Flow Measurement and Instrumentation 49 (2016) 46–51

Contents lists available at ScienceDirect

Flow Measurement and Instrumentation

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

Short communication

Open channel junction velocity prediction by using a hybrid self-neuron adjustable artificial neural network

Hossein Bonakdari, Amir Hossein Zaji *

Department of Civil Engineering, Razi University, Kermanshah, Iran

ARTICLE INFO

Article history:
Received 21 January 2015
Received in revised form
11 April 2015
Accepted 3 April 2016
Available online 4 April 2016

Keywords:
Artificial neural network
Genetic algorithm
Genetic programming
Neuron number determination
Open channel junction
Velocity prediction

ABSTRACT

Determining the appropriate hidden layers neuron number is one of the most important processes in modeling the Multi-Layer Perceptron Artificial Neural Network (MLP-ANN). Despite the significant effect of the MLP-ANN neurons number on predicting accuracy, there is no definite rule for its determination. In this study, a new self-neuron number adjustable, hybrid Genetic Algorithm-Artificial Neural Network (GA-ANN), is introduced and its application examined on the complex velocity field prediction of an open channel junction. The results of GA-ANN were compared with those got by the Genetic Programming (GP) method as two applications of the Genetic Algorithm (GA). The comparisons showed that the GA-ANN model can predict the open channel junction velocity with higher accuracy than the GP model, with Root Mean Squared Error (*RMSE*) of 0.086 and 0.156, respectively. Finally the equation, obtained by applying the GA-ANN model, predicting the velocity at the open channel junction is presented.

1. Introduction

Open channel junctions are widely used in hydraulic structures such as sewer networks, water transfer, and irrigation and drainage systems. Estimating the accurate velocity field in an open channel junction has a significant impact on the designing process for preventing the effects from erosion and deposition on the flow. Because of the complex 3D flow behavior in the open channel junctions, various experimental [1–4] and numerical modeling [5–9] studies have been conducted. The MLP-ANN is used for simulating various problems as a powerful computational intelligence method. The method consists of some cells, called neurons, to establish an interconnection between the input, hidden, and output layers. MLP-ANN is widely used in problems with a complex relationship between the input and output layers. Many studies have been carried out with the goal of improving the MLP-ANN prediction performance. The studies used other artificial intelligence methods such as other neural networks and optimization algorithms for hybridization with the MLP-ANN. Choi and Park [10] focused on a hybrid ANN that could reduce the dimensions of the input variables. Sarkar and Modak [11] used the Simulated Annealing (SA) method to construct a hybrid ANN-SA method and apply it on nonlinear and time series problems. Khashei, et al. [12] used a hybrid ANN-fuzzy regression model for time series forecasting problems. Lin and Wu [13] combined the Self-Organizing Map (SOM) and the MLP-ANN to propose a hybrid model for rainfall modeling. Ghalambaz, et al. [14] used Gravitational Self Algorithm (GSA) to train the ANN and used the results for solving Wessinger's equation. Mitra, et al. [15] used the GA algorithm for optimizing the ANN weights and biases.

The aim of this study is to apply a novel code of a hybrid GA-ANN on a simulation of the complex flow velocity field of the open channel junction. Selecting the appropriate hidden layers neurons number of the MLP-ANN method has a great impact on the model performance. The proposed GA-ANN is a hybrid method that has a self-adjustability of the hidden layers neurons number. The performance of the proposed model was then compared with the GP model. To train and validate the numerical models, the experimental study of Weber, et al. [3] was employed.

2. Material and methods

2.1. Self- adjustable hidden layers neuron artificial neural network (GA-ANN)

A typical MLP-ANN consists of one input, one or more hidden and one output layers. The neuron number of the input layer is equal to the input variables. The output layer neurons are equal to the model outputs. Hidden layers are used to establish a nonlinear

* Corresponding author.
E-mail address: amirzaji@gmail.com (A.H. Zaji).

http://dx.doi.org/10.1016/j.flowmeasinst.2016.04.003
0955-5986/

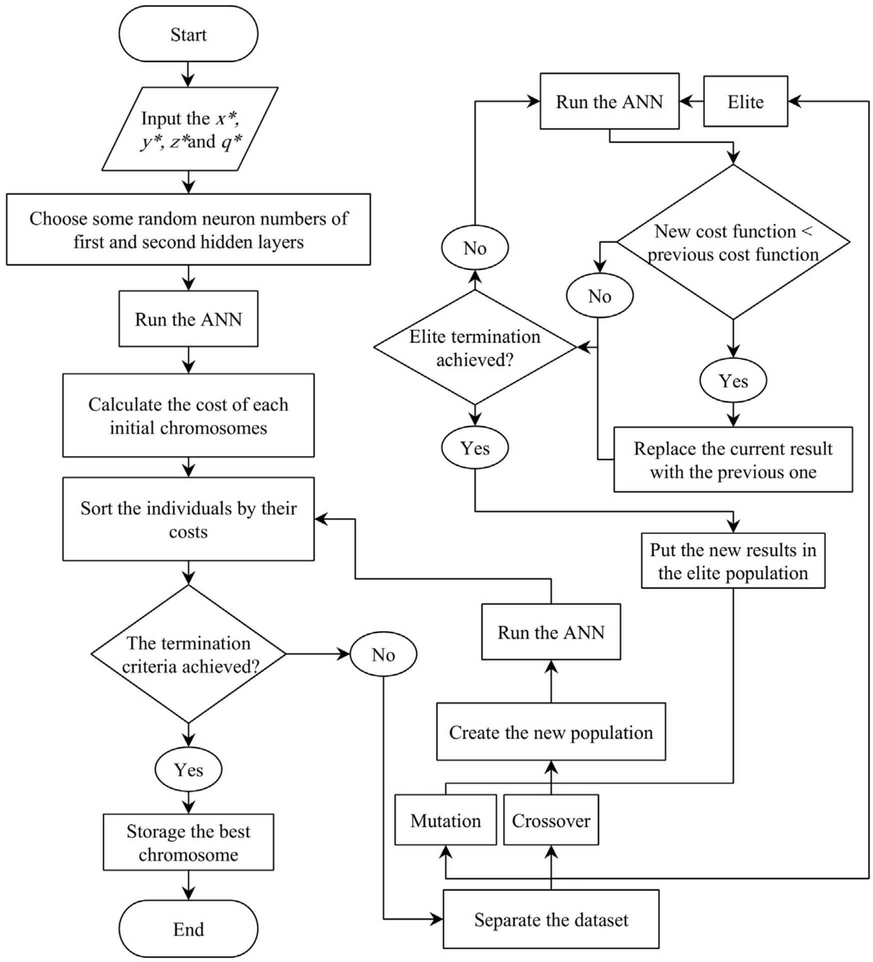

Fig. 1. GA-ANN flowchart.

connection between the input and output layers. In this study, two hidden layers were considered. One of the most puzzling processes of MLP-ANN modeling is determination the hidden layer neuron numbers. There is not a definite rule in neuron number determination. In the current study, a novel hybrid GA-ANN method is introduced that could self-adjust the hidden layer neurons. To do this, some modifications were done on the GA. The schematic overview of the proposed GA-ANN is shown in Fig. 1. The following procedure was carried out: (1) Giving the input variables to the model. (2) A population of the hidden neurons generated randomly. (3) Calculate the cost of each chromosome and sort them. (4) If the termination criteria are achieved, stop the process. If not, run GA to find the next generation of neuron numbers.

Because of the random nature of the Levenberg-Marquardt back-propagation algorithm used in this study to train the ANNs, it is probable that a good chromosome gets eliminated by the GA algorithm due to bad luck in the ANN training process. Therefore, according to Fig. 1, a modification was done in the GA. In the modified GA, the elite population is run repeatedly so that the minimum cost of the repetition process is achieved and saved as that chromosome's cost. Therefore, the bad luck chromosome elimination does not occur in the elite children (that have the best chromosomes).

2.2. Genetic programming method (GP)

In order to compare the accuracy of the proposed model with the other intelligence methods, the GP model of Koza [16] was made use of. The goal of selecting the GP method is to provide a comparison between two GA applications (GP and GA-ANN). The process of this algorithm is similar to GA optimization. However, GP uses the computer programs as chromosomes. Therefore, by optimizing the various computer programs, the best one with less cost was selected as the result of the GP. In the present GP model, the mathematical functions of $(+,\ -,\ \times,\ \div$, sin, cos, abs, e^x, power, sqrt and x^2) were used as the allowable functions for use in the computer programs. Another GP property that should be determined before modeling is the fitness function. The fitness function is used to calculate the cost of each computer program during the modeling process. In this study, the *MSE* fitness function was used. *MSE* statistical error is defined as follows:

$$MSE = \frac{\sum_{i=1}^{N} (GP_i - EXP_i)^2}{N} \tag{1}$$

where GP_i is the computer program output, EXP_i the measured non-dimensional longitudinal velocity from the experimental study and N is the number of the samples concerned. In the present GP model, the initial size of the random computer programs was determined as 64 bytes. To prevent the unexpected size increasing of the computer programs during the GP process, the maximum size of the computer programs was regarded as 256 bytes. The flowchart of the GP method is shown in Fig. 2.