Applied Mathematical Modelling 35 (2011) 5400–5412

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Fuzzy polynomial regression with fuzzy neural networks

M. Mosleh [a], M. Otadi [a], S. Abbasbandy [b,*]

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

ARTICLE INFO

Article history:
Received 25 September 2010
Received in revised form 17 April 2011
Accepted 26 April 2011
Available online 6 May 2011

Keywords:
Neural network
Fuzzy polynomial regression model
Learning algorithm

ABSTRACT

Recently, fuzzy linear regression is considered by Mosleh et al. [1]. In this paper, a novel hybrid method based on fuzzy neural network for approximate fuzzy coefficients (parameters) of fuzzy polynomial regression models with fuzzy output and crisp inputs, 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. Finally, we illustrate our approach by some numerical examples.

1. Introduction

The concept of fuzzy numbers and fuzzy arithmetic operations were first introduced by Zadeh [2], Dubois and Prade [3]. We refer the reader to [4] for more information on fuzzy numbers and fuzzy arithmetic.

Regression analysis is of the most popular methods of estimation. It is applied to evaluate the functional relationship between the dependent and independent variables. Fuzzy regression analysis is an extension of the classical regression analysis in which some elements of the model are represented by fuzzy numbers. Fuzzy regression methods have been successfully applied to various problems such as forecasting [5–7] and engineering [8]. Thus, it is very important to develop numerical procedures that can appropriately treat fuzzy regression models.

In the literature, several papers have addressed the issue of regression under fuzzy environment. Tanaka et al. [9] first formulated a linear programming problem to determine the regression coefficients as fuzzy numbers. Later, Tanaka [10], Tanaka and Watada [11] and Tanaka et al. [12] made some improvements. Kao and Chyu [13] the concept of least squares which was widely applied in the classical regression analysis was adopted to determine the regression coefficients. Very recently, Mosleh et al. [1] proposed a learning algorithm of fuzzy neural network with crisp inputs, fuzzy weights and fuzzy output for adjusting fuzzy weights of fuzzy linear regression model of the form

$$\overline{Y}_i = A_0 + A_1x_{i1} + \cdots + A_nx_{in},$$

where i indexes the different observations, $x_{i1}, x_{i2}, \ldots, x_{in} \in \mathbb{R}$, all coefficients and $\overline{Y}_i$ are fuzzy numbers. In this paper, firstly, the method of neural network under fuzzy environment is introduced. Then the approach of obtaining the estimates of the regression coefficients is described next. Three examples to illustrate how to apply the proposed method to determine the regression polynomials for fuzzy observations are described. The advantages of the proposed method over some other methods are discussed in follows.

* Corresponding author. Tel.: +98 912 1305326; fax: +98 2813780040.
E-mail address: abbasbandy@yahoo.com (S. Abbasbandy).

0307-904X/$ - see front matter
doi:10.1016/j.apm.2011.04.039

Ishibuchi et al. [14] proposed a learning algorithm of fuzzy neural networks with triangular fuzzy weights and Hayashi et al. [15] fuzzified the delta rule. Buckley and Eslami [16] consider neural net solutions to fuzzy problems. The topic of numerical solution of fuzzy polynomials by fuzzy neural network investigated by Abbasbandy and Otadi [17], consists of finding solution to polynomials like $a_1x + a_2x^2 + \cdots + a_nx^n = a_0$ where $x \in \mathbb{R}$ and $a_0, a_1, \ldots, a_n$ are fuzzy numbers, and finding solution to systems of s fuzzy polynomial equations such as [18]:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= a_{10}, \\ &\vdots \\ f_l(x_1, x_2, \ldots, x_n) &= a_{l0}, \\ &\vdots \\ f_s(x_1, x_2, \ldots, x_n) &= a_{s0}, \end{aligned}$$

where $x_1, x_2, \ldots, x_n \in \mathbb{R}$ and all coefficients are fuzzy numbers.

In this paper, we first propose an architecture of fuzzy neural network (FNN) with fuzzy weights for real input vectors and fuzzy targets to find approximate coefficients to fuzzy polynomial regression model

$$\overline{Y}_i = A_{l0} + \sum_{j=1}^{n} A_{lj} x_{ij} + \sum_{j=1}^{n} \sum_{k=1}^{n} A_{ljk} x_{ij} x_{ik} + \cdots,$$

where i indexes the different observations, $x_{i1}, x_{i2}, \ldots, x_{in} \in \mathbb{R}$, all coefficients and $\overline{Y}_i$ are fuzzy numbers. The input–output relation of each unit is defined by the extension principle of Zadeh [2]. Output from the fuzzy neural network, which is also a fuzzy number, is numerically calculated by interval arithmetic [19] for fuzzy weights and real inputs. Next, we define a cost function for the level sets of fuzzy outputs and fuzzy targets. Then, a crisp learning algorithm is derived from the cost function to find the fuzzy coefficients of the fuzzy polynomials regression models.

2. Preliminaries

In this section the basic notations used in fuzzy calculus are introduced. We start by defining the fuzzy number.

Definition 1. A fuzzy number is a fuzzy set $u : \mathbb{R}^1 \to I = [0, 1]$ such that

i. u is upper semi-continuous;
ii. $u(x) = 0$ outside some interval $[a, d]$;
iii. There are real numbers b and c, $a \leqslant b \leqslant c \leqslant d$, for which
   1. $u(x)$ is monotonically increasing on $[a, b]$,
   2. $u(x)$ is monotonically decreasing on $[c, d]$,
   3. $u(x) = 1, b \leqslant x \leqslant c$.

The set of all the fuzzy numbers (as given in Definition 1) is denoted by E^1.

An alternative definition which yields the same E^1 is given by Kaleva [20], Ma et al. [21].

Definition 2. A fuzzy number u is a pair $(\underline{u}, \bar{u})$ of functions $\underline{u}(r)$ and $\bar{u}(r)$, $0 \leqslant r \leqslant 1$, 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. $\bar{u}(r)$ is a bounded monotonically decreasing, left continuous function on $(0,1]$ and right continuous at 0;
iii. $\underline{u}(r) \leqslant \bar{u}(r)$, $0 \leqslant r \leqslant 1$.

A crisp number r is simply represented by $\underline{u}(\alpha) = \bar{u}(\alpha) = r$, $0 \leqslant \alpha \leqslant 1$. 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. Its parametric form is

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

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 $\widehat{FZ}$.

2.1. Operations on fuzzy numbers

We briefly mention fuzzy number operations defined by the extension principle [2]. Since input vector of feedforward neural network is fuzzified in this paper, the following addition, multiplication and nonlinear mapping of fuzzy numbers are necessary to define our fuzzy neural network: