Uncertain nonlinear system control with Fuzzy Differential Equations and Z-numbers

Raheleh Jafari, Wen Yu[*]

Departamento de Control Automático

CINVESTAV-IPN (National Polytechnic Institute)

Mexico City, Mexico.

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. We first transform the FDE into four ordinary differential equations (ODEs) with Hukuhara differentiability. Then we construct neural models with the structure of ODEs. With modified backpropagation method for $Z-$number variables, the neural networks are trained. The simulation results show that these new models, Bernstein neural networks, are effective to estimate the solutions of FDEs based on Z-numbers.

1 Introduction

Since the uncertainty in parameters can be transformed into fuzzy set theory [37], 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 [35]. When the parameter of an equation are changeable in the manner of fuzzy set, this equation becomes a fuzzy equation [8]. 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 [15]. The applications of these FDEs are in connection with nonlinear

[*]The corresponding author is Wen Yu, tel.: +52-55-57473734, email: yuw@ctrl.cinvestav.mx

modeling and control [20, 21, 22, 23]. 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 [10]. Then it is extended in [13]. In [27], the first order FDE with periodic boundary conditions is analyzed. Then higher order linear FDE are studied.

Too much complexity is involved in solving nonlinear FDE. By interval-valued method, [33] examines the basis solutions nonlinear FDEs with generalized differentiability. [16] suggests some suitable criterion to fuzzify the crisp solutions. [29] uses two-point fuzzy boundary value for FDE. [18] uses homotopy analysis technique 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 [32] with an iterative technique. [3] uses Laplace transform for second-order FDE. Euler numerical technique is used in [36] to solve FDE. Some other numerical techniques, such as Nystrom approach [26], Taylor method [1] and Runge-Kutta approach [31] can also be applied to solve FDEs. However the approximation accuracy of these numerical calculations are normally less [30].

The solution of FDE is uniformly continuous and inside compact sets [7]. Neural networks can give a good estimation for the solutions of FDEs. [2] shows that the solution of ODE can be approximated by neural network. [28] applies dynamics neural networks to approximate first-order ODE. There are few works on FDE. [14] 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 [38]. 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 [38]. Z-numbers can be regarded as to answer questions and carry out the decisions [24]. There are few structure based on the theoretical concept of Z-numbers [17]. [4] gives an inception extend Z-numbers. [25] proposes a theorem to transfer the Z-numbers to the usual fuzzy sets.

In this paper, we use a new model named Bernstein neural network, which has good properties of Bernstein polynomial for FDE based on Z-number. The Bernstein polynomial has good uniform approximation ability for continuous functions [12]. Also a very important property of the Bernstein polynomial is that it generates a smooth estimation for equal distance knots [11]. This property is suitable for FDE approximation.

We use two types of neural networks: 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, we use two real examples to show the effectiveness of our 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 $f_1(x_1, u)$ is unknown vector function, $x_1 \in \Re^n$ is an internal state vector and $u \in \Re^m$ is the input vector.

In this paper, we use the following differential equation (FDE) to model the uncertain nonlinear system (1),

$$\frac{d}{dt}x = f(x, u) \tag{2}$$

where $x \in \Re^n$ 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, we first introduce some concepts of fuzzy variables and Z-numbers.

Definition 1 (fuzzy variable) *If x is: 1) normal, there exists $\zeta_0 \in \mathbb{R}$ in such a manner $x(\zeta_0) = 1$, 2) convex, $x(\lambda\zeta + (1-\lambda)\zeta) \geq \min\{x(\zeta), x(\xi)\}$, $\forall \zeta, \xi \in \mathbb{R}, \forall \lambda \in [0, 1]$, 3) upper semi-continuous on $\mathbb{R}$, $x(\zeta) \leq x(\zeta_0) + \varepsilon$, $\forall \zeta \in N(\zeta_0)$, $\forall \zeta_0 \in \mathbb{R},$, $\forall \varepsilon > 0$, $N(\zeta_0)$ is a neighborhood, 4) $x^+ = \{\zeta \in \mathbb{R}, x(\zeta) > 0\}$ is compact, then x is a fuzzy variable, $x \in E : \mathbb{R} \to [0, 1]$.*

The fuzzy variable x can be also represented as

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

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

Definition 2 (Z-numbers) *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, we use the definition of Z^+-number, 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

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

and trapezoidal function

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

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

$$[Z]^\alpha = ([x]^\alpha, [p]^\alpha) \tag{8}$$

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

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

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

$$[Z]^\alpha = \left(\underline{Z}^\alpha, \overline{Z}^\alpha\right) = \left(\left(\underline{x}^\alpha, \underline{P}^\alpha\right), \left(\overline{x}^\alpha, \overline{P}^\alpha\right)\right) \tag{9}$$

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

Similar with the fuzzy numbers [20], 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 [37]. 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 [20]

$$\begin{aligned} [x_1 \oplus x_2]^\alpha &= [\underline{x_1}^\alpha + \underline{x_2}^\alpha, \overline{x_1}^\alpha + \overline{x_2}^\alpha] \\ [x_1 \ominus x_2]^\alpha &= [\underline{x_1}^\alpha - \underline{x_2}^\alpha, \overline{x_1}^\alpha - \overline{x_2}^\alpha] \\ [x_1 \odot x_2]^\alpha &= \left(\underline{x_1}^\alpha \underline{x_2}^\alpha + \underline{x_1}^\alpha \underline{x_2}^\alpha - \underline{x_1}^\alpha \underline{x_2}^\alpha, \overline{x_1}^\alpha \overline{x_2}^\alpha + \overline{x_1}^\alpha \overline{x_2}^\alpha - \overline{x_1}^\alpha \overline{x_2}^\alpha\right) \end{aligned} \tag{10}$$

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 [5],

$$\begin{gathered} Z_1 \ominus_H Z_2 = Z_{12} \\ Z_1 = Z_2 \oplus Z_{12} \end{gathered}$$

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

$$[Z_1 \ominus_H Z_2]^\alpha = [\underline{Z}_1^\alpha - \underline{Z}_2^\alpha, \overline{Z}_1^\alpha - \overline{Z}_2^\alpha]$$

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

Also the above definitions satisfy the generalized Hukuhara difference [6]

$$Z_1 \ominus_{gH} Z_2 = Z_{12} \Longleftrightarrow \begin{cases} 1)\ Z_1 = Z_2 \oplus Z_{12} \\ 2)\ Z_2 = Z_1 \oplus (-1) Z_{12} \end{cases} \tag{11}$$

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 $[Z_1 \ominus_{gH} Z_2]^\alpha = [\min\{\underline{Z}_1^\alpha - \underline{Z}_2^\alpha, \overline{Z}_1^\alpha - \overline{Z}_2^\alpha\}, \max\{\underline{Z}_1^\alpha - \underline{Z}_2^\alpha, \overline{Z}_1^\alpha - \overline{Z}_2^\alpha\}]$ 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 circumstances for the inerrancy of $Z_{12} = Z_1 \ominus_{gH} Z_2 \in E$ are

$$\begin{aligned} &1) \begin{cases} \underline{Z}_{12}^\alpha = \underline{Z}_1^\alpha - \underline{Z}_2^\alpha \text{ and } \overline{Z}_{12}^\alpha = \overline{Z}_1^\alpha - \overline{Z}_2^\alpha \\ \text{with } \underline{Z}_{12}^\alpha \text{ increasing, } \overline{Z}_{12}^\alpha \text{ decreasing, } \underline{Z}_{12}^\alpha \leq \overline{Z}_{12}^\alpha \end{cases} \\ &2) \begin{cases} \underline{Z}_{12}^\alpha = \overline{Z}_1^\alpha - \overline{Z}_2^\alpha \text{ and } \overline{Z}_{12}^\alpha = \underline{Z}_1^\alpha - \underline{Z}_2^\alpha \\ \text{with } \underline{Z}_{12}^\alpha \text{ increasing, } \overline{Z}_{12}^\alpha \text{ decreasing, } \underline{Z}_{12}^\alpha \leq \overline{Z}_{12}^\alpha \end{cases} \end{aligned} \tag{12}$$

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

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

$$|Z(\zeta)| = (|a_1| + |b_1| + |c_1|, p(|a_2| + |b_2| + |c_2|)) \tag{13}$$

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 [20]. $D(Z_1, Z_2)$ is incorporated with the following possessions:

$$\begin{aligned}
&D(Z_1 + Z, Z_2 + Z) = D(Z_1, Z_2)\\
&D(Z_2, Z_1) = D(Z_1, Z_2)\\
&D(kZ_1, kZ_2) = |k| D(Z_1, Z_2)\\
&D(Z_1, Z_2) \le D(Z_1, Z) + D(Z, Z_2)
\end{aligned}$$

where $k \in \mathbb{R}$, $Z = (x, p)$ is Z-number and x is triangle function.

Definition 4 ($\alpha-$level of Z-number valued function) *Let $\widetilde{Z}$ denotes the space of Z-numbers. The $\alpha-$level of Z-number valued function $F : [0, a] \to \widetilde{Z}$ is*

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

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

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

$$\frac{d}{dt} F(x_0) = \lim_{h \to 0} \frac{1}{h} [F(x_0 + h) \ominus_{gH} F(x_0)] \tag{14}$$

In (14), $F(x_0 + h)$ and $F(x_0)$ exhibits similar style with Z_1 and Z_2 respectively included in (11).

If we apply the $\alpha-$level (8) to $f(t, x)$ in (2), then we obtain two Z-number valued functions: $\underline{f}\,[t, \underline{x}(\zeta, \alpha), \bar{x}(\zeta, \alpha)]$ and $\overline{f}\,[t, \underline{x}(\zeta, \alpha), \bar{x}(\zeta, \alpha)]$.

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

$$\begin{aligned}
&1) \begin{cases} \frac{d}{dt}\underline{x} = \underline{f}\,[t, \underline{x}(\zeta, \alpha), \bar{x}(\zeta, \alpha)] \\ \frac{d}{dt}\bar{x} = \overline{f}\,[t, \underline{x}(\zeta, \alpha), \bar{x}(\zeta, \alpha)] \end{cases}\\
&2) \begin{cases} \frac{d}{dt}\underline{x} = \overline{f}\,[t, \underline{x}(\zeta, \alpha), \bar{x}(\zeta, \alpha)] \\ \frac{d}{dt}\bar{x} = \underline{f}\,[t, \underline{x}(\zeta, \alpha), \bar{x}(\zeta, \alpha)] \end{cases}
\end{aligned} \tag{15}$$

The fuzzy model of (1) can be regarded as four ordinary differential equations (15).

Figure 1: Nonlinear system modeling with fuzzy differential equation

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

$$\min_{f} \|x - x_1\| \tag{16}$$

This modeling object can be considered as: finding $\overline{f}$ and $\underline{f}$ in the fuzzy equations of (15) or finding the soultions of these models. It is impossible to obtain analytical solutions. In this paper, we use neural networks to approximate them, see Figure 1.

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.

3 Solving fuzzy differential equation with neural networks

In general, it is difficult to solve the four equations (15) or (2). In this paper, we use a special neural network named Bernstein neural network to approximate the solutions of the FDE (2).

The Bernstein neural network use the following Bernstein polynomial,

$$\begin{array}{c} B(x_1, x_2) = \sum_{i=0}^{N} \sum_{j=0}^{M} \binom{N}{i} \binom{M}{j} \\ W_{i,j} x_{1i} (T - x_{1i})^{N-i} x_{2j} (1 - x_{2j})^{M-j} \end{array} \tag{17}$$

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,

$$y = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j W_{i,j} x_{1i} (T - x_{1i})^{N-i} x_{2j} (1 - x_{2j})^{M-j} \tag{18}$$

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

Because the Bernstein neural network (18) has similar forms as (15), we use the Bernstein neural network (18) to approximate the solutions of four ODEs in (15).

If x_1 and x_2 in (17) 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

$$\begin{array}{c} x_m(t,\alpha) = x_m(0,\alpha) \\ \oplus t \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j W_{i,j} t_i (T - t_i)^{N-i} \alpha_j (1 - \alpha_j)^{M-j} \end{array} \tag{19}$$

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

so the derivative of (18) is

$$\begin{array}{l} 1) \begin{cases} \frac{d}{dt}\underline{x}_m = C_1 + C_2 \\ \frac{d}{dt}\bar{x}_m = D_1 + D_2 \end{cases} \\ 2) \begin{cases} \frac{d}{dt}\underline{x}_m = C_1 + C_2 \\ \frac{d}{dt}\bar{x}_m = D_1 + D_2 \end{cases} \end{array} \tag{20}$$

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

$$\begin{array}{l} C_1 = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} t_i (T - t_i)^{N-i} \alpha_j (1 - \alpha_j)^{M-j} \\ D_1 = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} t_i (T - t_i)^{N-i} \alpha_j (1 - \alpha_j)^{M-j} \\ C_2 = t_k \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} [i t_{i-1,j} (T - t_i)^{N-i} \\ - (N-i)\, t_{i,j} (T - t_i)^{N-i-1}] \alpha_j^i (1 - \alpha_j)^{M-j} \\ D_2 = t_k \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} [i t_{i-1,j} (T - t_i)^{N-i} \\ - (N-i)\, t_{i,j} (T - t_i)^{N-i-1}] \alpha_j^i (1 - \alpha_j)^{M-j} \end{array}$$

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

- input unit:

$$o_1^1 = t, \quad o_2^1 = \alpha$$

- the first hidden units:

$$\begin{array}{ll} o_{1,i}^2 = f_i^1(o_1^1), & o_{2,i}^2 = f_i^2(o_1^1) \\ o_{3,j}^2 = g_j^1(o_2^1), & o_{4,j}^2 = g_j^2(o_2^1) \end{array}$$

- the second hidden units:

$$o_{1,i}^3 = o_{1,i}^2(o_{2,i}^2), \quad o_{2,j}^3 = o_{3,j}^2(o_{4,j}^2)$$

Figure 2: Static Bernstein neural network

- 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 $f_i^1 = t^i$, $f_i^2 = (T-t)^{N-i}$, $\lambda_i = \frac{1}{T^N}\binom{N}{i}$, $g_j^1 = \alpha^j$, $g_j^2 = (1-\alpha)^{M-j}$, $\gamma_j = \binom{M}{j}$.
We define the training errors between (20) and (15) as

$$\begin{array}{l} 1)\left\{ \begin{array}{l} \underline{e}_1 = C_1 + C_2 - \underline{f} \\ \bar{e}_1 = D_1 + D_2 - \bar{f} \end{array} \right. \\ 2)\left\{ \begin{array}{l} \underline{e}_2 = C_1 + C_2 - \bar{f} \\ \bar{e}_2 = D_1 + D_2 - \underline{f} \end{array} \right. \end{array} \tag{21}$$

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

$$\begin{array}{l} \underline{W}_{i,j}(k+1) = \underline{W}_{i,j}(k) - \eta_1\left(\frac{\partial \underline{e}_1^2}{\partial \underline{W}_{i,j}} + \frac{\partial \bar{e}_1^2}{\partial \underline{W}_{i,j}}\right) \\ \overline{W}_{i,j}(k+1) = \overline{W}_{i,j}(k) - \eta_2\left(\frac{\partial \underline{e}_2^2}{\partial \overline{W}_{i,j}} + \frac{\partial \bar{e}_2^2}{\partial \overline{W}_{i,j}}\right) \end{array} \tag{22}$$

where η_1 and η_2 are positive learning rates.

Figure 3: Dynamic Bernstein nerual network

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

$$\begin{cases} \frac{d}{dt}\underline{x}_m(t,\alpha) = \underline{P}(t,\alpha)A(\underline{x}_m(t,\alpha),\bar{x}_m(t,\alpha)) + \underline{Q}(t,\alpha) \\ \frac{d}{dt}\bar{x}_m(t,\alpha) = \overline{P}(t,\alpha)A(\underline{x}_m(t,\alpha),\bar{x}_m(t,\alpha)) + \overline{Q}(t,\alpha) \end{cases} \tag{23}$$

Obviously this dynamic network has the form of

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

and it is closed to (2).

The training algorithm is similar as (22), only the training errors are changed as

$$\begin{array}{l} 1)\begin{cases} \underline{e}_1 = C_1 + C_2 - \underline{P}A(\underline{x}_m,\bar{x}_m) - \underline{Q} \\ \bar{e}_1 = D_1 + D_2 - \overline{P}A(\underline{x}_m,\bar{x}_m) - \overline{Q} \end{cases} \\ 2)\begin{cases} \underline{e}_2 = C_1 + C_2 - \overline{P}A(\underline{x}_m,\bar{x}_m) - \overline{Q} \\ \bar{e}_2 = D_1 + D_2 - \underline{P}A(\underline{x}_m,\bar{x}_m) - \underline{Q} \end{cases} \end{array} \tag{24}$$

4 Applications

In this section, we use several real applications to show how to use the Bernstein neural networks to approximate the solutions of the FDEs.

Figure 4: Vibration mass

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

$$\frac{d}{dt}x(t) = \frac{k}{m}x(t) \tag{25}$$

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 (25) can be formed into a FDE based on Z-number. It has the same form as (25), 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 (25) is [19]*

$$x(t, \alpha) = \left[((0.75 + 0.25\alpha)e^t, (1.125 - 0.125\alpha)e^t), p(0.8, 0.9, 1)\right] \tag{26}$$

where $t \in [0, 1]$*. Now we use the static Bernstein neural network (19), noted as SNN to approximate the Z-number solution* $[(\underline{x}_m(t, \alpha), \overline{x}_m(t, \alpha)), p(0.8, 0.94, 1)]$ *(26) where*

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

We also use dynamic Bernstein neural network (23), noted as DNN to approximate the solutions. The learning rates are $\eta = 0.01$, $\gamma = 0.01$*. To compare our results, we use the other two popular methods: Max-Min Euler method and Average Euler method [36]. The comparison results are shown in Table 1 and Table 2. Corresponding solution plots are shown in Figure 5.*

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

Figure 5: Comparison plots of SNN, DNN, Max-Min Euler, Average Euler and the exact solution 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)]*

Table 2. Approximation errors based on Z-numbers

α	*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)]*

We use the following to transfer the $Z-$numbers to fuzzy numbers,

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

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.*

Table 3. Solutions of different methods based on fuzzy numbers

Figure 6: Z-number and fuzzy number

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]*

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. We see 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 [9]:*

$$\frac{d}{dt}x(t) = 3Ax^2(t) \tag{27}$$

where transfer area A is uncertainty as $A = [(1+\alpha, 3-\alpha), p(0.8, 0.87, 0.95)]$, $\alpha \in [0, 1]$. So (27) 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 (19) has the form of*

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

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.*

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)]

5 Conclusions

In this paper, we use two types of Bernstein neural networks: static and dynamic models to approximate the solutions of FDEs on the basis of Z-numbers. We first transform the FDE into four ODEs with Hukuhara differentiability. Then we construct neural models with the structure of ODEs. With modified backpropagation method for Z-number variables, the neural networks are trained. Two real examples are employed to show the effectiveness of our 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] S. Abbasbandy, T. Allahvinloo, Numerical solutions of fuzzy differential equations by Taylor method, *J. Comput. Meth. Appl. Math.* Vol.2, pp.113-124, 2002.

[2] S. Agatonovic-Kustrin, R. Beresford, Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, *Journal of Pharmaceutical and Biomedical Analysis,* vol.22, pp.717-727, 2000.

[3] 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.* Vol.18, pp.2461-2469, 2014.

[4] R.A. Aliev, A.V. Alizadeh, O.H. Huseynov, The arithmetic of discrete Z-numbers, *Inform. Sci.* Vol.290, pp.134-155, 2015.

[5] R.A. Alieva, W. Pedryczb, V. Kreinovich, O.H. Huseynov, The general theory of decisions, *Inform. Sci.* Vol.327, pp.125-148, 2016.

[6] B. Bede, L. Stefanini, Generalized differentiability of fuzzy-valued functions, *Fuzzy Sets Syst.* Vol.230, pp.119-141, 2013.

[7] J.J. Buckley, T. Feuring, Fuzzy differential equations, *Fuzzy Sets Syst.* Vol.110, pp.69-77, 2000.

[8] J.J. Buckley, Y. Hayashi, Can fuzzy neural nets approximate continuous fuzzy functions, *Fuzzy Sets Syst.* Vol.61, pp.43-51, 1994.

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

[10] S.L. Chang, L.A. Zadeh, On fuzzy mapping and control, *IEEE Trans. Syst. Man Cybern.* Vol.2, pp.30-34, 1972.

[11] S. Curtis, S. Ghosh, A variable selection approach to monotonic regression with Bernstein polynomials, *J. Appl. Stat.* Vol.38, pp.961-976, 2011.

[12] P.J. Davis, Interpolation and approximation, *Dover Publications, Inc, New York,* 1975.

[13] D. Dubois, H. Prade, Towards fuzzy differential calculus: part 3 differentiation, *Fuzzy Sets Syst.* Vol.8, pp.225-233, 1982.

[14] S. Effati, M. Pakdaman, Artificial neural network approach for solving fuzzy differential equations, *Inform. Sci.* Vol.180, pp.1434-1457, 2010.

[15] S.S. Esmaili, A.M. Nasrabadi, Different initial conditions in fuzzy Tumor model, *Journal of Biomedical Science and Engineering,* vol.3, pp.1001-1005, 2010.

[16] O.S. Fard, A. Esfahani, A.V. Kamyad, On solution of a class of fuzzy BVPs, *Iran. J. Fuzzy Syst.* Vol.9, pp.49-60, 2012.

[17] L.A. Gardashova, Application of operational approaches to solving decision making problem using Z-Numbers, *Journal of Applied Mathematics,* Vol.5 pp.1323-1334, 2014.

[18] M.S. Hashemi, J. Malekinagad, Series solution of fuzzy wave-like equations with variable coefficients, *J. Intell. Fuzzy Syst.* Vol.25, pp.415-428, 2013.

[19] M. Hazewinkel, Oscillator harmonic, *Springer, ISBN,* 2001.

[20] R. Jafari, W. Yu, Fuzzy Control for Uncertainty Nonlinear Systems with Dual Fuzzy Equations, *J. Intell. Fuzzy. Syst.* Vol.29, pp.1229-1240, 2015.

[21] 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.

[22] A. Jafarian, R. Jafari, A. Khalili, D. Baleanud, Solving fully fuzzy polynomials using feed-back neural networks, *International Journal of Computer Mathematics*, Vol.92, pp.742-755, 2015.

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

[24] B. Kang, D. Wei, Y. Li, Y. Deng, A method of converting Z-number to classical fuzzy number, *Journal of Information and Computational Science,* Vol.9, pp.703-709, 2012.

[25] B. Kang, D. Wei, Y. Li, Y. Deng, Decision making using Z-Numbers under uncertain environemnt, *Journal of Computational Information Systems,* Vol.8, pp.2807-2814, 2012.

[26] A. Khastan, K. Ivaz, Numerical solution of fuzzy differential equations by Nystrom method, *Chaos, Solitons. Fractals.* Vol. 41, pp.859-868, 2009.

[27] 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.* Vol.222, pp.544-558, 2013.

[28] H. Lee, I.S. Kang, Neural algorithms for solving differential equations, *Journal of Computational Physics,* Vol.91, pp.110-131, 1990.

[29] H.K. Liu, Comparison results of two-point fuzzy boundary value problems, *Int. J. Comput. Math. Sci.* Vol.5, pp.1-7, 2011.

[30] W.A. Lodwick, K.D. Jamison, Special issue: interfaces between fuzzy set theory and interval analysis, *Fuzzy Sets Syst.* Vol.135, pp.1-3, 2003.

[31] S.C. Palligkinis, G. Papageorgiou, I.T. Famelis, Runge-Kutta methods for fuzzy differential equations, *Appl. Mathe. Comput.* Vol.209, pp.97-105, 2008.

[32] O. Solaymani Fard, N. Ghal-Eh, Numerical solutions for linear system of first-order fuzzy differential equations with fuzzy constant coeffcients, *Inform. Sci.* Vol.181, pp.4765-4779, 2011.

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

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

[35] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst. Man. Cybern.* Vol.15, 116-132, 1985.

[36] S. Tapaswini, S. Chakraverty, Euler-based new solution method for fuzzy initial value problems, *Int. J. Artificial. Intell. Soft. Comput.* Vol.4, pp.58-79, 2014.

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

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