INTELLIGENT AUTOMATION & SOFT COMPUTING, 2017
https://doi.org/10.1080/10798587.2017.1328812

AutoSoft®
Taylor & Francis
Taylor & Francis Group

Check for updates

Numerical Solution of Linear Regression Based on Z-Numbers by Improved Neural Network

Somayeh Ezadi[a] and Tofigh Allahviranloo[b]

[a]Department of Applied Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran; [b]Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran

ABSTRACT

In this article, the researcher at first focuses on introducing a linear regression based on the Z-number. In this regression, observations are real, but the coefficients and results of observations are unknown and in the form of Z-rating. Therefore, to estimate this type of regression, we have three distinct ways depending on different conditions dominating the problem. The three methods are a combination of artificial neural networks and fuzzy generalized improvements of the technique. Moreover the method of calculating the weights of the Z-number neural network has been mentioned and the stability of neural network weights is considered. In some examples, the answer is estimated compared with the original answer.

KEYWORDS
Z-numbers; linear regression; neural network; error analysis; weights; stability

1. Introduction

Regression analysis is one of the basic tools of scientific research that enables the identification of the performance of dependent and independent variables. In the analysis of the classic regression, both dependent and independent variables are real numbers. However, in many real conditions of life, where the complexity of the physical system has been dictated, adopting a more holistic view is required where the regression variables are given as non-numerical constructs such as linguistic variables (Cheng & Lee, 2001). Unfortunately, most of the real conditions in life are out of the classic regression analysis range (Bardossy, 1990; Bardossy, Bogardi, & Duckstein, 1990). After introducing the concept of fuzzy sets by Zadeh in 1965 (Zadeh, 1965, 1979, 1996), different researchers have developed the regression analysis. Fuzzy linear regression (FLR) was first suggested by Tanaka and colleagues (Tanaka, 1987), that is the extension of classic regression analysis that has turned into a powerful tool for the discovery of ambiguous relationships (Coppi, 2008). Indeed, in fuzzy regression, some of the elements of the regression model have been presented with ambiguous information. Different methods have been presented for solving these types of problems (Kao & Chyu, 2003; Modarres, Nasrabadi, & Nasrabadi, 2005; Mosleh, Allahviranloo, & Otadi, 2012; Mosleh, Otadi, & Abbasbandy, 2011; Tanaka, Havashi, & Watada, 1989). Among these methods is the use of the method of least squares for the study and fitness of fuzzy regression models that were first presented by Celmins (1987) and Diamond (1987). But, for this useful information, it must be reliable. Humans have a clear capacity for making logical decisions based on ambiguous, imprecise or incomplete information. Formalization of this capacity is at least to some degree a challenge that is estimated with difficulty. Zadeh proposed a subject called Z-number that has two components $\tilde{A}, \tilde{B}$ (Zadeh, 2011). The first component, $\tilde{A}$, is a restriction (constraint) on the values, which a real-valued uncertain variable, X, is allowed to take. The second component, $\tilde{B}$, is a measure of reliability (certainty) of the first component. Typically $\tilde{A}$ and $\tilde{B}$ are described in a natural language. For example: (About 45 min. is very reliable). Yager (2012) used Z-number to calculate the waiting time for the bus. Kang, Wei, Li, and Deng (2012) used the Z-number for making decision in an ambiguous environment. Regression analysis of the Z-number is an extension of the fuzzy regression analysis that some elements in this model have been shown with ambiguous information, which consist of a reliability degree and this type of regression has been introduced in this article. On the other hand, the artificial neural network is a powerful tool in estimating many functions, especially fuzzy regression (Mosleh et al., 2011, 2012). The neural network used in this article is a generalized neural system that is introduced later in this study.

The present article consists of following sections: In Section 2, the basic and required concepts have been stated. In Section 3, the Z-number neural network has been introduced. In Section 4, the estimate of ZLR regression has been stated using Z-number neural network. In Section 5, the analysis of recommended method error has been stated. In Section 6, the method of calculating the weights of Z-number neural network has been mentioned. In Section 7, the stability of neural network weights. In Section 8, a numerical example is presented. In Section 9, the conclusion is presented and in Section 10, the references are listed.

2. The Basic and Required Concepts

Definition 1: The parametric form of a fuzzy number that is introduced with a regular pair from function in the form of $(a_1(\alpha), b_1(\alpha)), 0 \le \alpha \le 1$, consists of following conditions:

$a_1(\alpha)$ is a bounded increasing function and is continuous on interval [0, 1] from right.

CONTACT Tofigh Allahviranloo tofigh@allahviranloo.com
© 2017 TSI® Press

$b_1(\alpha)$ is a descending bounded function and is continuous on interval [0, 1] from left.

Definition 2: If A and B be fuzzy numbers with $[A]_\alpha = [a_1(\alpha), a_2(\alpha)]$. $[B]_\alpha = [b_1(\alpha), b_2(\alpha)]$ and $\alpha \in [0,1]$, then fuzzy operation between them are defined as follows (Zimmermann, 1991):

$$[A+B]_\alpha = [a_1(\alpha) + b_1(\alpha), a_2(\alpha) + b_2(\alpha)],$$

$$[-A]_\alpha = [-a_2(\alpha), -a_1(\alpha)],$$

$$[A-B]_\alpha = [a_1(\alpha) - b_2(\alpha), a_2(\alpha) - b_1(\alpha)],$$

$$[\lambda A]_\alpha = [\lambda a_1(\alpha), \lambda a_2(\alpha)], \lambda > 0,$$

$$[\lambda A]_\alpha = [\lambda a_2(\alpha), \lambda a_1(\alpha)], \lambda < 0.$$

Definition 3: (Xu & Li, 2001) The distance between two fuzzy numbers of A and B based on weight function of $f(\alpha)$ is defined as:

$$d(A,B) = \left[\int_0^1 f(\alpha) d^2(A_\alpha, B_\alpha) d_\alpha\right]^{\frac{1}{2}},$$

$$d^2(A_\alpha, B_\alpha) = [a_1(\alpha) - b_1(\alpha)]^2 + [a_2(\alpha) - b_2(\alpha)]^2$$

Where index α is the α-cut, A and B are two fuzzy numbers, A and B are two fuzzy numbers, $A_\alpha = [a_1(\alpha), a_2(\alpha)]$ and $B_\alpha = [b_1(\alpha), b_2(\alpha)]$ are cuts of A and B, respectively. $f(\alpha)$ is an increasing function on the interval [0, 1] for which we have, $f(0) = 0$ and $\int_0^1 f(\alpha)d\alpha = \frac{1}{2}$. The amount $d(A_\alpha, B_\alpha)$ measures the distance between $\alpha - cut$ of fuzzy numbers from A and B. $f(\alpha)$ could be interpreted as weight of $d^2(A_\alpha, B_\alpha)$.

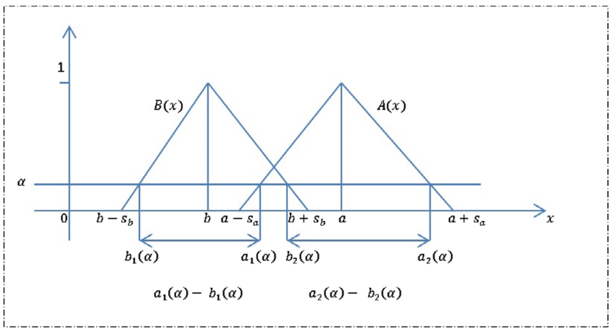
Definition 4: (Xu & Li, 2001) Let A be a symmetric trianggolar fuzzy number in LR form (a, s_a) such that a is center and s_a is width.

We write $A = (a, s_a^L, s_a^R)_T$, a is the center of A. s_a^L and s_a^R are the left and right widths, respectively. In a special case, if $s_a^L = s_a^R$, the A is called the symmetric fuzzy and we write $A = (a, s_a)_T$.

Theorem 5. Suppose $A = (a, s_a)_T$ and $B = (b, s_b)_T$ be two symmetric fuzzy numbers, then based on weight function $f(\alpha) = \alpha$, we have

$$d^2(A,B) = (a-b)^2 + \frac{1}{6}(s_a - s_b)^2$$

Proof: (Mohammadi & Taheri, 2004).



Definition 6: Definition of the Z-number

Zadeh (2011) introduced the Z-number along with an unknown variable X. A Z-number has two components (A, B). The first component, $\tilde{A}$, is a restriction (constraint) on the values, which a real-valued uncertain variable, X, is allowed to take. The second component, $\tilde{B}$, is a measure of reliability (certainty) of the first component. Zadeh introduced (x, A, B) as Z-valuation and showed that this amount is equal to this, that X is equal to (A, B). Here Z provides information about the value of variable X. An example of this valuation of Z is as follows:

Example: (about 45 min., very sure), (about 30 min., sure). This valuation for Z according to proposal of Zadeh is observed as a restriction in X and is interpreted as follows:

$$P(X \text{ is } A) \text{ is } B$$

Indeed, it means that

$$R(y): y \text{ } is \text{ } A \rightarrow Poss(y = u) = \mu_A(u)$$

$$P(y \text{ } is \text{ } A) = \int_R \mu_A(u) P_y(u) du \text{ } is \text{ } B$$

Where μ_A is the membership function of fuzzy set A and u is a part of y. $P_y(u)$ is the probability density function of y and $P(y = u)$ is the probability function of y. Whereas we do not know the basic probability distribution, it is clear from this information that the probability distribution function is itself a fuzzy number.

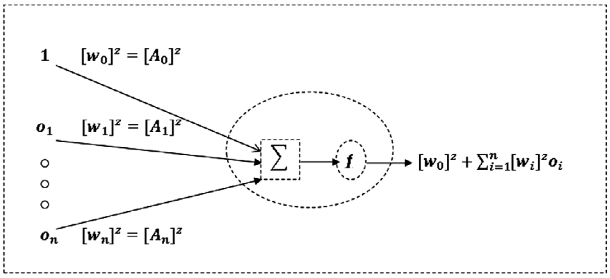
3. Z-number Neural Network (ZNN)

Although artificial neural networks are not comparable with the natural neural system, they have specifications where there is a need to learning linear mapping and nonlinear mapping, they are distinguished. Among the specifications of the artificial neural network they are ability to learn, distribution of information, generalizability, parallel processing (high speed) and being durable (ability to be repaired, error acceptance). A type of artificial neural network is the feed forward artificial neural network. This type of technology was first introduced by Rosenblatt (1962). This technology has for many years been widely used in different fields. In this article, a feed forward artificial neural network that is of a double-layer type is used. The first layer consists of inputs and the second layer forms the outputs with a linear transfer function. This network could be used for prediction, identification, and classification of the paradigm. In this study, the learning algorithm used for teaching of the artificial neural network is the algorithm after error dispersion. In designing the neural network, the section of the number of inputs is of special importance, because each input pattern consists of important information on the structure of the correlation itself and the complex structure of the data. To obtain input nodes, most researchers have used the error and trial method in this article, the number of input nodes is exactly equal to the number of unknowns in the problem, that is, the coefficients of regression and the number of nodes differs depending on the type of the problem.

Z-number neural network is defined with symbol $[Net]z$ as follows:

$$[Net]^z = (A, B)$$
$$[Net]^z = [w_0]^z + \sum_{i=1}^{n} [w_i]^z o_{ii}, \quad (1)$$

Where [.]z is the valuation symbol of Z-number, o_i is the input of the neural system that is of Z-number type and corresponding x_{ij}. $[w_i]^z$ is the weights of neural network with Z-number value.



Where f is the transformation function. For the case where the relationship between the first component restriction and the second component (the confidence limits) is not clear, the ZNN neural network cannot directly be used, but this neural network must be turned into two fuzzy neural networks. The first fuzzy neural network is related to the first section and the second one is related to the second section. The inputs of the neural network are common, but its outputs are distinct. The output of the first fuzzy neural network according to the first section of the variable is unknown and the output of the second fuzzy neural network based on the second section of the variable is unknown too. Hence, ZNN neural network consists of two independent target functions where the instruction method of both functions based on total squares of error have been defined according to distance in Theorem 2.5 and are independent from each other. This means that for solving problems where the relationship between the first and second components of the given numbers is not known, a Z-number neural network with a common function target cannot be used for real inputs and Z-number outputs.

4. Z-number Linear Regression (ZLR) Estimates using Neural Network

Consider the general model of ZLR as follows:

$$[\mathrm{Y_i}]^z = \left(A_Y, B_Y\right) \tag{2}$$

Where A_Y plays the role of restriction for $\mathrm{Y_i}$ and B_Y the role of degree of confidence and both of them have fuzzy values and have been defined as follows:

$$\begin{aligned} A_Y &= \left(\mathrm{a_0} + \mathrm{a_1 x_{i1}} + \mathrm{a_2 x_{i2}} + \cdots + \mathrm{a_n x_{in}}\right), \\ B_Y &= p\left(\mathrm{Y_i} is A_Y\right), \end{aligned} \tag{3}$$

$\mathrm{x_{i1}} \in \mathrm{R}, \mathrm{a_0}, \mathrm{a_1}, \mathrm{a_2}, \ldots, \mathrm{a_n}$ are fuzzy numbers and $[\mathrm{Y_i}]^z$ is Z - number.

We limit the discussion to the case where A_Y and B_Y are symmetric triangular fuzzy numbers. Since in the real world, ZLR regression given the different conditions (problem information), can have different states and in this section, we study and estimate some of those states.

4.1. First state

Suppose that $[\mathrm{Y}]^z$ in (2) consists of around $\mathrm{A_Y}$ value and density function $\mathrm{f}(X) = \lambda e^{-\lambda X}, \alpha \le \lambda \le \beta, X > 0$ (where α and β are fixed numbers and $\alpha < \beta$). The probability that Y, be $\mathrm{A_Y}$, it consists of $\mathrm{B_Y}$. That is, with the assumption that $\mathrm{A_Y} = (a, b)$ and $\mathrm{B_Y} = (c, d)$ where a and c be the center, b and d the fuzzy widths, $\mathrm{A_Y}$ and $\mathrm{B_Y}$ respectively. So we have

$$\begin{aligned} p(a \le Y \le b) &= \int_a^b \mathrm{f}(X)dX = \int_a^b \lambda e^{-\lambda X} DX \\ &= -e^{-\lambda X}\big|_a^b = e^{-\lambda a} - e^{-\lambda b} \end{aligned} \tag{4}$$

Where for $\alpha \le \lambda \le \beta$, in (4) will be as follows:

$$p(a \le Y \le b) = \begin{cases} c = e^{-\lambda_1 a} - e^{-\lambda_1 b}, \lambda_1 = \alpha \\ d = e^{-\lambda_2 a} - e^{-\lambda_2 b}, \lambda_2 = \beta \end{cases} \tag{5}$$

Where degree of membership for the above relation is introduced with $G(p)$ and is as follows:

$$G\left(p\right) = B\left(p(a \le Y \le b)\right) = \begin{cases} 1, & c \le e^{-\lambda a} - e^{-\lambda b} \le d \\ & 0, other \end{cases}$$

The objective based on observations in the form of

$$\left([\mathrm{y_i}]^z, \mathrm{x_{i1}}, \mathrm{x_{i2}}, \ldots, \mathrm{x_{in}}\right), \mathrm{i} = 1, \ldots, \mathrm{n}.$$

To obtain an optimal model with fuzzy co-efficients for describing and analyzing the data and predicting based on it where $\mathrm{x_{ij}}$ are real numbers and $[\mathrm{y_i}]^z$ are of Z-numbers type. For estimating regression with above conditions, we define the proposed method as follows:

$$y_T = \left(Net, p\left(y_T \text{ } is \text{ } Net\right)\right) = \left(A_{y_T}, B_{y_T}\right) \tag{6}$$

Where y_T is the proposed solution and Net is the feed forward artificial neural network that consists of two layers. The first layer is the inputs layer and the second layer is the outputs layer with linear transfer function that is introduced in the following form:

$$\begin{aligned} &Net = w_0 + w_1 \mathrm{o_{i1}} + \ldots + \mathrm{w_n o_{in}}, \\ &i = 1, \ldots, m, j = 1, \ldots, n \end{aligned} \tag{7}$$

Where w_0 and w_1 are artificial neural network weights. $\mathrm{o_{ij}}$ are inputs of neural network and correspond X. Now, let's suppose our proposed method consists of the same density function of the problem, that is, $\mathrm{f}(X) = \lambda e^{-\lambda X}, \alpha \le \lambda \le \beta$. It is clear that the neural network of relation (7) has fuzzy value and this means that we have fuzzy weights for real observations, thus equation (7) could be written as follows:

$$Net = w_0 + w_1 \mathrm{o_{i1}} + \cdots s.t w_i \text{ } is \text{ } fuzzy \tag{8}$$

We suppose $Net = (Net_{A_1}, Net_{A_2})$ where the value of Net_{A_1} is the center and the value of Net_{A_2} is the fuzzy width of Net neural network, so the relation (8) could be rewritten as follows:

$$\begin{aligned} Net &= \left(Net_{A_1}, Net_{A_2}\right) \\ &= \left(w_{0A_1}, w_{0A_2}\right) + \left(w_{1A_1}, w_{1A_2}\right)\mathrm{o_{i1}} + \cdots \end{aligned} \tag{9}$$

Where Net_{A_1} and Net_{A_2} are in the following form:

$$\begin{cases} Net_{A_1} = w_{0A_1} + w_{1A_1}\mathrm{o_{ij}} + \cdots \\ Net_{A_2} = w_{0A_2} + w_{1A_2}\left|\mathrm{o_{ij}}\right| + \cdots \end{cases} \tag{10}$$

Now, we should find the four weights of relation (9) that are met following two conditions:

1 - The value of Net that is almost the estimated answer y_T is close to the value limit of the main answer Y.
2 - The value of $p\left(y_T \text{ } is \text{ } Net\right)$ approaches the value of $p(a \le Y \le b) = (c, d)$.

To that end, we define the target function of the neural network as follows:

$$\mathrm{Min}(e^{-\left(Net_{A_1}\lambda_1\right)}-e^{-a\lambda_1})^2 + \frac{1}{6}\left(e^{-\left(Net_{A_2}\lambda_2\right)}-e^{-b\lambda_2}\right)^2 \tag{11}$$

Where in general, for n observations we will have x.

$$\mathrm{Min}\sum_{i=1}^{n}\left(\left(e^{-\left(Net_{iA_1}\lambda_1\right)}-e^{-a_i\lambda_1}\right)^2 + \frac{1}{6}\left(e^{-\left(Net_{iA_2}\lambda_2\right)}-e^{-b_i\lambda_2}\right)^2\right)$$

By minimizing equation (11) four weights of the neural network, namely, w_{0A_1}, w_{1A_1}, w_{0A_2} and w_{1A_2} are obtained. By substituting the weights obtained in equation (9), the values of Net_{A_1} and Net_{A_2} are obtained and eventually the value of Net in equation (8) will be obtained. Now, we calculate the probability that y_t may be a neural network (Net)

$$\begin{aligned} p\left(y_T \text{ is } Net\right) &= p\left(Net_{A_1}\le y_T \le Net_{A_2}\right) \\ &= \int_{Net_{A_1}}^{Net_{A_2}} \lambda e^{-\lambda o} do = e^{-(\lambda Net_{A_1})} - e^{-\left(\lambda Net_{A_2}\right)} \\ &\Rightarrow p\left(y_T \text{ is } Net\right) \\ &= \begin{cases} e^{-(\lambda_1 Net_{A_1})} - e^{-\left(\lambda_1 Net_{A_2}\right)}, \lambda_1 = \alpha \\ e^{-(\lambda_2 Net_{A_1})} - e^{-\left(\lambda_2 Net_{A_2}\right)}, \lambda_2 = \beta \end{cases} \end{aligned} \tag{12}$$

By substituting values obtained in equation (12) and in equation (6), the approximate solution with rating Z-number y_T is obtained. In the example section it is shown that the approximate solution is close to the main solution.

4.2. Second State

We consider the problem in such a way that the regression coefficients themselves be of the Z-number type objective based on following observations:

$$\left([y_i]^z,\ x_{i1}, x_{i2}, \cdots, x_{in}\right),\ i = 1, \ldots, n.$$

To obtain an optimal model with Z-number co-efficients for describing, analyzing and predicting based on it, where x_{ij} are real and $[y_i]^z$ are of Z-number type.

The basic model of Z-number linear regression (ZLR):

$$[Y_i]^z = [A_0]^z + [A_1]^z x_{i1} + [A_2]^z x_{i2} + \ldots + [A_n]^z x_{in} \tag{13}$$

Where

$x_{i1} \in R$, $[A_0]^z$, $[A_1]^z$, $[A_2]^z$, …, $[A_n]^z$, and $[Y_i]^z$ are Z-number.

We suppose $[A]^z$ and $[y]^z$ are respectively regular pairs $[A]^z = (A_k, A_H)$ and $[y]^z = (y_k, y_H)$ that A_H, A_k, y_H, y_k are fuzzy numbers. A_k and y_k play the role of restriction and A_H and y_H play the role of degree of confidence for restrictions, respectively. The discussion is limited to the state where A_H, A_k, y_H, Wy_k be symmetric fuzzy triangular numbers. Thus, the main problem with density function $f(X) = \lambda e^{-\lambda X}, \alpha \le \lambda \le \beta$(where a and b are fixed numbers greater than zero and $\alpha < \beta$ are shown as follows:

$$\left(y_k, y_H\right) = \left(A_{k0}, A_{H0}\right) + \left(A_{k1}, A_{H1}\right)x_1 + \cdots + \left(A_{kn}, A_{Hn}\right)x_n \tag{14}$$

That is the probability Y, be y_k and equals to y_H and in the form of relation (14). For estimating Z-number regression of equation (14) with above conditions, we define the proposed method as follows:

$$\begin{aligned} &[y_T]^z = (A', B') \\ &[y_T]^z = [w_0']^z + x_{i1}[w_0']^z + \cdots + x_{in}[w_0']^z \\ &\quad +[\varepsilon_n]^z, i = 1, \ldots, m., j = 1, \ldots., \end{aligned} \tag{15}$$

Where A' plays the role of output restriction of the neural network and B' the probability that y_T be A'. w'_i are the Z-number co-efficients of y_T where the value of the restriction section of these co-efficients are determined using a neural network introduced in the relationship (8). Suppose $[w_i']^z = (w_{iA}', w_{iB}')$ where w'_{iA} and w'_{iB} are symmetric fuzzy triangular numbers. Indeed iA means that the related component plays the role of restriction w_i and indeediB means that the related component plays the role of degree of confidence for the first component that has been shown with iA matrix. Supposing that $w_{iA}' = \left(w_{iA_1}', w_{iA_2}'\right)$ and $w_{iB}' = (w_{iB_1}', w_{iB_2}')$, equation (15) can be rewritten as follows:

$$\begin{aligned} [y_T]^z &= \left(w_{0A_1'}', w_{0A_2'}'\right)\left(w_{0B_1'}', w_{0B_2'}'\right) \\ &+\left(w_{1A_1'}', w_{1A_2'}'\right)\left(w_{1B_1'}', w_{1B_2'}'\right)x + \ldots \end{aligned} \tag{16}$$

Where A' is in below form and later in equation (20) we introduce B'.

$$A' = \begin{cases} A'1 = w_{0A_1}' + w_{1A_1}'x, \\ A'2 = w_{0A_2}' + w_{1A_2}'x. \end{cases} \tag{17}$$

The coefficients in equation (17) are obtained from relationships (8) and (11).

Since these coefficients in our proposed method are of Z-number value, now we should obtain the probabilities of these coefficients:

$$\begin{aligned} &p\left(w_{0A_1'}' \le w_0' \le w_{0A_2'}'\right) \\ &= \int_{w'_{0A_1}}^{w'_{0A'_2}} \lambda e^{-\lambda X} DX = e^{-(\lambda w'_{0A_1})} - e^{-\left(\lambda w'_{0A_2'}\right)} \\ &\Rightarrow p\left(w_{0A_1'}' \le w_0' \le w'_{0A_2'}\right) \\ &= \begin{cases} w_{0B_1'} = e^{-(\lambda_1 w'_{0A_1})} - e^{-\left(\lambda_1 w'_{0A'_2}\right)}, \lambda_1 = \alpha \\ w_{0B'_2} = e^{-(\lambda_2 w'_{0A_1})} - e^{-\left(\lambda_2 w'_{0A'_2}\right)}, \lambda_2 = \beta \end{cases} \end{aligned} \tag{18}$$

And

$$\begin{aligned} &p\left(w'_{1A_1'} \le w_1' \le w'_{1A'_2}\right) \\ &= \int_{w'_{1A_1}}^{w'_{1A'_2}} \lambda e^{-\lambda X} DX = e^{-(\lambda w'_{1A_1})} - e^{-\left(\lambda w'_{1A'_2}\right)} \\ &\Rightarrow p\left(w'_{1A_1'} \le w_1' \le w'_{1A'_2}\right) \\ &= \begin{cases} w_{1B_1'} = e^{-\left(\lambda_1 w'_{1A_1}\right)} - e^{-\left(\lambda_1 w'_{1A'_2}\right)}, \lambda_1 = \alpha \\ w_{1B_2'} = e^{-\left(\lambda_2 w'_{1A_1}\right)} - e^{-\left(\lambda_2 w'_{1A'_2}\right)}, \lambda_2 = \beta \end{cases} \end{aligned} \tag{19}$$

That there by, $w'_{0B'_1}, w'_{0B'_2}, w'_{1B'_1}$ and $w'_{1B'_2}$ are obtained from the neural network. Ultimately B' is computable as follows:

$$B' = \begin{cases} B'1 = w'_{0B_1} + \lambda_1 w'_{1B_1} o, \lambda_1 = \alpha \\ B'2 = w'_{0B_2} + \lambda_2 w'_{1B_2} o, \lambda_2 = \beta. \end{cases} \tag{20}$$

In the examples section, it is shown that the value of A' is close to the value of A and the value of B' is close to the value of B and this shows that our Z-number weights are almost the regression coefficients (ZLR).

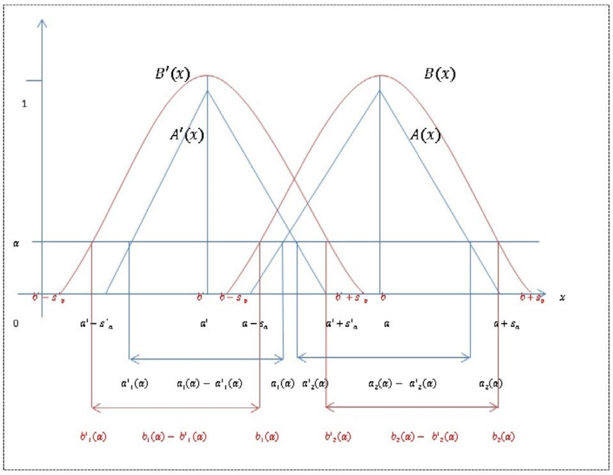
4.3. Third State

In this state we do not have enough information on the problem of regression, that is, we have been given the Z-number values Y based on observations and we do not have any information about the density function or the probability distribution function. We state following definition prior to solving regression.

Definition: suppose $Z_1 = (A, B)$ and $Z_2 = (A', B')$ be two Z-numbers in a way that $A = (a, s_a)_T$, $A' = (a', s'_a)_T$ and $B = (b, s_b)_T$, $B' = (b', s'_b)_T$, be symmetric and fuzzy. Then, based on weight function $f(\alpha) = \alpha$,

$$d^2(A, A') = (a - a')^2 + \frac{1}{6}(s_a - s'_a)^2$$

$$d^2(B, B') = (b - b')^2 + \frac{1}{6}(s_b - s'_b)^2$$



Now, based on the above definition for solving Z-number linear regression, we first divide the problem into two sections; where the first section is based on restriction and the second section based on the confidence scale. Thus, relation (2) could be written as follows:

$$\begin{aligned} &y_k = A_{k0} + A_{k1}x_1 + \ldots + A_{kn}x_n \\ &(y_k, x_{i1}, x_{i2}, \ldots, x_{in}); x_{ij} \in R, \\ &i = 1, \cdots, m, \ j = 1, \ldots, n. \end{aligned} \tag{21}$$

And

$$\begin{aligned} &y_H = A_{H0} + A_{H1}x_1 + \ldots + A_{Hn}x_n \\ &(y_H, x_{i1}, x_{i2}, \ldots, x_{in}); x_{ij} \in R, \\ &i = 1, \ldots, m, j = 1, \ldots, n. \end{aligned} \tag{22}$$

There are different methods for calculating each one of A_k and A_H. References (Diamond, 1988; Hojati, Bector, & Smimou, 2005; Peters, 1994; Savic & Pedrycz, 1991; Tanaka, Uejima, & Asai, 1982) could be used for more study.

In this work, since the correlation between known and unknown variables is not clear, thus the artificial neural network could be an appropriate method for an approximate answer. Therefore, we consider the double-layer progressive Z-number neural network that consists of n input units and one output unit. Input vectors are real and their related weights are of Z-number type. And the correlation between the inputs and outputs of the Z-number neural network could be written in below the form:

Input units:

$$o_{i0} = 1, o_{ij} = x_{ij}, j = 1, 2, \ldots, n, i = 1, \ldots, m.$$

Output units:

$$\begin{aligned} [y_i]^z &= f([Net_i]^z), i = 1, \ldots, m, [Net_i]^z \\ &= [w_0]^z \ldots + o_{in}[w_n]^z + [\varepsilon_n]^z \end{aligned} \tag{23}$$

As f is an increment function. $[\varepsilon_n]^z$ is the approximation method error.

We suppose $[w_i]^z = (u_i, v_i)$ where u and v are symmetric triangular fuzzy numbers. For answer approximation of equations (21) and (22) we will have:

$$\begin{aligned} [Net]^z &= (N_k, N_H) \\ (N_k, N_H) &= (u_0, v_0) + (u_1, v_1)o_1 + \cdots \\ &\quad + (u_n, v_n)o_n + (\varepsilon_k, \varepsilon_H). \end{aligned} \tag{24}$$

That N_k and N_H both are artificial fuzzy neural networks and are in the following forms:

$$N_k = u_0 + u_1 o_1 + \cdots + u_n o_n + \varepsilon_k \tag{25}$$

And

$$N_H = v_0 + v_1 o_1 + \cdots + v_n o_n + \varepsilon_H \tag{26}$$

5. Error Analysis

In this section, we study the error for the third state, that the first and second states can be studied in the same way.

The $[\varepsilon_i]^z$ error in equation (2) is stated in following form:

$$[\varepsilon_i]^z = [Y_i]^z - [Net_i]^z \tag{27}$$

$$[M_i]^z = (M_{ki}, M_{Hi}) = \left(\sum_{i=1}^{n} \varepsilon_{ki}^2, \sum_{i=1}^{n} \varepsilon_{Hi}^2 \right)$$

Where ε_{ki}, ε_{Hi} are symmetric triangular fuzzy numbers.

Now, we minimize the total square error given by distance d mentioned in previous section by using $[w_i]^z = (u_i, v_i)$.

$$\begin{aligned} \mathrm{Min}[M_i]^z &= (\mathrm{Min}M_{ki}, \mathrm{Min}M_{Hi}) \\ &= (\mathrm{Min}M(u_0, \ldots, u_n), \mathrm{Min}M(v_0, \ldots, v_n)) \end{aligned}$$

Given what was said for both equations (25) and (26) we have:

$$M(u_0, \ldots, u_n) = \sum_{i=1}^{m} d^2(N_{ki}, y_{ki}) \tag{28}$$

$$M\left(v_0,\ldots,v_n\right)=\sum_{i=1}^{m} d^2\left(N_{Hi},\,y_{Hi}\right) \tag{29}$$

The idea of finding the least squares is $\left[w_i\right]^z$ that are obtained by minimizing $\left[M_i\right]^z$, that is, the total square errors on distance d and this is done using the Matlab soft-ware and fminunc command is based on Quasi-Newton algorithm.

6. Weight Calculation Algorithm for the Third State

In this section, the algorithm for the calculation of neural network weights for the third state is studied. The first and second states resemble the third state too. To that end, we first explain the method of calculating u_i.

Suppose y_{ki} and u_i be as $y_{ki}=(y_{ki},s_{ki})_T$ and $u_i=(u_i,\sigma_i)_T$. Where y_{ki} and u_i are the centers, s_{ki} and σ_i the widths of the symmetric triangular fuzzy numbers y_{ki} and u_i, respectively. Given the equation (10) we have

$$N_{ki}=\left(u_0+u_1o_{i1}+\cdots+u_no_{in},\;\sigma_0+\sigma_1\left|o_{i1}\right|+\cdots\sigma_n\left|o_{in}\right|\right)_T. \tag{30}$$

Where

$$[Net_{ki}]_{Two}=u_0+u_1o_{i1}+\ldots+u_no_{in} \tag{31}$$

$$[Net_{ki}]_{Two}=\sigma_0+\sigma_1\left|o_{i1}\right|+\ldots\sigma_n\left|o_{in}\right| \tag{32}$$

Given the equation (31) and (32), relation (28) turns into following form:

$$M\left(u_0,\ldots,u_n\right)=\sum_{i=1}^{m}((\left[Net_{ki}\right]_{One}-y_{ki})^2+\frac{1}{6}(\left[Net_{ki}\right]_{Two}-S_{ki})^2) \tag{33}$$

To calculate v_i, in a similar way, suppose y_{Hi} and v_i are $y_{Hi}=(y_{Hi},s_{Hi})_T$ and $v_i=(v_i,c_i)_T$ where y_{Hi} and v_i are the center, s_{Hi} and c_i are the widths of symmetric triangular fuzzy number, y_{Hi} and v_i, respectively. Given the equation (10), we have

$$N_{Hi}=\left(v_0+v_1o_{ij}+\ldots+v_no_{in},\quad c_0+c_1\left|o_{i1}\right|+\ldots+c_n\left|o_{in}\right|\right)_T.$$

Where

$$[Net_{Hi}]_{One}=v_0+v_1o_{i1}+\ldots+v_no_{in} \tag{34}$$

$$[Net_{Hi}]_{Two}=c_0+c_1\left|o_{i1}\right|+\ldots+c_n\left|o_{in}\right| \tag{35}$$

Given the equations (34) and (35), equation (29) turns into following form:

$$M\left(v_0,\ldots,v_n\right)=\sum_{i=1}^{m}\left(\left(\left[Net_{Hi}\right]_{One}-y_{Hi}\right)^2+\frac{1}{6}\left(\left[Net_{Hi}\right]_{Two}-S_{Hi}\right)^2\right) \tag{36}$$

Now, by minimizing equations (33) and (36) with initial weights $u_i=v_i=0$, weights in the direction that the target function (the performance function) decreases, that is, contrary to its slope, they are updated, the algorithm of the neural network for the purpose of calculating the weights in this article is a quasi-Newton algorithm or BFGs (Broyden Fletcher Goldfarb Shanno), (Liu & Nocedal, 1989). The basic step in Quasi-Newton methods is calculated based on teewton formula. the N

$$\begin{aligned} u_{ij}(k+1) &= u_{ij}(k)-A_1^{-1}(k)\frac{\partial\hat{F}_{1i}(k)}{\partial u_{ij}(k)},\\ \sigma_{ij}(k+1) &= \sigma_{ij}(k)-A_1^{-1}(k)\frac{\partial\hat{F}_{1i}(k)}{\partial\sigma_{ij}(k)}. \end{aligned} \tag{37}$$

$$\begin{aligned} v_{ij}(k+1) &= v_{ij}(k)-A_2^{-1}(k)\frac{\partial\hat{F}_{2i}(k)}{\partial v_{ij}(k)},\\ c_{ij}(k+1) &= c_{ij}(k)-A_2^{-1}(k)\frac{\partial\hat{F}_{2i}(k)}{\partial c_{ij}(k)} \end{aligned} \tag{38}$$

In a way that $A_1(k)$ and $A_2(k)$ are respectively the matrices of the second derivatives from performance functions $\hat{F}1(k)$ and $\hat{F}2(k)$ for present values of the weights. Also

$$\hat{F}1(k)=\sum_{i=1}^{m}e_{ik1}^2+\frac{1}{6}\sum_{i=1}^{m}e_{ik2}^2;e_{ik1}\text{ and }e_{ik2}\text{ for }y_k,\,N_k \tag{39}$$

$$\hat{F}2(k)=\sum_{i=1}^{m}e_{iH1}^2+\frac{1}{6}\sum_{i=1}^{m}e_{iH2}^2;e_{iH1}\text{ and }e_{iH2}\text{ for }y_H,\,N_H \tag{40}$$

In the above equations, k is the number of repetitions. The disadvantage of Newton's method is that it is too complex and computationally too expensive and as a result they are not appropriate for neural networks, of course, there is a type of algorithm based on Newton's methods that since it does not require calculating the second derivative, its computational cost is lower. These methods are called Quasi-Newton's methods. They update the algorithm of approximate Hessian matrix for each repetition. Updating is conducted through a function from the slope. The Quasi-Newton method that has considerably been successful consists of BFGs method. This algorithm is usually converged faster and in lower number of repetitions. In the end, after obtaining weights, that is, u_i and v_i using the mentioned algorithm (that is by order of fminunc in subject), we substitute them in equation (24) and achieve relation (23).

For the first and second states the same process is applied.

7. The Stability of Neural Network Weights

In this section, we study the stability of artificial neural network weights. The repetitive relation of the weight calculation in equation (37) must be considered as follows:

$$u_{ij}(k+1)=u_{ij}(k)-\alpha\frac{\partial\hat{F}_{1i}(k)}{\partial u_{ij}(k)},$$

$$\sigma_{ij}(k+1)=\sigma_{ij}(k)-\alpha\frac{\partial\hat{F}_{1i}(k)}{\partial\sigma_{ij}(k)}$$

Where α is the Hessian matrix approximation in each repetition and k is representative of each repetition. Now, we consider the initial weights as $u_0^*(k)=u_0(k)+\Delta_0(k)$ and $\sigma_0^*(k)=\sigma_0(k)+\Delta_0{}'(k)$ that in this case for

$$\hat{F}1(k)=(u_0(k)+u_1(k)o_1-y_k)^2+\frac{1}{6}(\sigma_0(k)+\sigma_1(k)o_1-s_k)^2$$

We will have

$$\hat{\hat{F}}1=(u_0^*(k)+u_1(k)o_1-y_k)^2+\frac{1}{6}(\sigma_0^*(k)+\sigma_1(k)o_1-s_k)^2,$$

$$\tilde{\tilde{F}}1 = (u_0(k) + \Delta_0(k) + u_1(k)o_1 - y_k)^2 + \frac{1}{6}(\sigma_0(k) + \Delta_0'(k) + \sigma_1(k)o_1 - s_k)^2$$

$$\frac{\partial \tilde{\tilde{F}}1}{\partial u_0^*} = \frac{\partial}{\partial u_0^*}((u_0(k) + \Delta_0(k) + u_1(k)o_1 - y_k)^2 + \frac{1}{6}\left(\sigma_0(k) + \Delta_0'(k) + \sigma_1(k)o_1 - s_k\right)^2)$$

Thus, in the next repetition, that is, k \, + \, 1, we will have

$$\begin{aligned} u_0^*(k+1) &= u_0^*(k) - \alpha\frac{\partial \tilde{F}_1(k)}{\partial u_0^*(k)} \\ &= u_0(k) + \Delta_0(k) - \alpha\frac{\partial \tilde{F}_1(k)}{\partial u_0^*(k)} \end{aligned}$$

$$\begin{aligned} \Rightarrow u_0^*(k+1) &= u_0(k) + \Delta_0(k) \\ &- \alpha\frac{\partial}{\partial u_0^*}((u_0^*(k) + u_1(k)o_1 - y_k)^2 \\ &+ \frac{1}{6}\left(\sigma_0(k) + \Delta_0'(k) + \sigma_1(k)o_1 - s_k\right)^2) \\ &= u_0(k) + \Delta_0(k) - 2\alpha\left(u_0^*(k) + u_1(k)o_1 - y_k\right) \end{aligned}$$

Now, we determine the limitation of both sides of the above equation when Δ_0 tends towards zero.

$$\begin{aligned} \lim_{\Delta_0 \to 0} u_0^*(k+1) &= \lim_{\Delta_0 \to 0} u_0(k) \\ &+ \Delta_0(k) - 2\alpha \lim_{\Delta_0 \to 0}\left(u_0^*(k) + u_1(k)o_1 - y_k\right) \\ &= u_0(k) - 2\alpha(u_0(k) + u_1(k)o_1 - y_k) \\ &= u_0(k+1) \end{aligned}$$

$$\Rightarrow \lim_{\Delta_0 \to 0} u_0^*(k+1) = u_0(k+1)$$

That is

$$u_0^*(k+1) - u_0(k+1) \overset{\Delta_0 \to 0}{\to} 0$$

The procedure for the other weights is the same. For the first and second states it is similar, too.

8. Numerical Example

In this section, three numerical examples are presented for calculating Z-number regression coefficients according to three states mentioned above.

Example 1. For Z-number variables, consider the dependent variable Y and the independent real variable x_i, the values given in Table 1 (information in Table 1 has been adopted from reference (Tanaka et al., 1989)).

Suppose that the Z-number value of $[Y_i]^z$ for each real observation Xi is in table 2:

(1a) Using the data in Table 2 and Table 3a, develop an estimated fuzzy regression equation $[Y_i]^z = \left(A_Y, B_Y\right)$ by $y_T = \left(Net, p\left(y_T \; is \; Net\right)\right)$, (first state). Where stopping criteria; \, k=32 iterations of the learning algorithm. The training starts with $w_0 = (0,0), w_1 = (0,0)$ The value of $[y_{Ti}]^z$ for density function $\lambda e^{-\lambda X}$ in Table 3a for $0.2 \le \lambda \le 0.3$visible. The optimal weight of the neural network is as follows:

$$Net = (5.60, \; 1.79) + (1.34, \; 0.17)x$$

(1b) Using the data in Table 2 and Table 3b, develop an estimated fuzzy regression equation $[Y_i]^z = \left(A_Y, B_Y\right)$ by $[y_T]^z = (A', B')$, (two state). Where stopping criteria; k \, = \, 19 iterations of the learning algorithm. The training starts with $[w_0]^z = (0,0)(0,0), [w_1]^z = (0,0)(0,0)$. The value of $[y_{Ti}]^z$ in Table 3b is visible.

The optimal weight of the neural network is as follows:

$$w_{0A} = (5.60, \; 1.79)(0.37, \; 0.39),$$

$$w_{1A} = (1.34, \; 0.17)(0.20, \; 0.27).$$

(1c) Using the data in Table 2 and Table 3c, develop an estimated fuzzy regression equation $[Y_i]^z = \left(A_Y, B_Y\right)$ by $[Net_i]^z = [w_0]^z + o_{i1}[w_1]^z$, (three state). Where stopping criteria; k=19 iterations of the learning algorithm. The training starts with $[w_0]^z = (0,0)(0,0), [w_1]^z = (0,0)(0,0)$. The value of $[Net_i]^z$ in Table 3c is visible. Weights obtained in our proposed scheme (FLS network) can be obtained with weight FLS procedure (Diamond, 1987) comparable.

$$y_{ik} = (4.95, 1.84) + (1.70, 0.16)x \mathit{For\ FLS\ network}$$

$$y = (4.95, 1.84) + (1.71, 0.16)x \mathit{\ forFLSmethod}$$

Table 1. Crisp Input- Fuzzy Output Data-set from (Tanaka, Havashi & Watada, 1989).

Interval Yi	$Y_i = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(6.2, 9.8)	(8, 1.8)	1	1
(4.2, 8.6)	(6.4, 2.2)	2	2
(6.9, 12.1)	(9.5, 2.6)	3	3
(10.9, 16.1)	(13.5, 2.6)	4	4
(10.6, 15.4)	(13, 2.4)	5	5

Table 2. Crisp Input- Z-number Output.

$Y_i = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(8, 1.8)(0.495,0.492)	1	1
(6.4, 2.2)(0.365,0.370)	2	2
(9.5, 2.6)(0.444,0.400)	3	3
(13.5, 2.6)(0.527,0.440)	4	4
(13, 2.4)(0.544,0.466)	5	5

Table 3a. Crisp Input- Z-number Output data-set.

$[Y_T]^z = (Y_T, \underline{e}i, \bar{e}i)_T$	Xi	i
(6.9, 1.9)(0.432, 0.439)	1	1
(8.3, 2.1)(0.4669, 0.449)	2	2
(9.6, 2.3)(0.484, 0.445)	3	3
(10.9, 2.4)(0.5057, 0.448)	4	4
(12.3, 2.6)(0.5090, 0.4334)	5	5

Table 3b. Crisp Input- Z-number Output Data-set.

$y_{ik} = y_{ih} = (yi, \underline{e}i, \bar{e}i)_T$	Xi	i
(6.9, 1.9)(0.41, 0.474)	1	1
(8.3, 2.1)(0.45, 0.558)	2	2
(9.6, 2.3)(0.49, 0.624)	3	3
(10.9, 2.4)(0.53, 0.726)	4	4
(12.3, 2.6)(0.57, 0.81)	5	5

Table 3c. Crisp Input- Z-number Output Data-set.

$[Y_i^z = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(6.6, 2)(0.42, 0.43)	1	1
(8.37, 2.16)(0.44, 0.43)	2	2
(10.08, 2.32)(0.47, 0.43)	3	3
(11.78, 2.48)(0.49, 0.43)	4	4
(13.49, 2.64)(0.52, 0.43)	5	5

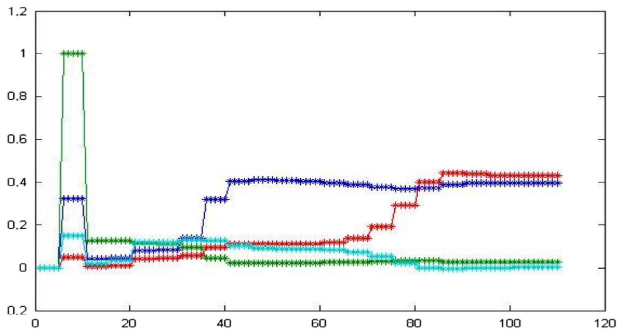


Figure 1a. The Convergence of Neural Network Weights to the Component likely Example (1c).

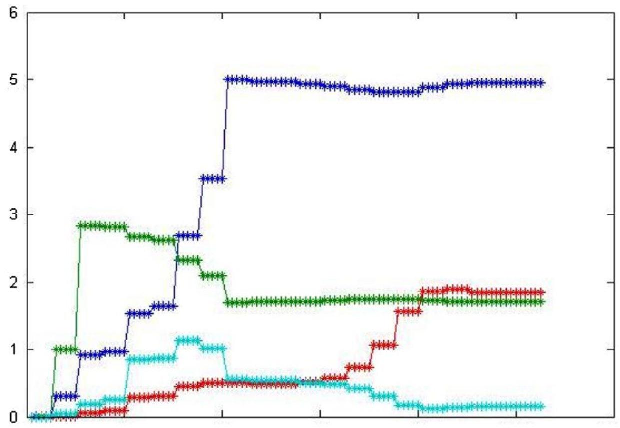


Figure 1b. The Convergence of Neural Network Weights of Components in such restrictions 1c.

To have y_{ik}:

Firstorderopt: 4.220008850097656e-05

$$\hat{F} = 9.219000000073168$$

TT = 0.436802800000002

Where TT is the time to calculate the weights of the neural network related to y_{ik}. The convergence of the neural network weights to y_{ik} for the third mode is visible in Figure 1b.

To have y_{iH}:

Firstorderopt: 2.019573003053665e-06

$$\hat{F} = 0.016321933333523$$

TT = 0.436802799999999

The convergence of the neural network weights to y_{iH} for the third mode is visible in Figure 1a.

In this case we have

$$\begin{aligned}[y]^z &= [A_0]^z + [A_1]^z x_1 \\ &= (4.95, 1.84)(0.398, 0.427) \\ &\quad + (1.70, 0.16)(0.025, 0.002)x.\end{aligned}$$

Example 2. For Z-number variables, consider dependent variable Y and independent real variable x_i, the values given in Table 4 (information in Table 4 has been adopted from reference (Diamond, 1988)).

Suppose that the Z-number value of $[Y_i]^z$ for each real observation Xi is in table 5a:

(2a) Using this data, develop an estimated fuzzy regression equation $[Y_i]^z = (A_Y, B_Y)$ by $y_T = (Net, p(y_T \text{ is } Net))$, (first state). Where stopping criteria; k=30 iterations of the learning algorithm. The training starts with $w_0 = (0,0), w_1 = (0,0)$.

Table 4. Crisp Input- Fuzzy Output Data-set from (Diamond, 1988).

$Y_i = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(4, 0.8)	21	1
(3, 0.3)	15	2
(3.5,0.35)	15	3
(2, 0.4)	4	4
(3, 0.45)	12	5
(3.5, 0.7)	18	6
(2.5, 0.38)	6	7
(2.5, 0.5)	12	8

Table 5a. Crisp Input- Z-number Output Data.

$[Y_i]^z = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(4, 0.8)(0.40, 0.48)	21	1
(3, 0.3)(0.39, 0.50)	15	2
(3.5,0.35)(0.43, 0.55)	15	3
(2, 0.4)(0.25, 0.33)	9	4
(3, 0.45)(0.36, 0.46)	12	5
(3.5, 0.7)(0.37, 0.46)	18	6
(2.5, 0.38)(0.32, 0.41)	6	7
(2.5, 0.5)(0.29, 0.38)	12	8

Table 5b. Relative Performance of y_{iK} and Other Methods (CPU Time).

Methods	SSE	CPU time (hh:mm:ss:ss)
Tan	3.6542	00:00:00:16
HBS	0.6666	00:00:00:94
Pet	1.1978	00:00:00:06
FLS	0.0846	00:00:00:19
SP	14.8243	00:00:00:16
FRBF	0.0046	00:02:06:56
Proposed method	**75.34**	**00:00:00:7**

Table 6a. Crisp Input- Z-number Output Data-set.

$[Y_{Ti}]^z = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(3.73, 0.58)(0.41, 0.51)	21	1
(3.07, 0.46)(0.37, 0.47)	15	2
(3.07, 0.46)(0.37, 0.47)	15	3
(2.41, 0.34)(0.31, 0.41)	9	4
(2.74, 0.4)(0.34, 0.44)	12	5
(3.40, 0.52)(0.39, 0.49)	18	6
(2.08, 0.28)(0.28, 0.38)	6	7
(2.74, 0.4)(0.34, 0.44)	12	8

The value of $[y_{Ti}]^z$ for density function $\lambda e^{-\lambda X}$ in Table 6a for $0.2 \leq \lambda \leq 0.3$ is visible. The optimal weight of the neural network is as follows:

$$Net = (1.42,\ 0.16) + (0.11,\ 0.02)x$$

(2b) Using this data, develop an estimated fuzzy regression equation $[Y_i]^z = (A_Y, B_Y)$ by $[y_T]^z = (A', B')$ (two state). Where stopping criteria; $k = 30$ iterations of the learning algorithm. The training starts with $[w_0]^z = (0,0)(0,0), [w_1]^z = (0,0)(0,0)$. The value of $[y_{Ti}]^z$ in Table 6b is visible.

The optimal weight of the neural network is as follows:

$$w_{0A} = (1.42,\ 0.16)(0.21,\ 0.30),$$

$$w_{1A} = (0.11,\ 0.02)(0.01,\ 0.02).$$

(2c) Using this data, develop an estimated fuzzy regression equation $[Y_i]^z = (A_Y, B_Y)$ by $[Net_i]^z = [w_0]^z + o_{i1}[w_1]^z$, (three state). The training starts with $[w_0]^z = (0,0)(0,0), [w_1]^z = (0,0)(0,0)$. The value of $[Net_i]^z$ in Table 6c is visible. For y_{ik}, weights obtained can be obtained with the weight FLS procedure (Diamond, 1987) comparable.

Table 6b. Crisp Input- Z-number Output Data-set.

$[Y_{Ti}]^z = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(3.73, 0.58)(0.25, 0.27)	21	1
(3.07, 0.46)(0.24, 0.25)	15	2
(3.07, 0.46)(0.24, 0.25)	15	3
(2.41, 0.34)(0.22, 0.23)	9	4
(2.74, 0.4)(0.23, 0.24)	12	5
(3.40, 0.52)(0.24, 0.26)	18	6
(2.08, 0.28)(0.222, 0.228)	6	7
(2.74, 0.4)(0.23, 0.24)	12	8

Table 6c. Crisp Input- Z-number Output Data-set.

$[Y_{Ti}]^z = (Y_i, \underline{e}i, \bar{e}i)_T$	Xi	i
(3.9, 0.6)(0.39, 0.48)	21	1
(3.1, 0.5)(0.35, 0.44)	15	2
(3.1, 0.5)(0.35, 0.44)	15	3
(2.4, 0.3)(0.30, 0.39)	9	4
(2.8, 0.4)(0.32, 0.41)	12	5
(3.5, 0.5)(0.37, 0.46)	18	6
(2, 0.2)(0.27, 0.36)	6	7
(2.8, 0.4)(0.32, 0.41)	12	8

$$y_{ik} = (1.375, 0.147) + (0.120, 0.024)x\ For\ FLSnetwork,$$

$$y_{ik} = (1.374, 0.147) + (0.120, 0.025)xforFLS.$$

Where stopping criteria; $k = 24$ iterations of the learning algorithm,

$$\hat{F} = 0.672169446070879$$

Firstorderopt: 2.999007701873779e-04

TT = 0.780004999999999

The average relative performance of the proposed method for y_{iK} and the other methods, measured by CPU time, is compared in Table 5b.

To have y_{iH}:

k=21,

$$\hat{F} = 0.017117129638503,$$

Firstorderopt: 6.845151074230671e-05

TT = 0.452402900000003

The convergence of the neural network weights first component of the first and second modes can be seen in Figure 2a. The convergence of the neural network weights third case shown in Figure 2b and 2c are visible.

In this case we have

$$[y]^z = [A_0]^z + [A_1]^z x_1 = (1.375, 0.147)(0.23, 0.32) + (0.120, 0.024)(00.8, 00.8)x$$

Example 3. For Z-number variables, consider dependent variable Y and independent real variable x_i, the values given in Table 7 (information in Table 7 has been adopted from reference (Mohammadi & Taheri, 2004)) with density function $\lambda e^{-\lambda X}$ for $0.1 \le \lambda \le 0.2$.

(3a) Using this data, develop an estimated fuzzy regression equation $[Y_i]^z = (A_Y, B_Y)$ by $y_T = (Net, p(y_T\ is\ Net))$, (first state). Where stopping criteria; k=26 iterations of the learning algorithm. The training starts with $w_0 = (0,0), w_1 = (0,0)$. The value of $[y_{Ti}]^z$ for density function $\lambda e^{-\lambda X}$ in Table 6a for $0.2 \le \lambda \le 0.3$ is visible. The optimal weight of the neural network is as follows:

$$Net = (4.41,\ 0.16) + (1.15,\ 0.54)x.$$

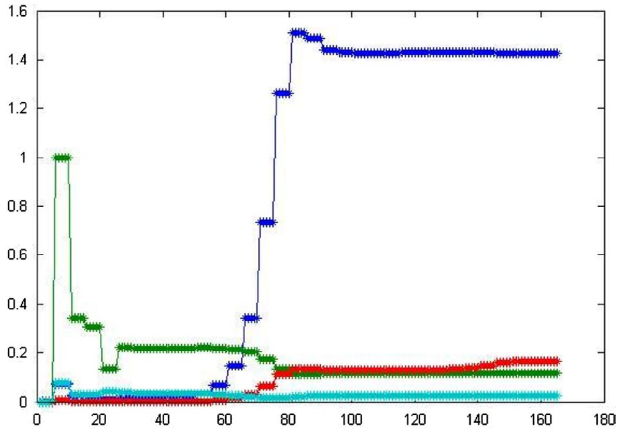


Figure 2a. The Convergence of Neural Network Weights First Component of the First and Second Modes.

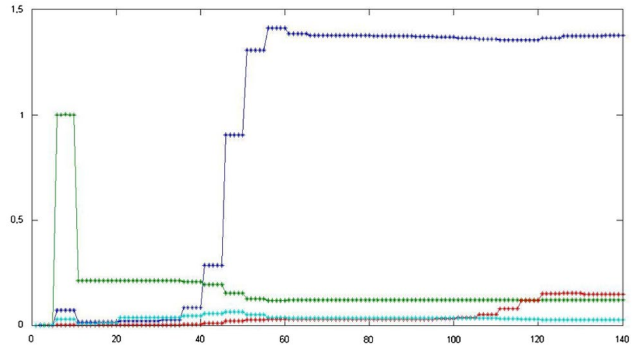


Figure 2b. The First Component Weights Neural Network Convergence ZNN Example 2.

The figure 3c presents the convergence of weights for y_{-iH}

(3b) Using this data, develop an estimated fuzzy regression equation $[Y_i]^z = (A_Y, B_Y)$by$[y_T]^z = (A', B')$ (two state). Where stopping criteria; k=26 iterations of the learning algorithm. The training starts with $[w_0]^z = (0,0)(0,0), [w_1]^z = (0,0)(0,0)$. The value of $[y_{Ti}]^z$ in Table 6b is visible. The optimal weight of the neural network is as follows:

$$w_{0A} = (4.41,\ 0.16)(0.55,\ 0.68),$$

$$w_{1A} = (1.15,\ 0.54)(0.10,\ 0.14).$$

(3c) Using this data, develop an estimated fuzzy regression equation $[Y_i]^z = (A_Y, B_Y)$by$[Net_i]^z = [w_0]^z + o_{i1}[w_1]^z$ (three state). The training starts with $[w_0]^z = (0,0)(0,0), [w_1]^z = (0,0)(0,0)$. The value of $[Net_i]^z$ in Table 6c is visible.

To have y_{ik}:

$$k = 16,$$

$$y_{ik} = (0.81,\ 0.06) + (6.90,\ 0.69)x,$$

$$\hat{F}1 = 2.677921909968936e + 02,$$

Firstorderopt: 3.013610839843750e-04,

TT = 0468003000000000

For example, if $x = 0.75$ the predicted value of y_{ik} will be $y_{ik} = (5.97, 0.57)$, where to (Mohammadi & Taheri, 2004):

$y_{ik} = (5.70, 0.60)$.

To have y_{iH}:

$$k = 13,$$

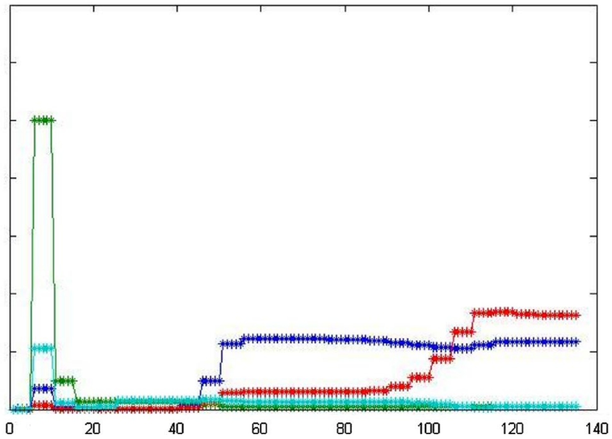


Figure 2c. The Second Component Weights Neural Network Convergence ZNN Example 2.

Table 7. Crisp Input- Z-number Output Data-set.

i	x_i	$[y_i]^z = (y_{ki}, s_i)(y_{Hi}, c_i)$
1	0.78	(3.08, 0.31)(0.23, 0.39)
2	0.64	(2.86, 0.29)(0.22, 0.37)
3	0.62	(6.25, 0.63)(0.40, 0.59)
4	0.49	(4.11, 0.41)(0.29, 0.48)
5	1.10	(1.04, 0.10)(0.08, 0.16)
6	0.61	(2.71, 0.27)(0.21, 0.36)
7	0.74	(4.45, 0.45)(0.31, 0.50)
8	1.15	(6.92, 0.69)(0.43, 0.62)
9	1.08	(7.41, 0.74)(0.45, 0.63)
10	0.38	(9.08, 0.91)(0.50, 0.67)
11	0.61	(6.56, 0.66)(0.41, 0.60)
12	0.98	(5.05, 0.51)(0.34, 0.53)
13	0.71	(5.23, 0.52)(0.35, 0.54)
14	0.51	(5.16, 0.52) (0.35, 0.54)
15	0.77	(11.10, 1.11)(0.56, 0.69)
16	0.99	(4.47, 0.45)(0.31, 0.50)
17	3.56	(28.84, 2.88) (0.69, 0.55)
18	0.86	(9.43, 0.94) (0.52, 0.67)
19	0.61	(4.50, 0.45)(0.31, 0.50)
20	0.64	(9.30, 0.94)(0.51, 0.67)
21	0.71	(9.48, 0.95)(0.52, 0.67)
22	0.61	(3.65, 0.37)(0.26, 0.44)
23	0.63	(10.14, 1.01)(0.54, 0.68)
24	1.13	(3, 0.3)(0.22, 0.39)

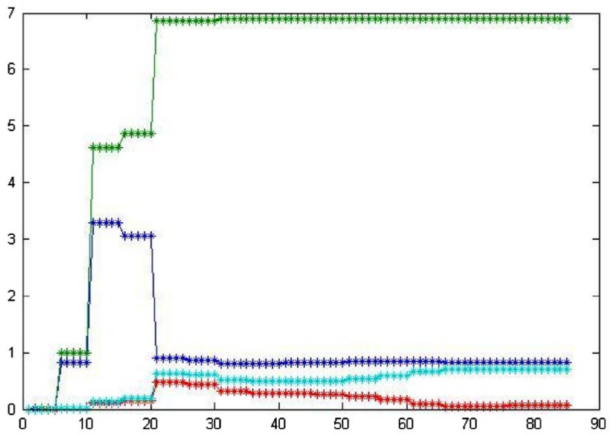


Figure 3a. The Convergence of Neural Network Weights First Case Example 3.

$$y_{iH} = (0.298,\ 0.511) + (0.088,\ 0.001)x,$$

$$\hat{F}1 = 0.509280169810204,$$

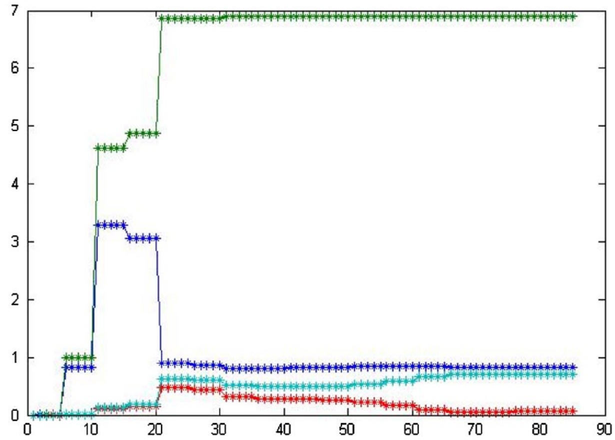


Figure 3b. Convergence Weights for y_{ik} Third Example 3.

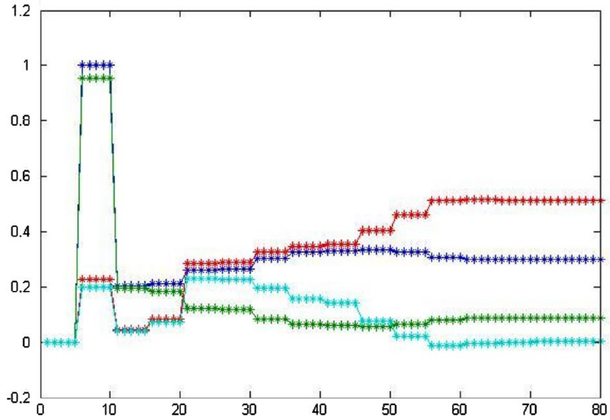


Figure 3c. Convergence Weights for y_{iH} Third Example 3.

Firstorderopt: 8.568167686462402e-06,
TT = 0.452402900000003

$$y_{iH} = (0.36, 0.51),\ \mathit{for}\ x = 0.75.$$

The convergence of the weights for y_{ik} in Figure 3b is visible.

9. Conclusion

In this article, we introduced a Mathematical model from regression with Z-number coefficients and a generalized neural system based on the given number. Then, we calculated the ZLR regression coefficients using the artificial neural network, the optimization technique and the least square error method based on the distance between two fuzzy numbers. It was shown that the calculation of regression coefficients (ZLR) is dependent on the coefficients of the neural network. The error of the method was studied and it was shown that the method error is of Z-number type. It was proven that the neural network weight section and regression co-efficient restriction section (ZLR) are convergent. Also, the stability of the neural network instruction algorithm in calculating optimal weights was proven and it was shown by examples that the proposed regression (ZLR) is a powerful method in solving complex problems.

Disclosure Statement

No potential conflict of interest was reported by the authors.

Notes on Contributors

Somayeh Ezadi received a B.S. degree in Mathematics in 2005 from Iran Islamic Azad University, Iran, Hamedan, and an M.S. degree in Mathematics in 2012 at the same university. She is currently a Ph.D. student in Mathematics at the Islamic Azad University, Iran, Hamedan. Her research interests are numerical solution of ordinary differential equations, numerical solutions of fuzzy differential equations, numerical solution of fuzzy linear regression, solve high order differential equations.

Tofigh Allahviranloo is a professor of Applied Mathematics at SRB-IAU, Iran. He has a B.Sc. level in Pure Mathematics, 1994, University of Tabriz, Iran and M.Sc. level in Applied Mathematics, (Operation Research), 1999, Lahijan Branch, IAU, Iran. He received a Ph.D. in Fuzzy Differential Equations, 2002, at the Science & Research Branch, IAU, Tehran, Iran. He has published more than 185 articles in international journals and 9 books.

ORCID

Tofigh Allahviranloo http://orcid.org/0000-0002-6673-3560

References

Bardossy, A. (1990). Note on fuzzy regression. *Fuzzy Sets and Systems, 37*, 65–75.

Bardossy, A., Bogardi, I., & Duckstein, L. (1990). Fuzzy regression in hydrology. *Water Resources, 26*, 1497–1508.

Celmins, A. (1987). Least squares model fitting to fuzzy vector data. *Fuzzy Sets and Systems, 22*, 260–269.

Cheng, C.B., & Lee, E.S. (2001). Fuzzy regression with radial basis function network. *Fuzzy Sets and Systems, 119*, 291–301.

Coppi, R. (2008). Management of uncertainty in statistical reasoning: The case of regression analysis. *International Journal of Approximate Reasoning, 47*, 284–305.

Diamond, P. (1987). *Least squares fitting of several fuzzy variables*. Proceedings Of the Second IFSA Congress (pp. 20-25). Tokyo, Japan: International Fuzzy Systems Association.

Diamond, P. (1988). Fuzzy least squares. *Information Sciences, 46*, 141–157.

Hojati, M., Bector, C.R., & Smimou, K. (2005). A simple method of fuzzy linear regression. *European Journal of Operational Research, 166*, 172–184.

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

Kao, C., & Chyu, C.L. (2003). Least-squares estimates in fuzzy regression analysis. *European Journal of Operational Research York, 148*, 426–435.

Liu, C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming, 45*, 503–528.

Modarres, M., Nasrabadi, E., & Nasrabadi, M.M. (2005). Fuzzy linear regression models with least square errors. *Applied Mathematics and Computing, 163*, 977–989.

Mohammadi, J., & Taheri, S.M. (2004). Pedomodels fitting with fuzzy least squares regression. *Iranian Journal of Fuzzy Systems, 1*, 45–61.

Mosleh, M., Otadi, M., & Abbasbandy, S. (2011). Fuzzy polynomial regression with fuzzy neural networks. *Applied Mathematical Modeling, 35*, 5400–5412.

Mosleh, M., Allahviranloo, T., & Otadi, M. (2012). Evaluation of fully fuzzy regression models by fuzzy neural network. *Neural Computing and Applications, 21*, 105–112.

Peters, G. (1994). Fuzy linear regression with fuzzy intervals. *Fuzzy Sets and Systems, 63*, 45–55.

Rosenblatt, F. (1962). *Principles of neurodynamics*. Washington, DC: Spartan Books.

Savic, D.A., & Pedrycz, W. (1991). Evaluation of fuzy linear regression models. *Fuzzy Sets and Systems, 39*, 51–63.

Tanaka, H. (1987). Fuzzy data analysis by possibilistic linear models. *Fuzzy Sets and Systems, 24*, 363–375.

Tanaka, H., Uejima, S., & Asai, K. (1982). Linear regression analysis with fuzzy model. *IEEE Transactions on Systems, Man and Cybernetics, 12*, 903–907.

Tanaka, H., Havashi, I., & Watada, J. (1989). Possibilistic liner regression analysis for fuzzy data. *European Journal of Operational Research, 40*, 389–396.

Xu, R., & Li, C. (2001). Multidimentional least-squares fitting with a fuzzy model. *fuzzy Sets and Systems, 119*, 215–223.

Yager, R. (2012). On Z-valuations using Zadeh's Z-numbers. *International Journal of Intelligent Systems, 27*, 259–278.

Zadeh, L.A. (1965). Fuzzy sets. *Inform and Control, 8*, 338–353.

Zadeh, L.A. (1979). Fuzzy sets and information granularity. In M.M. Gupta, R.K. Ragade, & R.R. Yager (Eds.), *Advances in fuzzy set theory and applications* (pp. 3–18). Amsterdam: North Holland.

Zadeh, L.A. (1996). Fuzzy logic = computing with words. *IEEE Transactions Fuzzy Systems, 4*, 103–111.

Zadeh, L.A. (2011). A note on Z-numbers. *Information Sciences, 181*, 2923–2932.

Zimmermann, H.J. (1991). *Fuzzy set theory and its applications*. Boston, MA: Kluwer Academic.