Advances in Engineering Software 97 (2016) 85–95

Contents lists available at ScienceDirect

Advances in Engineering Software

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

A self-tuning system for dam behavior modeling based on evolving artificial neural networks

B. Stojanovic [a,*], M. Milivojevic [b], N. Milivojevic [c], D. Antonijevic [a]

[a] Faculty of Science, University of Kragujevac, Radoja Domanovica 12, 34000 Kragujevac, Serbia
[b] Technical and Business College, Uzice, Trg Sv. Save 34, 31000 Uzice, Serbia
[c] "Jaroslav Cerni" Institute for the Development of Water Resources, 11000 Belgrade, Serbia

ARTICLE INFO

Article history:
Received 3 August 2015
Revised 8 January 2016
Accepted 20 February 2016
Available online 16 March 2016

Keywords:
Hybrid dam models
Artificial neural networks
Evolving networks
Genetic algorithms
Multiple linear regression
Dam stability

ABSTRACT

Most of the existing methods for dam behavior modeling presuppose temporal immutability of the modeled structure and require a persistent set of input parameters. In real-world applications, permanent structural changes and failures of measuring equipment can lead to a situation in which a selected model becomes unusable. Hence, the development of a system capable to automatically generate the most adequate dam model for a given situation is a necessity. In this paper, we present a self-tuning system for dam behavior modeling based on artificial neural networks (ANN) optimized for given conditions using genetic algorithms (GA). Throughout an evolutionary process, the system performs near real-time adjustment of ANN architecture according to currently active sensors and a present measurement dataset. The model was validated using the Grancarevo dam case study (at the Trebisnjica river located in the Republic of Srpska), where radial displacements of a point inside the dam structure have been modeled as a function of headwater, temperature, and ageing. The performance of the system was compared to the performance of an equivalent hybrid model based on multiple linear regression (MLR) and GA. The results of the analysis have shown that the ANN/GA hybrid can give rather better accuracy compared to the MLR/GA hybrid. On the other hand, the ANN/GA has shown higher computational demands and noticeable sensitivity to the temperature phase offset present at different geographical locations.

1. Introduction

To describe and predict the structural behavior of dams, a number of statistical, deterministic and hybrid mathematical models have been developed over the past decades. Statistical models based on multiple linear regression (MLR) and their advanced forms such as stepwise regression, robust regression, ridge regression and partial least squares regression have been shown to be more or less successful in dam modeling [1–3]. In contrast to statistical modeling, deterministic models require the solving of differential equations, for which closed form solutions could be difficult or impossible to obtain [4]. Therefore, many models that are based on numerical methods, such as the finite element method (FE), have also been developed [5]. Recently, numerical and statistical methods have been enriched with various heuristics from the artificial intelligence (AI) domain, creating hybrid models that combine their advantages.

Some of these artificial intelligence techniques and heuristic algorithms are artificial neural networks (ANN) [6–10], genetic algorithms (GA) [11–13] and particle swarm optimization (PSO). In his paper [8], Mata presented a comparison between MLR and ANN models for the characterization of dam behavior under environmental loads for the Alto Rabagao arch dam. Gholizadeh et al. [10] used a hybrid methodology with a combination of metaheuristics (GA and PSO) and neural networks to propose an efficient soft computing approach to achieve optimal shape design of arch dams that were subjected to natural frequency constraints. Gomes et al. [14] employed PSO for structural truss mass optimization on size and shape, considering frequency constraints. The results showed that the PSO algorithm performed similarly to other methods and even better in some cases. Several recent studies have also described the application of artificial immune algorithm (AIA) techniques, which imitate the function of a natural immune system [15,16]. Xi et al. [15] proposed an immune statistical model to resolve the data analysis problems of dam horizontal crest upstream-downstream displacements.

In a number of papers [17–22] researchers have made a significant effort to find optimal structures of neural networks for various problems. Majdi et al. [17] combined a neural network and genetic

* Corresponding author. Tel.: +381 69 11 543 75.
E-mail addresses: bobi@kg.ac.rs, boban.stojanovic@gmail.com (B. Stojanovic), milovan.milivojevic@vpts.edu.rs (M. Milivojevic), nikola.milivojevic@gmail.com (N. Milivojevic), dantonijevic.kg@gmail.com (D. Antonijevic).

http://dx.doi.org/10.1016/j.advengsoft.2016.02.010
0965-9978/

algorithm for predicting the deformation modulus of rock masses. GA is utilized to find the optimal number of neurons in a hidden layer, and the learning rates and momentum coefficients of hidden and output layers of the network. Using a standard backpropagation gradient descent algorithm, they tested networks with linear and sigmoid activation functions. Zhou et al. [19] presented a combined procedure of the orthogonal design (OD), FE analysis, ANN and GA for inverse modeling of the seepage/leakage problems. The chosen neural network used the sigmoid transfer function and had a fixed number of layers: one input layer, two hidden layers and one output layer. The number of neurons at the hidden layers was determined by minimizing an error function on a test dataset using a trial-and-error method. To obtain a quick training time and high generalization accuracy, the Levenberg–Marquardt backpropagation algorithm (LM) combined with Bayesian regularization is used for training of the network. In the study [18], a hybrid finite element–boundary element analysis (FE–BE) in conjunction with an ANN procedure is proposed for the prediction of dynamic characteristics of an existing concrete gravity dam. The conjugate gradient algorithm (CGA) and the LM algorithm are implemented for fast training of the ANNs. The authors tested neural networks with one hidden layer where the number of neurons was determined by a trial-and-error method. Hooshyaripor et al. [21] showed that a three-layer ANN model is appropriate to deal with a dam breach problem which has two inputs: the height of water behind the dam and the volume of water behind the dam at the failure time, and one output: the peak outflow discharge. In their study a feed-forward neural network model with a single hidden layer is used. Applying Hecht–Nielsen criterion [22], it was found that an ANN with four neurons in the hidden layer has higher performance. The LM algorithm was employed to train the ANN model. As a transfer function, tan-sigmoid and linear functions were employed in the hidden and output layers, respectively.

In our previous work we developed an adaptive system for dam behavior modeling based on a linear regression model optimized for given conditions using genetic algorithms [23]. Throughout the evolutionary process, the system performs near real-time adjustment of regressors in the MLR model according to currently active sensors. Following this idea, we developed a system for dam behavior modeling based on artificial neural networks, capable of adapting to persistent changes in the measuring system and measurement database. In order to achieve a full adaptability of the ANN model in real-world conditions, the system should be able to optimize all significant network parameters according to available input sensors and historical data. To the best of our knowledge, the existing solutions optimize a limited set of ANN parameters only, while the other parameters are chosen arbitrarily, based on experience, literature or trial-error methods. In this paper, we present a novel methodology and a system for automatic generation of an ANN dam model, which optimizes all significant elements of the ANN architecture. Guided by a variant set of input variables, the system permanently optimizes network topology, activation functions and learning algorithms in order to fit the growing measurement database. Optimization of the parameters is performed using genetic algorithms. The quality of the proposed ANN/GA hybrid dam models has been tested on a real-world case study and compared to equivalent dam models based on multiple linear regression and GA (MLR/GA).

2. Theoretical background

2.1. Artificial neural network

The ANN is a simplified mathematical model of a natural neural network. It is a computing system made up of a number of simple, interconnected processing elements or neurons, which process information by a dynamic state response to external inputs [17]. Processing elements are grouped in layers: an input layer, one or more hidden layers and an output layer. The neurons (nodes) are interconnected by weighted links. A special class of ANN are feed-forward networks, which propagate a signal from the input to output layer [24]. A schematic view of a feed-forward network is given in Fig. 1, where $\mathbf{X}$ denotes a vector of predictor variables, $\mathbf{Y}$ is a vector of response variables, $\mathbf{w}^{(\mathbf{i},\mathbf{h})}$ is a column-matrix of weighting coefficients between neurons in the input layer and the first hidden layer, and $\mathbf{w}^{(\mathbf{h},\mathbf{o})}$ is a column-matrix of weighting coefficients between neurons in the last hidden layer and the output layer. The term $w_{ij}^{(q-1,q)}$ denotes the weight between ith neuron from $(q-1)$th hidden layer and jth neuron from (q)th hidden layer. In order to improve performance of the neural network, there is an extra neuron assigned to each hidden layer and the output layer, with the role of sending a constant signal x_0 (bias) to all neurons in these layers.

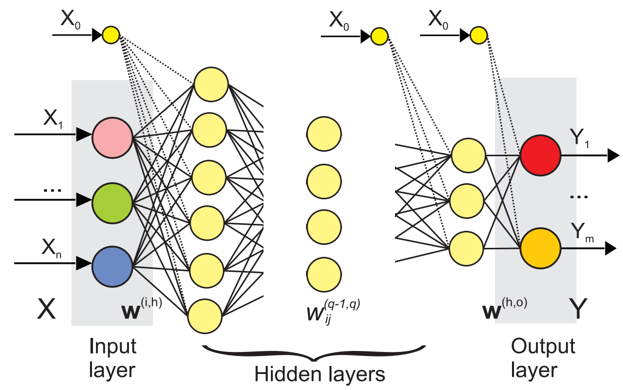

Fig. 1. Structure of feed-forward neural network.

According to the *sigma rule*, the total input $\alpha_j^{(q)}$ into processing element j in qth layer is a weighted sum of all outputs $x_i^{(q-1)}$ from the previous layer. In the same manner, the input signals into neurons of the output layer are calculated as a function of outputs from the last hidden layer. When input signal $\alpha_j^{(q)}$ passes through a neuron, it is processed and transformed to the output signal $x_j^{(q)}$ using an *activation function*. The activation function (AF) is necessary to transform the weighted sum of all signals hitting on a neuron so as to determine its firing intensity [17]. Some of the frequently used activation functions are: *Gaussian*, *log*, *sigmoid*, *bipolar sigmoid*, *sine*, *hyperbolic tangent (TANH)*. Characteristics of various AFs are given in details in [25].

The aim of the learning process is to set weights to the values for which the difference between desired and calculated values (error function) will be minimal. One of the most popular learning algorithms that minimizes error function is a *backpropagation algorithm*. According to the backpropagation algorithm, the error is propagated backward through the network and the weights are adjusted during a number of iterations. The procedure progresses until it reaches convergence of the calculated and expected outputs [17]. There are many variations of the backpropagation learning algorithm. In this work we used some of the best known: The Backpropagation gradient descent algorithm (BPGD) [24] and the Resilient propagation algorithm (RPROP) [26–29].

Design of an ANN is specified by network architecture (such as the number of hidden layers and neurons, type of AFs, etc.) and learning rules. Both the architecture and learning rules are very important, thus good selection of these will give better performance of the network [17]. This task is still an unsolved issue and most researchers use a trial-and-error method to find a