Int. J. Adv. Appl. Math. and Mech. 2(1) (2014) 53 - 63
ISSN: 2347-2529
Available online at www.ijaamm.com

International Journal of Advances in **Applied Mathematics and Mechanics**

Fuzzy nonlinear regression using artificial neural networks

Research Article

Purnima K. Pandit *

Department of Applied Mathematics, Faculty of Technology and Engineering, The M. S. University of Baroda, Vadodara, Gujarat, India

Received 18 July 2014; accepted (in revised version) 27 August 2014

Abstract: Fuzzy linear regression analysis with symmetric triangular fuzzy number coefficient has been introduced by Tanaka et al. In this work we propose to approximate the fuzzy nonlinear regression using Artificial Neural Networks. The working of the proposed method is illustrated by the case study with the data for temperature and evaporation for the IARI New Delhi division.

MSC: 62J86

Keywords: Fuzzy number • Nonlinear fuzzy regression • Artificial Neural Network

1. Introduction

In many real life phenomena one is required to estimate the functional dependency of one variable on the other. Regression analysis is the most popular tool for achieving such estimate. While collecting the data for the purpose of estimation there may be the possibilistic deviations in the measurements, such data can be more efficiently modeled using fuzzy numbers. Fuzzy regression was first introduced by Tanaka et al [16] in 1982. Thereafter, many investigations have focused on different variants of fuzzy regression, their properties and applications [[1], [2], [7], [10], [12], [15]].

The fuzzy regression models can be broadly classified into two classes, one of which uses fuzzy least square method (FLSM) to minimize the total square of errors as in P. Diamond [4] , M. Ming and et al. [6]. Another class includes Tanaka's method and its extension which minimizes the total vagueness of the estimated values of the dependent variables. Tanaka's model simply uses the programming and computation and FLSM uses minimum degree of fuzziness between the observed and estimated values.

In the proposed method we examine two variants of the relationship:

1. The independent variable is fuzzy whereas the dependent variable is crisp, that is:

$$y = f(\tilde{x}) \tag{1}$$

2. Secondly when both the dependent as well as the independent variables are fuzzy:

$$\tilde{y} = f(\tilde{x}) \tag{2}$$

* Corresponding author.
E-mail address: pkpandit@yahoo.com

Practically, there are mathematical relationships for which the fuzzy setup is more feasible, we take up one such scenario for study. In the irrigation domain the investigation of the effect of the parameters like sunshine hours, rainfall, wind speed and temperature on evaporation of water from soil is very vital as it directly affects the crop yield. We would take up the sub-problem wherein the influence of temperature (T) on evaporation (E) is studied. The data for study is downloaded from, http://www.iari.res.in

Since there is a variation in temperature throughout the day, instead of representing it by its average value, it would be more practical to represent it by a fuzzy number having support from minimum temperature to maximum temperature. Thus the relationship, for evaporation, which is to be estimated is given as, $E = f(\tilde{T})$, where, f is any nonlinear function and $\tilde{T}$ is temperature represented by fuzzy number.

We propose the use of Artificial Neural Network to give the best approximate of the above mentioned relationship. Such an implementation of the relationship is useful due to two reasons. Firstly, it is not required to know the nature of relationship a priori. Secondly, there is no need to give any functional form to such a relationship as done in regression. ANN implementation for fuzzy regression would be beneficial as they can easily and efficiently handle even the nonlinear relationship between the variables giving us better predictability and computations.

The paper starts with the brief of the required preliminaries on the fuzzy sets. Then the next section describes ANN. In fourth section we discuss the types of fuzzy regression relationships that can occur. In the fifth section we have give the details of problem under study, the numerical experiment and its results followed by conclusion.

2. Preliminaries

In this section, we give some primary definitions and notes, which are required in this work.

2.1. Definitions

2.1.1. Fuzzy number

Let R_F be the class of fuzzy subsets on the real axis (i.e. $u : R \to [0,1]$) satisfying the following properties:

1. $u \in R_F$, u is normal, i.e. $\exists x_0 \in R$ with $u(x_0) = 1$;
2. $u \in R_F$, u is convex fuzzy set, That is, $u(tx + (1-t)y) \geq min\{u(x), u(y)\}, \forall t \in [0,1], x, y \in R$;
3. $u \in R_F$, u is upper semicontinuous on R;
4. $\{\overline{x \in R; u(x) > 0}\}$ is compact, where $\overline{A}$ denotes the closure of A.

Then R_F is called the space of fuzzy numbers $\tilde{u}$, (refer [5]). Obviously $R \subset R_F$. Here, $R \subset R_F$ is obvious as R can be regarded as $\{\chi_x : x$ is any usual real number$\}$.

2.1.2. Non negative fuzzy number

A fuzzy number $\tilde{u}$ is said to be non-negative fuzzy number if $\tilde{u}(x) = 0, \forall x < 0$ and $\tilde{u}$ is said to be positive if $\tilde{u}(x) = 0, \forall x \leq 0$.

2.2. Operations on fuzzy numbers

2.2.1. α-cut / level-cut of fuzzy number

For $0 < \alpha \leq 1$, denote the α-cut as,
$[u]^\alpha = \{x \in R : u(x) \geq \alpha\}$ and $[u]^0 = \{\overline{x \in R; u(x) > 0}\}$.

Then it is well-known that for each $\alpha \in [0,1]$, the α-cut, $[u]^\alpha$ is a bounded closed interval $[{}^\alpha\underline{u}, {}^\alpha\overline{u}]$ in R.

The addition of the fuzzy numbers and the scalar multiplication is defined using α-cut, $\forall \alpha \in [0;1]$ as given below, refer [5].

2.2.2. Addition of two fuzzy numbers

For $u, v \in R_F$, the sum $u + v$ is obtained using α-cut, α-cut of sum of two fuzzy numbers is sum of their α-cuts.

That is, $[u+v]^\alpha = [u]^\alpha + [v]^\alpha = [{}^\alpha\underline{u}, {}^\alpha\overline{u}] + [{}^\alpha\underline{v}, {}^\alpha\overline{v}] = [{}^\alpha\underline{u} + {}^\alpha\underline{v}, {}^\alpha\overline{u} + {}^\alpha\overline{v}]$.

2.2.3. scalar multiplied with fuzzy number

For $\lambda \in R^+$, the product $\lambda \cdot u$ is given by $[\lambda \cdot u]^\alpha = \lambda[u]^\alpha = \lambda[{}^\alpha\underline{u}, {}^\alpha\overline{u}] = [\lambda\, {}^\alpha\underline{u}, \lambda\, {}^\alpha\overline{u}]$.

2.3. Triangular fuzzy number

A triangular fuzzy number is represented by $\tilde{u} = (u_1, u_2, u_3)$ and has membership function given as

$$\tilde{u} = \begin{cases} \frac{x-u_1}{u_2-u_1} & u_1 \le x \le u_2 \\ \frac{u_3-x}{u_3-u_2} & u_2 \le x \le u_3 \\ 0 & o.w. \end{cases} \tag{3}$$

Such a fuzzy number in its parametric form is given as $\tilde{u}(r) = (\underline{u}(r), \overline{u}(r))$, where $\underline{u}(r) = (u_2 - u_1)r + u_1$ and $\overline{u}(r) = u_3 - (u_2 - u_1)r$.

3. Artificial neural networks

Artificial Neural networks are massively parallel, distributed processing systems representing a computational technology built on the analogy to the human information processing system. Artificial Neural Network (ANN) is a massively connected network of simple processing elements called neurons. They have a natural propensity to learn and save the knowledge to make it available for use. ANNs can be used for classification, pattern recognition and function approximation and forecasting. Before the advent of ANN and other Artificial Intelligence models, these tasks were carried out by statistical methods such as the linear and nonlinear regression, principal components analysis and the Baysian classification models. The domain of application of such networks is vast and includes fields such as the finance, sales, economy, forensic science etc.

ANNs are being increasingly used for nonlinear regression and classification problems in meteorology due to their ability for learning from past data and do prediction. Recently statistical models, regression and Artificial Neural Networks, have been employed for predicting the Indian summer monsoon rainfall (ISMR) using lagged relationships between the ISMRI and various combinations of Niño indices, Shukla et al, [14].

The prediction of Indian summer monsoon rainfall (ISMR) on a seasonal time scales has been attempted by various research groups using different techniques. Such an attempt for prediction of ISMR is important for planning and devising agricultural strategies. The artificial neural network (ANN) technique with error- back-propagation algorithm has been successfully used to provide prediction of ISMR on monthly and seasonal time scales (Sahai et al, [13]).

'Artificial Neural Networks' plays a primary role in contemporary artificial intelligence and machine learning. The use of ANN in function approximation resulted from the following facts:

1. Cybenko [3] and Hornik [8] proved that the multi-layered Neural Network is universal approximator. It can approximate any continuous function defined on compact set.
2. The Back-propagation algorithm used for the training of the feed-forward Neural Networks with the hidden layers.

In the real practical situations when there is the lack of the mathematical model for the system, the ANN can be trained with the help of Input-Output pairs without fitting any kind of model to the system.

The fundamental element of neural network is neuron. We will refer to neuron as an operator, which maps and is explicitly, described by the equation

$$x_j = \Gamma\left(\sum_{i=1}^{n} w_{ji} u_i + w_0\right) \tag{4}$$

where, $U^T = [u_1, u_2, ..., u_n]$ is input vector, $W^T = [w_{j1}, w_{j2}, ..., w_{jn}]$ is weight vector to the j^{th} neuron and w_0 is the bias. $\Gamma(.)$ is a monotone continuous function, $\Gamma : R \to (-1,1)$ (eg. tanh(.) or signum(.)). Such neurons are interconnected to form a network.

In the feed-forward network neurons are organized in layers $h = 0,1,,L$. It is common practice to refer to the layer $l = 0$ as the input layer, to the $h = L$ as the output layer and to all other as hidden layers. A neuron at layer h receives its inputs only from neurons in the $h-1$ layer. The output of the i^{th} element in the layer h is given by

$$y_i^h = \Gamma\left(\sum_j w_{i,j}^h y_j^{h-1} + w_{i,0}^h\right) \tag{5}$$

where, $[w_i^h]^T = [w_{i,0}^h, w_{i,1}^h, ..., w_{i,n_{i-1}}^h]$ is the weight vector associated with i^{th} neuron at the h^{th} layer. Such a family of networks with n_i neurons at the i^{th} layer will be denoted by $N_{n_0,n_1,...,n_L}^L$.

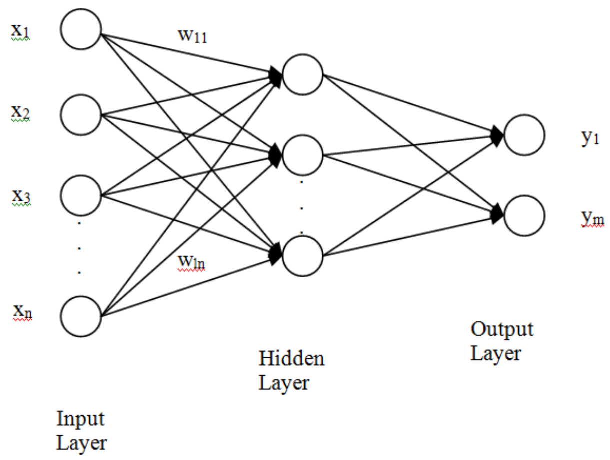

Fig. 1. Multilayer feedforward Neural Network

The general learning rule for weight updating is:

$$W(k+1) = W(k) + \Delta W(k)$$

where, $W(k+1)$ = weight in $(k+1)^{th}$ step, $W(k)$ = weight in $(k)^{th}$ step and $\Delta W(k)$ = change in weight in $(k)^{th}$ step. The change in weight for supervised delta rule is given by $\Delta w_{lj} = \eta f'(\sum_{i=0}^{n} w_{ji} x_i) f(\sum_{i=0}^{n} w_{ji} x_i)$.

The training in the feed-forward type multilayered neural network is done using back-propagation algorithm. Back-propagation algorithm is an extension of delta rule for training for multi-layer feed-forward networks. The algorithm is as described below.

3.1. Algorithm:

The Error Back Propagation learning algorithm for the above two-layer architecture can be derived as described below.

Weights are adjustable parameters of the network. We denote set of all weights of network by $\overrightarrow{W}$. Then the sum of squares of errors calculated by the network for current set of weights is:

$$E(\overrightarrow{W}) = \frac{1}{2}\sum_{l=1}^{L}(d_l - y_l)^2 \tag{6}$$

where, d_l is elements of desired output vector and y_l is elements of actual output vector. The weight updating takes place in two phases:

1. Output Layer Weight Updation
2. Hidden Layer Weight Updation

3.1.1. Output layer weight updation:

For the output layer neurons of the networks, the desired outputs 'd_i' are explicitly available. So, we can apply delta rule directly for updating the weights 'w_{lj}' of the output layer. The rule becomes

$$\Delta w_{lj} = w_{lj}^{new} - w_{lj}^{c}$$

$$\Delta w_{lj} = \eta_0 \frac{\partial E}{\partial w_{lj}}$$

$$\Delta w_{lj} = \eta_0 f_0^{'}(net_l) z_j \tag{7}$$

where, $l = 1,2,.....,L$; $j = 1,2,.....,J$ $net_l = \sum j = 0^J w_{lj} z_j$ = weighted sum for l^{th} output neuron
f_0 = Activation function for output layer neuron
$f_0^{'}$ = Derivative of the activation function f_0 with respect to net_l
w_{lj}^{new} = Updated weight of the output layer
w_{lj}^{c} = Current weight of the output layer

The z_j values are computed by propagating the input vector $\overrightarrow{X}$ through the hidden layer according to

$$z_j = f_h\left(\sum_{i=0}^{n} w_{ji} x_i\right)$$

$$z_j = f_h\left(net_j\right) \tag{8}$$

3.1.2. Hidden layer weight updation:

The derivation of an updation rule for the hidden layer weights cannot be done as in the output layer case since the desired outputs of the hidden layer are not specified by the patters. But a learning rule for hidden neuron can be obtained by attempting to minimize the output layer errors. This process propagates the output errors through the output layer back towards the hidden layer. The resulting weight updation rule is called "Error Back Propagation" learning rule. It is generalization of the 'delta rule'. The derivation follows the gradient descent method performed on the criterion function E defined by above equation. This time gradient calculated with respect to hidden weights:

$$\Delta w_{ji} = -\eta_h \frac{\partial E}{\partial w_{ji}} \tag{9}$$

where, $j = 1,2,.....,J$; $i = 1,2,.....,n$
η_h is the learning rate for the hidden neurons
Partial derivative is evaluated at current values of weights.

Now using the chain rule for differentiation, the partial derivative in (9) can be derived as:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial z_j}\frac{\partial z_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} \tag{10}$$

But, $net_j = \sum_{i=0}^{n} w_{ji} x_i$

$$\frac{\partial net_j}{\partial w_{ji}} = x_i \tag{11}$$

And $z_j = f_h(net_j)$

$$\frac{\partial z_j}{\partial net_j} = f_h^{'}(net_j) \tag{12}$$

Also,

$$\frac{\partial E}{\partial z_j} = \frac{\partial}{\partial z_j}\left[\frac{1}{2}\sum_{l=1}^{L}(d_l - f_0(net_l))^2\right]$$

$$\frac{\partial E}{\partial z_j} = -\sum_{l=1}^{L}[d_l - f_0(net_l)]\frac{\partial f_0(net_l)}{\partial z_j}$$

$$\frac{\partial E}{\partial z_j} = -\sum_{l=1}^{L}[d_l - y_l] f_0^{'}(net_l) w_{lj} \tag{13}$$

Putting (11), (12), (13) in (10) we get,

$$\frac{\partial E}{\partial w_{ji}} = \left\{ -\sum_{l=1}^{L} [d_l - y_l] f_0^{'}(net_l) w_{lj} \right\} \{f_h^{'}(net_j)\}\{x_i\} \tag{14}$$

Hence, by equation (9), the desired learning rule for the weights updation for hidden layers is obtained as:

$$\Delta w_{ji} = \eta_h \left\{ \sum_{l=1}^{L} [d_l - y_l] f_0^{'}(net_l) w_{lj} \right\} \{f_h^{'}(net_j)\}\{x_i\} \tag{15}$$

By, comparing Eqs. (7) and (15), it can be observed that an estimated target value d_j for the output of the j^{th} hidden neuron can be specified in terms of the back-propagated error signal as:

$$d_j - z_j = \sum_{l=1}^{L} [d_l - y_l] f_0^{'}(net_l) w_{lj} \tag{16}$$

Neural Network approximation for regression relation between the variables (x, y), can be given as:

$$y = NN_f(x) \tag{17}$$

The relation such as given by equation (17) can be efficiently approximated using multi-layer feed-forward neural network. This network is trained by Back-propagation algorithm and can be used for approximating the relationship between the variables. Once such a network is trained it can be easily used for forecasting. Such an ANN approximation of f will be minimum error approximation due to the in-built property of the network refer [11].

This implies that given any $\epsilon > 0$ a neural network with sufficiently large number of nodes can be determined such that

$$\|f(x) - NN(x)\| < \epsilon, for all x \in D,$$

where f is the function to be approximated and D is the compact domain of finite dimensional norm vector space.

4. Fuzzy regression

A general form of a linear fuzzy regression is represented by

$$\tilde{y} = \tilde{a}_0 + \tilde{a}_1 x_1 + \tilde{a}_2 x_2 + \dots + \tilde{a}_n x_n \tag{18}$$

where, $x_i, 1 \le i \le n$ are crisp independent variables (inputs) $\tilde{a}_i, 1 \le i \le n$ are fuzzy numbers as the regression coefficients (unknown parameters), $\tilde{y}$ is mainly a fuzzy dependent variable (output) represented by triangular fuzzy number. This fit will be useful if the data is linearly dependent, however, if the data is nonlinearly related then it is very difficult to give form to the relationship. Consider the general representation of the fuzzy nonlinear relationship

$$\tilde{y} = f(\tilde{x}) \tag{19}$$

Since the fuzzy numbers are represented by triangular membership, the relation (19) becomes

$$(y_1, y_2, y_3) = f((x_1, x_2, x_3)) \tag{20}$$

Taking α-cut on both the sides, we get,

$$\left[(y_2 - y_1)\alpha + y_1, y_3 - (y_3 - y_2)\alpha\right] = f([(x_2 - x_1)\alpha + x_1, x_3 - (x_3 - x_2)\alpha]) \tag{21}$$

Comparing the components we get $(y_2 - y_1)\alpha + y_1 = f((x_2 - x_1)\alpha + x_1)$ and $y_3 - (y_3 - y_2)\alpha = f(x_3 - (x_3 - x_2)\alpha)$ for all $\alpha \in (0, 1]$.

Also, the nonlinear relationship represented by (19) satisfies following conditions:

1. ${}^{\alpha}\tilde{y} = f({}^{\alpha}\tilde{x})$, for all $\alpha \in (0, 1]$.
2. For $\alpha < \beta$ in $(0, 1]$, ${}^{\beta}\tilde{y} = f({}^{\beta}\tilde{x}) \subset {}^{\alpha}\tilde{y} = f({}^{\alpha}\tilde{x})$.

Table 1. Data for crisp evaporation (E) and temperature $(\tilde{T})$

Evaporation E	Temp-min T1	Temp-avg T2	Temp-max T3
8	20.4	26.2	32
7	18.2	26	33.8
5	19	27.75	36.5
4.5	17.4	25.45	33.5
6.6	17.7	24.6	31.5
4.9	18	22.95	27.9
5	16.4	24	31.6
3.9	16.3	25	33.7
6.2	21.4	28.5	35.6
6.5	20.8	28.5	36.2
7	18.4	27.2	36
7	18.2	27.85	37.5
8.2	19.8	29.4	39
7	19.4	29.45	39.5
8.2	19	28.9	38.8
8.8	18.9	29.4	39.9
11.2	19	30.1	41.2

We propose the use of ANN for the approximation of unknown f involved in (19) from the given fuzzy data points. In equation (21), putting $\alpha = 1$ gives us,

$$y_2 = f(x_2) \tag{22}$$

and putting $\alpha = 0$ gives,

$$[y_1, y_3] = f([x_1, x_3]) \tag{23}$$

That is, $y_1 = f(x_1)$ and $y_3 = f(x_3)$.

Thus, the fuzzy relation f relates each data point $\tilde{x}$ to $\tilde{y}$.

We have done experiments for two cases: One, when only the independent variable is fuzzy and secondly when both the variables $\tilde{y}$ and $\tilde{x}$ are fuzzy represented by triangular membership functions given as, $\tilde{x}_i = (x_{i1}, x_{i2}, x_{i3})$ and $\tilde{y}_i = (y_{i1}, y_{i2}, y_{i3})$. We used multilayered ANN with appropriate architecture and trained it using Back-Propagation algorithm.

5. Numerical experiments

To test the proposed algorithm of fitting the data using ANN we have taken the monthly data for the evaporation and temperature from (http://www.iari.res.in, for the date: 14/4/2014 to 30/4/2014) for the IARI New Delhi division.

5.1. CASE - I:

Here the dependent variable evaporation (E) is crisp and temperature (T) is fuzzy represented triangular fuzzy number having values (minimum temperature, average temperature and maximum temperature) as shown in Table 1.

The network with configuration $N^4_{3,20,15,10,1}$ as shown in the Figure - 2 is trained with the above data.The training performance is as shown in the Fig. 4. Training state after the convergence of training error for the data is as shown in the Fig. 3. The testing of the training data is shown in the Fig. 5.

5.2. CASE - II:

In this case both the dependent variable evaporation (E) as well as the temperature (T) are fuzzy represented triangular fuzzy numbers as shown in Table 2.
The network with configuration $N^4_{3,20,15,10,1}$ trained with the above data. The testing results are shown in Fig. 6. These results of the test performed supports use of ANN for fuzzy regression.

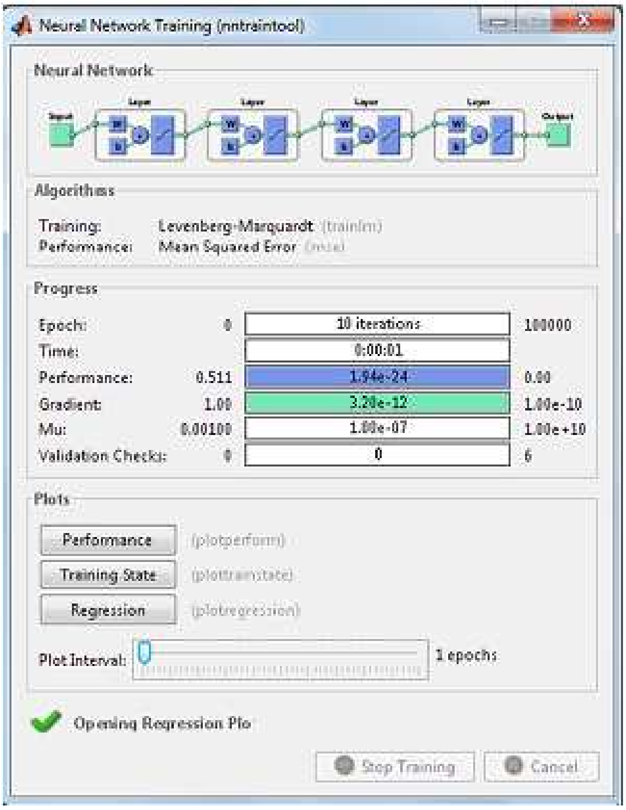

Fig. 2. Multilayer feedforward Neural Network for training evaporation and fuzzy temperature

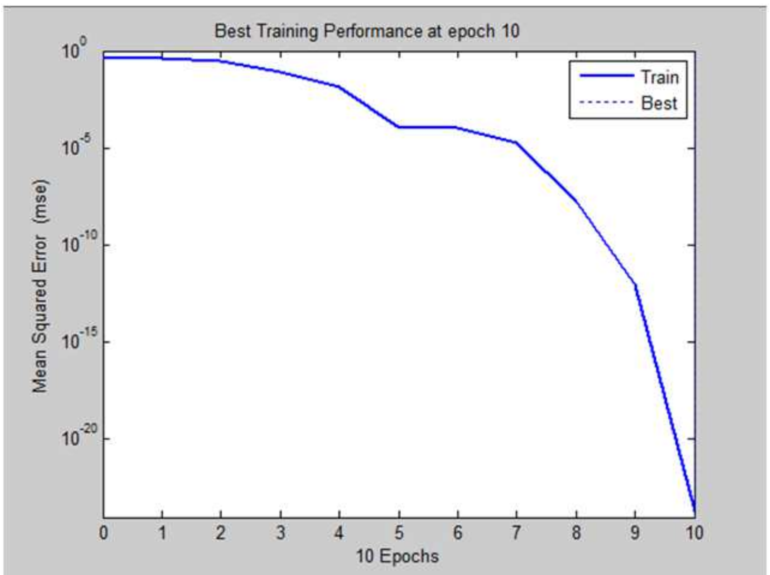

Fig. 3. Training convergence for ANN evaporation and fuzzy temperature

Table 2. Data for fuzzy evaporation ($\tilde{E}$) and temperature ($\tilde{T}$)

Evapo-min E1	Evaporation E2	Evapo-max E3	Temp-min T1	Temp-avg T2	Temp-max T3
5.472302	8	8.896571	20.4	26.2	32
6.43757	7	10.08082	18.2	26	33.8
1.021577	5	7.477222	19	27.75	36.5
4.116075	4.5	7.027687	17.4	25.45	33.5
2.032831	6.6	9.849161	17.7	24.6	31.5
2.411173	4.9	6.040306	18	22.95	27.9
1.240447	5	9.531918	16.4	24	31.6
2.837069	3.9	5.312859	16.3	25	33.7
3.728984	6.2	9.615074	21.4	28.5	35.6
3.844059	6.5	10.9005	20.8	28.5	36.2
3.500269	7	8.053951	18.4	27.2	36
2.12785	7	7.656706	18.2	27.85	37.5
6.818143	8.2	10.60297	19.8	29.4	39
3.205031	7	11.75971	19.4	29.45	39.5
3.639248	8.2	11.53486	19	28.9	38.8
6.77436	8.8	13.04875	18.9	29.4	39.9
7.739701	11.2	15.98851	19	30.1	41.2

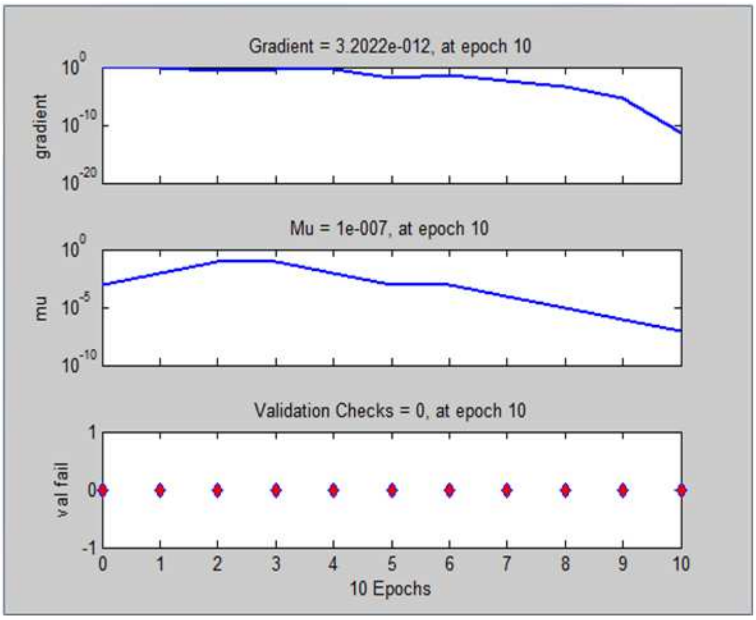

Fig. 4. Performance for ANN evaporation and fuzzy temperature

6. Conclusion

Fuzzy nonlinear regression based on the least squares of errors is a difficult because we are not able to give form to such nonlinear relationship. In this work, we propose to use ANN for approximation of relationship between the fuzzy data. The understanding and quantification of possibilistic uncertainties in temperature ? evaporation relationship is important for yield of crop. The paper presented a methodology for analysis of uncertainties in the temperature ? evaporation relationship using fuzzy regression and realizes it with Artificial Neural Networks.

The study shows that proposed fuzzy regression analysis provides an appropriate methodology for analysis of uncertainties in the temperature - evaporation relationship. The method allows quantification and aggregation of individual sources of uncertainty in temperature measurement and definition of uncertainty in the temperature evaporation relationship efficiently modeled in fuzzy setup.

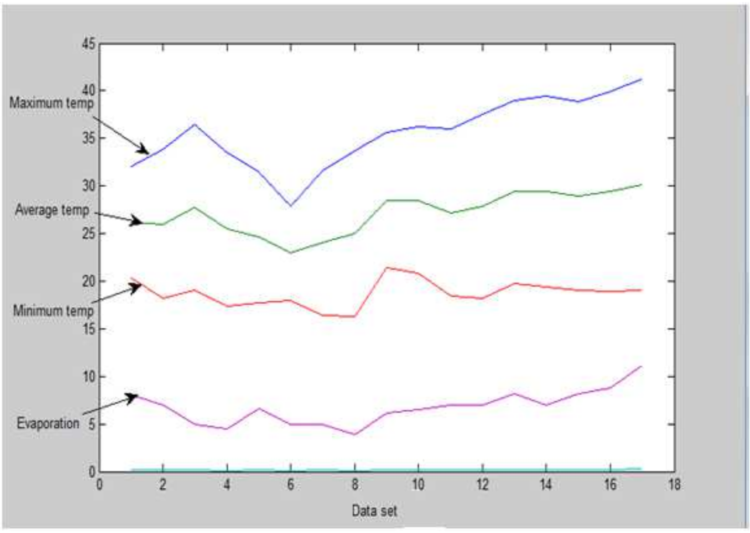

Fig. 5. Testing for ANN evaporation and fuzzy temperature

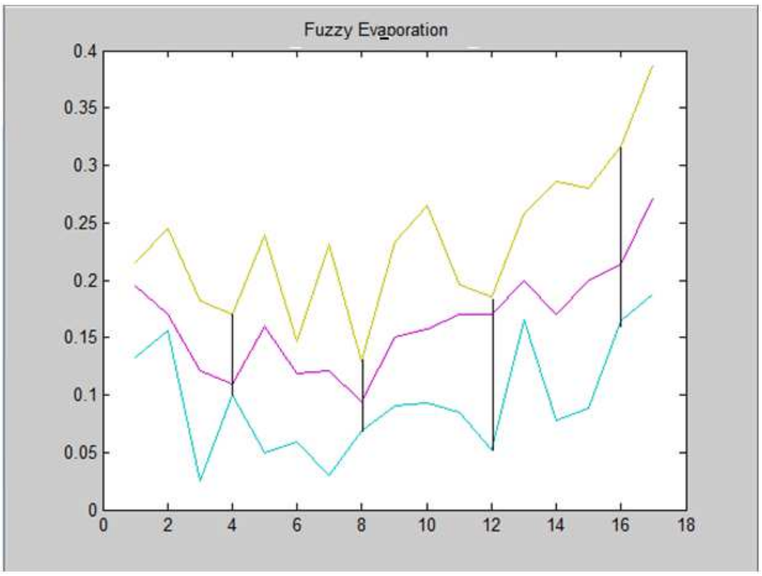

Fig. 6. Testing for ANN fuzzy evaporation and fuzzy temperature

References

[1] Y.H. Chang and B.M. Ayyub, Fuzzy regression methods- a comparative assessment, Fuzzy sets and systems 119 (2001) 187-203.

[2] L. H. Chen and C. C. Hsueh, A mathematical programming method for formuation a fuzzy regression model based on distance criterion, IEEE Trans. Syst. Man Cybernet. B-37, (2007) 705-712.

[3] G. Cybenko, Approximation by superposition id sigmoidal functions, Mathematics of control signals and systems 2(4) (1989) 303-314.

[4] P. Diamond, Fuzzy least squares, Information Sciences 46 (1988) 141-157.

[5] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, 1980.

[6] M. Friedman, M. Ming and A. Kandel, General Fuzzy least Squares, Fuzzy Sets and Systems 88 (1997) 107-118.
[7] D.H. Hong, C. Hwang, C. Ahn: Ridge estimation for regression models with crisp inputs and Gaussian fuzzy output. Fuzzy Sets and Systems 142(2) (2004) 307-319.
[8] K. Hornik and H. White H., Multi-layer feedforward networks are universal approximators, Neural Networks 2 (1989) 359-366.
[9] G. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.
[10] J. Mohamadian and S.M. Taheri, Pedomodels filling with fuzzy least squares regression. Iranian Journal of Fuzzy Systems 1(2) (2004), 45-61.
[11] A. U. Levin and Kumpati S. Narendra., Control of nonlinear dynamical systems using neural networks: Controllability and stabilization, IEEE Transactions on neural networks 4(2) (1993) 192-206.
[12] G. Peters, Fuzzy linear regression with fuzzy intervals, Fuzzy Sets and Systems 63(1) (1994) 45-55.
[13] A. K. Sahai, M..K. Soman, and V. Satyan, All India Summer Monsoon Rainfall Prediction Using an Artificial Neural Network, Climate Dynamics 16 (2000) 291-302.
[14] S. P. Shukla, M.A. chandler, D. Rind, L.E. Sohi, J .Jonas and J. Lerner, Teleconnections in a warmer climate:The Pliocene perspective. Climate Dynamics 37 (2011) 1869-1887.
[15] A. Shapiro, "Fuzzy Regression and the Term Structure of Interest Rates Revisited", 14th International AFIR Colloquium 1 (2004) 29-45.
[16] H. Tanaka, S. Uegima, K. Asai, Linear regression analysis with fuzzy model, IEEE Trans. Systems, Man and Cybernetics 12 (1982) 903-907.
[17] N. Wang, W.X. Zhang, C.L. Mei, Fuzzy nonparametric regression based on local linear smoothing technique, Information Sciences 177 (2007) 3882-3900.
[18] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338-353.
[19] Z. Zareamoghaddam and H. Zareamoghaddam, Solving Fuzzy Linear Regression by Huan's Algorithm, World Applied Sciences Journal 12(12) (2012) 2358-2364.