INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 20, Number 5, Pages 709–723

doi: 10.4208/ijnam2023-1031

FRACTIONAL ORDER LEARNING METHODS FOR NONLINEAR SYSTEM IDENTIFICATION BASED ON FUZZY NEURAL NETWORK

JIE DING, SEN XU, AND ZHIJIE LI

Abstract. This paper focuses on neural network-based learning methods for identifying nonlinear dynamic systems. The Takagi-Sugeno (T-S) fuzzy model is introduced to represent nonlinear systems in a linear way. Fractional calculus is integrated to minimize the cost function, yielding a fractional-order learning algorithm that can derive optimal parameters in the T-S fuzzy model. The proposed algorithm is evaluated by comparing it with an integer-order method for identifying numerical nonlinear systems and a water quality system. Both evaluations demonstrate that the proposed algorithm can effectively reduce errors and improve model accuracy.

Key words. Fractional calculus, T-S fuzzy neural network, gradient descent method, nonlinear systems.

1. Introduction

System identification (SID) is indispensable both in traditional industry (chemical industry, refinery, etc.) and modern engineering fields, e.g., mechanical engineering [1], robotics and quadrotors [2], etc., since it can provide effective models for adaptive control and prediction [3–5]. Neural network (NN) based SID methods have aroused the interest of many researchers since the activation function in NN has multiple choices so that the NN could bear excellent performance in fitting nonlinear systems [6]. One of the widely used NN is the fuzzy neural network (FNN) [7, 8], which is a hybrid intelligent system based on fuzzy logic and neural network. By applying fuzzy sets and rules to the neural network, the FNN can handle fuzzy and uncertain information and has good generalization ability. The FNN has been widely applied in many fields, including electric power [9], machinery [10], and mathematics [11].

Takagi-Sugeno (T-S) fuzzy model is one of the most commonly used fuzzy neural network models which is proposed by Takagi and Sugeno in 1985. It describes the rule of 'if-then' and consists of two parts: an antecedent network and a consequent network. The antecedent network is used to match the antecedent of fuzzy rules, and the consequent network is used to generate the consequent of fuzzy rules. The total output is the weighted sum of the consequences of each fuzzy rule, where the weighting coefficient is the applicability of each rule. The parameters of membership functions can be initialized by the training samples, while associated parameters are identified by the recursive least squares algorithm [13]. Muralisankar made use of the Lyapunov-Krasovskii functional and stochastic analysis approach and established new delay-dependent stability criteria in terms of linear matrix inequalities (LMIs) in T-S fuzzy stochastic neural networks [14]. Li proposed a recognition method and simplified scheme for T-S fuzzy neural networks. The basic idea is that the structure identification of fuzzy neural networks is guided by the output approximation error attenuation in each cluster, with input space clustering

Received by the editors on November 26, 2022 and, accepted on April 14, 2023.
2000 *Mathematics Subject Classification.* 65-XX.

and sub clustering as the main steps [15]. Li studied observer based distributed time-varying delay T-S fuzzy neural network dissipation rate control, obtained an observer-based controller for the T-S fuzzy delayed model [16].

In addition, there are many identification algorithms for neural networks, such as the gradient descent (GD) algorithm [17] and the least squares (LS) algorithm [18]. In order to solve the problem of inefficient traditional lower and upper bound estimation (LUBE) model training, Liu adopted a new training scheme based on the GD method, which improved the LUBE model and enhanced efficiency [19]. Zheng focused on the most commonly used Stochastic Gradient Descent (SGD) algorithm in a mild decentralized setting and proposed a robust algorithm to handle unstable networks. [20].

Recently, fractional order-based SID algorithms have attracted many researchers [21, 22, 24]. Fractional differential calculus also has been a famous notion in mathematics for many years. It is an extension of traditional calculus and difference theory, and can describe many nonlinear and nonlocal phenomena. Some classical fractional derivatives include the Grunwald Letnikov derivative, Riemann-Liouville derivative, and Caputo derivative. Lupupa proposed a fractional order identification algorithm for wireless communication which had smaller errors than the conventional method because of its long-term memory ability and the reduction of model parameters [21]. Gehring analyzed fractional models from the perspective of mathematical algebra, unknown parameters and fractional order were identified solely from input-output signals, and further elastic materials were selected to illustrate the effectiveness of the method [22]. Liang investigated the input-output finite-time stability of fractional-order positive switched systems [23]. Aguilar used fractional calculus to reduce the number of parameters of the proposed neural network model, which simplified the complexity of the model and reduced the time required for digital simulation [24]. Compared with the integer order, fractional calculus takes into account the influence of the variables at the previous time and constructs a long memory function, so that the fractional order system can make use of the past information, and has a better effect on control and identification [25–27].

Taking into account the advantages of T-S fuzzy neural networks and fractional calculus, this paper proposes a fuzzy neural network for complex nonlinear system identification. The T-S fuzzy neural network is trained to fit nonlinear system observations in a linear way, and the parameters of the membership functions in the T-S fuzzy neural network can be estimated using a fractional-order gradient descent method. The main contributions of this paper are as follows:

- T-S fuzzy model is combined with the neural network to analyze complex nonlinear problems in a linear way. A fractional-order gradient descent learning algorithm is proposed to deal with the T-S fuzzy neural network.
- Evaluation is performed by comparing with integer order method on the identification of numerical nonlinear system and a water quality system, both of which show that the proposed algorithm can effectively reduce the error of the results and improve the accuracy of the model.

The rest of this paper is organized as follows. In Section 2, a T-S fuzzy model is constructed. In Section 3, the fractional order gradient descent updating rule is proposed to optimize the weights in the T-S fuzzy neural network, and a detailed pseudo code is presented. In Section 4, examples of numerical nonlinear model and water quality system are provided to verify the proposed algorithm. Finally, the evaluation of this work is reviewed and open issues are discussed for future research in Section 5.

2. Problem Formulation

2.1. T-S Model of Fuzzy Systems. T-S fuzzy model is a model with strong adaptive ability, and can be modeled as follows [28]:

$$\begin{aligned} R^i:\ If \quad & x_1\ is\ A_1^i, x_2\ is\ A_2^i, \cdots, x_n\ is\ A_n^i \\ & then\ y^i = p_0^i + p_1^i x_1 + \cdots + p_n^i x_n, \end{aligned} \tag{1}$$

where R^i $(i = 1, 2, \cdots, l)$ is the ith rule of neural network, l is the maximum number of rules, $x_1, x_2, \cdots, x_n$ are the inputs of the network, n is the number of inputs, and A_j^i $(j = 1, 2, \cdots, n)$ are fuzzy sets of x_j in the ith rule, y^i is the output of the ith rule and is a linear combination of inputs $x_1, x_2, \cdots, x_n$, p_j^i is the parameter of ith rule. Generally, larger l will increase the complexity of the network, while smaller l may yield inaccurate network. Define $M(x_j)$ as the membership degree of the membership function of x_j in the fuzzy set A_j^i:

$$M(x_j) = \mu_{x_j}^i = e^{-\frac{(x_j - c_j^i)^2}{b_j^i}}, \tag{2}$$

where $\mu_{x_j}^i$ is the membership degree of x_j in the ith rule, which decides to what degree one rule integrates to the final output, and c_j^i and b_j^i are the mean and width of the corresponding membership function, respectively. In this paper, the membership function has the same distribution with Gaussian, which yields excellent performance in system identification [29]. Considering the fuzzy operator of each rule, the output y_f of FNN can be obtained by:

$$\begin{aligned} y_f &= \frac{\sum_{i=1}^{l} w^i y^i}{\sum_{i=1}^{l} w^i} \\ &= \frac{\sum_{i=1}^{l} w^i (p_0^i + p_1^i x_1 + \cdots + p_n^i x_n)}{\sum_{i=1}^{l} w^i}, \end{aligned} \tag{3}$$

where

$$w^i = \prod_{j=1}^{n} \mu_{x_j}^i \quad i = 1, 2, \cdots, l, \tag{4}$$

represents the fuzzy degree of the ith output in the total output. In this article, the T-S fuzzy neural network is a parallel-parallel identification mode, where both input and output variables are decomposed into multiple fuzzy subsets, each corresponding to a fuzzy rule. These fuzzy rules are independent of each other, and fuzzy inference is conducted in parallel, ultimately summarizing the output of the fuzzy neural network.

Based on the T-S model given earlier, we can design the block diagram of a fuzzy neural network shown in Figure 1. The network consists of two parts: the antecedent network and the consequent network. The antecedent network is designed to match the antecedent of fuzzy rules, while the consequent network generate the consequent of fuzzy rules.

(A) Antecedent network. The antecedent network consists of four layers. The first layer is the input layer, in which each node is directly connected to the individual component x_i of the input vector. The input layer serves to transmit the input value $x = \left[x_1\, x_2 \ldots x_n\right]^{\mathrm{T}}$ to the next layer. n is the number of nodes in this layer.

The second layer consists of nodes representing linguistic variable values, such as "NM" and "PS". Its purpose is to calculate the membership function μ_j^i of each input component belonging to the fuzzy set of each linguistic variable value, where

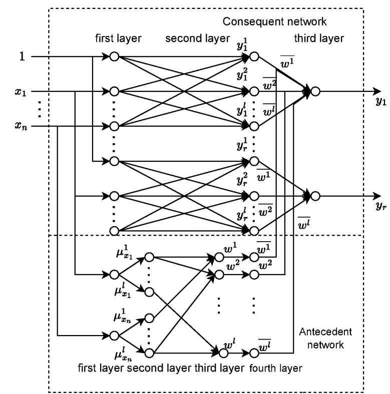


FIGURE 1. Fuzzy neural network structure based on T-S model.

$\mu_j^i = \mu_A^i(x_j)$, $j = 1, 2, \cdots, n; i = 1, 2, \cdots, l$. n is the dimension of the input, and l is the number of fuzzy partitions for x_j. In this paper, we use the Gaussian function (2) as the membership function.

Each node in the third layer represents a fuzzy rule, which is used to match the antecedents of the fuzzy rule and calculate the fitness of each rule, that is w^i, The total number of nodes in this layer is l. For a given input, only the language variable values near the input point have a large membership value, while the language variable values far from the input point have a small membership value.

The fourth layer has the same number of nodes as the third layer, which is used to achieve normalized calculation.

$$\overline{w^i} = w^j / \sum_{j=1}^{l} w^j, \quad i = 1, 2, \cdots, l. \tag{5}$$

(B) Consequent network. It consists of r parallel sub-networks with the same structure, each of which generates an output quantity.

The consequent network consists of r parallel sub-networks with the same structure. The first layer of each sub-network is the input layer, which transfers the input variables to the second layer. The input value of the 0th node in the input layer is $x_0 = 1$, which serves as the constant term in the fuzzy rule consequent.

The second layer of each sub-network has l nodes, each representing a rule. The function of this layer is to calculate the consequent of each rule.

$$\begin{gathered} y_j^i = p_{j0}^i + p_{j1}^i x_1 + \cdots + p_{jn}^i x_n = \sum_{k=0}^{n} p_{jk}^i x_k \\ j = 1, 2, \cdots, r; \quad i = 1, 2, \cdots, l; \quad x_0 = 1. \end{gathered} \tag{6}$$

The third layer of each sub-network calculates the output of the system.

$$y_j = \sum_{i=1}^{l} \overline{w^i} y_j^i \quad j = 1, 2, \cdots, r. \tag{7}$$

The output y_j of the fuzzy neural network is the weighted sum of the consequents of each rule, where the weighting coefficients are the normalized fitness of each fuzzy rule. In other words, the output of the antecedent network is used as the

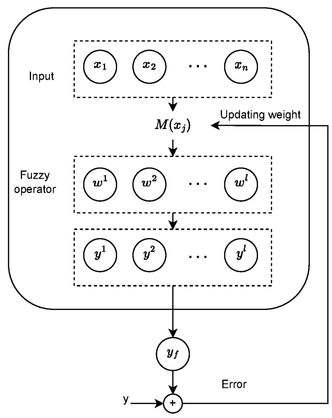


FIGURE 2. Flowchart of the fuzzy neural network.

connection weight of the consequent network. Figure 2 is the flowchart of the fuzzy neural network proposed in this paper.

To get a proper neural network is to find the optimal weights (p_j^i, c_j^i, b_j^i) in (1) and (2) that can minimize the following cost function

$$E(\theta) = \frac{1}{2}(y - y_f(\theta))^2, \tag{8}$$

where E is the cost function, y and y_f are real output and neural network output respectively, and θ represents the set of weights (p_j^i, c_j^i, b_j^i). Since fractional calculus has been a top choice to improve system performance both in control and identification, we take fractional order based algorithm into consideration in the next section.

2.2. Fractional calculus. The Riemann-Liouville (RL) fractional derivative, Grunwald-Letnikov (GL) fractional derivative, and Caputo fractional derivative are widely used definitions in fractional calculus.

(A) The RL fractional derivative. The RL fractional derivative is defined by Riemann-Liouville [21] as

$$D^\alpha f(t) = \begin{cases} \frac{\mathrm{d}^n}{\mathrm{d}t^n}\left[\frac{1}{\Gamma(n-\alpha)}\frac{\mathrm{d}}{\mathrm{d}t}\int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha+1-n}}\mathrm{d}\tau\right] & \\ & n-1 < \alpha < n \\ \frac{\mathrm{d}^n}{\mathrm{d}t^n} f(t) & \alpha = n \end{cases}, \tag{9}$$

where $f(\bullet)$ is the function of t, D is the fractional derivative operator, n is a positive integer $(n = 1, 2, \cdots)$, α is arbitrary order, and $\Gamma(\bullet)$ is the Gamma function defined as

$$\Gamma(\alpha) = \int_0^\infty e^{-\tau}\tau^{\alpha-1}\mathrm{d}\tau. \tag{10}$$

The RL fractional derivative is one of the earliest methods for defining fractional derivatives, which is defined by the Riemann-Liouville integral of a function and is mainly used to describe the initial conditions of fractional differential equations in practical applications. Its definition includes integrals, so the calculation is relatively complex. It is applicable to functions defined on the interval $(-\infty, \infty)$.

(B) The GL fractional derivative. The GL fractional derivative is proposed by Grunwald-Letnikov

$$D^{\alpha} f(t) = \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{m=0}^{\infty} (-1)^m \binom{\alpha}{m} f(t-mh) \approx \frac{1}{h^{\alpha}} \sum_{m=0}^{L} (-1)^m \binom{\alpha}{m} f(t-mh), \tag{11}$$

where h is the sampling time, L is the number of the past values considered, and the Newton's binomial $\binom{\alpha}{m}$ is defined as

$$\binom{\alpha}{m} = \frac{\Gamma(\alpha+1)}{\Gamma(m+1)\Gamma(\alpha-m+1)}. \tag{12}$$

Define $a_j^{\alpha} = (-1)^m \binom{\alpha}{m}$ $(j = 0, 1, 2, \cdots)$ as the forgetting factor, which represents the influence of the past moment on the future moment, then a_j^{α} can be written as:

$$a_j^{\alpha} = \begin{cases} 0 & j < 0 \\ 1 & j = 0 \\ (-1)^j \frac{\alpha(\alpha-1)\cdots(\alpha-j+1)}{j!} & j = 1, 2, 3, \ldots \end{cases}. \tag{13}$$

The recursive expression of a_j^{α} may be reformed as

$$a_j^{\alpha} = \begin{cases} 1 & j = 0 \\ \left(1 - \frac{\alpha+1}{j}\right) a_{j-1}^{\alpha} & j = 1, 2, 3, \ldots \end{cases}. \tag{14}$$

Grunwald-Letnikov fractional derivative, on the other hand, is defined by a recursive forward difference formula and is suitable for calculating the fractional derivative of discrete data. The definition of the GL fractional derivative is relatively simple and easy to compute and it can handle functions defined on any domain.

(C) The Caputo fractional derivative. In both the RL and GL fractional derivatives described above, the differential equation may be difficult to solve if the initial conditions are non-zero, so we need the other fractional derivative Caputo.

$$\mathscr{D}_t^{\alpha} y(t) = \frac{1}{\Gamma(m-\alpha)} \int_{t_0}^{t} \frac{y^{(m)}(\tau)}{(t-\tau)^{1+\alpha-m}} \, \mathrm{d}\tau, \tag{15}$$

where $m = \lceil \alpha \rceil$ is integer. Caputo fractional derivative is defined by a mixed method based on integer-order derivatives and Riemann-Liouville fractional derivatives, and is widely used to describe the boundary conditions of fractional differential equations in practical applications.

Overall, these three definitions of fractional derivatives have their own advantages and disadvantages. Depending on the specific problem, appropriate definition can be chosen to solve fractional calculus equations.

3. Techniques for fuzzy neural network

3.1. Fuzzy neural network based integer order algorithm. The integer order gradient descent method is a common way of implementing the machine learning process, particularly in neural network models, where the core of the BP backpropagation method is the gradient descent to optimise the weight parameters for each layer.

$$\theta_t = \theta_{t-1} - \eta \frac{\partial E}{\partial \theta},$$

(16) $$\triangle\theta = \theta_t - \theta_{t-1} = -\eta \frac{\partial E}{\partial \theta},$$

where θ_t and θ_{t-1} represent the weights of the current time and the previous time, respectively, η is the learning rate, and $\triangle\theta$ is the change of weight. Then, the modified integer order estimates of the model parameters can be derived as:

(17) $$p_j^i(t) = p_j^i(t-1) - \eta \frac{\partial E}{\partial y_f} \frac{\partial y_f}{\partial p_j^i},$$

(18) $$c_j^i(t) = c_j^i(t-1) - \eta \frac{\partial E}{\partial y_f} \frac{\partial y_f}{\partial c_j^i},$$

(19) $$b_j^i(t) = b_j^i(t-1) - \eta \frac{\partial E}{\partial y_f} \frac{\partial y_f}{\partial b_j^i},$$

where

$$\begin{aligned}
\frac{\partial E}{\partial y_f} &= -(y - y_f),\\
\frac{\partial y_f}{\partial p_j^i} &= \frac{w^i x_j}{w^1 + w^2 + \cdots + w^l} = \frac{w^i x_j}{\sum_{i=1}^l w^i},\\
\frac{\partial y_f}{\partial c_j^i} &= \frac{\partial y_f}{\partial w^i} \frac{\partial w^i}{\partial c_j^i}\\
&= \frac{\partial}{\partial w^i} \left(\frac{w^1 y^1 + w^2 y^2 + \cdots + w^l y^l}{\sum_{i=1}^l w^i}\right) \frac{\partial w^i}{\partial c_j^i}\\
&= \frac{y^i(\sum_{i=1}^l w^i) - \sum_{i=1}^l w^i y^i}{(\sum_{i=1}^l w^i)^2} \frac{\partial w^i}{\partial c_j^i},\\
\frac{\partial y_f}{\partial b_j^i} &= \frac{\partial y_f}{\partial w^i} \frac{\partial w^i}{\partial b_j^i}\\
&= \frac{\partial}{\partial w^i} \left(\frac{w^1 y^1 + w^2 y^2 + \cdots + w^l y^l}{\sum_{i=1}^l w^i}\right) \frac{\partial w^i}{\partial b_j^i}\\
&= \frac{y^i(\sum_{i=1}^l w^i) - \sum_{i=1}^l w^i y^i}{(\sum_{i=1}^l w^i)^2} \frac{\partial w^i}{\partial b_j^i},\\
\frac{\partial \mu_{x_j}^i}{\partial c_j^i} &= e^{-\frac{(x_j - c_j^i)^2}{b_j^i}} \left(\frac{2(x_j - c_j^i)}{b_j^i}\right) = \mu_{x_j}^i \left(\frac{2(x_j - c_j^i)}{b_j^i}\right),\\
\frac{\partial \mu_{x_j}^i}{\partial b_j^i} &= e^{-\frac{(x_j - c_j^i)^2}{b_j^i}} \left(\frac{(x_j - c_j^i)^2}{(b_j^i)^2}\right) = \mu_{x_j}^i \left(\frac{(x_j - c_j^i)^2}{(b_j^i)^2}\right),\\
\frac{\partial w^i}{\partial c_j^i} &= \frac{\partial w^i}{\partial \mu_{x_j}^i} \frac{\partial \mu_{x_j}^i}{\partial c_j^i} = \mu_{x_1}^i \mu_{x_2}^i \cdots \mu_{x_{j-1}}^i \mu_{x_{j+1}}^i \cdots \mu_{x_n}^i \frac{\partial \mu_{x_j}^i}{\partial c_j^i},\\
\frac{\partial w^i}{\partial b_j^i} &= \frac{\partial w^i}{\partial \mu_{x_j}^i} \frac{\partial \mu_{x_j}^i}{\partial b_j^i} = \mu_{x_1}^i \mu_{x_2}^i \cdots \mu_{x_{j-1}}^i \mu_{x_{j+1}}^i \cdots \mu_{x_n}^i \frac{\partial \mu_{x_j}^i}{\partial b_j^i}.
\end{aligned}$$

(20)

Then, we derive the iterative computation of the fuzzy parameters:

$$p_j^i(t) = p_j^i(t-1) + \eta\,(y - y_f)\,\frac{w^i x_j}{\sum_{i=1}^{l} w^i}, \tag{21}$$

$$c_j^i(t) = c_j^i(t-1) + \eta\,(y - y_f)\,\frac{2\left(y^i\left(\sum_{i=1}^{l} w^i\right) - \sum_{i=1}^{l} w^i y^i\right)\left(x_j - c_j^i\right) w^i}{b_j^i\left(\sum_{i=1}^{l} w^i\right)^2}, \tag{22}$$

$$b_j^i(t) = b_j^i(t-1) + \eta\,(y - y_f)\,\frac{\left(y^i\left(\sum_{i=1}^{l} w^i\right) - \sum_{i=1}^{l} w^i y^i\right)\left(x_j - c_j^i\right)^2 w^i}{\left(b_j^i\right)^2\left(\sum_{i=1}^{l} w^i\right)^2}, \tag{23}$$

where $i = 1, 2, \cdots, l,\ j = 1, 2, \cdots, n$. Equations (21)-(23) represent the integer correction method of parameters and coefficients in fuzzy neural network in (1), and the pseudo codes of the integer order gradient descent fuzzy neural network (IOFNN) algorithm are listed as follows:

Algorithm 1 The training process of the integer order learning algorithm

Input:
x_1, x_2, ..., x_n
Output:
y_f
Construct the model in (2)
Initialization: $p_j^i(0), b_j^i(0), c_j^i(0)$
Training times: N
Number of samples: M
$t = 1$
for $1 : N$ **do**
  for $1 : M$ **do**
    Calculate y_f by (3)
    Compute $p_j^i(t+1)$ by (21), $c_j^i(t+1)$ by (22), and $b_j^i(t+1)$ by (23)
    $t = t + 1$
  end for
end for
End

3.2. Fuzzy neural network based fractional order algorithm. The traditional integer order fuzzy neural network (IOFNN) has good performance in system identification, but the convergence rate and estimation accuracy haven't reached the desired level. Since fractional calculus has been a top choice to improve system performance both in control and identification, we take fractional order based algorithm into consideration in this section. The numerical solution of the fractional order differential equation given by the GL fractional derivative [30] can be written as:

$$f(t_m) = D_\theta^\alpha h^\alpha - \sum_{m=0}^{L} a_j^\alpha f(t_m - j), \tag{24}$$

where D_θ^α is equal to $D^\alpha\theta(t)$. Considering the GL fractional derivative (11) and its numerical solution (24), we rephrase the gradient descent method into a fractional order one:

$$\triangle\theta = \theta_t - \theta_{t-1} = -\eta\frac{\partial E}{\partial\theta}h^\alpha - D_\theta^\alpha h^\alpha. \tag{25}$$

Consider that E is a composite function of θ [31]:

$$\triangle\theta = -\eta\frac{\partial E}{\partial\theta}h^\alpha - D_\theta^\alpha h^\alpha = -\eta\frac{\partial E}{\partial y_f}\frac{\partial y_f}{\partial\theta}h^\alpha - D_\theta^\alpha h^\alpha. \tag{26}$$

Then, the modified fractional order estimates of the model parameters can be derived as:

$$p_j^i(t) = p_j^i(t-1) - \eta\frac{\partial E}{\partial y_f}\frac{\partial y_f}{\partial p_j^i}h^\alpha - D_{p_j^i}^\alpha h^\alpha, \tag{27}$$

$$c_j^i(t) = c_j^i(t-1) - \eta\frac{\partial E}{\partial y_f}\frac{\partial y_f}{\partial c_j^i}h^\alpha - D_{c_j^i}^\alpha h^\alpha, \tag{28}$$

$$b_j^i(t) = b_j^i(t-1) - \eta\frac{\partial E}{\partial y_f}\frac{\partial y_f}{\partial b_j^i}h^\alpha - D_{b_j^i}^\alpha h^\alpha, \tag{29}$$

where

$$\begin{aligned}
D_{p_j^i}^\alpha &= \frac{1}{h^\alpha}\sum_{m=0}^{L}(-1)^m\binom{\alpha}{m}p_j^i(t-mh),\\
D_{c_j^i}^\alpha &= \frac{1}{h^\alpha}\sum_{m=0}^{L}(-1)^m\binom{\alpha}{m}c_j^i(t-mh),\\
D_{b_j^i}^\alpha &= \frac{1}{h^\alpha}\sum_{m=0}^{L}(-1)^m\binom{\alpha}{m}b_j^i(t-mh).
\end{aligned}$$

Considering the Eq. (20), we derive the iterative computation of the fuzzy parameters:

$$\begin{aligned}
p_j^i(t) = {}& p_j^i(t-1) + \eta(y-y_f)\frac{w^i x_j}{\sum_{i=1}^{l} w^i}h^\alpha\\
&- \sum_{m=0}^{L}(-1)^m\binom{\alpha}{m}p_j^i(t-mh),
\end{aligned} \tag{30}$$

$$\begin{aligned}
c_j^i(t) = {}& c_j^i(t-1) + \eta(y-y_f)\frac{2(y^i(\sum_{i=1}^{l} w^i) - \sum_{i=1}^{l} w^i y^i)(x_j - c_j^i)w^i}{b_j^i(\sum_{i=1}^{l} w^i)^2}h^\alpha\\
&- \sum_{m=0}^{L}(-1)^m\binom{\alpha}{m}c_j^i(t-mh),
\end{aligned} \tag{31}$$

$$\begin{aligned}
b_j^i(t) = {}& b_j^i(t-1) + \eta(y-y_f)\frac{(y^i(\sum_{i=1}^{l} w^i) - \sum_{i=1}^{l} w^i y^i)(x_j - c_j^i)^2 w^i}{(b_j^i)^2(\sum_{i=1}^{l} w^i)^2}h^\alpha\\
&- \sum_{m=0}^{L}(-1)^m\binom{\alpha}{m}b_j^i(t-mh),
\end{aligned} \tag{32}$$

where $i = 1, 2, \cdots, l,\ j = 1, 2, \cdots, n$. Equations (30)-(32) represent the fractional correction method of parameters and coefficients in fuzzy neural network in (1), and the pseudo-code of the fractional order gradient descent fuzzy neural network (FOFNN) algorithm is listed as follows:

Algorithm 2 The training process of the fractional order learning algorithm

Input:
$x_1, x_2, ..., x_n$
Output:
y_f
Construct the model in (2)
Initialization: $p_j^i(0), b_j^i(0), c_j^i(0)$
Training times: N
Number of samples: M
$t = 1$
for $1 : N$ **do**
  for $1 : M$ **do**
    Calculate y_f by (3)
    Compute $p_j^i(t+1)$ by (30), $c_j^i(t+1)$ by (31), and $b_j^i(t+1)$ by (32)
    $t = t + 1$
  end for
end for
End

4. Simulation examples

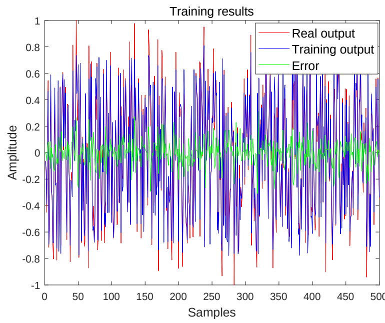
4.1. Nonlinear System Identification. Nonlinear systems have widely existed in modeling and analysis of complex systems, including but not limited to industrial processes, control systems, and mechanical systems. Traditional linear systems often cannot fully describe the complex dynamic characteristics of these systems, while nonlinear systems can more accurately characterize the nonlinear behavior and time-varying properties of the systems. Fuzzy neural networks have shown good performance in these cases [14, 15]. Consider a complex nonlinear system as follows:

$$y(k) = \frac{u(k-2)(y(k-3)-1)\sum_{i=1}^{3} y(k-i) + u(k-1)}{1 + y^2(k-2) + y^2(k-3)}, \tag{33}$$

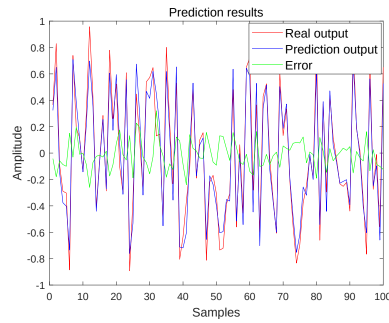
where $u(k)$ is a random sequence between (-1,1). The inputs of the FNN model are $y(k-1), y(k-2), y(k-3), u(k-1), u(k-2)$, and the output is $y(k)$. For comparison, appropriate integer order parameters are selected as $l = 5, \eta = 0.01$, the training times N of IOFNN is 10, the results are shown in Figure 3. The fractional order parameters are chosen as the same as integer one with extra fractional order $\alpha = 0.9878$, the results of fractional order gradient descent fuzzy neural network are shown in Figure 4. The root means square error (RMSE) is provided to measure the accuracy of the two algorithms, which can be defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y - y_f)^2}{N}}, \tag{34}$$

and the RMSEs of both algorithms are listed in Table 1. From Table 1, where the RMSE of fractional order gradient descent fuzzy neural network is smaller than that of integer order one. This implies that FOFNN can yield better accuracy.

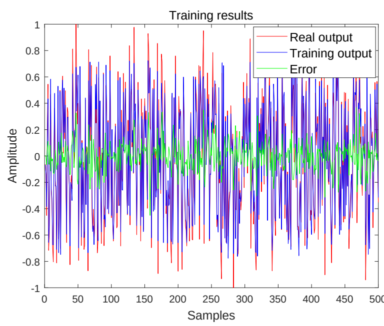


(a) Training results

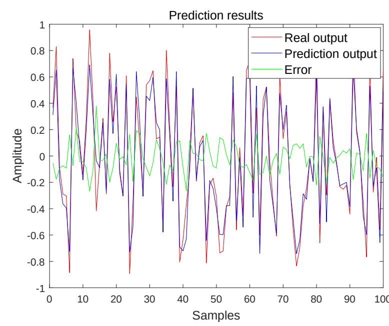


(b) Prediction results

FIGURE 3. Simulation results of IOFNN.



(a) Training results



(b) Prediction results

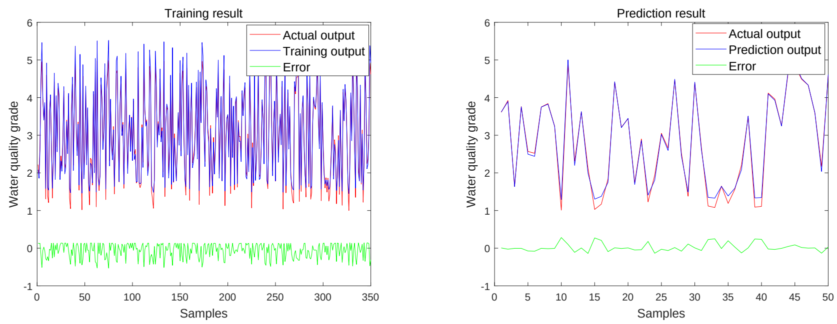
FIGURE 4. Simulation results of FOFNN.

TABLE 1. Performance of the IOFNN and FOFNN.

Model	n	l	RMSE
IOFNN	5	5	0.1036
FOFNN	5	5	0.0948

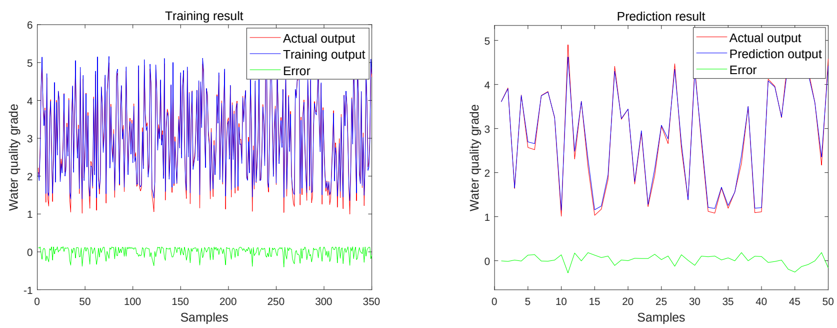
4.2. Water quality system. In order to verify the effectiveness of the proposed FOFNN, we select a group of data on water quality grade. The date comes from a book named MATLAB Neural Networks 43 Case Studies which is one of the famous dates for testing neural networks. The main factors affecting water quality include ammonia nitrogen, dissolved oxygen, chemical oxygen demand, permanganate index, total phosphorus, and total nitrogen. These indicators are input to the FNN model, while water quality grade is output. The data set contains 350 training data sets and 50 test data sets and the sample time is $h = 1$s. In this experiment, $n = 6$ inputs are determined to be the inputs of the FNN model, and the parameters by the integer-order algorithm are $l = 12, \eta = 0.01, N = 40$, additional parameter of the fractional-order model is set as $\alpha = 0.98$. Figures 5 and 6 show the simulation results of both algorithms, respectively.

In order to observe the difference between the two diagrams intuitively, we add the absolute value of the error of each training. The result is shown in Figure 7. From Figure 7 and Table 2, we can see that the performance of the fractional



(a) Training results (b) Prediction results

FIGURE 5. Water quality system simulation of IOFNN.



(a) Training results (b) Prediction results

FIGURE 6. Water quality system simulation of FOFNN.

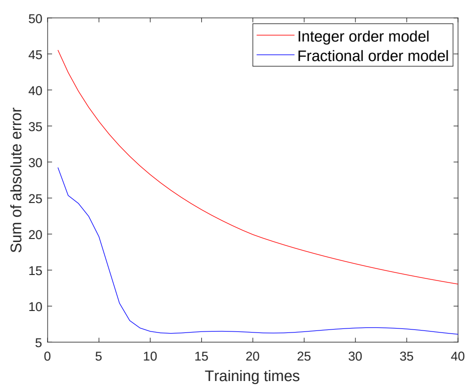


FIGURE 7. The sum of the absolute value of the error for each training.

TABLE 2. The performance of water quality system by IOFNN and FOFNN.

Model	n	l	RMSE
IOFNN	6	12	0.1381
FOFNN	6	12	0.1176

order is better than that of the positive integer order, and it has a better advantage in accuracy. Considering the energy loss in the actual system, the fractional-order system has better performance results.

Results:

As shown in the above two systems, the relevant fuzzy parameters are determined by well-designed rules and collected data, that is, p_j^i, c_j^i, b_j^i of each rule. Figures 3-6 show the training and prediction results of IOFNN and FOFNN, while Table 1 and 2 record the RMSEs of comparison of both algorithms in the two examples. We can conclude that FOFNN has a smaller RMSE than IOFNN, which means that FOFNN has a greater advantage in accuracy. In addition, Figure 7 is the sum of the absolute values of the errors in the model training process and the FOFNN algorithm has a faster convergence rate than IOFNN, indicating that FOFNN can appropriately reduce the complexity of the model as well as ensure the accuracy of identification.

5. Conclusions

In this paper, a fuzzy neural network is proposed to model complex nonlinear systems, which is trained and estimated by using fractional order gradient descent. The fractional learning algorithm has the advantage of multiple time points, which makes it more accurate than the integer order algorithm. In order to verify the effectiveness of the proposed algorithm, a numerical nonlinear model and a water quality system are considered. The experimental results show that the algorithm in this paper has better accuracy than the classic algorithm. Consider the novel fractional gradient descent-based learning algorithm in [32] and [33], which is the convex combination of the conventional and modified Riemann-Liouville derivative-based fractional gradient descent methods, convex combination in a fuzzy neural network will be discussed in our future work.

Acknowledgement

This work is supported by Natural Science Foundation of Jiangsu Province under Grant No. BK20211275 and Natural Science Foundation of NJUPT under Grant No. NY220217 and NY221079.

References

[1] T. C. Yuan, J. Yang and L. O. Chen, Nonparametric identification of nonlinear piezoelectric mechanical systems. *Journal of Applied Mechanics.* 2018; 85(11): 1-13.

[2] Y. Zhou, H. He, C. Sun, Fully distributed finite-time consensus of directed multiquadcopter systems via pinning control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems.* 2021; 51(8):5080-5089.

[3] X. Huo, H. R. Karimi, and X. Zhao et al., Adaptive Critic Design for Decentralized Event-Triggered Control of Constrained Nonlinear Interconnected Systems within An Identifier-Critic Framework. *IEEE Transactions on Cybernetics.* 2021; DOI:10.1109/TCYB.2020.3037321.

[4] Y. Wang, N. Xu, Y. Liu, and X. Zhao, Adaptive fault-tolerant control for switched nonlinear systems based on command filter technique. *Applied Mathematics and Computation.* 2021; 392, 125725.

[5] A. George and F. George, A Recurrent Neural Fuzzy Network. *Journal of Computational Analysis and Applications.* 2016; 20(1): 213-225.

[6] B. K. Andrews, Neural networks. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology.* 2018; 4(3): 19-22.

[7] D. Anuli and S. Smriti, Identification and control of dynamical systems using different architectures of recurrent fuzzy system. *ISA transactions.* 2019; 85: 107-118.

[8] A. Kulkarni and N. Kulkarni, Fuzzy Neural Network for Pattern Classification. *Procedia Computer Science.* 2020; 167: 2606-2616.

[9] Q. Pan, Y. L. Zhou and J. T. Fei, Feature Selection Fuzzy Neural Network Super-Twisting Harmonic Control. *Mathematics*. 2023; 11(6): 1495-1495.
[10] X. L. Zhou, X. K. Wang, H. T. Wang, L. L. Cao and Z. Y. Xing, Rotor Fault Diagnosis Method Based on VMD Symmetrical Polar Image and Fuzzy Neural Network. *Applied Sciences*. 2023; 13(2): 1134-1134.
[11] Q. Peng and J. G. Jian, Asymptotic synchronization of second-fractional-order fuzzy neural networks with impulsive effects. *Chaos, Solitons and Fractals*. 2023; 168
[12] T. Dam and A. K. Deb, A clustering algorithm based TS fuzzy model for tracking dynamical system data. *Journal of the Franklin Institute*. 2017; 354(13): 5617-5645.
[13] M. Li and X. Liu, Maximum likelihood hierarchical least squares-based iterative identification for dual-rate stochastic systems. *Journal of Computational Methods in Sciences and Engineering*. 2021; 18(2): 531-539.
[14] S. Muralisankar, N. Gopalakrishnan and P. Balasubramaniam, Robust Exponential Stability Criteria for T-S Fuzzy Stochastic Delayed Neural Networks of Neutral Type. *Circuits Syst Signal Process*. 2011; 30: 1617-1641.
[15] W. Li, J. F. Qiao, X. J. Zeng and S. L. Du, Identification and simplification of T-S fuzzy neural networks based on incremental structure learning and similarity analysis. *Fuzzy Sets and Systems*. 2020; 394(prepublish): 65-86.
[16] H. F. Li, C. D. Li, O. Y. Deqiang, S. K. Nguang and Z. L. He, Observer-Based Dissipativity Control for T-S Fuzzy Neural Networks With Distributed Time-Varying Delays. *IEEE transactions on cybernetics*. 2020; 51(11): 5248-5258.
[17] D.Q. Wang, L. Mao, F. Ding, Recasted models based hierarchical extended stochastic gradient method for MIMO nonlinear systems. *IET Control Theory and Applications*. 2017; 11(4): 476-485.
[18] D.Q. Wang, Z. Wei, Improved least squares identification algorithm for multivariable Hammerstein systems. *Journal of the Franklin Institute-Engineering and Applied Mathematics*. 2015; 352(11): 5292-5307.
[19] F. Liu, C. Li, and Y. Xu et al., A new lower and upper bound estimation model using gradient descend training method for wind speed interval prediction.*Journal of Wind Energy*. 2020; 24(3): 290-304.
[20] Y. W. Zheng, L. X. Zhang, S. Z. Chen, X. Zhang, Z. P. Cai and X. Z. Cheng, Robust decentralized stochastic gradient descent over unstable networks. *Computer Communications*. 2023; 203: 163-179.
[21] M. Lupupa, S. Hadjiloucas, Fractional-Order System Identification and Equalization in Massive MIMO Systems. *IEEE Access*. 2020; 8: 86481-86494.
[22] N. Gehring and J. Rudolph, An algebraic approach to the identification of linear systems with fractional derivatives. *IFAC-PapersOnLine*. 2017; 50(1): 6214-6219.
[23] J. Liang, B. Wu and Y. E. Wang YE et al., Input-Output Finite-Time Stability of Fractional-Order Positive Switched Systems. *Circuits Syst Signal Process*. 2019; 38: 1619-1638.
[24] C. J. Zuñiga Aguilar, J. F. Gómez-Aguilar, V. M. Alvarado-Martínez and H. M. Romero-Ugalde, Fractional order neural networks for system identification. *Chaos, Solitons and Fractals*. 2020; 130: 1-9.
[25] L. Liu, F. Pan and D. Xue, Variable-order fuzzy fractional PID controller. *ISA Transactions*. 2015; 55: 227-233.
[26] Q. Zou, J. F. Zhang, R. D. Zhang, F. Gao and A. Xue, Fractional order MPC design using improved state space model. *IFAC-PapersOnLine*. 2017; 50(1): 7535-7540.
[27] M. Kumar, A. Aggarwal, T. Rawat and H. Parthasarathy, Optimal nonlinear system identification using fractional delay second-order volterra system. *IEEE/CAA Journal of Automatica Sinica*. 2018; 1-18.
[28] S. J. Du and Q. B. Shen, Identification and control based on T-S fuzzy model. *Industrial Instrumentation and Automation*. 2011; 1(11): 105-108.
[29] E. N. Sadjadi, M. B. Menhaj, J. G. Herrero and J. M. Molina Lopez, How Effective are Smooth Compositions in Predictive Control of TS Fuzzy Models. *International Journal of Fuzzy Systems*. 2019; 21(6): 1669-1686.
[30] A. Coronel-Escamilla, J. F. Gómez-Aguilar, M. G. López-López, V. M. Alvarado-Martínez and G. V. Guerrero-Ramírez, Triple pendulum model involving fractional derivatives with different kernels. *Chaos, Solitons and Fractals*. 2016; 91:248-261.
[31] P. Hai and J. A. Rosenfeld, The gradient descent method from the perspective of fractional calculus. *Mathematical Methods in the Applied Sciences*. 2021; 44(7): 5520-5547.

[32] K. Shujaat, A. Jawwad, N. Imran and M. Muhammad, A Novel Fractional Gradient-Based Learning Algorithm for Recurrent Neural Networks. *Circuits Syst Signal Process.* 2018; 37: 593-612

[33] K. Shujaat, N. Imran, A. M. Muhammad, T. Roberto and B. Mohammed, A Fractional Gradient Descent-Based RBF Neural Network. *Circuits Syst Signal Process.* 2018; 37: 5311-5332

School of Automation and Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, 210023, PR China

E-mail: dingjie@njupt.edu.cn; 1022051421@njupt.edu.cn and 1019051233@njupt.edu.cn