IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 21, 2019, accepted September 5, 2019, date of publication September 11, 2019,
date of current version September 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940217

An Extended Multilayer Perceptron Model Using Reduced Geometric Algebra

YANPING LI[1] **AND WENMING CAO**[2]
[1]College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
[2]College of Information Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding author: Wenming Cao (wmcao@szu.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant U1701265 and Grant 61771322.

ABSTRACT An extended model of multilayer perceptron (MLP) based on reduced geometric algebra (RGA), namely RGA-MLP, is proposed for multi-dimensional signal processing. The RGA-MLP model treats multi-dimensional signals as multivectors in RGA space and all neuronal parameters such as inputs, connection weights, activation function and outputs, and also operators are encoded by RGA. The RGA-based back propagation (BP) algorithm is also provided. Thanks to the commutative property of RGA, multi-dimensional signals can be processed in a holistic manner which avoids losing relationship of multiple dimensions. The experiments demonstrate that the RGA-MLP model outperforms the traditional real-valued MLP model and quaternion based MLP model (QMLP) with faster convergence rate, higher classification accuracy and Lower computational complexity.

INDEX TERMS Reduced geometric algebra (RGA), multilayer perceptron (MLP), multivectors, multi-dimensional signals.

I. INTRODUCTION

Multilayer perceptron, known as multi-layer feed-forward neural network, is the most well-known artificial neural network (ANN) model. On account of its good nonlinear mapping ability, it has been widely applied in variety of applications like pattern recognition, image processing, fusion approximation, optimization computation, and so on [1]–[4]. MLP models achieve mappings from input space to output space, and implement its training utilizing mainly back propagation (BP) algorithm to accommodate the connection weights of the network, and are commonly applied in supervised learning tasks [5], [6]. Considerable researches have emerged towards both the optimization and extended applications of MLP models. Rossi *et al.* [7] proposed an extension of the multilayer perceptron (MLP) that works on functional inputs, and then they subsequently proposed to rely on a representation of input and weight functions by projection on a truncated base [8]. Zhang *et al.* [9] extended MLP to a new MLP-like neural network and has demonstrated to require fewer neural nodes and adjustable parameters in the same situation. Gas [10] gave an extension of a self-organizing map called self-organizing multilayer perceptron to achieve quantization of spaces of functions. Grippo *et al.* [11] considered the learning problem of multilayer perceptrons formulated as the problem of minimizing a smooth error function.

Multi-dimensional signal processing is an important issue for artificial neural networks. there does not exist efficient models for multi-dimensional signal processing.For traditional real-valued MLP models [12], [13], a single neuron can take only one real value as its input, thus several neurons are used for handling multi-dimensional signals in a network, which is sometimes unnatural in practical applications to engineering problem, such as image processing. As for color image, it is processed in an independent manner, in which the color image has been represented just as three gray-scale images to apply the real-valued MLP model for three times, and, the correlation of the three color channels has been lost inevitably.

Then, the complex-valued neural networks have been extensively investigated [14], [15], which have been well established in signal and image processing. Yang *et al.* [16] analyzed the sensitivity of a split-complex valued multilayer perceptron (split-CMLP) due to the errors of the inputs and the connection weights between neurons, and showed that the sensitivity is affected by the number of the layers and the number of the neurons adopted in each layer. Due to a growing number of studies concerning the use of quaternions in neural networks, multilayer perceptron models have been extended to quaternion domain [17]–[19]. The use

The associate editor coordinating the review of this manuscript and approving it for publication was Guitao Cao.

of quaternion in MLP models yielded to a wide range of applications, such as color image compression [20], signal classification [21], etc. quaternion-valued MLP models have shown the efficiency in dealing with color images. However, the quaternion-based models are known to suffer from high computational complexity and large data redundancy due to non-commutative multiplication.

Geometric algebra (GA) provides a powerful computing framework in signal and image processing and it gives a formidable way for multi-dimensional signals [22]–[30]. Buchholz and Sommer [31] proposed the clifford neurons and clifford multilayer perceptrons in GA space for different geometric entities processing. To address the problem of high computational complexity owing to non-commutative multiplication, Shen *et al.* [32] presented a novel theory of reduced geometric algebra (RGA) with commutative multiplication rules and a novel vector-valued sparse representation model for color image using RGA.

Inspired by the recent progress of GA based models in various fields of multi-dimensional signal processing, and the advantages of RGA theory, we present an extended multilayer perceptron model using reduced geometric algebra, which treats multi-dimensional signals as multivectors in RGA space, and simplifies the computation of the networks. All elements of proposed RGA-MLP model including inputs, outputs, activation function and operators are extended to the RGA domain with commutative multiplication rules. We also present a RGA-version of back propagation (BP) algorithm to train the network. Thus, the proposed RGA-MLP model is capable of achieving the state-of-art performance with lower computational complexity for multi-dimensional signal processing.

The rest of the paper is organized as follows. Section II reviews the related work, including the basics of multilayer perceptron (MLP) and geometric algebra (GA). Section III describes the basics of RGA. The structure of RGA-MLP model and the RGA-version of back propagation algorithm is introduced in Section IV. In Section V, the classification experiments are implemented to validate the superiority of the model. Section VI concludes the paper.

II. RELATED WORK

A. BASICS OF GEOMETRIC ALGEBRA

Geometric Algebra (GA) is introduced by William K. Clifford, also known as Clifford Algebra, which gives geometric insight and effective representation for multi-dimensional signals. Particularly, GA is regarded as a powerful framework for different geometric entities, offering a potential tool to solve many tasks connected to information science [33]–[35]. Technically speaking, GA subsume, for example, the real numbers, the complex numbers and the quaternions.

Suppose G_n is a 2^n-dimensional vector space with an orthogonal basis,

$$\left\{1, \left\{e_i\right\}, \left\{e_i e_j\right\}, \cdots, \left\{e_1 e_2 \cdots e_n\right\}\right\} \tag{1}$$

An arbitrary element of the geometric algebra is given as

$$x = x_0 + \sum_{i=1}^{n} x_i e_i \tag{2}$$

where $x_0, x_1 \cdots x_n \in \mathbb{R}$.

In general, the geometric product is non-commutative with two basis vectors e_i and e_j, it is denoted

$$\begin{cases} e_i^2 = 1, & i = 1, \ldots, p \\ e_i^2 = -1, & i = p+1, \ldots, p+q \\ e_i e_j = -e_j e_i, & i \neq j \end{cases} \tag{3}$$

where G_n can be denoted as $G_{p,q}$, and $n = p + q$.

The geometric product of geometric algebra is represented as follows.

$$xy = x \cdot y + x \wedge y \tag{4}$$

where x, y represent two vectors, $x \cdot y$ and $x \wedge y$ represent inner product and outer product, respectively.

And properties of the geometric product are given as follows

$$\begin{cases} x \cdot y = \dfrac{1}{2}(xy + yx) \\ x \wedge y = \dfrac{1}{2}(xy - yx) \end{cases} \tag{5}$$

It is clear to see that the mathematical relationship of geometric product between the two vectors can be converted calculate two parts, one is a scalar part $x \cdot y$, the other is a bivector part $x \wedge y$, which is neither fully anti-symmetric, nor fully symmetric. Multivectors are viewed as the the basic elements in GA space, extending vectors to higher dimensions.

Defining any multivector $H \in (G_n)^{M \times N}$ in GA space,

$$H = E_0(H) + \sum_{1 \leqslant i \leqslant n} H_i(H)\, e_i + \sum_{1 \leqslant i \leqslant j \leqslant n} E_{ij}(H)\, e_{ij} + \cdots + E_{1\cdots n}(H)\, e_{1\cdots n}, H(H) \in \mathbb{R}^{M \times N} \tag{6}$$

B. MULTILAYER PERCEPTRON MODEL

Multilayer Perceptron (MLP) is an efficient feed-forward neural network, constituting one of the most common and popular classes of neural networks for image processing and pattern recognition [36]–[38]. It consists of several subsequent layers which is of perceptron-type, including an input layer that simply obtains the external inputs, a set of hidden layers and one output layer. This model of neural network is known as a supervised network, which requires a desired output for learning. It is worth noting that MLP is applied to create a model that correctly maps the input to the output with historical data, therefore when the desired output is unknown, the model can be able to produce the output successfully. MLP is trained using back-propagation learning algorithm, which serves to minimize the squared error between the network outputs and the desired ones. Then, this is used to adjust weights error, back-propagating minimum squared error to the neural network.

Let x_l be the input signals to MLP, the output value from the j th hidden neuron is described as

$$y_{lj} = f\left(\sum_{i=1}^{n} x_{li} w_{ij}\right) \tag{7}$$

where f represents the activation function and is the connection weight from the i th input neuron to the j th hidden neuron.

Then, the final output value from the output neuron is computed as

$$y^{out} = f\left(\sum_{j=1}^{k} y_{lj} w_j\right) \tag{8}$$

where k represents the number of hidden neurons and w_j represents the connection weight from the j th hidden neuron to the output neuron.

The real-valued MLP is adaptable for scalar data processing especially. The complex-valued MLP can treat two-dimensional signal elements as a single entity. And the QMLP can handle three-channel signals in a whole manner, which preservers the interrelationship information. However, QMLP has a series of computational burden and large data redundancy, because of non-commutativity of the quaternion multiplication. The QMLP has been recently explored to naturally represent high-dimensional information, such as color and three-dimensional geometric signals, by a quaternionic neuron, rather than complex-valued or real-valued neurons.

III. REDUCED GEOMETRIC ALGEBRA (RGA)

A. THE BASICS OF RGA

Let $\mathbb{L}_n$ is n-dimensional reduced geometric algebra (RGA) space, it denoted

$$\mathbb{L}_n = \left\{\mathbb{L}_{n_0}, \mathbb{L}_{n_1}\right\} \tag{9}$$

where $\mathbb{L}_{n_0}$ and $\mathbb{L}_{n_1}$ are regards as two parallel RGA subspace which consist of RGA space $\mathbb{L}_n$, $n_0 = n_1 = \frac{n}{2}$. In detail, $\mathbb{L}_{n_0}$ is consisted with an orthogonal basis $\{\gamma_1, \gamma_2, \cdots, \gamma_{n_0-1}\}$, $\mathbb{L}_{n_1}$ is consisted with an orthogonal basis $\left\{\gamma_{1+n_0}, \gamma_{2+n_0}, \cdots, \gamma_{n_1+n_0-1}\right\}$.

The basis elements γ_i, γ_j in $\mathbb{L}_{n_0}$ and $\mathbb{L}_{n_1}$ are both obeying the following rules:

$$\gamma_i \gamma_j = \gamma_j \gamma_i, \quad i \neq j \tag{10}$$

Moreover, we define:

$$\gamma_i^2 = \begin{cases} -1, & i \neq 0 \\ 1, & i = 0 \end{cases} \tag{11}$$

Specially, there are $\alpha = e_{2n-1}$ in $\mathbb{L}_n$, which satisfy the following rules:

$$\begin{cases} \gamma_i \alpha = \alpha \gamma_i = \gamma_{i+n_0}, \gamma_i \in L_{n_0} \\ \alpha^2 = 1 \end{cases} \tag{12}$$

Take $n = 4$, for instance, $\alpha = e_2, e_1\alpha = \alpha e_1 = e_3$. Obviously, the multiplication of γ_i is commutative according to (10) and (11). Specifically, RGA is denoted as $\mathbb{L}_n$ and can be seen as the space that is generated by the collection of $\left\{\begin{array}{c}\left\{\gamma_1, \gamma_2, \cdots, \gamma_{n_0}\right\}, \left\{\gamma_{1+n_0}, \gamma_{2+n_0}, \cdots, \gamma_{n_1+n_0-1}\right\}, \\ \left\{\gamma_{ij} = \gamma_i \gamma_j, 1 \leqslant i \neq j \leqslant n\right\}\end{array}\right\}$.

The two special numbers, β_1 and β_2, are used to construct the RGA, which contains the following rules:

$$\begin{cases} \beta_1 = \dfrac{1-\alpha}{2} \\ \beta_1 = \dfrac{1-\alpha}{2} \\ \beta_1 \beta_2 = 0 \\ \beta_1^2 = \beta_1^3 \cdots = \beta_1^n = \beta_1 \\ \beta_2^2 = \beta_2^3 \cdots = \beta_2^n = \beta_2 \end{cases} \tag{13}$$

If $k \in \mathbb{L}_n$, then, a can be described as

$$k = k^{(1)} \beta_1 + k^{(2)} \beta_2 \tag{14}$$

and where

$$\begin{cases} k^{(1)} = k_0 + k_{n_0} + \sum_{i=1}^{n_0-1} [(k_i + k_{i+n_0})\gamma_i] \\ k^{(2)} = k_0 - k_{n_0} + \sum_{i=1}^{n_0-1} [(k_i - k_{i+n_0})\gamma_i] \end{cases} \tag{15}$$

Therefore, the element k in $\mathbb{L}_4$ is also defined as the following form:

$$k = k^{(1)} \beta_1 + k^{(2)} \beta_2 \tag{16}$$

Thus,

$$\begin{cases} \beta_1 = \dfrac{1+\alpha}{2} \\ \beta_2 = \dfrac{1-\alpha}{2} \end{cases} \tag{17}$$

and where

$$\begin{cases} k^{(1)} = k^{(0)} + k^{(2)}\gamma_2 + \gamma_1 \left(k^{(1)} + k^{(3)}\right) \\ k^{(2)} = k^{(0)} - k^{(2)}\gamma_2 + \gamma_1 \left(k^{(1)} - k^{(3)}\right) \end{cases} \tag{18}$$

B. THE PROPERTIES OF REDUCED GEOMETRIC ALGEBRA

The operations of addition and subtraction in $\mathbb{L}_n$ space are almost the same as $\mathbb{L}_4$ in GA space, here we just deduce the multiplication operation in $\mathbb{L}_n$ domain. $\forall v, z \in \mathbb{L}_4$, suppose $v = v^{(1)}\beta_1 + v^{(2)}\beta_2$ and $z = z^{(1)}\beta_1 + z^{(2)}\beta_2$, then the multiplication in $\mathbb{L}_4$ are defined as follows:

$$\begin{aligned} vz &= (v^1\beta_1 + v^2\beta_2)(z^1\beta_1 + z^2\beta_2) \\ &= v^1 z^1 \beta_1^2 + v^2 z^2 \beta_2^2 \end{aligned} \tag{19}$$

The theoretical results about RGA have been obtained, it is clear that computational complexity is reduced based on RGA, multiplication in $\mathbb{L}_n$ space is divided into two multiplication in subspace $\mathbb{L}_{n_0}$ and $\mathbb{L}_{n_1}$. For instance, in $\mathbb{L}_4$ space, computational complexity is given as table 1, the computational complexity of RGA is nearly half of GA. However, in practice, the computational complexity of the proposed

TABLE 1. Computational complexity in $\mathbb{L}_4$ space.

Models	$\times$ and $\div$	$+$ and $-$	Sum
Quaternion	16	12	28
RGA	8	4	12

RGA-MLP model can be attributed to the parameter estimation of β_1 and β_2, which is over half than GA.

We then define the norm of the elements in $\mathbb{L}_4$ as

$$\begin{aligned}\|k\| &= a + b\gamma_1 + c\gamma_2 + d\gamma_3 \\ &= |a + b + c + d| = \sqrt{a^2 + b^2 + c^2 + d^2} \end{aligned} \tag{20}$$

The conjugate of k is defined as

$$k^* = a' + b'\gamma_1 + c'\gamma_2 + d'\gamma_{12} \tag{21}$$

Then

$$\begin{aligned} kk^* &= \left(a + b\gamma_1 + c\gamma_2 + d'\gamma_{12}\right)\left(a' + b'\gamma_1 + c'\gamma_2 + d'\gamma_{12}\right) \\ &= \left(aa' + bb' + cc' + dd'\right) + \left(ab' + ab + cd' + c'd\right)\gamma_1 \\ &\quad + \left(ac' - bd' + ac - bd'\right)\gamma_2 + \left(ad' + bc' + b'c + ad'\right)\gamma_{12} \\ &= \|k\|^2 \end{aligned} \tag{22}$$

According to (2), the following equations are obtained

$$\begin{cases} ab' + ab + cd' + c'd = 0 \\ ac' - bd' + ac - bd' = 0 \\ ad' + bc' + b'c + ad' = 0 \\ aa' + bb' + cc' + dd' = \|k\|^2 \end{cases} \tag{23}$$

When solving the equations in (22), the values of the individual components in (21) are correspondingly yielded, but not all the elements of $\mathbb{L}_4$ are conjugate.

Thus, the inverse of element k in $\mathbb{L}_4$ can be defined as

$$k^{-1} = \frac{k^*}{\|k\|^2} \tag{24}$$

In RGA, multivectors, which are the extension of vectors to higher dimensions, are the basic units. Each multivector $K \in \mathbb{L}_4$ is so described by

$$K = K^0 + K^1\gamma_1 + K^2\gamma_2 + K^3\gamma_{12} \tag{25}$$

where $K^0, K^1, K^2, K^3 \in \mathbb{R}$.

IV. RGA-BASED MULTILAYER PERCEPTRON MODEL

A. RGA NEURON MODEL

In this section, we utilize RGA to obtain an extension neuron model for multi-dimensional signal processing, in which all the operators can be extended to RGA domain. The basic structure of the proposed RGA neuron model is shown in FIGURE 1.

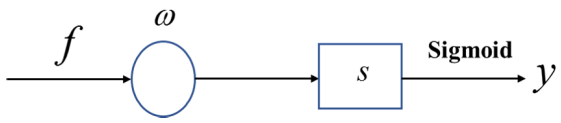

FIGURE 1. The structure of RGA neuron model.

For input $f \in \mathbb{L}_4$, the output $y \in \mathbb{L}_4$ can be formulated as follows

$$\begin{aligned} s &= \frac{f\omega}{\|\omega\|} - \theta \\ &= \frac{\left(f^{(1)}\gamma_1 + f^{(2)}\gamma_2 + f^{(3)}\gamma_{12}\right)\left(\omega^{(1)}\gamma_1 + \omega^{(2)}\gamma_2 + \omega^{(3)}\gamma_{12}\right)}{\sqrt{\left(\omega^{(1)}\right)^2 + \left(\omega^{(2)}\right)^2 + \left(\omega^{(3)}\right)^2}} \\ &\quad - \theta \\ &= \frac{\begin{bmatrix} \left(f^{(1)}\omega^{(3)} + f^{(2)}\omega^2 + f^{(3)}\omega^{(1)}\right)\gamma_1 \\ \left(f^{(1)}\omega^{(1)} + f^{(2)}\omega^3 + f^{(3)}\omega^{(2)}\right)\gamma_2 \\ \left(f^{(1)}\omega^{(2)} + f^{(2)}\omega^1 + f^{(3)}\omega^{(3)}\right)\gamma_{12} \end{bmatrix}}{\sqrt{\left(\omega^{(1)}\right)^2 + \left(\omega^{(2)}\right)^2 + \left(\omega^{(3)}\right)^2}} - \theta \end{aligned} \tag{26}$$

and

$$y = g(s) \tag{27}$$

where $\omega = \omega^{(1)}\gamma_1 + \omega^{(2)}\gamma_2 + \omega^{(3)}\gamma_{12} \in \mathbb{L}_4$ is the connection weight between neurons in the two layers, θ denotes the bias of the neuron and is also represented in $\mathbb{L}_4$. The function $g(\cdot)$ denotes a nonlinear activation function for neurons, here we adopt the sigmoid function and give its RGA version

$$g(s) = g(s^{(1)})\gamma_1 + g(s^{(2)})\gamma_2 + g(s^{(3)})\gamma_{12} \tag{28}$$

where $s = s^{(1)}\gamma_1 + s^{(2)}\gamma_2 + s^{(3)}\gamma_{12} \in \mathbb{L}_4$, $s^{(1)}, s^{(2)}, s^{(3)} \in \mathbb{R}$ and $g(u) = \frac{1}{1+e^{-u}}$, $u \in \mathbb{R}$.

For instance, the γ_2 part of the sigmoid activation function output is shown in FIGURE 2 while γ_1 and γ_2 components of input are given as the axis.

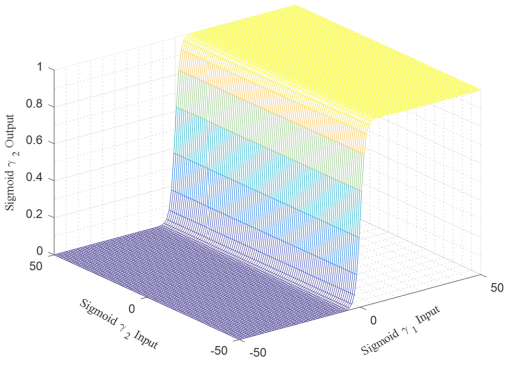

FIGURE 2. RGA sigmoid function $-\gamma_2$ part output.

As seen in (14), we can conclude that the input three-dimensional color image pixel $f \in \mathbb{L}_4$ is fully fused on different components of in $\mathbb{L}_4$ through multiplication rules RGA.

As always, neurons are the atoms of neuron network, multi-dimensional input and output are interpreted as tuples by RGA neuron with a manner of a left-side (right-side) weight association, which is generally inferior to spinor neuron in terms of two-sided weight association. However, spinor neuron is only meaningful for non-commutative algebras leading to more complicated computational complexity than that of the single-side one. It is fortunate that RGA is commutative to exhibit the same great performance as spinor neuron, while the network is simplified with lower computational complexity.

B. THE STRUCTURE OF RGA-MLP MODEL

Traditional MLP models for multi-dimensional signals treat each dimensional signal as a real number and process it independently leading to some unsatisfying results. In this section we present an extension model of traditional multi-layer perceptron (MLP) from real number to RGA domain for multi-dimensional signal processing, and the RGA neurons as defined above constitute the basis of the proposed RGA-MLP model. FIGURE 3 describes the structure of RGA-MLP model with one hidden layer.

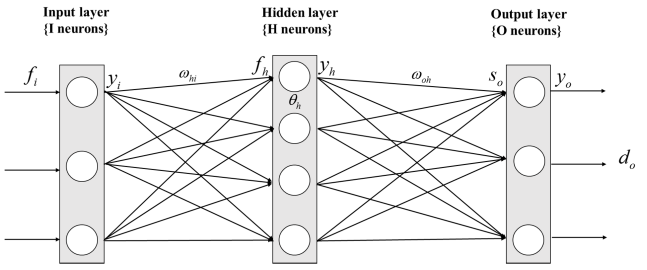

FIGURE 3. The structure of RGA-MLP model with one hidden layer.

The RGA-MLP model consists of three layers, that is input layer, hidden layer and output layer, and the numbers of neurons are set to I, H and O, respectively. Specifically, $f_i \in, i = 1, 2, \cdots, I$ and $y_i \in \mathbb{L}_4, i = 1, 2, \cdots, I$ are the input and output of the i-th neuron in the input layer respectively. Similarly, $f_h \in \mathbb{L}_4$, $y_h \in \mathbb{L}_4$, $h = 1, 2, \cdots, H$ and $f_o \in \mathbb{L}_4$, $y_o \in \mathbb{L}_4$, $o = 1, 2, \cdots, O$ are the input and output of the h-th and o-th neuron in the hidden and output layers respectively. Moreover, the connection weights between the i-th and h-th neurons in input and hidden layers, between the h-th and o-th neurons in hidden and output layers, are denoted as $\omega_{hi} \in \mathbb{L}_4$ and $\omega_{oh} \in \mathbb{L}_4$. Significantly, the desired output for the o-th neuron in the output layer is $d_o \in \mathbb{L}_4$.

Then, refer to (14) and (15), it can be clearly obtained as

$$f_h = g(\sum_{i=1}^{I} y_i \omega_{hi} + \theta_h) \tag{29}$$

C. LEARNING ALGORITHM

In order to achieve the desired output $d_o \in \mathbb{L}_4$, the connection weights are supposed to be modified by means of the so-called learning algorithms, here the back-propagation (BP) algorithm is extended to the RGA domain.

The RGA-MLP model with one hidden layer as provided, we define the error E between the desired output and the target training data as

$$\begin{aligned} E &= E(y_1, \cdots, y_o, \cdots, y_O) = \frac{1}{2}\sum_{o=1}^{O}(y_o - d_o)^2 \\ &= \frac{1}{2}\sum_{o=1}^{O}(\left(y^{(1)}\gamma_1 + y^{(2)}\gamma_2 + y^{(3)}\gamma_{12}\right)_o \\ &\quad - \left(d^{(1)}\gamma_1 + d^{(2)}\gamma_2 + d^{(3)}\gamma_{12}\right)_o)^2 \\ &= \frac{1}{2}\sum_{o=1}^{O}\left(\left(y_o^{(1)} - d_o^{(1)}\right)^2 + \left(y_o^{(2)} - d_o^{(2)}\right)^2 + \left(y_o^{(3)} - d_o^{3}\right)^2\right) \end{aligned} \tag{30}$$

where $y_o \in \mathbb{L}_4$ and $d_o \in \mathbb{L}_4$ are the output and the desired output of the o-th neuron in the output layer respectively.

In the drawback learning, the derivative of error function E with respect to any connection weights between two layers can be calculated as follows

$$\begin{aligned} \nabla E_{\omega_{ji}} &= \nabla E_{\left(\omega^{(1)}\gamma_1 + \omega^{(2)}\gamma_2 + \omega^{(3)}\gamma_{12}\right)_{ji}} \\ &= \frac{\partial E}{\partial(\omega^{(1)})_{ji}}\gamma_1 + \frac{\partial E}{\partial(\omega^{(2)})_{ji}}\gamma_2 + \frac{\partial E}{\partial(\omega^{(3)})_{ji}}\gamma_{12} \end{aligned} \tag{31}$$

Similarly, for the bias

$$\begin{aligned} \nabla E_{\theta_j} &= \nabla E_{\left(\theta^{(1)}\gamma_1 + \theta^{(2)}\gamma_2 + \theta^{(3)}\gamma_{12}\right)_j} \\ &= \frac{\partial E}{\partial(\theta^{(1)})_j}\gamma_1 + \frac{\partial E}{\partial(\theta^{(2)})_j}\gamma_2 + \frac{\partial E}{\partial(\theta^{(3)})_j}\gamma_{12} \end{aligned} \tag{32}$$

Then, the update of back-propagation for the RGA-MLP is defined as

$$\Delta\theta_j = -\eta\nabla E_{\theta_j} = -\eta\Psi_j \tag{33}$$

$$\Delta\omega_{ji} = -\eta\nabla E_{\omega_{ji}} = -\eta\Psi_j y_i^* \tag{34}$$

where $*$ is set as the conjugation and η is a constant denoting the learning coefficient. In the case of the update of the o-th neuron in the output layer, Ψ is given by

$$\begin{aligned} \Psi_o = &\left(g'(s)\left((y)_o - (d)_o\right)\right)^{(1)}\gamma_1 + \left(g'(s)\left((y)_o - (d)_o\right)\right)^{(2)}\gamma_2 \\ &+ \left(g'(s)\left((y)_o - (d)_o\right)\right)^{(3)}\gamma_{12} \end{aligned} \tag{35}$$

while Ψ of the h-th neuron in the hidden layer is given by

$$\begin{aligned} \Psi_h = &\left(g'(s)\left(\left(\sum_o(\omega)_{ho}\right)^*(\Psi)_o\right)\right)^{(1)}\gamma_1 \\ &+ \left(g'(s)\left(\left(\sum_o(\omega)_{ho}\right)^*(\Psi)_o\right)\right)^{(2)}\gamma_2 \\ &+ \left(g'(s)\left(\left(\sum_o(\omega)_{ho}\right)^*(\Psi)_o\right)\right)^{(3)}\gamma_{12} \end{aligned} \tag{36}$$

where

$$\begin{aligned} g'(s) &= g'(s^{(1)})\gamma_1 + g'(s^{(2)})\gamma_2 + g'(s^{(3)})\gamma_{12} \\ &= \frac{\partial g(s^{(1)})}{\partial s^{(1)}}\gamma_1 + \frac{\partial g(s^{(2)})}{\partial s^{(2)}}\gamma_2 + \frac{\partial g(s^{(3)})}{\partial s^{(3)}}\gamma_{12} \end{aligned} \tag{37}$$

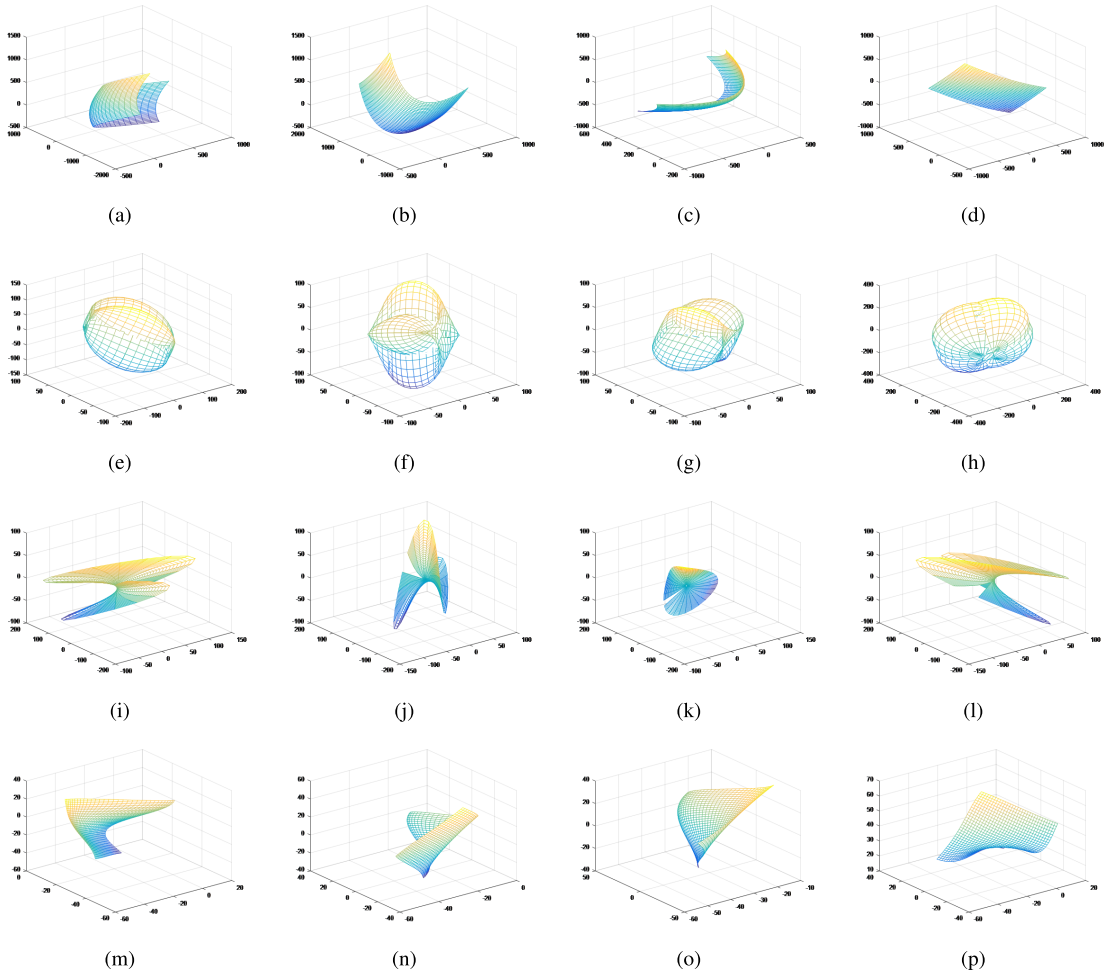

FIGURE 4. Four classes of plane curves.

V. EXPERIMENTS AND ANALYSIS

In this section, the classification performance of the proposed RGA-MLP model is evaluated on two datasets, compared with the traditional MLP model in the real and quaternion domain, quantitatively and visually.

A. DATA SETS

1) 3D GEOMETRICAL SHAPES DATA SETS

The 3*D* geometrical shapes dataset is formed, in which 16000 samples are obtained in 4 class. Patterns of the same style belong to the same class. Each class contains different patterns which are generated by various geometric transformations, such as the rotation of the first pattern with different angles and translation with different vectors around the origin. FIGURE 4 shows the example 3D geometrical shapes with 4 classes including 4 patterns for each of class, used to verify the performances of the RGA-MLP model.

2) COLOR IMAGE DATA SETS

The CIFAR-10 dataset consists of ten classes of objects with 6000 color images in each class. For each class, 5000 images are used for training and the rest 1000 images are kept for testing. And the size of all color image in this dataset is $32 * 32$, which means 1024 pixels for each color image. Each of pixel represents a color value (24 bits), describing the three components for color images including red, green and blue (denoted by 8 bits). More details of the CIFAR-10 dataset can be found at http://www.cs.toronto.edu/ kriz/cifar.html.

B. EXPERIMENTAL SETUP

The architectures of real-valued MLP, QMLP and the proposed RGA-MLP models are mainly similar and composed of an input layer, a hidden layer, and an output layer, where QMLP model treats signals as quaternion numbers, and RGA-MLP model treats signals as multivectors in RGA space.

All the experiments in this paper are performed using MATLAB under the condition of Intel(R) Core(TM) i5-6500 3.20GHz CPU and 8 GB memory, Windows 7. We choose sigmoid as the activation function, and the RGA-version has been given in FIGURE 2. The training error and the test errors are evaluated on the three networks for comparison experiments.

TABLE 2. Classification results of real-valued MLP, QMLP and RGA-MLP models on 3D geometrical shapes dataset.

MLP Models	Hidden Nodes	Error Rate (%)
Real-valued MLP	100	25.0
	150	18.7
	200	12.7
QMLP	100	21.3
	150	17.8
RGA-MLP	100	15.3
	150	12.5

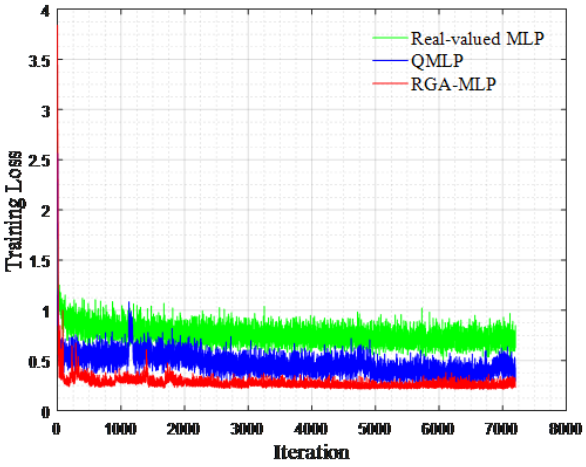

FIGURE 5. The training loss curves for a typical run of the real-valued MLP with 200 hidden nodes, the QMLP with 100 hidden nodes and the RGA-MLP with 100 hidden nodes on 3D shapes dataset.

C. 3D GEOMETRICAL SHAPES CLASSIFICATION

For 3D geometrical shapes dataset,we split 3D geometrical shapes into 4 classes, where 3000 shapes and 1000 of each class are used to training and testing, respectively. The network for 3D geometrical shapes is a three layer network, which has one hidden layer. Models are optimized with learning rate set at 0.04. The training ends at iteration 7200.

The optimal number of hidden nodes is explored as follows. Firstly, the QMLP and RGA-MLP model is trained with 100 and 150 neurons, respectively. Then, the real-valued MLP model followed with the same setting. We train the real-valued MLP model with a increasing number of hidden nodes until it reached the performance of the RGA-MLP model. Table 2 shows the test errors of the RGA-MLP model could not be reached by the real-valued MLP using 100 hidden nodes, both of them are roughly set the same number of parameters. It is important to note that more neurons within the hidden layer lead to a better performance as well, which means more features will be achieved. Altogether, the proposed RGA-MLP outperforms than the other methods in terms of classification accuracy, which can better capture the geometrical structure of 3D geometrical shapes.

The training loss curves for the three network with different number of hidden nodes are ploted in FIGURE 5 and

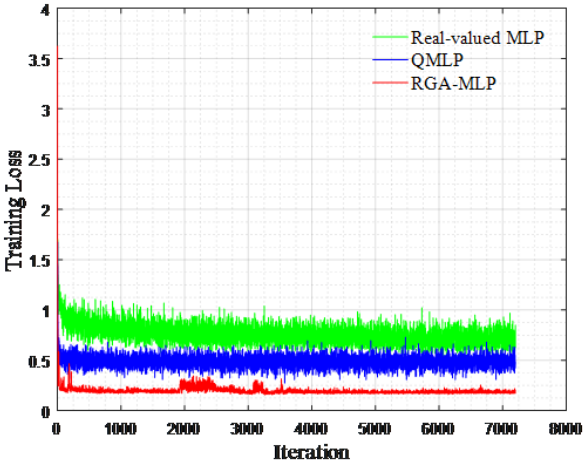

FIGURE 6. The training loss curves for a typical run of the real-valued MLP with 200 hidden nodes, the QMLP with 150 hidden nodes and the RGA-MLP with 150 hidden nodes on 3D shapes dataset.

FIGURE 6 for comparision. For 3D geometrical shapes database, the training loss curves of the RGA-MLP model converges more quickly and reaches smaller loss than that of real-valued MLP and QMLP finally. Furthermore, the training loss curves of real-valued MLP and QMLP models are very unstable and fluctuate greatly during the training process, in contrast, the training loss curves of RGA-MLP model is more stable convergence.

The results illustrate that the RGA-MLP model outperforms the traditional real-valued MLP. It can be seen that multi-dimensional signals processed by real-valued MLP tends to be single channel independently, which are not rich enough to preserves more discriminative information among multiple channels. In contrast, the proposed RGA-MLP model is capable of capturing the inter-relationship information between different channels, such as the scaling and the rotation of inputs in multi-dimensional space, this information is mort important for classification.

D. COLOR IMAGES CLASSIFICATION

In this section, the color images classification experiments with traditional real-valued MLP model, QMLP and RGA-MLP models are carried out based on the CIFAR-10 dataset. In this experiment, 2 classes of color images in CIFAR-10 are used, 6000 color images of each class for training and 2000 color images of each class for testing. The learning rate of the both models are fixed at 0.04 and the three models are trained for 3600 iterations while the batch size is set to 50. Since three color channels R,G,B are treated similarly as input, the labels with respect to three channels are set to the same,which ensures that the three channels (γ_1, γ_2, γ_{12}) are treated equally with the same priority. Therefore, the inter-relationship among the three color channels can be extracted more effectively using RGA theory.

More precisely, as shown in FIGURE 7 and FIGURE 8, the RGA-MLP model outperforms the real-valued MLP and QMLP model with the faster rate of convergence under

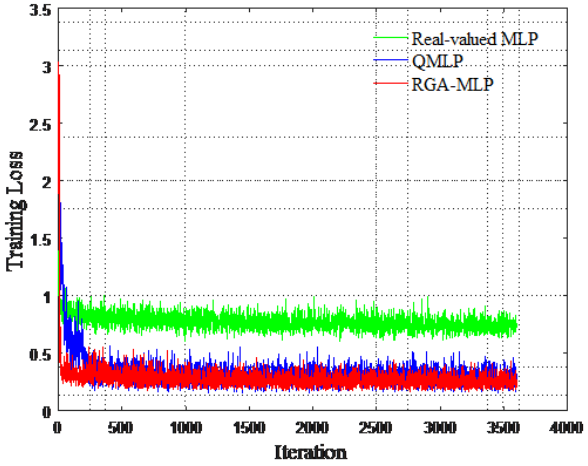

FIGURE 7. The training loss curves achieved by the real-valued MLP, QMLP and RGA-MLP models with one hidden layer on color image dataset.

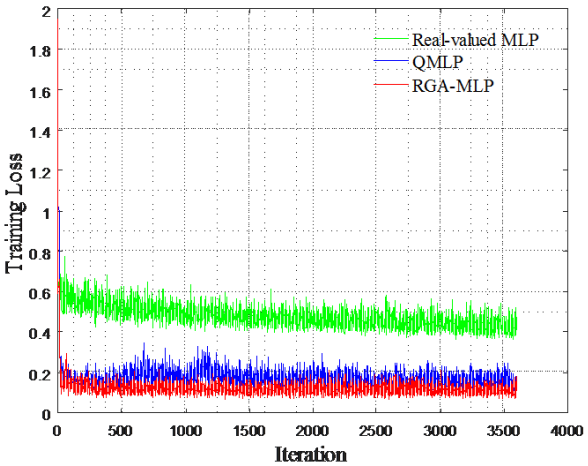

FIGURE 8. The training loss curves achieved by the real-valued MLP, QMLP and RGA-MLP models with two hidden layers on color image dataset.

smaller number of iterations, that is to say, the proposed RGA-MLP model has the formidable ability in approaching a stable value more rapidly. Moreover, it can be clearly concluded that the proposed RGA-MLP model achieves a more stable convergence trend and lower training losses while the real-valued MLP model and QMLP model fluctuate greatly.

In order to study the effects of different number of hidden layers on the performance of networks, we adopt different number of hidden layers of three networks in this experiment. FIGURE 7, FIGURE 8 and table 3 show a clear improvement in both the convergence rate and the test accuracy when increasing the number of hidden layers for each network.

For quantitatively comparison, test errors obtained by color images classification experiments performed by the real-valued MLP, QMLP and the proposed RGA-MLP models are listed in table 3. The second column denotes different topologies in hidden layers ranging from a single layer net with 100 output neurons to a net with two hidden layers each with 100 and 50 neurons, respectively. Expectedly, the proposed

TABLE 3. Classification results of real-valued MLP, QMLP and RGA-MLP models on color image dataset.

MLP Models	Hidden Nodes	Error Rate (%)
Real-valued MLP	100	26.2
	100-50	20.3
QMLP	100	20.1
	100-50	19.9
RGA-MLP	100	19.4
	100-50	19.2

TABLE 4. Computational time of QMLP and RGA-MLP models on different dataset.

Dataset	Models	Hidden Nodes	Training Time (sec)
3D Geometrical Shapes	QMLP	100	254.33
	RGA-MLP	100	166.99
	QMLP	150	401.47
	RGA-MLP	150	266.18
Color Images	QMLP	100	119.57
	RGA-MLP	100	84.45
	QMLP	100-50	149.91
	RGA-MLP	100-50	105.72

RGA-MLP exhibits lower test errors compared with the real-valued MLP and QMLP. As suggested in table 3, in the case of RGA-MLP model only with one hidden layer, the test errors of classification yielded by the proposed RGA-MLP reaches approximately 19.4% while that of real-valued MLP is only 26.2 and QMLP is 20.1%. When the number of hidden layer increases, the corresponding test accuracy of the traditional real-valued MLP go down to 20.3%, and the test errors of QMLP reduce to 19.9%, while that of the proposed RGA-MLP decrease to 19.2%.

To be more precise, the proposed model treats color images as RGA multivectors with the strong ability in capturing the relationship of color channels. Furthermore, a more simplified network and a powerful learning algorithm are presented in the proposed RGA-MLP model, thereby superior performance is achieved compared with the traditional real-valued MLP and QMLP models.

E. COMPLEXITY ANALYSIS

The computational complexity of QMLP and RGA-MLP models is evaluated in this section. The computation time for QMLP and RGA-MLP models with two different datasets are listed in table 4. The elapsed time for the RGA-MLP model is less than the elapsed time for QMLP model. Thanks to the commutative multiplication rule of RGA, the computational time of the RGA-MLP model is nearly the two-thirds of QMLP with lower computational complexity. The experiments have verified the effectiveness of RGA-MLP model, where the correlation among channels in multi-dimensional signals can be preserved well and the network

has been simplified using RGA theory. Moreover, this kind of computational workload reduction does not influence the classification results.

VI. CONCLUSION

In this paper, we have proposed an extended multilayer perceptron model for multi-dimensional signal processing and presented an error back-propagation algorithm for its learning scheme in RGA space. Taking advantage of RGA theory, the multi-dimensional signals are represented as RGA multivectors, multiple channels treated as a single unit instead of the separate components. And all elements and operators are extended into RGA domain, exploits strong ability in capturing the inherent structures and significantly acts as an efficient and simplified network with formidable training ability. The experiments demonstrate that our proposed model can achieve higher classification accuracy, faster convergence rate and lower computational complexity. Avenues for future work includes the combination of the RGA-MLP model with other different network and more excellent networks designed for multi-dimensional signal processing.

REFERENCES

[1] P. A. Castillo, M. G. Arenas, J. J. Merelo, and G. Romero, "Cooperative co-evolution of multilayer perceptrons," in *Proc. 7th Int. Work-Conf. Artif. Neural Netw.*, vol. 1. Berlin, Germany: Springer-Verlag, 2003, pp. 358–365.

[2] R. D. Dony and S. Haykin, "Neural network approaches to image compression," *Proc. IEEE*, vol. 83, no. 2, pp. 288–303, Feb. 1995.

[3] M. Gallagher and T. Downs, "Visualization of learning in multilayer perceptron networks using principal component analysis," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 33, no. 1, pp. 28–34, Feb. 2003.

[4] F.-L. Chung, S. Wang, Z. Deng, and D. Hu, "CATSMLP: Toward a robust and interpretable multilayer perceptron with sigmoid activation functions," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 36, no. 6, pp. 1319–1331, Dec. 2006.

[5] X. Hu and Q. Weng, "Estimating impervious surfaces from medium spatial resolution imagery using the self-organizing map and multi-layer perceptron neural networks," *Remote Sens. Environ.*, vol. 113, no. 10, pp. 2089–2102, 2009.

[6] G. S. V. S. Sivaram and H. Hermansky, "Sparse multilayer perceptron for phoneme recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 23–29, Jan. 2012.

[7] F. Rossi and B. Conan-Guez, "Theoretical properties of projection based multilayer perceptrons with functional inputs," *Neural Process. Lett.*, vol. 23, no. 1, pp. 55–70, 2006.

[8] F. Rossi, B. Conan-Guez, and A. El Golli, "Clustering functional data with the SOM algorithm," in *Proc. 12th Eur. Symp. Artif. Neural Netw. (ESANN)*, 2004, pp. 28–30.

[9] D. Zhang, X.-L. Bai, and K.-Y. Cai, "Extended neuro-fuzzy models of multilayer perceptrons," *Fuzzy Sets Syst.*, vol. 142, no. 2, pp. 221–242, 2004.

[10] B. Gas, "Self-organizing multilayer perceptron," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1766–1779, Nov. 2010.

[11] L. Grippo, A. Manno, and M. Sciandrone, "Decomposition techniques for multilayer perceptron training," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2146–2159, Nov. 2016.

[12] D. E. Rumelhart, B. Widrow, and M. A. Lehr, "The basic ideas in neural networks," *Commun. ACM*, vol. 37, no. 3, pp. 87–93, 1994.

[13] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, pp. 2627–2636, Aug. 1998.

[14] C. You and D. Hong, "Nonlinear blind equalization schemes using complex-valued multilayer feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1442–1455, Nov. 1998.

[15] A. Hirose, *Complex-Valued Neural Networks*. Berlin, Germany: Springer-Verlag, 2006.

[16] S.-S. Yang, C.-L. Ho, and S. Siu, "Sensitivity analysis of the split-complex valued multilayer perceptron due to the errors of the i.i.d. inputs and weights," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1280–1293, Sep. 2007.

[17] P. Arena, L. Fortuna, L. Occhipinti, and M. G. Xibilia, "Neural networks for quaternion-valued function approximation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 6, May/Jun. 1994, pp. 307–310.

[18] T. Isokawa, H. Nishimura, and N. Matsui, "Quaternionic multilayer perceptron with local analyticity," *Information*, vol. 3, no. 3, pp. 756–770, 2012.

[19] H. Kusamichi, T. Isokawa, and N. Matsui, "A new scheme for color night vision by quaternion neural network," in *Proc. 2nd Int. Conf. Auton. Robots Agents*, 2004, pp. 101–106.

[20] L. Luo, H. Feng, and L. Ding, "Color image compression based on quaternion neural network principal component analysis," in *Proc. Int. Conf. Multimedia Technol.*, Oct. 2010, pp. 1–4.

[21] S. Buchholz and N. L. Bihan, "Polarized signal classification by complex and quaternionic multi-layer perceptrons," *Int. J. Neural Syst.*, vol. 18, no. 2, pp. 75–85, Apr. 2008.

[22] E. Hitzer, T. Nitta, and Y. Kuroe, "Applications of clifford's geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 23, no. 2, pp. 377–404, 2013.

[23] W. Cao, F. Lyu, Z. He, G. Cao, and Z. He, "Multimodal medical image registration based on feature spheres in geometric algebra," *IEEE Access*, vol. 6, pp. 21164–21172, 2018.

[24] R. Wang, Z. Cao, X. Wang, W. Xue, and W. Cao, "GA-STIP: Action recognition in multi-channel videos with geometric algebra based spatio-temporal interest points," *IEEE Access*, vol. 6, pp. 56575–56586, 2018.

[25] W. Cao, J. Yuan, Z. He, Z. Zhang, and Z. He, "Fast deep neural networks with knowledge guided training and predicted regions of interests for real-time video object detection," *IEEE Access*, vol. 6, pp. 8990–8999, 2018.

[26] R. Wang, M. Shen, and W. Cao, "Multivector sparse representation for multispectral images using geometric algebra," *IEEE Access*, vol. 7, pp. 12755–12767, 2019.

[27] R. Wang, M. Shen, T. Wang, and W. Cao, "L1-norm minimization for multi-dimensional signals based on geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 29, p. 33, Apr. 2019. doi: 10.1007/s00006-019-0950-7.

[28] R. Wang, W. Zhang, Y. Shi, X. Wang, and W. Cao, "GA-ORB: A new efficient feature extraction algorithm for multispectral images based on geometric algebra," *IEEE Access*, vol. 7, pp. 71235–71244, 2019. doi: 10.1109/ACCESS.2019.2918813.

[29] W. Cao, Q. Lin, Z. He, and Z. He, "Hybrid representation learning for cross-modal retrieval," *Neurocomputing*, vol. 345, pp. 45–57, Jun. 2018. doi: 10.1016/j.neucom.2018.10.082.

[30] R. Wang, Y. He, C. Huang, X. Wang, and W. Cao, "A novel least-mean Kurtosis adaptive filtering algorithm based on geometric algebra," *IEEE Access*, vol. 7, pp. 78298–78310, 2019. doi: 10.1109/ACCESS.2019.2922343.

[31] S. Buchholz and G. Sommer, "On clifford neurons and clifford multi-layer perceptrons," *Neural Netw.*, vol. 21, no. 7, pp. 925–935, 2008.

[32] M. Shen, R. Wang, and W. Cao, "Joint sparse representation model for multi-channel image based on reduced geometric algebra," *IEEE Access*, vol. 6, pp. 24213–24223, 2018. doi: 10.1109/ACCESS.2018.2819691.

[33] W. B. Lopes, A. Al-Nuaimi, and C. G. Lopes, "Geometric-algebra lms adaptive filter and its application to rotation estimation," *IEEE Signal Process. Lett*, vol. 23, no. 6, pp. 858–862, Jun. 2016.

[34] L. Zou, J. Lasenby and Z. He, "Beamforming with distortionless co-polarisation for conformal arrays based on geometric algebra," *IET Radar, Sonar Navigat.*, vol. 5, no. 8, pp. 842–853, Oct. 2011. doi: 10.1049/iet-rsn.2011.0112.

[35] Y. Li, W. Liu, X. Li, Q. Huang, and X. Li, "GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra," *Inf. Sci.*, vol. 281, pp. 559–572, Oct. 2014.

[36] Q. Li, Q. Peng, and C. Yan, "Multiple VLAD encoding of CNNs for image classification," *Comput. Sci. Eng.*, vol. 20, no. 2, pp. 52–63, Mar./Apr. 2018.

[37] X. Zhang, J. Zou, K. He, and J. Sun, "Accelerating very deep convolutional networks for classification and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1943–1955, Oct. 2016.

[38] M. R. Silvestre, S. M. Oikawa, and F. H. T. Vieira, "A clustering based method to stipulate the number of hidden neurons of MLP neural networks: Applications in pattern recognition," *Trends Appl. Comput. Math.*, vol. 9, no. 2, pp. 351–361, 2008.