Numerical Solution of Fuzzy Differential Equations with Z-numbers Using Bernstein Neural Networks

Raheleh Jafari[1], Wen Yu[1*], Xiaoou Li [2], Sina Razvarz[1]

[1]*Departamento de Control Automático*
CINVESTAV-IPN (National Polytechnic Institute)
Mexico City, Mexico
E-mail: jafari3339@yahoo.com; yuw@ctrl.cinvestav.mx; srazvarz@yahoo.com

[2]*Departamento de Computación*
CINVESTAV-IPN (National Polytechnic Institute)
Mexico City, Mexico
E-mail: lixo@cs.cinvestav.mx

Abstract

The uncertain nonlinear systems can be modeled with fuzzy equations or fuzzy differential equations (FDEs) by incorporating the fuzzy set theory. The solutions of them are applied to analyze many engineering problems. However, it is very difficult to obtain solutions of FDEs.

In this paper, the solutions of FDEs are approximated by two types of Bernstein neural networks. Here, the uncertainties are in the sense of Z- numbers. Initially the FDE is transformed into four ordinary differential equations (ODEs) with Hukuhara differentiability. Then neural models are constructed with the structure of ODEs. With modified back propagation method for Z- number variables, the neural networks are trained. The theory analysis and simulation results show that these new models, Bernstein neural networks, are effective to estimate the solutions of FDEs based on Z-numbers.

Keywords: Fuzzy differential equations; Bernstein neural networks; Z- numbers; Uncertain nonlinear systems.

1. Introduction

Since the uncertainty in parameters can be transformed into fuzzy set theory [1], fuzzy set and fuzzy system theory are good tools to deal with uncertainty systems. Fuzzy models are applied for a large class of uncertainty nonlinear systems, for example Takagi-Sugeno fuzzy model [2]. When the parameter of an equation are changeable in the manner of fuzzy set, this equation becomes a fuzzy equation [3]. When the parameters or the states of the differential equations are uncertain, they can be modeled with FDE.

Many FDEs use fuzzy numbers as the coefficients of the differential equations to describe the uncertainties

[4]. The applications of these FDEs are connection with nonlinear modeling and control [5-8]. Another type of FED uses fuzzy variables to express the uncertainties. The study on the solutions of FDEs are applied into chaotic analysis, quantum system and many engineering problems, such as civil engineering and modeling actuators. The basic idea of fuzzy derivative was first introduced in [9]. Then it is extended in [10]. The linear first-order equation is the most generalized FDE. By generalizing the differentiability, [11] gives an analytical solution. In [12], the first order FDE with periodic boundary conditions is analyzed. Then higher order linear FDE are studied. In [13], the analytical solutions of second order FDE are obtained. The

---

analytical solutions of third order linear FDE are found in [14].

Too much complexity is involved in solving nonlinear FDE. By interval-valued method, [15] examines the basis solutions nonlinear FDEs with generalized differentiability. [16] uses periodic boundary and Hukuhara differentiability to the impulsive FDE. [17] suggests some suitable criterion to fuzzify the crisp solutions. [18] uses two-point fuzzy boundary value for FDE. However, all of above analytical methods for the solutions of FDEs are very difficult, especially for nonlinear FDEs.

Numerical solutions of FDEs have been discussed by many scientists recently. The numerical solutions of first-order FDE is proposed in [19] with an iterative technique. [20] uses Laplace transform for second-order FDE. By extending classical fuzzy set theory, [21] obtains numerical solution of an FDE. The predictor-corrector approach is applied in [22]. Euler numerical technique is used in [23] to solve FDE. Some other numerical techniques, such as Nystrom approach [24], Taylor method [25] and Runge-Kutta approach [26] can also be applied to solve FDEs. However the approximation accuracy of these numerical calculations are normally less.

The solution of FDE is uniformly continuous and inside compact sets [27]. Neural networks can give a good estimation for the solutions of FDEs. [28] shows that the solution of ODE can be approximated by neural network. [29] discusses differences between the exact solution and approximation solutions of ODEs. [30] applies dynamics neural networks to approximate first-order ODE. There are few works on FDE. [31] suggests a static neural network to solve FDE. Since the structure of the neural network is not suitable for FDE, the approximation accuracy is poor.

The decisions are carried out based on knowledge. In order to make the decision fruitful, the knowledge acquired must be credible. Z-numbers connect to the reliability of knowledge [32]. Many fields related to the analysis of the decisions are actually use the ideas of Z-numbers. Z -numbers are much less complex to calculate compared with nonlinear system modeling methods. The Z-number is abundantly adequate number compared with the fuzzy number. Although Z-numbers are implemented in many literatures, from theoretical point of view this approach is not certified completely.

The Z-number is a novel idea that is subjected to a higher potential to illustrate the information of the human being and to use in information processing [32]. Z-numbers can be regarded as to answer questions and carry out the decisions [33]. There are few structure based on the theoretical concept of Z-numbers [34]. [35] proposes a theorem to transfer the Z-numbers to the usual fuzzy sets.

In this paper, a new model named Bernstein neural network is used, which has good properties of Bernstein polynomial for FDE based on Z-number. The Bernstein polynomial has good uniform approximation ability for continuous functions [36]. Also it has innumerable drawing properties, homogeneous shape-sustaining approximation, bona fide estimation and low boundary bias. A very important property of the Bernstein polynomial is that it generates a smooth estimation for equal distance knots [37]. This property is suitable for FDE approximation. For more details regarding Bernstein neural networks, refer to [38, 39].

Two types of neural networks are used namely static and dynamic models, to approximate the solutions of FDEs based on z -numbers. These numerical methods use generalized differentiability of FDEs. The solutions of FDE is substituted into four ODEs. Then the corresponding Bernstein neural networks are applied. Finally, some real examples are used to show the effectiveness of the proposed approximation methods with the Bernstein neural networks.

2. Fuzzy differential equation for uncertain nonlinear system modeling

Consider the following controlled unknown nonlinear system
$$\dot{x} = f_1(x_1, u, t) \tag{1}$$
where $f_1(x_1, u, \mathrm{t})$ is unknown vector function, $x_1 \in \mathfrak{R}^n$ is an internal state vector and $u \in \mathfrak{R}^m$ is the input vector.

In this paper, the following differential equation (FDE) is used to model the uncertain nonlinear system (1),
$$\frac{d}{dt}x = f(t, x) \tag{2}$$
where $x \in \mathfrak{R}^n$ is the Z-number variable, which corresponds to the state x_1 in (1) , $f(t, x)$ is a Z-number vector function, which relates to $f_1(x_1, u)$, $\frac{d}{dt}x$ is the

derivative associated to the Z-number variable. Here the uncertainties of the nonlinear system (1) are in the sense of Z-numbers.

In order to use FDE based on Z- numbers, initially some concepts of fuzzy variables and Z-numbers.

Definition 1(fuzzy variable) If x is:

1) normal, there exists $\zeta_0 \in \mathsf{R}$ in such a manner $x(\zeta_0) = 1$,

2) convex, $x(\lambda\zeta + (1-\lambda)\zeta) \geq \min\{x(\zeta), x(\xi)\}$, $\forall \zeta, \xi \in \mathsf{R}, \forall \lambda \in [0,1]$,

3) upper semi-continuous on R , $x(\zeta) \leq x(\zeta_0) + \varepsilon$, $\forall \zeta \in N(\zeta_0), \forall \zeta_0 \in \mathsf{R}, \forall \varepsilon > 0, N(\zeta_0)$ is a neighborhood,

4) $x^+ = \{\zeta \in \mathbb{R}, \ x(\zeta) \in E | \mathbb{R} \to [0,1]\}$ is compact, then x is a fuzzy variable.

The fuzzy variable x can be also represented as [40]

$$x = A(\underline{x}, \bar{x}) \tag{3}$$

where $\underline{x}$ is the lower-bound variable, $\bar{x}$ is the upper-bound variable and A is a continuous function.

Definition 2 (Z- number) A Z-number has two components $Z = [x(\zeta), p]$. The primary component $x(\zeta)$ is termed as a restriction on a real-valued uncertain variable ζ . The secondary component p is a measure of reliability of x. p can be reliability, strength of belief, probability or possibility. When $x(\zeta)$ is a fuzzy number and p is the probability distribution of ζ, the Z-number is defined as Z^+-number. When both $x(\zeta)$ and p are fuzzy numbers, the Z-number is defined as Z^--number.

The Z^+ -number carries more information than the Z^--number. In this paper, the definition of Z^+-number is used, i.e., $Z = [x, p]$ x is a fuzzy number and p is a probability distribution.

We use so called membership functions to express the fuzzy number. The most popular membership functions are the triangular function

$$\mu_x(\zeta) = F(a,b,c) = \begin{cases} \frac{\zeta-a}{b-a} & a \leq \zeta \leq b \\ \frac{c-\zeta}{c-b} & b \leq \zeta \leq c \end{cases} \text{otherwise } \mu_x = 0 \tag{4}$$

and trapezoidal function

$$\mu_x(\zeta) = F(a,b,c,d) = \begin{cases} \frac{\zeta-a}{b-a} & a \leq \zeta \leq b \\ \frac{d-\zeta}{d-c} & c \leq \zeta \leq d \\ 1 & b \leq \zeta \leq c \end{cases} \text{otherwise } \mu_x = 0 \tag{5}$$

The probability measure is expressed as

$$P = \int_R \mu_x(\zeta) p(\zeta) d\zeta \tag{6}$$

where p is the probability density of ζ and R is the restriction on p. For discrete Z-numbers

$$P(x) = \sum_{i=1}^{n} \mu_x(\zeta_i) p(\zeta_i) \tag{7}$$

Definition 3 (α- level of fuzzy number) The fuzzy number x in association to the α -level is illustrated as

$$[x]^\alpha = \{a \in \mathsf{R} : x(a) \geq \alpha\} \tag{8}$$

where $0 \leq \alpha \leq 1$, $x \in E$.

Therefore $[x]^0 = x^+ = \{\zeta \in \mathsf{R}, x(\zeta) > 0\}$. Since $\alpha \in [0,1]$, $[x]^\alpha$ is a bounded mentioned as $\underline{x}^\alpha \leq [x]^\alpha \leq \bar{x}^\alpha$. The α -level of x in midst of $\underline{x}^\alpha$ and $\bar{x}^\alpha$ is explained as

$$[x]^\alpha = A(\underline{x}^\alpha, \bar{x}^\alpha) \tag{9}$$

$\underline{x}^\alpha$ and $\bar{x}^\alpha$ signify the function of α, and $\underline{x}^\alpha = d_M(\alpha), \ \bar{x}^\alpha = d_U(\alpha)$, $\alpha \in [0,1]$.

Definition 4 (α- level of Z-numbers) The α -level of the Z-number $Z = (x, P)$ is demonstrated as

$$[Z]^\alpha = ([x]^\alpha, [p]^\alpha) \tag{10}$$

where $0 \leq \alpha \leq 1$. $[p]^\alpha$ is calculated by the Nguyen's theorem

$$[p]^\alpha = p([x]^\alpha) = p([\underline{x}^\alpha, \bar{x}^\alpha]) = \left[\underline{P}^\alpha, \bar{P}^\alpha\right]$$

where $p([x]^\alpha) = \{p(\zeta) | \zeta \in [x]^\alpha\}$. So $[Z]^\alpha$ can be expressed as the form α -level of a fuzzy number

$$[Z]^\alpha = (\underline{Z}^\alpha, \bar{Z}^\alpha) = ((\underline{x}^\alpha, \underline{P}^\alpha), (\bar{x}^\alpha, \bar{P}^\alpha)) \tag{11}$$

where $\underline{P}^\alpha = \underline{x}^\alpha p(\underline{\zeta_i}^\alpha)$, $\bar{P}^\alpha = \bar{x}^\alpha p(\bar{\zeta_i}^\alpha)$, $[\zeta_i]^\alpha = (\underline{\zeta_i}^\alpha, \bar{\zeta_i}^\alpha)$.

Similar with the fuzzy numbers [5], the Z-numbers are also incorporated with three primary operations: $\oplus$, $\ominus$ and $\odot$. These operations are exhibited by: sum, subtract, multiply, and division. The operations in this paper are different definitions with [1]. The α-level of Z-numbers is applied to simplify the operations.

Let us consider $Z_1 = (x_1, p_1)$ and $Z_2 = (x_2, p_2)$ be two discrete Z-numbers illustrating the uncertain variables ζ_1 and ζ_2, $\sum_{k=1}^{n} p_1(\zeta_{1k}) = 1$, $\sum_{k=1}^{n} p_2(\zeta_{2k}) = 1$. The operations are defined

$$Z_{12} = Z_1 * Z_2 = (x_1 * x_2, p_1 * p_2)$$

where $* \in \{\oplus, \ominus, \odot\}$.

The operations for the fuzzy numbers are defined as [5]

$$[x_1 \oplus x_2]^\alpha = \left[\underline{x_1}^\alpha + \underline{x_2}^\alpha, \overline{x_1}^\alpha + \overline{x_2}^\alpha \right] \tag{12}$$

$$[x_1 \ominus x_2]^\alpha = \left[\underline{x_1}^\alpha - \underline{x_2}^\alpha, \overline{x_1}^\alpha - \overline{x_2}^\alpha \right]$$

$$[x_1 \odot x_2]^\alpha = \left[\underline{x_1}^\alpha \underline{x_2}^\alpha + \underline{x_1}^\alpha \underline{x_2}^\alpha - \underline{x_1}^\alpha \underline{x_2}^\alpha, \overline{x_1}^\alpha \overline{x_2}^\alpha + \overline{x_1}^\alpha \overline{x_2}^\alpha - \overline{x_1}^\alpha \overline{x_2}^\alpha \right]$$

For all $p_1 * p_2$ operations, we use convolutions for the discrete probability distributions

$$p_1 * p_2 = \sum_i p_1(\zeta_{1,i}) p_2(\zeta_{2,(n-i)}) = p_{12}(\zeta)$$

The above definitions satisfy the Hukuhara difference [41],

$$Z_1 \ominus_H Z_2 = Z_{12}$$
$$Z_1 = Z_2 \oplus Z_{12}$$

Here if $Z_1 \ominus_H Z_2$ exists, the α -level is

$$[Z_1 \ominus_H Z_2]^\alpha = \left[\underline{Z_1}^\alpha - \underline{Z_2}^\alpha, \overline{Z_1}^\alpha - \overline{Z_2}^\alpha \right]$$

Obviously, $Z_1 \ominus_H Z_1 = 0, Z_1 \ominus Z_1 \neq 0$

Also the above definitions satisfy the generalized Hukuhara difference [42]

$$Z_1 \ominus_{gH} Z_2 = Z_{12} \Leftrightarrow \begin{cases} 1) Z_1 = Z_2 \oplus Z_{12} \\ 2) Z_2 = Z_1 \oplus (-1) Z_{12} \end{cases} \tag{13}$$

It is convenient to display that 1) and 2) in combination are genuine if and only if Z_{12} is a crisp number. With respect to α -level what we got are $\left[Z_1 \ominus_{gH} Z_2 \right]^\alpha = [\min\{\underline{Z_1}^\alpha - \underline{Z_2}^\alpha, \overline{Z_1}^\alpha - \overline{Z_2}^\alpha\}, \max\{\underline{Z_1}^\alpha - \underline{Z_2}^\alpha, \overline{Z_1}^\alpha - \overline{Z_2}^\alpha\}]$ and If $Z_1 \ominus_{gH} Z_2$ and $Z_1 \ominus_H Z_2$ subsist, $Z_1 \ominus_H Z_2 = Z_1 \ominus_{gH} Z_2$. The conditions for the existence of $Z_{12} = Z_1 \ominus_{gH} Z_2 \in E$ are

$$1) \begin{cases} \underline{Z_{12}}^\alpha = \underline{Z_1}^\alpha - \underline{Z_2}^\alpha \text{ and } \overline{Z_{12}}^\alpha = \overline{Z_1}^\alpha - \overline{Z_2}^\alpha \\ \text{with } \underline{Z_{12}}^\alpha \text{ increasing, } \overline{Z_{12}}^\alpha \text{ decreasing, } \underline{Z_{12}}^\alpha \leq \overline{Z_{12}}^\alpha \end{cases} \tag{14}$$

$$2) \begin{cases} \underline{Z_{12}}^\alpha = \overline{Z_1}^\alpha - \overline{Z_2}^\alpha \text{ and } \overline{Z_{12}}^\alpha = \underline{Z_1}^\alpha - \underline{Z_2}^\alpha \\ \text{with } \underline{Z_{12}}^\alpha \text{ increasing, } \overline{Z_{12}}^\alpha \text{ decreasing, } \underline{Z_{12}}^\alpha \leq \overline{Z_{12}}^\alpha \end{cases}$$

where $\forall \alpha \in [0,1]$

If x is a triangular function, the absolute value of the Z-number $Z = (x, p)$ is

$$\left| Z(\zeta) \right| = (|a_1| + |b_1| + |c_1|, p(|a_2| + |b_2| + |c_2|)) \tag{15}$$

If x_1 and x_2 are triangular functions, the supremum metric for Z-numbers $Z_1 = (x_1, p_1)$ and $Z_2 = (x_2, p_2)$ is given as

$$D(Z_1, Z_2) = d(x_1, x_2) + d(p_1, p_2)$$

in this case $d(\cdot, \cdot)$ is the supremum metrics considering fuzzy sets [5]. $D(Z_1, Z_2)$ is incorporated with the following possessions:

$$D(Z_1 + Z, Z_2 + Z) = D(Z_1, Z_2)$$
$$D(Z_2, Z_1) = D(Z_1, Z_2)$$
$$D(kZ_1, kZ_2) = |k| D(Z_1, Z_2)$$
$$D(Z_1, Z_2) \leq D(Z_1, Z) + D(Z, Z_2)$$

where $k \in \mathsf{R}$, $Z = (x, p)$ is Z-number and x is triangle function, for proof refer to [43].

Definition 5 (α-level of Z-number valued function) Let $\tilde{Z}$ denotes the space of Z-numbers. The α −level of Z-number valued function $F: [0, a] \to \tilde{Z}$ is

$$F(x, \alpha) = \left[\underline{F}(x, \alpha), \overline{F}(x, \alpha) \right]$$

where $\in \tilde{Z}$, for each $\alpha \in [0,1]$.

With the definition of Generalized Hukuhara difference, the gH-derivative of F at x_0 is expressed as

$$\frac{d}{dt} F(x_0) = \lim_{h \to 0} \frac{1}{h} [F(x_0 + h) \ominus_{gH} F(x_0)] \tag{16}$$

In (16), $F(x_0 + h)$ and $F(x_0)$ exhibits similar style with Z_1 and Z_2 respectively included in (13).

If we apply the α − level (10) to $f(t, x)$ in (2), then we obtain two Z-number valued functions: $\underline{f}[t, \underline{x}(\zeta, \alpha), \overline{x}(\zeta, \alpha)]$ and $\overline{f}[t, \underline{x}(\zeta, \alpha), \overline{x}(\zeta, \alpha)]$.

The fuzzy differential equation (2) can be equivalent to the following four ODE

$$\begin{aligned} 1) & \begin{cases} \frac{d}{dt} \underline{x} = \underline{f}[t, \underline{x}(\zeta, \alpha), \overline{x}(\zeta, \alpha)] \\ \frac{d}{dt} \overline{x} = \overline{f}[t, \underline{x}(\zeta, \alpha), \overline{x}(\zeta, \alpha)] \end{cases} \\ 2) & \begin{cases} \frac{d}{dt} \underline{x} = \overline{f}[t, \underline{x}(\zeta, \alpha), \overline{x}(\zeta, \alpha)] \\ \frac{d}{dt} \overline{x} = \underline{f}[t, \underline{x}(\zeta, \alpha), \overline{x}(\zeta, \alpha)] \end{cases} \end{aligned} \tag{17}$$

The fuzzy model of (1) can be regarded as four ordinary differential equations (17).

In this paper, the FDE (2) is used to model the uncertain nonlinear system (1), such that the output of the plant x can follow the plant output x_1,

$$\min_f \|x - x_1\| \tag{18}$$

This modeling object can be considered as: finding $\overline{f}$ and $\underline{f}$ in the fuzzy equations of (17) or finding the solutions of these models. It is impossible to obtain analytical solutions [44].

In fact, the nonlinear system can be modeled by the neural network directly. However, this data-driven black box identification method does not use the model information. While the FDE use the model information

of the nonlinear system, such as the brief form of the differential equation.

The following theorems give theory support for nonlinear system modeling via FDEs based on Z-numbers.

Theorem 1 If the Z-number function f and its derivative $\frac{\partial f}{\partial x}$ are on the rectangle $[-p, p] \times [-q, q]$, here $p, q \in \tilde{Z}$, $\tilde{Z}$ is the space of Z-numbers, then there exists an unique Z-number solution for the following FDE based on Z-numbers

$$\frac{d}{dt} x = f(t, x), \quad x(t_0) = x_0 \tag{19}$$

for all $t \in (-b, b)$, $b \le p$

Proof: We utilize Picard's iteration technique [45] to develop a sequence of Z-number functions $\varphi_n(t)$ as

$$\varphi_{n+1}(t) = \varphi_0 \oplus \int_0^t f(s, \varphi_n(s)) ds =$$
$$\varphi_0 \ominus_H (-1) \int_0^t f(s, \varphi_n(s)) ds$$

We first validate that $\varphi_n(t)$ is continuous and exists for all n. Obviously, if $\varphi_n(t)$ exists then $\varphi_{n+1}(t)$ also exists as

$$\varphi_{n+1}(t) = \varphi_0 \oplus \int_0^t f(s, \varphi_n(s)) ds =$$
$$\varphi_0 \ominus_H (-1) \int_0^t f(s, \varphi_n(s)) ds$$

Since f is continuous, so there exists $N \in E$ such that $|f(t, x)| \le N$ for all $t \in [-p, p]$, as well as all $x \in [-q, q]$. If we set $t \in [-b, b]$ for $b \le \min(q/N, p)$ then it is possible

$$\| \varphi_{n+1} \ominus \varphi_0 \| = \left\| \int_0^t f(s, \varphi_n(s)) ds \right\| \le N|t| \le Nb \le q$$

This validates that $\varphi_{n+1}(t)$ acquires values in $[-q, q]$. Because

$$\varphi_n(t) = \sum_{k=1}^n (\varphi_k(t) \ominus \varphi_{k-1}(t))$$

for any $\gamma < 1$, we select $t \in (-b, b)$ such that $|\varphi_k(t) \ominus \varphi_{k-1}(t)| \le \gamma^k$ for all k. This signifies that there exists $\gamma < 1$ [46]

$$|\varphi_k(t) \ominus \varphi_{k-1}(t)| \le \gamma^k$$

From the mean value theorem [47],

$$\varphi_k(t) \ominus \varphi_{k-1}(t) =$$
$$\int_0^t [f(s, \varphi_{k-1}(s)) \ominus f(s, \varphi_{k-2}(s))] ds$$

Applying the mean value theorem into the Z-number function $h(x) = f(s, x)$ in the two points $\varphi_{k-1}(s)$ and, $\varphi_{k-2}(s)$,

$$h(\varphi_{k-1}(s)) \ominus h(\varphi_{k-2}(s)) = h'(\psi_k(s)) (\varphi_{k-1}(s)) \ominus \varphi_{k-2}(s))$$

Taking into consideration $h'(x) = \frac{\partial f}{\partial x}$, we obtain

$$\varphi_k(t) \varphi_{k-1}(t) = \int_0^t \frac{\partial f}{\partial x}(s, \psi_k(s))(\varphi_{k-1}(s) \ominus \varphi_{k-2}(s)) ds \tag{20}$$

Because $|\varphi_{k-1}(s) - \varphi_{k-2}(s)| \le \gamma^{k-1}$ for $s \le t$ and $b < \gamma / N$, by substituting the above relation in (20) and utilizing the boundess of $\frac{\partial f}{\partial x}$,

$$|\varphi_k(t)) \ominus \varphi_{k-1}(t)| \le \int_0^t N \gamma^{k-1} ds = Nt \gamma^{k-1} \le Nb \gamma^{k-1}$$

In order to validate that x is continuous, it is necessary to show that for any given $\varepsilon > 0$ there exists $\delta > 0$ in such a manner that $|t_2 - t_1| < \delta$ implies $|\varphi(t_2) - \varphi(t_1)| < \varepsilon$. At par with the notation convenience, we suppose that $t_1 < t_2$. It follows that

$$\varphi(t_2) - \varphi(t_1) = \lim_{n \to \infty} \varphi_n(t_2) - \lim_{n \to \infty} \varphi_n(t_1)$$
$$= \lim_{n \to \infty} (\varphi_n(t_2) - \varphi_n(t_1)) = \lim_{n \to \infty} \int_{t_1}^{t_2} f(s, \varphi_n(s)) ds$$

There exists N in such a manner that $|f(s, x)| \le N$. Hence

$$|\varphi(t_2) - \varphi(t_1)| \le \int_{t_1}^{t_2} N ds = N |t_2 - t_1| \le N\delta$$

henceforth by selecting $\delta < \varepsilon / N$ it is observed that $|\varphi(t_2) - \varphi(t_1)| < \varepsilon$. So $\lim_{n \to \infty} \varphi_n(t)$ exists for all t.

Now we demonstrate that $\lim_{n \to \infty} \varphi_n(t)$ is continuous. Since

$$\varphi(t) = \lim_{n \to \infty} \varphi_n(t) = \lim_{n \to \infty} \int_0^t f(s, \varphi_{n-1}(s)) ds$$
$$= \int_0^t \lim_{n \to \infty} f(s, \varphi_{n-1}(s)) ds = \int_0^t f(s, \lim_{n \to \infty} \varphi_{n-1}(s)) ds$$

where the last step (moving the limit inside the function) is at par with the concept that f is continuous in each variable. Hence it is clear that

$$\varphi(t) = \int_0^t f(s, \varphi(s)) ds$$

because all functions are continuous,

$$\frac{d}{dt} \varphi = f(s, \varphi(t))$$

If there exists another solution $\phi(t)$,

$$\varphi(t) - \phi(t) = \int_0^t (f(s, \varphi(t)) - f(s, \phi(t))) ds$$

Since the two functions are different, there exists $\varepsilon > 0$, $|\varphi(t) - \phi(t)| > \varepsilon$. We define

$$m = \max_{0 \le t \le b} |\varphi(t) - \phi(t)|$$

N is the bound for $\frac{\partial f}{\partial x}$. Utilizing the mean value theorem,

$$|\varphi(t) - \phi(t)| \le \int_0^t N |\varphi(t) - \phi(t)| ds \le N |t| m \le Nbm$$

If we select $b < \varepsilon/2mN$, it signifies that for all $t < b$, $|\varphi(t) - \phi(t)| < \varepsilon/2$, that contracts the fact that the least difference is ϵ. So there exists an unique Z-number solution. □

Theorem 2 Assume the following FDE based on Z-numbers

$$\frac{d}{dt}x = f(t,x) \tag{21}$$

here $f \in \bar{J}_{ab}$, $\bar{J}_{ab}$ is the set of linear strongly bounded operators, for every operators f there exists a function such that $|f(t,x)| \leq \tau(t)\|v\|_G$, $t \in [a,b]$ and and there exist $f_0, f_1 \in \varphi_{ab}$, φ_{ab} is a set of linear operators $f \in \bar{J}_{ab}$ from the set   to the set   such that

$$|\underline{f}(t,\underline{x},\bar{x}) + \underline{f}_1(t,\underline{x},\bar{x})| \leq \underline{f}_0(t,|x|,|\bar{x}|), \quad t \in [a,b] \tag{22}$$
$$|\bar{f}(t,\underline{x},\bar{x}) + \bar{f}_1(t,\underline{x},\bar{x})| \leq \bar{f}_0(t,|x|,|\bar{x}|), \quad t \in [a,b]$$

then (21) has an unique solution.

Proof. If x is a Z-number solution of (21) and $-\frac{1}{2}f_1 \in J_{ab}(a)$,

$$\frac{d}{dt}\beta = -\frac{1}{2}f_1(t,\beta) \oplus f_0(t,|x|) \oplus \frac{1}{2}f_1(t,|x|) \tag{23}$$

contains a unique Z-number solution β. Moreover as $f_0, f_1 \in \varphi_{ab}$

$$\underline{\beta}(t) \geq 0, \quad t \in [a,b] \tag{24}$$
$$\bar{\beta}(t) \geq 0, \quad t \in [a,b]$$

According to (22) and the condition $f_1 \in \varphi_{ab}$, from (23) we have

$$\frac{d}{dt}\underline{\beta} \geq -\frac{1}{2}\underline{f}_1(t,\underline{\beta},\bar{\beta}) + \underline{f}(t,\underline{x},\bar{x}) + \frac{1}{2}\underline{f}_1(t,\underline{x},\bar{x})$$
$$\frac{d}{dt}\bar{\beta} \geq -\frac{1}{2}\bar{f}_1(t,\underline{\beta},\bar{\beta}) + \bar{f}(t,\underline{x},\bar{x}) + \frac{1}{2}\bar{f}_1(t,\underline{x},\bar{x})$$

thus $t \in [a,b]$

$$\frac{d}{dt}(-\underline{\beta}) \leq -\frac{1}{2}\underline{f}_1(t,-\underline{\beta},-\bar{\beta}) + \underline{k}(t,\underline{x},\bar{x}) + \frac{1}{2}\underline{k}_1(t,\underline{x},\bar{x})$$
$$\frac{d}{dt}(-\bar{\beta}) \leq -\frac{1}{2}\bar{f}_1(t,-\underline{\beta},-\bar{\beta}) + \bar{f}(t,\underline{x},\bar{x}) + \frac{1}{2}\bar{f}_1(t,\underline{x},\bar{x})$$

The last two inequalities is on account of the presumption $-\frac{1}{2}f_1 \in J_{ab}(a)$

$$|\underline{x}(t)| \leq \underline{\beta}(t) \quad t \in [a,b] \tag{25}$$
$$|\bar{x}(t)| \leq \bar{\beta}(t) \quad t \in [a,b]$$

According to the (25) and the conditions $f_0, f_1 \in \varphi_{ab}$, (23) results in

$$\frac{d}{dt}\underline{\beta} \leq \underline{f}_0(t,\underline{\beta},\bar{\beta}), \quad t \in [a,b]$$
$$\frac{d}{dt}\bar{\beta} \leq \bar{f}_0(t,\underline{\beta},\bar{\beta}), \quad t \in [a,b]$$

As $f_0 \in J_{ab}(a)$, the last inequality with $\beta(a) = 0$ yields $\underline{\beta}(t) \leq 0$ and $\bar{\beta}(t) \leq 0$ for $t \in [a,b]$. (24) implies $\beta \equiv 0$. Thus based on (25) we have $x \equiv 0$. □

3. Solving fuzzy differential equation with neural networks

In general, it is difficult to solve the four equations (17) or (2). In this paper, a special neural network named Bernstein neural network is used to approximate the solutions of the FDE (2).

The Bernstein neural network use the following Bernstein polynomial,

$$B(x_1,x_2) = \sum_{i=0}^{N}\sum_{j=0}^{M}\binom{N}{i}\binom{M}{j}$$
$$W_{i,j}x_{1i}(T-x_{1i})^{N-i}x_{2j}(1-x_{2j})^{M-j} \tag{26}$$

where $\binom{N}{i} = \frac{N!}{i!(N-i)!}$, $\binom{M}{j} = \frac{M!}{j!(M-j)!}$, $W_{i,j}$ is the Z-number coefficient.

This two variable polynomial can be regarded as a neural network, which has two inputs x_{1i} and x_{2j} and one output y,

$$y = \sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j W_{i,j}x_{1i}(T-x_{1i})^{N-i}x_{2j}(1-x_{2j})^{M-j} \tag{27}$$

where $\lambda_i = \binom{N}{i}$, $\gamma_j = \binom{M}{j}$.

Because the Bernstein neural network (27) has similar forms as (17), the Bernstein neural network (27) is used to approximate the solutions of four ODEs in (17), see Figure 1.
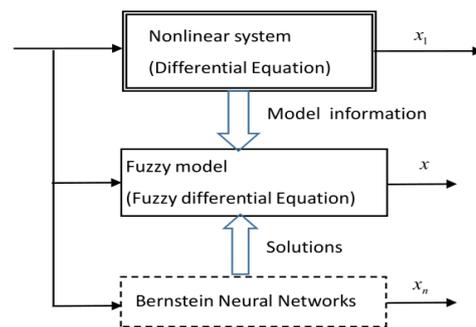

Fig. 1. Nonlinear system modeling with fuzzy differential equation

If x_1 and x_2 in (26) are defined as: x_1 is time interval t, x_2 is the α-level, the solution of (2) in the form of the Bernstein neural network is

$$x_m(t,\alpha) = x_m(0,\alpha)$$
$$\oplus t \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j W_{i,j} t_i (T-t_i)^{N-i} \alpha_j (1-\alpha_j)^{M-j} \quad (28)$$

where $x_m(0,\alpha)$ is the initial condition of the solution based on Z-number.

So the derivative of (27) is

$$1) \begin{cases} \frac{d}{dt}\underline{x}_m = C_1 + C_2 \\ \frac{d}{dt}\overline{x}_m = D_1 + D_2 \end{cases}$$
$$2) \begin{cases} \frac{d}{dt}\underline{x}_m = C_1 + C_2 \\ \frac{d}{dt}\overline{x}_m = D_1 + D_2 \end{cases} \quad (29)$$

where $t \in [0,T]$, $\alpha \in [0,1]$, $t_k = kh$, $h = \frac{T}{k}$, $k = 1,...,N$, $\alpha_j = jh_1$, $h_1 = \frac{1}{M}$, $j = 1,...,M$,

$C_1 = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} t_i (T-t_i)^{N-i} \alpha_j (1-\alpha_j)^{M-j}$

$D_1 = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} t_i (T-t_i)^{N-i} \alpha_j (1-\alpha_j)^{M-j}$

$C_2 = t_k \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} [it_{i-1,j}(T-t_i)^{N-i}$
$- (N-i)t_{i,j}(T-t_i)^{N-i-1}]\alpha_j^i (1-\alpha_j)^{M-j}$

$D_2 = t_k \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} [it_{i-1,j}(T-t_i)^{N-i}$
$- (N-i)t_{i,j}(T-t_i)^{N-i-1}]\alpha_j^i (1-\alpha_j)^{M-j}$

The above equations can be regarded as the neural network form, see Figure 2.
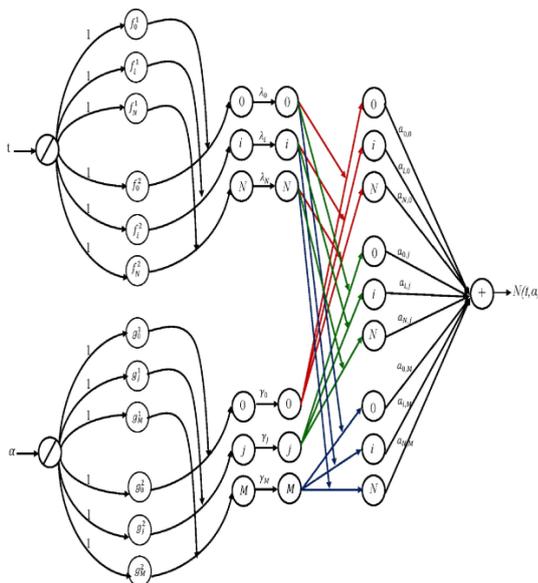

Fig. 2. Static Bernstein neural network

- input unit:
$$o_1^1 = t, \quad o_2^1 = \alpha$$

- the first hidden units:
$$o_{1,i}^2 = f_i^1(o_1^1), \quad o_{2,i}^2 = f_i^2(o_1^1)$$
$$o_{3,j}^2 = g_j^1(o_2^1), \quad o_{4,j}^2 = g_j^2(o_2^1)$$

- the second hidden units:
$$o_{1,i}^3 = o_{1,i}^2(o_{2,i}^2), \quad o_{2,j}^3 = o_{3,j}^2(o_{4,j}^2)$$

- the third hidden units:
$$o_{1,i}^4 = \lambda_i o_{1,i}^3, \quad o_{2,i'}^4 = \gamma_j o_{2,j}^3$$

- the forth hidden units:
$$o_{i,j}^5 = o_{1,i}^4 o_{2,j}^4$$

- output unit:
$$N(t,\alpha) = \sum_{i=0}^{N} \sum_{j=0}^{M} (a_{i,j} o_{i,j}^5)$$

where $f_i^1 = t^i$, $f_i^2 = (T-t)^{N-i}$, $\lambda_i = \frac{1}{T^N}\binom{N}{i}$,

$g_j^1 = \alpha^j$, $g_j^2 = (1-\alpha)^{M-j}$, $\gamma_j = \binom{M}{j}$.

The training errors between (29) and (17) are defined as

$$1) \begin{cases} \underline{e}_1 = C_1 + C_2 - \underline{f} \\ \overline{e}_1 = D_1 + D_2 - \overline{f} \end{cases}$$
$$2) \begin{cases} \underline{e}_2 = C_1 + C_2 - \overline{f} \\ \overline{e}_2 = D_1 + D_2 - \underline{f} \end{cases} \quad (30)$$

The standard back-propagation learning algorithm is utilized to update the weights with the above training errors

$$\underline{W}_{i,j}(k+1) = \underline{W}_{i,j}(k) - \eta_1 \left(\frac{\partial \underline{e}_1^2}{\partial \underline{W}_{i,j}} + \frac{\partial \overline{e}_1^2}{\partial \underline{W}_{i,j}} \right)$$
$$\overline{W}_{i,j}(k+1) = \overline{W}_{i,j}(k) - \eta_2 \left(\frac{\partial \underline{e}_2^2}{\partial \overline{W}_{i,j}} + \frac{\partial \overline{e}_2^2}{\partial \overline{W}_{i,j}} \right) \quad (31)$$

where η_1 and η_2 are positive learning rates.

The momentum terms, $\gamma \Delta \underline{W}_{i,j}(k-1)$ and $\gamma \Delta \overline{W}_{i,j}(k-1)$ can be added to stabilized the training process. The above Bernstein neural network can be converted to a recurrent (dynamic) form, see Figure 3. The dynamic Bernstein neural network is

$$\begin{cases} \frac{d}{dt}\underline{x}_m(t,\alpha) = \underline{P}(t,\alpha)A(\underline{x}_m(t,\alpha),\overline{x}_m(t,\alpha)) + \underline{Q}(t,\alpha) \\ \frac{d}{dt}\overline{x}_m(t,\alpha) = \overline{P}(t,\alpha)A(\underline{x}_m(t,\alpha),\overline{x}_m(t,\alpha)) + \overline{Q}(t,\alpha) \end{cases} \quad (32)$$

Obviously this dynamic network has the form of

$$f(t,x) = P(t)x + Q(t)$$

and it is closed to (2).

Table 1. Solutions of different methods based on Z-numbers.

α	Exact solution	SNN	DNN
0	[(2.1858,3.2787),p(0.8,0.87,0.95)]	[(2.2967,3.4240),p(0.7,0.81,0.85)]	[(2.2250,3.3883),p(0.71,0.85,0.87)]
0.2	[(2.2924,3.1521),p(0.81,0.9,1)]	[(2.3545,3.2570),p(0.7,0.82,0.9)]	[(2.3504,3.2467),p(0.75,0.83,0.9)]
0.6	[(2.5790,3.0088),p(0.81,0.9,1)]	[(2.6759,3.1461),p(0.7,0.8,0.87)]	[(2.6097,3.0872),p(0.75,0.83,0.9)]
1	[(2.9144,2.9144),p(0.8,0.87,0.95)]	[(2.9667,2.9667),p(0.7,0.8,0.87)]	[(2.9532,2.9532),p(0.71,0.85,0.87)]
α	Exact solution	Max-Min Euler	Average Euler
0	[(2.1858,3.2787),p(0.8,0.87,0.95)]	[(2.4847,3.4771),p(0.7,0.82,0.85)]	[(2.9921,3.4921),p(0.65,0.8,0.85)]
0.2	[(2.2924,3.1521),p(0.81,0.9,1)]	[(2.6100,3.5888),p(0.72,0.8,0.87)]	[(2.8137,3.2303),p(0.6,0.7,0.75)]
0.6	[(2.5790,3.0088),p(0.81,0.9,1)]	[(2.7137,3.1660),p(0.6,0.8,0.87)]	[(2.9565,3.1372),p(0.6,0.7,0.8)]
1	[(2.9144,2.9144),p(0.8,0.87,0.95)]	[(3.0152,3.0152),p(0.6,0.8,0.87)]	[(3.1249,3.1249),p(0.6,0.7,0.8)]

The training algorithm is similar as (31), only the training errors are changed as

$$1)\begin{cases} \underline{e}_1 = C_1 + C_2 - \underline{P}A(\underline{x}_m,\overline{x}_m) - \underline{Q} \\ \overline{e}_1 = D_1 + D_2 - \overline{P}A(\underline{x}_m,\overline{x}_m) - \overline{\overline{Q}} \end{cases}$$

$$2)\begin{cases} \underline{e}_2 = C_1 + C_2 - \overline{P}A(\underline{x}_m,\overline{x}_m) - \overline{Q} \\ \overline{e}_2 = D_1 + D_2 - \underline{P}A(\underline{x}_m,\overline{x}_m) - \underline{Q} \end{cases} \qquad (33)$$
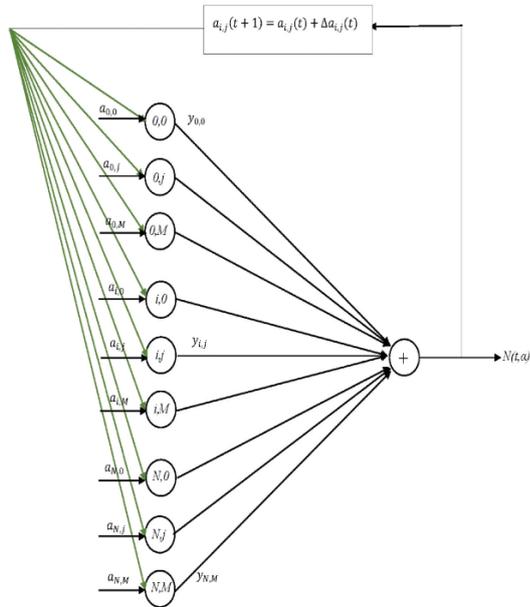

Fig. 3. Dynamic Bernstein neural network

4. Applications

In this section, several real applications are used to show how to use the Bernstein neural networks to approximate the solutions of the FDEs. These examples can be expressed by FDEs.

Example 1 The vibration mass system shown in Figure 4 can be modeled by the ODE

$$\frac{d}{dt}x(t) = \frac{k}{m}x(t) \qquad (34)$$

where the spring constant is $k = 1$. The mass m is changeable in $[(0.75,1.125), p(0.7,0.8,1)]$, so the position state $x(t)$ has some uncertainties, the ODE (34) can be formed into a FDE based on Z-number. It has the same form as (34), only $x(t)$ becomes a Z-number variable. If the initial position is $x(0) = [(0.75 + 0.25\alpha, 1.125 - 0.125\alpha), p(0.8,0.9,1)]$, $\alpha \in [0,1]$, then the exact solutions of the FDE (34) is [48]

$$x(t,\alpha) = \left[((0.75 + 0.25\alpha)e^t, (1.125 - 0.125\alpha)e^t), p(0.8,0.9,1)\right] \qquad (35)$$

where $t \in [0,1]$. Now the static Bernstein neural network (28) is used, noted as SNN to approximate the Z-number solution $[(\underline{x}_m(t,\alpha),\overline{x}_m(t,\alpha)), p(0.8,0.94,1)]$ (35) where

$$\begin{cases} \underline{x}_m(t,\alpha) = (0.75 + 0.25\alpha) \\ \quad + t\sum_{i=0}^N \sum_{j=0}^M \lambda_i \gamma_j \underline{W}_{i,j} t_i (T - t_i)^{N-i} \alpha_j (1 - \alpha_j)^{M-j} \\ \overline{x}_m(t,\alpha) = (1.125 - 0.125\alpha) \\ \quad + t\sum_{i=0}^N \sum_{j=0}^M \lambda_i \gamma_j \overline{W}_{i,j} t_i (T - t_i)^{N-i} \alpha_j (1 - \alpha_j)^{M-j} \end{cases}$$
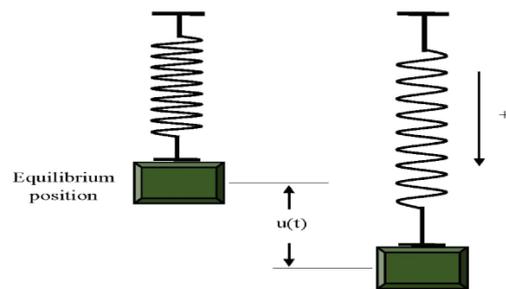

Fig. 4. Vibration mass

Also dynamic Bernstein neural network (32) is used, noted as DNN to approximate the solutions. The learning rates are $\eta = 0.01, \gamma = 0.01$. To compare the results, two other popular methods are used namely Max-Min Euler method and Average Euler method [28]. The comparison results are shown in Table 1 and Table

Table 2. Approximation errors based on Z-numbers

	SNN	DNN	Max-Min Euler	Average Euler
0	[(0.0684,0.1251),p(0.7,0.8,0.85)]	[(0.0231,0.0671),p(0.7,0.85,0.87)]	[(0.1064,0.1596),p(0.7,0.8,0.85)]	[(0.2404,0.5138),p(0.6,0.8,0.85)]
0.2	[(0.0735,0.1192),p(0.7,0.8,0.9)]	[(0.0266,0.0675),p(0.75,0.8,0.9)]	[(0.1127,0.1551),p(0.7,0.8,0.87)]	[(0.1588,0.4286),p(0.7,0.8,0.85)]
0.6	[(0.0855,0.1095),p(0.8,0.87,0.95)]	[(0.0339,0.0689),p(0.8,0.9,1)]	[(0.1253,0.1462),p(0.7,0.85,0.9)]	[(0.0082,0.2798),p(0.7,0.81,0.9)]
0.8	[(0.0833,0.0939),p(0.8,0.91,1)]	[(0.0345,0.0526),p(0.8,0.94,1)]	[(0.1247,0.1345),p(0.8,0.9,1)]	[(0.0628,0.2009),p(0.75,0.9,1)]
1	[(0.1029,0.1029),p(0.7,0.8,0.9)]	[(0.0572,0.0572),p(0.8,0.85,0.95)]	[(0.1410,0.1410),p(0.7,0.8,0.87)]	[(0.1410,0.1410),p(0.7,0.8,0.87)]

2. Corresponding solution plots are shown in Figure 5. We can see that both the Bernstein neural networks SNN and DNN (our methods), Max-Min Euler method and Average Euler method can approximate the solutions of the FDEs. The approximation errors of the DNN is much smaller than the other methods.
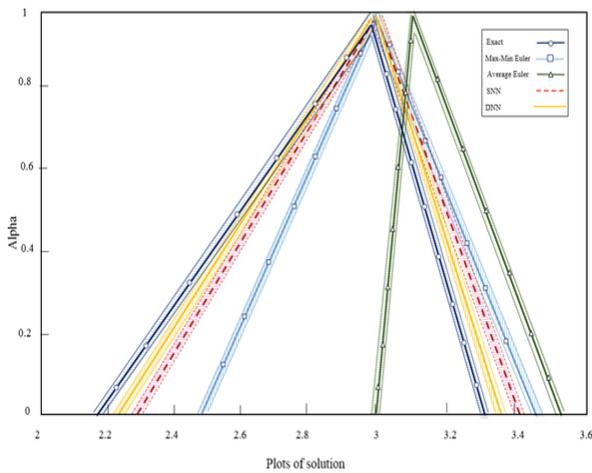

Fig. 5. Comparison plots of SNN, DNN, Max-Min Euler, Average Euler and the exact solution based on Z -numbers

The following relation is used to transfer the Z-numbers to fuzzy numbers,

$$\alpha = \frac{\int x \pi_{\tilde{P}}(x)dx}{\int \pi_{\tilde{P}}(x)dx}$$

Consider $Z = (A, p) = [(2.1858, 3.2787), p(0.8, 0.87, 0.95)],$

then $Z^{\alpha} = [2.1858, 3.2787; 0.87]$ and so $Z' = [\sqrt{0.87}\ 2.1858, \sqrt{0.87}\ 3.2787]$. The comparison results of different methods for the fuzzy numbers are shown in Table 3.

The Z-numbers increase degree of reliability of the information. The crucial factor is that incorporated information is not only the most generalized representation of information uncomplicated real world but also incorporated with greater narrative power extracted from human cognition perspective compared with fuzzy number. The comparison between the Z-number $Z = [(2.1858, 3.2787), p(0.8, 0.87, 0.95)]$ and fuzzy number [2.0387, 3.0581] is shown in Figure 6. It can be seen that the Z-number incorporates with various information and the solution of the Z-number is more accurate. The membership function for the restriction in the Z-number is $\mu_{A_z} = [2.1858, 3.2787]$. It can be in probability form.

Example 2 The heat treatment system in welding can be modeled as [49]:

$$\frac{d}{dt}x(t) = 3Ax^2(t) \tag{36}$$

where transfer area A is uncertainty as $A = [(1 + \alpha, 3 - \alpha), p(0.8, 0.87, 0.95)], \quad \alpha \in [0,1]$. So (ex3) is a FDE based on Z-number. If the initial condition is $x(0) = [(0.5\sqrt{\alpha}, 0.2\sqrt{1-\alpha} + 0.5), p(0.8, 0.92, 1)],$ the static Bernstein neural network (28) has the form of

Table 3. Solutions of different methods based on fuzzy numbers

α	Exact solution	SNN	DNN	Max-Min Euler	Average Euler
0	[2.0387,3.0581]	[1.9703,3.0043]	[1.9901,3.0305]	[1.9453,2.5980]	[2.2441,2.6191]
0.1	[2.1067,3.0241]	[2.0302,2.9415]	[2.0591,2.9752]	[2.0102,2.8855]	[2.2791,2.6166]
0.2	[2.1746,2.9901]	[2.1059,2.9131]	[2.1283,2.9399]	[2.0750,2.8531]	[2.3140,2.6140]
0.3	[2.2426,2.9561]	[2.1618,2.8707]	[2.1901,2.8931]	[2.1398,2.8207]	[2.3490,2.6115]
0.4	[2.3105,2.9222]	[2.2307,2.8453]	[2.2601,2.8799]	[2.2047,2.7883]	[2.3840,2.6090]
0.5	[2.3785,2.8882]	[2.2984,2.8088]	[2.3288,2.8337]	[2.2695,2.7559]	[2.4189,2.6064]
0.6	[2.4465,2.8542]	[2.3631,2.7784]	[2.3904,2.7955]	[2.3344,2.7234]	[2.4539,2.6039]
0.7	[2.5144,2.8202]	[2.4292,2.7449]	[2.4555,2.7691]	[2.3992,2.6910]	[2.4888,2.6013]
0.8	[2.5824,2.7862]	[2.4895,2.7067]	[2.5101,2.7302]	[2.4641,2.6586]	[2.5238,2.5988]
0.9	[2.6503,2.7523]	[2.5564,2.6769]	[2.5821,2.7001]	[2.5289,2.6262]	[2.5588,2.5963]
1	[2.7183,2.7183]	[2.6199,2.6399]	[2.6414,2.6614]	[2.5937,2.5937]	[2.5937,2.5937]

$$\begin{cases} \underline{x}_m(t,\alpha) = 0.5\sqrt{\alpha} \\ + t \sum_{i=0}^{N}\sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} t_i (T-t_i)^{N-i} \alpha_j (1-\alpha_j)^{M-j} \\ \overline{x}_m(t,\alpha) = 0.2\sqrt{1-\alpha} + 0.5 \\ + t \sum_{i=0}^{N}\sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} t_i (T-t_i)^{N-i} \alpha_j (1-\alpha_j)^{M-j} \end{cases}$$

where the approximate Z-number solution is termed as $[(\underline{x}_m(t,\alpha), \overline{x}_m(t,\alpha)), p(0.8,0.9,1)]$. With the learning rates $\eta = 0.002$ and $\gamma = 0.002$ the approximation results for Z-numbers are shown in Table 4. The results of Bernstein neural networks approximation for the fuzzy numbers are shown in Table 5.
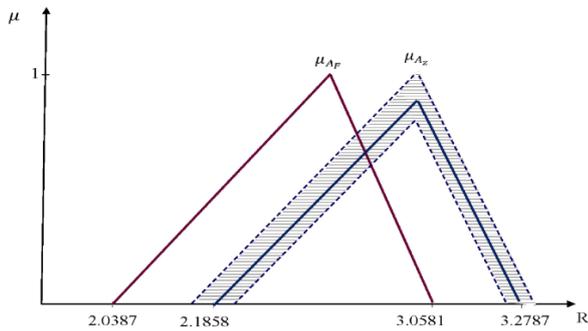

Fig. 6. Z-number and fuzzy number

Table 4. Bernstein neural networks approximate the Z-numbers

α	SNN	DNN
0	[(0.0582,0.0859),p(0.7,0.8,0.85)]	[(0.0250,0.0425),p(0.7,0.82,0.9)]
0.1	[(0.0449,0.0696),p(0.7,0.8,0.9)]	[(0.0224,0.0399),p(0.75,0.82,0.9)]
0.2	[(0.0419,0.0619),p(0.8,0.92,1)]	[(0.0207,0.0394),p(0.8,0.94,1)]
0.3	[(0.0250,0.0348),p(0.7,0.81,0.9)]	[(0.0226,0.0344),p(0.8,0.85,0.96)]
0.4	[(0.0487,0.0689),p(0.7,0.8,0.88)]	[(0.0271,0.0510),p(0.75,0.82,0.9)]
0.5	[(0.0534,0.0665),p(0.8,0.9,1)]	[(0.0160,0.0271),p(0.81,0.92,1)]
0.6	[(0.0494,0.0765),p(0.8,0.9,1)]	[(0.0201,0.0413),p(0.81,0.92,1)]
0.7	[(0.0630,0.0859),p(0.75,0.82,0.9)]	[(0.0303,0.0476),p(0.82,0.9,1)]
0.8	[(0.0393,0.0536),p(0.8,0.92,1)]	[(0.0164,0.0379),p(0.82,0.94,1)]
0.9	[(0.0422,0.0669),p(0.8,0.9,1)]	[(0.0212,0.0430),p(0.8,0.94,1)]
1	[(0.0443,0.0443),p(0.7,0.8,0.88)]	[(0.0186,0.0186),p(0.7,0.82,0.9)]

Example 3 A generalized model of a tank system is displayed in Figure 7. Assume $I = t + 1$ be inflow disturbances of the tank that will generate vibration in liquid level x, here $R = 1$ will be the flow obstruction that can be curbed using the valve and $A = 1$ is considered to be cross section of the mentioned tank. The expression in relation to the liquid level considering the tank can be described as [50]:

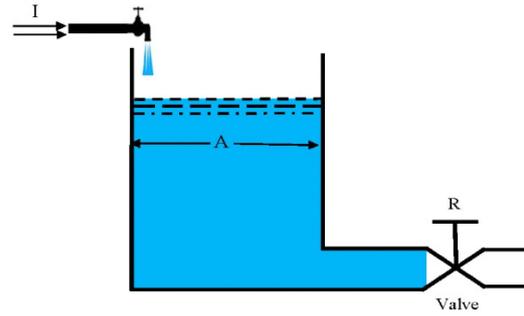$$\frac{d}{dt}x(t) = -\frac{1}{AR}x(t) + \frac{I}{A} \tag{37}$$

Fig. 7. Liquid tank system

If the initial condition is $x(0) = [(0.96 + 0.04\alpha, 1.01 - 0.01\alpha), p(0.75,0.82,0.9)]$, the static Bernstein neural network (28) has the form of

$$\begin{cases} \underline{x}_m(t,\alpha) = (0.96 + 0.04\alpha) \\ + t \sum_{i=0}^{N}\sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} t_i (T-t_i)^{N-i} \alpha_j (1-\alpha_j)^{M-j} \\ \overline{x}_m(t,\alpha) = (1.01 - 0.01\alpha) \\ + t \sum_{i=0}^{N}\sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} t_i (T-t_i)^{N-i} \alpha_j (1-\alpha_j)^{M-j} \end{cases}$$

where $t \in [0,1]$ and the approximate Z-number solution is termed as $[(\underline{x}_m(t,\alpha), \overline{x}_m(t,\alpha)), p(0.75,0.81,0.95)]$. Also dynamic Bernstein neural network (38) is used to approximate the solutions. To compare the results, the other generalization of neural network method [31] is used. The comparison results for Z-numbers are shown in Table 6. The specifications quoted here are $\eta = 0.001$ and $\gamma = 0.001$. Corresponding error plots are shown in Figure 8. These errors are the differences of the exact and the approximation solutions, for three different methods: SNN, DNN and NN Z-numbers. The results of Bernstein neural networks approximation for the fuzzy numbers are shown in Table 7. DNN is more accurate than the SNN and the other generalization of neural network method.
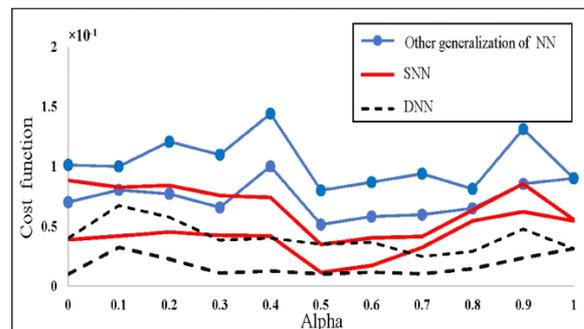

Fig. 8. The lower and upper bounds of absolute errors.

Table 5. Bernstein neural networks approximate the fuzzy numbers

α	SNN	DNN
0	[0.0511,0.0754]	[0.0224,0.0381]
0.1	[0.0402,0.0623]	[0.0203,0.0362]
0.2	[0.0398,0.0588]	[0.0197,0.0374]
0.3	[0.0224,0.0312]	[0.0211,0.0321]
0.4	[0.0433,0.0613]	[0.0246,0.0462]
0.5	[0.0507,0.0631]	[0.0152,0.0258]
0.6	[0.0469,0.0726]	[0.0191,0.0392]
0.7	[0.0571,0.0778]	[0.0288,0.0452]
0.8	[0.0373,0.0509]	[0.0157,0.0362]
0.9	[0.0401,0.0635]	[0.0202,0.0408]
1	[0.0394,0.0394]	[0.0167,0.0167]

5. Conclusions

In this paper, two types of Bernstein neural networks are used namely static and dynamic models to approximate the solutions of FDEs on the basis of Z-numbers. Initially the FDE is transformed into four ODEs with Hukuhara differentiability. Then neural models are constructed with the structure of ODEs. With modified back propagation method for Z-number variables, the neural networks are trained. Some real examples are employed to show the effectiveness of the proposed approximation methods with the Bernstein neural networks. The future works are the application of these mentioned methodologies for fuzzy partial differential equations on the basis of Z-numbers.

References

1. L.A. Zadeh, *Toward a generalized theory of uncertainty (GTU) an outline*, Inform. Sci., 172 (2005) 1-40.

2. T. Takagi, M. Sugeno, *Fuzzy identification of systems and its applications to modeling and control* ,IEEE Trans. Syst. Man. Cybern. 15 (1985) 116-132.

3. J.J. Buckley, Y. Hayashi, *Can fuzzy neural nets approximate continuous fuzzy functions*, Fuzzy. Sets. Syst. 61 (1994) 43-51.

4. S.S. Esmaili, A.M. Nasrabadi, *differential conditions in fuzzy Tumor model*, Journal of Biomedical Science and Engineering. 3 (2010)1001-1005.

5. R. Jafari, W. Yu, *Fuzzy Control for Uncertainty Nonlinear Systems with Dual Fuzzy Equations*, J. Intell. Fuzzy. Syst. 29 (2015) 1229-1240.

6. A. Jafarian, R. Jafari, *Approximate solutions of dual fuzzy polynomials by feed-back neural networks*, J. Soft. Comput. Appl., 2012, doi:10.5899/2012/jsca-00005.

7. A. Jafarian, R. Jafari, A. Khalili, D. Baleanud, *Solving fully fuzzy polynomials using feed-back neural networks*, International Journal of Computer Mathematics. 92(2015) 742-755.

8. A. Jafarian, R. Jafari, M. Mohamed Al Qurashi, D. Baleanud, *A novel computational approach to approximate fuzzy interpolation polynomials*, Springer Plus., 2016, 5:1428. doi:10.1186/s40064-016-3077-5.

9. S.L. Chang, L.A. Zadeh, *On fuzzy mapping and control*, IEEE Trans. Syst. Man Cybern. 2 (1972) 30-34.

10. D. Dubois, H. Prade, *Towards fuzzy differential calculus: part 3 differentiation,* Fuzzy Sets Syst. 8 (1982) 225-233.

11. B. Bede, I.J. Rudas, A.L. Bencsik, *First order linear fuzzy differential equations under generalized differentiability*, Inform. Sci. 177 (2007) 1648-1662.

12. A. Khastan, J.J. Nieto, R. *Rodríguez-López, Periodic boundary value problems for first order linear differential equations with uncertainty under generalized differentiability,* Inform. Sci. 222 (2013) 544-558.

13. T. Allahviranloo, N.A. Kiani, M. Barkhor- dari, *Toward the existence and uniqueness of solution of second-order fuzzy differential equations,* Inform. Sci. 179 (2009)1207-1215.

Table 6. Solutions of different methods based on Z-numbers

α	SNN	DNN	Neural network
0	[(0.0435, 0.0994),p(0.72,0.81,0.87)]	[(0.0112, 0.0442),p(0.75,0.82,0.88)]	[(0.0798, 0.1153),p(0.7,0.75,0.85)]
0.2	[(0.0504, 0.0940),p(0.7,0.8,0.9)]	[(0.0248, 0.0635),p(0.75,0.82,0.9)]	[(0.0878, 0.1375),p(0.7,0.8,0.85)]
0.4	[(0.0441, 0.0802),p(0.8,0.85,0.92)]	[(0.0131, 0.0422),p(0.8,0.9,1)]	[(0.1105, 0.1592),p(0.75,0.83,0.9)]
0.6	[(0.0178, 0.0423),p(0.8,0.92,1)]	[(0.0121, 0.0384),p(0.81,0.94,1)]	[(0.0613, 0.0915),p(0.8,0.9,1)]
0.8	[(0.0608, 0.0709),p(0.71,0.8,0.9)]	[(0.0154, 0.0309),p(0.8,0.87,0.95)]	[(0.0739, 0.0925),p(0.7,0.8,0.85)]
1	[(0.0611, 0.0611),p(0.75,0.82,0.91)]	[(0.0335, 0.0335),p(0.8,0.87,0.92)]	[(0.1007, 0.1007),p(0.7,0.8,0.9)]

Table 7. Solutions of different methods based on fuzzy numbers

α	SNN	DNN	Neural network
0	[0.0387, 0.0884]	[0.0101, 0.0398]	[0.0701, 0.1012]
0.2	[0.0451, 0.0841]	[0.0225, 0.0575]	[0.0771, 0.1207]
0.4	[0.0407, 0.0740]	[0.0125, 0.0401]	[0.1001, 0.1442]
0.6	[0.0169, 0.0402]	[0.0115, 0.0365]	[0.0582, 0.0868]
0.8	[0.0544, 0.0635]	[0.0144, 0.0289]	[0.0649, 0.0812]
1	[0.0554, 0.0554]	[0.0311, 0.0311]	[0.0901, 0.0901]

14. F. Hawrra, K.H. Amal, *On fuzzy Laplace transform for fuzzy differential equations of the third order, Journal of Kerbala University,* 11 (2013) 251-256.

15. L. Stefanini, B. Bede, *Generalized Hukuhara differentiability of interval-valued functions and interval differential equations*, J. Nonlinear Anal. (TMA). 71 (2009) 1311-1328.

16. J.J. Nieto, R. Rodríguez-López, M. Villanueva-Pesqueira, *Exact solution to the periodic boundary value problem for a first-order linear fuzzy differential equation with impulse*s, Fuzzy. Optim. Decis. Making. 10 (2011) 323-339.

17. O.S. Fard, A. Esfahani, A.V. Kamyad, *On solution of a class of fuzzy BVPs,* Iran. J. Fuzzy Syst. 9 (2012) 49-60.

18. H.K. Liu, *Comparison results of two-point fuzzy boundary value problems,* Int. J. Comput. Math. Sci. 5 (2011) 1-7.

19. O. Solaymani Fard, N. Ghal-Eh, *Numerical solutions for linear system of first-order fuzzy differential equations with fuzzy constant coefficients,* Inform. Sci. 181, (2011) 4765-4779.

20. M.B. Ahmadi, N.A. Kiani, N. Mikaeilvand, *Laplace transform formula on fuzzy nth order derivative and its application in fuzzy ordinary differential equations*, Soft. Comput. 18 (2014) 2461-2469.

21. E. Hullermeier, *An approach to modeling and simulation of uncertain dynamical systems,* Internat. J. Uncertainty Fuzzyness Knowledge-Based Syst. 5 (1997) 117-137.

22. T. Allahviranloo, N. Ahmadi, E. Ahmadi, *Numerical solution of fuzzy differential equations by predictor-corrector method,* Inform. Sci. 177 (2007) 1633-1647.

23. S. Tapaswini, S. Chakraverty, *Euler-based new solution method for fuzzy initial value problems*, Int. J. Artificial. Intell. Soft. Comput. 4 (2014) 58-79.

24. A. Khastan, K. Ivaz, *Numerical solution of fuzzy differential equations by Nystrom method*, Chaos, Solitons. Fractals. 41(2009) 859-868.

25. S. Abbasbandy, T. Allahvinloo, *Numerical solutions of fuzzy differential equations by Taylor method*, J. Comput. Meth. Appl. Math. 2 (2002) 113-124.

26. S.C. Palligkinis, G. Papageorgiou, I.T. Famelis*, Runge-Kutta methods for fuzzy differential equations,* Appl. Mathe. Comput. 209 (2008) 97-105.

27. J.J. Buckley, T. Feuring, *Fuzzy differential equations,* Fuzzy. Sets. Syst. 110 (2000) 69-77.

28. S. Agatonovic-Kustrin, R. Beresford, *Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research,* J. Pharm. Bio. Anal. 22 (2000) 717-727.

29. C. Filici, *On a neural approximate to ODEs*, IEEE Transactions on Neural Networks, 19 (2008) 539-543.

30. H. Lee, I.S. Kang, *Neural algorithms for solving differential equations,* Journal of Computational Physics. 91 (1990) 110-131.

31. S. Effati, M. Pakdaman*, Artificial neural network approach for solving fuzzy differential equations,* Inform. Sci. 180 (2010) 1434-1457.

32. L.A. Zadeh, *Generalized theory of uncertainty (GTU)-principal concepts and ideas*, Computational Statistics and Data Analysis. 51 (2006) 15-46.

33. B. Kang, D. Wei, Y. Li, Y. Deng, *A method of converting Z-number to classical fuzzy number*, Journal of Information and Computational Science.9 (2012)703-709.

34. L.A. Gardashova, *Application of operational approaches to solving decision making problem using Z-Numbers, Journal of Applied Mathematics.* 5 (2014) 1323-1334.

35. B. Kang, D. Wei, Y. Li, Y. Deng, *Decision making using Z-Numbers under uncertain environment*, Journal of Computational Information Systems. 8 (2012) 2807-2814.

36. P.J. Davis, *Interpolation and approximation*, Dover Publications, Inc, New York, 1975.

37. S. Curtis, S. Ghosh, *A variable selection approach to monotonic regression with Bernstein polynomials*, J. Appl. Stat. 38 (2011) 961-976.

38. R. Jafari, W. Yu, X. Li, *fuzzy differential equations for nonlinear system modeling with Bernstein neural networks*, IEEE Access, 4 (2016) 9428-9436,.

39. C. Wang, H. Zhang, W. Fan, X. Fan, *A new wind power prediction method based on chaotic theory and Bernstein neural network,* Energy, 117 (2016) 259-271.

40. S. Nahmias, *Fuzzy variables*, Fuzzy Sets Syst. 1 (1978) 97-110.

41. R.A. Alieva, W. Pedryczb, V. Kreinovich, O.H. Huseynov, *The general theory of decisions,* Inform. Sci., 327 (2016) 125-148.

42. B. Bede, L. Stefanini, *Generalized differentiability of fuzzy-valued functions,* Fuzzy. Sets. Syst. 23 (2013) 119-141.

43. R.A. Aliev, O.H. Huseynov, R.R. Aliyev, A.V. Alizadeh, *The arithmetic of Z-numbers. Theory and applications*, World Scientific, Singapore, 2015.

44. J.J. Buckley, T. Feuring, *Fuzzy differential equations*, Fuzzy Sets Syst. 110 (2000) 43-54.

45. S.S. Behzadi, T. Allahviranloo, *Solving fuzzy differential equations by using picard method*, Iranian Journal of Fuzzy Systems. 13 (2016) 71-81.

46. U. Kadak, F. Basar, *On some sets of fuzzy-valued sequences with the level sets,* Contemporary Analysis and Applied Mathematic. 1 (2013) 70-90.

47. D. Ralescu, *Toward a general theory of fuzzy variables, Journal of Mathematical Analysis and Applications.* 86 (1982) 176-193.

48. M. Hazewinkel, *Oscillator harmonic*, Springer, ISBN, 2001.

49. H.B. Cary, S.C. Helzer, *Modern Welding Technology*, Upper Saddle River, New Jersey: Pearson Education. 2005.

50. V.L. Streeter, E.B. Wylie, K.W. Bedford, *Fluid mechanics*, New York: McGraw Hill, 1998.