IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 17, 2017, accepted March 4, 2017, date of publication March 8, 2017, date of current version August 14, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2679185

Convergence Analysis of Caputo-Type Fractional Order Complex-Valued Neural Networks

JIAN WANG[1], GUOLING YANG[1], BINGJIE ZHANG[1], ZHANQUAN SUN[2], YUSONG LIU[1], AND JICHAO WANG[1]

[1]College of Science, China University of Petroleum, Qingdao 266580, China
[2]Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks, Jinan 250014, China

Corresponding authors: Jian Wang (wangjiannl@upc.edu.cn) and Zhanquan Sun (sunzhq@sdas.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61305075 and Grant 41406007, in part by the China Postdoctoral Science Foundation under Grant 2012M520624, in part by the Natural Science Foundation of Shandong Province under Grant ZR2013FQ004 and Grant ZR2013DM015, in part by the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20130133120014, and in part by the Fundamental Research Funds for the Central Universities under Grant 14CX05042A, Grant 15CX05053A, and Grant 15CX08011A.

ABSTRACT The complex-valued neural networks are the class of networks that solve complex problems by using complex-valued variables. The gradient descent method is one of the popular algorithms to train complex-valued neural networks. Essentially, the established networks are integer-order models. Compared with classical integer-order models, the built models in terms of fractional calculus possess significant advantages on both memory storage and hereditary characteristics. As one of commonly used fractional-order derivatives, Caputo derivative is more applicable in practical problems due to its simple requirements on initial condition. In this paper, we adopt this specific fractional-order derivative to train split-complex neural networks. As a result, the monotonicity and weak convergence of the presented model are rigorously proved. In addition, numerical simulation has effectively verified its competitive performance and also illustrated the theoretical results.

INDEX TERMS Complex-valued neural networks, Caputo fractional derivative, monotonicity, convergence.

I. INTRODUCTION

Recently, fractional-order complex neural networks have been extensively studied on both theory and practical applications. Fractional calculus has already been emerged three hundred years. It is a generalization of ordinary differential and integer operations from integer order to fractional order. Since the inherent fault tolerance and parallel computational characteristics of neural networks, the established fractional operations have the advantages in describing the memory and hereditary properties of various procedures [1]. As a natural extension of real number field, complex-valued problems are widely occurred in many research fields such as electromagnetics, image processing, voice processing. There have been increasing studies on complex-valued problems by employing fractional-order neural networks.

Neural networks have attracted an increasingly interest in the past decades, due to the powerful nonlinear mapping and parallel data processing abilities of them. Several classical neural networks, multilayer perceptions, Radical Basis function networks (RBF), bidirectional associative memory (BAM) and support vector machine (SVM) have been intensively investigated. Many good results have been obtained on various researches [2]–[5]. Constrained by the data volume and the dealing strategies, these models are the networks with shallow architectures. Since 2006, a breakthrough has been made in [12], which claimed the appearance of deep learning networks. A vast of significant achievements have been reached in a variety of research fields during the past ten years. However, most of these results are confined in real domain.

With the rapid development of electronic science, the complex-value signals are frequently appeared in the engineering practice and complex neural network is more and more widely used. The complex-valued neural networks (CVNNs) are the extension of real-valued neural networks. It is embodied in: the input, output and the weights are complex. Moreover, complex-valued neural networks have advantage on classification issues and abilities in reducing the number of parameters and operations. Therefore, more and more scholars focus their researches on complex-valued neural networks [6]–[11], [22], [23]. In order to solve nonlinear separation problems, authors concentrated on complex-value BP algorithm [6]. In [7], the complex-valued backpropagation neural networks (CBP) were trained by Complex Levengberg-Marquardt algorithm and different activation functions were compared. A fully complex-valued radial basis function network was applied to solve real-valued classification problems in [8]. Moreover, the FC-RBF classifier was investigated experimentally well by a set of real-valued benchmark problems. Based on wirtinger calculus [9], authors put forward an augmented algorithm for fully complex-valued neural networks.

Fractional calculus is a branch of calculus. Nowadays, fractional derivatives have been applied in many fields. Moreover, more and more researchers are concerned about the fractional calculus. In the respects of fractional pattern recognition and adaptive signal processing, integer-order adaptive learning method is not applicable. Then the fractional steepest descent approach was presented in [24]. There have some excellent results about the dynamic analyses on fractional calculus [1], [13]–[17], [25]. For example, Delavari *et al.* [15] investigated the stability of fractional-order nonlinear systems. Applying time domain scheme to the fractional order system [17], the author studied the unified system and synchronization of the fractional order.

Due to the good memory and heredity of fractional calculus, fractional calculus is widely applied to complex-valued neural networks. Currently, most of the results focus on fractional-order CVNNs with time delays [18]–[21]. For example, under sufficient conditions, fractional-order complex-valued neural networks for the existence and uniform steadiness are solved [18]. Bao *et al.* [19] discuss the synchronization problem with fractional-order complex-valued neural networks with time delays. In [20], the problems of leakage and discrete delays on fractional-order complex-valued neural networks were considered, in which authors handled the uniqueness and global uniform stability of the equilibrium point. In order to study the dissipativity of neural networks with time delay, authors proposed a kind of fractional-order complex-valued neural networks with time delays and discussed its dissipativity and stability analysis [21].

Motivated by the above discussion, we consider the convergence analysis of fractional-order complex-valued neural networks. Different from the training algorithm in [22], we use fractional order steepest descent method to train split-complex neural networks. In particular, Caputo-type fractional-order derivative is adopted. In the aspects of theories, the monotonicity and weak convergence of the presented model are rigorously proved. In addition, numerical simulations have effectively verified its competitive performance and also illustrated the theoretical results.

The structure of this paper is as follows: Section 2 presents a description of Caputo fractional-order derivative. The algorithm descriptions with two types of algorithm are then presented in Section 3. The main results based on Caputo-type fractional gradient descent method are drawn in Section 4. In Section 5, a numerical example on XOR problem is performed. Conclusions are presented in Section 6. Finally, the appendix shows the proof process of Caputo-type fractional order complex-valued neural networks algorithm.

II. FRACTIONAL-ORDER DERIVATIVE

There are several definitions used for fractional derivatives. The three most common fractional calculus definitions are Grünwald-Letnikov (GL), Riemann-Liouville (RL), and Caputo [26], [28]–[30].

Definition 1 (GL Fractional-Order Derivative): The GL derivative with order α of function $f(t)$ is defined as

$$\begin{aligned} {}^{GL}{}_aD_t^{\alpha}f(t) &= \sum_{k=0}^{n}\frac{f^{(k)}(a)(t-a)^{-\alpha+k}}{\Gamma(-\alpha+k+1)} \\ &\quad + \frac{1}{\Gamma(n-\alpha+1)}\int_a^t (t-\tau)^{n-\alpha}f^{(n+1)}(\tau)d\tau, \end{aligned} \tag{1}$$

where ${}^{GL}{}_aD_t^{\alpha}$ is the GL fractional derivative operator, $\alpha > 0$, $n-1 < \alpha < n, n \in N$, $f(t)$ is the objective function under consideration and $[a, t]$ is the interval of $f(t)$. $\Gamma(\cdot)$ is the Gamma function, that is, $\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1}e^{-t}dt$.

Definition 2 (RL Fractional-Order Derivative): The Riemann-Liouville derivative with order α of function $f(t)$ is defined as

$$\begin{aligned} {}^{RL}{}_aD_t^{\alpha}f(t) &= \frac{d^n}{dt^n}{}_aD_t^{-(n-\alpha)}f(t) \\ &= \frac{1}{\Gamma(n-\alpha)}\frac{d^n}{dt^n}\int_a^t (t-\tau)^{n-\alpha-1}f(\tau)d\tau, \end{aligned} \tag{2}$$

where ${}^{RL}{}_aD_t^{\alpha}$ is the RL fractional derivative operator. In addition, the GL derivative may be obtained based on the definition of the RL fractional derivative.

Definition 3 (Caputo Fractional-Order Derivative): The definition of the Caputo fractional-order derivative of order α is defined as follows

$${}^{Caputo}{}_aD_t^{\alpha}f(t) = \frac{1}{\Gamma(n-\alpha)}\int_a^t (t-\tau)^{n-\alpha-1}f^{(n)}(\tau)d\tau, \tag{3}$$

where ${}^{Caputo}{}_aD_t^{\alpha}$ is the Caputo derivative operator, α is the fractional order.

Specifically, when $\alpha \in (0, 1)$, the expression for Caputo derivative is as follows

$${}^{Caputo}{}_aD_t^{\alpha}f(t) = \frac{1}{\Gamma(1-\alpha)}\int_a^t (t-\tau)^{-\alpha}f'(\tau)d\tau. \tag{4}$$

When $\alpha \in (0, 1)$ and $a = 0$, there exists an evident difference of the RL and Caputo derivatives with respect to a constant. Assume that K is a constant, then the RL derivative can be expressed as follows

$${}^{RL}{}_aD_t^{\alpha}K = \frac{K}{\Gamma(1-\alpha)}x^{-\alpha} \neq 0, \tag{5}$$

however, the corresponding Caputo derivative is equal to zero, that is,

$$ {}^{Caputo}{}_{a}D_t^{\alpha} K = 0. \tag{6}$$

Since the initial values between the fractional differential equations with the Caputo derivative and the integer differential equations are the same: this derivative has a wide range of application in physical processes and engineering problems. In this paper, we just employ the Caputo fractional-order derivative to evaluate the CBP training algorithm for fractional-order complex-valued neural networks. For convenience, we use the notion ${}_aD_t^{\alpha}$ to denote Caputo fractional derivative operator instead of ${}^{Caputo}{}_aD_t^{\alpha}$.

Remark 1: There are close contacts between these three definitions. For example, the relationship between Riemann-Liouville definition and Grünwald-Letnikov definition [27]: Suppose function $u(t)$ is defined on the interval (a, b), $u(t)$ has $m+1$ order continuous derivatives and m at least takes $[\mu] = n-1$, then the definitions between G-L fractional calculus and R-L fractional calculus are equivalent. Moreover, G-L definition is equal to Caputo definition if the kth derivative of function $u(t)$ also satisfies $u^{(k)}(a) = 0, k = 0, 1, \cdots, n-1$. Wu and Huang [27] give more clear relationships of these three different fractional derivatives.

III. ALGORITHM DESCRIPTION

Without loss of generality, we consider a three-layered CVNN consisting of p input neurons, n hidden neurons, and 1 output neuron. Suppose that training samples $\{\mathbf{z}^j, d^j\}_{j=1}^J \subset \mathbb{C}^p \times \mathbb{C}^1$. For input signals $\mathbf{z} = (z_1, z_2, \ldots, z_p)^T = \mathbf{x} + i\mathbf{y} \in \mathbb{C}^p$, where $\mathbf{x} = (x_1, x_2, \ldots, x_p)^T \in \mathbb{R}^p$, and $\mathbf{y} = (y_1, y_2, \ldots, y_p)^T \in \mathbb{R}^p$. Let $\mathbf{v}_l = \mathbf{v}_l^R + i\mathbf{v}_l^I = (v_{l1}, v_{l2}, \ldots, v_{lp})^T \in \mathbb{C}^p$ as the weight vector between the input neurons and lth hidden neuron, where $v_{lm} = v_{lm}^R + iv_{lm}^I$, and v_{lm}^R, v_{lm}^I are the real part and the imaginary part of v_{lm}, respectively, $l = 1, \ldots, n, m = 1, \ldots, p$. The weight vector between the hidden neurons and the output neuron is denoted by $\mathbf{u} = \mathbf{u}^R + i\mathbf{u}^I = (u_1, u_2, \ldots, u_n)^T \in \mathbb{C}^n$, where $u_l = u_l^R + iu_l^I$, and u_l^R, u_l^I are the real part and the imaginary part of u_l, separately, $l = 1, \ldots, n$. For simplicity, we incorporate all the weight vectors into a total weight vector

$$\mathbf{w} = \left((\mathbf{v}_1)^T, (\mathbf{v}_2)^T, \ldots, (\mathbf{v}_n)^T, \mathbf{u}^T\right)^T \in \mathbb{C}^{n(p+1)}. \tag{7}$$

For the lth node of the hidden layer, the input is

$$\begin{aligned}\theta_l &= \mathbf{v}_l \cdot \mathbf{z} = (\mathbf{v}_l^R + i\mathbf{v}_l^I)\cdot(\mathbf{x}+i\mathbf{y}) = \theta_l^R + i\theta_l^I \\ &= \sum_{m=1}^{p}\left(v_{lm}^R x_m - v_{lm}^I y_m\right) + i\sum_{m=1}^{p}\left(v_{lm}^I x_m + v_{lm}^R y_m\right) \\ &= \begin{pmatrix}\mathbf{v}_l^R \\ -\mathbf{v}_l^I\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} + i\begin{pmatrix}\mathbf{v}_l^I \\ \mathbf{v}_l^R\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix},\end{aligned} \tag{8}$$

where "$\cdot$" denotes the inner product of two vectors.

We consider the following popular real-imaginary type activation function

$$g_C(\theta) = g_R\left(\theta^R\right) + ig_I\left(\theta^I\right), \tag{9}$$

for any $\theta = \theta^R + i\theta^I \in \mathbb{C}^1$, where g_R and g_I are real activation function, respectively. For simplicity, g_R and g_I are the same in this paper. For example, the output of the lth hidden neuron is given by

$$H_l = H_l^R + iH_l^I = g\left(\theta_l^R\right) + ig\left(\theta_l^I\right). \tag{10}$$

Write:

$$\begin{aligned}\mathbf{H} &= \mathbf{H}^R + i\mathbf{H}^I, \\ \mathbf{H}^R &= \left(H_1^R, H_2^R, \ldots, H_n^R\right)^T, \\ \mathbf{H}^I &= \left(H_1^I, H_2^I, \ldots, H_n^I\right)^T.\end{aligned} \tag{11}$$

For the output layer, the input is

$$\begin{aligned}S &= \mathbf{u}\cdot\mathbf{H} = (\mathbf{u}^R + i\mathbf{u}^I)\cdot(\mathbf{H}^R + i\mathbf{H}^I) = S^R + iS^I \\ &= \sum_{l=1}^{n}\left(u_l^R H_l^R - u_l^I H_l^I\right) + i\sum_{l=1}^{n}\left(u_l^I H_l^R + u_l^R H_l^I\right) \\ &= \begin{pmatrix}\mathbf{u}^R \\ -\mathbf{u}^I\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^R\\ \mathbf{H}^I\end{pmatrix} + i\begin{pmatrix}\mathbf{u}^I \\ \mathbf{u}^R\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^R\\ \mathbf{H}^I\end{pmatrix}.\end{aligned} \tag{12}$$

Similarly, the output of the network is given by

$$O = O^R + iO^I = f\left(S^R\right) + if\left(S^I\right). \tag{13}$$

For the jth sample, the input of the hidden neuron l is denoted by $\theta_l^j = \theta_l^{j,R} + i\theta_l^{j,I}(1 \le l \le n, 1 \le j \le J)$, the output of the hidden neuron l is represented by $H_l^j = H_l^{j,R} + iH_l^{j,I}(1 \le l \le n)$, $S^j = S^{j,R} + iS^{j,I}$ stands for the input of the output neuron, and $O^j = O^{j,R} + iO^{j,I}$ represents the actual output. The squared error function of CBP can be represented as follows:

$$\begin{aligned}E(\mathbf{w}) &= \frac{1}{2}\sum_{j=1}^{J}(O^j - d^j)(O^j - d^j)^* \\ &= \frac{1}{2}\sum_{j=1}^{J}\left[(O^{j,R} - d^{j,R})^2 + (O^{j,I} - d^{j,I})^2\right] \\ &= \sum_{j=1}^{J}\left[f_{jR}(S^{j,R}) + f_{jI}(S^{j,I})\right],\end{aligned} \tag{14}$$

where "$*$" signifies complex conjugate, and

$$f_{jR}(t) = \frac{1}{2}(f(t) - d^{j,R})^2, f_{jI}(t) = \frac{1}{2}(f(t) - d^{j,I})^2, \tag{15}$$

for $t \in \mathbb{R}^1, 1 \le j \le J$.

1) CBP ALGORITHM BASED ON GRADIENT DESCENT METHOD

The kth derivatives of the error function in regard to u_l^R, u_l^I, v_{lm}^R and v_{lm}^I are, respectively, given by

$$E_{u_l^{k,R}}(\mathbf{w}) = \sum_{j=1}^{J}(f'_{jR}(S^{j,R})H_l^{j,R} + f'_{jI}(S^{j,I})H_l^{j,I}), \tag{16}$$

$$E_{u_l^{k,I}}(\mathbf{w}) = \sum_{j=1}^{J}(-f'_{jR}(S^{j,R})H_l^{j,I} + f'_{jI}(S^{j,I})H_l^{j,R}), \tag{17}$$

$$E_{v_{lm}^{k,R}}(\mathbf{w}) = \sum_{j=1}^{J}\Big[f'_{jR}(S^{j,R})(u_l^{k,R}g'(\theta_l^{j,R})x_m^j - u_l^{k,I}g'(\theta_l^{j,I})y_m^j) + f'_{jI}(S^{j,I})(u_l^{k,I}g'(\theta_l^{j,R})x_m^j + u_l^{k,R}g'(\theta_l^{j,I})y_m^j)\Big], \tag{18}$$

$$E_{v_{lm}^{k,I}}(\mathbf{w}) = \sum_{j=1}^{J}\Big[f'_{jR}(S^{j,R})(-u_l^{k,R}g'(\theta_l^{j,R})y_m^j - u_l^{k,I}g'(\theta_l^{j,I})x_m^j) + f'_{jI}(S^{j,I})(-u_l^{k,I}g'(\theta_l^{j,R})y_m^j + u_l^{k,R}g'(\theta_l^{j,I})x_m^j)\Big], \tag{19}$$

where $k = 0, 1, 2, \cdots; l = 1, 2, \cdots, n; j = 1, 2, \cdots, J$.

Given an initial weight $\mathbf{w}^0$, the batch learning of standard CBP updates the weights iteratively by

$$u_l^{k+1} = u_l^k - \eta\left(E_{u_l^{k,R}}(\mathbf{w}) + iE_{u_l^{k,I}}(\mathbf{w})\right), \tag{20}$$

$$v_{lm}^{k+1} = v_{lm}^k - \eta\left(E_{v_{lm}^{k,R}}(\mathbf{w}) + iE_{v_{lm}^{k,I}}(\mathbf{w})\right), \tag{21}$$

where $k = 0, 1, 2, \cdots$; $l = 1, 2, \cdots, n$; $m = 1, 2, \cdots, p$; $\eta > 0$ is the learning rate.

2) FRACTIONAL-ORDER CBP ALGORITHM BASED ON CAPUTO FRACTIONAL-ORDER DERIVATIVE

Given any initial weight $\mathbf{w}^0 = (\mathbf{u}^0, \mathbf{V}^0)$, without loss of generality, assume that $c = \min\{u_l{}^{k,R}, u_l{}^{k,I}, v_{lm}{}^{k,R}, v_{lm}{}^{k,I}\}(k = 0, 1, 2, \ldots; l = 1, \ldots, n; m = 1, \ldots, p)$, where k represents the kth iteration. The CBP network with Caputo α-order derivative updates the weights $\{\mathbf{w}^k\}$ iteratively by

$$u_l^{k+1} = u_l^k - \eta\left({}_cD^{\alpha}_{u_l^{k,R}}E(\mathbf{w}) + i{}_cD^{\alpha}_{u_l^{k,I}}E(\mathbf{w})\right), \tag{22}$$

$$v_{lm}^{k+1} = v_{lm}^k - \eta\left({}_cD^{\alpha}_{v_{lm}^{k,R}}E(\mathbf{w}) + i{}_cD^{\alpha}_{v_{lm}^{k,I}}E(\mathbf{w})\right), \tag{23}$$

where $\eta > 0$ is the learning rate, $0 < \alpha < 1$ is the fractional order, ${}_cD^{\alpha}_{u_l^{k,R}}E(\mathbf{w})$, ${}_cD^{\alpha}_{u_l^{k,I}}E(\mathbf{w})$, ${}_cD^{\alpha}_{v_{lm}^{k,R}}E(\mathbf{w})$ and ${}_cD^{\alpha}_{v_{lm}^{k,I}}E(\mathbf{w})$ are the Caputo α-order derivative with respect to $u_l^{k,R}$, $u_l^{k,I}$, $v_{lm}^{k,R}$ and $v_{lm}^{k,I}$, respectively ($l = 1, 2, \ldots, n, m = 1, 2, \ldots, p$).

According to the definition of Caputo fractional derivative, the fractional-order differential of a compostive function can be inferred as the the product of an integer-order differential and a fractional order differential. Suppose that $h(s(t))$ is a composite function, the α-order ($0 < \alpha < 1$) differential with respect to t is as follows

$${}_aD_t^{\alpha}h(s) = \frac{\partial}{\partial s}(h(s))_aD_t^{\alpha}s(t). \tag{24}$$

Next, we will divide four parts to obtain the expressions of ${}_cD^{\alpha}_{u_l^{k,R}}E(\mathbf{w})$, ${}_cD^{\alpha}_{u_l^{k,I}}E(\mathbf{w})$, ${}_cD^{\alpha}_{v_{lm}^{k,R}}E(\mathbf{w})$ and ${}_cD^{\alpha}_{v_{lm}^{k,I}}E(\mathbf{w})$.

Part 1. ${}_cD^{\alpha}_{u_l^{k,R}}E(\mathbf{w})$.

For the jth sample, it is easily to know from (12),

$$\begin{aligned} S^j &= S^{j,R} + iS^{j,I} \\ &= \sum_{l=1}^{n}\left(u_l^R H_l^{j,R} - u_l^I H_l^{j,I}\right) + i\sum_{l=1}^{n}\left(u_l^I H_l^{j,R} + u_l^R H_l^{j,I}\right) \\ &= \begin{pmatrix}\mathbf{u}^R \\ -\mathbf{u}^I\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{j,R} \\ \mathbf{H}^{j,I}\end{pmatrix} + i\begin{pmatrix}\mathbf{u}^I \\ \mathbf{u}^R\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{j,R} \\ \mathbf{H}^{j,I}\end{pmatrix}. \end{aligned} \tag{25}$$

Since $S^{j,R}$ and $S^{j,I}$ all contain variable $u_l^{k,R}$, the fractional gradient of the error function with respect to $u_l^{k,R}$ includes two parts. According to (24) and (14), we have

$${}_cD^{\alpha}_{u_l^{k,R}}E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial S^{j,R}}{}_cD^{\alpha}_{u_l^{k,R}}(S^{j,R}) + \frac{\partial E(\mathbf{w})}{\partial S^{j,I}}{}_cD^{\alpha}_{u_l^{k,R}}(S^{j,I}), \tag{26}$$

where

$$\frac{\partial E(\mathbf{w})}{\partial S^{j,R}} = \sum_{j=1}^{J}f'_{jR}(S^{j,R}), \quad \frac{\partial E(\mathbf{w})}{\partial S^{j,I}} = \sum_{j=1}^{J}f'_{jI}(S^{j,I}). \tag{27}$$

In order to calculate ${}_cD^{\alpha}_{u_l^{k,R}}(S^{j,R})$, we need to utilize the definition of (4) and (25), then

$$\begin{aligned} {}_cD^{\alpha}_{u_l^{k,R}}(S^{j,R}) &= \frac{1}{\Gamma(1-\alpha)}\int_c^{u_l^{k,R}}(u_l^{k,R}-\tau)^{-\alpha}H_l^{j,R}d\tau \\ &= \frac{1}{\Gamma(1-\alpha)}H_l^{j,R}\int_c^{u_l^{k,R}}(u_l^{k,R}-\tau)^{-\alpha}d\tau \\ &= \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha}H_l^{j,R}(u_l^{k,R}-c)^{1-\alpha}. \end{aligned} \tag{28}$$

Similarly,

$$\begin{aligned} {}_cD^{\alpha}_{u_l^{k,R}}(S^{j,I}) &= \frac{1}{\Gamma(1-\alpha)}\int_c^{u_l^{k,R}}(u_l^{k,R}-\tau)^{-\alpha}H_l^{j,I}d\tau \\ &= \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha}H_l^{j,I}(u_l^{k,R}-c)^{1-\alpha}. \end{aligned} \tag{29}$$

Therefore, we obtain the expanded formula of ${}_cD^{\alpha}_{u_l^{k,R}}E(\mathbf{w})$, it follows from $(26)-(29)$ that

$$\begin{aligned} cD^{\alpha}_{u_l^{k,R}}E(\mathbf{w}) &= \frac{1}{(1-\alpha)\Gamma(1-\alpha)} \\ &\times\sum_{j=1}^{J}\left(f'_{jR}(S^{j,R})H_l^{j,R} + f'_{jI}(S^{j,I})H_l^{j,I}\right) \\ &\times(u_l^{k,R}-c)^{1-\alpha}. \end{aligned} \tag{30}$$

Part 2. ${}_cD^{\alpha}_{u_l^{k,I}}E(\mathbf{w})$.

Since $S^{j,R}$ and $S^{j,I}$ all contain variable $u_l^{k,I}$, the fractional gradient of the error function with respect to $u_l^{k,I}$ includes the following two parts. According to (24) and (14), we have

$$ {}_cD^{\alpha}_{u_l^{k,I}}E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial S^{j,R}}\,{}_cD^{\alpha}_{u_l^{k,I}}(S^{j,R}) + \frac{\partial E(\mathbf{w})}{\partial S^{j,I}}\,{}_cD^{\alpha}_{u_l^{k,I}}(S^{j,I}), \tag{31}$$

where

$$\frac{\partial E(\mathbf{w})}{\partial S^{j,R}} = \sum_{j=1}^{J} f'_{jR}(S^{j,R}), \quad \frac{\partial E(\mathbf{w})}{\partial S^{j,I}} = \sum_{j=1}^{J} f'_{jI}(S^{j,I}). \tag{32}$$

Employing (4) and (25), the Caputo fractional derivative of $S^{j,R}$ and $S^{j,I}$ with respect to $u_l^{k,I}$ are, respectively, expressed by

$${}_cD^{\alpha}_{u_l^{k,I}}(S^{j,R}) = \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha}(-H_l^{j,I})(u_l^{k,I}-c)^{1-\alpha}, \tag{33}$$

$${}_cD^{\alpha}_{u_l^{k,I}}(S^{j,I}) = \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha}H_l^{j,R}(u_l^{k,I}-c)^{1-\alpha}. \tag{34}$$

Therefore, based on (31) – (34), the expanded formula of ${}_cD^{\alpha}_{u_l^{k,I}}E(\mathbf{w})$ is as follow

$$\begin{aligned}{}_cD^{\alpha}_{u_l^{k,I}}E(\mathbf{w}) &= \frac{1}{(1-\alpha)\Gamma(1-\alpha)}\sum_{j=1}^{J} \\ &\quad\times\left(-f'_{jR}(S^{j,R})H_l^{j,I} + f'_{jI}(S^{j,I})H_l^{j,R}\right)(u_l^{k,I}-c)^{1-\alpha}. \end{aligned}\tag{35}$$

Part 3. ${}_cD^{\alpha}_{v_{lm}^{k,R}}E(\mathbf{w})$.

For the jth sample, it is easily to get from (8),

$$\begin{aligned}\theta_l &= \theta_l^R + i\theta_l^I \\ &= \sum_{m=1}^{p}\left(v_{lm}^R x_m - v_{lm}^I y_m\right) + i\sum_{m=1}^{p}\left(v_{lm}^I x_m + v_{lm}^R y_m\right) \\ &= \begin{pmatrix}\mathbf{v}_l^R \\ -\mathbf{v}_l^I\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} + i\begin{pmatrix}\mathbf{v}_l^I \\ \mathbf{v}_l^R\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}.\end{aligned}\tag{36}$$

Since $\theta_l^{j,R}$ and $\theta_l^{j,I}$ all contain variable $v_{lm}^{k,R}$, the fractional gradient of the error function with respect to $v_{lm}^{k,R}$ includes the following two parts. According to (24) and (14), we have

$${}_cD^{\alpha}_{v_{lm}^{k,R}}E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,R}}\,{}_cD^{\alpha}_{v_{lm}^{k,R}}(\theta_l^{j,R}) + \frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,I}}\,{}_cD^{\alpha}_{v_{lm}^{k,R}}(\theta_l^{j,I}), \tag{37}$$

where

$$\begin{aligned}\frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,R}} &= \sum_{j=1}^{J}\left(f'_{jR}(S^{j,R})u_l^{k,R}g'(\theta_l^{j,R}) + f'_{jI}(S^{j,I})u_l^{k,I}g'(\theta_l^{j,R})\right), \\ \frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,I}} &= \sum_{j=1}^{J}\left(f'_{jR}(S^{j,R})(-u_l^{k,I})g'(\theta_l^{j,I}) + f'_{jI}(S^{j,I})u_l^{k,R}g'(\theta_l^{j,I})\right).\end{aligned}\tag{38}$$

Employing (4) and (36), the Caputo fractional derivative of $\theta_l^{j,R}$ and $\theta_l^{j,I}$ with respect to $v_{lm}^{k,R}$ are, respectively, expressed by

$${}_cD^{\alpha}_{v_{lm}^{k,R}}(\theta_l^{j,R}) = \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha}x_m^j(v_{lm}^{k,R}-c)^{1-\alpha}, \tag{39}$$

$${}_cD^{\alpha}_{v_{lm}^{k,R}}(\theta_l^{j,I}) = \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha}y_m^j(v_{lm}^{k,R}-c)^{1-\alpha}. \tag{40}$$

Therefore, it follows from (37) – (40) that

$$\begin{aligned}{}_{c_c}D^{\alpha}_{v_{lm}^{k,R}}E(\mathbf{w}) &= \frac{1}{(1-\alpha)\Gamma(1-\alpha)} \\ &\quad\times\sum_{j=1}^{J}\Big[f'_{jR}(S^{j,R})\left(u_l^{k,R}g'(\theta_l^{j,R})x_m^j - u_l^{k,I}g'(\theta_l^{j,I})y_m^j\right) \\ &\quad + f'_{jI}(S^{j,I})\times\left(u_l^{k,I}g'(\theta_l^{j,R})x_m^j + u_l^{k,R}g'(\theta_l^{j,I})y_m^j\right)\Big] \\ &\quad\times(v_{lm}^{k,R}-c)^{1-\alpha}.\end{aligned}\tag{41}$$

Part 4. ${}_cD^{\alpha}_{v_{lm}^{k,I}}E(\mathbf{w})$.

Since $\theta_l^{j,R}$ and $\theta_l^{j,I}$ all contain variable $v_{lm}^{k,I}$, the fractional gradient of the error function with respect to $v_{lm}^{k,I}$ includes the following two parts. According to (24) and (14), we have

$${}_cD^{\alpha}_{v_{lm}^{k,I}}E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,R}}\,{}_cD^{\alpha}_{v_{lm}^{k,I}}(\theta_l^{j,R}) + \frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,I}}\,{}_cD^{\alpha}_{v_{lm}^{k,I}}(\theta_l^{j,I}), \tag{42}$$

where

$$\begin{aligned}\frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,R}} &= \sum_{j=1}^{J}\Big(f'_{jR}(S^{j,R})u_l^{k,R}g'(\theta_l^{j,R}) \\ &\quad + f'_{jI}(S^{j,I})u_l^{k,I}g'(\theta_l^{j,R})\Big), \\ \frac{\partial E(\mathbf{w})}{\partial \theta_l^{j,I}} &= \sum_{j=1}^{J}\Big(f'_{jR}(S^{j,R})(-u_l^{k,I})g'(\theta_l^{j,I}) \\ &\quad + f'_{jI}(S^{j,I})u_l^{k,R}g'(\theta_l^{j,I})\Big).\end{aligned}\tag{43}$$

Employing (4) and (36), the Caputo fractional derivative of $\theta_l^{j,R}$ and $\theta_l^{j,I}$ with respect to $v_{lm}^{k,I}$ are, respectively, expressed by

$${}_cD^{\alpha}_{v_{lm}^{k,I}}(\theta_l^{j,R}) = \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha} - y_m^j(v_{lm}^{k,I}-c)^{1-\alpha}, \tag{44}$$

$${}_cD^{\alpha}_{v_{lm}^{k,I}}(\theta_l^{j,I}) = \frac{1}{\Gamma(1-\alpha)}\frac{1}{1-\alpha}x_m^j(v_{lm}^{k,I}-c)^{1-\alpha}. \tag{45}$$

Therefore, it follows from (42) – (45) that

$$
\begin{aligned}
&{}_cD^{\alpha}_{v^{k,I}_{lm}}E(\mathbf{w}) \\
&= \frac{1}{(1-\alpha)\Gamma(1-\alpha)} \\
&\quad\times\sum_{j=1}^{J}\Big[f'_{jR}(S^{j,R})\Big(-u^{k,R}_l g'(\theta^{j,R}_l)y^j_m \\
&\qquad -u^{k,I}_l g'(\theta^{j,I}_l)x^j_m\Big)+f'_{jI}(S^{j,I})\Big(-u^{k,I}_l g'(\theta^{j,R}_l) \\
&\qquad y^j_m+u^{k,R}_l g'(\theta^{j,I}_l)x^j_m\Big)\Big](v^{k,I}_{lm}-c)^{1-\alpha}. \qquad (46)
\end{aligned}
$$

This then completes the CBP network with Caputo α-order derivative weights updates (22) and (23).

IV. MAIN RESULTS

In this section, the convergence behavior of the Caputo fractional-order derivative for CVBP algorithm is presented. The value range of α is $0 < \alpha < 1$. The following assumptions are imposed for the convergence of Caputo-type fractional-order CVNNs:

$(A1)$ There exists a constant $c_1 > 0$ such that $\max_{t\in\mathbb{R}^1}\{|g(t)|, |g'(t)|, |g''(t)|, |f(t)|, |f'(t)|, |f''(t)|\} \le c_1$;

$(A2)$ There exists a constant $c_2 > 0$ such that $||\mathbf{w}^{k,R}|| \le c_2$, $||\mathbf{w}^{k,I}|| \le c_2$ for all $k = 0, 1, 2, \dots$;

$(A3)$ η is chosen to satisfy $0 < \eta < \frac{c_9}{c_8}$, where $\frac{c_9}{c_8}$ is a constant defined in (76) below.

Theorem 1: Suppose that the error function $E(\mathbf{w})$ is defined by (14), $\mathbf{w}^0 \in \mathbb{C}^{n(p+1)}$ be an arbitrary initial value, the weight sequence $\{\mathbf{w}^k\}$ be generated by (22) and (23). Assume that conditions $(A1)-(A3)$ are valid, then we have,

(i) $E(\mathbf{w}^{k+1}) \le E(\mathbf{w}^k), \quad k = 0, 1, \dots;$

(ii) There exists $E^* \ge 0$ such that $\lim_{k\to\infty} E(\mathbf{w}^k) = E^*$;

(iii) $\lim_{k\to\infty} \|{}_cD^{\alpha}_{\mathbf{u}^R}E(\mathbf{w}^k)\| = 0$, $\lim_{k\to\infty} \|{}_cD^{\alpha}_{\mathbf{u}^I}E(\mathbf{w}^k)\| = 0$, $\lim_{k\to\infty} \|{}_cD^{\alpha}_{\mathbf{v}^R_l}E(\mathbf{w}^k)\| = 0$, $\lim_{k\to\infty} \|{}_cD^{\alpha}_{\mathbf{v}^I_l}E(\mathbf{w}^k)\| = 0 (k = 0, 1, 2, \dots; l = 1, 2, \dots, n)$.

V. NUMERICAL EXAMPLE

To verify the convergence of the proposed fractional steepest descent method for split-complex neural networks, we demonstrate it by a numerical simulation. Simulation has been done on a well-known XOR problem.

The training samples of the XOR problem for complex-value neural networks are presented as follows:

$$
\begin{aligned}
&\{\mathbf{z}^1 = (0, -1-i)^T, d^1 = 0\},\\
&\{\mathbf{z}^2 = (i, -1-i)^T, d^2 = 1\},\\
&\{\mathbf{z}^3 = (1, -1-i)^T, d^3 = 1+i\},\\
&\{\mathbf{z}^4 = (1+i, -1-i)^T, d^4 = i\}.
\end{aligned}
$$

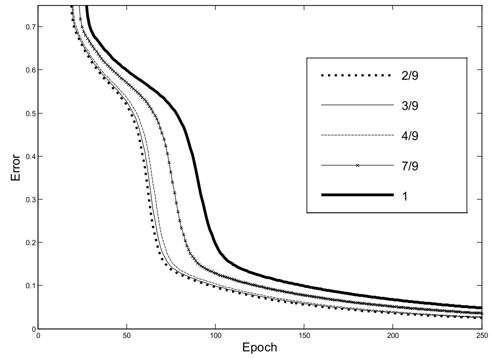


FIGURE 1. The comparison of different fractional and integer order CBP algorithms for the same learning rates with fixed numbers of hidden nodes.

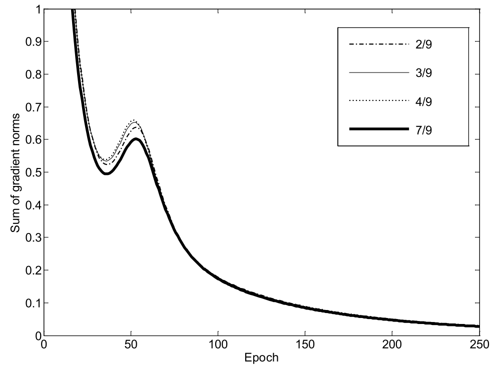


FIGURE 2. The sum of gradient norms for the same learning rates with fixed numbers of hidden nodes.

We use a network with two input neurons, ten hidden neurons, and one output neuron. We set the learning rate $\eta = 0.05$. The activation function is *tansig*$(\cdot)$ for hidden layer and output layer function. In this example, we employ different fractional α-order derivatives, where $\alpha = \frac{2}{9}, \frac{3}{9}, \frac{4}{9}, \frac{7}{9}$ and $\frac{9}{9} = 1$, separately ($\alpha = 1$ is the equal of integer order derivative for the common CBP).

Fig. 1 shows that errors change for different fractional and integer order CBP algorithms. It demonstrates that the simulation of fractional orders perform better than that of integer order, and we can also find a phenomenon that the larger the order is, the slower the convergence rate in this simulation. Therefore, we can infer that fractional order CVNN performs better than integer order CVNN in our example. Fig. 2 implies that the sum of gradient norms change for different fractional CBP algorithms. Fig. 2 shows that the sum of the gradient norms go to zero for $\frac{2}{9}, \frac{3}{9}, \frac{4}{9}, \frac{7}{9}$. In addition, it also verifies the week convergence **theorem 1**.

VI. CONCLUSIONS

In this paper, we have extended the fractional steepest descent approach to CBP training of FNNs. The batch split-complex fractional order gradient descent method (BSC-FGD) for training complex value backpropagation is investigated. The monotonicity of error function and weak convergence of the proposed Caputo fractional-order CBP algorithm are derived. Finally, the numerical results support the theoretical conclusions very well.

APPENDIX

For the sake of description, we introduce the following notations:

$$\begin{aligned}\Delta u_l^k &= u_l^{k+1} - u_l^k \\ &= -\eta\left({}_cD^\alpha_{u_l^{k,R}}E(\mathbf{w}) + i\,{}_cD^\alpha_{u_l^{k,I}}E(\mathbf{w})\right),\end{aligned} \quad (47)$$

$$\begin{aligned}\Delta v_{lm}^k &= v_{lm}^{k+1} - v_{lm}^k \\ &= -\eta\left({}_cD^\alpha_{v_{lm}^{k,R}}E(\mathbf{w}) + i\,{}_cD^\alpha_{v_{lm}^{k,I}}E(\mathbf{w})\right),\end{aligned} \quad (48)$$

where

$$\Delta u_l^{k,R} = u_l^{k+1,R} - u_l^{k,R} = -\eta\,{}_cD^\alpha_{u_l^{k,R}}E(\mathbf{w}), \quad (49)$$

$$\Delta u_l^{k,I} = u_l^{k+1,I} - u_l^{k,I} = -\eta\,{}_cD^\alpha_{u_l^{k,I}}E(\mathbf{w}), \quad (50)$$

$$\Delta v_{lm}^{k,R} = v_{lm}^{k+1,R} - v_{lm}^{k,R} = -\eta\,{}_cD^\alpha_{v_{lm}^{k,R}}E(\mathbf{w}), \quad (51)$$

$$\Delta v_{lm}^{k,I} = v_{lm}^{k+1,I} - v_{lm}^{k,I} = -\eta\,{}_cD^\alpha_{v_{lm}^{k,I}}E(\mathbf{w}), \quad (52)$$

$$\begin{aligned}\theta_l^{k,j} &= \theta_l^{k,j,R} + i\theta_l^{k,j,I} \\ &= \begin{pmatrix}\mathbf{v}_l^{k,R}\\ -\mathbf{v}_l^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix} + i\begin{pmatrix}\mathbf{v}_l^{k,I}\\ \mathbf{v}_l^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}, \\ H_l^{k,j} &= H_l^{k,j,R} + iH_l^{k,j,I} = g\left(\theta_l^{k,j,R}\right) + ig\left(\theta_l^{k,j,I}\right), \\ \mathbf{H}^{k,j,R} &= \left(H_1^{k,j,R}, H_2^{k,j,R}, \ldots, H_n^{k,j,R}\right)^T, \\ \mathbf{H}^{k,j,I} &= \left(H_1^{k,j,I}, H_2^{k,j,I}, \ldots, H_n^{k,j,I}\right)^T, \\ S^{k,j} &= S^{k,j,R} + iS^{k,j,I} \\ &= \begin{pmatrix}\mathbf{u}^{k,R}\\ -\mathbf{u}^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{k,j,R}\\ \mathbf{H}^{k,j,I}\end{pmatrix} + i\begin{pmatrix}\mathbf{u}^{k,I}\\ \mathbf{u}^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{k,j,R}\\ \mathbf{H}^{k,j,I}\end{pmatrix}, \\ \psi^{k,j,R} &= \mathbf{H}^{k+1,j,R} - \mathbf{H}^{k,j,R}, \\ \psi^{k,j,I} &= \mathbf{H}^{k+1,j,I} - \mathbf{H}^{k,j,I},\end{aligned} \quad (53)$$

where $k \in \mathbb{N}$; $l = 1, 2, \ldots, n$; $m = 1, 2, \ldots, p$.

Lemma 1: Suppose Assumptions $A1, A2$ hold, desired outputs $\{d^j\}_{j=1}^J$ satisfy $|d^{j,R}| \le c_0$, $|d^{j,I}| \le c_0$, for $1 \le j \le J$, $k = 0, 1, 2, \ldots$, then, there are constants $c_i (i = 3, \ldots, 7)$ such that

$$\|\mathbf{H}^{k,j,R}\| \le c_0, \|\mathbf{H}^{k,j,I}\| \le c_0, \quad (54)$$

$$\begin{aligned}&|f'_{jR}(t)| \le c_3, |f'_{jI}(t)| \le c_3, \\ &|f''_{jR}(t)| \le c_3, |f''_{jI}(t)| \le c_3,\end{aligned} \quad (55)$$

$$\begin{aligned}&\max\{\|\psi^{k,j,R}\|^2, \|\psi^{k,j,I}\|^2\} \\ &\quad\le c_4\sum_{l=1}^n(\|\Delta\mathbf{v}_l^{k,R}\|^2 + \Delta\mathbf{v}_l^{k,I}\|^2),\end{aligned} \quad (56)$$

$$\begin{aligned}\delta_1 &= \sum_{j=1}^J\left(f'_{jR}(S^{k,j,R})\begin{pmatrix}\Delta\mathbf{u}^{k,R}\\ -\Delta\mathbf{u}^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{k,j,R}\\ \mathbf{H}^{k,j,I}\end{pmatrix}\right. \\ &\quad\left.+f'_{jI}(S^{k,j,I})\begin{pmatrix}\Delta\mathbf{u}^{k,I}\\ \Delta\mathbf{u}^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{k,j,R}\\ \mathbf{H}^{k,j,I}\end{pmatrix}\right) \\ &\le -\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}(||\Delta\mathbf{u}^{k,R}||^2 \\ &\quad+||\Delta\mathbf{u}^{k,I}||^2),\end{aligned} \quad (57)$$

$$\begin{aligned}\delta_2 &= \sum_{j=1}^J\left(f'_{jR}(S^{k,j,R})\begin{pmatrix}\mathbf{u}^{k,R}\\ -\mathbf{u}^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\psi^{k,j,R}\\ \psi^{k,j,I}\end{pmatrix}\right. \\ &\quad\left.+f'_{jI}(S^{k,j,I})\begin{pmatrix}\mathbf{u}^{k,I}\\ \mathbf{u}^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\psi^{k,j,R}\\ \psi^{k,j,I}\end{pmatrix}\right) \\ &\le \left(c_5 - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\right) \\ &\quad\sum_{l=1}^n(||\Delta\mathbf{v}_l^{k,R}||^2 + ||\Delta\mathbf{v}_l^{k,I}||^2),\end{aligned} \quad (58)$$

$$\begin{aligned}\delta_3 &= \sum_{j=1}^J\left(f'_{jR}(S^{k,j,R})\begin{pmatrix}\Delta\mathbf{u}^{k,R}\\ -\Delta\mathbf{u}^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\psi^{k,j,R}\\ \psi^{k,j,I}\end{pmatrix}\right. \\ &\quad\left.+f'_{jI}(S^{k,j,I})\begin{pmatrix}\Delta\mathbf{u}^{k,I}\\ \Delta\mathbf{u}^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\psi^{k,j,R}\\ \psi^{k,j,I}\end{pmatrix}\right) \\ &\le c_6\left(\sum_{l=1}^n(||\Delta\mathbf{v}_l^{k,R}||^2 + ||\Delta\mathbf{v}_l^{k,I}||^2) + ||\Delta\mathbf{u}^{k,R}||^2\right. \\ &\quad\left.+||\Delta\mathbf{u}^{k,I}||^2\right),\end{aligned} \quad (59)$$

$$\begin{aligned}\delta_4 &= \frac{1}{2}\sum_{j=1}^J\left(f''_{jR}(t_1^{k,j})(S^{k+1,j,R} - S^{k,j,R})^2\right. \\ &\quad\left.+f''_{jI}(t_2^{k,j})(S^{k+1,j,I} - S^{k,j,I})^2\right) \\ &\le c_7\left(\sum_{l=1}^n(||\Delta\mathbf{v}_l^{k,R}||^2 + ||\Delta\mathbf{v}_l^{k,I}||^2) + ||\Delta\mathbf{u}^{k,R}||^2\right. \\ &\quad\left.+||\Delta\mathbf{u}^{k,I}||^2\right),\end{aligned} \quad (60)$$

where $t_1^{k,j} \in \mathbb{R}^1$ is a constant between $S^{k+1,j,R}$ and $S^{k,j,R}$, and $t_2^{k,j} \in \mathbb{R}^1$ is a constant between $S^{k+1,j,I}$ and $S^{k,j,I}$.

Proof: For (54)-(56), (59) and (60), the results is almost the same as the [22, Lemma 5.2], and the detail of the proof is omitted.

Because of the properties of the scalar product, δ_1 can be deduced as below:

$$\begin{aligned}\delta_1 &= \sum_{j=1}^J\left(f'_{jR}(S^{k,j,R})\begin{pmatrix}\Delta\mathbf{u}^{k,R}\\ -\Delta\mathbf{u}^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{k,j,R}\\ \mathbf{H}^{k,j,I}\end{pmatrix}\right. \\ &\quad\left.+f'_{jI}(S^{k,j,I})\begin{pmatrix}\Delta\mathbf{u}^{k,I}\\ \Delta\mathbf{u}^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{H}^{k,j,R}\\ \mathbf{H}^{k,j,I}\end{pmatrix}\right) \\ &= \sum_{j=1}^J\left(f'_{jR}(S^{k,j,R})H_1^{k,j,R} + f'_{jI}(S^{k,j,I})H_1^{k,j,I}\right)\Delta u_1^{k,R} \\ &\quad+\ldots \\ &\quad+\sum_{j=1}^J\left(f'_{jR}(S^{k,j,R})H_n^{k,j,R} + f'_{jI}(S^{k,j,I})H_n^{k,j,I}\right)\Delta u_n^{k,R} \\ &\quad+\sum_{j=1}^J\left(-f'_{jR}(S^{k,j,R})H_1^{k,j,I} + f'_{jI}(S^{k,j,I})H_1^{k,j,R}\right)\Delta u_1^{k,I} \\ &\quad+\cdots+\sum_{j=1}^J\left(-f'_{jR}(S^{k,j,R})H_n^{k,j,I} + f'_{jI}(S^{k,j,I})H_n^{k,j,R}\right) \\ &\quad\times\Delta u_n^{k,I}.\end{aligned} \quad (61)$$

It is easy to find that there is a close relationship between the two factors of each item. For example, $f'_{jR}(S^{k,j,R})H_1^{k,j,R} + f'_{jI}(S^{k,j,I})H_1^{k,j,I}$ is a part of $\Delta u_1^{k,R}$ in first item.

Next, for avoiding the situation that numerator equals 0 during the calculation, we will divide the value of δ_1 into two situations.

Case 1. If for all $(u_l^{k,R}-c)^{1-\alpha} \neq 0$, $(u_l^{k,I}-c)^{1-\alpha} \neq 0$. $l = 1, \cdots, n$. Substituting (30), (35) and (49), (50) into (61) and according to ($A2$), we have

$$\begin{aligned}\delta_1 &= -\frac{1}{\eta}(\Delta u_1^{k,R})^2(1-\alpha)\Gamma(1-\alpha)(u_1^{k,R}-c)^{\alpha-1}\\ &\quad - \cdots - \frac{1}{\eta}(\Delta u_n^{k,R})^2(1-\alpha)\Gamma(1-\alpha)(u_n^{k,R}-c)^{\alpha-1}\\ &\quad - \frac{1}{\eta}(\Delta u_1^{k,I})^2(1-\alpha)\Gamma(1-\alpha)(u_1^{k,I}-c)^{\alpha-1}\\ &\quad - \cdots - \frac{1}{\eta}(\Delta u_n^{k,I})^2(1-\alpha)\Gamma(1-\alpha)(u_n^{k,I}-c)^{\alpha-1}\\ &\leq -\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\quad \times\left(\sum_{l=1}^{n}(\Delta u_l^{k,R})^2 + \sum_{l=1}^{n}(\Delta u_l^{k,I})^2\right)\\ &= -\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\quad \times\left(\|\Delta \mathbf{u}^{k,R}\|^2 + \|\Delta \mathbf{u}^{k,I}\|^2\right). \end{aligned} \tag{62}$$

Case 2. If there exists $(u_l^k - c)^{1-\alpha} = 0$, $l = 1, \ldots, n$. Without loss of generality, we assume that $(u_1^{k,R}-c)^{1-\alpha} = 0$. By (30) and (49), it is easy to prove that ${}_cD^{\alpha}_{u_1^{k,R}}E(\mathbf{w}) = 0$ and $\Delta u_1^{k,R} = 0$ are valid. Hence, (61) is induced as follows

$$\begin{aligned}\delta_1 &= 0 - \frac{1}{\eta}(\Delta u_2^{k,R})^2(1-\alpha)\Gamma(1-\alpha)(u_2^{k,R}-c)^{\alpha-1}\\ &\quad - \cdots - \frac{1}{\eta}(\Delta u_n^{k,R})^2(1-\alpha)\Gamma(1-\alpha)(u_n^{k,R}-c)^{\alpha-1}\\ &\quad - \frac{1}{\eta}(\Delta u_1^{k,I})^2(1-\alpha)\Gamma(1-\alpha)(u_1^{k,I}-c)^{\alpha-1}\\ &\quad - \cdots - \frac{1}{\eta}(\Delta u_n^{k,I})^2(1-\alpha)\Gamma(1-\alpha)(u_n^{k,I}-c)^{\alpha-1}\\ &\leq -\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\quad \times\left(\sum_{l=1}^{n}(\Delta u_l^{k,R})^2 + \sum_{l=1}^{n}(\Delta u_l^{k,I})^2\right)\\ &= -\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\quad \times\left(\|\Delta \mathbf{u}^{k,R}\|^2 + \|\Delta \mathbf{u}^{k,I}\|^2\right). \end{aligned} \tag{63}$$

Consequently, we can reach the the upper bound of δ_1 following by the above **Case 1** and **Case 2**

$$\begin{aligned}\delta_1 &\leq -\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\quad\times\left(\|\Delta \mathbf{u}^{k,R}\|^2 + \|\Delta \mathbf{u}^{k,I}\|^2\right). \end{aligned} \tag{64}$$

$$\begin{aligned}\delta_2 &= \sum_{j=1}^{J}\Bigg(f'_{jR}(S^{k,j,R})\begin{pmatrix}\mathbf{u}^{k,R}\\ -\mathbf{u}^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\psi^{k,j,R}\\ \psi^{k,j,I}\end{pmatrix}\\ &\qquad + f'_{jI}(S^{k,j,I})\begin{pmatrix}\mathbf{u}^{k,I}\\ \mathbf{u}^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\psi^{k,j,R}\\ \psi^{k,j,I}\end{pmatrix}\Bigg)\\ &= \sum_{j=1}^{J}\sum_{l=1}^{n}\Big(f'_{jR}(S^{k,j,R})u_l^{k,R}(H_l^{k+1,j,R} - H_l^{k,j,R})\\ &\qquad - f'_{jR}(S^{k,j,R})u_l^{k,I}(H_l^{k+1,j,I} - H_l^{k,j,I})\\ &\qquad + f'_{jI}(S^{k,j,I})u_l^{k,I}(H_l^{k+1,j,R} - H_l^{k,j,R})\\ &\qquad + f'_{jI}(S^{k,j,I})u_l^{k,R}(H_l^{k+1,j,I} - H_l^{k,j,I})\Big)\\ &= \sum_{j=1}^{J}\sum_{l=1}^{n}\Bigg(f'_{jR}(S^{k,j,R})u_l^{k,R}g'(\theta_l^{k,j,R})\begin{pmatrix}\Delta\mathbf{v}_l^{k,R}\\ -\Delta\mathbf{v}_l^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\\ &\qquad - f'_{jR}(S^{k,j,R})u_l^{k,I}g'(\theta_l^{k,j,I})\begin{pmatrix}\Delta\mathbf{v}_l^{k,I}\\ \Delta\mathbf{v}_l^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\\ &\qquad + f'_{jI}(S^{k,j,I})u_l^{k,I}g'(\theta_l^{k,j,R})\begin{pmatrix}\Delta\mathbf{v}_l^{k,R}\\ -\Delta\mathbf{v}_l^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\\ &\qquad + f'_{jI}(S^{k,j,I})u_l^{k,R}g'(\theta_l^{k,j,I})\begin{pmatrix}\Delta\mathbf{v}_l^{k,I}\\ \Delta\mathbf{v}_l^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\Bigg)\\ &\quad + \gamma\\ &= \varsigma + \gamma. \end{aligned} \tag{65}$$

The reason of the third equation in (65) establishes is the virtue of the Taylor expansion. For convenience, we note the first item as ς in the last equation, and

$$\begin{aligned}\gamma &= \frac{1}{2}\sum_{j=1}^{J}\sum_{l=1}^{n}\\ &\quad\times\Bigg(f'_{jR}(S^{k,j,R})u_l^{k,R}g''(t_l^{k,j,R})\left(\begin{pmatrix}\Delta\mathbf{v}_l^{k,R}\\ -\Delta\mathbf{v}_l^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\right)^2\\ &\qquad - f'_{jR}(S^{k,j,R})u_l^{k,I}g''(t_l^{k,j,I})\left(\begin{pmatrix}\Delta\mathbf{v}_l^{k,I}\\ \Delta\mathbf{v}_l^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\right)^2\\ &\qquad + f'_{jI}(S^{k,j,I})u_l^{k,I}g''(t_l^{k,j,R})\left(\begin{pmatrix}\Delta\mathbf{v}_l^{k,R}\\ -\Delta\mathbf{v}_l^{k,I}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\right)^2\\ &\qquad + f'_{jI}(S^{k,j,I})u_l^{k,R}g''(t_l^{k,j,I})\left(\begin{pmatrix}\Delta\mathbf{v}_l^{k,I}\\ \Delta\mathbf{v}_l^{k,R}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}^j\\ \mathbf{y}^j\end{pmatrix}\right)^2\Bigg). \end{aligned} \tag{66}$$

Firstly, we will discuss the first part of the right hand side of (65). Based on the properties of the scalar product and similar terms, ς can be inferred as follows

$$\begin{aligned}\varsigma = &\sum_{l=1}^{n}\sum_{j=1}^{J}\Big[f'_{jR}(S^{k,j,R})\big(u_l^{k,R}g'(\theta_l^{k,j,R})x_1^j - u_l^{k,I}g'(\theta_l^{k,j,I})\\ &y_1^j\big) + f'_{jI}(S^{k,j,I})\left(u_l^{k,I}g'(\theta_l^{k,j,R})x_1^j + u_l^{k,R}g'(\theta_l^{k,j,I})y_1^j\right)\Big]\\ &\Delta v_{l1}^{k,R} + \dots\\ &+\sum_{l=1}^{n}\sum_{j=1}^{J}\Big[f'_{jR}(S^{k,j,R})\big(u_l^{k,R}g'(\theta_l^{k,j,R})x_p^j - u_l^{k,I}g'(\theta_l^{k,j,I})\\ &y_p^j\big) + f'_{jI}(S^{k,j,I})\left(u_l^{k,I}g'(\theta_l^{k,j,R})x_p^j + u_l^{k,R}g'(\theta_l^{k,j,I})y_p^j\right)\Big]\\ &\Delta v_{lp}^{k,R}\\ &+\sum_{l=1}^{n}\sum_{j=1}^{J}\Big[f'_{jR}(S^{k,j,R})\big(-u_l^{k,R}g'(\theta_l^{k,j,R})y_1^j - u_l^{k,I}g'(\theta_l^{k,j,I})\\ &x_1^j\big) + f'_{jI}(S^{k,j,I})\left(-u_l^{k,I}g'(\theta_l^{k,j,R})y_1^j + u_l^{k,R}g'(\theta_l^{k,j,I})x_1^j\right)\Big]\\ &\Delta v_{l1}^{k,I} + \dots\\ &+\sum_{l=1}^{n}\sum_{j=1}^{J}\Big[f'_{jR}(S^{k,j,R})\big(-u_l^{k,R}g'(\theta_l^{k,j,R})y_p^j - u_l^{k,I}g'(\theta_l^{k,j,I})\\ &x_p^j\big) + f'_{jI}(S^{k,j,I})\left(-u_l^{k,I}g'(\theta_l^{k,j,R})y_p^j + u_l^{k,R}g'(\theta_l^{k,j,I})x_p^j\right)\Big]\\ &\Delta v_{lp}^{k,I}. \end{aligned} \tag{67}$$

It is easy to find that there is a close relationship between the two factors of each item. For example,

$$\begin{aligned}&f'_{jR}(S^{k,j,R})\left(u_l^{k,R}g'(\theta_l^{k,j,R})x_1^j - u_l^{k,I}g'(\theta_l^{k,j,I})y_1^j\right)\\ &\quad + f'_{jI}(S^{k,j,I})\left(u_l^{k,I}g'(\theta_l^{k,j,R})x_1^j + u_l^{k,R}g'(\theta_l^{k,j,I})y_1^j\right)\end{aligned}$$

is a part of $\Delta v_{l1}^{k,R}$ in first item.

Similarly as the reason of δ_1, We will divide the value of ς into two situations .

Case 3. If for all $(v_{lm}^{k,R} - c)^{1-\alpha} \neq 0, (u_{lm}^{k,I} - c)^{1-\alpha} \neq 0, l = 1, \dots, n; m = 1, \dots, p$. Substituting (41), (46) and (51), (52) into (65) and according to $(A2)$, we have

$$\begin{aligned}\delta_2 = &\sum_{l=1}^{n}\Big(-\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{l1}^{k,R}-c)^{\alpha-1}(\Delta v_{l1}^{k,R})^2\\ &- \dots - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{lp}^{k,R}-c)^{\alpha-1}(\Delta v_{lp}^{k,R})^2\\ &- \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{l1}^{k,I}-c)^{\alpha-1}(\Delta v_{l1}^{k,I})^2\\ &- \dots - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{lp}^{k,I}-c)^{\alpha-1}(\Delta v_{lp}^{k,I})^2\Big) + \gamma\\ \leq &-\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\times\left(\sum_{l=1}^{n}\sum_{m=1}^{p}(\Delta v_{lm}^{k,R})^2 + \sum_{l=1}^{n}\sum_{m=1}^{p}(\Delta v_{lm}^{k,I})^2\right) + \gamma\\ = &-\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\sum_{l=1}^{n}\\ &\times\left(\|\Delta \mathbf{v}_l^{k,R}\|^2 + \|\Delta \mathbf{v}_l^{k,I}\|^2\right) + \gamma. \end{aligned} \tag{68}$$

Case 4. If there exists $(v_{lm}^{k} - c)^{1-\alpha} = 0, l = 1, \dots, n$. Without loss of generality, we assume that $(v_{l1}^{k,R} - c)^{1-\alpha} = 0$. By (41) and (51), it is easy to prove that ${}_cD^{\alpha}_{v_{l1}^{k,R}}E(\mathbf{w}) = 0$ and $\Delta v_{l1}^{k,R} = 0$ are valid. Hence, (65) is induced as follows

$$\begin{aligned}\delta_2 = &\sum_{l=1}^{n}\Big(0 - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{l2}^{k,R}-c)^{\alpha-1}(\Delta v_{l2}^{k,R})^2\\ &- \dots - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{lp}^{k,R}-c)^{\alpha-1}(\Delta v_{lp}^{k,R})^2\\ &- \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{l1}^{k,I}-c)^{\alpha-1}(\Delta v_{l1}^{k,I})^2\\ &- \dots - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(v_{lp}^{k,I}-c)^{\alpha-1}(\Delta v_{lp}^{k,I})^2\Big) + \gamma\\ \leq &-\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\times\left(\sum_{l=1}^{n}\sum_{m=1}^{p}(\Delta v_{lm}^{k,R})^2 + \sum_{l=1}^{n}\sum_{m=1}^{p}(\Delta v_{lm}^{k,I})^2\right) + \gamma\\ = &-\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\times\sum_{l=1}^{n}\left(\|\Delta \mathbf{v}_l^{k,R}\|^2 + \|\Delta \mathbf{v}_l^{k,I}\|^2\right) + \gamma. \end{aligned} \tag{69}$$

From **Case 3** and **Case 4**, we obtain that

$$\begin{aligned}\varsigma \leq &\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\\ &\times\sum_{l=1}^{n}(||\Delta \mathbf{v}_l^{k,R}||^2 + ||\Delta \mathbf{v}_l^{k,I}||^2). \end{aligned} \tag{70}$$

Then we will discuss the second part of the right hand side of (65). On the basis of Cauchy-Schwarz inequality, (VI) can be deduced that

$$\begin{aligned}\gamma \leq &\frac{1}{2}\sum_{j=1}^{J}\sum_{l=1}^{n}\Big(4c_3c_2c_1\left(\|\Delta \mathbf{v}_l^{k,R}\|^2\|\mathbf{x}^j\|^2 + \|\Delta \mathbf{v}_l^{k,I}\|^2\|\mathbf{y}^j\|^2\right)\\ &+ 4c_3c_2c_1\left(\|\Delta \mathbf{v}_l^{k,I}\|^2\|\mathbf{x}^j\|^2 + \|\Delta \mathbf{v}_l^{k,R}\|^2\|\mathbf{y}^j\|^2\right)\Big)\\ \leq &2c_1c_2c_3J\max_{1\leq j\leq J}\{\|\mathbf{x}^j\|^2 + \|\mathbf{y}^j\|^2\}\\ &\times\sum_{l=1}^{n}\left(\|\Delta \mathbf{v}_l^{k,R}\|^2 + \|\Delta \mathbf{v}_l^{k,I}\|^2\right)\\ = &c_5\sum_{l=1}^{n}\left(\|\Delta \mathbf{v}_l^{k,R}\|^2 + \|\Delta \mathbf{v}_l^{k,I}\|^2\right), \end{aligned} \tag{71}$$

where $c_5 = 2c_1c_2c_3J\max_{1\leq j\leq J}\{\|\mathbf{x}^j\|^2 + \|\mathbf{y}^j\|^2\}$.

Substituting (70) and (71) into (65), we have

$$\begin{aligned}\delta_2 \leq &\left(c_5 - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2-c)^{\alpha-1}\right)\\ &\times\sum_{l=1}^{n}\left(||\Delta \mathbf{v}_l^{k,R}||^2 + ||\Delta \mathbf{v}_l^{k,I}||^2\right). \end{aligned} \tag{72}$$

Therefore, we get the the upper bound of δ_2. This completes the proof of Lemma 1. ■

Proof to (i) of Theorem 1: By the error function (14), we have

$$E(\mathbf{w}^{k+1}) = \sum_{j=1}^{J} \left[f_{jR}(S^{k+1,j,R}) + f_{jI}(S^{k+1,j,I}) \right],$$

$$E(\mathbf{w}^{k}) = \sum_{j=1}^{J} \left[f_{jR}(S^{k,j,R}) + f_{jI}(S^{k,j,I}) \right]. \qquad (73)$$

By using the Taylor mean value theorem with Lagrange remainder, we have the following estimation

$$\begin{aligned} &E(\mathbf{w}^{k+1}) - E(\mathbf{w}^{k}) \\ &= \sum_{j=1}^{J} \Big[f_{jR}(S^{k+1,j,R}) - f_{jR}(S^{k,j,R}) \\ &\qquad + f_{jI}(S^{k+1,j,I}) - f_{jI}(S^{k,j,I}) \Big] \\ &= \sum_{j=1}^{J} \Big[f'_{jR}(S^{k,j,R})(S^{k+1,j,R} - S^{k,j,R}) \\ &\quad + f'_{jI}(S^{k,j,I})(S^{k+1,j,I} - S^{k,j,I}) \\ &\quad + \frac{1}{2} f''_{jR}(t_1^{k,j})(S^{k+1,j,R} - S^{k,j,R})^2 \\ &\quad + \frac{1}{2} f''_{jI}(t_2^{k,j})(S^{k+1,j,I} - S^{k,j,I})^2 \Big] \\ &= \sum_{j=1}^{J} \Bigg(f'_{jR}(S^{k,j,R}) \begin{pmatrix} \Delta\mathbf{u}^{k,R} \\ -\Delta\mathbf{u}^{k,I} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{H}^{k,j,R} \\ \mathbf{H}^{k,j,I} \end{pmatrix} \\ &\qquad + f'_{jI}(S^{k,j,I}) \begin{pmatrix} \Delta\mathbf{u}^{k,I} \\ \Delta\mathbf{u}^{k,R} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{H}^{k,j,R} \\ \mathbf{H}^{k,j,I} \end{pmatrix} \\ &\qquad + f'_{jR}(S^{k,j,R}) \begin{pmatrix} \mathbf{u}^{k,R} \\ -\mathbf{u}^{k,I} \end{pmatrix} \cdot \begin{pmatrix} \psi^{k,j,R} \\ \psi^{k,j,I} \end{pmatrix} \\ &\qquad + f'_{jI}(S^{k,j,I}) \begin{pmatrix} \mathbf{u}^{k,I} \\ \mathbf{u}^{k,R} \end{pmatrix} \cdot \begin{pmatrix} \psi^{k,j,R} \\ \psi^{k,j,I} \end{pmatrix} \\ &\qquad + f'_{jR}(S^{k,j,R}) \begin{pmatrix} \Delta\mathbf{u}^{k,R} \\ -\Delta\mathbf{u}^{k,I} \end{pmatrix} \cdot \begin{pmatrix} \psi^{k,j,R} \\ \psi^{k,j,I} \end{pmatrix} \\ &\qquad + f'_{jI}(S^{k,j,I}) \begin{pmatrix} \Delta\mathbf{u}^{k,I} \\ \Delta\mathbf{u}^{k,R} \end{pmatrix} \cdot \begin{pmatrix} \psi^{k,j,R} \\ \psi^{k,j,I} \end{pmatrix} \\ &\qquad + \frac{1}{2} f''_{jR}(t_1^{k,j})(S^{k+1,j,R} - S^{k,j,R})^2 \\ &\qquad + \frac{1}{2} f''_{jI}(t_2^{k,j})(S^{k+1,j,I} - S^{k,j,I})^2 \Bigg) \\ &= \delta_1 + \delta_2 + \delta_3 + \delta_4, \end{aligned} \qquad (74)$$

The third equation in (74) is evaluated according to (49), (50) and (53).

In terms of Lemma 1, we can conclude that

$$\begin{aligned} &E(\mathbf{w}^{k+1}) - E(\mathbf{w}^{k}) \\ &\leq -\frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2 - c)^{\alpha-1} \left(\|\Delta\mathbf{u}^{k,R}\|^2 + \|\Delta\mathbf{u}^{k,I}\|^2 \right) \\ &\quad + \left(c_5 - \frac{1}{\eta}(1-\alpha)\Gamma(1-\alpha)(c_2 - c)^{\alpha-1} \right) \sum_{l=1}^{n} \left(\|\Delta\mathbf{v}_l^{k,R}\|^2 \right. \\ &\quad \left. + \|\Delta\mathbf{v}_l^{k,I}\|^2 \right) \\ &\quad + c_6 \Bigg(\sum_{l=1}^{n} (||\Delta\mathbf{v}_l^{k,R}||^2 + ||\Delta\mathbf{v}_l^{k,I}||^2) + ||\Delta\mathbf{u}^{k,R}||^2 \\ &\quad + ||\Delta\mathbf{u}^{k,I}||^2 \Bigg) \\ &\quad + c_7 \Bigg(\sum_{l=1}^{n} (||\Delta\mathbf{v}_l^{k,R}||^2 + ||\Delta\mathbf{v}_l^{k,I}||^2) + ||\Delta\mathbf{u}^{k,R}||^2 \\ &\quad + ||\Delta\mathbf{u}^{k,I}||^2 \Bigg) \\ &\leq (c_8 - \frac{1}{\eta} c_9) \Bigg(\sum_{l=1}^{n} (||\Delta\mathbf{v}_l^{k,R}||^2 + ||\Delta\mathbf{v}_l^{k,I}||^2) + ||\Delta\mathbf{u}^{k,R}||^2 \\ &\quad + ||\Delta\mathbf{u}^{k,I}||^2 \Bigg), \end{aligned} \qquad (75)$$

where $c_8 = c_5 + c_6 + c_7$, $c_9 = (1-\alpha)\Gamma(1-\alpha)(c_2 - c)^{\alpha-1}$. If the learning rate η satisfies that $0 < \eta < \frac{c_9}{c_8}$, we get

$$E(\mathbf{w}^{k+1}) \leq E(\mathbf{w}^{k}), \qquad k = 0, 1, \cdots. \qquad (76)$$

Proof to (ii) of Theorem 1: From the above analysis of (i), we learn $E(\mathbf{w}^{k+1}) \leq E(\mathbf{w}^{k})$, along with $E(\mathbf{w}^{k}) \geq 0$. So, we derive that $E(\mathbf{w}^{k})$ is convergent. Namely, there exists $E^* \geq 0$ such that

$$\lim_{k\to\infty} E(\mathbf{w}^{k}) = E^*. \qquad (77)$$

Proof to (iii) of Theorem 1:

$$\begin{aligned} &\sum_{l=1}^{n} (||\Delta\mathbf{v}_l^{k,R}||^2 + ||\Delta\mathbf{v}_l^{k,I}||^2) \\ &\quad + ||\Delta\mathbf{u}^{k,R}||^2 + ||\Delta\mathbf{u}^{k,I}||^2 \\ &\quad = \sum_{l=1}^{n} \sum_{m=1}^{p} ((\Delta v_{lm}^{k,R})^2 + (\Delta v_{lm}^{k,I})^2) \\ &\quad + \sum_{l=1}^{n} \left((\Delta u_l^{k,R})^2 + (\Delta u_l^{k,I})^2 \right) \\ &\quad = \eta^2 \sum_{l=1}^{n} \Bigg(\sum_{m=1}^{p} \left(({}_cD^{\alpha}_{v^R_{lm}} E(\mathbf{w}^k))^2 + ({}_cD^{\alpha}_{v^I_{lm}} E(\mathbf{w}^k))^2 \right) \\ &\quad + ({}_cD^{\alpha}_{u^R_l} E(\mathbf{w}^k))^2 + ({}_cD^{\alpha}_{u^I_l} E(\mathbf{w}^k))^2 \Bigg). \end{aligned} \qquad (78)$$

So, (76) can be writen

$$
\begin{aligned}
E(\mathbf{w}^{k+1}) \leq E(\mathbf{w}^k) - \alpha\Bigg(\sum_{l=1}^{n}\sum_{m=1}^{p}\Big(({}_cD^{\alpha}_{v^R_{lm}} E(\mathbf{w}^k))^2 \\
+ ({}_cD^{\alpha}_{v^I_{lm}} E(\mathbf{w}^k))^2\Big) + \sum_{l=1}^{n}({}_cD^{\alpha}_{u^R_l} E(\mathbf{w}^k))^2 \\
+ ({}_cD^{\alpha}_{u^I_l} E(\mathbf{w}^k))^2\Bigg) \\
\cdots \\
\leq E(\mathbf{w}^0) - \alpha\sum_{q=0}^{k}\Bigg(\sum_{l=1}^{n}\sum_{m=1}^{p}\Big(({}_cD^{\alpha}_{v^R_{lm}} E(\mathbf{w}^q))^2 \\
+ ({}_cD^{\alpha}_{v^I_{lm}} E(\mathbf{w}^q))^2\Big) + \sum_{l=1}^{n}({}_cD^{\alpha}_{u^R_l} E(\mathbf{w}^q))^2 \\
+ ({}_cD^{\alpha}_{u^I_l} E(\mathbf{w}^q))^2\Bigg), \qquad (79)
\end{aligned}
$$

where $\alpha = c_8\eta^2 - c_9\eta$. Since $E(\mathbf{w}^{k+1}) \geq 0$. There holds that

$$
\begin{aligned}
\alpha\sum_{q=0}^{k}\Bigg(\sum_{l=1}^{n}\sum_{m=1}^{p}\Big(({}_cD^{\alpha}_{v^R_{lm}} E(\mathbf{w}^q))^2 + ({}_cD^{\alpha}_{v^I_{lm}} E(\mathbf{w}^q))^2\Big) \\
+ \sum_{l=1}^{n}({}_cD^{\alpha}_{u^R_l} E(\mathbf{w}^q))^2 + ({}_cD^{\alpha}_{u^I_l} E(\mathbf{w}^q))^2\Bigg) \\
\leq E(\mathbf{w}^0). \qquad (80)
\end{aligned}
$$

Let $k \to \infty$,then

$$
\begin{aligned}
\alpha\sum_{q=0}^{\infty}\Bigg(\sum_{l=1}^{n}\sum_{m=1}^{p}\Big(({}_cD^{\alpha}_{v^R_{lm}} E(\mathbf{w}^q))^2 + ({}_cD^{\alpha}_{v^I_{lm}} E(\mathbf{w}^q))^2\Big) \\
+ \sum_{l=1}^{n}({}_cD^{\alpha}_{u^R_l} E(\mathbf{w}^q))^2 + ({}_cD^{\alpha}_{u^I_l} E(\mathbf{w}^q))^2\Bigg) \\
\leq E(\mathbf{w}^0) < \infty. \qquad (81)
\end{aligned}
$$

So there holds that

$$
\begin{aligned}
\lim_{k\to\infty}\Bigg(\sum_{l=1}^{n}\sum_{m=1}^{p}\Big(({}_cD^{\alpha}_{v^R_{lm}} E(\mathbf{w}^k))^2 + ({}_cD^{\alpha}_{v^I_{lm}} E(\mathbf{w}^k))^2\Big) \\
+ \sum_{i=1}^{n}({}_cD^{\alpha}_{u^R_l} E(\mathbf{w}^k))^2 + ({}_cD^{\alpha}_{u^I_l} E(\mathbf{w}^k))^2\Bigg) = 0, \qquad (82)
\end{aligned}
$$

which implies that

$$
\lim_{k\to\infty} D^{\alpha}_{v^R_{lm}} E(\mathbf{w}^k) = 0, \quad \lim_{k\to\infty} D^{\alpha}_{v^I_{lm}} E(\mathbf{w}^k) = 0, \qquad (83)
$$

$$
\lim_{k\to\infty} D^{\alpha}_{u^R_l} E(\mathbf{w}^k) = 0, \quad \lim_{k\to\infty} D^{\alpha}_{u^I_l} E(\mathbf{w}^k) = 0, \qquad (84)
$$

where $k \in \mathbb{N}$; $l = 1, 2, \cdots, n$; $m = 1, 2, \cdots, p$. This then completes the proof of (**iii**) of **1**.

REFERENCES

[1] E. Kaslik and S. Sivasundaram, "Nonlinear dynamics and chaos in fractional-order neural networks," *Neural Netw.*, vol. 32, no. 1, pp. 245–256, Aug. 2012.

[2] M. Wang, S. Yang, S. Wu, and F. Luo, "A RBFNN approach for DoA estimation of ultra wideband antenna array," *Neurocomputing*, vol. 71, nos. 4–6, pp. 631–640, Jan. 2008.

[3] W. Wu, D.-P. Xu, and Z.-X. Li, "Convergence of gradient method for Elman networks," *Appl. Math. Mech.*, vol. 29, no. 9, pp. 1231–1238, Nov. 2008.

[4] H. Zhang, W. Wu, F. Liu, and M. Yao, "Boundedness and convergence of online gradient method with penalty for feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 6, pp. 1050–1054, Jun. 2009.

[5] W. Wu, J. Wang, M. Cheng, and Z. Li, "Convergence analysis of online gradient method for BP neural networks," *Neural Netw.*, vol. 24, no. 1, pp. 91–98, 2011.

[6] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 2101–2104, Sep. 1991.

[7] A. R. Hafiz, M. F. Amin, and K. Murase, "Using complex-valued Levenberg–Marquardt algorithm for learning and recognizing various hand gestures," in *Proc. IEEE World Congr. Comput. Intell. (WCCI)*, Jun. 2012, pp. 1–5.

[8] R. Savitha, S. Suresh, N. Sundararajan, and H. J. Kim, "A fully complex-valued radial basis function classifier for real-valued classification problems," *Neurocomputing*, vol. 78, no. 1, pp. 104–110, Feb. 2012.

[9] D. Xu, H. Zhang, and D. P. Mandic, "Convergence analysis of an augmented algorithm for fully complex-valued neural networks," *Neural Netw.*, vol. 69, pp. 44–50, Sep. 2015.

[10] B. Widrow, J. McCool, and M. Ball, "The complex LMS algorithm," *Proc. IEEE*, vol. 63, no. 4, pp. 719–720, Apr. 1975.

[11] H. Zhang, D. Xu, and Y. Zhang, "Boundedness and convergence of split-complex back-propagation algorithm with momentum and penalty," *Neural Process. Lett.*, vol. 39, no. 3, pp. 297–307, Jun. 2014.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[13] X. Wu, J. Li, and G. Chen, "Chaos in the fractional order unified system and its synchronization," *J. Franklin Inst.*, vol. 345, no. 4, pp. 392–401, Jul. 2008.

[14] C. P. Li, W. H. Deng, and D. Xu, "Chaos synchronization of the Chua system with a fractional order," *Phys. A, Statist. Mech. Appl.*, vol. 360, no. 2, pp. 171–185, Feb. 2006.

[15] H. Delavari, D. Baleanu, and J. Sadati, "Stability analysis of Caputo fractional-order nonlinear systems revisited," *Nonlinear Dyn.*, vol. 67, no. 4, pp. 2433–2439, Mar. 2012.

[16] B. N. Lundstrom, M. H. Higgs, W. J. Spain, and A. L. Fairhall, "Fractional differentiation by neocortical pyramidal neurons," *Nature Neurosci.*, vol. 11, no. 11, pp. 1335–1342, Nov. 2008.

[17] G. A. Anastassiou, "Fractional neural network approximation," *Comput. Math Appl.*, vol. 64, no. 6, pp. 1655–1676, Sep. 2012.

[18] R. Rakkiyappan, J. Cao, and G. Velmurugan, "Existence and uniform stability analysis of fractional-order complex-valued neural networks with time delays," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 84–97, Jan. 2015.

[19] H. Bao, J. H. Park, and J. Cao, "Synchronization of fractional-order complex-valued neural networks with time delay," *Neural Netw.*, vol. 81, pp. 16–28, Sep. 2016.

[20] L. Zhang, Q. Song, and Z. Zhao, "Stability analysis of fractional-order complex-valued neural networks with both leakage and discrete delays," *Appl. Math. Comput.*, vol. 298, pp. 296–309, Apr. 2017.

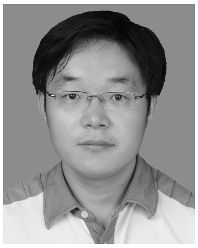
[21] G. Velmurugan, R. Rakkiyappan, V. Vembarasan, J. Cao, and A. Alsaedi, "Dissipativity and stability analysis of fractional-order complex-valued neural networks with time delay," *Neural Netw.*, vol. 86, pp. 42–53, Feb. 2017.

[22] H. Zhang, C. Zhang, and W. Wu, "Convergence of batch split-complex backpropagation algorithm for complex-valued neural networks," *Discrete Dyn. Nature Soc.*, vol. 2, 2009, Art. no. 329173.

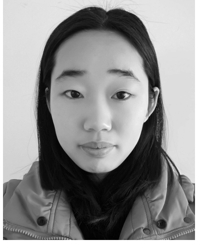
[23] D. P. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*. New York, NY, USA: Wiley, 2009.

[24] Y.-F. Pu, J.-L. Zhou, Y. Zhang, N. Zhang, G. Huang, and P. Siarry, "Fractional extreme value adaptive training method: Fractional steepest descent approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 653–662, Apr. 2015.

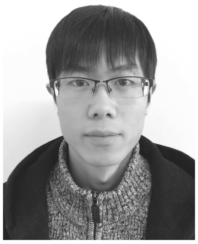
[25] H. Li, "State estimation for fractional-order complex dynamical networks with linear fractional parametric uncertainty," *Abstract Appl. Anal.*, vol. 6, 2013, Art. no. 178718.
[26] E. R. Love, "Fractional derivatives of imaginary order," *J. London Math. Soc.*, vol. s2-3, no. 2, pp. 241–259, Feb. 1971.
[27] Q. Wu and J. Huang, *Fractional Calculus*. Beijing, China: Tsinghua Univ. Press, 2016.
[28] K. B. Oldham and J. Spanier, *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*. New York, NY, USA: Academic, 1974.
[29] A. McBride, *Fractional Calculus*. New York, NY, USA: Halsted, 1986.
[30] K. Nishimoto, *Fractional Calculus: Integrations and Differentiations of Arbitrary Order*. New Haven, CT, USA: New Haven Univ. Press, 1989.



JIAN WANG received the B.S. degree in computational mathematics from the China University of Petroleum, Dongying, China, in 2002, and the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China, in 2012. From 2002 to 2006, he was an Instructor with the School of Mathematics and Computational Science, China University of Petroleum. From 2010 to 2011, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Louisville, USA. He is currently an Associate Professor with the China University of Petroleum (East China), Qingdao, China. His current research interests include machine learning, regularization theory, and neural networks.



GUOLING YANG is currently pursuing the master's degree in mathematics with the China University of Petroleum, China. Her research mainly concentrates on neural networks.



BINGJIE ZHANG is currently pursuing the master's degree with the school of mathematics major with the College of Science, China University of Petroleum, Qingdao. His research focuses on neural networks.

ZHANQUAN SUN received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences. He is an Associate Professor with the Shandong Computer Science Center. Major on big data mining, cloud computing, and intelligent transportation systems. Has presided and attended ten research projects and authored about 50 academic papers.

YUSONG LIU is currently pursuing the Ph.D. degree with the School of Mathematical Sciences, Dalian University of Technology, China. He is also a Lecturer with the College of Science, China University of Petroleum (Huadong). His research interests include neural network, shape modeling, and digital geometry processing.

JICHAO WANG was born in 1979. He received the Ph.D. degree from the Institute of Oceanology, Chinese Academy of Sciences. His current research interests include numerical simulation of ocean wave and data assimilation.