Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2025 (43)** : 1–24.
ISSN-0037-8712
doi:10.5269/bspm.76732

The role of residual neural networks for advancing fractional differential equations

Alice Antony and Lakshmi Narayan Mishra*

ABSTRACT: This study provides the first comprehensive demonstration of how to utilise ResNets to estimate a family of generalized Caputo-type fractional differential equations and their solutions, and how to limit the quantity of parameters present in these ResNets. The basis of our evidence is the variational iteration method. It determines the differential equation's exact solution with the use of the variational iteration method. Then it shows how to estimate these equations using residual neural networks, using the structure produced by the variational iteration method.

Key Words: Fractional differential equations, ResNets, *VIM*, close approximation.

Contents

1. Introduction

In the study of FDEs, researchers have noted that fractional-order operators display non-local properties, unlike integer-order operators. It also involves fractional derivatives of functions. FDEs find use in diverse domains of physics and engineering, such as electromagnetism [30], heat transfer engineering [32], viscoelasticity [35], and seepage flow [12]. Fractional differential equations are essential in the domains of engineering, environmental phenomena, and the physical and engineering sciences [2,27,28]. Delay differential equations also play a crucial role in these fields, contributing to a deeper understanding of complex phenomena. It's a growing area of study and importance as an extension of integer differential equations. The effective memory function of fractional derivatives has made FDEs a popular tool for representing a wide range of physical phenomena. For FDEs, a wide range of approximate analytical techniques have been developed, including *VIM* [34], the Adomian decomposition method [36], the rectangle decomposition method [25], the trapezoidal method [6] as well as the differential transform method [3]. Recent studies have analyzed weakly singular nonlinear Volterra integral equations using discretization techniques. The solvability and approximation of weakly singular nonlinear Volterra integral equations using discretization techniques have been demonstrated by a thorough investigation [7]. Given their close relationship to fractional differential equations, these works highlight the power of classical numerical approaches. Many authors have spent years significantly refining *VIM*. The author He [13] created the strong analytical method known as the *VIM*, which is used to solve FDEs. The Fokker-Planck equation [20], the Lotka-Volterra formula [14], the fuzzy differential formula [19], and other differential formulas of an integer nature can all be successfully solved with it. The fractional order differential equations that it also works well with are the following: partially differential equations of fractional third order dispersive [9]; Riesz differential equations [22]; left or right Caputo fractional derivatives [31]; left or right Riemann-Liouville derivatives [5]; Erdélyi-Kober fractional derivatives [26]; left and right sided generalized k-fractional formula [37] and other fractional derivatives [10,16]. Fractional differential equations

* Corresponding author.
2020 *Mathematics Subject Classification:* 68T07, 26A33, 47J30.
Submitted May 10, 2025. Published September 24, 2025

have been successfully utilized in interpolation problems, providing a powerful tool for modelling and estimating missing data points in a more accurate and efficient manner [8]. FDEs are widely used to represent fluid dynamics phenomena like turbulence, anomalous diffusion, and viscoelastic flows because of their ability to capture memory and heredity effects [17]. More and more, fluid dynamics is simulating complicated fluid behaviours with memory-dependent and non-local features using fractional calculus.

These days, open source machine learning software frameworks like PyTorch, TensorFlow, and Keras allow us to build neural networks, even ones with tremendous structural complexity, with just a few lines of code. The complex method of creating a neural network involves giving careful thought to a number of variables, such as optimization techniques, activation functions, and network architecture. It is possible to create, train, and evaluate these complex structures with the help of a variety of tools and programs used in neural network development. This process involves carefully calibrating the level of model complexity to achieve optimal generalizability, with the ultimate aim of developing a network that can learn, adapt, and respond to new data.

Lately, numerical approximations of partial differential equations have been successfully achieved through the use of neural networks. Neural networks form the foundation of deep learning, delivering unparalleled results in tasks such as image classification, object detection, and language translation. Also, numerous studies have shown that they can effectively approximate solutions to integro-differential equations [15] and partial differential equations [33,23]. Jentzen et al. presented proof that deep artificial neural networks can approximate broad models numerically, such as Kolmogorov partial differential equations, and Baggenstos et al. [4] showed that ResNets can also approximate these models. It was developed in order to solve the issue of vanishing gradients in deep networks, which was a major barrier to the development of deep neural networks. A common problem with deep networks, local minima, prevents the network from learning additional layers of features. This is made possible with the ResNets architecture. The ResNet neural network architecture, introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 paper 'Deep Residual Learning for Image Recognition', has been a game-changer in computer vision research. ResNet's innovative approach has significantly altered the landscape of deep convolutional neural network training, with major implications for computer vision tasks. The residual learning framework of ResNets is its key innovation, permitting the network to learn residual functions, which alleviates the vanishing gradient problem and enables deeper networks. The ResNets design is computationally more efficient since it produces better results with fewer parameters. A lot of partial differential equations, including the Poisson equation, the Burgers equation [21], and others, have been numerically approximated using ResNets.

In this work aims to analyse FDEs in the following format:

$$\begin{cases} {}_v\mathcal{I}_V^{\delta}({}_0^c\mathcal{D}_v^{\delta,\rho}w)(v) + w(v) & = -f(v),\ v \in [0,V], \\ w(0) = w(V) = 0, \end{cases} \tag{1.1}$$

where $\delta \in [\frac{1}{2}, 1), {}_v\mathcal{I}_V^{\delta}, {}_0^c\mathcal{D}_v^{\delta,\rho}$ denote the right RLFI and left GCTFD, correspondingly. Based on the work, it can be estimated that ResNets can solve Eq.(1.1). ResNets help resolve the issue of the vanishing gradient and enable the training of much deeper networks. Deeper networks can be enabled generally and efficiently with residual connections. To the best of our knowledge, this paper presents a pioneering proof of numerical approximation for fractional differential equations using deep learning techniques, marking a significant first in this field. ResNets can approximate the solution to illustrate Eq.(1.1) by means of the *VIM* mediation [11]. Explicit or approximate solutions are found for both linear and nonlinear differential equations using this method. Using the *VIM*, we shall establish the applicable theorem of FDE with GCTFD. According to our findings, the ResNets need a polynomial number of parameters to approximate the answer to Eq.(1.1) in both N, which is the number of the approximative solution's iterations obtained through VI, and K, which is the number of points chosen in the defining domain.

The study's second section starts with some basic fractional calculus equations and characteristics. It has the required Laplace transform as well. It also demonstrates the convergence of the *VIM* and provides an iterative framework for precisely solving Eq.(1.1). Section 3 demonstrates how to use ResNets to estimate the iterated value in Section 2, as it closely resembles the exact solution to Eq.(1.1). To help

explain the findings, Section 4 offers a few instances. A presentation of the conclusion wraps up Section 5. In the sequel, we utilize the following notations:
• $\mathbb{R}$: The collection of all real numbers,
• $\mathbb{N}$: The collection of all natural numbers,
• $\mathbb{C}$: The collection of all continuous function,
• FDEs: Fractional differential equations,
• GCTFD: Generalized Caputo-type fractional derivatives,
• RLFI: Riemann-Liouville fractional integrals,
• RLFD: Riemann-Liouville fractional derivatives,
• LT: Laplace transform,
• T': Transpose of a matrix,
• $\mathscr{I}$: Identity matrix,
• VI: Variational iteration,
• VIF: Variational iteration formula,
• *VIM*: Variational iteration method,
• ResNets: Residual neural networks,
• FNNs: Neural networks with feedforward,
• RFNNs: FNN Realization,
• RResNets: ResNets Realization.

2. Initial and key findings

Definition 2.1 *(RLFI on the left and right [7]). Given a function f that is defined on the interval $[a, b]$, when $\delta > 0$ is a positive integer, then fractional integrals of $f(v)$ on the left and right are defined as*

$$
{}_a\mathcal{I}_v^{\delta} f(w) = \frac{1}{\Gamma(\delta)} \int_a^v (v-s)^{\delta-1} f(s)\, ds,
$$

$$
{}_v\mathcal{I}_b^{\delta} f(w) = \frac{1}{\Gamma(\delta)} \int_v^b (s-v)^{\delta-1} f(s)\, ds.
$$

Definition 2.2 *(RLFD on the left and right [38]). Consider a function f that is defined on the interval $[a, b]$. The RLFD of function $f(v)$ on the left and right, of order δ, are defined as*

$$
{}_a\mathcal{D}_v^{\delta} f(v) = \frac{d^m}{dv^m}({}_a\mathcal{D}_v^{\delta-m}) f(v) = \frac{1}{\Gamma(m-\delta)} \frac{d^m}{dv^m}\left(\int_a^v (v-s)^{m-\delta-1} f(s)\, ds \right),
$$

$$
{}_v\mathcal{D}_b^{\delta} f(v) = (-1)^m \frac{d^m}{dv^m}({}_v\mathcal{D}_b^{\delta-m}) f(v) = \frac{(-1)^m}{\Gamma(m-\delta)} \frac{d^m}{dv^m}\left(\int_v^b (s-v)^{m-\delta-1} f(s)\, ds \right).
$$

where $m-1 \le \delta < m$ and $m \in \mathbb{N}$.

Definition 2.3 *(GCTFD on left and right [24]). Given a function f that is defined on the interval $[a, b]$ of order $\delta > 0$, where $r-1 < \delta < r$, $r \in \mathbb{N}$, then the left and right GCTFD of $f(v)$ are defined as*

$$
{}_a^c\mathcal{D}_v^{\delta,\rho} f(v) = \frac{\rho^{\delta-r+1}}{\Gamma(r-\delta)} \left(\int_a^v s^{\rho-1} (v^\rho - s^\rho)^{r-\delta-1} \left(s^{1-\rho} \frac{d}{ds} \right)^r f(s) ds \right),
$$

$$
{}_v^c\mathcal{D}_b^{\delta,\rho} f(v) = \frac{(-1)^r \rho^{\delta-r+1}}{\Gamma(r-\delta)} \left(\int_v^b s^{\rho-1} (s^\rho - v^\rho)^{r-\delta-1} \left(s^{1-\rho} \frac{d}{ds} \right)^r f(s) ds \right).
$$

Remark 2.1 *When $\delta \in (0,1)$ and f is defined on the interval $[0,V]$, the right RLFI can be expressed in terms of the right GCTFD*

$$ {}_v\mathcal{I}_V^{-\delta} f(v) = I(v) - {}_v^c\mathcal{D}_V^{\delta,\rho} f(v), $$

$$ \textit{where} \quad I(v) = \frac{f(V)}{\Gamma(1-\delta)(V-v)^{\delta}}. $$

It is possible to express the right GCTFD with respect to the left GCTFD.

$$ \begin{aligned} {}_v^c\mathcal{D}_V^{\delta,\rho} f(v) =& \frac{-\rho^{\delta}}{\Gamma(1-\delta)} \left(\int_v^V s^{\rho-1}(s^{\rho}-v^{\rho})^{-\delta}(s^{1-\rho}\frac{d}{ds}) f(s) ds \right) \\ =& (-1)^{1-\delta} \ \frac{\rho^{\delta}}{\Gamma(1-\delta)} \ \left(\int_v^V (v^{\rho}-s^{\rho})^{-\delta} \frac{d}{ds} f(s) ds \right) \\ =& \frac{(-1)^{1-\delta}\rho^{\delta}}{\Gamma(1-\delta)} \left(\int_0^V (v^{\rho}-s^{\rho})^{-\delta} f'(s) ds - \int_0^v (v^{\rho}-s^{\rho})^{-\delta} f'(s) ds \right) \\ =& \frac{(-1)^{1-\delta}\rho^{\delta}}{\Gamma(1-\delta)} \int_0^V (v^{\rho}-s^{\rho})^{-\delta} f'(s) ds - \frac{(-1)^{1-\delta}\rho^{\delta}}{\Gamma(1-\delta)} \int_0^v (v^{\rho}-s^{\rho})^{-\delta} f'(s) ds \\ =& \frac{(-1)^{1-\delta}\rho^{\delta}}{\Gamma(1-\delta)} \int_0^V (v^{\rho}-s^{\rho})^{-\delta} f'(s) ds + (-1)^{-\delta} \ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v). \end{aligned} $$

The left GCTFD can be used to express the right RLFI,

$$ {}_v\mathcal{I}_V^{-\delta} f(v) = (-1)^{1-\delta} \ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v) + X(v) - Y(v), \tag{2.1} $$

where $\quad X(v) = \dfrac{f(V)}{\Gamma(1-\delta)(V-v)^{\delta}}, \ Y(v) = \dfrac{(-1)^{1-\delta}\rho^{\delta}}{\Gamma(1-\delta)} \displaystyle\int_0^V (v^{\rho}-s^{\rho})^{-\delta} f'(s) ds.$

Definition 2.4 *([29]) LT of the term ${}_0^c\mathcal{D}_v^{\delta,\rho}\mathscr{U}(v)$ is given as*

$$ \mathscr{L}[{}_0^c\mathcal{D}_v^{\delta,\rho}\mathscr{U}(v)] = s^{\delta}\bar{\mathscr{U}}(s) - \sum_{j=0}^{m-1} \mathscr{U}^{(j)}(0^+) s^{\delta-1-j}, $$

where $\mathscr{L}$ is denoted as the LT, $\bar{\mathscr{U}}(s) = \mathscr{L}[\mathscr{U}(v)]$.

Letting $\mathscr{L}[a(v)] = \bar{a}(s)$ and $\mathscr{L}[b(v)] = \bar{b}(s)$ as assumptions, then the convolution theorem is defined as

$$ a(v) * b(v) = \int_0^v a(v-\tau')b(\tau')d\tau', $$

and

$$ \mathscr{L}[a(v) * b(v)] = \mathscr{L}[a(v)]\mathscr{L}[b(v)] = \bar{a}(s)\bar{b}(s). $$

We create a new equation by converting Eq.(1.1):

$$ \begin{cases} {}_0^c\mathcal{D}_v^{\delta,\rho} w(v) - y(v) = 0, & \text{(a)} \\ {}_v\mathcal{I}_V^{\delta} y(v) + w(v) + f(v) = 0, & \text{(b)} \\ w(0) = w(v) = 0. \end{cases} \tag{2.2} $$

Theorem 2.1 *Eq.(2.2)(a) and Eq.(2.2)(b) have exact solutions that are represented by the VIFs $w(v)$ and $y(v)$, respectively*

$$w_{n+1}(v) = w_n(v) - {}_0\mathcal{I}_v^{\delta}[{}_0^c\mathcal{D}_{\tau'}^{\delta;\rho} w_n(\tau') - y_n(\tau')], \tag{2.3}$$

$$y_{n+1}(v) = y_n(v) + {}_v\mathcal{I}_V^{\delta}[{}_{\tau'}\mathcal{I}_V^{-\delta} y_n(\tau') + w_n(\tau') + f(\tau')]. \tag{2.4}$$

Proof: One method that can be used to discover the convergent successive approximation or the exact solution of a differential equation in confined as well as unconfined domains is *VIM*. To begin with, we demonstrate that the VIF for $w(v)$ is given by Eq.(2.3). With the use of the Caputo-type derivative, *VIM* approximates the solutions to the FDE and we quote a theorem in it.

Let us define $\mathscr{U}(v)$ on $[0, V]$. Linear and nonlinear operators are represented by the symbols L_a and N_a, respectively. In the event that the functional correction for

$${}_0^c\mathcal{D}_v^{\delta,\rho}\mathscr{U}(v) + L_a[\mathscr{U}(v)] + N_a[\mathscr{U}(v)] = f(v), \tag{2.5}$$

is determined by the left RLFI

$$\mathscr{U}_{n+1}(v) = \mathscr{U}_n(v) + {}_0\mathcal{I}_v^{\delta}\lambda(v,\tau')[{}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau') + L_a[\mathscr{U}_n(\tau')] + N_a[\mathscr{U}_n(\tau')] - f(\tau')],$$

where the multiplier of Lagrange is known to be $\lambda(v,\tau') = -1$ in which the terms $L_a[\mathscr{U}_n]$ and $N_a[\mathscr{U}_n]$, which are constrained variations.

The iteration formula for Eq.(2.5) can thus be found here:

$$\mathscr{U}_{n+1}(v) = \mathscr{U}_n(v) - {}_0\mathcal{I}_v^{\delta}[{}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau') + L_a[\mathscr{U}_n(\tau')] + N_a[\mathscr{U}_n(\tau')] - f(\tau')].$$

The approximate solution $\mathscr{U}_n(v)$ can be obtained using any zeroth approximation $\mathscr{U}_0(v)$. Thus, for $w(v)$, the VIF is

$$w_{n+1}(v) = w_n(v) - {}_0\mathcal{I}_v^{\delta}[{}_0^c\mathcal{D}_{\tau'}^{\delta;\rho} w_n(\tau') - y_n(\tau')]. \tag{2.6}$$

Then, we demonstrate that $y(v)$ has a VIF of Eq.(2.4). The following equation's correction functional asks us to figure out the Lagrange multiplier. Consider the equation:

$${}_v\mathcal{I}_V^{-\delta}\mathscr{U}(v) + L_a[\mathscr{U}(v)] + N_a[\mathscr{U}(v)] = f(v), \tag{2.7}$$

where $v \in [0, V]$. From Eq.(2.1), right RLFD can be expressed in terms of the left GCTFD, we have

$${}_v\mathcal{I}_V^{-\delta}\mathscr{U}(v) = (-1)^{1-\delta}\ {}_0^c\mathcal{D}_v^{\delta,\rho}\mathscr{U}(v) + X(v) - Y(v),$$

where $\quad X(v) = \dfrac{\mathscr{U}(V)}{\Gamma(1-\delta)(V-v)^{\delta}}, \ Y(v) = \dfrac{(-1)^{1-\delta}\rho^{\delta}}{\Gamma(1-\delta)}\displaystyle\int_0^b (v^{\rho} - s^{\rho})^{-\delta}\mathscr{U}'(s)ds.$

Now the correction functional for Eq.(2.7) can be established using the right RLFI, where $0 < \delta < 1$,

$$\begin{aligned}\mathscr{U}_{n+1}(v) =& \mathscr{U}_n(v) + {}_v\mathcal{I}_V^{\delta}\lambda(v,\tau')[(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau') + X_n(\tau') \\ &- Y_n(\tau') + L_a[\mathscr{U}_n(\tau')] + N_a[\mathscr{U}_n(\tau')] - f(\tau')],\end{aligned} \tag{2.8}$$

thus, $\lambda(v,\tau') = 1$ may be used to determine the Lagrange multiplier, where Y_n, X_n, $L_a[\mathscr{U}_n]$, and $N_a[\mathscr{U}_n]$ are restricted variations. Applying LT on both sides of Eq.(2.8),

$$\begin{aligned}\bar{\mathscr{U}}_{n+1}(s) =& \bar{\mathscr{U}}_n(s) + \mathscr{L}[{}_v\mathcal{I}_V^{\delta}\lambda(v,\tau')[(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau') + X_n(\tau') \\ &- Y_n(\tau') + L_a[\mathscr{U}_n(\tau')] + N_a[\mathscr{U}_n(\tau')] - f(\tau')]].\end{aligned} \tag{2.9}$$

Now,

$$\begin{aligned}
{}_v\mathcal{I}_V^{\delta}\lambda(v,\tau')[(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau')] &= \frac{1}{\Gamma(\delta)}\int_v^V (\tau'-v)^{\delta-1}\lambda(v,\tau')(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau')\,d\tau' \\
&= \frac{1}{\Gamma(\delta)}\int_0^V (\tau'-v)^{\delta-1}\lambda(v,\tau')(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau')\,d\tau' \\
&\quad - \frac{1}{\Gamma(\delta)}\int_0^v (\tau'-v)^{\delta-1}\lambda(v-\tau')(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau')\,d\tau' \\
&= F(v) - \frac{1}{\Gamma(\delta)}\int_0^v (v-\tau')^{\delta-1}(-1)^{\delta-1}\lambda(v-\tau')(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau')\,d\tau' \\
&= F(v) - \frac{1}{\Gamma(\delta)}\int_0^v (v-\tau')^{\delta-1}\lambda(v,\tau')\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau')\,d\tau' \\
&= F(v) - \frac{\lambda(v)v^{\delta-1}}{\Gamma(\delta)} *_0^c \mathcal{D}_v^{\delta,\rho}\mathscr{U}_n(v) \\
&= F(v) - \mathfrak{m}(v) *_0^c \mathcal{D}_v^{\delta,\rho}\mathscr{U}_n(v), \qquad (2.10)
\end{aligned}$$

where $F(v) = \frac{1}{\Gamma(\delta)}\int_0^V (\tau'-v)^{\delta-1}\lambda(v,\tau')(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau')\,d\tau', \mathfrak{m}(v) = \frac{\lambda(v)v^{\delta-1}}{\Gamma(\delta)}$ and $\lambda(v,\tau') = \lambda(v-\tau')$.

Applying LT on both side of Eq.(2.10) is given as

$$\mathscr{L}[{}_v\mathcal{I}_V^{\delta}\lambda(v,\tau')[(-1)^{1-\delta}\ {}_0^c\mathcal{D}_v^{\delta,\rho}\mathscr{U}_n(\tau')]] = F(s) - \bar{\mathfrak{m}}(s)\left[s^{\delta}\bar{\mathscr{U}}_n(s) - \mathscr{U}_n(0^+)s^{\delta-1}\right].$$

On each side of Eq.(2.9), take the derivative of the classical variation δ. Since X_n, Y_n, $L_a[\mathscr{U}_n]$ and $N_a[\mathscr{U}_n]$ are seen as limited variants; hence, $F(v)$ is not the functional of $\mathscr{U}_n(v)$, Eq.(2.9) is computed as

$$\begin{aligned}
\delta\bar{\mathscr{U}}_{n+1}(s) &= \delta\bar{\mathscr{U}}_n(s) + \delta[-\bar{\mathfrak{m}}(s)[s^{\delta}\bar{\mathscr{U}}_n(s) - \bar{\mathscr{U}}_n(0^+)s^{\delta-1}]] \\
&= [1 - \bar{\mathfrak{m}}(s)s^{\delta}]\delta\bar{\mathscr{U}}_n(s).
\end{aligned}$$

Then $1 - \bar{\mathfrak{m}}(s)s^{\delta} = 0, \bar{\mathfrak{m}}(s) = \frac{1}{s^{\delta}}, \mathfrak{m}(v) = \frac{v^{\delta-1}}{\Gamma(\delta)}$. As $\mathfrak{m}(v) = \frac{\lambda(v)v^{\delta-1}}{\Gamma(\delta)}$, $\lambda(v) = 1$.

Then, for Eq.(2.7), the iterations formula is

$$\begin{aligned}
\mathscr{U}_{n+1}(v) &= \mathscr{U}_n(v) + {}_v\mathcal{I}_V^{\delta}[(-1)^{1-\delta}\ {}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}\mathscr{U}_n(\tau') + X_n(\tau') - Y_n(\tau') + L_a[\mathscr{U}_n(\tau')] + N_a[\mathscr{U}_n(\tau')] - f(\tau')] \\
&= \mathscr{U}_n(v) + {}_v\mathcal{I}_V^{\delta}[{}_{\tau'}\mathcal{I}_V^{-\delta}\mathscr{U}_n(\tau') + L_a[\mathscr{U}_n(\tau')] + N_a[\mathscr{U}_n(\tau')] - f(\tau')].
\end{aligned}$$

Thus, VI calculation of $y(v)$ is:

$$y_{n+1}(v) = y_n(v) + {}_v\mathcal{I}_V^{\delta}[{}_{\tau'}\mathcal{I}_V^{-\delta}y_n(\tau') + w_n(\tau') + f(\tau')]. \qquad (2.11)$$

This brings Theorem 2.1's proof to a close. □

Theorem 2.2 *Let $w_0(v) = \phi(v)$, where $\phi(0) = w(0)$. The precise answers $w(v)$ and $y(v)$ can be reached by this $\{w_n(v)\}$ and $\{y_n(v)\}$ sequence that were derived from Theorem 2.1.*

Proof: We will demonstrate in this section that the order in which $\{y_n(v)\}$ and $\{w_n(v)\}$ originated converge to $w(v)$ and $y(v)$ the exact from Theorem 2.1.

It is evidently possible to acquire that

$$\begin{aligned}
w(v) &= w(v) - {}_0\mathcal{I}_v^{\delta}[{}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}w(\tau') - y(\tau')], \\
y(v) &= y(v) + {}_v\mathcal{I}_V^{\delta}[{}_{\tau'}\mathcal{I}_V^{-\delta}y(\tau') + w(\tau') + f(\tau')].
\end{aligned}$$

Let $e_n(v) = w_n(v) - w(v)$, $\epsilon_n(v) = y_n(v) - y(v)$, we can get that

$$e_{n+1}(v) = e_n(v) - {}_0\mathcal{I}_v^{\delta}[{}_0^c\mathcal{D}_{\tau'}^{\delta;\rho}e_n(\tau') - \epsilon_n(\tau')], \qquad (2.12)$$

$$\epsilon_{n+1}(v) = \epsilon_n(v) + {}_v\mathcal{I}_V^{\delta}[\,{}_{\tau'}\mathcal{I}_V^{-\delta}\epsilon_n(\tau') + e_n(\tau')]. \tag{2.13}$$

When $\delta \in (0,1)$, ${}_0\mathcal{I}_v^{\delta}[\,{}_0^c\mathcal{D}_v^{\delta,\rho}w(v)] = w(v) - w(0)$,
and ${}_v\mathcal{I}_V^{\delta}[\,{}_v\mathcal{I}_V^{-\delta}w(v)] = {}_v\mathcal{I}_V^{1-\delta}\,w(v)|_{v=V}\frac{(V-v)^{\delta-1}}{\Gamma(\delta)} - {}_v\mathcal{I}_V^{1+\delta}w(v)|_{v=V}\frac{(V-v)^{-\delta-1}}{\Gamma(-\delta)}$. If $w(v)$ is bounded on $[0,V]$, then

$$\begin{aligned}\left|{}_v\mathcal{I}_V^{1-\delta}w(v)\right| &= \left|\frac{1}{\Gamma(1-\delta)}\int_v^V (s-v)^{-\delta}w(s)ds\right| \le \frac{1}{\Gamma(1-\delta)}\int_v^V (s-v)^{-\delta}\,|w(s)|\,ds \\ &\le \max_{s\in[t,V]}|w(s)|\,\frac{1}{\Gamma(1-\delta)}\int_v^V (s-v)^{-\delta}ds = \max_{s\in[v,V]}|w(s)|\,\frac{1}{\Gamma(1-\delta)}\,\frac{(V-v)^{1-\delta}}{1-\delta}.\end{aligned}$$

When $v = V, \left|{}_v\mathcal{I}_V^{1-\delta}w(v)\right|_{v=V} \le 0$, which means $\left|{}_v\mathcal{I}_V^{1-\delta}w(v)\right| = 0$. Similarly, when $v = V, \left|{}_v\mathcal{I}_V^{1+\delta}w(v)\right|_{v=V} \le 0$, which means $\left|{}_v\mathcal{I}_V^{1+\delta}w(v)\right| = 0$.

It is proved that $e_n(0) = 0$ and ϵ_n is bounded on $[0,V]$. Then Eq.(2.12) and Eq.(2.13) can be made easier to

$$\begin{aligned} e_{n+1}(v) =& {}_0\mathcal{I}_v^{\delta}\epsilon_n(\tau'), \\ \epsilon_{n+1}(v) =& \epsilon_n(v) + {}_v\mathcal{I}_V^{\delta}e_n(\tau').\end{aligned}$$

Taking the absolute value of $e_{n+1}(v)$ and $\epsilon_{n+1}(v)$, we have

$$\begin{aligned}|e_{n+1}(v)| &= \left|{}_0\mathcal{I}_v^{\delta}\epsilon_n(\tau')\right| \\ &= \left|\frac{1}{\Gamma(\delta)}\int_0^v (v-\tau')^{\delta-1}\epsilon_n(\tau')d\tau'\right| \\ &\le \frac{1}{\Gamma(\delta)}\int_0^v (v-\tau')^{\delta-1}\,|\epsilon_n(\tau')|\,d\tau' \\ &\le \frac{1}{\Gamma(\delta)}\int_0^v (v-\tau')^{\delta-1}(\max|\epsilon_n(\tau')|)d\tau' = {}_0\mathcal{I}_v^{\delta}||\epsilon_n(\tau')||. \\ ||e_{n+1}(v)|| &= \max_{v\in[0,1]}|e_{n+1}(v)| \le \max {}_0\mathcal{I}_v^{\delta}||\epsilon_n|| = \max\frac{||\epsilon_n||v^{\delta}}{\Gamma(\delta)\delta} = \frac{||\epsilon_n||V^{\delta}}{\Gamma(\delta)\delta} = {}_0\mathcal{I}_V^{\delta}||\epsilon_n||,\end{aligned}$$

and

$$\begin{aligned}|\epsilon_{n+1}(v)| &= \left|\epsilon_n(v) + {}_v\mathcal{I}_V^{\delta}e_n(\tau')\right| \le |\epsilon_n(v)| + |{}_v\mathcal{I}_V^{\delta}e_n(\tau')| \le |{}_v\mathcal{I}_V^{\delta}e_n(\tau')| \le {}_v\mathcal{I}_V^{\delta}||e_n||. \\ ||\epsilon_{n+1}(v)|| &= \max_{t\in[0,1]}|\epsilon_{n+1}(v)| \le \max {}_v\mathcal{I}_V^{\delta}||e_n|| = \max\frac{||e_n||(V-t)^{\delta}}{\Gamma(\delta)\delta} = {}_0\mathcal{I}_V^{\delta}||e_n||.\end{aligned}$$

Because ${}_a\mathcal{I}_v^{\delta}[\,{}_a\mathcal{I}_v^{\beta}f(v)] = {}_a\mathcal{I}_v^{\delta+\beta}f(v)$ holds if function f is continuous for almost all $v \in [a,b]$ and for any $\delta, \beta > 0$.

$$\begin{aligned}||e_{n+1}(v)|| + ||\epsilon_{n+1}(v)|| &\le {}_0\mathcal{I}_V^{\delta}||\epsilon_n|| + {}_0\mathcal{I}_V^{\delta}||e_n|| = {}_0\mathcal{I}_V^{\delta}(||\epsilon_n|| + ||e_n||) \le {}_0\mathcal{I}_V^{2\delta}(||\epsilon_{n-1}|| + ||e_{n-1}||) \\ &\le \cdots \le {}_0\mathcal{I}_V^{(n+1)\delta}(||\epsilon_0|| + ||e_0||) = \frac{||\epsilon_0|| + ||e_0||}{\Gamma(n\delta+\delta)}\int_0^V (V-s)^{n\delta+\delta-1}ds \\ &= \frac{||\epsilon_0|| + ||e_0||}{\Gamma(n\delta+\delta)}\left[\frac{V^{n\delta+\delta}}{n\delta+\delta}\right].\end{aligned} \tag{2.14}$$

When $\Gamma(n\delta+\delta) \sim \sqrt{2\pi}e^{-n\delta}(n\delta)^{n\delta+\delta-\frac{1}{2}}$, then

$$\begin{aligned}\frac{1}{\Gamma(n\delta+\delta)}\left[\frac{V^{n\delta+\delta}}{n\delta+\delta}\right] &\sim \frac{V^{n\delta+\delta}}{\sqrt{2\pi}e^{-n\delta}(n\delta)^{n\delta+\delta-\frac{1}{2}}}\frac{1}{(n\delta+\delta)}\\ &\sim \frac{V^{n\delta+\delta}.e^{n\delta}}{\sqrt{2\pi}(n\delta)^{n\delta+\delta}}\frac{(n\delta)^{\frac{1}{2}}}{(n\delta+\delta)}\\ &= \frac{1}{e^{\delta}\sqrt{2\pi}}\left(\frac{eV}{n\delta}\right)^{n\delta+\delta}\frac{(n\delta)^{\frac{1}{2}}}{(n\delta+\delta)}. \end{aligned} \tag{2.15}$$

Putting Eq.(2.15) in Eq.(2.14), we get

$$\max_{v\in[0,V]}(|e_{n+1}(v)|+|\epsilon_{n+1}(v)|) \le ||e_{n+1}(v)||+||\epsilon_{n+1}(v)|| \le \frac{||\epsilon_0||+||e_0||}{e^{\delta}\sqrt{2\pi}}\left(\frac{eV}{n\delta}\right)^{n\delta+\delta}\frac{(n\delta)^{\frac{1}{2}}}{(n\delta+\delta)}.$$

Since $||\epsilon_0||+||e_0||$ and V are constants, let $n\to\infty$, then we have

$$\begin{aligned}\lim_{n\to\infty}\max_{v\in[0,V]}(|e_{n+1}(v)|+|\epsilon_{n+1}(v)|) &\le \lim_{n\to\infty}\frac{||\epsilon_0||+||e_0||}{e^{\delta}\sqrt{2\pi}}\left(\frac{eV}{n\delta}\right)^{n\delta+\delta}\frac{(n\delta)^{\frac{1}{2}}}{(n\delta+\delta)}\\ &\le \frac{||\epsilon_0||+||e_0||}{e^{\delta}\sqrt{2\pi}}\lim_{n\to\infty}\left(\frac{eV}{n\delta}\right)^{n\delta+\delta}\frac{(n\delta)^{\frac{1}{2}}}{(n\delta+\delta)}=0.\end{aligned}$$

As n approaches infinity, the sum of these two terms converges to 0. This completes the proof of Theorem 2.2. □

3. Approximation based on ResNets

After the proof of Theorem 2.2, we identify $w_N(v)$ and $y_N(v)$ as the approximate solutions of Eq.(2.2)(a) and Eq.(2.2)(b), respectively. Let's discuss about various concepts and lemmas related to FNNs and ResNets as follows:

Definition 3.1 *(FNNs [18]). The whole collection of FNNs is known as*

$$\mathbb{F} = \cup_{H\in\mathbb{N}}\cup_{\ell_0,\ell_1,\ldots,\ell_H\in\mathbb{N}}(\times_{j=1}^{H}(\mathbb{R}^{\ell_j\times\ell_{j-1}}\times\mathbb{R}^{\ell_j})),$$

and $\mathcal{P}_{\mathfrak{a}},\mathcal{L}_{\mathfrak{a}},\mathcal{I}_{\mathfrak{a}},\mathcal{O}_{\mathfrak{a}}:\mathbb{F}\to\mathbb{N},\mathcal{T}:\mathbb{F}\to(\cup_{H\in\mathbb{N}}\mathbb{N}^H),\mathbb{D}_n:\mathbb{F}\to\mathbb{N}_0$ *are identified as the functions that satisfy* $H\in\mathbb{N}$, $l_0,l_1,\ldots,l_H\in\mathbb{N}$, $\theta=((Z_1,\mathscr{B}_1),(Z_2,\mathscr{B}_2),\ldots,(Z_H,\mathscr{B}_H))\in(\times_{j=1}^{H}(\mathbb{R}^{l_j\times l_{j-1}}\times\mathbb{R}^{l_j}))$ *that* $\mathcal{L}_{\mathfrak{a}}(\theta)=H$, $\mathcal{I}_{\mathfrak{a}}(\theta)=l_0$, $\mathcal{O}_{\mathfrak{a}}(\theta)=l_H$, $\mathcal{P}_{\mathfrak{a}}(\theta)=\sum_{j=1}^{H}l_j(l_{j-1}+1)$, $\mathcal{T}(\theta)=(l_0,l_1,\ldots,l_H)$, *and* $\mathbb{D}_r(\theta)=l_r, r=0,1,\ldots,H$. *The depth of* θ *and the layer size* i *are* $\mathcal{L}_{\mathfrak{a}}(\theta)$ *and* l_i*, respectively.* θ *is referred to as a FNN. The input and output dimensions are* $\mathcal{I}_{\mathfrak{a}}(\theta)$ *and* $\mathcal{O}_{\mathfrak{a}}(\theta)$*, respectively; for* θ *and* $\mathcal{T}(\theta)$*, the complexity is* $\mathcal{P}_{\mathfrak{a}}(\theta)$*, and the structure of* θ *is represented by* $\mathbb{D}_r(\theta)$*.*

Definition 3.2 *(RFNNs [1]). Let's* $a'\in\mathbb{C}(\mathbb{R},\mathbb{R})$. $\mathcal{R}_{a'}$ *is displayed as something that satisfy the desires of all* $w_0\in\mathbb{R}^{\ell_0},w_1\in\mathbb{R}^{\ell_1},\ldots,w_{H-1}\in\mathbb{R}^{\ell_{H-1}}$

$$(\mathcal{R}_{a'}\theta)(w_0)=Z_Hw_{H-1}+\mathscr{B}_H,$$

where $w_j=\mathscr{M}_a(Z_jw_{j-1}+\mathscr{B}_j)$, $j=1,\ldots,H-1$. $\mathcal{R}_{a'}\theta\in\mathbb{C}(\mathbb{R}^{\ell_0},\mathbb{R}^{\ell_1})$. *The function of* a' *is commonly known as the activation function, while the function of* θ *is* $\mathcal{R}_{a'}\theta$*.*

Lemma 3.1 *([18]) If* $(\cdot)\bullet(\cdot)$:$\{(\theta_1,\theta_2)\in\mathbb{F}\times\mathbb{F}|\mathcal{I}_{\mathfrak{a}}(\theta_2)=\mathcal{O}_{\mathfrak{a}}(\theta_1)\}\to\mathbb{F}$ *is represented as the function which occurs for*

$$\theta_1=((\mathcal{Z}_1,\mathcal{B}_1),(\mathcal{Z}_2,\mathcal{B}_2),\ldots,(\mathcal{Z}_{\mathcal{L}_{\mathfrak{a}}},\mathcal{B}_{\mathcal{L}_{\mathfrak{a}}})),$$
$$\theta_2=((Z_1,\mathscr{B}_1),(Z_2,\mathscr{B}_2),\ldots,(Z_H,\mathscr{B}_H)),$$

that

$$\theta_2 \bullet \theta_1 = ((\mathcal{Z}_1, \mathcal{B}_1), (\mathcal{Z}_2, \mathcal{B}_2), \ldots, (\mathcal{Z}_{\mathcal{L}_\mathfrak{a}-1}, \mathcal{B}_{\mathcal{L}_\mathfrak{a}-1}), (Z_1\mathcal{Z}_{\mathcal{L}_\mathfrak{a}}, Z_1\mathcal{B}_{\mathcal{L}_\mathfrak{a}} + \mathscr{B}_1), (Z_2, \mathscr{B}_2), \ldots, (Z_H, \mathscr{B}_H)),$$

then

$$\begin{aligned}\mathcal{T}(\theta_2 \bullet \theta_1) &= (\mathbb{D}_0(\theta_1), \mathbb{D}_1(\theta_1), \ldots, \mathbb{D}_{\mathcal{L}(\theta_1)-1}(\theta_1), \mathbb{D}_1(\theta_2), \mathbb{D}_2(\theta_2), \ldots, \mathbb{D}_{\mathcal{L}_\mathfrak{a}(\theta_2)}(\theta_2)),\\ \mathcal{R}_a(\theta_2 \bullet \theta_1) &= [\mathcal{R}_a(\theta_2)] \circ [\mathcal{R}_a(\theta_1)].\end{aligned}$$

Lemma 3.2 *([18]) Let $r \in \mathbb{N}$, then*

$$\boldsymbol{P}_r : \{(\theta_1, \theta_2, \ldots, \theta_r) \in \mathbb{F} | \mathcal{L}_\mathfrak{a}(\theta_1) = \mathcal{L}_\mathfrak{a}(\theta_2) = \cdots = \mathcal{L}_\mathfrak{a}(\theta_r)\} \to \mathbb{F}$$

is denoted as the function

$$\begin{aligned}\theta_1 &= ((Z_1^1, \mathscr{B}_1^1), (Z_2^1, \mathscr{B}_2^1), \ldots, (Z_H^1, \mathscr{B}_H^1)),\\ \theta_2 &= ((Z_1^2, \mathscr{B}_1^2), (Z_2^2, \mathscr{B}_2^2), \ldots, (Z_H^2, \mathscr{B}_H^2)),\\ &\vdots\\ \theta_r &= ((Z_1^r, \mathscr{B}_1^r), (Z_2^r, \mathscr{B}_2^r), \ldots, (Z_H^r, \mathscr{B}_H^r)).\end{aligned}$$

$$\begin{aligned}\boldsymbol{P}_r(\theta_1, \theta_2, \ldots, \theta_r) = \Bigg(&\left(\begin{pmatrix} Z_1^1 & 0 & \ldots & 0 \\ 0 & Z_1^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & Z_1^r \end{pmatrix}, \begin{pmatrix} \mathscr{B}_1^1 \\ \mathscr{B}_1^2 \\ \vdots \\ \mathscr{B}_1^r \end{pmatrix}\right), \left(\begin{pmatrix} Z_2^1 & 0 & \ldots & 0 \\ 0 & Z_2^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & Z_2^r \end{pmatrix}, \begin{pmatrix} \mathscr{B}_2^1 \\ \mathscr{B}_2^2 \\ \vdots \\ \mathscr{B}_2^r \end{pmatrix}\right),\\ &\ldots, \left(\begin{pmatrix} Z_H^1 & 0 & \ldots & 0 \\ 0 & Z_H^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & Z_H^r \end{pmatrix}, \begin{pmatrix} \mathscr{B}_H^1 \\ \mathscr{B}_H^2 \\ \vdots \\ \mathscr{B}_H^r \end{pmatrix}\right)\Bigg).\end{aligned}$$

Let $\theta = (\theta_1, \theta_2, \ldots, \theta_r) \in \mathbb{F}^r$, then

$$\begin{aligned}\mathcal{T}(\boldsymbol{P}_r(\theta)) &= \left(\sum_{j=1}^r \mathbb{D}_0(\theta_j), \sum_{j=1}^r \mathbb{D}_1(\theta_j), \ldots, \sum_{j=1}^r \mathbb{D}_{\mathcal{L}_\mathfrak{a}(\theta_1)}(\theta_j)\right),\\ \mathcal{R}_a(\boldsymbol{P}_r(\theta))(w_1, w_2, \ldots, w_r) &= (\mathcal{R}_a\theta_1)(w_1), (\mathcal{R}_a\theta_2)(w_2), \ldots, (\mathcal{R}_a\theta_r)(w_r).\end{aligned}$$

Lemma 3.3 *([18]) Let $\theta = ((Z_1, \mathscr{B}_1), (Z_2, \mathscr{B}_2), \ldots, (Z_H, \mathscr{B}_H)), \mathcal{Z} \in \mathbb{R}^{m \times \ell_H}, \mathbb{Z} \in \mathbb{R}^{\ell_0 \times n}$, FNNs $\mathcal{Z} \otimes \theta$ as well as $\theta \otimes \mathcal{Z}$ are indicated correspondingly,*

$$\begin{aligned}\mathcal{Z} \otimes \theta &= ((Z_1, \mathscr{B}_1), (Z_2, \mathscr{B}_2), \ldots, (Z_{H-1}, \mathscr{B}_{H-1}), (\mathcal{Z}Z_H, \mathcal{Z}\mathscr{B}_H)),\\ \theta \otimes \mathbb{Z} &= ((Z_1\mathbb{Z}, \mathscr{B}_1), (Z_2, \mathscr{B}_2), \ldots, (Z_H, \mathscr{B}_H)),\end{aligned}$$

then $\mathcal{R}_a(\mathcal{Z} \otimes \theta) \in \mathbb{C}(\mathbb{R}^{\mathcal{I}_\mathfrak{a}(\theta)}, \mathbb{R}^m), \mathcal{R}_a(\theta \otimes \mathcal{Z}) \in \mathbb{C}(\mathbb{R}^n, \mathbb{R}^{\mathcal{O}_\mathfrak{a}(\theta)})$,

$$\begin{aligned}(\mathcal{R}_a(\mathcal{Z} \otimes \theta))(w) &= \mathcal{Z}(\mathcal{R}_a\theta(w)),\\ (\mathcal{R}_a(\theta \otimes \mathcal{Z}))(w) &= \mathcal{R}_a\theta(\mathbb{Z}w).\end{aligned}$$

Lemma 3.4 *([18]) Let $K \in \mathbb{N}, g_1, g_2, ..., g_K \in \mathbb{R}, \theta_1, \theta_2, \ldots, \theta_K \in \mathbb{F}$ satisfy that $\mathcal{T}(\theta_1) = \mathcal{T}(\theta_2) = \ldots = \mathcal{T}(\theta_K) = (\ell_0, \ell_1, \ldots, \ell_H)$. Then $\exists\, \psi \in \mathbb{F}$ such that $\forall\, w \in \mathbb{R}^{\mathcal{I}_\mathfrak{a}(\theta_1)}$ it holds that $\mathcal{R}_a\psi \in \mathbb{C}(\mathbb{R}^{\mathcal{I}_\mathfrak{a}(\theta_1)}, \mathbb{R}^{\mathcal{O}_\mathfrak{a}(\theta_1)})$, $\mathcal{P}_\mathfrak{a}(\psi) \leq K^2\mathcal{P}_\mathfrak{a}(\theta_1)$ and*

$$(\mathcal{R}_a\psi)(w) = \sum_{i=1}^K g_i[(\mathcal{R}_a\theta_i)(w)], \tag{3.1}$$

where

$$\psi = A_2 \otimes \boldsymbol{P}_K(\theta_1, \theta_2, \dots, \theta_K) \otimes A_1, \tag{3.2}$$

with

$$A_1 = \begin{pmatrix} I_{\ell_0} \\ \vdots \\ I_{\ell_0} \end{pmatrix}, A_2 = (g_1 I_H \dots g_K I_H). \tag{3.3}$$

Next, we define ResNets and discuss some of its qualities. FNNs and general ResNets differ primarily in their use of shortcut connections between layers.

Definition 3.3 *(ResNets [4]) The collection of all ResNets is denoted as*

$$\mathcal{N} := \cup_{n \in \mathbb{N}} \mathcal{N}_n,$$

where $\forall\ n \in \mathbb{N}$,

$$\mathcal{N}_n := \cup_{(d_0, d_1, \dots, d_n) \in \mathbb{N}^n} \left\{ (\Gamma_1, \theta_1, \Gamma_2, \theta_2, \dots, \Gamma_n, \theta_n) \middle| \begin{array}{l} \theta_j \in \mathbb{F}, \mathcal{I}_{\mathfrak{a}}(\theta_j) = d_{j-1}, \mathcal{O}_{\mathfrak{a}}(\theta_j) = d_j, \\ \Gamma_j \in \mathbb{R}^{d_j \times d_{j-1}}, \forall j \in \{1, 2, \dots, n\} \end{array} \right\}.$$

Let $n \in \mathbb{N}, d_0, d_1, \dots, d_n \in \mathbb{N}$, *and let the ResNets* $\zeta \in \mathcal{N}$ *be given by*

$$\zeta = (\Gamma_1, \theta_1, \Gamma_2, \theta_2, \dots, \Gamma_n, \theta_n),$$

for every $j \in 1, 2, ..., n$ *we have* $\mathcal{I}_{\mathfrak{a}}(\theta_j) = d_{j-1}$, $\mathcal{O}_{\mathfrak{a}}(\theta_j) = d_j$ *and* $\Gamma_j \in \mathbb{R}^{d_j \times d_{j-1}}$. *We call* $\theta = (\theta_1, \theta_2, ..., \theta_n) \in \mathbb{F}^n$ *the residual blocks of* ζ, *and* Γ_j *is referred to as the shortcut. Also,* $\mathscr{L}_{\mathfrak{r}}, \mathscr{P}_{\mathfrak{r}} : \mathcal{N} \to \mathbb{N}$ *are the length and complexity of* ζ *by the functions that satisfy* $\mathscr{L}_{\mathfrak{r}}(\zeta) = n$ *and* $\mathscr{P}_{\mathfrak{r}}(\zeta) = \sum_{i=1}^{n} (\mathcal{P}_{\mathfrak{a}}(\theta_i) + d_i d_{i-1})$, *respectively. Moreover* $\mathscr{D}_{\mathfrak{r}} : \mathcal{N} \to \cup_{H \in \mathbb{N}} \mathbb{N}^H$ *are denoted by the function that fulfills* $\mathscr{D}_{\mathfrak{r}}(\zeta) = (d_0, d_1, ..., d_{\mathscr{L}_{\mathfrak{r}}(\zeta)})$.

Definition 3.4 *(RResNets [4]).* $a \in \mathbb{C}(\mathbb{R}, \mathbb{R})$ *and* $\mathscr{R}_a$ *is the function it is applicable to all* $\zeta = (\Gamma_1, \theta_1, \Gamma_2, \theta_2, \dots,$
$\Gamma_r, \theta_r) \in \mathcal{N}$, *and* $w_0 \in \mathbb{R}^{\mathcal{I}_{\mathfrak{a}}(\theta_1)}, w_1 \in \mathbb{R}^{\mathcal{I}_{\mathfrak{a}}(\theta_2)} = \mathbb{R}^{\mathcal{O}_{\mathfrak{a}}(\theta_1)}, \dots, w_r \in \mathbb{R}^{\mathcal{O}_{\mathfrak{a}}(\theta_r)}$ *with* $w_i = \Gamma_i w_{i-1} + \mathcal{R}_a \theta_i(w_{i-1})$ $\forall i \in \{1, 2, ..., r\}$ *that* $(\mathscr{R}_a \zeta)(w_0) = w_r$. *The activation function is denoted by* a, *while the function* $\mathscr{R}_a \zeta \in \mathbb{C}(\mathbb{R}^{\mathcal{I}_{\mathfrak{a}}(\theta_1)}, \mathbb{R}^{\mathcal{O}_{\mathfrak{a}}(\theta_r)})$ *is called the RResNets* ζ.

Lemma 3.5 *([4]) Let* $r, m \in \mathbb{N}$, *the ResNets* $\zeta^1 = (\Gamma_1^1, \theta_1^1, \Gamma_2^1, \theta_2^1, \dots, \Gamma_r^1, \theta_r^1)$, $\zeta^2 = (\Gamma_1^2, \theta_1^2, \Gamma_2^2, \theta_2^2, \dots, \Gamma_m^2, \theta_m^2)$ *satisfy* $\mathcal{O}_{\mathfrak{a}}\left(\theta_r^1\right) = \mathcal{I}_{\mathfrak{a}}\left(\theta_1^2\right)$, *the composition of* ζ^1 *and* ζ^2 *is defined as*

$$\zeta^2 \bullet \zeta^1 = (\Gamma_1^1, \theta_1^1, \Gamma_2^1, \theta_2^1, \dots, \Gamma_r^1, \theta_n^1, \Gamma_1^2, \theta_1^2, \Gamma_2^2, \theta_2^2, \dots, \Gamma_m^2, \theta_m^2) \in \mathcal{N},$$

subsequently the subsequent characteristics are true

$$\begin{aligned} \mathscr{R}_a(\zeta^2 \bullet \zeta^1) =& \mathscr{R}_a(\zeta^2) \circ \mathscr{R}_a(\zeta^1), \\ \mathscr{P}_a(\zeta^2 \bullet \zeta^1) =& \mathscr{P}_a(\zeta^2) + \mathscr{P}_a(\zeta^1). \end{aligned}$$

Lemma 3.6 *([4])Let* $\mathscr{U}, n \in \mathbb{N}$, *the ResNets* $\zeta^j = (\Gamma_1^j, \theta_1^j, \Gamma_2^j, \theta_2^j, \dots, \Gamma_n^j, \theta_n^j) \in \mathcal{N}$ *for all* $j \in \{1, 2, ..., \mathscr{U}\}$. *Assume that* $\mathcal{T}\left(\theta_i^1\right) = \dots = \mathcal{T}\left(\theta_i^{\mathscr{U}}\right)$ *for all* $i \in \{1, 2, ..., n\}$. *Then* $\mathbb{P}_{\mathscr{U}}$ *is defined as*

$$\begin{aligned} \mathbb{P}_{\mathscr{U}}(\zeta^1, \zeta^2, \dots, \zeta^{\mathscr{U}}) = \Bigg(& \begin{pmatrix} \Gamma_1^1 & 0 & \dots & 0 \\ 0 & \Gamma_1^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Gamma_1^{\mathscr{U}} \end{pmatrix}, \boldsymbol{P}_{\mathscr{U}}(\theta_1^1, \theta_1^2, \dots, \theta_1^{\mathscr{U}}), \begin{pmatrix} \Gamma_2^1 & 0 & \dots & 0 \\ 0 & \Gamma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Gamma_2^{\mathscr{U}} \end{pmatrix}, \\ & \boldsymbol{P}_{\mathscr{U}}(\theta_2^1, \theta_2^2, \dots, \theta_2^{\mathscr{U}}), \dots, \begin{pmatrix} \Gamma_n^1 & 0 & \dots & 0 \\ 0 & \Gamma_n^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Gamma_n^{\mathscr{U}} \end{pmatrix}, \boldsymbol{P}_{\mathscr{U}}(\theta_n^1, \theta_n^2, \dots, \theta_n^{\mathscr{U}}) \Bigg). \end{aligned}$$

$\mathbb{P}_{\mathscr{U}}$ is still a ResNets. Let $w_0^1, w_0^2, \ldots, w_0^{\mathscr{U}} \in \mathbb{R}^{\mathcal{I}_{\mathfrak{a}}(\theta_1^1)}$, then

$$\begin{aligned}\mathscr{R}_a\left(\mathbb{P}_{\mathscr{U}}(\zeta^1, \zeta^2, \ldots, \zeta^{\mathscr{U}})\right)\left(w_0^1, w_0^2, \ldots, w_0^{\mathscr{U}}\right) = \left((\mathscr{R}_a\zeta^1)(w_0^1), (\mathscr{R}_a\zeta^2)(w_0^2), \ldots, (\mathscr{R}_a\zeta^{\mathscr{U}})(w_0^{\mathscr{U}})\right),\\ \mathscr{P}_{\mathfrak{r}}\left(\mathbb{P}_{\mathscr{U}}(\zeta^1, \zeta^2, \ldots, \zeta^{\mathscr{U}})\right) \leq \mathscr{U}^2 \mathscr{P}_{\mathfrak{r}}(\zeta^1).\end{aligned}$$

The claim asserts a parallelized ResNets realization is equivalent to the total of its realizations of individual ResNets that make up the ResNets.

Proposition 3.1 *Let $\mathscr{U}, n \in \mathbb{N}$ and let $\zeta^j := (\Gamma_1^j, \theta_1^j, \Gamma_2^j, \theta_2^j, \ldots, \Gamma_n^j, \theta_n^j) \in \mathcal{N}$, $j \in 1, 2, ..., \mathscr{U}$, satisfy $\mathcal{T}\left(\theta_i^1\right) = \cdots = \mathcal{T}\left(\theta_i^{\mathscr{U}}\right)$ for all $i \in \{1, 2, ..., n\}$, i.e., all the ith residual blocks have the same architecture. Then there exists a ResNets ψ such that for all $w \in \mathbb{R}^{\mathcal{I}_{\mathfrak{a}}(\theta_1^1)}$,*

$$(\mathscr{R}_a\psi)(w) = \left((\mathscr{R}_a\zeta^1)(w), (\mathscr{R}_a\zeta^2)(w), \ldots, (\mathscr{R}_a\zeta^{\mathscr{U}})(w)\right),$$

and ψ's complexity, which meets the following property: $\mathscr{P}_{\mathfrak{r}}(\psi) \leq \mathscr{U}^2 \mathscr{P}_{\mathfrak{r}}(\zeta^1)$.

Proof: Let $d_0, d_1, ..., d_n \in \mathbb{N}\ \forall\ j \in \{1, 2, ..., u\}, \mathscr{D}_{\mathfrak{r}}(\zeta^j) = (d_0, d_1, ..., d_n)$. Set

$$\begin{aligned}\psi := \Bigg(&\left(\begin{pmatrix}\Gamma_1^1 & 0 & \ldots & 0\\ 0 & \Gamma_1^2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \Gamma_1^{\mathscr{U}}\end{pmatrix} A, \mathbf{P}_{\mathscr{U}}(\theta_1^1, \theta_1^2, \ldots, \theta_1^{\mathscr{U}}) \otimes A, \begin{pmatrix}\Gamma_2^1 & 0 & \ldots & 0\\ 0 & \Gamma_2^2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \Gamma_2^{\mathscr{U}}\end{pmatrix}, \mathbf{P}_{\mathscr{U}}(\theta_2^1, \theta_2^2, \ldots, \theta_2^{\mathscr{U}}),\right.\\ &\left.\ldots, \begin{pmatrix}\Gamma_n^1 & 0 & \ldots & 0\\ 0 & \Gamma_n^2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \Gamma_n^{\mathscr{U}}\end{pmatrix}, \mathbf{P}_{\mathscr{U}}(\theta_n^1, \theta_n^2, \ldots, \theta_n^{\mathscr{U}})\right),\end{aligned}$$

where $A \in \mathbb{R}^{\mathscr{U} d_0 \times d_0}$ satisfy

$$A = \begin{pmatrix}I_{d_0}\\ \vdots\\ I_{d_0}\end{pmatrix}.$$

Then, setting

$$\begin{aligned}z_0 =& w,\\ z_1 =& \begin{pmatrix}\Gamma_1^1 & 0 & \ldots & 0\\ 0 & \Gamma_1^2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \Gamma_1^{\mathscr{U}}\end{pmatrix}(Az_0) + [\mathcal{R}_a\left(\mathbf{P}_{\mathscr{U}}(\theta_n^1, \theta_n^2, \ldots, \theta_n^{\mathscr{U}}) \otimes A\right)](z_0)\\ =& \begin{pmatrix}\Gamma_1^1 & 0 & \ldots & 0\\ 0 & \Gamma_1^2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \Gamma_1^{\mathscr{U}}\end{pmatrix}(Az_0) + \mathcal{R}_a\left(\mathbf{P}_{\mathscr{U}}(\theta_n^1, \theta_n^2, \ldots, \theta_n^{\mathscr{U}})\right)(Az_0)\\ =& \begin{pmatrix}\Gamma_1^1 & 0 & \ldots & 0\\ 0 & \Gamma_1^2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \Gamma_1^{\mathscr{U}}\end{pmatrix}\begin{pmatrix}w\\ \vdots\\ w\end{pmatrix} + \mathcal{R}_a[\mathbf{P}_{\mathscr{U}}(\theta_n^1, \theta_n^2, \ldots, \theta_n^{\mathscr{U}})]\begin{pmatrix}w\\ \vdots\\ w\end{pmatrix}.\end{aligned}$$

It is in line with the circumstances in Lemma 3.6, thus $(\mathscr{R}_a\psi)(w) = \left((\mathscr{R}_a\zeta^1)(w), (\mathscr{R}_a\zeta^2)(w), \ldots (\mathscr{R}_a\zeta^{\mathscr{U}})(w)\right)$. Let $\mathcal{T}\left(\theta_i^1\right) = \mathcal{T}\left(\theta_i^2\right) = \cdots = \mathcal{T}\left(\theta_i^{\mathscr{U}}\right) = \left(d_{i-1}, \ell_1^i, \ell_2^i, \ldots, d_i\right) \in \mathbb{N}^{\mathcal{L}(\theta_i^1)}\ \forall\, i \in \{1, 2, \ldots, n\}$.

$$
\begin{aligned}
\mathscr{P}(\psi) =& \sum_{k=2}^{n}[\mathcal{P}_{\mathfrak{a}}\left(\mathbf{P}_{\mathscr{U}}(\theta_k^1, \theta_k^2, \ldots, \theta_k^{\mathscr{U}})\right) + \mathscr{U} d_k \mathscr{U} d_{k-1}] + \mathcal{P}_{\mathfrak{a}}\left(\mathbf{P}_{\mathscr{U}}(\theta_1^1, \theta_1^2, \ldots, \theta_1^{\mathscr{U}}) \otimes A\right) + d_0 \mathscr{U} d_1 \\
=& \mathscr{U}\ell_1^1 d_0 + \mathscr{U}\ell_1^1 + \left[\sum_{j=2}^{\mathcal{L}_{\mathfrak{a}}(\theta_1^1)-1} (\mathscr{U}\ell_j^1 \mathscr{U}\ell_{j-1}^1 + \mathscr{U}\ell_j^1)\right] + \mathscr{U} d_1 \mathscr{U}\ell^1_{\mathcal{L}_{\mathfrak{a}}(\theta_1^1)-1} + \mathscr{U} d_1 + d_0 \mathscr{U} d_1 \\
&+ \sum_{k=2}^{n}[\mathcal{P}_{\mathfrak{a}}\left(\mathbf{P}_{\mathscr{U}}(\theta_k^1, \theta_k^2, \ldots, \theta_k^{\mathscr{U}})\right) + \mathscr{U} d_k \mathscr{U} d_{k-1}] \\
\leq& \mathscr{U}^2 \mathcal{P}_{\mathfrak{a}}\left(\theta_i^1\right) + \mathscr{U}^2 d_1 d_0 + \sum_{k=2}^{n}[\mathscr{U}^2 \mathcal{P}_{\mathfrak{a}}\left(\theta_k^1\right) + \mathscr{U}^2 d_k d_{k-1}] \\
\leq& \mathscr{U}^2[\mathcal{P}_{\mathfrak{a}}\left(\theta_i^1\right) + d_1 d_0 + \sum_{k=2}^{n}[\mathcal{P}_{\mathfrak{a}}\left(\theta_k^1\right) + d_k d_{k-1}]] = \mathscr{U}^2 \mathscr{P}_{\mathfrak{r}}(\zeta^1).
\end{aligned}
$$

This completes the proof. □

Theorem 3.1 *The approximate answers to Eqs. (2.2)(a) and (b) are $w_N(v)$ and $y_N(v)$, respectively, which were derived from Eq.(2.3) and Eq.(2.4). It is then possible to obtain $w_N(v)$ and $y_N(v)$ using a ResNets. The complexity of the ResNets increases multinomially in K, which is the quantity of points obtained in $[0, V]$, and N, which is the approximate solution's iteration count.*

Proof: After studying ResNets and FNNs, we proceed to demonstrate Theorem 3.1. To be more precise, it implies we demonstrate the existence of a ResNets $\mho$, whose input is $(w_0\ y_0)$ and $(w_N\ y_N)$ being the output, where $w_r = (w_r(v_0), ..., w_r(v_{K+1}))^{T'}$, $y_r = (y_r(v_0), ..., y_r(v_{K+1}))^{T'}$, $r = 0, 1, ..., N$. On the interval $[0, V]$, K points are noted, $\Delta v = \frac{V}{K+1}, v_j = j\Delta v$, $j = 0, 1, \ldots, K+1$. K is a massive number. This is the structure of the proof.

(i) We demonstrate how $(w_r\ y_r)$ is converted to w_{n+1}, $r = 0, 1, ..., N-1$, via the ResNets ϖ.

(ii) We outline how to build the ResNets $\varkappa$, which takes $r = 0, 1, ..., N-1$ and transforms (w_r, y_r) into y_{r+1}.

(iii) We explain the construction of $\mho$ via ϖ and $\varkappa$.

The three items mentioned above are then independently verified.
(i): Initially, we demonstrate the conversion of $(w_r\ y_r)$ to w_{r+1} using a ResNets ϖ. To be more specific, ϖ's input is $(wy)_r = (w_r(v_0), \ldots, w_r(v_{K+1}), y_r(v_0), \ldots, y_r(v_{K+1}))^{T'}$ and the output is $(w_{r+1}(v_0), \ldots, w_{r+1}(v_{K+1}))^{T'}$.
Figure 1 depicts ResNets ϖ's structure. A shortcut Γ' and four FNNs, $\zeta_1, \theta_{sub}, \theta_{LRI}$, and θ_o, make up ResNets ϖ. Next, we demonstrate a five-step method for creating ϖ.

Step 1 (Constructing ζ_1): ζ_1 is composed with FNN θ_{Caputo} and FNN θ_I and we first describe how to construct θ_{Caputo}.

Proposition 3.2 *There exists a FNN θ_{Caputo} that $(\mathcal{R}_a\theta_{Caputo})(w_r) = {}_0^c\mathcal{D}_v^{\delta,\rho} w_r$.*

Proof: If $\delta \in (0, 1)$, the GCTFD can be discretized by L1 method. With a step size of dx=h, discretize

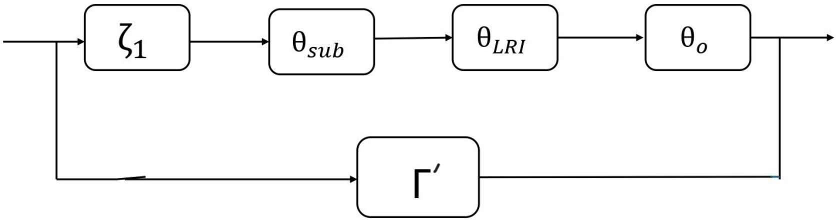

Figure 1: ResNets ϖ's structure

the interval into n points as $v_1, \ldots, v_r$ and consider $v = v_r$.

$$\begin{aligned}
{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v) &= \frac{\rho^{\delta-m+1}}{\Gamma(m-\delta)} \left(\int_0^v t^{\rho-1} \left(v^\rho - t^\rho\right)^{m-\delta-1} \left(t^{1-\rho} \frac{d}{dt} \right)^m w_r(t) \right) dt \\
&= \frac{\rho^{\delta}}{\Gamma(1-\delta)} \left(\int_0^v t^{\rho-1} \left(v^\rho - t^\rho\right)^{-\delta} \left(t^{1-\rho} \frac{d}{dt} \right) w_r(t) \right) dt \\
&= \frac{\rho^{\delta}}{\Gamma(1-\delta)} \int_0^v \left(v^\rho - t^\rho\right)^{-\delta} w_r'(t) dt \\
&= \frac{\rho^{\delta}}{\Gamma(1-\delta)} \int_0^{v_r} \left(v_r^\rho - t^\rho\right)^{-\delta} w_r'(t) dt \\
&= \frac{\rho^{\delta}}{\Gamma(1-\delta)} \left(\int_0^{v_1} \left(v_r^\rho - t^\rho\right)^{-\delta} w_r'(t) dt + \cdots + \int_{v_{n-1}}^{v_r} \left(v_r^\rho - t^\rho\right)^{-\delta} w_r'(t) dt \right) \\
&= \frac{\rho^{\delta}}{\Gamma(1-\delta)} \sum_{j=0}^{r-1} \int_{v_j}^{v_{j+1}} \left(v_r^\rho - t^\rho\right)^{-\delta} w_r'(t) dt.
\end{aligned}$$

Now we replace first order derivative by forward difference quotient,

$$\begin{aligned}
{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v) &= \frac{\rho^{\delta}}{\Gamma(1-\delta)} \sum_{j=0}^{r-1} \int_{v_j}^{v_{j+1}} \left(v_r^\rho - t^\rho\right)^{-\delta} \left(\frac{w_r(v_{j+1}) - w_r(v_j)}{h} \right) dt \\
&= \frac{\rho^{\delta}}{h\Gamma(1-\delta)} \sum_{j=0}^{r-1} \left(w_r(v_{j+1}) - w_r(v_j)\right) \int_{v_j}^{v_{j+1}} \left(v_r^\rho - t^\rho\right)^{-\delta} dt \\
&= \frac{-\rho^{\delta-1}}{h(1-\delta)\Gamma(1-\delta)} \sum_{j=0}^{r-1} \left(w_r(v_{j+1}) - w_r(v_j)\right) \left(\frac{\left(v_r^\rho - t^\rho\right)^{1-\delta}}{t^{\rho-1}} \right)_{v_j}^{v_{j+1}}.
\end{aligned}$$

By gamma property $\Gamma(2-\delta) = (1-\delta)\Gamma(1-\delta)$ and assuming $v_r = rh, v_j = jh$ and $v_{j+1} = (j+1)h$, then we get

$$\begin{aligned}
({}_0^c\mathcal{D}_v^{\delta,\rho}w_r)(v) &= \frac{-\rho^{\delta-1}}{h\Gamma(2-\delta)}\sum_{j=0}^{r-1}(w_r(v_{j+1})-w_r(v_j))\left(\frac{(v_r^\rho-v_{j+1}^\rho)^{1-\delta}}{v_{j+1}^{\rho-1}}-\frac{(v_r^\rho-v_j^\rho)^{1-\delta}}{v_j^{\rho-1}}\right)\\
&= \frac{\rho^{\delta-1}}{h\Gamma(2-\delta)}\sum_{j=0}^{r-1}(w_r(v_{j+1})-w_r(v_j))\left(\frac{-\left[(rh)^\rho-((j+1)h)^\rho\right]^{1-\delta}}{((j+1)h)^{\rho-1}}+\frac{\left[(rh)^\rho-(jh)^\rho\right]^{1-\delta}}{(jh)^{\rho-1}}\right)\\
&= \frac{\rho^{\delta-1}h^{1-\delta\rho}}{h\Gamma(2-\delta)}\sum_{j=0}^{r-1}(w_r(v_{j+1})-w_r(v_j))\left(\frac{-(r^\rho-(j+1)^\rho)^{1-\delta}}{(j+1)^{\rho-1}}+\frac{(r^\rho-j^\rho)^{1-\delta}}{j^{\rho-1}}\right)\\
({}_0^c\mathcal{D}_v^{\delta,\rho}w_r)(v) &= \frac{\rho^{\delta-1}h^{-\delta\rho}}{\Gamma(2-\delta)}\sum_{j=0}^{r-1}w_j\left(w_r(v_{j+1})-w_r(v_j)\right),\\
{}_0^c\mathcal{D}_v^{\delta,\rho}w_r(v_{j+1}) &\approx \frac{\rho^{\delta-1}h^{-\delta\rho}}{\Gamma(2-\delta)}\sum_{j=0}^{r-1}w_j\left(w_r(v_{j+1})-w_r(v_j)\right),
\end{aligned}$$

where $w_j=\frac{-(n^\rho-(j+1)^\rho)^{1-\delta}}{(j+1)^{\rho-1}}+\frac{(n^\rho-j^\rho)^{1-\delta}}{j^{\rho-1}}$.

$$Z_{Caputo}=$$

$$\frac{\rho^{\delta-1}h^{-\delta\rho}}{\Gamma(2-\delta)}\cdot\begin{pmatrix}0&0&0&\dots&0&0\\0&w_0&0&\dots&0&0\\0&w_1&w_0&\dots&0&0\\\vdots&\vdots&\vdots&&\vdots&\vdots\\0&w_{K-1}&w_{K-2}&\dots&w_0&0\\0&w_K&w_{K-1}&\dots&w_1&w_0\end{pmatrix}\cdot\begin{pmatrix}0&0&0&\dots&0&0\\-1&1&0&\dots&0&0\\0&-1&1&\dots&0&0\\\vdots&\vdots&\vdots&&\vdots&\vdots\\0&0&0&\dots&1&0\\0&0&0&\dots&-1&1\end{pmatrix}\in\mathbb{R}^{(K+2)\times(K+2)},$$

$\mathscr{B}_{Caputo}=(0,\dots,0)^{T'}\in\mathbb{R}^{(K+2)}$ such that

$$\begin{pmatrix}{}_0^c\mathcal{D}_v^{\delta,\rho}w_i(v_0)\\\vdots\\{}_0^c\mathcal{D}_v^{\delta,\rho}w_i(v_{K+1})\end{pmatrix}=Z_{Caputo}\begin{pmatrix}w_i(v_0)\\\vdots\\w_i(v_{K+1})\end{pmatrix}+\mathscr{B}_{Caputo}.$$

When θ_{Caputo} receives an input of w_r, then the output is

$$\left({}_0^c\mathcal{D}_v^{\delta,\rho}w_r(v_0),\dots,{}_0^c\mathcal{D}_v^{\delta,\rho}w_r(v_{K+1})\right)^{T'}.$$

□

Of certainly, there is a FNN $\theta_1=(Z_{\mathscr{I}},\mathscr{B}_{\mathscr{I}})$ where $Z_{\mathscr{I}}=\mathscr{I}\in\mathbb{R}^{(K+2)\times(K+2)}$, $\mathscr{B}_{\mathscr{I}}=(0,...,0)^{T'}\in\mathbb{R}^{(K+2)}$ and $(\mathcal{R}_a\theta_1)(y_r)=y_r$ resulting in the input and output being both y_r.
From Lemma 3.2 $\mathcal{L}_{\mathfrak{a}}(\theta_{Caputo})=\mathcal{L}_{\mathfrak{a}}(\theta_1)=1$, then there exists a FNN $\zeta_1=\mathbf{P}_2(\theta_{Caputo},\theta_1)$, $\mathcal{T}(\zeta_1)$=$(2(K+2),2(K+2))$. Definition 3.1 provides an explanation of $\mathcal{L}_{\mathfrak{a}}$ and $\mathcal{T}$. Let $\zeta_1=((Z_1,\mathscr{B}_1))$, then

$$Z_1=\begin{pmatrix}Z_{Caputo}&0\\0&Z_{\mathscr{I}}\end{pmatrix},\mathscr{B}_1=\begin{pmatrix}\mathscr{B}_{Caputo}\\\mathscr{B}_{\mathscr{I}}\end{pmatrix}.$$

When the input of ζ_1 is $(wy)_r$, then the output is

$$\left({}_0^c\mathcal{D}_v^{\delta,\rho}w_r(v_0),\dots,{}_0^c\mathcal{D}_v^{\delta,\rho}w_r(v_{K+1}),y_r(v_0),\dots,y_r(v_{K+1})\right)^{T'}.$$

Step 2 (Constructing θ_{sub}): Set FNN $\theta_{sub} = (Z_{sub}, \mathscr{B}_{sub})$, where $Z_{sub} = (\mathscr{I} - \mathscr{I}) \in \mathbb{R}^{(K+2)\times 2(K+2)}$ and $\mathscr{B}_{sub} = (0, \ldots, 0)^{T'} \in \mathbb{R}^{(K+2)}$ then $(\mathcal{R}_a \theta_{sub})(w, y) = w - y, \forall w, y \in \mathbb{R}^{(K+2)}$.
When the input of θ_{sub} is

$$\left({}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0), \ldots, {}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}), y_r(v_0), \ldots, y_r(v_{K+1})\right)^{T'},$$

then the output is

$$\left({}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0) - y_r(v_0), \ldots, {}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}) - y_r(v_{K+1})\right)^{T'}.$$

Step 3 (Constructing θ_{LRI}):

Proposition 3.3 *There exists a FNN θ_{LRI} that $(\mathcal{R}_a \theta_{LRI})(h) = {}_0\mathcal{I}_v^{\delta} h$, $\forall h = (h(v_0), \ldots, h(v_{K+1}))^{T'}$.*

Proof: One way to discretize the left RLFI is as

$$\begin{aligned} {}_0\mathcal{I}_v^{\delta} h(v_j) &= \frac{1}{\Gamma(\delta)} \int_0^{v_j} (v_j - \tau)^{\delta - 1} h(\tau)\, d\tau \\ &\approx \frac{\Delta v}{\Gamma(\delta)} [v_j^{\delta-1} h(0) + (v_j - v_1)^{\delta-1} h(v_1) + (v_j - v_2)^{\delta-1} h(v_2) + \cdots + (v_j - v_{j-1})^{\delta-1} h(v_{j-1})]. \end{aligned}$$

Then, a FNN is present $\theta_{LRI} = (Z_{LRI}, \mathscr{B}_{LRI})$ with the input $(h(v_0), \ldots, h(v_{K+1}))^{T'}$ and the output $({}_0\mathcal{I}_v^{\delta} h(v_0), \ldots, {}_0\mathcal{I}_v^{\delta} h(v_{K+1}))^{T'}$, where

$$Z_{LRI} = \frac{\Delta v}{\Gamma(\delta)} \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ v_1^{\delta-1} & 0 & 0 & 0 & \cdots & 0 & 0 \\ v_2^{\delta-1} & (v_2 - v_1)^{\delta-1} & 0 & 0 & \cdots & 0 & 0 \\ v_3^{\delta-1} & (v_3 - v_1)^{\delta-1} & (v_3 - v_2)^{\delta-1} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ v_K^{\delta-1} & (v_K - v_1)^{\delta-1} & (v_K - v_2)^{\delta-1} & (v_K - v_3)^{\delta-1} & \cdots & 0 & 0 \\ V^{\delta-1} & (V - v_1)^{\delta-1} & (V - v_2)^{\delta-1} & (V - v_3)^{\delta-1} & \cdots & (V - v_K)^{\delta-1} & 0 \end{pmatrix} \in \mathbb{R}^{(K+2)\times(K+2)},$$

$\mathscr{B}_{LRI} = (0, \ldots, 0)^{T'} \in \mathbb{R}^{(K+2)}$ such that

$$\begin{pmatrix} {}_0\mathcal{I}_v^{\delta} h(v_0) \\ \vdots \\ {}_0\mathcal{I}_v^{\delta} h(v_{K+1}) \end{pmatrix} = Z_{LRI} \begin{pmatrix} h(v_0) \\ \vdots \\ h(v_{K+1}) \end{pmatrix} + \mathscr{B}_{LRI}.$$

Hence proved. □

When the input of θ_{LRI} is

$$\left({}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0) - y_r(v_0), \ldots, {}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}) - y_r(v_{K+1})\right)^{T'},$$

then the output is

$$\left({}_0\mathcal{I}_v^{\delta} \left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0) - y_r(v_0)\right], \ldots, {}_0\mathcal{I}_v^{\delta} \left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}) - y_r(v_{K+1})\right]\right)^{T'}.$$

Step 4 (Developing θ_o): Let FNN $\theta_o = (Z_o, \mathscr{B}_o)$, where $Z_o = -(\mathscr{I}) \in \mathbb{R}^{(K+2)\times(K+2)}$, $\mathscr{B}_o = (0, ..., 0)^{T'} \in \mathbb{R}^{(K+2)}$, then $(\mathcal{R}_a \theta_o)(w) = -w, \forall w \in \mathbb{R}^{(K+2)}$. Thus when the input of θ_o is

$$\left({}_0\mathcal{I}_v^{\delta} \left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0) - y_r(v_0)\right], \ldots, {}_0\mathcal{I}_v^{\delta} \left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}) - y_r(v_{K+1})\right]\right)^{T'},$$

then the output is

$$\left(-{}_0\mathcal{I}_v^{\delta} \left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0) - y_r(v_0)\right], \ldots, -{}_0\mathcal{I}_v^{\delta} \left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}) - y_r(v_{K+1})\right]\right)^{T'}.$$

Step 5 (Constructing ϖ):The notation for ResNets ϖ is $\varpi = (\Gamma', \Omega)$, $\Gamma' = (\mathscr{I}, 0) \in \mathbb{R}^{(K+2)\times 2(K+2)}$ with $\mathscr{I}, 0 \in \mathbb{R}^{(K+2)\times(K+2)}$ where 0 is the null matrix. An FNN can be inferred from Lemma 3.1. $\Omega = \theta_o \bullet \theta_{LRI} \bullet \theta_{sub} \bullet \zeta_1$ when the input of Ω is $(wy)_r$, then the output is

$$\left(-{}_0\mathcal{I}_v^{\delta}\left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0) - y_r(v_0)\right], \dots, -{}_0\mathcal{I}_v^{\delta}\left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}) - y_r(v_{K+1})\right]\right)^{T'},$$

where $\Omega = (Z_o\Omega, \mathscr{B}_\Omega), Z_\Omega = Z_o Z_{LRI} Z_{sub} Z_1 \in \mathbb{R}^{(K+2)\times 2(K+2)}, \mathscr{B}_\Omega = Z_o\left[Z_{LRI}(Z_{sub}\mathscr{B}_1 + \mathscr{B}_{sub}) + \mathscr{B}_{LRI}\right] + \mathscr{B}_o \in \mathbb{R}^{(K+2)}$ and $\mathcal{T}(\Omega) = (2(K+2), K+2)$. Thus $(\mathscr{R}_a\varpi)(w) = \Gamma' w + \mathcal{R}_a\Omega(w), \forall w \in \mathbb{R}^{2(K+2)}$ when the input of ϖ is $(wy)_r$, then the output is

$$\left(w_r(v_0) - {}_0\mathcal{I}_v^{\delta}\left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_0) - y_r(v_0)\right], \dots, w_r(v_{K+1}) - {}_0\mathcal{I}_v^{\delta}\left[{}_0^c\mathcal{D}_v^{\delta,\rho} w_r(v_{K+1}) - y_r(v_{K+1})\right]\right)^{T'}.$$

Furthermore, ϖ's complexity is

$$\begin{aligned}\mathscr{P}_{\mathfrak{r}}(\varpi) &= 2(K+2)\cdot(K+2) + (K+2) + 2(K+2)\cdot(K+2)\\ &= (K+2)\left[4(K+2)+1\right].\end{aligned} \tag{3.4}$$

(ii): Following the construction of ϖ, we will present the processes for ResNets $\varkappa$ where $\varkappa$ converts (w_r, y_r) to y_{r+1}. Particularly, the input of $\varkappa$ is $(wy)_r$, then the output is

$$\left(y_{r+1}(v_0), \dots, y_{r+1}(v_{K+1})\right)^{T'}.$$

The structure of ResNets $\varkappa$ is seen in Figure 2. Five FNNs, $\zeta_3, \theta_{add}, \theta_f, \theta_{RRI}$, and θ_o, plus a shortcut Γ', make up ResNets $\varkappa$. We will show you how to construct $\varkappa$ in 5 steps.

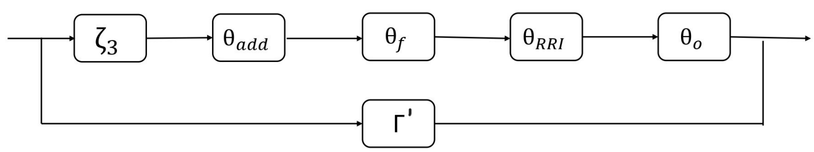

Figure 2: ResNets $\varkappa$'s structure

Step 1 (Constructing ζ_3): ζ_3 is made up of FNN ζ_2 and FNN θ_I. ζ_2 is composed with FNN θ'_{Caputo}, FNN θ_η and FNN θ_ϕ. Firstly we find how to construct θ_η and θ_ϕ.
$X_r(v) + Y_r(v) = \dfrac{y_r(V)}{\Gamma(1-\delta)(V-v)^{\delta}} + \dfrac{(-1)^{1-\delta}\rho^{\delta}}{\Gamma(1-\delta)}\displaystyle\int_0^V (v^{\rho} - \tau^{\rho})^{-\delta} y'_r(\tau)d\tau.$
Let consider $\rho = 1$

$$\begin{aligned}X_r(v) + Y_r(v) &= \frac{y_r(V)}{\Gamma(1-\delta)(V-v)^{\delta}} + \frac{(-1)^{1-\delta}}{\Gamma(1-\delta)}\int_0^V (v-\tau)^{-\delta} y'_r(\tau)d\tau\\ &= \frac{y_r(V)}{\Gamma(1-\delta)(V-v)^{\delta}} + \frac{(-1)^{1-\delta}}{\Gamma(1-\delta)}(v-V)^{-\delta} y_r(V) - \frac{(-1)^{1-\delta}}{\Gamma(1-\delta)}(t)^{-\delta} y_r(0)\\ &\quad - \frac{(-1)^{1-\delta}}{\Gamma(1-\delta)}\delta\int_0^V (v-\tau)^{-\delta-1} y_r(\tau)d\tau\\ &= \frac{(-1)^{-\delta}}{\Gamma(1-\delta)}(v)^{-\delta} y_r(0) + \frac{(-1)^{-\delta}}{\Gamma(1-\delta)}\delta\int_0^V (v-\tau)^{-\delta-1} y_r(\tau)d\tau\\ &= \phi_r(v) + \eta_r(v),\end{aligned}$$

where $\phi_r(v) = \dfrac{(-1)^{-\delta}}{\Gamma(1-\delta)}(v)^{-\delta} y_r(0)$ and $\eta_r(v) = \dfrac{(-1)^{-\delta}}{\Gamma(1-\delta)}\delta \displaystyle\int_0^V (v-\tau)^{-\delta-1} y_r(\tau) d\tau$.

There is a FNN $\theta_\phi = (Z_\phi, \mathscr{B}_\phi)$ with $(\mathcal{R}_a\theta_\phi)(y_r) = \phi_r$ that convert the input y_r to the output $(\phi_r(v_0), \dots, \phi_r(v_{K+1}))^{T'}$, where

$$Z_\phi = \frac{(-1)^{-\delta}}{\Gamma(1-\delta)} \begin{pmatrix} v_0^{-\delta} & 0 & \dots & 0 \\ v_1^{-\delta} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ v_{K+1}^{-\delta} & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{(K+2)\times(K+2)}, \mathscr{B}_\phi = (0, 0, \dots, 0)^{T'} \in \mathbb{R}^{(K+2)}.$$

Additionally, FNN exists $\theta_\eta = (Z_\eta, \mathscr{B}_\eta)$ with $(\mathcal{R}_a\theta_\eta)(y_r) = \eta_r$ that convert the input y_r to the output $(\eta_r(v_0), \dots, \eta_r(v_{K+1}))^{T'}$, where

$$Z_\eta = \frac{(-1)^{-\delta}}{\Gamma(1-\delta)}\delta\Delta v \begin{pmatrix} 0 & (-v_1)^{-\delta-1} & (-v_2)^{-\delta-1} & \dots & (-v_K)^{-\delta-1} & 0 \\ v_1^{-\delta-1} & 0 & (v_1-v_2)^{-\delta-1} & \dots & (v_1-v_K)^{-\delta-1} & 0 \\ v_2^{-\delta-1} & (v_2-v_1)^{-\delta-1} & 0 & \dots & (v_2-v_K)^{-\delta-1} & 0 \\ v_3^{-\delta-1} & (v_3-v_1)^{-\delta-1} & (v_3-v_2)^{-\delta-1} & \dots & (v_3-v_K)^{-\delta-1} & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ v_K^{-\delta-1} & (v_K-v_1)^{-\delta-1} & (v_K-v_2)^{-\delta-1} & \dots & 0 & 0 \\ V^{-\delta-1} & (V-v_1)^{-\delta-1} & (V-v_2)^{-\delta-1} & \dots & (V-v_K)^{-\delta-1} & 0 \end{pmatrix}$$
$$\in \mathbb{R}^{(K+2)\times(K+2)},$$

and $\mathscr{B}_\eta = (0, 0, \dots, 0)^{T'} \in \mathbb{R}^{(K+2)}$.
In keeping with preposition 3.2, a FNN $\theta'_{Caputo} = ((-1)^{-\delta} Z_{Caputo}, \mathscr{B}_{Caputo})$ exists that convert the input y_r
to the output

$$\left((-1)^{-\delta}\,{}_0^c\mathcal{D}_t^{\delta,\rho} y_r(t_0), \dots, (-1)^{-\delta}\,{}_0^c\mathcal{D}_t^{\delta,\rho} y_r(t_{K+1})\right)^{T'}.$$

As $\mathcal{T}(\theta'_{Caputo}) = \mathcal{T}(\theta_\phi) = \mathcal{T}(\theta_\eta) = (K+2, K+2)$, therefore, based on Lemma 3.4, there is a FNN $\zeta_2 = (Z_2, \mathscr{B}_2)$, where

$$Z_2 = \begin{pmatrix} \mathscr{I} & \mathscr{I} & \mathscr{I} \end{pmatrix} \begin{pmatrix} (-1)^{-\delta} Z_{Caputo} & 0 & 0 \\ 0 & Z_\phi & 0 \\ 0 & 0 & Z_\eta \end{pmatrix} \begin{pmatrix} \mathscr{I} \\ \mathscr{I} \\ \mathscr{I} \end{pmatrix} \in \mathbb{R}^{(K+2)\times(K+2)},$$
$$\mathscr{B}_2 = \begin{pmatrix} \mathscr{I} & \mathscr{I} & \mathscr{I} \end{pmatrix} \begin{pmatrix} \mathscr{B}_{Caputo} \\ \mathscr{B}_\phi \\ \mathscr{B}_\eta \end{pmatrix} \in \mathbb{R}^{(K+2)}.$$

$(\mathcal{R}_a\zeta_2)(w) = \left(\mathcal{R}_a\theta'_{Caputo}\right)(w) + (\mathcal{R}_a\theta_\phi)(w) + (\mathcal{R}_a\theta_\eta)(w), \forall w \in \mathbb{R}^{(K+2)}$. When the input of ζ_2 is y_r, then the output is

$$\left((-1)^{-\delta}\,{}_0^c\mathcal{D}_v^{\delta,\rho} y_r(v_0) + \phi_r(v_0) + \eta_r(v_0), \dots, (-1)^{-\delta}\,{}_0^c\mathcal{D}_v^{\delta,\rho} y_r(v_{K+1}) + \phi_r(v_{K+1}) + \eta_r(v_{K+1})\right)^{T'},$$
$$\text{i.e., } \left(-{}_v\mathcal{I}_T^{-\delta} y_r(v_0), \dots, -{}_v\mathcal{I}_T^{-\delta} y_r(v_{K+1})\right)^{T'}.$$

From Lemma 3.2, $\mathcal{L}_\mathfrak{a}(\zeta_2) = \mathcal{L}_\mathfrak{a}(\theta_1) = 1$, then there exists a FNN $\zeta_3 = \mathbf{P}_2(\theta_1, \zeta_2)$, $\mathcal{T}(\zeta_3) = (2(K+2), 2(K+2))$. Let $\zeta_3 = ((Z_3, \mathscr{B}_3))$, then

$$Z_3 = \begin{pmatrix} Z_\mathscr{I} & 0 \\ 0 & Z_2 \end{pmatrix}, B_3 = \begin{pmatrix} \mathscr{B}_\mathscr{I} \\ \mathscr{B}_2 \end{pmatrix}.$$

Thus when the input of ζ_3 is $(wy)_r$, then the output is

$$\left(w_r(v_0), \ldots, w_r(v_{K+1}), -{}_v\mathcal{I}_V^{-\delta} y_r(v_0), \ldots, -{}_v\mathcal{I}_V^{-\delta} y_r(v_{K+1})\right)^{T'}.$$

Step 2 (Constructing θ_{add}): Set FNN $\theta_{add} = (Z_{add}, \mathscr{B}_{add})$, where $Z_{add} = (\mathscr{I}, \mathscr{I}) \in \mathbb{R}^{(K+2)\times 2(K+2)}$ and $\mathscr{B}_{add} = (0, ..., 0)^{T'} \in \mathbb{R}^{(K+2)}$ then $(\mathcal{R}_a\theta_{add})(w \;\; y) = w + y$, $\forall w, y \in \mathbb{R}^{(K+2)}$. When the input of θ_{add} is

$$\left(w_r(v_0), \ldots, w_r(v_{K+1}), -{}_v\mathcal{I}_V^{\delta} y_r(v_0), \ldots, -{}_v\mathcal{I}_V^{\delta} y_r(v_{K+1})\right)^{T'},$$

then the output is

$$\left(w_r(v_0) - {}_v\mathcal{I}_V^{\delta} y_r(v_0), \ldots, w_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta} y_r(v_{K+1})\right)^{T'}.$$

Step 3 (Constructing θ_f): Let us consider FNN $\theta_f = (Z_f, B_f)$, where $Z_f = \mathscr{I} \in \mathbb{R}^{(K+2)\times 2(K+2)}$ and $\mathscr{B}_f = (f(v_0), f(v_1) \ldots, f(v_{K+1}))^{T'} \in \mathbb{R}^{(K+2)}$, then $(\mathcal{R}_a\theta_f)(w) = w + f$, $f = (f(v_0)f(v_1) \ldots, f(v_{K+1}))^{T'}$, $\forall w \in \mathbb{R}^{(K+2)}$. When the input of θ_f is

$$\left(w_r(v_0) - {}_v\mathcal{I}_V^{\delta} y_r(v_0), \ldots, w_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta} y_r(v_{K+1})\right)^{T'},$$

then the output is

$$\left(w_r(v_0) - {}_v\mathcal{I}_V^{\delta} y_r(v_0) + f(v_0), \ldots, w_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta} y_r(v_{K+1}) + f(v_{K+1})\right)^{T'}.$$

Step 4 (Constructing θ_{RRI}):

Proposition 3.4 *There exists a FNN θ_{RRI} that $(\mathcal{R}_a\theta_{RRI})(h) =_v \mathcal{I}_V^{\delta} h$, $\forall h = (h(v_0), \ldots, h(v_{K+1}))^{T'}$.*

Proof: The right RLFI can be discretized as

$$\begin{aligned} {}_v\mathcal{I}_V^{\delta} h(v_j) &= \frac{1}{\Gamma(\delta)} \int_{v_j}^{V} (\tau - v_j)^{\delta-1} h(\tau)\, d\tau \\ &\approx \frac{\Delta v}{\Gamma(\delta)} [(v_{j+1} - v_j)^{\delta-1} h(v_{j+1}) + (v_{j+2} - v_j)^{\delta-1} h(v_{j+2}) + \cdots + (V - v_j)^{\delta-1} h(V)]. \end{aligned}$$

Then, a FNN is present $\theta_{RRI} = (Z_{RRI}, \mathscr{B}_{RRI})$ with the input $(h(v_0), \ldots, h(v_{K+1}))^{T'}$, and the output $({}_v\mathcal{I}_V^{\delta} h(v_0), \ldots, {}_v\mathcal{I}_V^{\delta} h(v_{K+1}))^{T'}$, where

$$\begin{aligned} Z_{RRI} = \frac{\Delta v}{\Gamma(\delta)} &\begin{pmatrix} 0 & (v_1 - v_0)^{\delta-1} & (v_2 - v_0)^{\delta-1} & (v_3 - v_0)^{\delta-1} & \ldots & (v_K - v_0)^{\delta-1} & (V - v_0)^{\delta-1} \\ 0 & 0 & (v_2 - v_1)^{\delta-1} & (v_3 - v_1)^{\delta-1} & \ldots & (v_K - v_1)^{\delta-1} & (V - v_1)^{\delta-1} \\ 0 & 0 & 0 & (v_3 - v_2)^{\delta-1} & \ldots & (v_K - v_2)^{\delta-1} & (V - v_2)^{\delta-1} \\ 0 & 0 & 0 & 0 & \ldots & (v_K - v_3)^{\delta-1} & (V - v_3)^{\delta-1} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & (V - v_K)^{\delta-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix} \\ &\in \mathbb{R}^{(K+2)\times(K+2)}, \end{aligned}$$

and $\mathscr{B}_{RRI} = (0, \ldots, 0)^{T'} \in \mathbb{R}^{(K+2)}$ such that

$$\begin{pmatrix} {}_v\mathcal{I}_V^{\delta} h(v_0) \\ \vdots \\ {}_v\mathcal{I}_V^{\delta} h(v_{K+1}) \end{pmatrix} = Z_{RRI} \begin{pmatrix} h(v_0) \\ \vdots \\ h(v_{K+1}) \end{pmatrix} + \mathscr{B}_{RRI}.$$

Hence proved. □

When the input of θ_{RRI} is

$$\left(w_r(v_0) - {}_v\mathcal{I}_V^{\delta} y_r(v_0) + f(v_0), \ldots, w_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta} y_r(v_{K+1}) + f(v_{K+1})\right)^{T'},$$

then the output is

$$\left({}_v\mathcal{I}_V^{\delta}\left[w_r(v_0) - {}_v\mathcal{I}_V^{\delta} y_r(v_0) + f(v_0)\right], \ldots, {}_v\mathcal{I}_V^{\delta}\left[w_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta} y_r(v_{K+1}) + f(v_{K+1})\right]\right)^{T'}.$$

Step 5 (Constructing $\varkappa$): ResNets $\varkappa$ can be denoted as $\varkappa = (\Gamma, \Psi)$. As we can see from Lemma 3.1, a FNN exists $\Psi = \theta_o \bullet \theta_{RRI} \bullet \theta_f \bullet \theta_{add} \bullet \zeta_3$ with the input of Ψ is $(wy)_r$, then the output is

$$\left(-{}_v\mathcal{I}_V^{\delta}\left[w_r(v_0) - {}_v\mathcal{I}_V^{\delta} y_r(v_0) + f(v_0)\right], \ldots, -{}_v\mathcal{I}_V^{\delta}\left[w_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta} y_r(v_{K+1}) + f(v_{K+1})\right]\right)^{T'},$$

where $\Psi = (W_\Psi, B_\Psi)$, $W_\Psi = W_o W_{RRI} W_f W_{add} W_3 \in \mathbb{R}^{(K+2)\times 2(K+2)}$, $B_\Psi = W_o\{W_{RRI}[W_f(W_{add}B_3 + B_{add}) + B_f] + B_{RRI}\} + B_o \in \mathbb{R}^{(K+2)}$ and $\mathcal{T}(\Psi) = (2(K+2), K+2)$. The notation for the ResNets $\varkappa$ is $\varkappa = (\Gamma, \Psi)$ then when the input of $\varkappa$ is $(wy)_r$, then the output is

$$\left(y_r(v_0) - {}_v\mathcal{I}_V^{\delta}\left[w_r(v_0) - {}_v\mathcal{I}_V^{\delta} y_r(v_0) + f(v_0)\right], \ldots, y_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta}\left[w_r(v_{K+1}) - {}_v\mathcal{I}_V^{\delta} y_r(v_{K+1}) + f(v_{K+1})\right]\right)^{T'}.$$

(iii) Now, let us explain how to create $\mho$ using ϖ and $\varkappa$. When the input is $(wy)_r$, the output of ϖ and $\varkappa$ are $(w_r(v_{K+1}, \ldots, w_r(v_{K+1})^{T'}$ and $(y_r(v_{K+1}, \ldots, w_r(v_{K+1})^{T'}$, respectively. As $\mathcal{T}(\Omega) = \mathcal{T}(\Psi) = (2(K+2), K+2)$, in line with Proposition 3.1, a ResNets Λ made out of ϖ and $\varkappa$ exists, and when the input be $(wy)_r$, then the output be

$$\left(w_{r+1}(v_0), \ldots, w_{r+1}(v_{K+1}), y_{r+1}(v_0), \ldots, y_{r+1}(v_{K+1})\right)^{T'},$$

and $\Lambda = \left(\begin{pmatrix}\Gamma & 0\\ 0 & \Gamma\end{pmatrix} A,\ \mathbf{P}_2(\Omega, \Psi) \otimes A\right)$, $A = \begin{pmatrix}\mathscr{I}_{2(K+2)}\\ \mathscr{I}_{2(K+2)}\end{pmatrix}$.
As $\mho = \Lambda \bullet \cdots \bullet \Lambda$, N numbers of Λ combined into one composition, we express $\mho$. According to Lemma 3.5, when $\mho$'s input is

$$\left(w_0(v_0), \ldots, w_0(v_{K+1}), y_0(v_0), \ldots, y_0(v_{K+1})\right)^{T'},$$

then the output is

$$\left(w_N(v_0), \ldots, w_N(v_{K+1}), y_N(v_0), \ldots, y_N(v_{K+1})\right)^{T'}.$$

We thus demonstrate that a ResNets may be able to provide the approximate answers $w_N(t)$ and $y_N(t)$. When used with Eq.(2.15), $\mho$'s complexity

$$\mathscr{P}(\mho) = N\mathscr{P}(\Lambda) \leq N \cdot 4\mathscr{P}(\varphi) = 4N(K+2)[4(K+2)+1].$$

This suggests that the parameter count of the estimated ResNets has a superior constraint and expand polynomially in K and N. Theorem 3.1 has a thorough proof. □

To gain a better understanding the previously mentioned notions, let's solve a few problems.

4. Examples

Example 4.1 Approach the function $f(x) = x^2 + x$ using the idea $\left(\mathcal{R}_a \theta_{Caputo}\right)(w_n) = {}_0^c\mathcal{D}_v^{\delta,\rho} w_n$.

Solution : To solve this, we use the Proposition 3.2, i.e.,

$$ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_{j+1}) \approx \frac{\rho^{\delta-1}h^{-\delta\rho}}{\Gamma(2-\delta)} \sum_{j=0}^{n-1} w_j \times \left[f(v_{j+1}) - f(v_j)\right], $$

where $w_j = \frac{-(n^\rho-(j+1)^\rho)^{1-\delta}}{(j+1)^{\rho-1}} + \frac{(n^\rho-j^\rho)^{1-\delta}}{j^{\rho-1}}$.
Let us assume for $\delta = 0, n = 1, \rho = 1, h = 1$ and $K = 1$, then we have $w_0 = 1$ and $w_1 = 1$. Then,

$$ \begin{aligned} Z_{Caputo} &= \frac{1}{\Gamma(2)} \cdot \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}. \end{aligned} $$

Now,

$$ \begin{aligned} \begin{pmatrix} {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_0) \\ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_1) \\ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_2) \end{pmatrix} &= Z_{Caputo} \begin{pmatrix} f(v_0) \\ f(v_1) \\ f(v_2) \end{pmatrix} + \mathscr{B}_{Caputo}, \qquad \textit{where } \mathscr{B}_{Caputo} = (0,\ldots,0)^{T'} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ -v_0 + v_1 \\ -v_0 + v_2 \end{pmatrix}. \end{aligned} $$

By using the Definitions 2.3, we get

$$ \begin{aligned} {}_0^c\mathcal{D}_{v_i}^{\delta,\rho} f(v_i) &= \frac{1}{\Gamma(1)} \int_0^{v_i} (v_i - x)^{1-0-1} \frac{d}{dx} f(x)dx \\ &= \int_0^{v_i} f'(x)dx = \int_0^{v_i} (2x+1)dx \\ &= (x^2 + x)_0^{v_i} = v_i(v_i + 1). \end{aligned} $$

When $i = 0$,

$$ \begin{aligned} &{}_0^c\mathcal{D}_{v_0}^{\delta,\rho} f(v_0) = 0 \\ &v_0(v_0 + 1) = 0 \Rightarrow v_0 = 0, -1. \end{aligned} $$

When $i = 1$,

$$ \begin{aligned} &{}_0^c\mathcal{D}_{v_1}^{\delta,\rho} f(v_1) = -v_0 + v_1 \\ &v_1(v_1 + 1) = -v_0 + v_1 \Rightarrow v_1^2 = -v_0. \end{aligned} $$

If $v_0 = 0$ then $v_1 = 0$ and if $v_0 = -1$ then $v_1 = 1, -1$, thus we get $v_1 = -1, 0, 1$.

$$ \begin{aligned} &{}_0^c\mathcal{D}_{v_2}^{\delta,\rho} f(v_2) = -v_0 + v_2 \\ &v_2(v_2 + 1) = -v_0 + v_2 \Rightarrow v_2^2 = -v_0. \end{aligned} $$

If $v_0 = 0$ then $v_2 = 0$ and if $v_0 = -1$ then $v_2 = 1, -1$. We get $v_2 = -1, 0, 1$.

Example 4.2 This problem displays the outcome for a matrix when $f(z) = z^2 + 3z + 2$ is used in $(\mathcal{R}_a \theta_{Caputo})(w_n)$
$= {}_0^c \mathcal{D}_v^{\delta,\rho} w_n$.

Solution: To solve this, we use the Proposition 3.2, i.e.,

$$ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_{j+1}) \approx \frac{\rho^{\delta-1}h^{-\delta\rho}}{\Gamma(2-\delta)} \sum_{j=0}^{n-1} w_j \times [f(v_{j+1}) - f(v_j)], $$

where $w_j = \frac{-(n^\rho-(j+1)^\rho)^{1-\delta}}{(j+1)^{\rho-1}} + \frac{(n^\rho-j^\rho)^{1-\delta}}{j^{\rho-1}}$.
Let us assume for $\delta = 1/2, n = 3, \rho = 1, h = 1$ and $K = 2$, we get

$$ w_j = -(3-(j+1))^{1/2} + (3-j)^{1/2}, $$

when $j = 0, 1, 2$, then we have $w_0 = \sqrt{3}-\sqrt{2}, w_1 = \sqrt{2}-1, w_2 = 1$. Now

$$ \begin{aligned} Z_{Caputo} &= \frac{\rho^\delta h^{\rho(1-\delta)-1}}{\Gamma(2-\delta)} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & w_0 & 0 & 0 \\ 0 & w_1 & w_0 & 0 \\ 0 & w_2 & w_1 & w_0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \\ &= \frac{1}{\Gamma(3/2)} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \sqrt{3}-\sqrt{2} & 0 & 0 \\ 0 & \sqrt{2}-1 & \sqrt{3}-\sqrt{2} & 0 \\ 0 & 1 & \sqrt{2}-1 & \sqrt{3}-\sqrt{2} \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \\ &= \frac{2}{\sqrt{\pi}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ \sqrt{2}-\sqrt{3} & \sqrt{3}-\sqrt{2} & 0 & 0 \\ 1-\sqrt{2} & 2\sqrt{2}-\sqrt{3}-1 & \sqrt{3}-\sqrt{2} & 0 \\ -1 & 2-\sqrt{2} & 2\sqrt{2}-\sqrt{3}-1 & \sqrt{3}-\sqrt{2} \end{pmatrix}. \end{aligned} $$

$$ \begin{aligned} \begin{pmatrix} {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_0) \\ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_1) \\ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_2) \\ {}_0^c\mathcal{D}_v^{\delta,\rho} f(v_3) \end{pmatrix} &= Z_{Caputo} \begin{pmatrix} f(v_0) \\ f(v_1) \\ f(v_2) \\ f(v_3) \end{pmatrix} + \mathscr{B}_{Caputo}, \qquad where \ \ \mathscr{B}_{Caputo} = (0,0,0,0)^{T'} \\ &= \frac{2}{\sqrt{\pi}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ \sqrt{2}-\sqrt{3} & \sqrt{3}-\sqrt{2} & 0 & 0 \\ 1-\sqrt{2} & 2\sqrt{2}-\sqrt{3}-1 & \sqrt{3}-\sqrt{2} & 0 \\ -1 & 2-\sqrt{2} & 2\sqrt{2}-\sqrt{3}-1 & \sqrt{3}-\sqrt{2} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ &= \frac{2}{\sqrt{\pi}} \begin{pmatrix} 0 \\ -(\sqrt{3}-\sqrt{2})v_0 + (\sqrt{3}-\sqrt{2})v_1 \\ (1-\sqrt{2})v_0 + (2\sqrt{2}-\sqrt{3}-1)v_1 + (\sqrt{3}-\sqrt{2})v_2 \\ -v_0 + (2-\sqrt{2})v_1 + (2\sqrt{2}-\sqrt{3}-1)v_2 + (\sqrt{3}-\sqrt{2})v_3 \end{pmatrix}. \end{aligned} $$

By using the Definitions 2.3, we get

$$ \begin{aligned} {}_0^c\mathcal{D}_{v_i}^{\delta,\rho} f(v_i) &= \frac{1}{\Gamma(5/2)} \int_0^{v_i} (v_i - z)^{3-\frac{1}{2}-1} \left(\frac{d}{dz}\right)^3 f(z)dz \\ &= \frac{4}{3\sqrt{\pi}} \int_0^{v_i} (v_i - z)^{3/2} f'''(z)dz \\ &= 0. \end{aligned} $$

When $i = 0$,

$$ {}_0^c\mathcal{D}_{v_0}^{\delta,\rho} f(v_0) = 0 \Rightarrow 0 = 0. $$

When $i = 1$,

$$
\begin{aligned}
{}_0^c\mathcal{D}_{v_1}^{\delta,\rho} f(v_1) &= \frac{2}{\sqrt{\pi}}\left[-(\sqrt{3}-\sqrt{2})v_0 + (\sqrt{3}-\sqrt{2})v_1\right] \\
0 &= \frac{2}{\sqrt{\pi}}\left[-(\sqrt{3}-\sqrt{2})v_0 + (\sqrt{3}-\sqrt{2})v_1\right] \\
v_1 &= v_0.
\end{aligned}
$$

When $i = 2$,

$$
\begin{aligned}
{}_0^c\mathcal{D}_{v_2}^{\delta,\rho} f(v_2) &= \frac{2}{\sqrt{\pi}}(1-\sqrt{2})v_0 + (2\sqrt{2}-\sqrt{3}-1)v_1 + (\sqrt{3}-\sqrt{2})v_2 \\
0 &= \frac{2}{\sqrt{\pi}}(1-\sqrt{2})v_0 + (2\sqrt{2}-\sqrt{3}-1)v_1 + (\sqrt{3}-\sqrt{2})v_2 \\
v_2 &= v_0.
\end{aligned}
$$

Similarly when $i = 3$, we get $v_3 = v_0$.

5. Conclusions

In this paper, we have demonstrated that *VIM* handles residual neural network approximations to fractional differential equation solutions. The *VIM* has been applied to FDEs containing RLFI and GCTFD and it mainly focused on left RLFI and left GCTFD. The technique first defines VIF for the solution and then uses a residual neural network to establish the iteration format's convergence. This study demonstrates that it is possible to approximate a solution using a neural network by applying VI approach. Lastly, some examples are provided to help you grasp the improved understanding.

References

1. Agarwal, S., Mishra, L. N., *Attributes of residual neural networks for modeling fractional differential equations*, Heliyon. 10, e38332, (2024).
2. Ansari, M., Mishra, L. N., *Common solution to a coupled system of fractional differential equations and nonlinear integral equations via weakly altering distance functions and w-distance*, Adv. Stud.: Euro-Tbil. Math. J. 18(2), 149-174, (2025).
3. Arikoglu, A., Ozkol, I., *Solution of fractional differential equations by using differential transform method*, Chaos Solit. Fractals. 34, 1473-1481, (2007).
4. Baggenstos, J., Salimova, D., *Approximation properties of residual neural networks for Kolmogorov PDEs*, Discrete Contin. Dyn. Syst. - Ser. B. 28, 3193–215, (2023).
5. Bhadana, K. G., Meena, A. K., Mishra, V. N., *Some unified integral formulae involving with general class of polynomials and generalized Hurwitz-Lerch zeta function*, AIP Conf. Proc. 2768, 020018, (2023).
6. Bhat, I. A., Mishra, L. N., Mishra, V. N., *Comparative analysis of nonlinear Urysohn functional integral equations via Nyström method*, Appl. Math. Comput. 494, 129287, (2025).
7. Bhat, I. A., Mishra, L. N., Mishra, V. N., Abdel-Aty, M., Oasymeh, M., *A comprehensive analysis for weakly singular nonlinear functional Volterra integral equations using discretization techniques*, Alex. Eng. J. 104, 564-575, (2024).
8. Bhat, I. A., Mishra, L. N., Mishra, V. N., Tunç, C., Tunç, O., *Precision and efficiency of an interpolation approach to weakly singular integral equations*, Int. J. Numer. Methods Heat Fluid Flow. 34, 1479-1499, (2024).
9. Chu, Y. M., Bani Hani, E. H., El-Zahar, E. R., Ebaid, A., Shah, N. A., *Combination of Shehu decomposition and variational iteration transform methods for solving fractional third order dispersive partial differential equations*, Numer. Methods Partial Differ. Equations. 40(2), e22755, (2024).
10. Farid, G., Bibi, S., Rathour, L., Mishra, L. N., Mishra, V. N., *Fractional versions of Hadamard inequalities for strongly (s, m)-convex functions via Caputo fractional derivatives*, Korean J. Math. 31(1), 75-94, (2023).
11. Ghorbani, A., Momani, S., *An effective variational iteration algorithm for solving Riccati differential equations*, Appl. Math. Lett. 23, 922-927, (2010).
12. He, J. H., *Approximate analytical solution for seepage flow with fractional derivatives in porous media*, Comput. Methods Appl. Mech. Eng. 167, 57-68, (1998).
13. He, J. H., *Variational iteration method-some recent results and new interpretations*, J. Comput. Appl. Math. 207, 3-17, (2007).

14. Higazy, M., Alsallami, S. A. M., Abdel-Khalek, S., El-Mesady, A., *Dynamical and structural study of a generalized Caputo fractional order Lotka-Volterra model*, Results Phys. 37,105478, (2022).

15. Hogeme, M. S., Woldaregay, M. M., Rathour, L., Mishra, V. N., *A stable numerical method for singularly perturbed Fredholm integro differential equation using exponentially fitted difference method*, J. Comput. Appl. Math. 441, 115709, (2024).

16. Inc, M., Parto-Haghighi, M., Akinlar, M. A., Chu, Y., *New numerical solutions of fractional-order Korteweg-de Vries equation*, Results Phys. 19, 103326, (2022).

17. Jamshed, W., Nisar, K. S., *Computational single-phase comparative study of a Williamson nanofluid in a parabolic trough solar collector via the Keller box method*, Int. J. Energy Res. 45, 10696-10718, (2021).

18. Jentzen, A., Salimova, D., Welti, T., *A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients*, Commun. Math. Sci. 19, 1167–205, (2021).

19. Kumar, S., Nieto, J. J., Ahmad, B., *Chebyshev spectral method for solving fuzzy fractional Fredholm–Volterra integro-differential equation*, Math. Comput. Simul. 192, 501-513, (2022).

20. Liu, F., Anh, V., Turner, I., *Numerical solution of the space fractional Fokker–Planck equation*, J. Comput. Appl. Math. 166, 209-219, (2004).

21. Liu, Z., Yang, Y., Cai, Q., *Neural network as a function approximator and its application in solving differential equations*, Appl. Math. Mech. 40, 237-248, (2019).

22. Mao, Z., Chen, S., Shen, J., *Efficient and accurate spectral method using generalized Jacobi functions for solving Riesz fractional differential equations*, Appl. Numer. Math. 106, 165–181, (2016).

23. Negero, N. T., Duressa, G. F., Rathour, L., Mishra, V. N., *A novel fitted numerical scheme for singularly perturbed delay parabolic problems with two small parameters. Partial Differ*, Equations Appl. Math. 8, 100546, (2023).

24. Odibat, Z., Baleanu, D., *Numerical simulation of initial value problems with generalized Caputo-type fractional derivatives*, Appl. Numer. Math. 156, 94105, (2020).

25. Odibat, Z. M., *Rectangular decomposition method for fractional diffusion-wave equations*, Appl. Math. Comput. 179, 92-97, (2006).

26. Pathak, V. K., Mishra, L. N., Mishra, V. N., *On the solvability of a class of nonlinear functional integral equations involving Erdélyi–Kober fractional operator*, Math. Methods Appl. Sci. 46, 14340-14352, (2023).

27. Paul, S. K., Mishra, L. N., Mishra, V. N., *Results on integral inequalities for a generalized fractional integral operator unifying two existing fractional integral operators*, Nonlinear Anal.: Model. Control. 29(6), 1080-1105, (2024).

28. Paul, S. K., Mishra, L. N., Mishra, V. N., Baleanu, D., *An effective method for solving non linear integral equations involving the Riemann-Liouville fractional operator*, AIMS Math. 8(8), 17448–17469, (2023).

29. Sheng, H., Li, Y., Chen, Y. Q., *Application of numerical inverse Laplace transform algorithms in fractional calculus*, J. Franklin Inst. 348, 315-330, (2011).

30. Stefanski, T. P., Gulgowski, J., *Signal propagation in electromagnetic media described by fractional-order models*, Commun. Nonlinear Sci. Numer. Simul. 82, 105029, (2020).

31. Tavares, D., Almeida, R., Torres, D. F. M., *Caputo derivatives of fractional variable order: numerical approximations*, Commun. Nonlinear Sci. Numer. Simul. 35, 69-87, (2016).

32. Ullah, A., Ali, S., Awwad, F. A., Ismail, E. A., *Analysis of the convective heat transfer through straight fin by using the Riemann-Liouville type fractional derivative: Probed by machine learning*, Heliyon. 10(4), (2024).

33. Woldaregay, M. M., Hunde, T. W., Mishra, V. N., *Fitted exact difference method for solutions of a singularly perturbed time delay parabolic PDE*, Partial Differ. Equations Appl. Math. 8, 100556, (2023).

34. Wu, G., *A fractional variational iteration method for solving fractional nonlinear differential equations*, Comput. Math. Appl. 61, 2186-2190, (2011).

35. Yang, X., Gao, F., Ju, Y., *General Fractional Derivatives with Applications in Viscoelasticity*, Academic Press. (2020).

36. Wu, G., Lee, E. W. M., *Fractional variational iteration method and its application*, Phys. Lett. A. 374, 2506-2509, (2010).

37. Yadav, A. K., Pandey, R. M., Mishra, V. N., Agarwal, R., *Some Integral Inequalities Involving a Fractional Integral Operator with Extended Hypergeometric Function*, Dolomites Res. Notes Approx. 18(1), 17–26, (2025).

38. Zuo, J., Yang, J., *Approximation properties of residual neural networks for fractional differential equations*, Commun. Nonlinear Sci. Numer. Simul. 125, 107399, (2023).

Alice Antony,
Department of Mathematics,
Vellore Institute of Technology,
Vellore 632 014, Tamil Nadu, India.
E-mail address: aliceantony13@gmail.com

and

Lakshmi Narayan Mishra,
Department of Mathematics,
Vellore Institute of Technology,
Vellore 632 014, Tamil Nadu, India.
E-mail address: lakshminarayanmishra04@gmail.com, lakshminarayan.mishra@vit.ac.in