the corresponding color images decoded by the real and Cayley-Dickson ELM auto-encoders. The first and second rows correspond respectively to training and testing images. Visually, the five auto-encoders performed well in both training and test images.

To evaluate quantitatively the performance of the ELM auto-encoders two metrics have been used: the peak noise-to-signal ratio (PSNR) and the structural similarity index (SSIM) [50]. The PSNR is a logarithmic scale inversely proportional to the mean squared error. Thus, a higher PSNR value means a higher quality of reconstruction. The structural similarity index is a value in the interval $[-1, 1]$, in which 1 represents perfect similarity between the original and the reconstructed image. We consider the PSNR and SSIM to be fairly different yet robust metrics for the auto-encoding task. The results obtained by the ELM auto-encoders are reported in Table III as well as the boxplots shown in Fig. 2.

It is clear from Table III that all models presented similar PSNR and SSIM rates when comparing training and test set results. Therefore, the five ELM auto-encoder exhibited adequate generalization performance. Let us now compare the real-valued and Cayley-Dickson models. In agreement with the results provided by Minemoto et al. [28], the hypercomplex-valued networks outperformed the real-valued model by a significant margin. Except for the quaternion-valued ELM, the Cayley-Dickson models showed minimal loss in terms of the SSIM index. Moreover, the ELM auto-encoder based on the Cayley-Dickson algebra $\mathbb{R}[+1, -1]$ presented the largest PSNR rates, slightly outperforming the ones based on the algebras $\mathbb{R}[-1, +1]$ and $\mathbb{R}[+1, +1]$. In fact, based on paired Stundent t-test as well as Wilcoxon signed-rank test, there is enough evidence to claim that the Cayley-Dickson ELM based on $\mathbb{R}[+1, -1]$ outperformed the other auto-encoders considered in this experiment with respect to both PSNR and SSIM measures on the test set at 99% confidence level.

TABLE III

AVERAGE PSNR AND SSIM RATES ACHIEVED BY REAL-VALUED AND CAYLEY-DICKSON ELM AUTO-ENCODERS IN THE TRAIN AND TEST SETS.

	Train Set		**Test Set**	
Model	PSNR	SSIM	PSNR	SSIM
Real-valued	27.3 ± 2.4	$0.90 \pm .05$	26.8 ± 2.7	$0.89 \pm .05$
$\mathbb{R}[-1, -1]$	28.9 ± 2.5	$0.93 \pm .04$	28.5 ± 2.7	$0.92 \pm .05$
$\mathbb{R}[-1, +1]$	31.0 ± 2.5	$\mathbf{0.95 \pm .03}$	30.5 ± 2.7	$\mathbf{0.95 \pm .04}$
$\mathbb{R}[+1, -1]$	$\mathbf{31.2 \pm 2.5}$	$\mathbf{0.95 \pm .03}$	$\mathbf{30.7 \pm 2.7}$	$\mathbf{0.95 \pm .04}$
$\mathbb{R}[+1, +1]$	30.9 ± 2.5	$\mathbf{0.95 \pm .03}$	30.4 ± 2.7	$\mathbf{0.95 \pm .04}$

Finally, from the computational point of view, the Cayley-Dickson ELMs are much more time demanding than the real-valued ELM model mainly due to the transformations Φ_L and Φ_R required on (18), (19), and (25). Precisely, in our computational experiments, the training phase of the real-valued ELM have been concluded in approximately 3.77s while the training of the hypercomplex-valued ELM on $\mathbb{R}[+1, -1]$ required approximately 263.05s, approximately 70 times more time demanding than the real-valued model. Moreover, 77% of the time required to train the Cayley-Dickson ELM (204.24s of 263.05s) was spent to determine the real-valued matrix $\Phi_L(\boldsymbol{H})$. At this point, however, we would like to point out that we implemented general purpose recursive codes for computing the Cayley-Dickson multiplication and conjugation based on Definition 1. We believe that the time required by the Cayley-Dickson ELMs can be significantly reduced by implementing application specific hypercomplex-valued operations.

VI. Concluding Remarks

In this paper we explored the concept of extreme learning machines (ELMs) on Cayley-Dickson algebras. It has been shown that quaternion ELMs are a valuable tool and perform better than real-valued ELMs in some situations [28]. Definition 1 encompasses a more general family of algebras. We showed that quaternions are a particular element of this family on Example 1. By means of (12)-(17) we established a

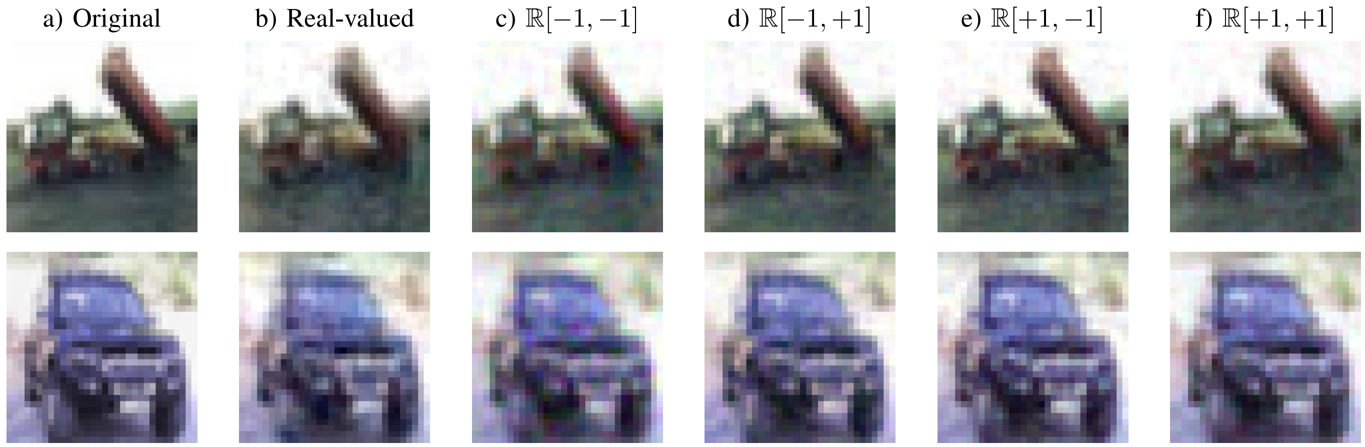

Fig. 1. Original color images and the corresponding images decoded by the real-valued and Cayley-Dickson ELM auto-encoders. First row: training image. Second row: testing image.

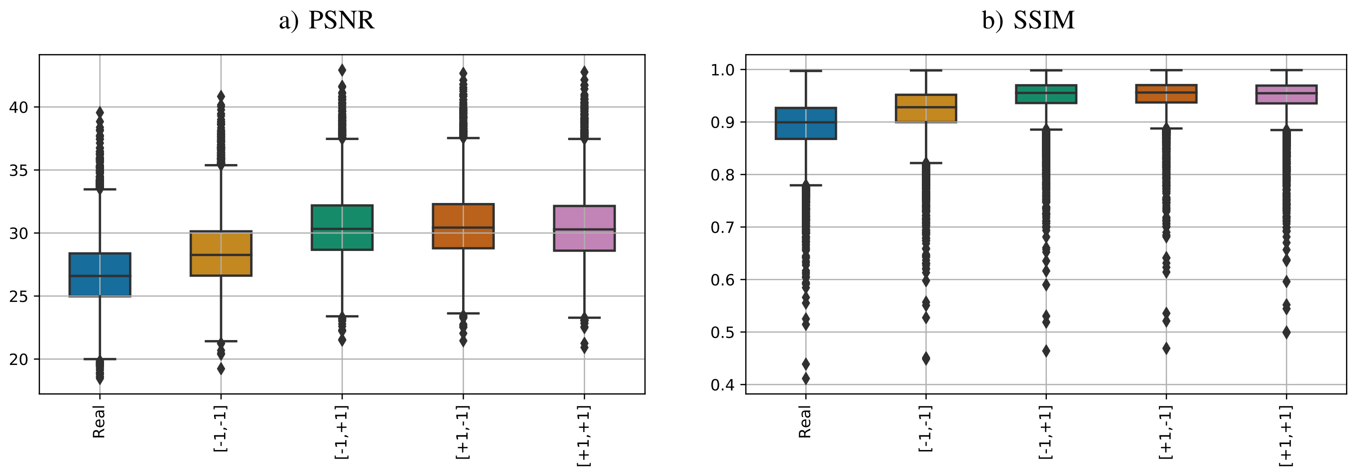

Fig. 2. PSNR and SSIM rates produced by the real-valued and Cayley-Dickson ELM auto-encoders in the test set.

framework to evaluate matrix products on Cayley-Dickson algebras as simpler real-valued matrix products. We extended the ELM concept to Cayley-Dickson algebras, explicitly showing the equivalence of the training step to a Cayley-Dickson least squares problem (30) that can be solved using (31). We then proceeded to evaluate four ELMs defined on four-dimensional Cayley-Dickson algebras as well as a real-valued ELM on a color image auto-encoding task on the CIFAR dataset.

Our preliminary experiments showed clearly that hypercomplex-valued networks vastly outperform their real-valued counterpart using a similar number of trainable parameters. This stems from the compact representation of information regarding the same object, a core characteristic of hypercomplex algebras (see Definition 1). Furthermore, the best results have been obtained from an unusual Cayley-Dickson algebra, namely the algebra $\mathbb{R}[+1,-1]$. The reason for this fact is still unclear and the results reported in this paper may instigate further research on Cayley-Dickson neural networks for which at least one generator $\gamma_1,\ldots,\gamma_k$ is positive.

REFERENCES

[1] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[2] J. Kuiper, *Quaternions and Rotation Sequences: A primer with applications to robotics, aerospace and virtual reality*. Princeton University Press, 1999.

[3] J. C. Baez, "The octonions," *Bulletin of the American Mathematical Society*, vol. 39, pp. 145–205, 2002.

[4] K. Morita, "Quasi-associativity and Cayley–Dickson algebras," *Progress of Theoretical and Experimental Physics*, vol. 2014, no. 1, 01 2014.

[5] R. D. Schafer, "On the algebras formed by the cayley-dickson process," *American Journal of Mathematics*, vol. 76, no. 2, pp. 435–446, 1954.

[6] R. B. Brown, "On generalized Cayley-Dickson algebras," *Pacific Journal of Mathematics*, vol. 20, no. 3, pp. 415–422, 1967.

[7] I. N. Aizenberg, *Complex-Valued Neural Networks with Multi-Valued Neurons*, ser. Studies in Computational Intelligence. Springer, 2011, vol. 353.

[8] A. Hirose, *Complex-Valued Neural Networks*, 2nd ed., ser. Studies in Computational Intelligence. Heidelberg, Germany: Springer, 2012.

[9] S. Buchholz and G. Sommer, "Hyperbolic multilayer perceptron," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, 2000, pp. 129–133.

[10] T. Nitta and S. Buchholz, "On the decision boundaries of hyperbolic neurons," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 2974–2980.

[11] T. Nitta and Y. Kuroe, "Hyperbolic gradient operator and hyperbolic back-propagation learning algorithms," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1689–1702, 2018.

[12] Y. Xia, M. Xiang, Z. Li, and D. P. Mandic, "Echo state networks for multidimensional data: Exploiting noncircularity and widely linear models," in *Adaptive Learning Methods for Nonlinear System Modeling*, D. Comminiello and J. C. Príncipe, Eds. Butterworth-Heinemann, 2018, pp. 267 – 288.

[13] T. Parcollet, M. Morchid, and G. Linarès, "A survey of quaternion neural networks," *Artificial Intelligence Review*, 2019.

[14] T. Minemoto, T. Isokawa, H. Nishimura, and N. Matsui, "Quaternionic multistate Hopfield neural network with extended projection rule," *Artificial Life and Robotics*, vol. 21, no. 1, pp. 106–111, Mar 2016.

[15] Y. Xia, C. Jahanchahi, and D. Mandic, "Quaternion-Valued Echo State Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 663–673, 2015.

[16] D. Xu, Y. Xia, and D. P. Mandic, "Optimization in Quaternion Dynamic Systems: Gradient, Hessian, and Learning Algorithms," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 249–261, Feb 2016.

[17] F. Z. Castro and M. E. Valle, "Continuous-Valued Quaternionic Hopfield Neural Network for Image Retrieval: A Color Space Study," in *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, Oct 2017, pp. 186–191.

[18] B. Chen, X. Qi, X. Sun, and Y.-Q. Shi, "Quaternion pseudo-Zernike moments combining both of RGB information and depth information for color image splicing detection," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 283–290, 2017.

[19] J. P. Papa, G. H. Rosa, D. R. Pereira, and X.-S. Yang, "Quaternion-based Deep Belief Networks fine-tuning," *Applied Soft Computing*, vol. 60, pp. 328–335, 2017.

[20] L. Xiaodong, L. Aijun, Y. Changjun, and S. Fulin, "Widely Linear Quaternion Unscented Kalman Filter for Quaternion-Valued Feedforward Neural Network," *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1418–1422, Sept 2017.

[21] K. Kinugawa, F. Shang, N. Usami, and A. Hirose, "Isotropization of Quaternion-Neural-Network-Based PolSAR Adaptive Land Classification in Poincare-Sphere Parameter Space," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 8, pp. 1234–1238, Aug 2018.

[22] I. Aizenberg and A. Gonzalez, "Image Recognition using MLMVN and Frequency Domain Features," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–8.

[23] G. Wang and R. Xue, "Quaternion filtering based on quaternion involutions and its application in signal processing," *IEEE Access*, vol. 7, pp. 149 068–149 079, 2019.

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

[25] F. Shang and A. Hirose, "Quaternion Neural-Network-Based PolSAR Land Classification in Poincare-Sphere-Parameter Space," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 5693–5703, 2014.

[26] S. Talebi, S. Kanna, and D. Mandic, "Real-time estimation of quaternion impropriety," in *IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, Jul. 2015, pp. 557–561.

[27] C.-A. Popa, "Scaled Conjugate Gradient Learning for Quaternion-Valued Neural Networks," in *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part III*, A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, and D. Liu, Eds. Cham: Springer International Publishing, 2016, pp. 243–252.

[28] T. Minemoto, T. Isokawa, H. Nishimura, and N. Matsui, "Feed forward neural network with random quaternionic neurons," *Signal Processing*, vol. 136, pp. 59–68, 2017.

[29] A. B. Greenblatt and S. S. Agaian, "Introducing quaternion multi-valued neural networks with numerical examples," *Information Sciences*, vol. 423, pp. 326–342, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025516316383

[30] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew *et al.*, "Extreme learning machine: a new learning scheme of feedforward neural networks," *Neural networks*, vol. 2, pp. 985–990, 2004.

[31] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[32] G.-B. Huang, D. Wang, and Y. Lan, "Extreme learning machines: a survey." *Int. J. Machine Learning & Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[33] Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2483–2490, 2011.

[34] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, 2005.

[35] H. Lv and H. Zhang, "Quaternion extreme learning machine," in *Proceedings of ELM-2016*. Springer, 2018, pp. 27–36.

[36] C. Culbert, "Cayley-dickson algebras and loops," *J. Gen. Lie Theory Appl*, vol. 1, no. 1, pp. 1–17, 2007.

[37] A. A. Albert, "Quadratic forms permitting composition," *Annals of Mathematics*, vol. 43, no. 1, pp. 161–177, 1942.

[38] A. Shenitzer, I. Kantor, and A. Solodovnikov, *Hypercomplex Numbers: An Elementary Introduction to Algebras*. Springer New York, 1989.

[39] F. Z. de Castro and M. E. Valle, "A broad class of discrete-time hypercomplex-valued hopfield neural networks," *Neural Networks*, vol. 122, pp. 54 – 67, 2020.

[40] J. Cockle, "Iii. on a new imaginary in algebra," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 34, no. 226, pp. 37–47, 1849.

[41] A. Balankin, J. Bory-Reyes, M. Luna-Elizarrarás, and M. Shapiro, "Cantor-type sets in hyperbolic numbers," *Fractals*, vol. 24, no. 04, p. 1650051, 2016.

[42] D. Rochon and M. Shapiro, "On algebraic properties of bicomplex and hyperbolic numbers," *Anal. Univ. Oradea, fasc. math*, vol. 11, no. 71, p. 110, 2004.

[43] F. Catoni, R. Cannata, E. Nichelatti, and P. Zampetti, "Commutative hypercomplex numbers and functions of hypercomplex variable: a matrix study," *Advances in Applied Clifford Algebras*, vol. 15, no. 2, pp. 183–212, 2005.

[44] G. Golub and C. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: John Hopkins University Press, 1996.

[45] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.

[46] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[47] J. Tang, C. Deng, and G. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, April 2016.

[48] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 708–717.

[49] A. Géron, *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Incorporated, 2017.

[50] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, Jan. 2009.