Alexandria Engineering Journal (2020) 59, 2803–2810

HOSTED BY
ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

A kernel least mean square algorithm for fuzzy differential equations and its application in earth's energy balance model and climate

M. Pakdaman [a,*], **Y. Falamarzi** [b], **H. Sadoghi Yazdi** [c], **A. Ahmadian** [d,*], **S. Salahshour** [e], **F. Ferrara** [f]

[a] *Disasters and Climate Change Group, Climatological Research Institute, Mashhad, Iran*
[b] *Climate Change Division, Climatological Research Institute, Mashhad, Iran*
[c] *Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*
[d] *Institute of Industry Revolution 4.0, National University of Malaysia, 43600 UKM, Bangi, Selangor, Malaysia*
[e] *Faculty of Engineering and Natural Sciences, Bahcesehir University, Istanbul, Turkey*
[f] *Department of Law, Economics and Human Sciences& Decisions Lab University Mediterranea of Reggio Calabria, 89125 Reggio Calabria, Italy*

Received 18 May 2020; revised 1 June 2020; accepted 7 June 2020
Available online 2 July 2020

KEYWORDS

Least mean square;
Kernel space;
Fuzzy dynamical differential equation;
Energy balance model;
BFGS optimization algorithm

Abstract This paper concentrates on solving fuzzy dynamical differential equations (FDDEs) by use of unsupervised kernel least mean square (UKLMS). UKLMS is a nonlinear adaptive filter which works by applying kernel trick to LMS adaptive filter. UKLMS estimates multivariate function which is embedded to estimate the solution of FDDE. Adaptation mechanism of UKLMS helps for finding solution of FDDE in a recursive scenario. Without any desired response, UKLMS finds nonlinear functions. For this purpose, an approximate solution of FDDE is constructed based on adaptable parameters of UKLMS. An optimization algorithm, optimizes the values of adaptable parameters of UKLMS. The proposed algorithm is applied for solving Earth energy balance model (EBM) which is considered as a fuzzy differential equation for the first time. The method in comparison with the other existing approaches (such as numerical methods) has some advantages such as more accurate solution and also that the obtained solution has a functional form, thus the solution can be obtained at each time in training interval. Low error and applicability of developed algorithm are examined by applying it for solving several problems. After comparing the numerical results, with relative previous works, the superiority of the proposed method will be illustrated.

© 2020 The Authors. Published by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

A lot of real world practical models, can be modeled as a fuzzy differential equations (FDEs) (e.g. [7,10]). FDEs are an

* Corresponding authors.
E-mail addresses: pakdaman@cri.ac.ir (M. Pakdaman), ali.ahmadian@ukm.edu.my (A. Ahmadian).

☆ Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2020.06.016
1110-0168 © 2020 The Authors. Published by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

appropriate tool to construct a dynamical system with fuzzy uncertainty [54]. The first definition of fuzzy function was proposed by Chang and Zadeh [12]. By employing the extension principle [53], Dubois and Prade [15] introduced a type of fuzzy mathematics. Two special types of fuzzy derivative for fuzzy function were proposed by Puri and Ralescu [37] which definition of Hukuhara difference was used in the first type (and later extended by Kaleva [23]). Later, multiple different definitions were extended for the solution concepts for FDEs (e. g. [22,30,36,43,14]). Unfortunately, the Hukuhara based definitions for fuzzy derivative has a drawback that their proposed solutions have an increasing length of support. To overcome this lack, the authors of [8,9] defined the generalized differentiability. Based on generalized differentiability, the length of support of solution of FDE may decreases. However, there can be found other approaches and definitions for the solution set of FDE (e. g. [11,30]).

There are several numerical algorithms in literature for solving a FDE (e.g. [17,21,4]). Most of proposed numerical algorithms are extended forms of existing related approaches for ordinary differential equations (For example see [5,42,31,13,35]). Based on using the extension principle, Ahmad et.al [1] introduced an analytical and numerical approach for solving FDEs. Lupulescu and Abbas [29] discussed about fuzzy delay differential equations. Xu et al. [48] used a complex number representation for fuzzy system to analyze the solution of linear fuzzy dynamical systems containing fuzzy matrices. Khastan and Ivaz [25] analyzed proposed numerical methods along with their stability analysis for fuzzy first-order initial value problems. It must also be mentioned that the use of numerical methods was extended for solving fuzzy fractional differential equations (see [32,3,38,34]). Another group of authors tried to use the neural networks as a powerful tool in approximation theory. Lagaris et al. [26] proposed a method for the solution of ordinary and partial differential equations (ODEs and PDEs), based on perceptron neural networks. It is proved that multilayer perceptron is universal approximator [20]. The authors of this paper in [52] used a new format of unsupervised adaptive network-based fuzzy inference system (ANFIS) for the solution of ODEs. Widrow and Hoff [46] introduced the least mean squares (LMS) algorithm which has a wide applications in adaptive learning. Moreover, in last decade, kernel methods are extended and used in machine learning, such as support vector machines and K-PCA (kernel based principal component analysis) [40], regularization networks [18] and K-ICA (kernel based independent component analysis) [6]). The kernel based algorithms are tools for extending linear adaptive filters [6,41,45]. Pokharel et al. [27] proposed a "kernel trick" [45] to the LMS algorithm [47,19] to study the nonlinear adaptive filters in RKHS (reproducing kernel Hilbert spaces). This method which is a fusion of LMS and kernel tricks is the KLMS algorithm. The author of this paper in [51] applied an unsupervised form of KLMS algorithm (which there is no desired signal from user) for the solution of ODEs. They also in [16] proposed a new approach based on neural networks, for solving FDEs. Continuing our previous works [16,52,51,50,49], in the current paper, we develop a KLMS approach for estimating the solution of FDEs. However, there are several important applications of fuzzy differential equations and fuzzy dynamical systems in real world, in this paper we focus on Earth's Energy Balance Model [24]. The Earth surface temperature, which is an important parameter in climatology and Earth climate, is determined by the global energy balance between radiative energy coming from the Sun and radiative energy emitted back to space by the Earth [39]. In his paper we try to consider Earth's Energy Balance Model as a fuzzy differential equation.

The current paper focuses to extend the capability of KLMS algorithm to solve FDEs. The important preliminaries of KLMS and FDEs are presented in Section 2 and Section 3 presents the proposed new approach. Numerical examples are solved in Section 4 which contains an application in circuit analysis. Finally, Section 5 contains conclusions.

2. Preliminaries

2.1. Kernel least mean square

Consider an unknown system $K(n)$ which must adapt the filter $\widehat{K}(n)$ via LMS algorithm. Suppose that we denote by $O(n)$ the desired output, $y(n)$ as the input and $e(n)$ as error. Based on steepest-descent method for LMS algorithm, the weights can be updated by using the following recursive algorithm:

$$v(n+1) = v(n) + 2\mu \times e(n) \times y(n) \tag{1}$$

Here $v(n)$ and $y(n)$ stand for weight vector and input vector respectively. μ is a step size. For simplicity we denote filter $\widehat{K}(n)$, by $K(n)$ which can be calculated by:

$$K(n) = v \times y(n) \tag{2}$$

For more details about LMS algorithm see [46]. In kernel trick, Kernels approaches are used for mapping the input data into a high dimensional space (HDS) via Φ functions (see Fig. 1). Finally in a HDS to find a linear relation in data, several methods can be employed.

The base of KLMS algorithm is using the linear LMS algorithm in kernel space.

$$w(n+1) = w(n) + 2\mu \times e(n) \times \psi(y(n)), \tag{3}$$

where $w(n)$ is a vector containing the weights HDS. The approximated output $K(n)$ can be calculated as follows:

$$K(n) = \langle W(n), \psi(y(n)) \rangle. \tag{4}$$

Based on Fig. 1 the input variable $y(n)$ is transformed to an infinite feature vector $\psi(y(n))$, whose elements are a linear combination of infinite dimensional weight vector. We can write (3) as the following non-recursive equation:

$$w(n) = w(0) + 2\mu \sum_{i=0}^{n-1} e(i)\psi(y(i)). \tag{5}$$

Let $w(0) = 0$, then:

$$w(n) = 2\mu \sum_{i=0}^{n-1} e(i)\psi(y(i)). \tag{6}$$

Considering (4) and (6) we have:

$$K(n) = \langle W(n), \psi(y(n)) \rangle = \left\langle 2\mu \sum_{i=0}^{n-1} e(i)\psi(y(i)), \psi(y(n)) \right\rangle$$
$$= 2\mu \sum_{i=0}^{n-1} e(i) \langle \psi(y(i)), \psi(y(n)) \rangle. \tag{7}$$

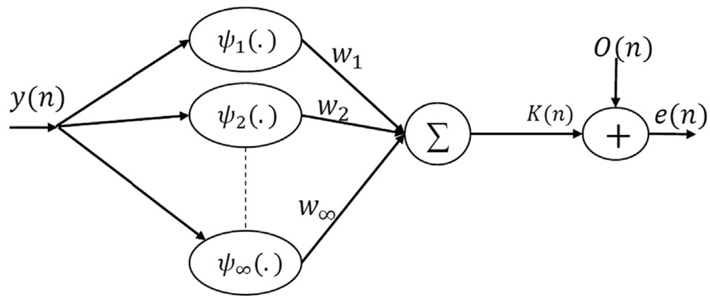

Fig. 1 Kernel estimation system.

Now, to calculate $K(n)$, we can apply kernel tricks as follows:

$$K(n) = \mu\sum_{i=0}^{n-1} e(i)ker(y(i), y(n)) \tag{8}$$

In (8), which is the KLMS algorithm, stands for kernel function. If we ignore $e(n)$ after a number of m samples and predict new input by using th previous error:

$$K(n) = \mu\sum_{i=0}^{m} e(i)ker(y(i), y(n)) \tag{9}$$

This trick discounts the difficulties of the proposed approach. For simplicity in notation, if we replace $e(i)$ in (9) by $w(i)$, then $K(n)$ can be considered as a function of x and a vector W which contains the error items $w(i)$. Thus (9) can be written as follows [51,52]:

$$K(y, W) = \mu\sum_{i=0}^{m} w(i)ker(y(i), y(n)). \tag{10}$$

2.2. Fuzzy set theory

Now, some requirements from fuzzy mathematics will be mentioned here.

Definition 2.1. (See [44])

We can denote a fuzzy number m by $(\underline{m}, \overline{m})$ of two functions $\underline{m}(r), \overline{m}(r) : [0, 1] \to R$ which they must satisfy the following conditions: a. $\underline{m}(r)$ is a function with three conditions: monotonicity, boundedness, increasing (non-decreasing) and for all $r \in (0, 1])$ it is a left-continuous function while for $r = 0$ it is right-continuous. b. $\overline{m}(r)$ is a function with three conditions: monotonicity, boundedness, decreasing (non-increasing) and for all $r \in (0, 1])$ it is a left-continuous function while for $r = 0$ it is right-continuous. c. For all $r \in [0, 1]$ we have $\underline{m}(r) \leqslant \overline{m}(r)$.

Also, for any two fuzzy numbers $m = (\underline{m}, \overline{m})$ and $n = (\underline{n}, \overline{n})$ addition and multiplication can be defined as follows:

$$\begin{aligned}(\underline{m+n})(r) &= \underline{m}(r) + \underline{n}(r), (\overline{m+n})(r)\\ &= \overline{m}(r) + \overline{n}(r), (\underline{km})(r) = k\underline{m}(r), (\overline{km})(r)\\ &= k\overline{m}(r).\end{aligned} \tag{11}$$

We denote by E^1, the set of fuzzy numbers with the multiplication and also the addition which were defined in (11). For $0 < r \leq 1$, we introduce the r-cuts of fuzzy number m with $[m]^r = \{x \in r | m(x) \geqslant r\}$ and for $r = 0$ the support of m is defined as $[m]^0 = \overline{\{x \in r | m(x) \geqslant 0\}}$.

Definition 2.2. The distance between two given fuzzy numbers $m = (\underline{m}, \overline{m})$ and $n = (\underline{n}, \overline{n})$ can be defined as follows:

$$d(m, n) = \sup_{0 \leqslant r \leq 1} \{\max[|\underline{m}(r) - \underline{n}(r)|, |\overline{m}(r) - \overline{n}(r)|]\} \tag{12}$$

Definition 2.3. The function $f : R \to E^1$ can be called as a fuzzy function. If for an arbitrary fixed $\hat{x}$and $\epsilon > 0$ there exists a $\delta > 0$ such that:

$$|x - \hat{x}| < \delta \Rightarrow d(f(x), f(\hat{x})) < \epsilon \tag{13}$$

then f is said to be continuous.

Definition 2.4. If there exists a fuzzy number $u \in E^1$, for $m, n \in E^1$, such that $m = n + u$ then u is said to be the H-difference of m and n which is denoted by $m - n$.

Definition 2.5. (see [2]) A fuzzy function $G : (a, b) \to E^1$ is called differentiable at $\hat{x} \in (a, b)$ if there exist an element $G'(\hat{x}) \in E^1$ such that:

(I) for all $h > 0$ sufficiently small, the H-differences $G(\hat{x} + h) - G(\hat{x})$, $G(\hat{x}) - G(\hat{x} - h)$ exist and:

$$\begin{aligned}&\lim_{h\to 0^+} d\left(\frac{G(\hat{x} + h) - G(\hat{x})}{h}, G'(\hat{x})\right)\\ &\quad = \lim_{h\to 0^+} d\left(\frac{G(\hat{x}) - G(\hat{x} - h)}{h}, G'(\hat{x})\right) = 0.\end{aligned} \tag{14}$$

(II) for all $h > 0$ sufficiently small, the H-differences $G(\hat{x}) - G(\hat{x} + h)$ and $G(\hat{x} - h) - G(\hat{x})$ exist and:

$$\begin{aligned}&\lim_{h\to 0^+} d\left(\frac{G(\hat{x}) - G(\hat{x} + h)}{h}, G'(\hat{x})\right)\\ &\quad = \lim_{h\to 0^+} d\left(\frac{G(\hat{x} - h) - G(\hat{x})}{h}, G'(\hat{x})\right) = 0.\end{aligned} \tag{15}$$

In this situation, $G'(\hat{x})$ is called the type-I and type-II fuzzy derivative of G at $\hat{x}$ respectively.

Theorem 2.1. (see [2]) Let $G : (a, b) \to E^1$ and suppose that $[G(x)]^r = [\underline{G}(x, r), \overline{G}(x, r)]$ for $r \in [0, 1]$.

(i) *If G is (I)-differentiable at all* $x \in [a,b]$ *then* $G_L(x,r)$ *and* $G_R(x,r)$ *are differentiable functions and we have* $[G'(x)]^r = [\underline{G}'(x,r), \overline{G}'(x,r)]$.

(ii) *If G is (II)-differentiable at all* $x \in [a,b]$ *then* $G_L(x,r)$ *and* $G_R(x,r)$ *are differentiable functions and we have* $[G'(t)]^r = [\overline{G}'(x,r), \underline{G}'(x,r)]$.

2.3. Fuzzy differential equations

After introducing a first order FDE, to show the importance and applicability of FDEs we will discuss about some applications in this section. A first order FDE can be denoted as follows (see [33]):

$$\begin{cases} x'(t) = G(t,x), t \in [t_0, T] \\ x(t_0) = x_0, \end{cases} \tag{16}$$

where x_0 is a fuzzy number, x is a fuzzy function of crisp variable t and also, $G(t,x)$ can be considered as a fuzzy function. Here x' denotes fuzzy derivative (based on Definition 2.5) of x. For more details about the sufficient conditions for the existence and uniqueness of solution to Eq. (16), see [23]. These conditions can be expressed as follows:

- G is continuous
- G should satisfy the Lipschitz condition $d(G(t,x), G(t,y)) \leqslant Md(x,y)$ for some $M > 0$ $(x, y \in E^1)$.

If we suppose that G is (I)-differentiable, then system (16) can be replaced by the following equations:

$$\begin{cases} \underline{x}'(t,r) = \underline{G}(t,x) = L(t,\underline{x},\overline{x}), \underline{x}(t_0,r) = \underline{x}_0(r) \\ \overline{x}'(t,r) = \overline{G}(t,x) = U(t,\underline{x},\overline{x}), \overline{x}(t_0,r) = \overline{x}_0(r) \end{cases} \tag{17}$$

where

$$\begin{cases} \underline{G}(t,x) = min\{G(t,u) | u \in [\underline{x}(r), \overline{x}(r)]\} \\ \overline{G}(t,x) = max\{G(t,u) | u \in [\underline{x}(r), \overline{x}(r)]\} \end{cases} \tag{18}$$

Thus, the parametric form of (17) is as follows:

$$\begin{cases} \underline{x}'(t,r) = L(t,\underline{x}(t,r),\overline{x}(t,r)), \underline{x}(t_0,r) = \underline{x}_0(r) \\ \overline{x}'(t,r) = U(t,\underline{x}(t,r),\overline{x}(t,r)), \overline{x}(t_0,r) = \overline{x}_0(r) \end{cases} \tag{19}$$

where $t \in [t_0, T]$ and $r \in [0,1]$. Now we select some points $t_i, i = 1,2,\ldots,m$ from the interval $[t_0, T]$. Thus for any $t_i \in [t_0, T]$, Eq. (19) must satisfied as follows:

$$\begin{cases} \underline{x}'(t_i,r) - L(t_i,\underline{x}(t_i,r),\overline{x}(t_i,r)) = 0, \\ \overline{x}'(t_i,r) - U(t_i,\underline{x}(t_i,r),\overline{x}(t_i,r)) = 0, \end{cases} \tag{20}$$

along with the initial conditions:

$$\begin{cases} \underline{x}(t_0,r) = \underline{x}_0(r) \\ \overline{x}(t_0,r) = \overline{x}_0(r) \end{cases} \tag{21}$$

for $0 \leqslant r \leq 1$. In next section, we try to solve (20) with a method based on KLMS approach. Note that type(II) differentiability similarly can be considered.

3. New method

In the current section, we use (10) as the main part of the trial solutions for FDE. We employ (10) as an approximator for the solution of FDE (20). To solve ((19)) it is enough to find the solution $x(t,r) = (\underline{x}(t,r), \overline{x}(t,r))$ of system (20). For this purpose first we introduce the trial solutions $\underline{x}_T(t,r,\underline{w})$ and $\overline{x}_T(t,r,\overline{w})$ for $\underline{x}(t,r)$ and $\overline{x}(t,r)$ respectively. These trial solutions have this property that satisfy the initial conditions and have parameters which can be adjusted so as they also satisfy the FDE (20). Indeed we use the terms $w(i)$ in (10) as the adjustable parameters of the trial solutions. Now to solve (20) by using KLMS, we propose the following trial solutions:

$$\begin{cases} \underline{x}_T(t,r,\underline{w}) = \underline{x}(t_0,r) + (t-t_0)\underline{K}(t,r,\underline{w}), \\ \overline{x}_T(t,r,\overline{w}) = \overline{x}(t_0,r) + (t-t_0)\overline{K}(t,r,\overline{w}), \end{cases} \tag{22}$$

where $\underline{x}_T(t,r,\underline{w})$ and $\overline{x}_T(t,r,\overline{w})$ are the trial solutions for $\underline{x}(t,r)$ and $\overline{x}(t,r)$ respectively. $\underline{K}$ and $\overline{K}$ are KLMS parts in the following forms:

$$\begin{cases} \underline{K}(t,r,\underline{w}) = \mu \sum_{i=0}^{n-1} \underline{w}(i) \underline{ker}(t_i,t) \\ \overline{K}(t,r,\overline{w}) = \mu \sum_{i=0}^{n-1} \overline{w}(i) \overline{ker}(t_i,t). \end{cases} \tag{23}$$

As we can see, the KLMS parts $\underline{K}$ and $\overline{K}$, contain appropriate kernel functions $\underline{ker}$ and $\overline{ker}$ which can be different. Here $\underline{w} = (\underline{w}_0, \underline{w}_1, \ldots, \underline{w}_{n-1})$ and $\overline{w} = (\overline{w}_0, \overline{w}_1, \ldots, \overline{w}_{n-1})$ are error vector as in (10). Based on this approach, each trial solution $\underline{x}_T(t,r,\underline{w})$ and $\overline{x}_T(t,r,\overline{w})$ in (22), is contained of two parts. The first term is for satisfaction in the initial conditions and has no adaptable parameters. The second part, contains of KLMS functions which the errors vector $(\underline{w}, \overline{w})$ must be adapted such that the trial solutions satisfy (22). By this approach, the error vector $W = (\underline{w}, \overline{w})$ must be adapted such that $x_T(t,r,W) = (\underline{x}_T(t,r,\underline{w}), \overline{x}_T(t,r,\overline{w}))$ satisfies the FDE in (16) with related initial values. Based on (22) we have:

$$\begin{cases} \underline{x}'_T(t,r,\underline{w}) = \underline{K}(t,r,\underline{w}) + (t-t_0)\frac{\partial}{\partial t}\underline{K}(t,r,\underline{w}), \\ \overline{x}'_T(t,r,\overline{w}) = \overline{K}(t,r,\overline{w}) + (t-t_0)\frac{\partial}{\partial t}\overline{K}(t,r,\overline{w}), \end{cases} \tag{24}$$

where

$$\begin{cases} \frac{\partial}{\partial t}\underline{K}(t,r,\underline{w}) = \mu \sum_{i=0}^{n-1} \underline{w}(i) \frac{\partial}{\partial t}\underline{ker}(t_i,t), \\ \frac{\partial}{\partial t}\overline{K}(t,r,\overline{w}) = \mu \sum_{i=0}^{n-1} \overline{w}(i) \frac{\partial}{\partial t}\overline{ker}(t_i,t), \end{cases} \tag{25}$$

Now to solve (15) first, we discretize interval $[t_0, T]$ into m equal parts and propose the following optimization problem:

$$\min_W \psi(W) = \sum_{i=1}^{m} \Big\{ \big(\underline{x}'_T(t_i,r,\underline{w}) - F[t_i, \underline{x}(t_i,r,\underline{w}), \overline{x}(t_i,r,\overline{w})]\big)^2 + \big(\overline{x}'_T(t_i,r,\overline{w}) - G[t_i, \underline{x}(t_i,r,\underline{w}), \overline{x}(t_i,r,\overline{w})]\big)^2 \Big\} \tag{26}$$

Substituting (24) and (25) in (26), it can be rewritten in the following form:

$$\begin{aligned} \min_W \psi(W) = \sum_{i=1}^{m} \Bigg\{ & \bigg(\underline{K}(t_i,r,\underline{w}) + (t_i - t_0)\frac{\partial}{\partial t}\underline{K}(t_i,r,\underline{w}) \\ & - F[t_i, \underline{x}(t_i,r,\underline{w}), \overline{x}(t_i,r,\overline{w})] \bigg)^2 \\ & + \bigg(\overline{K}(t_i,r,\overline{w}) + (t_i - t_0)\frac{\partial}{\partial t}\overline{K}(t_i,r,\overline{w}) \\ & - G[t_i, \underline{x}(t_i,r,\underline{w}), \overline{x}(t_i,r,\overline{w})] \bigg)^2 \Bigg\} \end{aligned} \tag{27}$$

The optimization problem (27) is unconstrained. Thus to solve it, we can employ any optimization algorithm such as conjugate gradient, steepest descent method or quasi Newton methods. Also, we can employ any heuristic algorithm such as Genetic Algorithm or Particle Swarm Optimization algorithm. Here, we used a Quasi-Newton method (BFGS algorithm) which is quadratically convergent (see [28]). After termination of the optimization section, we can replace the optimal value of vector in (22). Finally, the trial solution (22) is the approximated solution of (16). In the following remark, we discuss about the motivation and the advantages of using KLMS as a tool for solving FDEs.

Remark 3.1. Comparing the obtained results in KLMS approach in Section 4, with existing numerical algorithms, illustrates the accuracy of the method. One important property of obtained solution in KLMS approach is that the proposed solution is a smooth function. If we need to calculate the derivatives or integral of the final solution of FDE, we can calculate them directly without using numerical methods. In comparison with neural network methodology [16], we must mention that in KLMS approach we just use one type of weights (vector $W = (\underline{w}, \overline{w})$) and the number of weights are less than the weights of neural networks (which have at least three types of weights: input and output layers weights and bias weights). Finally the method has a flexibility that can be adopted to solve several types of the FDE (e.g. we can change the number of points for training, as well as the number of adjustable parameters w_i, or we can change the kernel function). Also we can use several optimization algorithms to obtain more accurate results or the number of iterations of the optimization step can be increased.

Remark 3.2. In (23) we can use any well-known kernel function. In this article we used the Gaussian kernel function with the following formula:

$$ker(a,b) = exp\left(\frac{||a-b||^2}{-2\sigma^2}\right) \tag{28}$$

We can use other kernel functions.

Remark 3.3.

The proposed method is capable for applying other types and definitions of fuzzy derivatives. In this case we may change system (20) for other definitions of fuzzy derivative.

4. Numerical simulations

For the sake of illustrate, four problems will be discussed and solved here. For all problems the training intervals are discretized to 20 identical sections. For minimizing the performance function in (25), we used MATLAB8 optimization toolbox by using BFGS algorithm. For each example we compared the numerical results with other existing algorithms and illustrated the advantages. Note that to attain a more exact solution we can use more grid points or a greater number of weights and change the optimization algorithm.

Example 4.1. Consider the following FDE:

$$\begin{cases} y'(x) = y(x), x \in [0,1] \\ y(0) = (0.75 + 0.25r, 1.125 - 0.125r) \end{cases} \tag{29}$$

Exact solution for $x = 1$ is $y(1,r) = ((0.75 + 0.25r)e, (1.125 - 0.125r)e$, where $r \in [0,1]$. According to (22) the constructed trial solutions are as follows:

$$\begin{cases} \underline{y}_T(x,r,\underline{w}) = 0.75 + 0.25r + x\underline{K}(x,r,\underline{w}), \\ \overline{y}_T(x,r,\overline{w}) = 1.125 - 0.125r + x\overline{K}(x,r,\overline{w}), \end{cases} \tag{30}$$

The approximated solution of Example 1, for $x = 1$ and several values for r, is depicted in Fig. 2. Error for $x = 1$ is shown in Table 1. In Table 1, The KLMS numerical results are compared with neural network method [16]. Note that in neural network algorithm, the number of weights are more that KLMS algorithm, thus the optimization step is longer that KLMS optimization step.

Example 4.2. Consider the following fuzzy initial value problem:

$$\begin{cases} x'(t) = 3t^2x(t), t \in [0,1] \\ x(0) = \left(0.5\sqrt{r}, 0.5 + 0.2\sqrt{1-r}\right) \end{cases} \tag{31}$$

For $t = 1$, Fig. 3 depicts the final approximated solution. Also the details of numerical computations are illustrated in Table 2. (see Fig. 4)

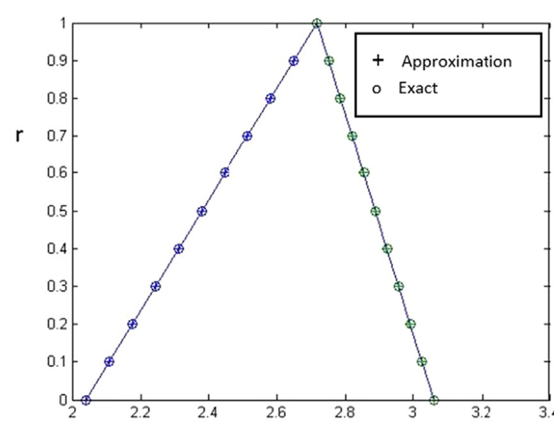

Fig. 2 Exact and approximated solution of Example 4.1 at $x = 1$.

Table 1 Numerical results for different values of r for Example 4.1 at $x = 1$.

r	KLMS Output		NN Output [16]	
	Lower	Upper	Lower	Upper
0	9.044042e−07	3.379464e−07	8.895270e−5	6.003329e−5
0.2	3.914999e−06	1.457328e−05	4.693243e−5	2.582699e−5
0.4	1.147732e−05	1.508659e−05	4.739291e−6	3.384699e−5
0.6	1.219494e−05	1.423798e−05	2.827934e−5	4.194040e−5
0.8	1.444412e−06	1.450936e−05	9.712388e−5	3.025500e−5
1	2.620011e−06	1.498573e−05	5.574679e−5	2.787937e−5

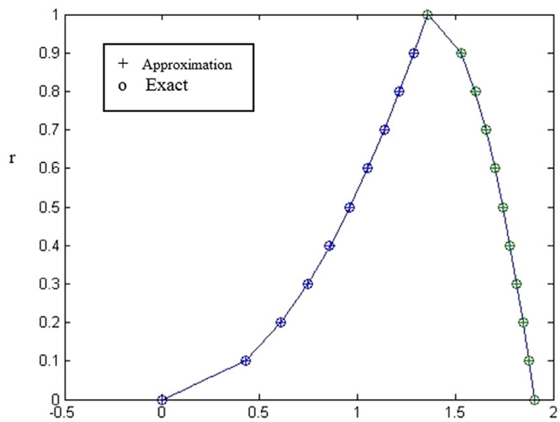

Fig. 3 Exact and approximated solution of Example 4.2 at $t = 1$.

Table 2 Numerical results for different values of r for Example 4.2 at $t = 1$.

r	KLMS Output		NN Output [16]	
	Lower	Upper	Lower	Upper
0	5.841994e−12	3.316067e−06	3.787978e−7	3.798493e−4
0.2	8.063511e−06	1.3979e−06	1.213900e−4	1.427181e−3
0.4	8.802233e−06	1.1875e−06	7.511922e−4	2.649743e−4
0.6	4.796803e−06	1.0540e−06	6.748295e−5	2.935789e−5
0.8	6.757338e−06	9.025472e−06	1.773549e−5	4.297191e−4
1	4.467423e−06	7.162632e−06	2.244555e−6	3.295712e−4

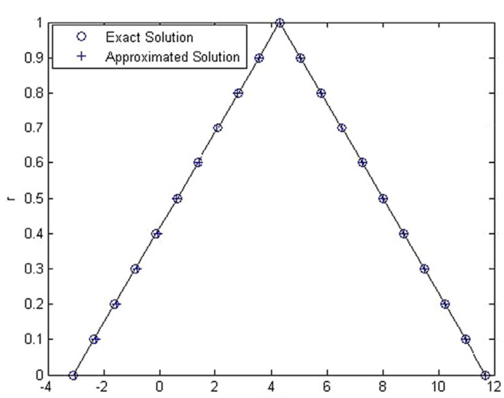

Fig. 4 Exact and approximated solution of Example 4.3 at $t = 4$.

Example 4.3. Consider the following FDE:

$$\begin{cases} x'(t) = 0.5x(t) + 2sin(t), t \in [0,4] \\ x(0) = (-1+r, 1-r) \end{cases} \tag{32}$$

Final solution for $t = 4$ is calculated in [1]] for several values of r via a numerical algorithm. Our results and the numerical results from [1] are compared in Table 3.

Example 4.4. The classical energy balance model (EBM) can be expressed as follows [24]:

$$C\frac{dT}{dt} = E_{in} - E_{out}, \tag{33}$$

Table 3 Comparison of our numerical results with the results in [1] for Example 4.3, at $t = 4$.

r	KLMS Output		Error of numerical method [1]	
	Lower	Upper	Lower	Upper
0	4.902254e−04	5.374543e−03	0.0140	0.3029
0.2	1.537021e−03	9.638080e−03	0.0429	0.2740
0.4	2.144774e−03	4.556884e−03	0.0718	0.2451
0.6	5.627119e−03	8.982244e−03	0.1007	0.2162
0.8	3.991684e−03	6.074451e−03	0.1296	0.1873
1	3.529961e−03	4.498548e−03	0.1584	0.1584

where, E_{in} is the average amount of solar energy reaching one square meter of the Earth's surface per unit time and similarly, E_{out} is the average amount of energy reflected by one square meter of the Earth's surface and released into the stratosphere per unit time [24]. $C = 2.912$ is the averaged heat capacity of the Earth/atmosphere system. The basic form of the EBM (33) is in the following form [24]:

$$C\frac{dT}{dt} = (1-\alpha)Q - \sigma T^4, \tag{34}$$

where T is the earth surface temperature, σ is the Stefan's constant, $\sigma = 5.67 \times 10^{-8}\ \mathrm{W\,m^{-2}\,K^{-4}}$. Time ($t$) is in year and $Q = 342\ \mathrm{W/m^2}$ is the annual global mean incoming solar radiation per square meter of the Earth's surface and finally, α is albedo which is a dimensionless parameter. Based on several physical reasons such as global warming, greenhouse gases and imprecise initial values, the earth surface temperature T is not a crisp function and has uncertainty. In this case, if we consider the earth surface temperature T as a fuzzy function of time, then Eq. (34) can be rewritten as follows:

$$C\frac{d\tilde{T}}{dt} = (1-\alpha)Q - \sigma\tilde{T}^4, \tilde{T}(0) = \tilde{T}_0. \tag{35}$$

which is a fuzzy initial value problem and can be solved by proposed algorithm. For $\alpha = 0.7$ the final fuzzy solution $\tilde{T}(1)$, which is a fuzzy number, is depicted in Fig. 5.

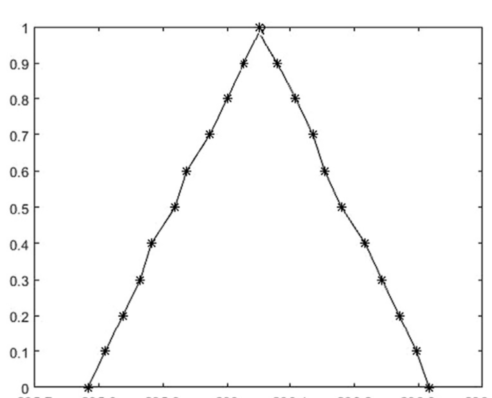

Fig. 5 Approximated solution of Example 4.4 at $t = 1$.

5. Concluding remarks

This paper developed a new method with a kernel least mean square structure, to approximate the solution of fuzzy differential equations. Here we used an optimization technique to adjust the parameters of KLMS. The proposed algorithm was evaluated by solving four problems and the final results compared with the other algorithms. Numerical results show that the method in comparison with the other numerical methods has more accuracy ([41,44]). Also in comparison with our previous paper (which was based on neural networks [36]), the structure of the trial solution is more simple, for example, here we don't need the vector of bias weight or a weight vector for output and input data (which they are used in neural networks models). As future works, this method can be applied to solve higher order FDEs and fuzzy integral equations. To attain more accurate solutions, we can use more points in the proposed interval, change the kernel functions or employ other existing optimization algorithms. Work is in progress to apply the method for solving n-order FDEs as well as for partial FDEs.

Author's Contribution

All authors contributed equally to this wok.

Availability of data

The data that supports the findings of this study are available within the article.

Funding

The first author acknowledges the financial support from Iran National Elites Foundation under Dr. Kazemi Ashtiani Award.

Declaration of Competing Interest

The authors declare that they have no conflict of interest.

References

[1] M.Z. Ahmad, M.K. Hasan, B. De Baets, Analytical and numerical solutions of fuzzy differential equations, Inf. Sci. 236 (2013) 156–167.
[2] M.Z. Ahmad, M.K. Hasan, B. De Baets, Analytical and numerical solutions of fuzzy differential equations, Inf. Sci. 236 (2013) 156–167.
[3] A. Ahmadian, C.S. Chan, S. Salahshour, V. Vaitheeswaran, Ftfbe: A numerical approximation for fuzzy time-fractional bloch equation, in: 2014 IEEE international conference on fuzzy systems (FUZZ-IEEE), IEEE, 2014, pp. 418–423.
[4] A. Ahmadian, S. Salahshour, C.S. Chan, D. Baleanu, Numerical solutions of fuzzy differential equations by an efficient runge–kutta method with generalized differentiability, Fuzzy Sets Syst. 331 (2018) 47–67.
[5] T. Allahviranloo, B. Ghanbari, On the fuzzy fractional differential equation with interval atangana–baleanu fractional derivative approach, Chaos, Solit. Fract. 130 (2020) 109397.
[6] F.R. Bach, M.I. Jordan, Kernel independent component analysis, J. Mach. Learn. Res. 3 (Jul) (2002) 1–48.
[7] L. Barros, R. Bassanezi, P. Tonelli, Fuzzy modelling in population dynamics, Ecol. Model. 128 (1) (2000) 27–33.
[8] B. Bede, S.G. Gal, Generalizations of the differentiability of fuzzy-number-valued functions with applications to fuzzy differential equations, Fuzzy Sets Syst. 151 (3) (2005) 581–599.
[9] B. Bede, I.J. Rudas, A.L. Bencsik, First order linear fuzzy differential equations under generalized differentiability, Inform. Sci. 177 (7) (2007) 1648–1662.
[10] J.J. Buckley, T. Feuring, Fuzzy differential equations, Fuzzy Sets Syst. 110 (1) (2000) 43–54.
[11] Y. Chalco-Cano, H. Román-Flores, On new solutions of fuzzy differential equations, Chaos, Solit. Fract. 38 (1) (2008) 112–119.
[12] S.S. Chang, L.A. Zadeh, On fuzzy mapping and control. In Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh, World Scientific, 1996, pp. 180–184.
[13] M. Chehlabi, T. Allahviranloo, Positive or negative solutions to first-order fully fuzzy linear differential equations under generalized differentiability, Appl. Soft Comput. 70 (2018) 359–370.
[14] W. Congxin, S. Shiji, Existence theorem to the cauchy problem of fuzzy differential equations under compactness-type conditions, Inf. Sci. 108 (1–4) (1998) 123–134.
[15] D. Dubois, H. Prade, Towards fuzzy differential calculus part 3: Differentiation, Fuzzy Sets Syst. 8 (3) (1982) 225–233.
[16] S. Effati, M. Pakdaman, Artificial neural network approach for solving fuzzy differential equations, Inf. Sci. 180 (8) (2010) 1434–1457.
[17] M. Friedman, M. Ma, A. Kandel, Numerical solutions of fuzzy differential and integral equations, Fuzzy Sets Syst. 106 (1) (1999) 35–48.
[18] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, Neural Comput. 7 (2) (1995) 219–269.
[19] A. Gunduz, J.-P. Kwon, J.C. Sanchez, J.C. Principe, Decoding hand trajectories from ecog recordings via kernel least-mean-square algorithm, in: 2009 4th International Ieee/Embs Conference on Neural Engineering, IEEE, 2009, pp. 267–270.
[20] K. Hornik, M. Stinchcombe, H. White, et al, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359–366.
[21] E. Hüllermeier, Numerical methods for fuzzy initial value problems, Int. J. Uncert., Fuzziness Knowledge-Based Syst. 7 (05) (1999) 439–461.
[22] L.J. Jowers, J.J. Buckley, K.D. Reilly, Simulating continuous fuzzy systems, Inf. Sci. 177 (2) (2007) 436–448.
[23] O. Kaleva, Fuzzy differential equations, Fuzzy Sets Syst. 24 (3) (1987) 301–317.
[24] H. Kaper, H. Engler, Mathematics and Climate, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2013.
[25] A. Khastan, K. Ivaz, Numerical solution of fuzzy differential equations by nyström method, Chaos, Solit. Fract. 41 (2) (2009) 859–868.
[26] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Networks 9 (5) (1998) 987–1000.
[27] W. Liu, P.P. Pokharel, J.C. Principe, The kernel least-mean-square algorithm, IEEE Trans. Signal Process. 56 (2) (2008) 543–554.
[28] D.G. Luenberger, Y. Ye, et al, Linear and nonlinear programming, volume 2, Springer, 1984.
[29] V. Lupulescu, U. Abbas, Fuzzy delay differential equations, Fuzzy Optim. Decis. Making 11 (1) (2012) 99–111.
[30] M.T. Mizukoshi, L. d. Barros, Y. Chalco-Cano, H. Román-Flores, and R.C. Bassanezi. Fuzzy differential equations and the extension principle. Information Sciences, 177(17):3627–3635, 2007.

[31] R.G. Moghaddam, T. Allahviranloo, On the fuzzy poisson equation, Fuzzy Sets Syst. 347 (2018) 105–128.
[32] N. Najafi, T. Allahviranloo, Combining fractional differential transform method and reproducing kernel hilbert space method to solve fuzzy impulsive fractional differential equations, Comput. Appl. Math. 39 (2020) 1–25.
[33] J.J. Nieto, The cauchy problem for continuous fuzzy differential equations, Fuzzy Sets Syst. 102 (2) (1999) 259–262.
[34] M. Pakdaman, A. Ahmadian, S. Effati, S. Salahshour, D. Baleanu, Solving differential equations of fractional order using an optimization technique based on training artificial neural network, Appl. Math. Comput. 293 (2017) 81–95.
[35] M. Paripour, M. Ferrara, M. Salimi, Approximate solutions by artificial neural network of hybrid fuzzy differential equations, Adv. Mech. Eng. 9 (9) (2017), 1687814017717429.
[36] P. Prakash, J.J. Nieto, J. Kim, R. Rodríguez-LÓPEZ, Existence of solutions of fuzzy neutral differential equations in banach spaces, Dynam. Syst. Appl. 14 (3–4) (2005) 407–418.
[37] M.L. Puri, D.A. Ralescu, Differentials of fuzzy functions, J. Math. Anal. Appl. 91 (2) (1983) 552–558.
[38] S. Salahshour, A. Ahmadian, C.S. Chan, Successive approximation method for caputo q-fractional ivps, Commun. Nonlinear Sci. Numer. Simul. 24 (1–3) (2015) 153–158.
[39] G. Schaub, T. Turek, Energy Balance of the Earth, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 53–78.
[40] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[41] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[42] A. Seifi, T. Lotfi, T. Allahviranloo, A new efficient method using fibonacci polynomials for solving of first-order fuzzy fredholm–volterra integro-differential equations, Soft. Comput. 23 (19) (2019) 9777–9791.
[43] S. Song, C. Wu, Existence and uniqueness of solutions to cauchy problem of fuzzy differential equations, Fuzzy Sets Syst. 110 (1) (2000) 55–67.
[44] L. Stefanini, L. Sorini, M.L. Guerra, Parametric representation of fuzzy numbers and application to fuzzy calculus, Fuzzy Sets Syst. 157 (18) (2006) 2423–2455.
[45] V. Vapnik. The nature of statistical learning theory, springer science. & business media, 2013.
[46] B. Widrow. Adaptive filters i. fundamentals (tr 6764–6), 1966.
[47] B. Widrow, J.R. Glover, J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, J.E. Dong, R.C. Goodlin, Adaptive noise cancelling: principles and applications, Proc. IEEE 63 (12) (1975) 1692–1716.
[48] J. Xu, Z. Liao, J.J. Nieto, A class of linear differential dynamical systems with fuzzy matrices, J. Math. Anal. Appl. 368 (1) (2010) 54–68.
[49] H.S. Yazdi, H. Modaghegh, M. Pakdaman, Ordinary differential equations solution in kernel space, Neural Comput. Appl. 21 (1) (2012) 79–85.
[50] H.S. Yazdi, M. Pakdaman, S. Effati, Fuzzy circuit analysis, Int. J. Appl. Eng. Res. 3 (8) (2008) 1061–1072.
[51] H.S. Yazdi, M. Pakdaman, H. Modaghegh, Unsupervised kernel least mean square algorithm for solving ordinary differential equations, Neurocomputing 74 (12–13) (2011) 2062–2071.
[52] H.S. Yazdi, R. Pourreza, Unsupervised adaptive neural-fuzzy inference system for solving differential equations, Appl. Soft Comput. 10 (1) (2010) 267–275.
[53] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (3) (1965) 338–353.
[54] L.A. Zadeh, Toward a generalized theory of uncertainty (gtu)—-an outline, Inform. Sci. 172 (1–2) (2005) 1–40.