Resolution of Minimal Solutions for Max-Lukasiewicz Fuzzy Relation Equation by Fuzzy Neural Network

Kaiyan Zhou, Xinyi Liang, Wenyi Zeng and Qian Yin*

School of Artificial Intelligence, Beijing Normal University, Beijing, 100875, PR China

*Corresponding author e-mail: yinqian@bnu.edu.cn

Abstract. Fuzzy relation equation (FRE) was introduced by Sanchez in 1976, which was extensively applied in fuzzy logic, approximate reasoning, intelligent control, artificial intelligence and so on. It is well known that solving the minimal solutions of FRE is a NP-hard problem because of its computational complexity. Many scholars have studied some algorithms to solve the minimal solutions of fuzzy relation equation. Considering that max-lukasiewicz composition operator is widely used, in this paper, we investigate the resolution of minimal solutions for max-lukasiewicz fuzzy relation equation based on fuzzy neural network and propose a novel algorithm. Through continuous downward iteration and conditional constraints, all minimum solutions of FRE are finally obtained. Both mathematical conclusion and simulation experiment show that our algorithm is effective and valid.

Keywords: Fuzzy Relation Equation, Fuzzy Neural Network, Max-Lukasiewicz, Minimal solution

1. Introduction

Fuzzy relation equation (FRE) was extensively applied in many fields, such as fuzzy decision, fuzzy control, transport system, data compression, fault diagnosis and medical diagnosis and so on[1,2,3,4].

FRE based on the max-min composition was introduced by Sanchez [5] in 1976. Aimed at the maximum solution of FRE, some scholars obtained some meaningful conclusion [6,7]. And for the minimal solutions of FRE, many researchers investigated it and proposed some different algorithms [8-11]. Considering the diversity of the minimal solution, thus the resolution of minimal solution for FRE is a NP-hard problem.

In 1994, Fuzzy neural networks were introduced to solve FRE [12]. Li and Ruan [13,14] proposed some novel neural algorithms to solve the maximum solution of FRE based on fuzzy δ rule. Aimed at that max-lukasiewicz operator has been widely applied in many fields, in this paper, inspired by Li and Ruan’s δ rule, we investigate the minimal solutions of FRE with max-lukasiewicz operator by using fuzzy neural network, and propose a novel algorithm.

The rest of our work is organized as follows: in section2, we briefly recall some notions of FRE and fuzzy neural network. In Section 3, we propose a novel algorithm for solving the FRE. In Section 4, we propose some related mathematical conclusions and investigate the convergence of our proposed
algorithm. In Section 5, we give the simulation experiment of the algorithm. The conclusion is given in the last section.

2. Preliminaries
To better understand the introduced algorithm, in this section we will give some basic definitions.

2.1. Fuzzy Operator
Definition 1. Lukasiewicz-norm [15]: \( f_{boun}(a,b) = \max\{0,a + b - 1\} \) is a binary operation t-norm(triangular norm), and denote by \( \ast \) in this paper.

Definition 2. max-conorm [15]: \( g_{\max}(a,b) = \max\{a,b\} \) is a t-conorm operation, and denote by \( \lor \) in this paper.

2.2. Fuzzy Relation Equation
Definition 3. Binary fuzzy relation[5]: Let \( X \) and \( Y \) be nonempty set, a binary fuzzy relation \( R \) between \( X \) and \( Y \) is a mapping \( R:X \times Y \rightarrow [0,1] \) is a fuzzy set in the universe \( X \times Y \), i.e. \( R \in [0,1]^{X \times Y} \).

Definition 4. Fuzzy relation equation[5]: Given the deterministic domain \( X, Y, Z \), given the binary fuzzy relation \( A \in [0,1]^{Y \times Z}, B \in [0,1]^{X \times Z} \), find the fuzzy relation \( W \in [0,1]^{X \times Y} \) satisfies the equation

\[
W \circ A = B\tag{1}
\]

where \( \circ \) in equation represents the fuzzy composition operator, it is composed of t-norm and t-conorm in definition 1 and 2 respectively, that is

\[
W \circ A(x,y) = \bigvee_{y \in Y} (\max\{W(x,y) + A(y,z) - 1, 0\})\tag{2}
\]

2.3. Fuzzy Neural Network
Here, we give a common neural network in Figure.1. If we set the input node as each sample of \( A \), ignore all hidden layers, output layer as each sample of \( B \), and change the \( \times \) and \( + \) of the intermediate operator into “ \( \ast \)” and “ \( \lor \)” respectively, then we will require \( W \) to be the weight of the neural network.

![Figure 1. A common neural network](image)

3. Novel Algorithm
In this section we will present the objective and specific steps of the algorithm.

3.1. The Objective
Assume the fuzzy relational equation is Eq.(1), that is \( W \circ A = B \), where \( \circ = (\lor, \ast) \).

\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{p1} & a_{p2} & \cdots & a_{pn}
\end{bmatrix}
\]
\[ W = (w_1 \ w_2 \ \cdots \ w_n) \]
\[ B = (b_1 \ b_2 \ \cdots \ b_n) \]

Here, we call the relationship \( A \) and relationship \( B \) as \( n \) different training samples \((a_1,b_1),(a_2,b_2),\cdots,(a_n,b_n)\), where \( a_j = (a_{1j},a_{2j},\cdots,a_{nj})^T (j = 1,2,\cdots,n) \) are fuzzy vectors, and \( b_1,b_2,\cdots,b_n \) are the expected outputs samples, respectively. And our objective is to train these samples in turn through the neural network to adjust the value of weight \( W \).

3.2. Algorithm

a. Initializing
\[ W(0) = (w_{10}(0),w_{20}(0),\cdots,w_{p0}(0)) = (0,0,\cdots,0), \quad U(0) = \{W(0)\}, \quad \text{where} \quad U(0) \quad \text{represents} \quad \text{the} \quad 0^{\text{th}} \quad \text{set} \quad \text{of} \quad \text{all} \quad \text{weight} \quad \text{vectors}, \quad W(0) \quad \text{is} \quad \text{the} \quad \text{Initial} \quad \text{weight}. \]

\[ W = (\overline{w_1},\overline{w_2},\cdots,\overline{w_p}) \] is the maximum solution of Eq.(1).

b. For \( j=1 \) to \( n \):
Input the \( j^{\text{th}} \) sample: \((a_{1j},a_{2j},\cdots,a_{pj})^T \) is the \( j^{\text{th}} \) input pattern and \((b_j)\) is the \( j^{\text{th}} \) output pattern. The result of \( j-1^{\text{th}} \) is \( U(j-1) = [W_s(j-1),W_s(j-1),\cdots,W_s(n)] \). For every \( W_s(j) = (w_{s_1}(j),w_{s_2}(j-1),\cdots,w_{s_p}(j-1)) (s=1,2,\cdots,k_j-1) \) in \( U(j-1) \):
Step1. Calculating the actual outputs:
\[ (b_j)' = \sqrt[p]{\max_k \{w_{s_k}(j-1) + a_{kj} - 1,0\}} \quad \text{(3)} \]
where \((b_j)'\) represents the actual output when the \( j^{\text{th}} \) data pair is being trained, and \( w_{s_k}(j-1) \) is the \( j-1^{\text{th}} \) connection weight from the \( k^{\text{th}} \) input node to the output node, and \( a_{kj} \) is the \( k^{\text{th}} \) component of the input pattern.
Step2. \((b_j)\) has several cases:
- Case1. \((b_j) = b_j\), then \( W(j) = W_s(j-1) \), and add \( W(j) \) to \( U(j) \).
- Case2. When \( j > 1 \), if \((b_j) > b_j\), then \( W_s(j-1) \) will not enter \( U(j) \), because it will not be the answer for Eq.(1).
- Case3. \((b_j) < b_j\), then \( I_{1j} = \{i_1,i_2,\cdots,i_{p_j}\} = \{i | a_{ij} + \overline{w}_i + 1 \geq b_j, i = 1,2,\cdots,p\} \),
where \( i_1 < i_2 < \cdots < i_{p_j} \) obviously \(|I_{1j}| = p_j \leq p \), then we adjust \( W_s(j-1) \) in \( p_j \) ways.

for \( r \) in \( I_{1j} \):
Let \( Y_r(j-1) = W_s(j-1), W_r(j) = W_s(j-1) \), adjusting \( y_{i1}(j-1) \) in \( Y_r(j-1) \):
- A. Calculate the Error \( E_i = |(b_j)' - b_j| \).
- B. If \( E_j < \varepsilon \), then compare \( Y_r(j-1) \) with the solutions of \( U(j-1) \) except \( W_s(j-1) \), and if \( \exists W_r(j-1) \in U(j-1) \backslash W_s(j-1), W_r(j-1) \leq Y_r(j-1) \), then discards \( Y_r(j-1) \).
- Otherwise change \( W_r(j) \), make \( w_{i1}(j) = y_{i1}(j-1) \), and add \( W_r(j) \) to \( U(j) \), then go to next way.

Else go to C.
- C. \( y_{i1}(j-1) = y_{i1}(j-1) + \eta E_i \), if \( y_{i1}(j-1) > \overline{w}_i \), move on the next way, else calculate \( (b_j)' = \sqrt[p]{\max_k \{y_{i1}(j-1) + a_{kj} - 1,0\}} \) and return to the step A.

where \( \varepsilon \) represents the margin of error that we can accept, and \( \eta \) represents learning rate or step length.

c. When \( j = n \), we can get the result \( U(n) = \{W_1(n),W_2(n),\cdots,W_k(n)\} \).

4. Theoretical Results

**Theorem 1.** \( \forall W \in U, \{W(t) = (w_t)_{1xp}\} \) is a monotone increasing sequence.

**Proof.** Because the absolute error \( E_j = |(b_j)' - b_j| \geq 0, \forall E \in (0,1) \), then
(i) When \( i = k_{j_0} \), \( w_i(t + 1) = w_i(t) + \eta \tilde{e}_j \geq w_i(t) \).
(ii) When \( i \neq k_{j_0} \), \( w_i(t + 1) = w_i(t) \).
(iii) In addition, if \( w_i(t + 1) > \overline{w_i} \) (\( \overline{w_i} \) is the \( i^{th} \) component of the maximum solution \( \overline{W} \)), for \( \forall i: w_i(t + 1) = \overline{w}_i \).

Known by (i),(ii) and (iii) we always have \( \forall W \in U \), \( \{W(t)\} \) is a monotone increasing sequence.

**Theorem 2.** \( \forall W \in U \), \( \{W(t) = (w_i)_{1 \leq i \leq p}\} \), \( \{W(t)\} \) in the algorithm of section 3 is surely convergent.

**Proof.** Know by Theorem 1, the training sequence \( \{W(t)\} \) is monotone ascending, and we also know \( \{W(t)\} \) is a bounded sequence, that is, \( 0 \leq W(t) \leq W \). Where \( W \) is the maximum solution of Eq.(1), and \( O \) is a matrix with all components being 0. So \( \forall W \in U \), \( \{W(t)\} \) is surely convergent.

**Theorem 3.** \( \forall W \in U \), \( W \) is the solution of Eq.(1).

**Proof.** \( \forall W \in U \), known by step C in the section 3, we have \( W \leq W \), then \( W \circ A \leq W \circ A = B \), and now we need to prove that this equals sign holds. If this is not true, then \( \exists j_0 \in \{1,2,\cdots,n\} \) makes that \( V^p_k = \{\max \{w_i + a_{ij_0} - 1,0\} < b_{j_0} \} \). Since \( W \) is the maximum solution to Eq.(1), \( W \circ A = B \), there must be \( i_0 \) to make \( a_{ij_0} + \overline{w}_{i_0} + 1 \geq b_{j_0} \). In the process of algorithm iteration, there must be a generation of solution \( W(j_0) \), where one element \( w_{i_0}(j_0) \) satisfies \( \max \{w_{i_0}(j_0) + a_{ij_0} - 1,0\} = b_{j_0} \).

Known by Theorem 1, the training sequence \( \{W(t)\} \) is monotone increasing, so \( W \geq W(0) \), \( w_{i_0} \geq w_{i_0}(j_0) \), then we have \( V^p_k = \{\max \{w_i + a_{ij_0} - 1,0\} > b_{j_0} \} \) \( \forall W \in U \), \( \{W(t)\} \) is surely convergent.

**Theorem 4.** \( \forall W \in U \), if \( W \) is not the minimal solution, there exists a \( V = (v_1,v_2,\cdots,v_p) \) \( < W \) is the minimal solution to Eq.(1), \( \exists i_0 \in \{1,2,\cdots,p\} \), \( v_{i_0} < w_{i_0} \). Assume that the solution generated by the algorithm in the section 3 in each iteration is \( W(0),W(1),\cdots,W(n) = W \).

Known by \( 0 \leq v_{i_0} < w_{i_0} \) and \( w_{i_0}(0) = 0 \), we have: \( \exists j_0 \), such that \( w_{i_0}(j_0) > w_{i_0}(j_0 - 1) \), and there has \( w_{i_0}(t) = w_{i_0}(j_0) \forall t \in \{j_0 + 1,\cdots,n\} \), then we have \( \max \{w_{i_0}(j_0 - 1) + a_{ij_0} - 1,0\} < b_{j_0} \), \( \max \{w_{i_0}(j_0) + a_{i_0} - 1,0\} = b_{j_0} \).

Suppose \( W_{i_0}(j_0) = \begin{cases} v_{i_0} & i = i_0 \\ w_{i_0} & i \neq i_0 \end{cases} \)

Then we have \( V \leq W(j_0) \leq W(0) \leq W \). Since \( W(j_0 - 1) \circ A < b_{j_0} \) \( w_{i_0}(j_0) = w_{i_0}(j_0 - 1) \) \( (i \neq i_0) \), then \( V^p_k = \{\max \{w_i(j_0) + a_{ij_0} - 1,0\} < b_{j_0} \} \). Therefore, we have, \( V \circ A < W(j_0) \circ A < W \), then \( Y \) is not the solution of Eq.(1).

**Theorem 5.** The solutions in \( U \) are all minimal solutions to Eq.(1).

**Proof.** When \( \eta = 1 \), this algorithm is going to be similar to Li's[16] algorithm, and to sum up, the conclusion is valid.

5. **Simulation Experiments**

Example 1. Suppose \( FRE \ W \circ A = B \) where \( \circ \) is max- lukasiewicz operator.

\[
A = \begin{pmatrix}
0.8 & 0.9 & 0.2 & 0.3 \\
0.1 & 0.7 & 0.8 & 0.1 \\
0.7 & 0.2 & 0.9 & 0.0 \\
0.9 & 0.8 & 0.7 & 0.9 \\
\end{pmatrix}
\]

\[
B = \begin{pmatrix}
0.5 & 0.6 & 0.7 & 0.0 \\
0.7 & 0.9 & 0.8 & 0.1 \\
0.7 & 0.0 & 0.8 & 0.0 \\
0.0 & 0.9 & 0.8 & 0.0 \\
\end{pmatrix}
\]

The maximum solution of this FRE is \((0.7 \ 0.9 \ 0.8 \ 0.1) \), then when training them with the algorithm in section 3,2, we can get all the minimal solutions are \( U = \{(0.7 \ 0.9 \ 0.0 \ 0.0) , \ (0.7 \ 0.0 \ 0.8 \ 0.0) , (0.0 \ 0.9 \ 0.8 \ 0.0) \} \).
It is easy to verify that the every solution in satisfies equation $W^*A = B$, furthermore they are the whole minimal solutions.

Now Suppose $t$ represents the number of iteration steps when a stable point arrives, then $t$ is related to the size of $\eta$. The results are shown in Table 1, and $t_i$ represents the iteration of example 1. We find that the number of iterations increased with the decrease of $\eta$.

Table 1. The relation between $\eta$ and $t$ for max-lukasiewicz FRE

| IDX | $\eta$ | $t_i$ | IDX | $\eta$ | $t_i$ |
|-----|--------|-------|-----|--------|-------|
| 1   | 1.0    | 12    | 6   | 0.5    | 183   |
| 2   | 0.9    | 60    | 7   | 0.4    | 246   |
| 3   | 0.8    | 82    | 8   | 0.3    | 351   |
| 4   | 0.7    | 108   | 9   | 0.2    | 557   |
| 5   | 0.6    | 138   | 10  | 0.1    | 1175  |

6. Conclusion
In this paper, we investigate the minimal solutions of fuzzy relation equation and propose a novel algorithm base on neural network where the operator is max-lukasiewicz operator, we can obtain all minimum solutions of the FRE through automatic iteration process. Finally, we give some theoretical mathematical results and simulation experiment to illustrate that our algorithm is effective and valid.

Acknowledgments
The research work is supported by the grant from Joint Research Fund in Astronomy (U2031136) under cooperative agreement between the NSFC and CAS. Prof. Qian Yin and Prof. Wenyi Zeng are the authors to whom all correspondence should be addressed.

References
[1] A. Di Nola, S. Sessa, W. Pedrycz, E. Sanchez, Fuzzy relation equations and their applications to knowledge engineering, Vol. 3, Springer Science & Business Media, 2013.
[2] H. Mai, B.-y. Cao, X.-G. Zhou, The application of fuzzy relational equations and genetic algorithm in fault diagnosis problem, in: B.-y. Cao (Ed.), Fuzzy Information and Engineering-2019, Springer Singapore, Singapore, 2020, pp. 197–205.
[3] Ferdinando Di Martino, Salvatore Sessa, Digital Watermarking Strings with Images Compressed by Fuzzy Relation Equations. Springer Berlin Heidelberg, 2013.
[4] A. P. Rotshtein, H. B. Raktytanska, Fundamentals of Intellectual Technologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 1–37.
[5] E. Sanchez, Resolution of composite fuzzy relation equations, Information and control 30 (1) (1976) 38–48.
[6] M. Higashi, G. J. Klir, Resolution of finite fuzzy relation equations, Fuzzy Sets and Systems 13 (1) (1984) 65–82.
[7] G. Klir, B. Yuan, Fuzzy sets and fuzzy logic, Vol. 4, Prentice hall New Jersey, 1995.
[8] J. C. Diaz-Moreno, J. M. E. T, Minimal solutions of general fuzzy relation equations on linear carriers, an algebraic characterization, Fuzzy Sets and Systems 311 (2017) 112–123.
[9] Xiao-Peng Yang, Resolution of bipolar fuzzy relation equations with max-Łukasiewicz composition, Fuzzy Sets and Systems, Volume 397, Pages 41-60, ISSN 0165-0114 (2020).
[10] B.-S. Shieh, Solution to the covering problem, Information Sciences 222 (2013) 626–633, including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems.
[11] J. C. D’iaz-Moreno, J. Medina, Using concept lattice theory to obtain the set of solutions of multi-adjoint relation equations, Information Sciences 266 (2014) 218–225.
[12] M. Gupta, D. Rao, On the principles of fuzzy neural networks, Fuzzy Sets and Systems 61 (1) (1994) 1–290.18.
[13] X. Li, D. Ruan, Novel neural algorithms based on fuzzy $\delta$ rules for solving fuzzy rela-tion
equations: Part 295 i, Fuzzy Sets and Systems 90 (1) (1997) 11–23.
[14] X. Li, D. Ruan, Novel neural algorithms based on fuzzy δ rules for solving fuzzy rela-tion
equations: Part iii, 300 Fuzzy Sets and Systems 109 (3) (2000) 355–362.
[15] M. B, B. J, Fuzzy implications, series:Studies in fuzziness and soft computing (2008).
[16] Yanping Li, Hongxing Li, Fuzzy relational equation solution based on neural network, Journal
of Beijing Normal University (Natural Science),1998.(in Chinese)