Journal of Mathematical Extension
Vol. 5, No. 2 (1), (2011), 47-66

Solving Fuzzy Linear System by Fuzzy Neural Network and Applications in Economics

M. Otadi
Islamic Azad University-Firoozkooh Branch
M. Mosleh
Islamic Azad University-Firoozkooh Branch
S. Abbasbandy
Islamic Azad University-Science and Research Tehran Branch

Abstract. In this paper, a novel hybrid method based on fuzzy neural network for estimate fuzzy coefficients (parameters) of fuzzy linear supply and demand function, is presented. Here a neural network is considered as a part of a large field called neural computing or soft computing. Moreover, in order to find the approximate parameters, a simple algorithm from the cost function of the fuzzy neural network is proposed.

AMS Subject Classification: 34k28; 97N40; 62M45; 82C32.
Keywords and Phrases: Fuzzy neural networks, Fuzzy number, learning algorithm.

1. Introduction

Supply and demand is an economic model of price determination in a market. The price p of a product is determined by a balance between production at each price (supply S) and the desires of those with purchasing power at each price (demand D). The market price of a good and the quantity produced are determined by the equality between supply

Received September 2010; Final Revised January 2011

and demand. Suppose that demand and supply are linear functions of the price ([20]):

$$\begin{cases} q_d = bp + c, \\ q_s = ep + f, \end{cases}$$

where q_s is the quantity supplied, which is required to be equal to q_d, the quantity requested, p is the fuzzy price and b, c, e and f are coefficients to be estimated, where the coefficients b, c, e and f are represented by fuzzy triangular numbers. By imposing the equality between quantity supplied and requested, the following general fuzzy linear system should be solved:

$$\begin{cases} x_1 = bx_2 + c, \\ x_1 = ex_2 + f. \end{cases}$$

We refer the reader to Ooldridje [23] for more information on supply and demand linear function.

The concept of fuzzy numbers and fuzzy arithmetic operations were first introduced by Zadeh ([25]), Dubois and Prade ([10]). We refer the reader to ([16]) for more information on fuzzy numbers and fuzzy arithmetic. One of the major applications of fuzzy number arithmetic is treating fuzzy linear systems and fuzzy regression models ([21,22,23]). Friedman *et al.* ([12]) introduced a general model for solving a fuzzy $n \times n$ linear system whose coefficient matrix is crisp and the right-hand side column is an arbitrary fuzzy number vector. They used the parametric form of fuzzy numbers and replaced the original fuzzy $n \times n$ linear system by a crisp $2n \times 2n$ linear system and studied duality in fuzzy linear systems $AX = BX + Y$ where A , B are real $n \times n$ matrix, the unknown vector X is vector consisting of n fuzzy numbers and the constant Y is vector consisting of n fuzzy numbers, in [13]. In [1, 2, 3, 4, 8] the authors presented conjugate gradient, LU decomposition method for solving general fuzzy linear systems or symmetric fuzzy linear systems. Also, Wang *et al.* ([24]) presented an iterative algorithm for solving dual linear system of the form $X = AX + U$, where A is real $n \times n$ matrix, the unknown vector X and the constant U are all vectors consisting of fuzzy numbers. In this paper, we first propose an architecture of fuzzy neural network (FNN) with fuzzy weights for fuzzy input vectors and fuzzy targets to find approximate solution to fully fuzzy linear systems like $Ax = Bx + d$,

where A, B are fuzzy matrices and d is a fuzzy vector.

2. Preliminaries

In this section the basic notations used in fuzzy calculus are introduced.

Definition 2.1. [15, 18]. *A fuzzy number u is a pair $(\underline{u}, \overline{u})$ of functions $\underline{u}(r)$ and $\overline{u}(r)$, $0 \leqslant r \leqslant 1$, which satisfy the following requirements:*

i. $\underline{u}(r)$ is a bounded monotonically increasing, left continuous function on $(0, 1]$ and right continuous at 0.

ii. $\overline{u}(r)$ is a bounded monotonically decreasing, left continuous function on $(0, 1]$ and right continuous at 0.

iii. $\underline{u}(r) \leqslant \overline{u}(r), 0 \leqslant r \leqslant 1$.

A crisp number r is simply represented by $\underline{u}(\alpha) = \overline{u}(\alpha) = r, 0 \ \leqslant \alpha \leqslant 1$. The set of all the fuzzy numbers is denoted by E^1.
A popular fuzzy number is the triangular fuzzy number $u = (u_m, u_l, u_r)$ where u_m denotes the modal value and the real values $u_l > 0$ and $u_r > 0$ represent the left and right fuzziness, respectively. The membership function of a triangular fuzzy number is defined by:

$$\mu_u(x) = \begin{cases} \frac{x-u_m}{u_l} + 1, & u_m - u_l \leqslant x \leqslant u_m, \\ \frac{u_m-x}{u_r} + 1, & u_m \leqslant x \leqslant u_m + u_r, \\ 0, & otherwise. \end{cases}$$

Its parametric form is

$$\underline{u}(\alpha) = u_m + u_l(\alpha - 1), \quad \overline{u}(\alpha) = u_m + u_r(1 - \alpha).$$

Triangular fuzzy numbers are fuzzy numbers in LR representation where the reference functions L and R are linear. The set of all triangular fuzzy numbers on $\mathbb{R}$ is called $\hat{FZ}$.

2.1 Operations on Fuzzy Numbers

We briefly mention fuzzy number operations defined by the extension principle ([25]). Since input vector of feedforward neural network is

fuzzified in this paper, the operations we use in our fuzzy neural network are fuzzified by means of the extension principle as follows:

$$\mu_{A+B}(z) = max\{\mu_A(x) \wedge \mu_B(y) | z = x + y\},$$

$$\mu_{AB}(z) = max\{\mu_A(x) \wedge \mu_B(y) | z = xy\},$$

$$\mu_{f(Net)}(z) = max\{\mu_{Net}(x) | z = f(x)\},$$

where A, B and Net are fuzzy numbers, $\mu_*(.)$ denotes the membership function of each fuzzy number, $\wedge$ is the minimum operator, and $f(.)$ is a continuous activation function (such as f(x)=x) inside units of our fuzzy neural network.

The above operations of fuzzy numbers are numerically performed on level sets. The h-level set of a fuzzy number X is defined by

$$[X]_h = \{x \in \mathbb{R} | \mu_X(x) \geqslant h\} \quad for \quad 0 < h \leqslant 1,$$

and $[X]_0 = \overline{\bigcup_{h\in(0,1]}[X]_h}$. Since level sets of fuzzy numbers become closed intervals, we denote $[X]_h$ by

$$[X]_h = [[X]_h^L, [X]_h^U],$$

where $[X]_h^L$ and $[X]_h^U$ are the lower and the upper limits of the h-level set $[X]_h$, respectively.

From interval arithmetic ([5]), the above operations on fuzzy numbers are written for h-level sets as follows:

$$A = B \Longleftrightarrow [A]_h = [B]_h \quad for \quad 0 < h \leqslant 1, \tag{1}$$

$$[A + B]_h = [[A]_h^L + [B]_h^L, [A]_h^U + [B]_h^U], \tag{2}$$

$$[A.B]_h = [[A]_h^L, [A]_h^U].[[B]_h^L, [B]_h^U] =$$

$$[min\{[A]_h^L.[B]_h^L, [A]_h^L.[B]_h^U, [A]_h^U.[B]_h^L, [A]_h^U.[B]_h^U\}, \tag{3}$$

$$max\{[A]_h^L.[B]_h^L, [A]_h^L.[B]_h^U, [A]_h^U.[B]_h^L, [A]_h^U.[B]_h^U\}],$$

$$f([Net]_h) = f([[Net]_h^L, [Net]_h^U]) = [f([Net]_h^L), f([Net]_h^U)], \tag{4}$$

where f is an increasing function. In the case of $0 \leqslant [A]_h^L \leqslant [A]_h^U$, (3) can be simplified as

$$[A.B]_h = [min\{[A]_h^L.[B]_h^L, [A]_h^L.[B]_h^U\}, max\{[A]_h^U.[B]_h^L, [A]_h^U.[B]_h^U\}]. \quad (5)$$

The result of a fuzzy addition of triangular fuzzy numbers is a triangular fuzzy number again. So we only have to compute the following equation:

$$(a_m, a_l, a_r) + (b_m, b_l, b_r) = (a_m + b_m, a_l + b_l, a_r + b_r) \quad (6)$$

Considering the fuzzy multiplication, some computational expense problems can be investigated. The result of a fuzzy multiplication is a fuzzy number in LR representation, but it is difficult to compute the new functions L and R because they are not necessarily linear. We approximate this fuzzy multiplication such that it computes a triangular fuzzy number too. This fuzzy multiplication is denoted by $\hat{*}$ ([11]).
This fuzzy multiplication is based on the extension principle but is a bit different from the classical fuzzy multiplication. We compute our operation by the following equation:

$$(a_m, a_l, a_r)\hat{*}(b_m, b_l, b_r) = (c_m, c_l, c_r) \quad (7)$$

with

$$c_m = a_m.b_m, c_l = c_m - c_\lambda, c_r = c_\rho - c_m,$$
$$c_\lambda := min(a_\lambda.b_\lambda, a_\lambda.b_\rho, a_\rho.b_\lambda, a\rho.b_\rho)$$
$$c_\rho := max(a_\lambda.b_\lambda, a_\lambda.b_\rho, a_\rho.b_\lambda, a\rho.b_\rho),$$

where $a_\lambda = a_m - a_l$ and $a_\rho = a_m + a_r$. a_λ and a_ρ denote the left and right limits of the support of fuzzy the number a.
The use of these fuzzy operations has some advantages:

- The distributivity of these operations is retained. This is very important for our theoretical examinations.
- The computational expense is acceptable.
- The idea of fuzzy sets is preserved even if a fuzzy number is characterized by only three values.

We describe the classical definition of distance between fuzzy numbers [11]:

Definition 2.1.1. *For arbitrary fuzzy number* $u = (u_m + u_l(\alpha - 1), u_m + u_r(1-\alpha))$ *and real positive number* k*, we define a new operation* $\circledcirc$ *such that:*

$$(u_m + u_l(\alpha-1), u_m + u_r(1-\alpha)) \circledcirc k = (-k(u_m + u_l(\alpha-1)), -k(u_m + u_r(1-\alpha))).$$

Definition 2.1.2. *The mapping* $\hat{d} : \hat{FZ} \times \hat{FZ} \longrightarrow \mathbb{R}^+$ *is defined by*

$$\hat{d}(A, B) = max(|a_m - b_m|, |a_\lambda - b_\lambda|, |a_\rho - b_\rho|),$$

where $A = (a_m, a_l, a_r)$ *and* $B = (b_m, b_l, b_r)$*. It can be proved that* $\hat{d}$ *is a metric on* $\hat{FZ}$ *and so* $(\hat{FZ}, \hat{d})$ *becomes a metric space.*

3. Fully Fuzzy Linear Systems and Applications in Economics

We are interested in finding approximate solution of fuzzy linear equation systems of the form

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_{11}x_1 + b_{12}x_2 + \cdots + b_{1n}x_n + d_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_{21}x_1 + b_{22}x_2 + \cdots + b_{2n}x_n + d_2, \\ \quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_{n1}x_1 + b_{n2}x_2 + \cdots + b_{nn}x_n + d_n. \end{cases} \tag{8}$$

Let $A = (a_{ij})_{n\times n}$, $B = (b_{ij})_{n\times n}$, $d = (d_1, d_2, \ldots, d_n)^T$ and $x = (x_1, x_2, \ldots, x_n)^T$. Then Eq.(8) can be written as:

$$Ax = Bx + d, \tag{9}$$

where A and B are two square matrices of fuzzy number entries and the unknown x and the known d are two vectors whose components are n fuzzy numbers.

Usually, there is no inverse element for an arbitrary fuzzy number $u \in E^1$, i.e., there exists no element $v \in E^1$ such that

$$u + v = 0.$$

Actually, for all non-crisp fuzzy number $u \in E^1$ we have

$$u + (-u) \neq 0.$$

Therefore, the fuzzy linear equation system (9) cannot be equivalently replaced by the fuzzy linear equation system

$$(A - B)x = d$$

which had been investigated. In the sequel, we will call the fuzzy linear system (9), a fully fuzzy linear system.
How does the FNN_3 (fuzzy neural network with fuzzy input, output signals and fuzzy weights) solve the dual fully fuzzy linear systems? First let us build a FNN_3 equivalent to the fully fuzzy linear system. The network is shown in Fig.1. Neurons $1, 2, \ldots, 2n$ have output equal to input, and neurons $2n + 1$ and $2n + 3$ add their inputs to produce output.

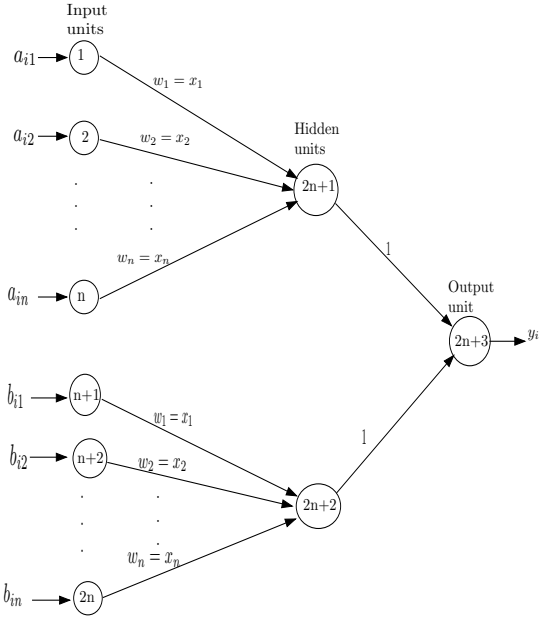

Fig. 1. Fuzzy neural network for solving fully fuzzy linear systems.

Now suppose that we do not know $x = (x_1, x_2, \ldots, x_n)$ and we try to find $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T$ such that if

$$Ax = Bx + y, \tag{10}$$

so that the output from the network y_i, when the inputs are

$$a_{i1}, a_{i2}, \ldots, a_{in}, b_{i,1}, b_{i2}, \ldots, b_{in},$$

is approximately d_i, all i, i.e.,

$$min \ \|[y]_h^L - [d]_h^L\| \ \ and \ \ min \ \|[y]_h^U - [d]_h^U\|, \quad h \in [0,1], \tag{11}$$

or

$$min \ \|[Ax]_h^L - [Bx]_h^L - [d]_h^L\| \ \ and \ \ min \ \|[Ax]_h^U - [Bx]_h^U - [d]_h^U\|, \quad h \in [0,1],$$

or

$$min \ \ \hat{d}(y_i, d_i) \quad for \ \ all \ \ i = 1, 2, \ldots, n, \tag{12}$$

then, it becomes a problem of optimization. Therefore, neuron $2n + 2$ add their inputs and use $\circledcirc$ for $k = 1$.

3.1 Input-Output Relation of Each Unit

Let us fuzzify a three-layer feedforward neural network with $2n$ input units, two hidden units and one output unit. Input vectors, targets and connection weights are fuzzified (i.e., extended to fuzzy numbers). In order to derive a crisp learning rule, we restrict fuzzy weights, fuzzy inputs and fuzzy target within triangular fuzzy numbers.
The input-output relation of each unit of the fuzzified neural network can be written as follows:

- Input units:

$$\begin{array}{ll} o_{ij} = a_{ij}, & i, j = 1, 2, \ldots, n, \\ m_{ij} = b_{ij}, & i, j = 1, 2, \ldots, n. \end{array} \tag{13}$$

- Hidden units:

$$\begin{aligned} &f(U_i) = U_i, \\ &U_i = o_{i1}\hat{*}w_1 + \ldots + o_{in}\hat{*}w_n, \qquad i = 1, 2, \ldots, n, \end{aligned} \tag{14}$$

and

$$\begin{aligned} &g(V_i) = V_i \circledcirc 1, \\ &V_i = m_{i1}\hat{*}w_1 + \ldots + m_{in}\hat{*}w_n, \qquad i = 1, 2, \ldots, n. \end{aligned} \tag{15}$$

- Output unit:

$$y_i = f(U_i) + g(V_i), \tag{16}$$

where a_{ij} and b_{ij} are fuzzy inputs and w_j is a fuzzy weight (see Fig.1).

3.2 Learning Fuzzy Neural Network

Consider the learning algorithm of the fuzzy feedforward neural network as shown in Figure 1. Let the h-level sets of the target output $d_i, i = 1, \ldots, n$ be denoted

$$[d_i]_h = [[d_i]_h^L, [d_i]_h^U], \quad i = 1, \ldots, n, \tag{17}$$

where $d_i^L(h)$ shows the left-hand side and $d_i^U(h)$ the right-hand side of the h-level sets of the desired output.
A cost function to be minimized is defined for each h-level sets as follows:

$$[E(w_1, \ldots, w_n)]_h = [E(w_1, \ldots, w_n)]_h^L + [E(w_1, \ldots, w_n)]_h^U, \tag{18}$$

where

$$[E(w_1, \ldots, w_n)]_h^L = \frac{1}{2}\sum_{i=1}^{n}([y_i]_h^L - [d_i]_h^L)^2,$$

$$[E(w_1, \ldots, w_n)]_h^U = \frac{1}{2}\sum_{i=1}^{n}([y_i]_h^U - [d_i]_h^U)^2.$$

Hence $[E(w_1, \ldots, w_n)]_h^L$ denotes the error between the left-hand sides of the h-level sets of the desired and the computed output, and $[E(w_1, \ldots, w_n)]_h^U$ denotes

the error between the right-hand sides of the h-level sets of the desired and the computed output.
In the research of neural networks, the norm $\|.\|$ is often defined as follows:

$$[E(w_1,\ldots,w_n)]_h^L = \frac{1}{2}\sum_{i=1}^{n}([y_i]_h^L - [d_i]_h^L)^2 = \frac{1}{2}\sum_{i=1}^{n}(\sum_{j=1}^{n}[a_{ij}w_j]_h^L - \sum_{j=1}^{n}[b_{ij}w_j]_h^L - [d_i]_h^L)^2, \tag{19}$$

$$[E(w_1,\ldots,w_n)]_h^U = \frac{1}{2}\sum_{i=1}^{n}([y_i]_h^U - [d_i]_h^U)^2 = \frac{1}{2}\sum_{i=1}^{n}(\sum_{j=1}^{n}[a_{ij}w_j]_h^U - \sum_{j=1}^{n}[b_{ij}w_j]_h^U - [d_i]_h^U)^2.$$

Clearly, this is a problem of optimization of quadratic functions without constrains that can usually be solved by gradient descent algorithm. In fact, denoting

$$[\nabla E(W)]_h^L = ([\frac{\partial E(W)}{\partial w_1}]_h^L,\ldots,[\frac{\partial E(W)}{\partial w_n}]_h^L)^T,$$

$$[\nabla E(W)]_h^U = ([\frac{\partial E(W)}{\partial w_1}]_h^U,\ldots,[\frac{\partial E(W)}{\partial w_n}]_h^U)^T,$$

in order to solve equation (11), assume k iterations to have been done and get the k^{th} iteration point W_k.

Remark 3.2.1. Since the equations (19) are quadratic functions, supposing $0 \leqslant [a_{ij}]_h^L \leqslant [a_{ij}]_h^U$, $0 \leqslant [b_{ij}]_h^L \leqslant [b_{ij}]_h^U$ and $0 \leqslant [w_j]_h^L \leqslant [wj]_h^U$ for $i,j = 1,\ldots,n$, we rewrite them as follows:

$$[E(W)]_h^L = \tfrac{1}{2}\textstyle\sum_{i=1}^{n}(\sum_{j=1}^{n}[a_{ij}w_j]_h^L - \sum_{j=1}^{n}[b_{ij}w_j]_h^L - [d_i]_h^L)^2$$

$$= \tfrac{1}{2}([W]_h^L)^T[Q]_h^L[W]_h^L + \tfrac{1}{2}([W]_h^L)^T[P]_h^L[W]_h^L + [C]_h^L$$

$$+([W]_h^L)^T[\acute{Q}]_h^L[W]_h^L + ([\acute{A}]_h^L)^T[W]_h^L + ([\acute{B}]_h^L)^T[W]_h^L,$$

where

$$[Q]_h^L = [(q_{ij})_{n\times n}]_h^L, \qquad [\acute{Q}]_h^L = [(\acute{q}_{ij})_{n\times n}]_h^L,$$

$$[P]_h^L = [(p_{ij})_{n\times n}]_h^L, \qquad [\acute{B}]_h^L = ([\acute{b}_1]_h^L, [\acute{b}_2]_h^L, \ldots, [\acute{b}_n]_h^L)^T,$$

$$[\acute{A}]_h^L = ([\acute{a}_1]_h^L, [\acute{a}_2]_h^L, \ldots, [\acute{a}_n]_h^L)^T, \qquad [C]_h^L = \tfrac{1}{2}\sum_{i=1}^n ([d_i]_h^L)^2,$$

$$[q_{ij}]_h^L = \sum_{k=1}^n [a_{ki}]_h^L [a_{kj}]_h^L, \qquad [p_{ij}]_h^L = \sum_{k=1}^n [b_{ki}]_h^L [b_{kj}]_h^L,$$

$$[\acute{q}_{ij}]_h^L = -\sum_{k=1}^n [a_{ki}]_h^L [b_{kj}]_h^L,$$

with $[q_{ij}]_h^L = [q_{ji}]_h^L$, $[p_{ij}]_h^L = [p_{ji}]_h^L$, $[\acute{a}_i]_h^L = -\sum_{k=1}^n [a_{ki}]_h^L [d_k]_h^L$ and $[\acute{b}_i]_h^L =$

$\sum_{k=1}^n [b_{ki}]_h^L [d_k]_h^L$.

We have

$$[\nabla E(W)]_h^L = [Q]_h^L [W]_h^L + [P]_h^L [W]_h^L + [Q'']_h^L [W]_h^L + ([Q''']_h^L)^T [W]_h^L + [\acute{A}]_h^L + [\acute{B}]_h^L, \tag{20}$$

where

$$[Q'']_h^L = [(q''_{ij})_{n\times n}]_h^L, \qquad [Q''']_h^L = [(q'''_{ij})_{n\times n}]_h^L,$$

$$[q''_{ij}]_h^L = \begin{cases} 2[\acute{q}_{ij}]_h^L, & i = j, \\ [\acute{q}_{ij}]_h^L, & i \neq j, \end{cases}$$

$$[q'''_{ij}]_h^L = \begin{cases} 0, & i = j, \\ [\acute{q}_{ij}]_h^L, & i \neq j, \end{cases}$$

and

$$[E(W)]_h^U = \tfrac{1}{2}\sum_{i=1}^n (\sum_{j=1}^n [a_{ij}w_j]_h^U - \sum_{j=1}^n [b_{ij}w_j]_h^U - [d_i]_h^U)^2$$

$$= \tfrac{1}{2}([W]_h^U)^T [Q]_h^U [W]_h^U + \tfrac{1}{2}([W]_h^U)^T [P]_h^U [W]_h^U + [C]_h^U$$

$$+([W]_h^U)^T [\acute{Q}]_h^U [W]_h^U + ([\acute{A}]_h^U)^T [W]_h^U + ([\acute{B}]_h^U)^T [W]_h^U,$$

where

$$[Q]_h^U = [(q_{ij})_{n\times n}]_h^U, \qquad [\acute{Q}]_h^U = [(\acute{q}_{ij})_{n\times n}]_h^U,$$

$$[P]_h^U = [(p_{ij})_{n\times n}]_h^U, \qquad [\acute{B}]_h^U = ([\acute{b}_1]_h^U, [\acute{b}_2]_h^U, \ldots, [\acute{b}_n]_h^U)^T,$$

$$[\acute{A}]_h^U = ([\acute{a}_1]_h^U, [\acute{a}_2]_h^U, \ldots, [\acute{a}_n]_h^U)^T, \qquad [C]_h^U = \tfrac{1}{2}\sum_{i=1}^n ([d_i]_h^U)^2,$$

$$[q_{ij}]_h^U = \sum_{k=1}^n [a_{ki}]_h^U [a_{kj}]_h^U, \qquad [p_{ij}]_h^U = \sum_{k=1}^n [b_{ki}]_h^U [b_{kj}]_h^U,$$

$$[\acute{q}_{ij}]_h^U = -\sum_{k=1}^n [a_{ki}]_h^U [b_{kj}]_h^U,$$

with $[q_{ij}]_h^U = [q_{ji}]_h^U$, $[p_{ij}]_h^U = [p_{ji}]_h^U$, $[\acute{a}_i]_h^U = -\sum_{k=1}^n [a_{ki}]_h^U [d_k]_h^U$ and $[\acute{b}_i]_h^U =$

$\sum_{k=1}^n [b_{ki}]_h^U [d_k]_h^U$.

We have

$$[\nabla E(W)]_h^U = [Q]_h^U [W]_h^U + [P]_h^U [W]_h^U + [Q'']_h^U [W]_h^U + ([Q''']_h^U)^T [W]_h^U + [\acute{A}]_h^U + [\acute{B}]_h^U, \tag{21}$$

where

$$[Q'']_h^U = [(q''_{ij})_{n\times n}]_h^U, \qquad [Q''']_h^U = [(q'''_{ij})_{n\times n}]_h^U,$$

$$[q''_{ij}]_h^U = \begin{cases} 2[\acute{q}_{ij}]_h^U, & i = j, \\ [\acute{q}_{ij}]_h^U, & i \neq j, \end{cases}$$

$$[q'''_{ij}]_h^U = \begin{cases} 0, & i = j, \\ [\acute{q}_{ij}]_h^U, & i \neq j. \end{cases}$$

To find the stationary point of $[E(W)]_h = ([E(W)]_h^L, [E(W)]_h^U)$, we should put

$$[\nabla E(W)]_h^L = [\nabla E(W)]_h^U = 0 \triangleq (0, 0, \ldots, 0)^T.$$

When

$$[Q]_h^L + [P]_h^L + [Q'']_h^L + ([Q''']_h^L)^T$$

and

$$[[Q]_h^U + [P]_h^U + [Q'']_h^U + ([Q''']_h^U)^T$$

are positive definite matrices, the stationary point can be obtained as follows ([17]):

$$[W^*]_h^L = -([Q]_h^L + [P]_h^L + [Q'']_h^L + ([Q''']_h^L)^T)^{-1}(-[\acute{A}]_h^L - [\acute{B}]_h^L), \quad (22)$$

$$[W^*]_h^U = -([Q]_h^U + [P]_h^U + [Q'']_h^U + ([Q''']_h^U)^T)^{-1}(-[\acute{A}]_h^U - [\acute{B}]_h^U).$$

The Hessian matrices at this point are

$$[\nabla^2 E(W^*)]_h^L = [\nabla(\nabla E(W^*))]_h^L = [Q]_h^L + [P]_h^L + [Q'']_h^L + ([Q''']_h^L)^T,$$

and

$$[\nabla^2 E(W^*)]_h^U = [\nabla(\nabla E(W^*))]_h^U = [Q]_h^U + [P]_h^U + [Q'']_h^U + ([Q''']_h^U)^T,$$

if $[\nabla^2 E(W^*)]_h^L$ and $[\nabla^2 E(W^*)]_h^U$ be positive definite matrices, from optimization theory, we known that

$$[W^*]_h = ([W^*]_h^L, [W^*]_h^U) = (-([Q]_h^L + [P]_h^L + [Q'']_h^L + ([Q''']_h^L)^T)^{-1}(-[\acute{A}]_h^L - [\acute{B}]_h^L),$$
$$-([Q]_h^U + [P]_h^U + [Q'']_h^U + ([Q''']_h^U)^T)^{-1}(-[\acute{A}]_h^U - [\acute{B}]_h^U)),$$

is the unique solution of the problem.

Remark 3.2.2. The above method is not very convenient in applications. Now we consider its explicit scheme. Since

$$[\nabla E(W)]_h^L = [Q]_h^L[W]_h^L + [P]_h^L[W]_h^L + [Q'']_h^L[W]_h^L + ([Q''']_h^L)^T[W]_h^L + [\acute{A}]_h^L + [\acute{B}]_h^L$$

and

$$[\nabla E(W)]_h^U = [Q]_h^U[W]_h^U + [P]_h^U[W]_h^U + [Q'']_h^U[W]_h^U + ([Q''']_h^U)^T[W]_h^U + [\acute{A}]_h^U + [\acute{B}]_h^U,$$

then

$$[\nabla E(W_k)]_h^L = [Q]_h^L[W_k]_h^L + [P]_h^L[W_k]_h^L + [Q'']_h^L[W_k]_h^L + ([Q''']_h^L)^T[W_k]_h^L + [\acute{A}]_h^L + [\acute{B}]_h^L$$

and

$$[\nabla E(W_k)]_h^U = [Q]_h^U[W_k]_h^U + [P]_h^U[W_k]_h^U + [Q'']_h^U[W_k]_h^U + ([Q''']_h^U)^T[W_k]_h^U + [\acute{A}]_h^U + [\acute{B}]_h^U.$$

We know that [17]

$$([\nabla E(W_{k+1})]_h^L)^T[\nabla E(W_k)]_h^L = 0, \quad ([\nabla E(W_{k+1})]_h^U)^T[\nabla E(W_k)]_h^U = 0,$$

therefore we have

$$\{[Q]_h^L([W_k]_h^L - [\mu_k]_h^L[\nabla E(W_k)]_h^L) + [P]_h^L([W_k]_h^L - [\mu_k]_h^L[\nabla E(W_k)]_h^L) + [Q'']_h^L([W_k]_h^L - [\mu_k]_h^L[\nabla E(W_k)]_h^L) + ([Q''']_h^L)^T([W_k]_h^L - [\mu_k]_h^L[\nabla E(W_k)]_h^L) + [\acute{A}]_h^L + [\acute{B}]_h^L\}^T\{[Q]_h^L[W_k]_h^L + [P]_h^L[W_k]_h^L + [Q'']_h^L[W_k]_h^L + ([Q''']_h^L)^T[W_k]_h^L + [\acute{A}]_h^L + [\acute{B}]_h^L\} = 0$$

and

$$\{[Q]_h^U([W_k]_h^U - [\mu_k]_h^U[\nabla E(W_k)]_h^U) + [P]_h^U([W_k]_h^U - [\mu_k]_h^U[\nabla E(W_k)]_h^U) + [Q'']_h^U([W_k]_h^U - [\mu_k]_h^U[\nabla E(W_k)]_h^U) + ([Q''']_h^U)^T([W_k]_h^U - [\mu_k]_h^U[\nabla E(W_k)]_h^U) + [\acute{A}]_h^U + [\acute{B}]_h^U\}^T\{[Q]_h^U[W_k]_h^U + [P]_h^U[W_k]_h^U + [Q'']_h^U[W_k]_h^U + ([Q''']_h^U)^T[W_k]_h^U + [\acute{A}]_h^U + [\acute{B}]_h^U\} = 0$$

Rearranging them, we have:

$$([\nabla E(W_k)]_h^L - [\mu_k]_h^L[Q]_h^L[\nabla E(W_k)]_h^L - [\mu_k]_h^L[P]_h^L[\nabla E(W_k)]_h^L - [\mu_k]_h^L[Q'']_h^L[\nabla E(W_k)]_h^L - [\mu_k]_h^L([Q''']_h^L)^T[\nabla E(W_k)]_h^L)^T[\nabla E(W_k)]_h^L = 0,$$

and

$$([\nabla E(W_k)]_h^U - [\mu_k]_h^U[Q]_h^U[\nabla E(W_k)]_h^U - [\mu_k]_h^U[P]_h^U[\nabla E(W_k)]_h^U - [\mu_k]_h^U[Q'']_h^U[\nabla E(W_k)]_h^U - [\mu_k]_h^U([Q''']_h^U)^T[\nabla E(W_k)]_h^U)^T[\nabla E(W_k)]_h^U = 0.$$

From these equations, we can easily get an expression for $[\mu_k]_h^L$ and $[\mu_k]_h^U$:

$$[\mu_k]_h^L = \frac{([\nabla E(W_k)]_h^L)^T[\nabla E(W_k)]_h^L}{[\Gamma]_h^L} \tag{23}$$

and

$$[\mu_k]_h^U = \frac{([\nabla E(W_k)]_h^U)^T[\nabla E(W_k)]_h^U}{[\Gamma]_h^U}. \tag{24}$$

where

$$[\Gamma]_h^L = ([\nabla E(W_k)]_h^L)^T[Q]_h^L[\nabla E(W_k)]_h^L + ([\nabla E(W_k)]_h^L)^T[P]_h^L[\nabla E(W_k)]_h^L +$$

$$([\nabla E(W_k)]_h^L)^T([Q'']_h^L)^T[\nabla E(W_k)]_h^L + ([\nabla E(W_k)]_h^L)^T[Q''']_h^L[\nabla E(W_k)]_h^L,$$

and

$$[\Gamma]_h^U = ([\nabla E(W_k)]_h^U)^T[Q]_h^U[\nabla E(W_k)]_h^U + ([\nabla E(W_k)]_h^U)^T[P]_h^U[\nabla E(W_k)]_h^U +$$

$$([\nabla E(W_k)]_h^U)^T([Q'']_h^U)^T[\nabla E(W_k)]_h^U + ([\nabla E(W_k)]_h^U)^T[Q''']_h^U[\nabla E(W_k)]_h^U.$$

Substituting these into equations, we obtain ([14])

$$\begin{array}{l} W_{k+1} = W_k + \Delta W_k, \\ \Delta W_k = -\mu_k \nabla E(W_k), \end{array} \tag{25}$$

where k indexes the number of adjustments and $\mu_k = ([\mu_k]_h^L, [\mu_k]_h^U)$ is a learning rate, we have the explicit scheme

$$[W_{k+1}]_h^L = [W_k]_h^L - \frac{([\nabla E(W_k)]_h^L)^T[\nabla E(W_k)]_h^L}{[\Gamma]_h^L}[\nabla E(W_k)]_h^L, \tag{26}$$

and

$$[W_{k+1}]_h^U = [W_k]_h^U - \frac{([\nabla E(W_k)]_h^U)^T[\nabla E(W_k)]_h^U}{[\Gamma]_h^U}[\nabla E(W_k)]_h^U. \tag{27}$$

We can also obtain similar relations for $[a_{ij}]_h^L \leqslant [a_{ij}]_h^U \leqslant 0$ and $[w_j]_h^L \leqslant [wj]_h^U \leqslant 0$, $i, j = 1, \ldots, n$, and other cases.
The fully fuzzy linear system may have no solution. In this case there is no hope to make the error measure close to zero.

3.3 Estimation of Fuzzy Linear Demand and Supply

Suppose that demand and supply are linear functions of the price:

$$\left\{ \begin{array}{l} q_d = b * p + c, \\ q_s = e * p + f, \end{array} \right.$$

where q_s is the quantity supplied, which is required to be equal to q_d, the quantity requested, p is the fuzzy price and b, c, e and f are coefficients to be estimated, where the coefficients b, c, e and f are represented by fuzzy triangular numbers. By imposing the equality between quantity supplied and requested, the following general dual fuzzy linear system should be solved:

$$\begin{cases} x_1 = bx_2 + c, \\ x_1 = ex_2 + f. \end{cases}$$

Shortcoming of the Existing Methods

In this section, the shortcoming of the existing methods ([1, 2, 3, 4, 8, 6, 7, 12, 13, 24]) for solving fuzzy linear systems are pointed out.

1. Friedman *et al.* ([12, 13]) introduced a general model for solving a fuzzy $n \times n$ linear system whose coefficient matrix is crisp and the right-hand side column is an arbitrary fuzzy number vector also in ([1, 2, 3, 4, 8, 6, 7]) the authors presented conjugate gradient, LU decomposition method for solving general fuzzy linear systems or symmetric fuzzy linear systems. Also, Wang *et al.* ([24]) presented an iterative algorithm for solving dual linear system of the form $X = AX + U$, where A is real $n \times n$ matrix, the unknown vector X and the constant U are all vectors consisting of fuzzy numbers. The existing methods ([1, 2, 3, 4, 8, 6, 7, 12, 13, 24]) are applicable only if all the elements of the coefficient matrix are real numbers, e.g., it is not possible to find the solution of FFLS, chosen in following Example.

2. Dehghan et al. ([9]) considered fully fuzzy linear systems (FFLS) of the form $\tilde{A} \otimes \tilde{x} = \tilde{b}$ where $\tilde{A}$ is a fuzzy $n \times n$ matrix, the unknown vector $\tilde{x}$ is a vector consisting of n fuzzy numbers and the constant $\tilde{b}$ is a vectors consisting of n fuzzy numbers . The existing method ([9]) is applicable only if all the elements of the coefficient matrix and right hand side vector are non-negative fuzzy numbers, e.g., it is not possible to find the solution of FFLS, chosen in following Example by using the existing method ([9]) due to the existence

of $-(3,1,1)$ which is not the non-negative fuzzy number.

In this paper, we suppose that $b = -(3,1,1), c = (16,1,2), e = (2,1,1)$ and $f = (6,4,1)$. In order to get the classical solution, we solve the following system:

$$\begin{cases} x_1 = -(3,1,1)x_2 + (16,1,2), \\ x_1 = (2,1,1)x_2 + (6,4,1). \end{cases}$$

The training starts with $W_1(1) = (7, 0.5, 0.5)$,

$$W_2(1) = (1, 0.3, 0.2)$$

We get approximate solution of fuzzy linear system of demand and supply in Matlab code $W_1(15) = (10, 6.9989, 6.0010), W_2(15) = (2, 0.9991, 1.001)$.
We can apply this method, for another case study.

3.4 Algorithm

Step 1: Read K (number of iterations), w_j (fuzzy weights w_j are initialized values), $(a_{i1}, \ldots, a_{in}, b_{i1}, \ldots, b_{in}, d_1, ..., d_n)$ for $i = 1, 2, ..., n$.
Step 2: Compute

$$[E(w_1, \ldots, w_n)]_h = [E(w_1, \ldots, w_n)]_h^L + [E(w_1, \ldots, w_n)]_h^U.$$

Step 3: Compute $\mu_k = ([\mu_k]_h^L, [\mu_k]_h^U)$ is a learning rate.
Step 4: The fuzzy weight $W^T = (w_1, \ldots, w_n)^T$ is updated by the Eq. (26) or Eq. (27).
Step 5: If $k < K$ then $k := k + 1$ and we continue the training by going back to step 2, otherwise we go to step 6.
Step 6: The training cycle is completed.

References

[1] S. Abbasbandy and M. Alavi, *A method for solving fuzzy linear systems, Iranian J. fuzzy systems,* 2 (2005), 37-43.

[2] S. Abbasbandy and M. Alavi, A new method for solving symmetric fuzzy linear systems, *Mathematics Scientific Journal, Islamic Azad University of Arak,* 1 (2005), 55-62.

[3] S. Abbasbandy, A. Jafarian, and R. Ezzati, Conjugate gradient method for fuzzy symmetric positive definite system of linear equations, *Appl. Math. Comput.,* 171 (2005), 1184-1191.

[4] S. Abbasbandy, R. Ezzati, and A. Jafarian, LU decomposition method for solving fuzzy system of linear equations, *Appl. Math. Comput.,* 172 (2006), 633-643.

[5] G. Alefeld and J. Herzberger, *Introduction to Interval Computations,* Academic Press, New York, 1983.

[6] T. Allahviranloo, Numerical methods for fuzzy system of linear equations, *Appl. Math. Comput.,* 155 (2004), 493-502.

[7] T. Allahviranloo, Successive over relaxation iterative method for fuzzy system of linear equations, *Appl. Math. Comput.,* 162 (2005), 189-196.

[8] B. Asady, S. Abbasbandy, and M. Alavi, Fuzzy general linear systems, *Appl. Math. Comput.,* 169 (2005), 34-40.

[9] M. Dehghan, B. Hashemi, and M. Ghatee, Solution of the fully fuzzy linear systems using iterative techniques, *Chaos Solitons & Fractals,* 34 (2007), 316-336.

[10] D. Dubois and H. Prade, Operations on fuzzy numbers, *J. Systems Sci.,* 9 (1978), 613-626.

[11] TH. Feuring and W. M. Lippe, Fuzzy neural networks are universal approximators, *IFSA World Congress, Sao Paulo, Brasil,* 2 (1995), 659-662.

[12] M. Friedman, M. Ming, and A. Kandel, Fuzzy linear systems, *Fuzzy Sets and Systems,* 96 (1998), 201-209.

[13] M. Friedman, M. Ming, and A. Kandel, Duality in fuzzy linear systems, *Fuzzy Sets and Systems,* 109 (2000), 55-58.

[14] H. Ishibuchi and M. Nii, Numerical analysis of the learning of fuzzified neural networks from fuzzy if-then rules, *Fuzzy Sets and Systems,* 120 (2001), 281-307.

[15] O. Kaleva, Fuzzy differential equations, *Fuzzy Sets and Systems,* 24 (1987), 301-317.

[16] A. Kaufmann and M. M. Gupta, *Introduction Fuzzy Arithmetic,* Van Nostrand Reinhold, New York, 1985.

[17] H. X. Li, L. X. Li, and J. Y. Wang, Interpolation functions of feedforward neural networks, *Computers and mathematics with applications,* 46 (2003), 1861-1874.

[18] M. Ma, M. Friedman, and A. Kandel, A new fuzzy arithmetic, *Fuzzy Sets and Systems,* 108 (1999), 83-90.

[19] J. Ooldridje, *Introductry econometrics a modern approach,* Tompson, Edition 2, 2003.

[20] H. Varian, Micro economic analysis, *Norton, Edition 3*, 1992.

[21] M. Mosleh, M. Otadi, and S. Abbasbandy, Evaluation of fuzzy regression models by fuzzy neural network, *Journal of computational and Applied Mathematics,* 234 (2010), 825-834.

[22] M. Mosleh, M. Otadi, and S. Abbasbandy, Fuzzy polynomial regression with fuzzy neural networks, *Applied Mathematical Modelling,* 35 (2011), 5400-5412.

[23] M. Mosleh, T. Allahviranloo, and M. Otadi, Evaluation of fully fuzzy regression models by fuzzy neural network, *Neural Comput and Applications,* In press.

[24] X. Wang, Z. Zhong, and M. Ha, Iteration algorithms for solving a system of fuzzy linear equations, *Fuzzy Sets and Systems,* 119 (2001), 121-128.

[25] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, *Inform. Sci.,* 8 (1975), 199-249.

Mahmood Otadi
Department of Mathematics
Assistant Professor of Mathematics
Islamic Azad University-Firoozkooh Branch
Firoozkooh, Iran
E-mail: otadi@iaufb.ac.ir

Maryam Mosleh
Department of Mathematics
Assistant Professor of Mathematics
Islamic Azad University-Firoozkooh Branch
Firoozkooh, Iran.
E-mail: mosleh@iaufb.ac.ir

Saeed Abbasbandy
Department of Mathematics
Science and Research Brench
Professor of Mathematics
Islamic Azad University-Tehran Branch
Tehran, Iran.
E-mail: abassbandy@yahoo.com