SOLVING FUZZY VOLterra-FREDholm INTEGRAL EQUATION BY FUZZY ARTIFICIAL NEURAL NETWORK

SEIYED HADI ABTAHI AND HAMIDREZA RAHIMI∗
Department of Mathematics, Faculty of Science, Central Tehran Branch
Islamic Azad University
Tehran, Iran

MARYAM MOSLEH
Department of Mathematics, Faculty of Basic Science, West Tehran Branch
Islamic Azad University
Tehran, Iran

(Communicated by Ana Maria Acu)

Abstract. The volterra-fredholm integral equation in all forms are arose from physics, biology and engineering problems which is derived from differential equation modelling. On the other hand, the trained programming algorithm by the fuzzy artificial neural networks has effective solution to find the best answer. In this article we try to estimate the equation and its answer by developed fuzzy artificial neural network to fuzzy volterra-fredholme integral. Our attempts would lead to benchmark other extended forms of this type of equation.

1. Introduction. What we use in this article is a constructive interference of the artificial intelligence and fuzzy logic which appear in the form of fuzzy artificial neural network (FANN) through out history. Initial attempt to construct artificial intelligence’s (AI) and neural network concept were accomplished by Werren McCulloach and Walter Pitts in 1943 [32]. In 1949 Donald Hebb developed a neuropsychological theory and its rule [17] known as “Hebbian learning” [34]. In 1950 Marvin Minsky and Dean Edmonds designed the first neurocomputer [48]. We can call the year 1956 “the birth of artificial intelligence”, when McCarthy, Minsky Rochester and Shavron as AI’s pioneer held Dartmouth summer conference about AI [3, 36]. Afterwards and during 1950s’ some extensions about AI were accomplished. Arthur Sumuel wrote checkers player machine which defeated its creator in the game and developed his studies in machine learning [45]. During those years, McCarthy by developing of the first complete AI system [31].

We want to proceed AI history in point of view of ANN since the advent. J. Kelly, H. Arthur and E. Bryson described the basics concept of continuous back propagation which is a common method of algorithm learning and trained system in ANN recently [47, 53]. Then multi-stage dynamic system optimization described

2020 Mathematics Subject Classification. Primary: 45D05; Secondary: 92B20.
Key words and phrases. Fuzzy volterra-fredholm integral equation, fuzzy artificial neural network, trained algorithm.
∗ Corresponding author.
by Bryson and Yuchi ho in 1969 [8]. In 1974 Werbos stated the possibility of applying backpropagation in an artificial neural network [52]. In 1982 Hopfield [19] and in 1988 Rumehart et al [43] developed the ANN learning algorithm. Finally E. A. Wan was the one who won international pattern recognition contest through back propagation [49]. So far we introduce the system that its’ input, processing unit and output with crisp number. What we want to attain is a system that deal with relative number. This idea arised from the fuzzy logic which expressed by L. Zadeh in 1965 [54]. Actually we need hybrid system to use it as a final tool. Hybrid systems are combination on fuzzy logic, neural network, genetic algorithms and learned machine that called expert system created by humans [13]. Wang and Chang emphasized on the crucial rule of fuzzy set applications on the information systems [50]. Applications of fuzzy function to differentiation of data was the next step in developing of the fuzzy machine learning. The first work in developing the NN learning methods with fuzzy system was proposed by Keller and Hunt in 1985 [23]. They applied fuzzy membership function in the perceptron algorithm. In 1993 Khan and Venkatapuram classified the structure of a complete FANN which is included fuzzy inferencing and defuzzification, and application of fuzzy membership in processing levels [24]. Hayashi and Buckley presented an advanced model of FANN algorithm which deals with fuzzy biases and fuzzy weights. These parameters are symmetric triangular form of fuzzy numbers [16]. Solving problems, equations or approximation of functions are common and useful application of FANN. Approximating between fuzzy expert system and neural network was the starting point of this challenge [9]. In the following we deal with volterra-fredholm integral equation. This kind of equation is combination of disjoint volterra and fredholm integral [51]. Integral equations in kind of volterra-fredholm can appear in two forms, mixed form and disjoint form. This integra equations come from physical, biological and engineering problems when it confront with differential system in parabolic forms [40]. Researchers have always been attempting to solve physics problems with least error [1]. Further more to solve this type of integral equation much efforts has been made [7, 10, 30, 35]. What made us enthusiasm to solve equation by fuzzy perspective was initial value. Using deterministic initial value which isn’t determined exactly in the function with unknown parameters, would increase error or solving time [46]. Therefore, we would implement method in solving fuzzy volterra-fredholm integral equation with fuzzy initial number instead of deterministic form of this integral equation. The initial form to solve fuzzy equations was studied by Kalera [22] in 1987. Jeong and Park proofed the exitance of Volterra-Fredholm integral equations [41]. This article focus on finding solution to this integral equation in disjoint form and joinable kernels by FANN method. This method has been used in other integral equations in recent years [38]. In this paper, we present a method for solving a type of integral equations that can be considered as an applicable approach for solving similar equations in all related sciences. The proposed method is an extension of former methods and the ANN solution. The main challenge in solving these equations by ANN method emerges when the machine is trained to identify unknowns and move towards solving them. The main obstacle is in training an IT-based system to recognize the analysis method and problem solving. Moreover, achieving the result with maximum accuracy and minimum duration is another obstacle. Despite the existing obstacles, we overcame the problem. We implemented an effective learning system by applying ANN method, upon making use of integral linearization. The main idea was in designing the learning algorithm and performing its steps. The
result of our effort is design of an intelligent system that has the ability to solve applicable integral equations with the least error and the highest speed. This method is unique to solve Volterra-Fredholm integral equations.

2. Preliminaries.

2.1. Fuzzy number. In this part we come with a brief explanation of fuzzy number and operations on it. At first we define fuzzy set.

Definition 2.1. Let V be nonempty set. The fuzzy set M is characterized in V by η as a membership function $\eta_M : V \rightarrow [0, 1]$, $\eta_M(z)$ allocate a degree to every member in M for $z \in V$. So $M = \{(z, \eta_M(z)) | z \in V\}$.

Definition 2.2. Let $M$ is a fuzzy set in $V$ as a reference set. Let $\eta_m : V \rightarrow [0, 1]$ is membership continues function. This function characterized $M$ by assigning a number in $[0,1]$ to each element on $M$ [37]. We can define union and intersection operations for fuzzy sets $K$ and $N$ by following:

$$\forall l \in V : \eta_{K \cap N} = \min \{\eta_K(l), \eta_N(l)\}$$

$$\forall l \in V : \eta_{K \cup N} = \min \{\eta_K(l), \eta_N(l)\}.$$  

Definition 2.3. The presentation $(K, \eta)$ is a fuzzy number if it has following properties:

1. $K$ is compact in $\mathbb{R}$
2. $\eta$ is continues over $\mathbb{R}$
3. $\forall l \in K : \eta(l) \geq 0$; $\forall l \in \mathbb{R} - K : \eta(l) = 0$
4. $\exists l_0 \in K : (l_0) = 1$
5. $\eta$ is ascending over $[-\infty, l_0]$ and descending over $[l_0, +\infty]$, [11].

Definition 2.4. [2] Three points $l_1$, $l_2$ and $l_3$ make presentation $\tilde{A} = (l_1, l_2, l_3)$ A triangular fuzzy number if it has following conditions:

1. $l_1$ and $l_2$ is increasing function
2. $l_2$ and $l_3$ is decreasing function
3. $l_1 \leq l_2 \leq l_3$
4. $\eta_{\tilde{A}}(x) = \begin{cases} 0 & \text{if } x < l_1 \text{ and } x > l_3 \\ \frac{x - l_1}{l_2 - l_1} & \text{if } l_1 \leq x \leq l_2 \\ \frac{l_3 - x}{l_3 - l_2} & \text{if } l_2 \leq x \leq l_3 \end{cases}$

Definition 2.5. [44] $\alpha$-cut on triangular fuzzy number $K$, is a function on it and present a representation in form of $K_{\alpha} = [k_{L_{\alpha}}, k_{R_{\alpha}}]$ for $\alpha \in (0, 1]$. $[k_{L_{\alpha}}, k_{R_{\alpha}}] = [(k_2 - k_1)\alpha + k_1, -(k_3 - k_2)\alpha + k_3]$.

2.2. Arithmetic of fuzzy number. Here we come with definition of fuzzy arithmetic and operation on two fuzzy number which applied in current article.

Definition 2.6. [5] Let $A_{\alpha} = [a_{\alpha}, a_{\alpha}]$ and $B_{\alpha}$ be two fuzzy number with $\alpha$-cut, then we have:

$$A_{\alpha} + B_{\alpha} = [a_{L_{\alpha}}, a_{R_{\alpha}} + b_{R_{\alpha}}]$$
$$A_{\alpha} - B_{\alpha} = [a_{L_{\alpha}} - b_{L_{\alpha}}, a_{R_{\alpha}} - b_{R_{\alpha}}]$$
$$A_{\alpha} \times B_{\alpha} = [a_{L_{\alpha}} b_{L_{\alpha}}], a_{R_{\alpha}} b_{R_{\alpha}}]$$
$$A_{\alpha} \div B_{\alpha} = [a_{L_{\alpha}} / b_{L_{\alpha}}, a_{R_{\alpha}} / b_{R_{\alpha}}]$$
$$\forall k > 0 : (k.A)_{\alpha} = k.A_{\alpha} = [ka_{L_{\alpha}}, ka_{R_{\alpha}}].$$
In particular for character function we have:

\[
\eta_{(A+B)}(c) = \sup_{(c=a+b)} \min(\eta_A(a), \eta_B(b)) \\
\eta_{(A-B)}(c) = \sup_{(c=a-b)} \min(\eta_A(a), \eta_B(b)) \\
\eta_{(A.B)}(c) = \sup_{(c=ab)} \min(\eta_A(a), \eta_B(b)) \\
\eta_{(A+\bar{B})}(c) = \sup_{(c=a/\bar{b})} \min(\eta_A(a), \eta_B(b)) 
\]

It is necessary to mention two type of fuzzy number:

**Definition 2.7.** A symmetric triangular fuzzy number has same distance from left and right to center number by any horizon from left and right to center and its presentation is \(A = \left[ a_L, a_C, a_R \right] \).

2.3. **Artificial fuzzy neural network.** Artificial neural network is one of the most special method in programming which is presented by scientists. Computer with use of this method can solve complex problems easily [39]. An artificial neural network has an interconnection structure which is inspired by processing unit and nodes whose based on human brain [15].

**Definition 2.8.** Let \( N \) is the set of neurons and \( \{ (k,l) | k,l \in N \} \) whose elements connect between neuron \( k \) and neuron \( l \). A neural network is \( (N,V,w) \) where a function \( w : V \rightarrow R \) defines the weights. We called \( w((k,l)) \), the weight of the connection between two neurons \( k \) and \( l \) and it is shortened to \( w_{k,l} \) [25].

In the neural network two function play an important role in hidden unit and output unit, propagation function and activation function [33]. The structure of neural network has many common form. the simple one include: input layer which receives data, hidden layer which processes the data and output layer( brings out the results).neural networks can also have many hidden layers by the complexity of problems. Duty of propagation function is receiving and transferring outputs of other neurons to processing layer. This transferring is accompanied by tagging weights to data in terms of level of importance. Activation function process these data in the layer. In ANN threshold values assign to certain neuron uniquely. Threshold value labels the conditions of activation function in terms of maximum of gradient value. So for the set of neurons \( I = \{ i_1, i_2, ..., i_n \} \) and for \( w_{ij} \) as weight of the neurons connection, the network input of \( j \) called \( net_j \). So for \( (o_{i1}, ..., o_{in}) \) we have:

\[
net_j = f_{prop}(o_{i1}, ..., o_{in}, w_{(i1,j)}, ..., w_{(in,j)})
\]

and

\[
a_j(t) = f_{act}(net_j(t), a_j(t-1), \theta_j).
\]

Conversions the ANN inputs \( net_j \) along to the former activation state \( a_j(t-1) \) to the next activation condition \( a_j(t) \) with threshold value \( \theta \).

There are many kind of neural networks which used by the type of problem, changing process units, input and outputs. By the kolomogorov existence theorem, a three layered perceptron which include \( (2n+1)n \) nodes can calculate \( n \) variable continuous functions [18]. Number of neurons in hidden layers determine approximating in ANN structure, so we can solve and equation by ANN algorithms. As we told the expansion of ANN in the form of fuzzy was appeared in modern form when ishibuchi et al used fuzzy weights in network [16]. In the following we will define FANN principals. Let \( w_{ij} = (w_{ij}^L, w_{ij}^C, w_{ij}^R) \) and \( v_{ij} = (v_{ij}^L, v_{ij}^C, v_{ij}^R) \) are fuzzy connections weights . then we have: Input units:

\[
x = (o_{i1}, ..., o_{in})
\]
Hidden units:
\[ z_{ij} = f(\text{net}_{ij}) \text{ and } \text{net}_{ij} = xw_{ij} + b_{ij} \]

Output unit:
\[ N_i = f(\text{Net}_i) \text{ and } \text{Net}_i = \sum_{j=1}^{m} z_{ij}v_{ij} \]

where \( f \) is activation function and \( b_{ij} \) is bias. We can also write below structure for \( h \)-level sets [21]:

Input units:
\[ x = (o_{i1}, ..., o_{in}) \]

Hidden units:
\[ z_{ij} = f(\text{net}_{ij}) \text{ and } \text{net}_{ij} = \sum_{i=1}^{n} xw_{ij} + b_{ij} \]

Output unit:
\[ [N_i]_h = [[N_i]_h^L, [N_i]_h^U] = [f[N_i]_h^L, f[N_i]_h^U] \]

where \( f \) is an activation function. In the following we define \( h \)-level set of \( \tilde{F} \) as a fuzzy number [27]:
\[ [\tilde{F}]_h = \{ \beta | \mu_{\tilde{F}}(\beta) \geq h, \beta \in R \text{ for } 0 < h \leq 1 \} \]

3. Volterra-fredholm integral equation. In this paper we considers volterra fredholm integral equation in first kind of the forms:
\[ f(x) = \int_a^b k_1(x, y)u(y)dy + \int_a^x k_2(x, y)u(y)dy \]
which \( f(x) \) and \( u(y) \) are polynomials and \( k_1 \) and \( k_2 \) are separable kernels [51]. These kernels can separate in form of:
\[ k_1(x, y) = \sum_{j=1}^{n} g_j(x)h_j(y) \]
\[ k_2(x, y) = \sum_{i=1}^{m} m_i(x)n_i(y) \]

[14, 28]. So:
\[ f(x) = u(x) + \sum_{j=1}^{n} g_j(x)\int_a^b h_j(y)u(y)dy + \sum_{i=1}^{m} m_i(x)\int_a^x n_i(y)u(y)dy. \]

By newton-cotes formula and Bools’ rule [4, 26] we approximate equation. The general form of newton-cotes approximated integral in form of:
\[ \int_a^b f(x)dx \approx \sum_{j=0}^{n} w_jf(x_j) \]

with:
\[ x_j = a + j \frac{b-a}{n}. \]
We call $\frac{b-a}{n}$ step size and by changing this step we can derive some different methods. The Bool’s rule use size four for step $h = \frac{b-a}{4}$, so for supposed integral $\int_{a}^{b} f(x) \, dx$ we have:

$$x_j = a + j \frac{b-a}{4} = a + j h \text{ for } j = \{0, ..., 4\},$$

so:

$$a = (x_0), (x_1) = (x_0) + h, (x_2) = (x_1) + h, (x_3) = (x_2) + h, b = (x_4) = (x_3) + h$$

so expanding integral by this rule we have:

$$\int_{a}^{b} f(x) \, dx \approx \frac{2h}{45} (7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)).$$

In following we continue approximating for volterra fredholm integral equation:

$$f(x) = u(x) + \sum_{j=1}^{n} g_j(x)(\sum_{\alpha=1}^{4} 7h_j (y_{4\alpha-4}) u(y_{4\alpha-4})$$

$$+ 32h_j (y_{4\alpha-3}) u(y_{4\alpha-3}) + 12h_j (y_{4\alpha-2}) u(y_{4\alpha-2})$$

$$+ 32h_j (y_{4\alpha-1}) u(y_{4\alpha-1}) + 7h_j (y_{4\alpha}) u(y_{4\alpha}))$$

$$+ \sum_{i=1}^{m} m_i(x)(\sum_{\beta=1}^{4} 7n_i (y_{4\beta-4}) u(y_{4\beta-4})$$

$$+ 32n_i (y_{4\beta-3}) u(y_{4\beta-3}) + 12n_i (y_{4\beta-2}) u(y_{4\beta-2})$$

$$+ 32n_i (y_{4\beta-1}) u(y_{4\beta-1}) + 7n_i (y_{4\beta}) u(y_{4\beta}))$$

where

$$l = 4k''$$, $$k'' \in \mathbb{N}$$

$$v = \frac{x-a}{b-a}, \quad y_0 = a, \quad y_l = x$$

$$z = 4k', \quad k' \in \mathbb{N}$$

$$p = \frac{b-a}{4}, \quad y_0 = a, \quad y_z = b.$$}

4. **Fuzzy neural network operations on fuzzy volterra-fredholm integral equation.** To solving equation, we should fuzzificate the parameters and function in approximated form of volterra-fredholm integral equation. We define the fuzzy volterra-fredholm integral equation by left and right fuzzy bond and $h$-level set. We have:

$$[f(x)]_h^L = \left[ \int_{a}^{b} k_1(x, y) u(y) \, dy \right]_h^L + \left[ \int_{a}^{x} k_2(x, y) u(y) \, dy \right]_h^L$$

$$[f(x)]_h^R = \left[ \int_{a}^{b} k_1(x, y) u(y) \, dy \right]_h^R + \left[ \int_{a}^{x} k_2(x, y) u(y) \, dy \right]_h^R$$

so:

$$[f(x)]_h^L = \int_{a}^{b} [k_1(x, y)]_h^L [u(y)]_h^L \, dy + \int_{a}^{x} [k_2(x, y)]_h^L [u(y)]_h^L \, dy$$

$$[f(x)]_h^R = \int_{a}^{b} [k_1(x, y)]_h^R [u(y)]_h^R \, dy + \int_{a}^{x} [k_2(x, y)]_h^R [u(y)]_h^R \, dy$$

where $[f(x)]_h^L$ and $[f(x)]_h^R$ are left and right fuzzy polynomials by $h$-level and $[k_1]_h^L, [k_1]_h^R$ and $[k_2]_h^L, [k_2]_h^R$ are separable fuzzy kernels.
Some solution to this work done in some papers [6] but we follow fuzzification in our way to equation (1) by left and right fuzzy bonds and h-level set

\[
[f(x)]_h^L = [u(x)]_h^L + \sum_{j=1}^n [g_j(x)]_h^L \left( \frac{2}{35} \sum_{\alpha=1}^7 \tau [h_j(y_{4\alpha-4})]_h^L [u(y_{4\alpha-4})]_h^L \right.
+ 32[h_j(y_{4\alpha-3})]_h^L [u(y_{4\alpha-3})]_h^L
+ 12[h_j(y_{4\alpha-2})]_h^L [u(y_{4\alpha-2})]_h^L
+ 32[h_j(y_{4\alpha-1})]_h^L [u(y_{4\alpha-1})]_h^L
+ 7[h_j(y_{4\alpha})]_h^L [u(y_{4\alpha})]_h^L
\]

\[
[f(x)]_h^R = [u(x)]_h^R + \sum_{j=1}^n r[g_j(x)]_h^R \left( \frac{2}{35} \sum_{\beta=1}^7 \tau [h_j(y_{4\beta-4})]_h^R [u(y_{4\beta-4})]_h^R \right.
+ 32[h_j(y_{4\beta-3})]_h^R [u(y_{4\beta-3})]_h^R
+ 12[h_j(y_{4\beta-2})]_h^R [u(y_{4\beta-2})]_h^R
+ 32[h_j(y_{4\beta-1})]_h^R [u(y_{4\beta-1})]_h^R
+ 7[h_j(y_{4\beta})]_h^R [u(y_{4\beta})]_h^R
\]

We want to design a general solution to above equation. In ANN system the most important part is action function which can be named and appeared in “fuzzy training function” [12]. By training function, we can train three layered AANs’ algorithm and solve problem. Our train function is in the form of [29]:

\[ t_L(x, P) = \gamma(x) + \delta[x, N(x, P)] \]

where \( x \) is arbitrary input, \( t_L \) is a function by values \( x \) and \( P \) where \( P \) included weights and biases in the FANN layered structure. Fuzzy initial condition created by \( \gamma \) and \( \delta \) tuning input and output on \( N(x, P) \) in three layered feed forward FANN. We act the function on collection of points in arbitrary interval by set of discrete spaced grid points.

Here our FANN is 1*1*1 dimension and every neuron input \( x \) weighted by parameters to hidden neurons in hidden layer in network like:

\[ net_i = x.w_i + b_i \]

where \( b_i \) is a bias which is activate in hidden layer for f(x) estimating. The outputs from hidden layer are:

\[ z_i = A(net_i) \quad i = 1, ..., n \]

where A is activation function on network. Let \( V_i \) is a weight parameter between hidden layer and output layer. No changing appear by output neuron in its input so:

\[ N_i = V_iz_i \]
Output layer in FANN has combined form with right and left bond in h-level like [20]:

\[ [N_i]_h = [[N_i]_h^L, \ [N_i]_h^R] = [[v_i]_h^L, z_i, [v_i]_h^R, z_i]. \]

Learning of fuzzy weights is another subject that should be illustrate. Let fuzzy vector \( v = (v_1, \ldots , v_c) \) as a target, corresponding to fuzzy output vector \( o = (o_1, \ldots , o_c) \) in FANN. To learn of FANN, we offer a function \( e_h \) that measures the difference for h-level sets of target and output vector same components in the interval arithmetic FANN:

\[
e_h = \sum_{k=1}^c \left( \frac{[v_k]_h^L - [o_k]_h^L}{2} \right)^2 + \sum_{k=1}^c \left( \frac{[v_k]_h^R - [o_k]_h^R}{2} \right)^2
\]

where \([v_k]_h = [[v_k]_h^L, [v_k]_h^R]\) and \([o_k]_h = [[o_k]_h^L, [o_k]_h^R]\). The difference of vector \( v \) and \( t \) is computing by function \( e_h \) in the h-level sets, then we summing up \( e_h \) for various of \( h \). So [42]:

\[
e = \sum_h e_h = \sum_h h.e_h.
\]

To describe the proposed method, we bring up the following numerical example.

**Example:** Consider the following fuzzy volterra-fredholm integral equation

\[
f(x) = u(x) + \int_0^1 xyu(y)dy + \int_0^x x^2(y + 1)u(y)dy
\]

where

\[
f(x) = (0.5 + 0.5r, 2 - r)(e^x - \frac{x}{3} + x^3e^x - \frac{x^6}{3} - \frac{x^4}{2}).
\]

The trail function for this problem is \( t_L(x, p) = xN(x, p) \) and we use allocate followind properties to the learning algorithm:

1. Allocation of five hidden units
2. Selection 200 repetition for stopping condition of the learning algorithm
3. We use 0.1 and 0.2 to Learning and momentum constants: \( \eta = 0.1 \)

A comparison between the exact and approximate solution at \( x = 0.5 \) is shown in the following table:

| \( r \) | \( u_0 \) | \( u_1 \) | \( \text{ERROR} \) | \( u_0 \) | \( u_1 \) | \( \text{ERROR} \) | \( u_0 \) | \( u_1 \) | \( \text{ERROR} \) |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.109140 | 1.1091409142 | 9.14E-07 | 4.43656 | 4.436563569 | 3.66E-06 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.1 | 1.220025 | 1.2200258465 | 8.47E-07 | 4.214735 | 4.214735548 | 2.00E+00 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.2 | 1.330096 | 1.3300960992 | 9.09E-06 | 3.99291 | 3.992907276 | 2.72E-06 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.3 | 1.44218 | 1.441883183 | -2.97E-04 | 3.77108 | 3.771079094 | 9.06E-07 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.4 | 1.55279 | 1.552797274 | 7.27E-06 | 3.549251 | 3.549250912 | -8.80E-08 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.5 | 1.66371 | 1.663711365 | 1.36E-06 | 3.32742 | 3.327422273 | 2.73E-06 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.6 | 1.774625 | 1.774625456 | 4.66E-07 | 3.10559 | 3.105594548 | 4.55E-06 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.7 | 1.88553 | 1.885539547 | 9.55E-06 | 2.88376 | 2.883766366 | 6.37E-06 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.8 | 1.99645 | 1.996453638 | 3.64E-06 | 2.661862 | 2.66186184 | 1.60E-07 | 2.214735 | 2.214735458 | 9.52E-07 |
| 0.9 | 2.10736 | 2.107367729 | 7.73E-06 | 2.440110 | 2.440110002 | 2.00E-09 | 2.214735 | 2.214735458 | 9.52E-07 |
| 1 | 2.218282 | 2.21828182 | -1.80E-07 | 2.218282 | 2.21828182 | -1.80E-07 | 2.214735 | 2.214735458 | 9.52E-07 |
4.1. Conclusions. In this paper we follow up the method that lead to the best approximate of fuzzy volterra-fredholm integral equation in the rapid way. Less error and best approximating is the features of AFNN and can be extend to other type of integral and differential equation. We claim that the three layered feedforward AFNN is optimized to solve these kind of problems. We think Less complexity in designing network is lead to more flexibility in solving problems, but struggled with other complex equation by training more than three layered AFNN is avoidable.

Acknowledgments. The authors sincerely thank the anonymous reviewers for their careful reading, constructive comments and fruitful suggestions that improved the manuscript substantially.

REFERENCES

[1] G. Adomian, Solution of physical problems by decomposition, Comput. Math. Appl., 27 (1994), 145–154.
[2] M. Y. Ali, A. Sultana and A. Khan, Comparison of fuzzy multiplication operation on triangular fuzzy number, IOSR J. Math (IOSR-JM), 12, (2016), 35–41.
[3] A. B. Badiru and J. Cheung, Fuzzy Engineering Expert Systems with Neural Network Applications, Vol. 11, John Wiley & Sons, 2002.
[4] E. Balagurusamy, Computer Oriented Statistical and Numerical Methods, Macmillan India Limited, 1988.
[5] C. Bector and S. Chandra, Fuzzy numbers and fuzzy arithmetic, Fuzzy Mathematical Programming and Fuzzy Matrix Game, (2005), 39–56.
[6] S. S. Behzadi, Solving fuzzy nonlinear Volterra-Fredholm integral equations by using homotopy analysis and Adomian decomposition methods, J. Fuzzy Set Valued Anal., 2011 (2011), 1–13.
[7] H. Brunner, On the numerical solution of nonlinear Volterra-Fredholm integral equations by collocation methods, SIAM J. Numer. Anal., 27 (1990), 987–1000.
[8] A. E. Bryson and Y. C. Ho, Applied Optimal Control: Optimization, Estimation, and Control, Hemisphere Publishing Corp. Washington, D. C.; Distributed by Halsted Press [John Wiley & Sons], New York-London-Sydney, 1975.
[9] J. J. Buckley and Y. Hayashi, Can fuzzy neural nets approximate continuous fuzzy functions?, Fuzzy Sets and Systems, 61 (1994), 43–51.
[10] A. Cardone, E. Messina and A. Vecchio, An adaptive method for Volterra-Fredholm integral equations on the half line, J. Comput. Appl. Math., 228 (2009), 538–547.
[11] J. Dijkman, H. V. Haeringen and S. D. Lange, Fuzzy numbers, Journal of Mathematical Analysis and Applications, 92 (1983), 301–341.
[12] D. Dumitrescu, B. Lazzerini and L. C. Jain, Fuzzy Sets & their Application to Clustering & Training, CRC Press, 2000.
[13] R. Fullér, Introduction to Neuro-Fuzzy Systems, Advances in Soft Computing, Physica-Verlag, Heidelberg, 2000.
[14] I. Gohberg, S. Goldberg and M. A. Kaashoek, Classes of Linear Operator, Vol. 63, Birkhäuser, 2013.
[15] K. Gurney, An Introduction to Neural Networks, CRC Press, 2014.
[16] Y. Hayashi, J. J. Buckley and E. Czogala, Fuzzy neural network with fuzzy signals and weights, International Journal of Intelligent Systems, 8 (1993), 527–537.
[17] D. O. Hebb and D. Hebb, The Organization of Behavior, Vol. 65, Wiley New York, 1949.
[18] R. Hecht-Nielsen, Kolmogorov’s mapping neural network existence theorem, IEEE Press, 3 (1987), 11–14.
[19] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sci. U. S. A., 79 (1982), 2554-2558.
[20] H. Ishibuchi, K. Morioka and I. Turksen, Learning by fuzzified neural networks, International Journal of Approximate Reasoning, 13 (1995), 327–358.
[21] H. Ishibuchi, H. Tanaka and H. Okada, Fuzzy neural networks with fuzzy weights and fuzzy biases, IEEE International Conference on Neural Networks, (1993), 1650–1655.
[22] O. Kaleva, Fuzzy differential equations, Fuzzy sets and systems, 24(3)(1987), 301–317.
[23] J. M. Keller and D. J. Hunt, Incorporating fuzzy membership functions into the perceptron algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 (1985), 693–699.

[24] E. Khan and P. Venkatapuram, Neufuz: Neural network based fuzzy logic design algorithms, *Second IEEE International Conference on Fuzzy Systems*, (1993), 647–654, https://ieeexplore.ieee.org/abstract/document/327412.

[25] D. Kriese, *A Brief Introduction on Neural Network*, 2007. Available from: http://www.dkriesel.com/en/science.

[26] A. R. Krommer and C. W. Ueberhuber, *Computational Integration*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[27] C. T. Leondes, *Fuzzy Logic and Expert Systems Applications*, Vol. 6, Elsevier, 1998.

[28] T. Mach, *Eigenvalue Algorithms for Symmetric Hierarchical Matrices*, Thomas Mach, 2012.

[29] A. Malek and R. S. Beidokhti, Numerical solution for high order differential equations using a hybrid neural network-optimization method, *Appl. Math. Comput.*, 183 (2006), 260–271.

[30] K. Maleknejard and M. Hadizadeh, A new computational method for Volterra-Fredholm integral equations, *Comput. Math. Appl.*, 37 (1999), 1–8.

[31] J. McCarthy, *The dartmouth college artificial intelligence conference: The next fifty years*, *Ai Magazine*, 27 (2006), 87–87.

[32] F. Mora-Camino and C. A. N. Cosenza, Fuzzy dual numbers, in fuzzy dual numbers, Springer, (2018), 11–16.

[33] A. L. Samuel, Some studies in machine learning using the game of checkers, *IBM J. Res. Develop.*, 3 (1959), 211–229.

[34] V. Raju and R. Jayagopal, An arithmetic operations of icosaogonal fuzzy number using Alpha cut, *International Journal of Pure and Applied Mathematics*, 120 (2018), 137–145.

[35] P. Sibi, S. A. Jones and P. Siddarth, Analysis of different activation functions using back propagation neural networks, *Journal of Theoretical and Applied Information Technology*, 47 (2013), 1264–1268.

[36] R. Subramanian, Emergent AI, Social Robots and the Law: Security, Privacy and Policy Issues, *Journal of International, Technology and Information Management*, 26 (2017).

[37] E. A. Wan, *Time Series Prediction by using a Connectionist Network with Internal Delay Lines*, Addison-Wesley Publishing Co, 1993.

[38] A.-M. Wazwaz, *A First Course in Integral Equations*, 2nd edition, World Scientific Publishing Company, Co. Pte. Ltd., Hackensack, NJ, 2015.

[39] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph. D. dissertation, Harvard University, 1975.
[53] J. Yuan and S. Yu, Privacy preserving back-propagation neural network learning made practical with cloud computing, *IEEE Transactions on Parallel and Distributed Systems*, 25 (2013), 212–221.

[54] L. A. Zadeh and F. sets, Fuzzy sets, *Information and Control*, 8 (1965), 338–353.

Received March 2021; revised June 2021; early access August 2021.

E-mail address: hadiabtahi@gmail.com
E-mail address: rahimi@iauctb.ac.ir
E-mail address: maryammosleh79@yahoo.com