Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Fractional-order global optimal backpropagation machine trained by an improved fractional-order steepest descent method*

Yi-fei PU‡[1], Jian WANG‡[2]

[1]College of Computer Science, Sichuan University, Chengdu 610065, China
[2]College of Science, China University of Petroleum, Qingdao 266580, China
E-mail: puyifei@scu.edu.cn; wangjiannl@upc.edu.cn

Received Oct. 31, 2019; Revision accepted Jan. 13, 2020; Crosschecked Mar. 31, 2020

Abstract: We introduce the fractional-order global optimal backpropagation machine, which is trained by an improved fractional-order steepest descent method (FSDM). This is a fractional-order backpropagation neural network (FBPNN), a state-of-the-art fractional-order branch of the family of backpropagation neural networks (BPNNs), different from the majority of the previous classic first-order BPNNs which are trained by the traditional first-order steepest descent method. The reverse incremental search of the proposed FBPNN is in the negative directions of the approximate fractional-order partial derivatives of the square error. First, the theoretical concept of an FBPNN trained by an improved FSDM is described mathematically. Then, the mathematical proof of fractional-order global optimal convergence, an assumption of the structure, and fractional-order multi-scale global optimization of the FBPNN are analyzed in detail. Finally, we perform three (types of) experiments to compare the performances of an FBPNN and a classic first-order BPNN, i.e., example function approximation, fractional-order multi-scale global optimization, and comparison of global search and error fitting abilities with real data. The higher optimal search ability of an FBPNN to determine the global optimal solution is the major advantage that makes the FBPNN superior to a classic first-order BPNN.

Key words: Fractional calculus; Fractional-order backpropagation algorithm; Fractional-order steepest descent method; Mean square error; Fractional-order multi-scale global optimization
https://doi.org/10.1631/FITEE.1900593 **CLC number:** O235; N93

1 Introduction

Backpropagation, an abbreviation for "backward propagation of errors," is a common method for training artificial neural networks and is used in conjunction with an optimization method such as gradient descent (Battiti, 1992). The classic first-order backpropagation algorithm (BA) to train multi-layer networks was first described by Werbos (1974); the algorithm was presented in the context of general networks, with neural networks as a special case. It was not until the mid-1980s that the classic BA was publicized, rediscovered independently by LeCun (1985), Parker (1985), and Rumelhart et al. (1986a). The classic BA was popularized by its inclusion in the book "Parallel Distributed Processing" (Rumelhart et al., 1986b). The publication of this book triggered a significant amount of research into neural networks. Multi-layer perceptrons, trained by the classic BA, are currently the most widely used neural networks. The classic BA can refer to the result of a playout that is propagated up the search tree in a Monte Carlo tree search (Browne et al., 2012). It has been demonstrated that classic two-layer first-order backpropagation neural networks (BPNNs), with sigmoid activation functions in the hidden layer and linear transfer

‡ Corresponding authors
* Project supported by the National Key Research and Development Program of China (No. 2018YFC0830300) and the National Natural Science Foundation of China (No. 61571312)
ORCID: Yi-fei PU, https://orcid.org/0000-0003-2975-4976; Jian WANG, https://orcid.org/0000-0002-4316-932X

functions in the output layer, can approximate any function of interest to any degree of accuracy when a sufficient number of hidden units are available (Hornik et al., 1989). Research related to faster BPNN algorithms can be broadly classified into two categories. The first category involves the development of heuristic techniques. These heuristic techniques include concepts such as varying the learning rate using momentum and resealing variables (Jacobs, 1988; Vogl et al., 1988; Tollenaere, 1990; Rigler et al., 1991). The second category focuses on standard numerical optimization techniques. These techniques include concepts such as the conjugate gradient algorithm and Levenberg-Marquardt algorithm (Shanno, 1990; Barnard, 1992; Battiti, 1992; Charalambous, 1992; Hagan and Menhaj, 1994). Combinations of neural networks and evolutionary computation procedures have been widely explored. This area addresses a wide range of topics such as the cutting angle method (Andramonov et al., 1999), simulated annealing (Treadgold and Gedeon, 1998; Chuang et al., 2000; Ludermir et al., 2006), swarm algorithms (Yeh, 2013), genetic algorithms (Maniezzo, 1994; Leung et al., 2003; Nikolaev and Iba, 2003; Palmes et al., 2005), and hybrid training methods (Cantu-Paz and Kamath, 2005; Zanchettin et al., 2011), which are superior to traditional local techniques. In all these methods, the computational complexity increases rapidly with an increase in the number of variables. Moreover, the classic first-order BA-based BPNNs are easily trapped into a local optimal solution, whose optimization performance must be improved.

The application of fractional calculus to neural networks and cybernetics is an emerging discipline of research, and a small number of studies have been conducted in this area. Fractional calculus has become an important novel branch in mathematical analyses (Oldham and Spanier, 1974; Podlubny, 1998). Recently, fractional calculus has become a promising mathematical method for physical scientists and engineering technicians. Promising results and ideas have demonstrated that fractional calculus can be an interesting and useful tool in many scientific fields such as diffusion processes (Özdemir and Karadeniz, 2008), viscoelasticity theory (Koeller, 1984), fractal dynamics (Rossikhin and Shitikova, 1997), fractional control (Manabe, 2002; Podlubny et al., 2002), fractance (Elwakil, 2010), fracmemristor (Pu et al., 2018a), image processing (Pu et al., 2010, 2018b), and neural networks (Kaslik and Sivasundaram, 2011; Pu et al., 2015, 2016, 2017). Fractional calculus has been applied to neural networks and cybernetics primarily due to its inherent advantages of long-term memory, non-locality, and weak singularity, which are important properties of fractional calculus (Oldham and Spanier, 1974; Podlubny, 1998). The basic characteristic of fractional calculus is that it extends the concepts of integer-order difference and Riemann sums. The characteristics of fractional calculus are considerably different from those of classic integer-order calculus. For example, the fractional differential, except based on the Caputo definition, of a Heaviside function is nonzero, whereas its integer-order differential must be zero (Oldham and Spanier, 1974; Podlubny, 1998). Thus, the properties of the fractional-order steepest descent method (FSDM) are also different from those of the traditional first-order steepest descent method (Pu et al., 2015). For example, the FSDM can determine the fractional-order extreme points of the energy norm, which do not overlap the traditional first-order stationary points (Pu et al., 2015). It is known that classic first-order BPNNs demonstrate a tendency to be trapped into a local optimal solution.

The application of the improved FSDM to training BPNNs has the potential to overcome such deficits. Therefore, to improve the optimization performance of classic first-order BPNNs, in this study we investigate whether it is possible to apply the improved FSDM to generalize classic first-order BPNNs to the fractional-order backpropagation neural networks (FBPNNs). Based on this inspiration, we introduce an FBPNN that is trained by an improved FSDM, whose reverse incremental search is in the negative directions of the approximate fractional-order partial derivatives of the square error during the iterative search process. This introduced fractional-order branch of the family of BPNNs, which differs from the majority of the previous classic first-order BPNNs and as such represents an interesting theoretical contribution. The FBPNN and the first-order BPNN are trained by the improved FSDM and the traditional first-order steepest descent method, respectively. The higher optimal search ability of an FBPNN to determine the global optimal solution is

the major advantage that makes the FBPNN superior to a classic first-order BPNN.

2 Background of fractional calculus and FSDM

In this section, we briefly introduce the necessary mathematical background of fractional calculus and FSDM.

The Grünwald-Letnikov definition of fractional calculus for a causal differintegrable function $f(x)$ can be represented in a convenient form as follows (Oldham and Spanier, 1974; Podlubny, 1998):

$$ {}_{a}^{\mathrm{G\text{-}L}}D_x^v f(x)=\lim_{N\to\infty}\left\{\frac{\left(\frac{x-a}{N}\right)^{-v}}{\Gamma(-v)}\sum_{k=0}^{N-1}\frac{\Gamma(k-v)}{\Gamma(k+1)}f\left[x-k\left(\frac{x-a}{N}\right)\right]\right\}, \tag{1}$$

where $[a, x]$ is the duration of $f(x)$, N is the number of partitions of the duration, v is an arbitrary real number, $\Gamma(\alpha)=\int_0^\infty \mathrm{e}^{-x}x^{\alpha-1}\mathrm{d}x$ is the Gamma function, and ${}_{a}^{\mathrm{G\text{-}L}}D_x^v$ denotes the Grünwald-Letnikov defined fractional differential operator. In this study, we use the equivalent notations $D_x^v={}_{0}^{\mathrm{G\text{-}L}}D_x^v$ in an interchangeable manner.

The reverse incremental search of the FSDM is in the negative direction of the v-order fractional derivative of the quadratic energy norm E, which can be represented as (Pu et al., 2015)

$$\xi_{k+1}=\xi_k-\mu\left(D_{\xi_k}^v E\right), \tag{2}$$

where k is the step size or the number of iterations, ξ is the independent variable of E, and μ is the constant coefficient that controls the stability and the rate of convergence of the FSDM.

3 Fractional-order backpropagation neural networks

3.1 Fractional-order partial derivatives of the square error

In this subsection, to achieve the improved FSDM for training the FBPNNs, the fractional-order partial derivatives of the square error of a BPNN should be described first.

Fig. 1 displays the model of a BPNN, which is represented by abbreviated symbols denoting its three layers.

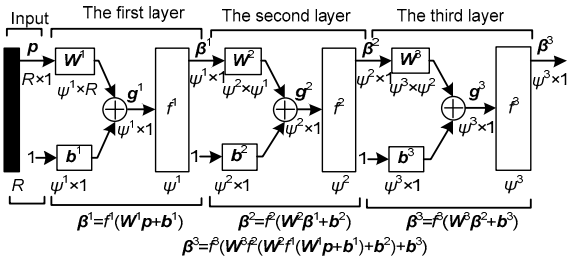

Fig. 1 A backpropagation neural network (BPNN) model with abbreviated notations (the superscript denotes the layer number of a BPNN)

In Fig. 1, the superscript denotes the layer number of a BPNN. $\boldsymbol{p}\in\mathbb{R}^{R\times 1}$ is the input matrix. $\boldsymbol{W}^1\in\mathbb{R}^{\psi^1\times R}$, $\boldsymbol{W}^2\in\mathbb{R}^{\psi^2\times\psi^1}$, and $\boldsymbol{W}^3\in\mathbb{R}^{\psi^3\times\psi^2}$ are the weight matrices of its first, second, and third layers, respectively. $\boldsymbol{b}^1\in\mathbb{R}^{\psi^1\times 1}$, $\boldsymbol{b}^2\in\mathbb{R}^{\psi^2\times 1}$, and $\boldsymbol{b}^3\in\mathbb{R}^{\psi^3\times 1}$ are the bias matrices of its first, second, and third layers, respectively. f^1, f^2, and f^3 are the activation functions of its first, second, and third layers, respectively. $\boldsymbol{g}^1\in\mathbb{R}^{\psi^1\times 1}$, $\boldsymbol{g}^2\in\mathbb{R}^{\psi^2\times 1}$, and $\boldsymbol{g}^3\in\mathbb{R}^{\psi^3\times 1}$ are the net input matrices of f^1, f^2, and f^3, respectively. $\boldsymbol{\beta}^1\in\mathbb{R}^{\psi^1\times 1}$, $\boldsymbol{\beta}^2\in\mathbb{R}^{\psi^2\times 1}$, and $\boldsymbol{\beta}^3\in\mathbb{R}^{\psi^3\times 1}$ are the output matrices of the first, second, and third layers, respectively. In a multi-layer BPNN, the output of its upper layer is the input of the next layer. Thus, it follows that

$$\begin{aligned}\boldsymbol{\beta}^{m+1}&=f^{m+1}\left(\boldsymbol{g}^{m+1}\right)\\&=f^{m+1}\left(\boldsymbol{W}^{m+1}\boldsymbol{\beta}^m+\boldsymbol{b}^{m+1}\right),\ m=0,\ 1,\ \ldots,\ M-1,\end{aligned} \tag{3}$$

where M is the number of layers of a BPNN, $\boldsymbol{g}^{m+1}=\boldsymbol{W}^{m+1}\boldsymbol{\beta}^m+\boldsymbol{b}^{m+1}$, $\boldsymbol{\beta}^0=\boldsymbol{p}$, and $\boldsymbol{\beta}=\boldsymbol{\beta}^M$ represents the actual output of the entire BPNN. Assume that the sample sets of the target output matrix $\boldsymbol{q}_\eta$ corresponding to the actual input matrix $\boldsymbol{p}_\eta$ of a BPNN are $\{\boldsymbol{p}_1, \boldsymbol{q}_1\}$, $\{\boldsymbol{p}_2, \boldsymbol{q}_2\}$, …, $\{\boldsymbol{p}_\eta, \boldsymbol{q}_\eta\}$. Then, the mean square error of a BPNN is as follows:

$$F(\boldsymbol{x})=E\left(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}\right)=E\left[(\boldsymbol{q}-\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{q}-\boldsymbol{\beta})\right], \tag{4}$$

where $\boldsymbol{x}$ is the weight and bias vector and $\boldsymbol{e}=[e_i]$ is the error vector of a BPNN. A BPNN uses the iterative approximate calculation of variance as follows:

$$\begin{aligned}\hat{F}(k)&=\left[\boldsymbol{q}(k)-\boldsymbol{\beta}(k)\right]^{\mathrm{T}}\left[\boldsymbol{q}(k)-\boldsymbol{\beta}(k)\right]\\&=\left[\boldsymbol{e}(k)\right]^{\mathrm{T}}\boldsymbol{e}(k),\end{aligned}\tag{5}$$

where k is the iteration index and $\hat{F}(k)=\sum_{i=1}^{\psi^M}\left(q_i-\beta_i\right)^2$. Eq. (5) indicates that the mathematical expectation of the mean square error F is replaced by a quadratic energy norm $\hat{F}(k)$, the square error at iteration k of the iterative search process, of a BPNN.

In addition, $\sum_{m=0}^{\infty}\sum_{n=0}^{m}\equiv\sum_{n=0}^{\infty}\sum_{m=n}^{\infty}$ and $\binom{v}{r+n}\binom{r+n}{n}\equiv\binom{v}{n}\binom{v-n}{r}$ can be derived, where v is a fraction and $\binom{v}{n}=\frac{(-1)^n\Gamma(n-v)}{\Gamma(-v)\Gamma(1+n)}=\frac{\Gamma(1+v)}{\Gamma(1-n+v)\Gamma(1+n)}$. Thus, from Eq. (1), it follows that

$$D_{x-a}^{v}\left(\hat{F}g_i^m\right)=\sum_{n=0}^{\infty}\left[\binom{v}{n}\left(D_{x-a}^{v-n}\hat{F}\right)\left(D_{x-a}^{n}g_i^m\right)\right],\tag{6}$$

where x is an independent variable, g_i^m is the i^{th} net input of f^m, and a is a constant. When $n>v$, $\binom{v}{n}=\frac{(-1)^n\Gamma(n-v)}{\Gamma(-v)\Gamma(n+1)}=\frac{\Gamma(1+v)}{\Gamma(1-n+v)\Gamma(1+n)}\neq 0$ (Oldham and Spanier, 1974). Thus, from Eq. (6) and Faà di Bruno's formula, the fractional-order partial derivatives of the square error of a BPNN can be derived as

$$\begin{aligned}&D_{\left(w_{i,j}^m-w_{\inf}\right)}^{v}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]=D_{w_{i,j}^m}^{v}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]\\&=\frac{\left(w_{i,j}^m-w_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(w_{i,j}^m-w_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}n!\\&\cdot\sum_{h=1}^{n}\left(D_{g_i^m}^{h}\hat{F}\right)\sum\prod_{k=1}^{n}\frac{1}{P_k!}\left(\frac{D_{w_{i,j}^m}^{k}g_i^m}{k!}\right)^{P_k},\end{aligned}\tag{7}$$

$$\begin{aligned}&D_{\left(b_i^m-b_{\inf}\right)}^{v}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]=D_{b_i^m}^{v}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]\\&=\frac{\left(b_i^m-b_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(b_i^m-b_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}n!\\&\cdot\sum_{h=1}^{n}\left(D_{g_i^m}^{h}\hat{F}\right)\sum\prod_{k=1}^{n}\frac{1}{P_k!}\left(\frac{D_{b_i^m}^{k}g_i^m}{k!}\right)^{P_k},\end{aligned}\tag{8}$$

where $w_{i,j}^m\in\left(w_{\inf},w_{\sup}\right)$ and $b_i^m\in\left(b_{\inf},b_{\sup}\right)$ are the domains of definitions of $w_{i,j}^m$ and b_i^m, respectively. $\hat{F}=\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]$ is a composite function, $D_{g_i^m}^{h}$, $D_{w_{i,j}^m}^{k}$, and $D_{b_i^m}^{k}$ are the integer-order differential operators, and the nonnegative integer P_k satisfies

$$\begin{cases}\sum_{k=1}^{n}kP_k=n,\\\sum_{k=1}^{n}P_k=h.\end{cases}\tag{9}$$

The third summation notations "Σ" in Eqs. (7) and (8) denote the summation of the corresponding $\left\{\prod_{k=1}^{n}\left(D_{w_{i,j}^m}^{k}g_i^m/k!\right)^{P_k}/P_k!\right\}\Big|_{h=1\to n}$ and $\left\{\prod_{k=1}^{n}\left(D_{b_i^m}^{k}g_i^m/k!\right)^{P_k}/P_k!\right\}\Big|_{h=1\to n}$ of all the combinations of $P_k|_{h=1\to n}$ that satisfy Eq. (9).

From Fig. 1, we can observe that $\boldsymbol{g}^{m+1}=\boldsymbol{W}^{m+1}\boldsymbol{\beta}^m+\boldsymbol{b}^{m+1}$. Thus, the net input g_i^m of f^m is as follows:

$$g_i^m\left(w_{i,j}^m,b_i^m\right)=\sum_{j=1}^{\psi^{m-1}}w_{i,j}^m\beta_j^{m-1}+b_i^m.\tag{10}$$

From Eq. (10), it follows that

$$D_{w_{i,j}^m}^{1}g_i^m\left(w_{i,j}^m,b_i^m\right)=\beta_j^{m-1},\tag{11}$$

$$D_{b_i^m}^{1}g_i^m\left(w_{i,j}^m,b_i^m\right)=1.\tag{12}$$

From Eqs. (11) and (12), it follows that

$$D_{w_{i,j}^m}^{k>1}g_i^m\left(w_{i,j}^m,b_i^m\right)\equiv 0,\tag{13}$$

$$D_{b_i^m}^{k>1}g_i^m\left(w_{i,j}^m,b_i^m\right)\equiv 0.\tag{14}$$

Using mathematical induction, from Eq. (9), we can derive the following:

(1) When h=1, we have

$$P_i=\begin{cases}0, & i=1,2,...,n-1,\\ 1, & i=n.\end{cases}$$

(2) When h=2, 3, …, n−1, we have

$$P_i=\begin{cases}0, & i=1,2,...,n, i\neq j, n-j,\\ 1, & i=j, n-j,\\ & j=1,2,...,n-1.\end{cases}$$

(3) When h=n, we have

$$P_i=\begin{cases}n, & i=1,\\ 0, & i=2,3,...,n.\end{cases}$$

Thus, from Eqs. (7) and (8), if and only if

$$\begin{cases}D^1_{w_{i,j}^m} g_i^m\left(w_{i,j}^m,b_i^m\right)\neq 0,\\ D^1_{b_i^m} g_i^m\left(w_{i,j}^m,b_i^m\right)\neq 0,\\ D^{k>1}_{w_{i,j}^m} g_i^m\left(w_{i,j}^m,b_i^m\right)\equiv 0,\\ D^{k>1}_{b_i^m} g_i^m\left(w_{i,j}^m,b_i^m\right)\equiv 0,\end{cases}$$

and only when h=n, can we obtain

$$\begin{cases}\sum\prod_{k=1}^{n}\left(D^k_{w_{i,j}^m} g_i^m/k!\right)^{P_k}\Big/P_k!\neq 0,\\ \sum\prod_{k=1}^{n}\left(D^k_{b_i^m} g_i^m/k!\right)^{P_k}\Big/P_k!\neq 0.\end{cases}$$

Thus, from Eqs. (11)–(14), Eqs. (7) and (8) can be simplified as follows:

$$D^v_{w_{i,j}^m}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]=\frac{\left(w_{i,j}^m-w_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(w_{i,j}^m-w_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}n!\left(D^h_{g_i^m}\hat{F}\right)\frac{1}{P_1!}\left(D^1_{w_{i,j}^m} g_i^m\right)^{P_1}\Bigg|_{h=n},\tag{15}$$

$$D^v_{b_i^m}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]=\frac{\left(b_i^m-b_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(b_i^m-b_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}n!\left(D^h_{g_i^m}\hat{F}\right)\frac{1}{P_1!}\left(D^1_{b_i^m} g_i^m\right)^{P_1}\Bigg|_{h=n}.\tag{16}$$

Recall that from Eq. (9), when h=n,

$$\begin{cases}P_1=n,\\ P_2=P_3=\cdots=P_{n-1}=P_n=0.\end{cases}\tag{17}$$

Thus, from Eqs. (11), (12), and (17), Eqs. (15) and (16) can be further simplified as follows:

$$\begin{aligned}D^v_{w_{i,j}^m}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]&=\frac{\left(w_{i,j}^m-w_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(w_{i,j}^m-w_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}\left(D^n_{g_i^m}\hat{F}\right)\left(D^1_{w_{i,j}^m} g_i^m\right)^n\\ &=\frac{\left(w_{i,j}^m-w_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(w_{i,j}^m-w_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}\cdot\left(D^n_{g_i^m}\hat{F}\right)\left(\beta_j^{m-1}\right)^n,\end{aligned}\tag{18}$$

$$\begin{aligned}D^v_{b_i^m}\hat{F}\left[g_i^m\left(w_{i,j}^m,b_i^m\right)\right]&=\frac{\left(b_i^m-b_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(b_i^m-b_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}\left(D^n_{g_i^m}\hat{F}\right)\left(D^1_{b_i^m} g_i^m\right)^n\\ &=\frac{\left(b_i^m-b_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}+\sum_{n=1}^{\infty}\binom{v}{n}\frac{\left(b_i^m-b_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}\left(D^n_{g_i^m}\hat{F}\right).\end{aligned}\tag{19}$$

Note that first, Eqs. (18) and (19) are the chain rule of the fractional derivatives of the square error (a composite function) of a BPNN (Oldham and Spanier, 1974), which gives an infinite series that offers the minimal expectation of being expressible in the closed form, except for trivially simple instances of functions $\hat{F}$ and g_i^m. When Eqs. (18) and (19) are with zero initial conditions, they continue to hold (Oldham and Spanier, 1974). Second, the nonzero initial value problem is a fundamental issue of the application of fractional calculus. The practical applicability of Eqs. (18) and (19) is limited by the absence of the physical interpretation of the limit values of fractional derivatives at the lower bound $w_{i,j}^m=w_{\inf}$ and $b_i^m=b_{\inf}$, respectively. To date, such an interpretation was partially solved by Heymans and Podlubny (2006) and Petráš (2011). Finally, the

summation notations in Eqs. (18) and (19) denote that both $D^{v}_{w^{m}_{i,j}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$ and $D^{v}_{b^{m}_{i}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$ express the long-term memory and non-locality of the fractional differential of the square error $\hat{F}$ of a BPNN.

3.2 Achievement of improved FSDM and FBPNNs trained by an improved FSDM

In this subsection, the achievement of the improved FSDM and the FBPNNs trained by the improved FSDM is further discussed.

To simplify the infinite series calculation of the fractional-order partial derivative of the square error of a BPNN in Eqs. (18) and (19), the approximate fractional-order partial derivatives of the square error of a BPNN, $\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$ and $\tilde{D}^{v}_{b^{m}_{i}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$, are suggested to merely take the first four terms of Eqs. (18) and (19) into account, which can be given as

$$\begin{aligned}\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]&=\frac{\left(w^{m}_{i,j}-w_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}\\&+\sum_{n=1}^{3}\binom{v}{n}\frac{\left(w^{m}_{i,j}-w_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}\left(D^{n}_{g^{m}_{i}}\hat{F}\right)\left(\beta^{m-1}_{j}\right)^{n},\end{aligned}\tag{20}$$

$$\begin{aligned}\tilde{D}^{v}_{b^{m}_{i}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]&=\frac{\left(b^{m}_{i}-b_{\inf}\right)^{-v}}{\Gamma(1-v)}\hat{F}\\&+\sum_{n=1}^{3}\binom{v}{n}\frac{\left(b^{m}_{i}-b_{\inf}\right)^{n-v}}{\Gamma(n-v+1)}\left(D^{n}_{g^{m}_{i}}\hat{F}\right).\end{aligned}\tag{21}$$

Eqs. (20) and (21) indicate that although only the first four terms of $D^{v}_{w^{m}_{i,j}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$ and $D^{v}_{b^{m}_{i}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$ are taken into account, $\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$ and $\tilde{D}^{v}_{b^{m}_{i}}\hat{F}\left[g^{m}_{i}\left(w^{m}_{i,j},b^{m}_{i}\right)\right]$ are essentially the approximate fractional differentials of the square error $\hat{F}$ of a BPNN, preserving the strengths of fractional calculus such as long-term memory, non-locality, and weak singularity. Therefore, similar to the FSDM, from Eqs. (2), (20), and (21), the improved FSDM for the family of the BPNNs can be implemented as follows:

$$w^{m}_{i,j}(k+1)=w^{m}_{i,j}(k)-\mu\left[\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)\right],\tag{22}$$

$$b^{m}_{i}(k+1)=b^{m}_{i}(k)-\mu\left[\tilde{D}^{v}_{b^{m}_{i}}\hat{F}(k)\right],\tag{23}$$

where $w^{m}_{i,j}$ is the weight between the j^{th} output of the $(m-1)^{\text{th}}$ layer and the i^{th} input of the m^{th} layer, b^{m}_{i} is the i^{th} bias of the m^{th} layer, and μ is the learning rate of an improved FSDM (a small positive number). Note that in Eqs. (22) and (23), there may exist a point $\left(w^{m^{o}}_{i,j},b^{m^{o}}_{i}\right)$ that exactly satisfies $\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)=0$ and $\tilde{D}^{v}_{b^{m}_{i}}\hat{F}(k)=0$, but keeps $\hat{F}(k)\neq 0$. Even if $\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)=0$ and $\tilde{D}^{v}_{b^{m}_{i}}\hat{F}(k)=0$, but $\hat{F}(k)\neq 0$, the corresponding point $\left(w^{m^{o}}_{i,j},b^{m^{o}}_{i}\right)$ is merely a saddle point, rather than a real fractional-order optimal minimum point of an improved FSDM. In Eqs. (22) and (23), on a saddle point $\left(w^{m^{o}}_{i,j},b^{m^{o}}_{i}\right)$, we should force the search process to continue by adding small perturbations. Eqs. (22) and (23) can be expressed in the vector form, $\boldsymbol{W}^{m}(k+1)=\boldsymbol{W}^{m}(k)-\mu\tilde{\boldsymbol{D}}^{v}_{w^{m}}\hat{F}$ and $\boldsymbol{b}^{m}(k+1)=\boldsymbol{b}^{m}(k)-\mu\tilde{\boldsymbol{D}}^{v}_{b^{m}}\hat{F}$, where $\tilde{\boldsymbol{D}}^{v}_{w^{m}}\hat{F}=\left[\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}\right]_{\psi^{m}\times\psi^{m-1}}$ and $\tilde{\boldsymbol{D}}^{v}_{b^{m}}\hat{F}=\left[\tilde{D}^{v}_{b^{m}_{i}}\hat{F}\right]_{\psi^{m}\times 1}$. We define n-order sensibility, ${}_{n}\rho^{m}_{i}(k)$, at iteration k of the iterative search process of the improved FSDM for the family of the BPNNs as follows:

$${}_{n}\rho^{m}_{i}(k)=D^{n}_{g^{m}_{i}}\hat{F}(k).\tag{24}$$

Thus, from Eq. (24), Eqs. (20) and (21) can be rewritten as Eqs. (25) and (26), respectively. Thus, from Eqs. (22), (23), (25), and (26), we have Eqs. (27) and (28).

$$\begin{aligned}\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)&=\frac{\left[w^{m}_{i,j}(k)-w_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)\\&+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w^{m}_{i,j}(k)-w_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}\,{}_{n}\rho^{m}_{i}(k)\left(\beta^{m-1}_{j}\right)^{n},\end{aligned}\tag{25}$$

$$\tilde{D}^{v}_{b_i^m}\hat{F}(k)=\frac{\left[b_i^m(k)-b_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}{}_n\rho_i^m(k). \tag{26}$$

$$w_{i,j}^m(k+1)=w_{i,j}^m(k)-\mu\left(\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right), \tag{27}$$

$$b_i^m(k+1)=b_i^m(k)-\mu\left(\frac{\left[b_i^m(k)-b_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\right). \tag{28}$$

Eqs. (27) and (28) and Fig. 1 indicate that by employing the improved FSDM to achieve the training process for the family of the BPNNs, a BPNN is indeed an FBPNN. The architecture of an FBPNN is identical to that of a traditional first-order BPNN; however, the reverse incremental searches of an FBPNN that is trained by an improved FSDM are in the negative directions of the approximate v-order fractional derivatives of quadratic energy norm $\hat{F}(k)$. As we know, the classic first-order BA for the approximate mean square error of a first-order BPNN is as follows (Werbos, 1974; LeCun, 1985; Parker, 1985; Rumelhart et al., 1986a, 1986b; Battiti, 1992):

$$\begin{aligned}w_{i,j}^m(k+1)&=w_{i,j}^m(k)-\mu\left[D^1_{w_{i,j}^m}\hat{F}(k)\right]\\&=w_{i,j}^m(k)-\mu_1\rho_i^m(k)\beta_j^{m-1},\end{aligned} \tag{29}$$

$$\begin{aligned}b_i^m(k+1)&=b_i^m(k)-\mu\left[D^1_{b_i^m}\hat{F}(k)\right]\\&=b_i^m(k)-\mu_1\rho_i^m(k),\end{aligned} \tag{30}$$

where $D^1_{w_{i,j}^m}\hat{F}(k)={}_1\rho_i^m(k)\beta_j^{m-1}=\left[D^1_{g_i^m}\hat{F}(k)\right]\beta_j^{m-1}$ and $D^1_{b_i^m}\hat{F}(k)={}_1\rho_i^m(k)=D^1_{g_i^m}\hat{F}(k)$. The reverse incremental search of a classic first-order BPNN is in the negative direction of the first-order derivative of the square error $\hat{F}$. Comparing Eqs. (27) and (28) with Eqs. (29) and (30), we can observe that when v=1, an FBPNN converts to a classic first-order BPNN. The first-order BPNN is a special case of an FBPNN. The classic first-order optimal minimum point is a particular case of the fractional-order minimum one (Pu et al., 2015).

3.3 Fractional-order global optimal convergence of an improved FSDM-based FBPNN

In this subsection, the mathematical proof of the fractional-order global optimal convergence of an FBPNN that is trained by an improved FSDM is presented.

Assume that the square error $\hat{F}(k)$ of an FBPNN is a smooth convex function with at least one of the equivalent fractional-order optimal extreme points $\left(w_{i,j}^{m^*},b_i^{m^*}\right)$. Each step of the iterative search processes of an FBPNN that is trained by an improved FSDM is formulated as Eqs. (22), (23), (27), and (28). The fingerprint of a fractional-order optimal minimum point is that its $\tilde{D}^{v}_{w_{i,j}^m}\hat{F}(k)$ and $\tilde{D}^{v}_{b_i^m}\hat{F}(k)$ are equal to zero. Therefore, we can obtain the following lemma:

Lemma 1 If the number of neurons and the number of hidden layers of an FBPNN that is trained by an improved FSDM are such that at least one fractional-order minimum point of the square error $\hat{F}(k)$ exists, and at the same time if on a saddle point $\left(w_{i,j}^{m^o},b_i^{m^o}\right)$ (even if $\tilde{D}^{v}_{w_{i,j}^m}\hat{F}(k)=0$ and $\tilde{D}^{v}_{b_i^m}\hat{F}(k)=0$, but $\hat{F}(k)\neq 0$), the iterative search processes are constrained artificially to keep going on, then the improved FSDM described in Eqs. (22), (23), (27), and (28) can guarantee that this FBPNN converges to a fractional-order optimal minimum point $\left(w_{i,j}^{m^*},b_i^{m^*}\right)$, a global minimum point, of the square error $\hat{F}(k)$.

Proof Eqs. (22), (23), and (25)–(28) indicate that on a fractional-order optimal minimum point $\left(w_{i,j}^{m^*},b_i^{m^*}\right)$ of the square error $\hat{F}(k)$, one can obtain

$\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)=0$ and $\tilde{D}^{v}_{b^{m}_{i}}\hat{F}(k)=0$. Thus, the criteria for the convergence of Eqs. (27) and (28) are as follows:

$$\begin{cases}\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)=\dfrac{\left[w^{m}_{i,j}(k)-w_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)\\ \quad+\displaystyle\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w^{m}_{i,j}(k)-w_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}\,{}_{n}\rho^{m}_{i}(k)\left(\beta^{m-1}_{j}\right)^{n}=0,\\ \tilde{D}^{v}_{b^{m}_{i}}\hat{F}(k)=\dfrac{\left[b^{m}_{i}(k)-b_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)\\ \quad+\displaystyle\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b^{m}_{i}(k)-b_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}\,{}_{n}\rho^{m}_{i}(k)=0.\end{cases}\tag{31}$$

Eq. (31) gives the criteria for the convergence of Eqs. (22), (23), (27), and (28). Note that in Eqs. (22) and (23), even if $\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)=0$ and $\tilde{D}^{v}_{b^{m}_{i}}\hat{F}(k)=0$, but $\hat{F}(k)\neq 0$, the iterative search processes are constrained to keep going on. Therefore, when $\tilde{D}^{v}_{w^{m}_{i,j}}\hat{F}(k)=0$ and $\tilde{D}^{v}_{b^{m}_{i}}\hat{F}(k)=0$, but $\hat{F}(k)\neq 0$, the corresponding point $\left(w^{m^{o}}_{i,j},b^{m^{o}}_{i}\right)$ is merely a saddle point, rather than a real fractional-order optimal minimum point $\left(w^{m^{*}}_{i,j},b^{m^{*}}_{i}\right)$ of an improved FSDM. The solution to Eq. (31) depends on both the square error $\hat{F}(k)$ and the n-order sensibility ${}_{n}\rho^{m}_{i}(k)$ at iteration k of the iterative search process of an FBPNN.

For an FBPNN that is trained by an improved FSDM, because the bias $\boldsymbol{b}^{m+1}$ ($\boldsymbol{b}^{m+1}\in\mathbb{R}^{\psi^{m+1}\times 1}$) of the $(m+1)^{\text{th}}$ layer is not related to the net input $\boldsymbol{g}^{m}$ ($\boldsymbol{g}^{m}\in\mathbb{R}^{\psi^{m}\times 1}$) of the m^{th} layer, from Eqs. (3) and (10), we can derive that

$$\begin{aligned}D^{1}_{g^{m}_{j}}g^{m+1}_{i}&=w^{m+1}_{i,j}\left(D^{1}_{g^{m}_{j}}\beta^{m}_{j}\right)=w^{m+1}_{i,j}\left[D^{1}_{g^{m}_{j}}f^{m}\left(g^{m}_{j}\right)\right]\\&=w^{m+1}_{i,j}\left[D^{1}_{g^{m}_{j}}f^{m}\left(\sum_{k=1}^{\psi^{m-1}}w^{m}_{j,k}\beta^{m-1}_{k}+b^{m}_{j}\right)\right].\end{aligned}\tag{32}$$

Thus, from Eq. (32), it follows that

$$\begin{aligned}{}_{2}\rho^{m}_{i}(k)&=\left(D^{1}_{g^{m}_{j}}g^{m+1}_{i}\right)\left[D^{1}_{g^{m+1}_{i}}\left(D^{1}_{g^{m}_{i}}\hat{F}(k)\right)\right]\\&=\left(D^{1}_{g^{m}_{j}}g^{m+1}_{i}\right)\left[D^{1}_{g^{m+1}_{i}}\left(\left(D^{1}_{g^{m}_{j}}g^{m+1}_{i}\right)\left(D^{1}_{g^{m+1}_{i}}\hat{F}(k)\right)\right)\right]\\&=\left(D^{1}_{g^{m}_{j}}g^{m+1}_{i}\right)^{2}\left(D^{2}_{g^{m+1}_{i}}\hat{F}(k)\right)\\&=\left\{w^{m+1}_{i,j}\left[D^{1}_{g^{m}_{j}}f^{m}\left(\sum_{k=1}^{\psi^{m-1}}w^{m}_{j,k}\beta^{m-1}_{k}+b^{m}_{j}\right)\right]\right\}^{2}{}_{2}\rho^{m+1}_{i}(k),\end{aligned}\tag{33}$$

which is closely related to the activation function of the m^{th} layer of an FBPNN, f^{m}. Similar to Eq. (33), from Eq. (24) and Fig. 1, we can derive that

$$\begin{aligned}{}_{n}\rho^{m}_{i}(k)&=\left(D^{1}_{g^{m}_{j}}g^{m+1}_{i}\right)^{n}\left(D^{n}_{g^{m+1}_{i}}\hat{F}(k)\right)\\&=\left\{w^{m+1}_{i,j}\left[D^{1}_{g^{m}_{j}}f^{m}\left(\sum_{k=1}^{\psi^{m-1}}w^{m}_{j,k}\beta^{m-1}_{k}+b^{m}_{j}\right)\right]\right\}^{n}{}_{n}\rho^{m+1}_{i}(k).\end{aligned}\tag{34}$$

Eq. (34) indicates that the sensibility backpropagates from the last layer to the first layer of the improved FSDM-based FBPNN. In particular, with regard to the M^{th} layer, the last layer of an FBPNN, from Eqs. (3), (5), and (24), we can derive that

$$\begin{aligned}{}_{1}\rho^{M}_{i}(k)&=D^{1}_{g^{M}_{i}}\hat{F}(k)=D^{1}_{g^{M}_{i}}\left[\sum_{j=1}^{\psi^{M}}\left(q_{j}-\beta_{j}\right)^{2}\right]\\&=-2\left(q_{i}-\beta_{i}\right)\left(D^{1}_{g^{M}_{i}}\beta_{i}\right),\end{aligned}\tag{35}$$

$$D^{1}_{g^{M}_{i}}\beta_{i}=D^{1}_{g^{M}_{i}}\beta^{M}_{i}=D^{1}_{g^{M}_{i}}f^{M}\left(g^{M}_{i}\right).\tag{36}$$

Substituting Eq. (36) into Eq. (35), we obtain

$$\begin{aligned}{}_{1}\rho^{M}_{i}(k)&=-2\left(q_{i}-\beta_{i}\right)\left(D^{1}_{g^{M}_{i}}\beta_{i}\right)\\&=-2\left(q_{i}-\beta_{i}\right)\left[D^{1}_{g^{M}_{i}}f^{M}\left(g^{M}_{i}\right)\right].\end{aligned}\tag{37}$$

From Eqs. (24), (35), and (37), we have

$$\begin{aligned}{}_{2}\rho^{M}_{i}(k)&=D^{1}_{g^{M}_{i}}\left[{}_{1}\rho^{M}_{i}(k)\right]\\&=-2\left[\left(q_{i}-\beta_{i}\right)\left(D^{2}_{g^{M}_{i}}\beta_{i}\right)-\left(D^{1}_{g^{M}_{i}}\beta_{i}\right)^{2}\right],\end{aligned}\tag{38}$$

$$
\begin{aligned}
&{}_3\rho_i^M(k)=D^1_{g_i^M}\left[{}_2\rho_i^M(k)\right]\\
&=-2\left[\left(q_i-\beta_i\right)\left(D^3_{g_i^M}\beta_i\right)-3\left(D^1_{g_i^M}\beta_i\right)\left(D^2_{g_i^M}\beta_i\right)\right].
\end{aligned}
\tag{39}
$$

Eqs. (37)–(39) are the initial point values of the backpropagation recurrence relations, which can be expressed by Eq. (34).

In addition, if $\tilde{D}^v_{w^m_{i,j}}\hat{F}(k)=0$ and $\tilde{D}^v_{b^m_i}\hat{F}(k)=0$, but $\hat{F}(k)\neq 0$, as long as the iterative search processes are constrained to keep going on, when the iterative search processes converge to a fractional-order optimal minimum point $\left(w^{m^*}_{i,j},b^{m^*}_i\right)$ of the square error $\hat{F}(k)$, with respect to the M^{th} layer (the last layer) of an FBPNN, to enable Eq. (31) to be set up, from Eqs. (37)–(39), except for saddle points $\left(w^{m^o}_{i,j},b^{m^o}_i\right)$, a necessary and sufficient condition can be given as

$$
\begin{cases}
\hat{F}(k)=\sum\limits_{i=1}^{\psi^M}\left(q_i-\beta_i\right)^2=0,\\
{}_1\rho_i^M(k)=-2\left(q_i-\beta_i\right)\left(D^1_{g_i^M}\beta_i\right)=0,\\
{}_2\rho_i^M(k)=-2\left[\left(q_i-\beta_i\right)\left(D^2_{g_i^M}\beta_i\right)-\left(D^1_{g_i^M}\beta_i\right)^2\right]=0,\\
{}_3\rho_i^M(k)=-2\left[\left(q_i-\beta_i\right)\left(D^3_{g_i^M}\beta_i\right)-3\left(D^1_{g_i^M}\beta_i\right)\left(D^2_{g_i^M}\beta_i\right)\right]=0,
\end{cases}
\tag{40}
$$

where $\beta_i=\beta_i^M$ and i=1, 2, …, ψ^M. To enable Eq. (40) to be set up, a necessary and sufficient condition can be given as

$$
\begin{cases}
q_i-\beta_i=0,\\
D^1_{g_i^M}\beta_i=0.
\end{cases}
\tag{41}
$$

Furthermore, when Eq. (41) is set up, we have

$$
\begin{cases}
\sum\limits_{i=1}^{\psi^M}\left(q_i-\beta_i\right)^2=0,\\
-2\left(q_i-\beta_i\right)\left(D^1_{g_i^M}\beta_i\right)=0.
\end{cases}
\tag{42}
$$

Therefore, when the iterative search processes converge to a fractional-order optimal minimum point $\left(w^{m^*}_{i,j},b^{m^*}_i\right)$ of the square error $\hat{F}(k)$ of an FBPNN, from Eqs. (5), (35), and (42), we obtain

$$
\begin{cases}
\hat{F}(k)=\sum\limits_{i=1}^{\psi^M}\left(q_i-\beta_i\right)^2=0,\\
{}_1\rho_i^M(k)=D^1_{g_i^M}\hat{F}(k)=-2\left(q_i-\beta_i\right)\left(D^1_{g_i^M}\beta_i\right)=0.
\end{cases}
\tag{43}
$$

Thus, from Eq. (34), when Eq. (43) is set up, we have

$$
\begin{cases}
\hat{F}(k)=0,\\
{}_1\rho_i^m(k)=D^1_{g_i^m}\hat{F}(k)=0,
\end{cases}
\tag{44}
$$

where m=0, 1, …, M−1, $D^1_{w^m_{i,j}}\hat{F}(k)=\left[D^1_{g_i^m}\hat{F}(k)\right]\beta_j^{m-1}$, and $D^1_{b_i^m}\hat{F}(k)=D^1_{g_i^m}\hat{F}(k)$. Note that in Eqs. (41), (43), and (44), when $\hat{F}(k)=0$, β_i is constant. Thus, if $\hat{F}(k)=0$, we obtain $D^1_{g_i^M}\beta_i=0$ and ${}_1\rho_i^m(k)=0$. In fact, in Eqs. (41), (43), and (44), the second formula can be derived from the first one, but not vice versa. As we know, when and only when Eqs. (43) and (44) are set up, is the point of convergence of the iterative search processes (the final convergence result) the same as the global minimum point of the square error $\hat{F}(k)$ of an FBPNN; namely, the iterative search algorithms in Eqs. (27) and (28) can be guaranteed to converge to a fractional-order optimal minimum point $\left(w^{m^*}_{i,j},b^{m^*}_i\right)$, a global minimum point, of the square error $\hat{F}(k)$ of an FBPNN. This completes the proof.

Lemma 1 indicates that first, $D^1_{w^m_{i,j}}\hat{F}(k)={}_1\rho_i^m(k)\beta_j^{m-1}$ and $D^1_{b_i^m}\hat{F}(k)={}_1\rho_i^m(k)$ consider merely the local characteristics of the square error $\hat{F}(k)$ of a classic first-order BPNN. Thus, a classic first-order BPNN is likely to converge to a local extreme point of its square error $\hat{F}(k)$. Assume that the iterative search process of a classic first-order BPNN converges to a local extreme point $\left(w^{m^l}_{i,j},b^{m^l}_i\right)\neq\left(w^{m^*}_{i,j},b^{m^*}_i\right)$. Thus, on this local extreme point

$\left(w_{i,j}^{m^l}, b_i^{m^l}\right)$, one can obtain

$$\begin{cases} \hat{F}(k) \neq 0, \\ {}_1\rho_i^m(k) = D_{g_i^m}^1 \hat{F}(k) = 0. \end{cases} \tag{45}$$

However, in Eqs. (25) and (26), $\tilde{D}_{w_{i,j}^m}^v \hat{F}(k)$ and $\tilde{D}_{b_i^m}^v \hat{F}(k)$ consider the non-local characteristics and the weak singularity of the square error $\hat{F}(k)$ of an FBPNN. Thus, an FBPNN that is trained by an improved FSDM can be guaranteed to converge to a global minimum point, of its square error $\hat{F}(k)$. The higher optimal search ability of an FBPNN to determine the global optimal solution is the major advantage that makes the FBPNN superior to a classic first-order BPNN.

Second, if the square error $\hat{F}(k)$ is mixed with white noise, the white noise should increase the convergence time of the reverse incremental search of an FBPNN, which could cause Eq. (41) not to be set up. In this case, Eq. (41) could be invalid and hence the FBPNN may not converge to the fractional-order optimal point.

Third, the computational complexity of an algorithm can be typically measured by the number of its multiplications and additions, and the related memory space, so the computational complexity of a BPNN is linear with $\boldsymbol{W}^m = \left[w_{i,j}^m\right]_{\psi^m \times \psi^{m-1}}$ and $\boldsymbol{b}^m = \left[b_i^m\right]_{\psi^m \times 1}$ (i=1→ψ^m and j=1→ψ^{m-1}), which is in direct proportion to $O[\boldsymbol{W}^m]+O[\boldsymbol{b}^m]$ ($O[\cdot]$ is the number of free parameters of a matrix). Therefore, for n=1, 2, 3 in Eqs. (27) and (28), the computational complexity of an FBPNN that is trained by an improved FSDM is in direct proportion to $4\{O[\boldsymbol{W}^m]+O[\boldsymbol{b}^m]\}$. Therefore, with the same number of neurons, the computational complexity of an FBPNN is three times greater than that of a classic first-order BPNN. In particular, Eqs. (27), (28), and (41) indicate that the final convergence result of an FBPNN with n=1 is the same as that with n=1, 2, 3. Thus, to further simplify the calculation of Eqs. (27) and (28), without loss of generality, in the actual computations of an FBPNN that is trained by an improved FSDM, we can set n=1. In this case, with the same number of neurons, the computational complexity of an FBPNN is the same as that of a classic first-order BPNN.

Fourth, in general, the fractional-order extreme value of a normalized quadratic energy norm determined by the fractional-order partial derivatives is not equal to its integer-order one (Pu et al., 2015). However, with respect to a specific quadratic function $\hat{F}(k)=\sum_{i=1}^{\psi^M}\left(q_i-\beta_i\right)^2$, Eqs. (3), (5), (41), and (44) indicate that if the global minimum value of the square error $\hat{F}(k)$ is equal to zero, the fractional-order optimal minimum point $\left(w_{i,j}^{m^*}, b_i^{m^*}\right)$ determined by the approximate fractional-order partial derivatives expressed by Eqs. (25) and (26) is identical to the global minimum value of the square error $\hat{F}(k)$.

3.4 Assumption of the structure of an improved FSDM-based FBPNN

In this subsection, an assumption of the structure of an FBPNN that is trained by an improved FSDM is made.

First, in a manner similar to a classic first-order BPNN, an FBPNN's behavior is highly dependent on the number of neurons and that of hidden layers. The aforementioned properties highlighted for an FBPNN regarding the fractional-order multi-scale global optimization search assume that the size of an FBPNN is sufficient, such that at least a fractional-order minimum point (a global minimum point) exists. With regard to an undersized improved FSDM-based FBPNN, there does not exit a zero square error $\hat{F}(k)$. Thus, it is possible that the domain of attraction of the fractional-order optimal minimum point of an undersized improved FSDM-based FBPNN does not contain the attractor $\left(w_{i,j}^{m^*}, b_i^{m^*}\right)$, or this domain of attraction is not included in the state space of an undersized improved FSDM-based FBPNN at all. Assume that there is a local extreme point $\left(w_{i,j}^{m^l}, b_i^{m^l}\right)$ different from the fractional-order optimal minimum point $\left(w_{i,j}^{m^*}, b_i^{m^*}\right)$ of the square error $\hat{F}(k)$. Thus, on this local extreme point $\left(w_{i,j}^{m^l}, b_i^{m^l}\right)$, substitution of Eq. (45) into Eqs. (27) and (28) results in

$$\begin{cases}\left|w_{i,j}^{m}(k+1)-w_{i,j}^{m}(k)\right|=\left|w_{i,j}^{m}(k)-w_{i,j}^{m^{l}}\right|\neq 0,\\ \left|b_{i}^{m}(k+1)-b_{i}^{m}(k)\right|=\left|b_{i}^{m}(k)-b_{i}^{m^{l}}\right|\neq 0.\end{cases} \tag{46}$$

Eq. (46) indicates that the iterative search algorithms in Eqs. (27) and (28) on a local extreme point $\left(w_{i,j}^{m^{l}}, b_{i}^{m^{l}}\right)$ cannot be ultimately terminated by themselves. On the one hand, if the domain of attraction of point $\left(w_{i,j}^{m^{*}}, b_{i}^{m^{*}}\right)$ is not complete and point $\left(w_{i,j}^{m^{l}}, b_{i}^{m^{l}}\right)$ is only in this domain of attraction, the iterative search process of an undersized improved FSDM-based FBPNN should ultimately somewhat oscillate around point $\left(w_{i,j}^{m^{l}}, b_{i}^{m^{l}}\right)$ and attempt to identify point $\left(w_{i,j}^{m^{*}}, b_{i}^{m^{*}}\right)$. Otherwise, it should deviate from point $\left(w_{i,j}^{m^{l}}, b_{i}^{m^{l}}\right)$ to further determine point $\left(w_{i,j}^{m^{*}}, b_{i}^{m^{*}}\right)$. Conversely, if the domain of attraction of point $\left(w_{i,j}^{m^{*}}, b_{i}^{m^{*}}\right)$ is not included in the state space at all, the iterative search process of an undersized improved FSDM-based FBPNN should deviate from point $\left(w_{i,j}^{m^{l}}, b_{i}^{m^{l}}\right)$ to make a continual attempt to search for point $\left(w_{i,j}^{m^{*}}, b_{i}^{m^{*}}\right)$.

Second, regarding a multi-layer perceptron, Cybenko (1989) first mathematically proved that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with a single hidden layer and any continuous sigmoidal nonlinearity. Sontag (1992) derived a general result demonstrating that nonlinear control systems can be stabilized using two hidden layers, however, not in general using a single hidden layer in a classic first-order BPNN. Barron (1993) examined how the approximation error is related to the number of nodes in a classic first-order BPNN. In a similar manner to a first-order BPNN, the approximation properties related to the number of neurons of the hidden layer in an FBPNN must be established. For convenience of discussion, without loss of generality, assume that the actual output function $\boldsymbol{\beta}_{\psi^{1}\times 1}^{1}(\boldsymbol{p}_{R\times 1})$ on $\mathbb{R}^{R\times 1}$ formulated in Eq. (3) is approximated by an FBPNN with a single layer of sigmoidal units, given as

$$\boldsymbol{\beta}_{\psi^{1}\times 1}^{1}\left(\boldsymbol{p}_{R\times 1}\right)=f^{1}\left(\boldsymbol{W}_{\psi^{1}\times R}^{1}\boldsymbol{p}_{R\times 1}+\boldsymbol{b}_{\psi^{1}\times 1}^{1}\right), \tag{47}$$

where R denotes the dimensionality of the input matrix, $\psi^{1}\geq 1$ denotes the number of neurons of the hidden layer, and f^{1} is an arbitrarily fixed sigmoidal function. Assume that $\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}(\boldsymbol{p}_{R\times 1})$ is a target output function of an FBPNN. If $\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}(\boldsymbol{p}_{R\times 1})$ is a causal signal with fractional primitives zero, from Eq. (1), we can derive the Fourier transform of the v-order derivative of $\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)$, given as

$$\mathrm{FT}\left[D_{\boldsymbol{p}_{R\times 1}}^{v}\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)\right]=\left(\varsigma\boldsymbol{\varpi}_{R\times 1}\right)^{v}\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{\varpi}_{R\times 1}\right), \tag{48}$$

where FT[·] denotes the Fourier transform, ς denotes an imaginary unit, $\boldsymbol{\varpi}_{R\times 1}$ denotes the angular frequency, and $\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{\varpi}_{R\times 1}\right)=\iint_{\mathbb{R}^{R\times 1}}\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)\mathrm{e}^{-\varsigma\boldsymbol{\varpi}_{R\times 1}\boldsymbol{p}_{R\times 1}}\mathrm{d}\boldsymbol{p}_{R\times 1}$ is the Fourier transform of $\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)$. From Eq. (48), the smoothness property of $\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)$ can be measured by the integrability of $\mathrm{FT}\left[D_{\boldsymbol{p}_{R\times 1}}^{v}\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)\right]$ on $\boldsymbol{\varpi}^{R\times 1}$, given as

$$\begin{aligned}C_{\beta}^{v}&=\iint_{\boldsymbol{\varpi}^{R\times 1}}\left|\mathrm{FT}\left[D_{\boldsymbol{p}_{R\times 1}}^{v}\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)\right]\right|\mathrm{d}\boldsymbol{\varpi}_{R\times 1}\\&=\iint_{\boldsymbol{\varpi}^{R\times 1}}\left|\left(\boldsymbol{\varpi}_{R\times 1}\right)^{v}\right|\left|\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{\varpi}_{R\times 1}\right)\right|\mathrm{d}\boldsymbol{\varpi}_{R\times 1},\end{aligned} \tag{49}$$

where $\left|\left(\boldsymbol{\varpi}_{R\times 1}\right)^{v}\right|=\left[\left(\boldsymbol{\varpi}_{R\times 1}\right)^{v}\cdot\left(\boldsymbol{\varpi}_{R\times 1}\right)^{v}\right]^{1/2}$ and "·" denotes the inner product. Assume that the approximation error of an FBPNN that is trained by an improved FSDM can be measured by the integrated square error regarding a probability ρ on the hyper-ball $B_r=\{\boldsymbol{p}_{R\times 1}: \|\boldsymbol{p}_{R\times 1}\|\leq r\}$ of radius $r=\sup\|\boldsymbol{p}_{R\times 1}\|>0$. Therefore, we have the following lemma:

Lemma 2 If a target output function $\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)$ with $C_{\beta}^{v}\geq 0$ finite is in the closure of the convex hull of a set $\boldsymbol{G}$ in a Hilbert space, with $\|\boldsymbol{g}\|<b$ for each $\boldsymbol{g}\in\boldsymbol{G}$ and $\bar{C}_{\beta}^{v}=\left(2rC_{\beta}^{v}\right)^{2}\geq b^{2}-\left\|\boldsymbol{\beta}_{\psi^{1}\times 1}^{*}\left(\boldsymbol{p}_{R\times 1}\right)\right\|^{2}\geq 0$, there

should be an actual output function $\boldsymbol{\beta}^{1}_{\psi^1\times 1}(\boldsymbol{p}_{R\times 1})$ of an FBPNN formulated in Eq. (47) in the convex hull of $\psi^1\geq 1$ points in $\boldsymbol{G}$, such that

$$\iint_{B_r}\left[\boldsymbol{\beta}^{*}_{\psi^1\times 1}(\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{1}_{\psi^1\times 1}(\boldsymbol{p}_{R\times 1})\right]^2\rho\mathrm{d}\boldsymbol{p}_{R\times 1}\leq\frac{\bar{C}^{v}_{\beta}}{\psi^1},\quad(50)$$

where $\|\cdot\|$ denotes the norm of the Hilbert space. If $\boldsymbol{\beta}^{1}_{\psi^1\times 1}(\boldsymbol{p}_{R\times 1})$ is observed at sites $\{{}_i\boldsymbol{p}_{R\times 1}\}|_{i=1\to S}$ restricted to B_r that obey the uniform distribution, inequality (50) provides a restricted condition on the approximation error, given as

$$\frac{1}{S}\sum_{i=1}^{S}\left\|\boldsymbol{\beta}^{*}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{1}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\leq\frac{\bar{C}^{v}_{\beta}}{\psi^1},\quad(51)$$

where S is the sample size.

Proof We prove Lemma 2 based on the law of large numbers. Assume that $\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})$ is a point in the convex hull of $\boldsymbol{G}$ being extremely close to $\boldsymbol{\beta}^{*}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})$, with $\left\|\boldsymbol{\beta}^{*}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|\leq\delta/\psi^1$ and $\delta\geq 0$. $\boldsymbol{g}$ is stochastically drawn from the set $\left\{\boldsymbol{g}^{\mathrm{o}}_k\in\boldsymbol{G}\right\}\big|_{k=1\to m}$ with probability $P\left(\boldsymbol{g}=\boldsymbol{g}^{\mathrm{o}}_k\right)=\rho_k\geq 0$. Then, if m is sufficiently large, we have $\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})=\sum_{k=1}^{m}\rho_k\boldsymbol{g}^{\mathrm{o}}_k$ with $\sum_{k=1}^{m}\rho_k=1$. Furthermore, assume that $\left\{\boldsymbol{g}^{1}_i\right\}\big|_{i=1\to\psi^1}$ is independently drawn from the same distribution as $\boldsymbol{g}$. For the uniform distribution, $\rho=\rho_k=1/\psi^1$. If ψ^1 is sufficiently large, $\boldsymbol{\beta}^{1}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})=\sum_{i=1}^{\psi^1}\boldsymbol{g}^{1}_i\Big/\psi^1$ is a sample average. Thus, $E\left[\boldsymbol{\beta}^{1}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right]=\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})$, where $E[\cdot]$ denotes the mathematical expectation. Therefore,

$$\begin{aligned}&E\left[\left\|\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{1}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\right]\\&=E\left[\left\|\boldsymbol{g}-\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\right]\Big/\psi^1,\end{aligned}$$

which equals $\left[E\|\boldsymbol{g}\|^2-\left\|\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\right]\Big/\psi^1$ restricted by $\left[b^2-\left\|\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\right]\Big/\psi^1$. For the mathematical expectation to be bounded in this manner, there must be $\left\{\boldsymbol{g}^{\mathrm{o}}_k\right\}\big|_{k=1\to m}$, for which $\frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{1}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\leq\frac{1}{\psi^1}\left[b^2-\left\|\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}(\boldsymbol{p}_{R\times 1})\right\|^2\right]$. Because $\left\|\boldsymbol{\beta}^{*}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{\mathrm{o}}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|\leq\delta/\psi^1$ and $\bar{C}^{v}_{\beta}>b^2-\left\|\boldsymbol{\beta}^{*}_{\psi^1\times 1}(\boldsymbol{p}_{R\times 1})\right\|^2$, the proof of Lemma 2 is completed by the choice of a sufficiently small δ.

Furthermore, let $\bar{C}^{v}_{\beta}\overset{r=1/2}{=}\left(C^{v}_{\beta}\right)^2$ and add a statistically estimated regularization item $-R\psi^1\log S/S$ (Barron, 1993) to the restricted condition on the approximation error of an FBPNN formulated in inequality (51), given as

$$\frac{1}{S}\sum_{i=1}^{S}\left\|\boldsymbol{\beta}^{*}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{1}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\leq\frac{\left(C^{v}_{\beta}\right)^2}{\psi^1}-\frac{R\psi^1}{S}\log S,\quad(52)$$

where R is the number of input nodes. Inequality (52) indicates that if the approximation error $\sum_{i=1}^{S}\left\|\boldsymbol{\beta}^{*}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})-\boldsymbol{\beta}^{1}_{\psi^1\times 1}({}_i\boldsymbol{p}_{R\times 1})\right\|^2\Big/S=0$, the number of neurons of the hidden layer in an FBPNN can be derived as

$$\psi^1\cong C^{v}_{\beta}\left[S/\left(R\log S\right)\right]^{1/2}.\quad(53)$$

Eq. (53) indicates that when the number of neurons of the hidden layer $\psi^1\cong C^{v}_{\beta}\left[S/(R\log S)\right]^{1/2}$, the size of an FBPNN is sufficient such that at least one fractional-order minimum exists.

3.5 Fractional-order multi-scale global optimization of an improved FSDM-based FBPNN

In this subsection, the fractional-order multi-scale global optimization of an FBPNN that is trained by an improved FSDM is analyzed.

Eqs. (25)–(28) indicate that the variation of the fractional order v of $\tilde{D}^{v}_{w^m_{i,j}}\hat{F}$ and $\tilde{D}^{v}_{b^m_i}\hat{F}$ can actually nonlinearly influence the learning process of an

FBPNN to a certain degree. We restrict the square error $\hat{F}(k)$ to be a nonlinearly increasing function of fractional order $v(k)$. Furthermore, to simplify the calculation, we restrict $0\le v\le 2$.

First, Eqs. (5), (25), and (26) indicate that on a given local extreme point and in its neighborhood, we have $0<\sigma_{\mathrm{L}}^2\le\hat{F}(k)=e^2(k)<\sigma_{\mathrm{U}}^2, \tilde{D}^v_{w_{i,j}^m}\hat{F}\neq 0$ and $\tilde{D}^v_{b_i^m}\hat{F}\neq 0$, where σ_{L} and σ_{U} are a sufficiently small positive scalar and a sufficiently large positive scalar, respectively. To enable escape from a local extreme point and its neighborhood, the iterative search process of an FBPNN that is trained by an improved FSDM should have climbing capacity on $\hat{F}(k)$ by itself; i.e., $\tilde{D}^v_{w_{i,j}^m}\hat{F}$ and $\tilde{D}^v_{b_i^m}\hat{F}$ should be in nearly opposite directions of $D^1_{w_{i,j}^m}\hat{F}$ and $D^1_{b_i^m}\hat{F}$, respectively. On the one hand, if $D^1_{w_{i,j}^m}\hat{F}>0$ or $D^1_{b_i^m}\hat{F}>0$, we should restrict $\tilde{D}^v_{w_{i,j}^m}\hat{F}<0$ or $\tilde{D}^v_{b_i^m}\hat{F}<0$ correspondingly. Thus, from Eqs. (25)–(28), we can derive

$$\tilde{D}^v_{w_{i,j}^m}\hat{F}(k)=\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n<0, \tag{54}$$

$$\tilde{D}^v_{b_i^m}\hat{F}(k)=\frac{\left[b_i^m(k)-b_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}{}_n\rho_i^m(k)<0. \tag{55}$$

If restricting $0<v<1$ in this case, we have $0<1/\Gamma(1-v)<1$. Thus, if $0<v<1$, inequalities (54) and (55) can be simplified as

$$\hat{F}(k)<-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n, \tag{56}$$

$$\hat{F}(k)<-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k). \tag{57}$$

For $0<\sigma_{\mathrm{L}}^2\le\hat{F}(k)=e^2(k)<\sigma_{\mathrm{U}}^2$, to enable inequalities (56) and (57) to be set up, a necessary condition is given as

$$0<-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n=\Gamma(1-v)\left|\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|, \tag{58}$$

$$0<-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)=\Gamma(1-v)\left|\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\right|. \tag{59}$$

Furthermore, when $0<v<1$, $\left|\binom{v}{n}\frac{1}{\Gamma(n-v+1)}\right|_{n=1,2,3}$ obtains its maximum value when $n=1$. Thus, according to the properties of inequality, if $0<v<1$, from inequalities (58) and (59), inequalities (56) and (57) can be further simplified as inequalities (60) and (61), respectively. From inequalities (60) and (61), one can further derive inequalities (62) and (63).

$$\hat{F}(k)<\Gamma(1-v)\sum_{n=1}^{3}\left|\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|\le\frac{v}{1-v}\sum_{n=1}^{3}\left|\left[w_{i,j}^m(k)-w_{\inf}\right]^n{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|=\sigma_1^2(v), \tag{60}$$

$$\hat{F}(k)<\Gamma(1-v)\sum_{n=1}^{3}\left|\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\right|\le\frac{v}{1-v}\sum_{n=1}^{3}\left|\left[b_i^m(k)-b_{\inf}\right]^n{}_n\rho_i^m(k)\right|=\sigma_2^2(v). \tag{61}$$

$$0<v_{T_1}=\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[w_{i,j}^m(k)-w_{\inf}\right]^n{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|+1\right\}^{-1}<v<1, \tag{62}$$

$$0<v_{T_2}=\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[b_i^m(k)-b_{\inf}\right]^n{}_n\rho_i^m(k)\right|+1\right\}^{-1}<v<1. \tag{63}$$

In inequalities (60) and (62), $\sigma_{\mathrm{L}}^2=\min\left(\sigma_1^2(v),\sigma_1^2\left(v_{T_1}\right)\right)$ and $\sigma_{\mathrm{U}}^2=\max\left(\sigma_1^2(v),\sigma_1^2\left(v_{T_1}\right)\right)$. In inequalities (61) and (63), $\sigma_{\mathrm{L}}^2=\min\left(\sigma_2^2(v),\sigma_2^2\left(v_{T_2}\right)\right)$ and $\sigma_{\mathrm{U}}^2=\max\left(\sigma_2^2(v),\sigma_2^2\left(v_{T_2}\right)\right)$.

In addition, if restricting $1<v<2$ in this case, we have $-0.3<1/\Gamma(1-v)<0$. Thus, if $1<v<2$, inequalities (54) and (55) can be simplified as

$$\hat{F}(k)>-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n, \tag{64}$$

$$\hat{F}(k)>-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k). \tag{65}$$

For $0<\sigma_{\mathrm{L}}^2\le\hat{F}(k)=e^2(k)<\sigma_{\mathrm{U}}^2$, to enable inequalities (54) and (55) to be set up, a weak restriction is given as

$$\begin{aligned}&-\Gamma(1-v)\left|\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|\\&=-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n>0,\end{aligned} \tag{66}$$

$$\begin{aligned}&-\Gamma(1-v)\left|\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\right|\\&=-\Gamma(1-v)\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)>0.\end{aligned} \tag{67}$$

Furthermore, when $1<v<2$, $\left|\binom{v}{n}\frac{1}{\Gamma(n-v+1)}\right|_{n=1,2,3}$ obtains its maximum value when $n=1$. Thus, if $1<v<2$, from inequalities (66) and (67), inequalities (64) and (65) can be further simplified as inequalities (68) and (69), respectively. From inequalities (68) and (69), one can further derive inequalities (70) and (71).

$$\begin{aligned}\hat{F}(k)>\sigma_3^2(v)&=\frac{-v}{1-v}\sum_{n=1}^{3}\left|\left[w_{i,j}^m(k)-w_{\inf}\right]^n{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|\\&\ge-\Gamma(1-v)\left|\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|,\end{aligned} \tag{68}$$

$$\begin{aligned}\hat{F}(k)>\sigma_4^2(v)&=\frac{-v}{1-v}\sum_{n=1}^{3}\left|\left[b_i^m(k)-b_{\inf}\right]^n{}_n\rho_i^m(k)\right|\\&\ge-\Gamma(1-v)\left|\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^n}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\right|.\end{aligned} \tag{69}$$

$$1<v_{T_3}=-\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[w_{i,j}^m(k)-w_{\inf}\right]^n{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n\right|-1\right\}^{-1}<v<2, \tag{70}$$

$$1<v_{T_4}=-\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[b_i^m(k)-b_{\inf}\right]^n{}_n\rho_i^m(k)\right|-1\right\}^{-1}<v<2. \tag{71}$$

In inequalities (68) and (70), $\sigma_{\mathrm{L}}^2=\min\left(\sigma_3^2(v),\sigma_3^2\left(v_{T_3}\right)\right)$ and $\sigma_{\mathrm{U}}^2=\max\left(\sigma_3^2(v),\sigma_3^2\left(v_{T_3}\right)\right)$. In inequalities (69) and (71), $\sigma_{\mathrm{L}}^2=\min\left(\sigma_4^2(v),\sigma_4^2\left(v_{T_4}\right)\right)$ and $\sigma_{\mathrm{U}}^2=\max\left(\sigma_4^2(v),\sigma_4^2\left(v_{T_4}\right)\right)$.

On the other hand, if $D^1_{w_{i,j}^m}\hat{F}<0$ or $D^1_{b_i^m}\hat{F}<0$, we should restrict $\tilde{D}^v_{w_{i,j}^m}\hat{F}>0$ or $\tilde{D}^v_{b_i^m}\hat{F}>0$ correspondingly. Thus, from Eqs. (25)–(28), we can derive

$$\begin{aligned}\tilde{D}^v_{w_{i,j}^m}\hat{F}(k)&=\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)\\&+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[w_{i,j}^m(k)-w_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}{}_n\rho_i^m(k)\left(\beta_j^{m-1}\right)^n>0,\end{aligned} \tag{72}$$

$$\begin{aligned}\tilde{D}^v_{b_i^m}\hat{F}(k)&=\frac{\left[b_i^m(k)-b_{\inf}\right]^{-v}}{\Gamma(1-v)}\hat{F}(k)\\&+\sum_{n=1}^{3}\binom{v}{n}\frac{\left[b_i^m(k)-b_{\inf}\right]^{n-v}}{\Gamma(n-v+1)}{}_n\rho_i^m(k)>0.\end{aligned} \tag{73}$$

In a similar way, if we restrict 0<v<1 in this case, inequalities (74) and (75) are true.

In addition, if we restrict 1<v<2 in this case, inequalities (76) and (77) are true.

$$0<v<\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[w_{i,j}^{m}(k)-w_{\inf}\right]^{n}{}_{n}\rho_{i}^{m}(k)\left(\beta_{j}^{m-1}\right)^{n}\right|+1\right\}^{-1}=v_{T_1}<1, \tag{74}$$

$$0<v<\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[b_{i}^{m}(k)-b_{\inf}\right]^{n}{}_{n}\rho_{i}^{m}(k)\right|+1\right\}^{-1}=v_{T_2}<1. \tag{75}$$

$$1<v<-\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[w_{i,j}^{m}(k)-w_{\inf}\right]^{n}{}_{n}\rho_{i}^{m}(k)\left(\beta_{j}^{m-1}\right)^{n}\right|-1\right\}^{-1}=v_{T_3}<2, \tag{76}$$

$$1<v<-\left\{\frac{1}{\hat{F}(k)}\sum_{n=1}^{3}\left|\left[b_{i}^{m}(k)-b_{\inf}\right]^{n}{}_{n}\rho_{i}^{m}(k)\right|-1\right\}^{-1}=v_{T_4}<2. \tag{77}$$

Second, when $0\le\hat{F}(k)<\sigma_{\mathrm{L}}^{2}$, the iterative search process of an FBPNN is in the neighborhood of a fractional-order optimal minimum point. Inequalities (62), (63), (74), and (75) further show that, to encourage the iterative search process of an FBPNN to converge to this given fractional-order optimal minimum point, the square error $\hat{F}(k)$ should decrease until it is equal to zero. Thus, when $0\le\hat{F}(k)<\sigma_{\mathrm{L}}^{2}$, if $D_{w_{i,j}^{m}}^{1}\hat{F}>0$ or $D_{b_i^{m}}^{1}\hat{F}>0$, we restrict 0≤v<v_{T_1}<1 for $w_{i,j}^{m}$ or 0≤v<v_{T_2}<1 for b_i^{m}. Note that from Eq. (1), $D_{x}^{v}\big|_{v=0}$ is an identity operator, which implements neither differential nor integral. Thus, we have $D_{x}^{0}0=0$. Eqs. (22) and (23) restrict v=0 on a fractional-order optimal minimum point with $\hat{F}(k)=0$. When v=0, the iterative search process of an FBPNN can be terminated on a fractional-order optimal minimum point of the square error $\hat{F}(k)$ by itself; if $D_{w_{i,j}^{m}}^{1}\hat{F}<0$ or $D_{b_i^{m}}^{1}\hat{F}<0$, we set 0<v_{T_1}<v≤1 for $w_{i,j}^{m}$ or 0<v_{T_2}<v≤1 for b_i^{m}.

Third, when $\hat{F}(k)\ge\sigma_{\mathrm{U}}^{2}$, we can directly set v=1. Note that first, when $0<\sigma_{\mathrm{L}}^{2}\le\hat{F}(k)<\sigma_{\mathrm{U}}^{2}$, $\tilde{D}_{w_{i,j}^{m}}^{v}\hat{F}$ and $\tilde{D}_{b_i^{m}}^{v}\hat{F}$ are restricted in the near opposite directions of $D_{w_{i,j}^{m}}^{1}\hat{F}$ and $D_{b_i^{m}}^{1}\hat{F}$, respectively. As we know, the first-order gradient of a point is in the direction of the fastest growth of the scalar field, and its module value is the maximum rate of change. In numerical implementation, the directions of $D_{w_{i,j}^{m}}^{1}\hat{F}$ and $D_{b_i^{m}}^{1}\hat{F}$ are those of their maximum module values, respectively. For example, on the two-dimensional discrete plane, there are eight directional derivatives (their interval is 45°) of a point $\left(w_{i,j}^{m},b_i^{m}\right)$ in its neighborhood. We select the maximum directional derivative of a point $\left(w_{i,j}^{m},b_i^{m}\right)$ to be $D_{w_{i,j}^{m}}^{1}\hat{F}$ and $D_{b_i^{m}}^{1}\hat{F}$. Second, to enhance the multi-scale search capability, we should also set the learning rate, $\mu(k)$, of an FBPNN to be an appropriate correlation function with $\tilde{D}_{w_{i,j}^{m}}^{v}\hat{F}$, $\tilde{D}_{b_i^{m}}^{v}\hat{F}$, and $\hat{F}(k)$. Third, to guarantee to escape from the domain of attraction of a given local extreme point, as long as $\tilde{D}_{w_{i,j}^{m}}^{v}\hat{F}$ and $\tilde{D}_{b_i^{m}}^{v}\hat{F}$ begin to climb, they should keep climbing until arriving at the top of the hill in the neighborhood of this local extreme point. On the top of the hill, we should restrict the downward direction to be different from the previous uphill direction of $\tilde{D}_{w_{i,j}^{m}}^{v}\hat{F}$ and $\tilde{D}_{b_i^{m}}^{v}\hat{F}$.

4 Experiment and analysis

4.1 Example function approximation of an improved FSDM-based FBPNN

In this subsection, we discuss an example of function approximation of an FBPNN that is trained by an improved FSDM, and it will be used for the four examples in the next subsection.

As we know, multi-layer networks can be used to approximate virtually any function if we have a sufficient number of neurons in the hidden layers. We choose a network for an FBPNN and apply it to a

particular problem. Without loss of generality, we illustrate the characteristics with a 1-ψ^1-1 improved FSDM-based FBPNN (Fig. 2).

In Fig. 2, ψ^1 denotes the number of neurons in the first layer (hidden layer) of an FBPNN. The activation functions for the first and second layers are log-sigmoid, and can be expressed as follows:

$$f^1(x)=f^2(x)=\frac{1}{1+\mathrm{e}^{-x}}. \tag{78}$$

To simplify the analysis, we assign a specific problem to an FBPNN. We know the global optimization solution to this problem. In Fig. 2, we set ψ^1=2. It is also assumed that the function to be approximated is the response of a 1-2-1 improved FSDM-based FBPNN, with the following values for the weights and biases: $w^1_{1,1}=10$, $w^1_{2,1}=10$, $b^1_1=-5$, $b^1_2=5$, $w^2_{1,1}=1$, $w^2_{1,2}=1$, and b^2=−1. The response for these parameters is displayed in Fig. 3, which consists of the plot of this improved FSDM-based FBPNN output β^2 as the input p varies over the range [−2, 2].

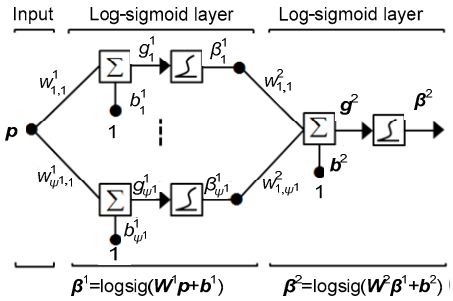

Fig. 2 An example of function approximation of an FBPNN

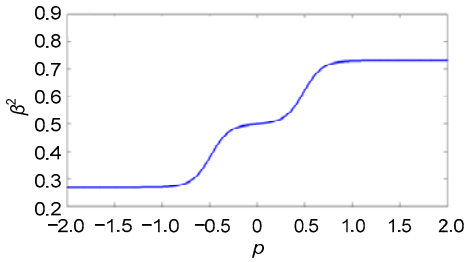

Fig. 3 Function to be approximated

The improved FSDM-based FBPNN described in Fig. 2 must be trained to approximate the function displayed in Fig. 3. The approximation is exact when the parameters are set as in the previous paragraph. We assume that the function to be approximated is sampled at the values p=−2.0, −1.9, …, 2.0 with an equal probability. The mean square error of this improved FSDM-based FBPNN is equal to the average sum of the square errors at these 41 points. To plot the mean square error of this improved FSDM-based FBPNN in a three-dimensional space, we vary only two parameters at the same time. Fig. 4 illustrates the mean square error of this improved FSDM-based FBPNN when only two parameters, $w^1_{1,1}$ and $w^2_{1,1}$, are adjusted; the other parameters are set to their aforementioned optimal values.

It is observed that the optimal minimum error occurs when $w^1_{1,1}=10$ and $w^2_{1,1}=1$, as indicated by the solid green circle in Fig. 4.

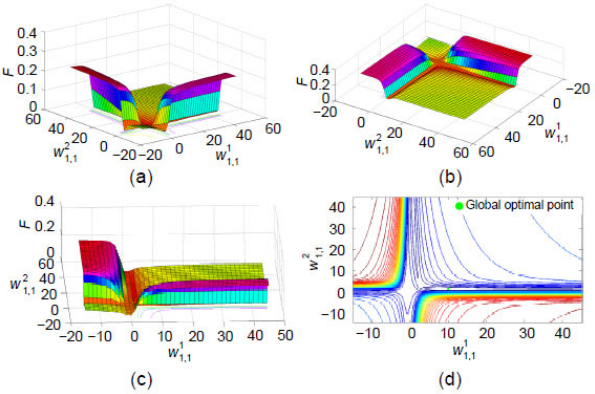

Fig. 4 Mean square error of an FBPNN: (a) front view; (b) back view; (c) lateral view; (d) contour map (References to color refer to the online version of this figure)

4.2 Fractional-order multi-scale global optimization of an improved FSDM-based FBPNN

In this subsection, we analyze the fractional-order multi-scale global optimization of an FBPNN that is trained by an improved FSDM.

Inspired by the aforementioned mathematical analysis, the fractional order $v(k)$ of $\tilde{D}^v_{w^m_{i,j}}\hat{F}(k)$ and $\tilde{D}^v_{b^m_i}\hat{F}(k)$ of an FBPNN can be self-adaptively determined. To simplify the multi-scale search process in the following simulations, we construct an imperfect adaptive kernel function of the fractional order v at the k^{th} iteration of an FBPNN as follows:

$$v(k)=2\left|\frac{1-\Phi^{-e(k)}}{1+\Phi^{-e(k)}}\right|+|e(k)|, \tag{79}$$

where $\Phi=\left|\rho^{M}\right|^{2+e(k)}$, $e(k)=\sum_{i=1}^{\psi^{M}}e_i(k)\Big/\psi^{M}$ is the average error at the k^{th} iteration, and $\rho^{M}=\sum_{i=1}^{\psi^{M}}{}_1\rho_i^{M}\Big/\psi^{M}$ is the first-order average sensibility in the M^{th} layer (the last layer) of an FBPNN.

From Eqs. (3), (5), (10), (24), and (78), we can derive the first-order average sensibility in the output layer of the improved FSDM-based FBPNN in Fig. 2, ${}_1\rho_i^2(k)\big|_{i=1\to1}={}_1\rho_1^2(k)=-2e(k)\beta_1^2\left(1-\beta_1^2\right)$. The output $\beta_1^2=\beta_i^2\big|_{i=1\to1}$ of an FBPNN changes dynamically with the initial condition, weight matrices, bias matrices, and average error $e(k)$. For convenience of illustration, we analyze the multi-scale adjustment of the adaptive kernel function of $v(k)$ by assuming the output $\beta_1^2=\beta_i^2\big|_{i=1\to1}=1+e(k)$, $\beta_j^1\big|_{j=1\to2}=1$, $w_{i,j}^2(k)\big|_{i=1\to1,\ j=1\to2}=1$, and $b_i^2(k)\big|_{i=1\to1}=1$ in Eq. (79), which is displayed in Fig. 5.

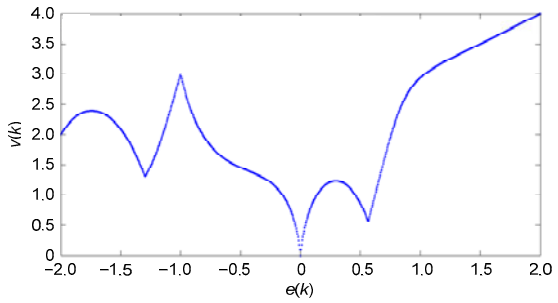

Fig. 5 Adaptive kernel function of fractional order $v(k)$

From Fig. 5 and Eq. (79), we can observe that the fractional order $v(k)$ adaptively varies with the error $e(k)$ during the entire iterative search process of an FBPNN. As in the previous discussion, if the minimum value of the square error $\hat{F}(k)$ is equal to zero, the fractional order $v(k)$ approaches zero on a fractional-order optimal extreme point (global optimal minimum point) with zero average error $e(k)$=0.

Example 1 In this example, we select an initial condition in the specific zone from a random sample of cases, where the reverse incremental searches of an FBPNN and a classic first-order BPNN can both converge to the global optimal minimum point of the square error $\hat{F}(k)$ at the k^{th} iteration. We set the same parameters, the rate of convergence μ=5.50 and the number of iterations 2000, for both the FBPNN and first-order BPNN in this simulation; the initial condition is at the point where $w_{1,1}^1=-4$ and $w_{1,1}^2=-4$. From Eqs. (27)–(30), (78), and (79), the iterative search processes of the FBPNN and first-order BPNN can be represented as in Fig. 6.

In Fig. 6a, we observe two convergence trajectories of the FBPNN and first-order BPNN (batch mode) when only two parameters ($w_{1,1}^1$ and $w_{1,1}^2$) vary. Figs. 6a and 6b indicate that regarding the

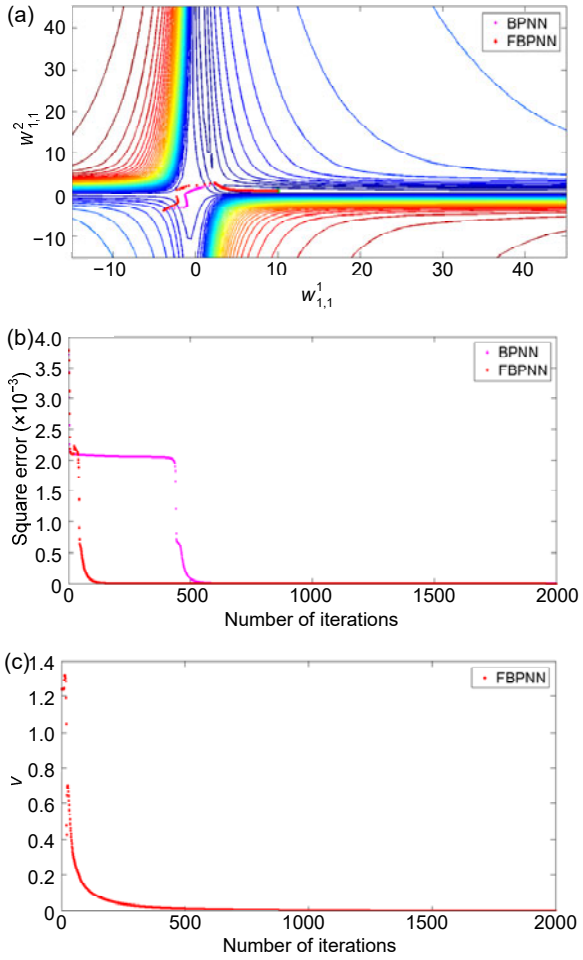

Fig. 6 Comparison of iterative search processes of an FBPNN and a first-order BPNN in Example 1: (a) convergence trajectories; (b) convergence patterns of the square error of the k^{th} iteration; (c) fractional order v of the FBPNN

aforementioned initial condition ($w_{1,1}^1=-4$ and $w_{1,1}^2=-4$), the FBPNN and first-order BPNN eventually converge to an optimal point ($w_{1,1}^1=10$ and $w_{1,1}^2=1$). The square errors $\hat{F}(k)$ of the FBPNN and first-order BPNN on an optimal point are equal to zero. The two convergence trajectories of the FBPNN and first-order BPNN bypass an initial flat surface and then fall into a gently sloping valley, as observed in Fig. 6a. From Fig. 6b, we can observe that the FBPNN requires fewer iterations than the first-order BPNN to converge to the optimal point. Fig. 6c indicates that as the value of $\hat{F}(k)$ changes over the convergence trajectory of the FBPNN, the fractional order $v(k)$ of the FBPNN varies according to the adaptive kernel function in Eq. (79). Thus, the fractional order $v(k)$ of the FBPNN in its entire iterative search process is not constant, demonstrating the fractional-order multi- scale global optimization of the FBPNN trained by an improved FSDM.

Example 2 In this example, we select an initial condition in the gently sloping zone of the square error $\hat{F}(k)$ from a random sample of cases. We set the same parameters, the rate of convergence μ=5.50, the number of iterations 9000, for both an FBPNN and a first-order BPNN in this simulation; the initial condition is at the point where $w_{1,1}^1=5$ and $w_{1,1}^2=30$. From Eqs. (27)–(30), the iterative search processes of the FBPNN and the first-order BPNN can be represented as in Fig. 7.

In Fig. 7a, the convergence trajectory of the first-order BPNN illustrates the manner in which it can converge to a local extreme point ($w_{1,1}^1=0.7003$ and $w_{1,1}^2=35.2626$). The convergence trajectory of the first-order BPNN is trapped in a valley and diverges from the global optimal solution. However, the convergence trajectory of the FBPNN illustrates the manner in which it can converge to a global optimal minimum point ($w_{1,1}^1=10$ and $w_{1,1}^2=1$). The convergence trajectory of the FBPNN passes a valley and converges to the global optimal solution. Fig. 7b indicates that when the number of iterations of the first-order BPNN is approximately 6500, the square error $\hat{F}(k)$ of the first-order BPNN reduces to a nonzero minimum, which is approximately 0.25×10^{-3}. As the number of iterations increases, this nonzero minimum cannot continue to decrease. Thus, the first-order BPNN can be trapped in a local extreme point. Fig. 7b also indicates that when the number of iterations of the FBPNN increases from 1 to 10, its square error $\hat{F}(k)$ decreases sharply from 3.4×10^{-3} to 0.35×10^{-3}. When the number of iterations increases from 11 to 3000, its square error $\hat{F}(k)$ does not decrease; instead, it increases gradually from 0.35×10^{-3} to 0.82×10^{-3}. Furthermore, when the number of iterations increases from 3001 to 4000, its square error $\hat{F}(k)$ decreases sharply from 0.82×10^{-3} to a zero minimum. As the number of iterations increases, this

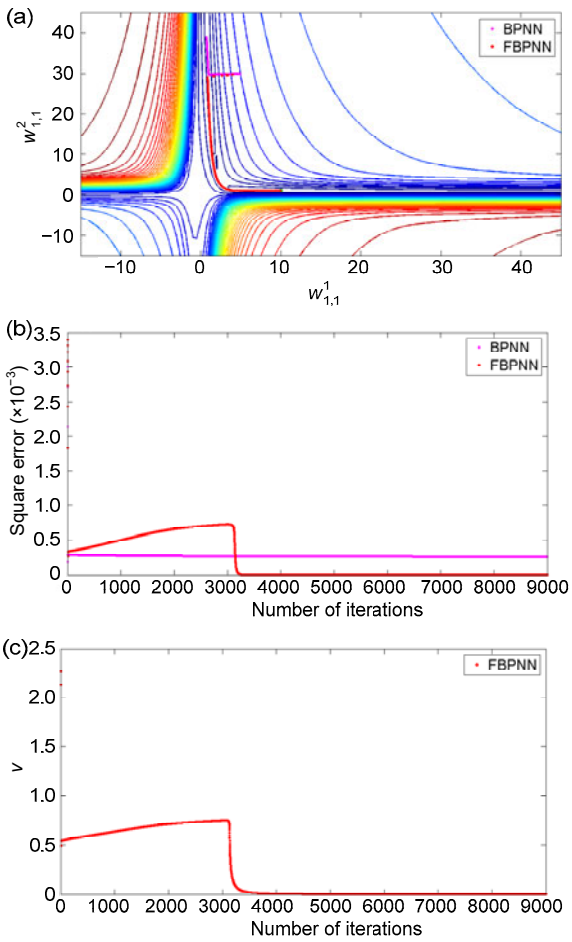

Fig. 7 Comparison of iterative search processes of an FBPNN and a first-order BPNN in Example 2: (a) convergence trajectories; (b) convergence patterns of the square error of the k^{th} iteration; (c) fractional order v of the FBPNN

minimum remains at zero. Thus, the FBPNN converges to a global optimal minimum point. From Eqs. (1), (27), (28), and (79), we observe that when $0<v<1$, the square error $\hat{F}(k)$ of the FBPNN may not always decrease; it can increase at certain points, which helps the FBPNN bypass the first-order local extreme points of $\hat{F}(k)$. Fig. 7c indicates that when the number of iterations of the FBPNN increases from 1 to 10, its fractional order v decreases sharply from 2.30 to 0.51. When the number of iterations increases from 11 to 3000, its fractional order v increases gradually from 0.51 to 0.80. Furthermore, when the number of iterations increases from 3001 to 4000, its fractional order v decreases sharply from 0.80 to zero. From Eq. (79), we can observe that the fractional order v of the FBPNN varies with its average error $e(k)$ changing during the entire iterative process. Thus, the fractional order v of the FBPNN in its entire iterative search process is not constant, demonstrating the fractional-order multi-scale global optimization of the FBPNN trained by an improved FSDM.

Example 3 To further verify the fractional-order multi-scale global optimization of the FBPNN trained by an improved FSDM, we select another initial condition in the sharply sloping zone of the square error $\hat{F}(k)$ from a random sample of cases. We set the same parameters, the rate of convergence μ=5.50, the number of iterations 6000, for both the FBPNN and first-order BPNN in this simulation; the initial condition is at the point $w_{1,1}^1=-8$ and $w_{1,1}^2=9$. From Eqs. (27)–(30), the iterative search processes of the FBPNN and first-order BPNN can be represented as in Fig. 8.

In Fig. 8a, the convergence trajectory of the first-order BPNN illustrates the manner in which it can converge to a local extreme point ($w_{1,1}^1=0.7003$ and $w_{1,1}^2=35.2626$). However, the convergence trajectory of the FBPNN illustrates the manner in which it can converge to a global optimal minimum point ($w_{1,1}^1=10$ and $w_{1,1}^2=1$). Fig. 8b indicates that when the number of iterations of the first-order BPNN is approximately 2000, the square error $\hat{F}(k)$ reduces to a nonzero minimum, which is approximately 0.25×10^{-3}. Thus, the first-order BPNN can be trapped in a local extreme point. However, when the number of iterations of the FBPNN is approximately 2300, the square error $\hat{F}(k)$ reduces to a zero minimum. Thus, the FBPNN converges to a global optimal minimum point. Fig. 8b also indicates that when $0<v<1$, the square error $\hat{F}(k)$ of the FBPNN may not always decrease; instead, it can increase at certain points, which helps the FBPNN bypass the first-order local extreme points of $\hat{F}(k)$. Fig. 8c indicates that the fractional order v of the FBPNN in its entire iterative search process is not constant; instead, it varies according to Eq. (79), demonstrating the fractional-order multi-scale global optimization of the FBPNN trained by the improved FSDM.

Example 4 We present an extreme example where the initial condition is directly on a local extreme

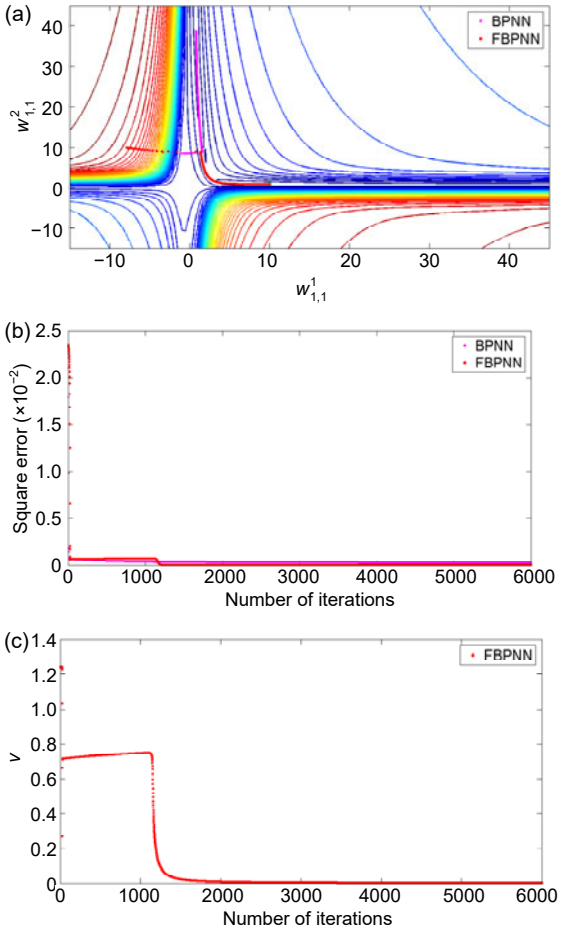

Fig. 8 Comparison of iterative search processes of an FBPNN and a first-order BPNN in Example 3: (a) convergence trajectories; (b) convergence patterns of the square error of the k^{th} iteration; (c) fractional order v of the FBPNN

point of the square error $\hat{F}(k)$. Assume that we have set the same parameters, the rate of convergence μ=5.50, the number of iterations 9000, for both the FBPNN and first-order BPNN in this simulation, and the initial condition is directly at the local extreme point ($w_{1,1}^1=0.7003$ and $w_{1,1}^2=35.2626$) of $\hat{F}(k)$. Thus, from Eqs. (27)–(30), the iterative search processes of the FBPNN and first-order BPNN can be represented as in Fig. 9.

Figs. 9a and 9c indicate that if the initial condition is directly at a local extreme point of $\hat{F}(k)$, the convergence trajectory of the first-order BPNN is trapped at this local extreme point and the square error $\hat{F}(k)$ remains unchanged. Fig. 9b indicates that even if the initial condition is directly at a local extreme point of $\hat{F}(k)$, the convergence trajectory of the FBPNN can converge to a global optimal minimum point ($w_{1,1}^1=10$ and $w_{1,1}^2=1$). Fig. 9d indicates that as the number of iterations of the FBPNN increases from 1 to 3500, its $\hat{F}(k)$ does not decrease; instead, it increases gradually from 2.5×10^{-4} to 7.4×10^{-4}. Furthermore, when the number of iterations increases from 3501 to 5200, $\hat{F}(k)$ decreases sharply from 7.4×10^{-4} to a zero minimum. As the number of iterations increases, this zero minimum remains at zero. Thus, the FBPNN converges to a global optimal minimum point. Fig. 9e indicates that the fractional order v of the FBPNN in its entire iterative search process is not constant; instead, it varies according to Eq. (79), demonstrating the fractional-order multi-scale global optimization of the FBPNN trained by an improved FSDM.

4.3 Comparative performance of an improved FSDM-based FBPNN with real data

In this subsection, to further verify the fractional-order multi-scale global optimization, we compare the global search and error fitting abilities of the FBPNN and a classic first-order BPNN with real data.

Example 5 In this example, the data presented in Table 1 display the output as a transfer function of the input of a nonlinear signal processing filter.

We choose an FBPNN and a Levenberg-Marquardt algorithm based first-order BPNN (Hagan and Menhaj, 1994) to provide two nonlinear least-square approximations to the data in Table 1. We illustrate the characteristics with a 1-ψ^1-1 improved FSDM-based FBPNN and a same structure first-order BPNN. The numbers of neurons in the first layer (hidden layer) of the FBPNN and first-order BPNN are both ψ^1=15. The activation functions for the first and second layers are tan-sigmoid $f^1(x)=2/(1+e^{-2x})-1$ and pure linear function $f^2(x)=x$, respectively. $\boldsymbol{W}_{15\times1}^1=[w_{1,1}^1,\ 20.8597, 21.2543, -21.0232, -21.3975, 21.0826, -21.0743, 21.0052, 21.0272, 20.9446,\ w_{11,1}^1,\ -21.1307, 21.2419,\ 20.9357,\ 21.0157]^{\mathrm{T}}$ and $\boldsymbol{W}_{1\times15}^2=[-0.7629, -0.7168,\ 1.1592,\ 0.4330,\ 0.9470,\ 0.5903,\ -1.1983,$

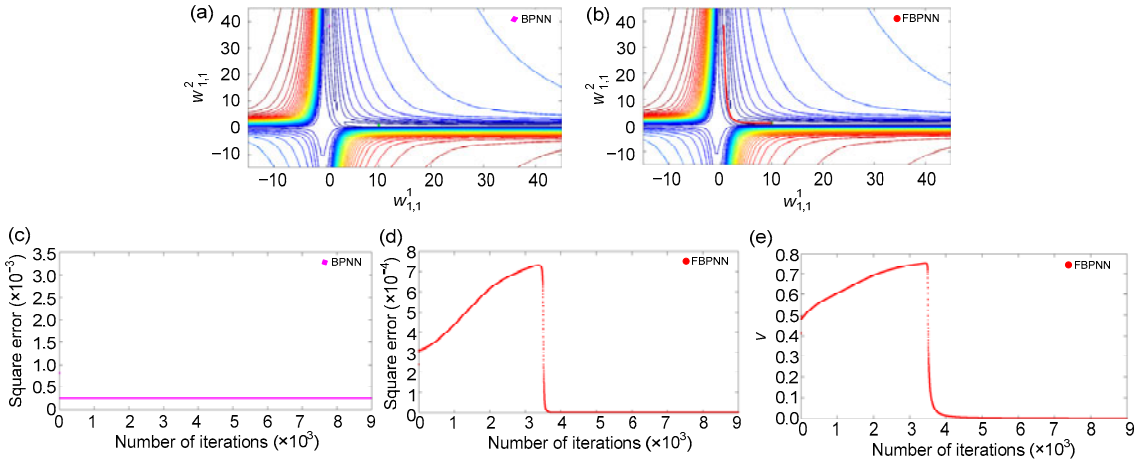

Fig. 9 Comparison of iterative search processes of an FBPNN and a first-order BPNN in Example 4: (a) convergence trajectory of the BPNN; (b) convergence trajectory of the FBPNN; (c) convergence patterns of the square error of the k^{th} iteration of the BPNN; (d) convergence patterns of the square error of the k^{th} iteration of the FBPNN; (e) fractional order v of the FBPNN

−0.7002, −0.3756, −1.0144, −0.2451, −1.3834, 0.4546, 0.2460, 0.3230] are randomly selected as the weight matrices of the first and second layers of the FBPNN and first-order BPNN, respectively. $\boldsymbol{b}^1_{15\times1}$ = [−21.0070, −18.1627, −14.6449, 11.9684, 8.0087, −5.7329, 2.0816, 0.7399, 2.7071, 6.1967, −8.9802, −11.7774, 14.6532, 18.0707, 20.9846]$^{\mathrm{T}}$ and $b^2_{1,1}$=−0.4954 are also randomly selected as the biases of the first and second layers of the FBPNN and first-order BPNN, respectively. For convenience of illustration, we simultaneously vary only two parameters of the FBPNN. Fig. 10 illustrates the mean square error of the FBPNN when only two parameters, $w^1_{1,1}$ and $w^1_{11,1}$, are adjusted; the other parameters are set to their aforementioned randomly selected values.

It can be observed that a near optimal minimum error occurs when $w^1_{1,1}$=19.3065 and $w^1_{11,1}$= −20.4575, as indicated by the solid green circle in Fig. 10. We set the same parameters, the rate of convergence μ=3.50, the number of iterations 3000, for both the FBPNN and first-order BPNN in this simulation, and the initial conditions are at the points where $w^1_{1,1}$=108 and $w^1_{11,1}$=116, $w^1_{1,1}$=−110 and $w^1_{11,1}$=−106, and $w^1_{1,1}$=−95 and $w^1_{11,1}$=100. From Eqs. (27)–(30), the iterative search processes of the FBPNN and first-order BPNN can be represented as in Fig. 11.

In Figs. 11a, 11d, and 11g, the convergence trajectories illustrate that the first-order BPNN can converge to three local extreme points, $w^1_{1,1}$=86.0196 and $w^1_{11,1}$=84.169, $w^1_{1,1}$=−83.8969 and $w^1_{11,1}$= −74.0297, and $w^1_{1,1}$=−94.9789 and $w^1_{11,1}$=74.0876. However, all convergence trajectories illustrate that the FBPNN can converge to a near optimal global minimum point ($w^1_{1,1}$=19.3065 and $w^1_{11,1}$= −20.4575). Figs. 11b, 11e, and 11h display that first, when the number of iterations increases, the square error $\hat{F}(k)$ of the first-order BPNN reduces to a nonzero minimum, which is clearly greater than that of the FBPNN. Second, because we vary only two parameters ($w^1_{1,1}$ and $w^1_{11,1}$) simultaneously, the maximum adjustment of the FBPNN converges to the near optimal global minimum point ($w^1_{1,1}$=19.3065 and $w^1_{11,1}$= −20.4575), where its minimum square error $\hat{F}(k)$ $\left(\hat{F}_{\min}=0.001115\right)$ approaches zero (but not equal to zero). If we vary all the parameters simultaneously, the convergence trajectory of the FBPNN can converge to a global optimal minimum point, where its minimum square error $\hat{F}(k)$ is equal to zero. Figs. 11c, 11f, and 11i, and Eq. (79) indicate that the minimum square error $\hat{F}(k)$ $\left(\hat{F}_{\min}=0.001115\right)$ approaches zero and that the minimum of the fractional order v of the FBPNN trained by an improved FSDM approaches zero (but not equal to zero).

Table 1 Input and output of a nonlinear signal processing filter

Input	−1.0	−0.9	−0.8	−0.7	−0.6	−0.5	−0.4	−0.3	−0.2	−0.1
Output	−0.832	−0.423	−0.024	0.344	1.282	3.456	4.020	3.232	2.102	1.504
Input	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Output	0.248	1.242	2.344	3.262	2.052	1.684	1.022	2.224	3.022	1.984

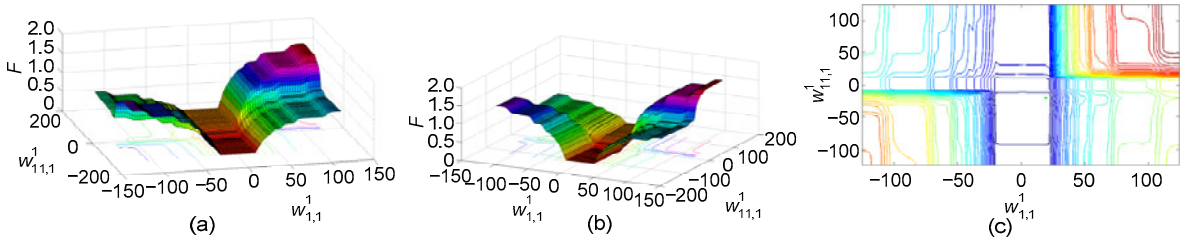

Fig. 10 Mean square error of an FBPNN: (a) left side view; (b) right side view; (c) contour map (References to color refer to the online version of this figure)

In the following extreme example, the initial condition is directly on a local extreme point of the square error $\hat{F}(k)$. We set the same parameters, the rate of convergence μ=3.50, the number of iterations 3000 for both the FBPNN and first-order BPNN in this simulation; the initial condition is directly at a local extreme point ($w^1_{1,1}$=−9.00 and $w^1_{11,1}$=8.2676) of $\hat{F}(k)$. Thus, from Eqs. (27)–(30), the iterative search processes of the FBPNN and first-order BPNN can be represented as in Fig. 12.

Fig. 12 indicates that if the initial condition is directly at a local extreme point of $\hat{F}(k)$, the convergence trajectory of the first-order BPNN is trapped at this local extreme point and the square error $\hat{F}(k)$ remains unchanged. Conversely, even if the initial condition is directly at a local extreme point of $\hat{F}(k)$, the convergence trajectory of the FBPNN can converge to a near optimal global minimum point ($w^1_{1,1}$=19.3065 and $w^1_{11,1}$=−20.4575).

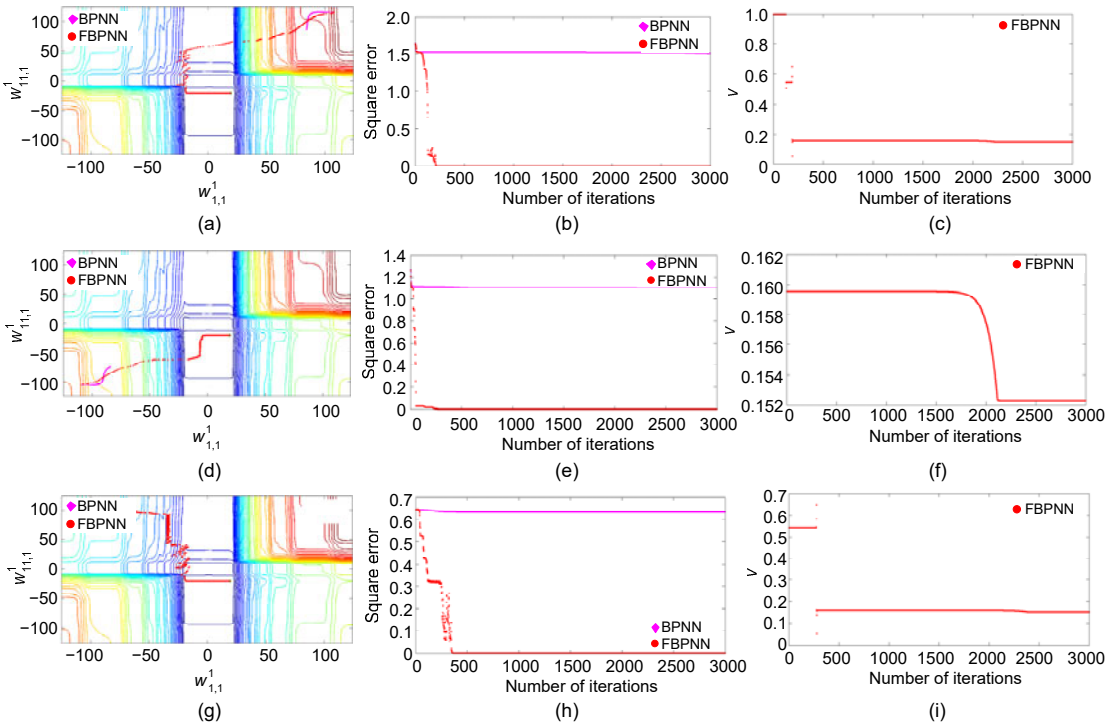

Fig. 11 Comparison of iterative search processes of an FBPNN and a first-order BPNN in Example 5: (a) convergence trajectories ($w^1_{1,1}$=108 and $w^1_{11,1}$=116); (b) convergence patterns of the square error of the k^{th} iteration ($w^1_{1,1}$=108 and $w^1_{11,1}$=116); (c) fractional order v of the FBPNN ($w^1_{1,1}$=108 and $w^1_{11,1}$=116); (d) convergence trajectories ($w^1_{1,1}$=−110 and $w^1_{11,1}$=−106); (e) convergence patterns of the square error of the k^{th} iteration ($w^1_{1,1}$=−110 and $w^1_{11,1}$=−106); (f) fractional order v of the FBPNN ($w^1_{1,1}$=−110 and $w^1_{11,1}$=−106); (g) convergence trajectories ($w^1_{1,1}$=−95 and $w^1_{11,1}$=100); (h) convergence patterns of the square error of the k^{th} iteration ($w^1_{1,1}$=−95 and $w^1_{11,1}$=100); (i) fractional order v of the FBPNN ($w^1_{1,1}$=−95 and $w^1_{11,1}$=100)

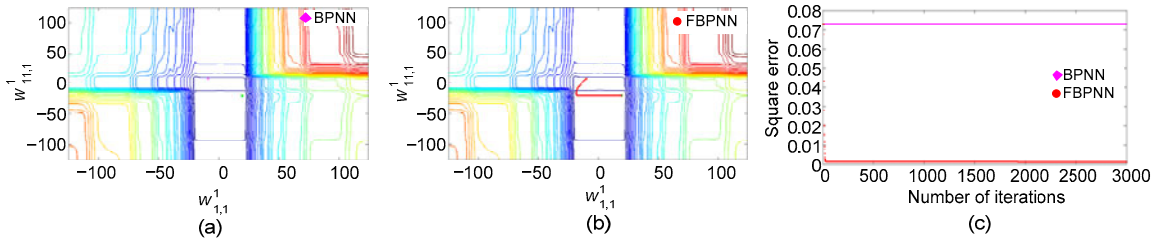

Fig. 12 Comparison of iterative search processes of an FBPNN and a first-order BPNN: (a) convergence trajectory of the BPNN; (b) convergence trajectory of the FBPNN; (c) convergence patterns of the square error of the k^{th} iteration

The responses of the FBPNN and first-order BPNN for the convergence parameters in Fig. 11 are displayed in Fig. 13, consisting of the plots of the outputs of the FBPNN and first-order BPNN as the inputs vary over the range in Table 1.

Fig. 13 indicates that for the convergence parameters in Fig. 11, the sample data in Table 1 can be fitted well by the FBPNN. The fitting error of the first-order BPNN is clearly greater than that of the FBPNN trained by an improved FSDM.

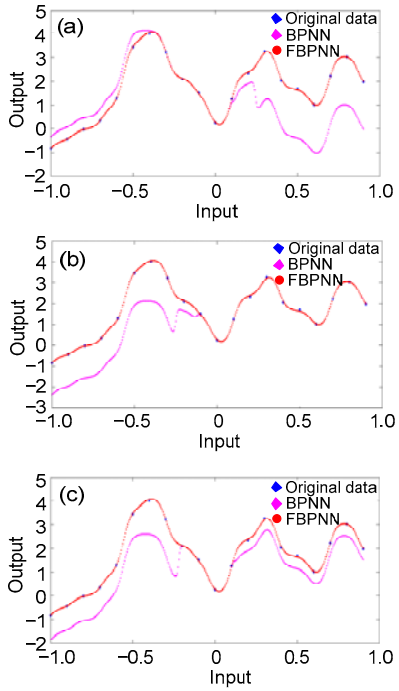

Fig. 13 Responses of an FBPNN and a first-order BPNN for the convergence parameters in Fig. 11a (a), Fig. 11d (b), and Fig. 11g (c)

5 Conclusions

The application of fractional calculus to neural networks and cybernetics is an emerging field of study and only a small number of studies have been conducted in this area. The properties of the fractional calculus of a signal are considerably different from those of its integer-order calculus. Fractional calculus has been applied to neural networks and cybernetics primarily due to its inherent advantages of long-term memory, non-locality, and weak singularity. Therefore, to improve the optimization performance of the ordinary first-order BPNNs, it is logical to generalize a first-order BPNN to an FBPNN by applying a state-of-the-art application of a promising mathematical method, fractional calculus. From this inspiration, in this study, an FBPNN trained by an improved FSDM was achieved, whose reverse incremental search was in the negative directions of the approximate fractional-order partial derivatives of the square error $\hat{F}(k)$. The higher optimal search ability of an FBPNN to determine the global optimal solution is the major advantage that makes the FBPNN superior to a classic first-order BPNN.

From the aforementioned discussion, we also observe that there are other problems that must be further studied. For example, the imperfect adaptive kernel function of the fractional order v at the k^{th} iteration of the FBPNN, $v(k)$, is not sufficient for an arbitrary quadratic energy norm $\hat{F}(k)$. Therefore, it is evident that other topics, such as how to construct an efficient appropriate correlation function of $\mu(k)$ and how to construct a more efficient adaptive kernel function of fractional order $v(k)$ of the FBPNN trained by an improved FSDM, must be studied further. These topics will be discussed in our future work.

Contributors

Yi-fei PU designed the research and drafted the manuscript. Jian WANG helped organize the manuscript. Yi-fei PU and Jian WANG processed the data, and revised and finalized the paper.

Compliance with ethics guidelines

Yi-fei PU and Jian WANG declare that they have no conflict of interest.

References

Andramonov M, Rubinov A, Glover B, 1999. Cutting angle methods in global optimization. *Appl Math Lett*, 12(3): 95-100. https://doi.org/10.1016/S0893-9659(98)00179-7

Barnard E, 1992. Optimization for training neural nets. *IEEE Trans Neur Netw*, 3(2):232-240. https://doi.org/10.1109/72.125864

Barron AR, 1993. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans Inform Theory*, 39(3):930-945. https://doi.org/10.1109/18.256500

Battiti R, 1992. First- and second-order methods for learning: between steepest descent and Newton's method. *Neur Comput*, 4(2):141-166. https://doi.org/10.1162/neco.1992.4.2.141

Browne CB, Powley E, Whitehouse D, et al., 2012. A survey

of Monte Carlo tree search methods. *IEEE Trans Comput Intell AI Games*, 4(1):1-43. https://doi.org/10.1109/tciaig.2012.2186810

Cantu-Paz E, Kamath C, 2005. An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. *IEEE Trans Syst Man Cybern*, 35(5):915-927. https://doi.org/10.1109/TSMCB.2005.847740

Charalambous C, 1992. Conjugate gradient algorithm for efficient training of artificial neural networks. *IEE Proc G*, 139(3):301-310. https://doi.org/10.1049/ip-g-2.1992.0050

Chuang CC, Su SF, Hsiao CC, 2000. The annealing robust backpropagation (ARBP) learning algorithm. *IEEE Trans Neur Netw*, 11(5):1067-1077. https://doi.org/10.1109/72.870040

Cybenko G, 1989. Approximation by superpositions of a sigmoidal function. *Math Contr Signals Syst*, 2(4):303-314. https://doi.org/10.1007/bf02551274

Elwakil AS, 2010. Fractional-order circuits and systems: an emerging interdisciplinary research area. *IEEE Circ Syst Mag*, 10(4):40-50. https://doi.org/10.1109/MCAS.2010.938637

Hagan MT, Menhaj MB, 1994. Training feedforward networks with the Marquardt algorithm. *IEEE Trans Neur Netw*, 5(6):989-993. https://doi.org/10.1109/72.329697

Heymans N, Podlubny I, 2006. Physical interpretation of initial conditions for fractional differential equations with Riemann-Liouville fractional derivatives. *Rheol Acta*, 45(5):765-771. https://doi.org/10.1007/s00397-005-0043-5

Hornik K, Stinchcombe M, White H, 1989. Multilayer feedforward networks are universal approximators. *Neur Netw*, 2(5):359-366. https://doi.org/10.1016/0893-6080(89)90020-8

Jacobs RA, 1988. Increased rates of convergence through learning rate adaptation. *Neur Netw*, 1(4):295-307. https://doi.org/10.1016/0893-6080(88)90003-2

Kaslik E, Sivasundaram S, 2011. Dynamics of fractional-order neural networks. Proc Int Joint Conf on Neural Networks, p.1375-1380. https://doi.org/10.1109/IJCNN.2011.6033277

Koeller RC, 1984. Applications of fractional calculus to the theory of viscoelasticity. *J Appl Mech*, 51(2):299-307. https://doi.org/10.1115/1.3167616

LeCun Y, 1985. Une procedure d'apprentissage pour reseau a seuil assymetrique. *Proc Cogn*, 85:599-604 (in French).

Leung FHF, Lam HK, Ling SH, et al., 2003. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Trans Neur Netw*, 14(1):79-88. https://doi.org/10.1109/TNN.2002.804317

Ludermir TB, Yamazaki A, Zanchettin C, 2006. An optimization methodology for neural network weights and architectures. *IEEE Trans Neur Netw*, 17(6):1452-1459. https://doi.org/10.1109/TNN.2006.881047

Manabe S, 2002. A suggestion of fractional-order controller for flexible spacecraft attitude control. *Nonl Dynam*, 29(1-4):251-268. https://doi.org/10.1023/a:1016566017098

Maniezzo V, 1994. Genetic evolution of the topology and weight distribution of neural networks. *IEEE Trans Neur Netw*, 5(1):39-53. https://doi.org/10.1109/72.265959

Nikolaev NY, Iba H, 2003. Learning polynomial feedforward neural networks by genetic programming and backpropagation. *IEEE Trans Neur Netw*, 14(2):337-350. https://doi.org/10.1109/TNN.2003.809405

Oldham KB, Spanier J, 1974. The Fractional Calculus: Integrations and Differentiations of Arbitrary Order. Academic Press, New York, USA, p.1-234.

Özdemir N, Karadeniz D, 2008. Fractional diffusion-wave problem in cylindrical coordinates. *Phys Lett A*, 372(38):5968-5972. https://doi.org/10.1016/j.physleta.2008.07.054

Palmes PP, Hayasaka T, Usui S, 2005. Mutation-based genetic neural network. *IEEE Trans Neur Netw*, 16(3):587-600. https://doi.org/10.1109/tnn.2005.844858

Parker DB, 1985. Learning-Logic: Casting the Cortex of the Human Brain in Silicon. Technical Report, No. TR-47, Center for Computational Research in Economics and Management Science, MIT, USA.

Petráš I, 2011. Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation. Springer Berlin Heidelberg, Berlin, Germany, p.1-218.

Podlubny I, 1998. Fractional Differential Equations: an Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications. Academic Press, San Diego, USA, p.1-340.

Podlubny I, Petráš I, Vinagre BM, et al., 2002. Analogue realizations of fractional-order controllers. *Nonl Dynam*, 29(1-4):281-296. https://doi.org/10.1023/a:1016556604320

Pu YF, Zhou JL, Yuan X, 2010. Fractional differential mask: a fractional differential-based approach for multiscale texture enhancement. *IEEE Trans Image Process*, 19(2):491-511. https://doi.org/10.1109/TIP.2009.2035980

Pu YF, Zhou JL, Zhang Y, et al., 2015. Fractional extreme value adaptive training method: fractional steepest descent approach. *IEEE Trans Neur Netw Learn Syst*, 26(4):653-662. https://doi.org/10.1109/TNNLS.2013.2286175

Pu YF, Yi Z, Zhou JL, 2016. Defense against chip cloning attacks based on fractional Hopfield neural networks. *Int J Neur Syst*, 27(4):1750003. https://doi.org/10.1142/S0129065717500034

Pu YF, Yi Z, Zhou JL, 2017. Fractional Hopfield neural networks: fractional dynamic associative recurrent neural networks. *IEEE Trans Neur Netw Learn Syst*, 28(10):2319-2333. https://doi.org/10.1109/TNNLS.2016.2582512

Pu YF, Yuan X, Yu B, 2018a. Analog circuit implementation of fractional-order memristor: arbitrary-order lattice scaling fracmemristor. *IEEE Trans Circ Syst I*, 65(9):

2903-2916. https://doi.org/10.1109/TCSI.2018.2789907

Pu YF, Siarry P, Chatterjee A, et al., 2018b. A fractional-order variational framework for retinex: fractional-order partial differential equation-based formulation for multi-scale nonlocal contrast enhancement with texture preserving. *IEEE Trans Image Process*, 27(3):1214-1229. https://doi.org/10.1109/TIP.2017.2779601

Rigler AK, Irvine JM, Vogl TP, 1991. Rescaling of variables in back propagation learning. *Neur Netw*, 4(2):225-229. https://doi.org/10.1016/0893-6080(91)90006-q

Rossikhin YA, Shitikova MV, 1997. Applications of fractional calculus to dynamic problems of linear and nonlinear hereditary mechanics of solids. *Appl Mech Rev*, 50(1): 15-67. https://doi.org/10.1115/1.3101682

Rumelhart DE, Hinton GE, Williams RJ, 1986a. Learning representations by back-propagating errors. *Nature*, 323(6088):533-536. https://doi.org/10.1038/323533a0

Rumelhart DE, McClelland JL, PDP Research Group, 1986b. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1, MIT Press, Cambridge, USA, p.547-611.

Shanno DF, 1990. Recent advances in numerical techniques for large-scale optimization. In: Miller WT, Sutton RS, Werbos PJ (Eds.), Neural Networks for Control. MIT Press, Cambridge, USA, p.171-178.

Sontag ED, 1992. Feedback stabilization using two-hidden-layer nets. *IEEE Trans Neur Netw*, 3(6):981-990. https://doi.org/10.1109/72.165599

Tollenaere T, 1990. SuperSAB: fast adaptive back propagation with good scaling properties. *Neur Netw*, 3(5):561-573. https://doi.org/10.1016/0893-6080(90)90006-7

Treadgold NK, Gedeon TD, 1998. Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm. *IEEE Trans Neur Netw*, 9(4):662-668. https://doi.org/10.1109/72.701179

Vogl TP, Mangis JK, Rigler AK, et al., 1988. Accelerating the convergence of the back-propagation method. *Biol Cybern*, 59(4-5):257-263. https://doi.org/10.1007/bf00332914

Werbos PJ, 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD Thesis, Harvard University, Cambridge, USA.

Yeh WC, 2013. New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series. *IEEE Trans Neur Netw Learn Syst*, 24(4):661-665. https://doi.org/10.1109/TNNLS.2012.2232678

Zanchettin C, Ludermir TB, Almeida LM, 2011. Hybrid training method for MLP: optimization of architecture and training. *IEEE Trans Syst Man Cybern*, 41(4):1097-1109. https://doi.org/10.1109/TSMCB.2011.2107035