Advances in Engineering Software 88 (2015) 63–72

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Assessment of artificial neural network and genetic programming as predictive tools

Amir H. Gandomi [a,b,*], David A. Roke [b,1]

[a] BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48824, USA
[b] Department of Civil Engineering, The University of Akron, Akron, OH 44325, USA

ARTICLE INFO

Article history:
Received 14 July 2014
Received in revised form 9 April 2015
Accepted 26 May 2015
Available online 19 June 2015

Keywords:
Artificial neural networks
Genetic programming
Over-fitting
Explicit formulation
Punching shear
RC slabs
Parametric study

ABSTRACT

Soft computing techniques have been widely used during the last two decades for nonlinear system modeling, specifically as predictive tools. In this study, the performances of two well-known soft computing predictive techniques, artificial neural network (ANN) and genetic programming (GP), are evaluated based on several criteria, including over-fitting potential. A case study in punching shear prediction of RC slabs is modeled here using a hybrid ANN (which includes simulated annealing and multi-layer perception) and an established GP variant called gene expression programming. The ANN and GP results are compared to values determined from several design codes. For more verification, external validation and parametric studies were also conducted. The results of this study indicate that model acceptance criteria should include engineering analysis from parametric studies.

1. Introduction

Empirical modeling and formulation by soft computing techniques remain highly-researched topics, especially for engineering modeling [25]. Soft computing-based models differ from conventional models that are based on engineering principles (e.g., elasticity and plasticity theories), as they are based on experimental data rather than theoretical derivations. Soft computing-based models are usually complex and often cannot build an explicit formula. Therefore, they are most appropriate for use as a part of a computer program, limiting their applicability.

The most well-known soft computing predictive tool is the artificial neural network (ANN), which has been used successfully in structural engineering modeling (e.g. [36]. ANNs are inspired by biological neural networks [30]. Although ANNs typically build "black box" models, explicit formulas can be derived for a trained ANN model. A derivative-free optimization algorithm should be added to the training process of the ANN algorithm to avoid local minima, which lead to false convergence of the ANN model [38]. Some researchers have already combined ANN and global optimization algorithms to improve ANN efficiency (e.g., [41,46].

Another robust soft computing technique for modeling is genetic programming (GP), which is inspired by the principle of Darwinian natural selection. The machine code generated by GP can be translated as a mathematical formula, which makes it very suitable for mathematical modeling. GPs, especially new variants such as gene expression programming (GEP), have been successfully applied to several engineering problems, particularly in structural engineering (e.g., [21].

In this study, the performance of ANN and GP techniques are evaluated based on several criteria, including over-fitting potential, parametric study results, and simplicity of the generated formulas. To demonstrate this performance comparison, the punching shear strength of reinforced concrete slabs is modeled using a comprehensive database containing 241 experimental test results. We present the explicit slab strength prediction formulas from a well-trained ANN and from a proposed GP model. A subsequent parametric study was carried out to evaluate the trends of the ANN and GP models with respect to each parameter. The results show that although the ANN model outperforms the GP model in terms of error and correlation, it tends to be overfitted (with respect to design code values) due to its complexity. The GP models tend to have acceptable error and correlation characteristics

* Corresponding author at: BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48824, USA. Tel.: +1 (234) 788 0619; fax: +1 (330) 972 2060.

E-mail addresses: a.h.gandomi@gmail.com, ag72@uakron.edu (A.H. Gandomi), roke@uakron.edu (D.A. Roke).

URLs: http://gozips.uakron.edu/~ag72/ (A.H. Gandomi), http://www.uakron.edu/engineering/about-us/people-directory/profile.dot?identity=1641796 (D.A. Roke).

1 Tel.: +1 330 972 6813; fax: +1 330 972 6020.

http://dx.doi.org/10.1016/j.advengsoft.2015.05.007
0965-9978/

while performing well in the parametric studies (with respect to the physics of the problem as verified by the design codes).

2. Methodologies: Soft computing techniques

Soft computing includes, but is not limited to, evolutionary algorithms, ANNs, support vector machine and fuzzy logic. Soft computing predictive tools have wide-ranging applications and are often used to model the nonlinear relationship between input parameters and output value(s). Advances in computer hardware have made soft computing techniques more efficient. In addition, soft computing techniques may be used to model problems where conventional approaches, such as regression analysis, fail or perform poorly (e.g., [34]. In this study, two of the most well-known soft computing techniques, ANN and GP, are applied to an engineering case study and their results are analyzed and compared.

2.1. Artificial neural networks

Artificial neural networks (ANNs) were first developed in the early 1940s [42]. ANNs are predictive tools used to build a mathematical model for a complex system. Multi-layer perception (MLP) ANN [14] is the most well-known class of ANNs. MLP ANNs usually have feed-forward architectures and are typically trained with back-propagation algorithms. MLP networks consist of one input layer and one output layer, with at least one additional hidden layer. Each layer has a number of nodes and contains one or more processing units. Each unit in the MLP is fully interconnected with weighted connections (w_{ij}) to the units in the subsequent layer [6]. The output (Y) is obtained by passing the sum of the products of the inputs and weights through an activation function. Fig. 1 shows a schematic of a simple MLP ANN.

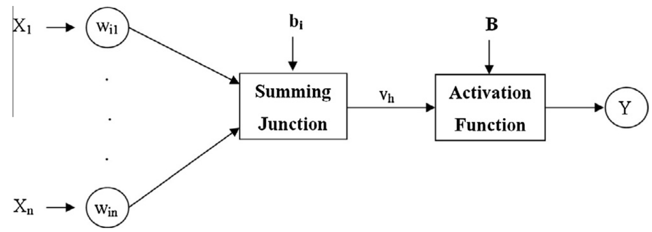

Fig. 1. Schematic of an MLP neural network.

2.1.1. Hybrid simulated annealing-artificial neural network

ANNs are "trained" from a set of data known as the training set. During the training process, the network's weights are optimized. The training procedure consists of two main steps: initialization and optimization [4]. In the initialization process, initial values are assigned to the weights of the network, either randomly or via a global optimization method such as simulated annealing (SA). The optimization process typically utilizes a gradient-based algorithm that is suitable for local search. Therefore, to be successful, the optimization process requires a starting point obtained from a global search. A robust training process needs both the initialization and optimization processes [38]. A schematic of the hybrid ANN algorithm is presented in Fig. 2. In the first step, the initial weights are optimized by SA. In the second step, the MLP network is used to find the final weights of the network.

SA is very useful for solving nonlinear problems with multiple local optima [1]. SA is a global search algorithm that makes use of the Metropolis algorithm [40] for computer simulation of annealing. Annealing is a process in which a metal is heated to a high temperature and thereafter it is gradually cooled to relieve thermal stresses. The cooling process is simulated by SA to optimize a function in a certain design space. The objective function relates to the energy state, and changing the set of design variables corresponds to changing the crystalline structural state [22]. The abilities and shortcomings of SA are well summarized by Ingber [33]. For the initialization step, SA randomly perturbs the weights of the network during the iterations. When the weights are perturbed, the network performance is evaluated based on the defined objective function. The cooling schedule during iterations can be linear or exponential, and additional iterations at a specific temperature may occur if the objective function is improved [32]. Each time a new solution is generated, the algorithm decides whether the new solution should be accepted or rejected. Metropolis et al. [40] expressed the probability (P_a) of accepting a new solution as:

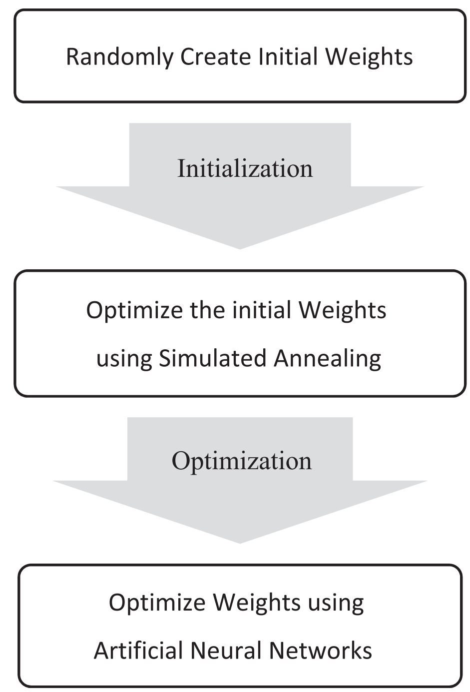

Fig. 2. Schematic of the hybrid ANN algorithm.

$$P_a(\Delta E, y) = \begin{cases} e^{-\frac{\kappa \Delta E}{y}} & \Delta E > 0 \\ 1 & \Delta E \leqslant 0 \end{cases} \tag{1}$$

where ΔE is the error, y is the current temperature, and κ is an acceptance constant that depends on the network's weights range and inputs. As y decreases, the algorithm becomes more selective [38].

When implementing SA for ANN initialization, the most important factor is adjusting the acceptance constant, which is a function of the temperature range, the training dataset, and the allowed weight values [26]. Generally, cooling schedules progress gradually from high temperatures to lower temperatures until a specified target temperature is reached [39]. However, the hybrid SA-ANN method requires the temperature to increase and decrease periodically, following a linear or temperature cycling cooling schedule. In some cases, the temperature cycling schedule can outperform the linear cooling schedule (e.g., [4,38]. Therefore, the temperature cycling cooling schedule is used in this study.

2.2. Genetic programming

Genetic programming (GP) is a predictive tool that creates computer programs by emulating the biological evolution of living