Expert Systems with Applications 38 (2011) 161–168

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Weighted operation structures to program strengths of concrete-typed specimens using genetic algorithm

Hsing-Chih Tsai *

Department of Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan

ARTICLE INFO

Keywords:
Weighted formula
Prediction
Concrete strength
Genetic programming

ABSTRACT

This study introduces weighted operation structures (WOS) to program engineering problems, in which each WOS adopts a fixed binary tree topology. The first WOS layer serves as the parameter input entrance. The target is produced at the eventual layer using both values and a mathematical formula. Each WOS element is operated by two front nodal inputs, an undetermined function, and two undetermined weights to produce one nodal output. This study proposes the novel concept of introducing weights into a WOS. Doing so provides two unique advantages: (1) achieving a balance between the influences of two front inputs and (2) incorporating weights throughout the generated formulas. Such a formula is composed of a certain quantity of optimized functions and weights. To determine function selections and proper weights, genetic algorithm is employed for optimization. Case studies herein focused on three kinds of concrete-typed specimen strengths: (1) concrete compressive strength, (2) deep beam shear strength, and (3) squat wall shear strength. Results showed that the proposed WOS can provide accurate results that nearly equal the results obtainable using the familiar neural network. The weighted formula, however, offers a distinct advantage in that it can be programmed for practical cases.

1. Introduction

Artificial intelligence (AI) approaches have been applied in neural networks (NN), fuzzy logic, support vector machines, genetic algorithms (GA), genetic programming (GP) and so on. Each approach offers merits in particular applications. NN is the most familiar AI approach for tasks involving AI learning, and many NN derivatives have been developed and applied in various categories (Behzad, Asghari, Eazi, & Palhang, 2009; Mehrjoo, Khaji, Moharrami, & Bahreininejad, 2008; Tsai, 2009). However, NN is primarily argued as a black box model due to the massive size of nodes and connections within its structure. Since being first proposed by Koza (1992), GP has earned significant attention in terms of its ability to model nonlinear relationships for input–output mappings. GP creates solutions as programs (formulas) to solve problems with an operation tree. However, coefficient constants are quite important to balance nodal input influences in a programmed formula. This paper introduces weights to tree connections and finally generates a fully weighted formula. The proposed model is a weighted operation structure (WOS) and is optimized by GA.

Concrete is a complex material widely used due to its strong capacity to withstand compression. However, identifying the specific mechanics of concrete and its derivatives is a difficult task. Predicting strength capacity is made more difficult still due to the reliance of such on identified mechanics. Using an AI approach represents an alternative method for predicting strengths that is able to achieve a high level of accuracy.

The main purpose of this paper was to develop a WOS and then apply it to predict the specimen strengths of concrete and its derivatives. Results focused on assessing WOS performance and predicted strengths with values and formulas.

The remaining sections of this paper include Section 2: proposed WOS and GA optimization; Section 3: predicting specimen strengths of concrete cylinders, reinforced-concrete deep beams, and reinforced-concrete squat walls; Section 4: suggestions for further studies and future work, and Section 5: conclusions.

2. Weighted operation structure optimized using a genetic algorithm

The WOS adopted a layer number (NL) setting (see Fig. 1), with each node x_i^1 in the first layer one of the input parameters (including a unit parameter "1"):

$$x_i^1 = \text{one}(1 \quad P_1 \quad P_2 \quad \ldots \quad P_j \quad \ldots \quad P_{NI}), \quad j = 0 \sim NI, \tag{1}$$

* Tel.: +886 2 27301277; fax: +886 2 27301074.
E-mail address: tsaihsingchih@gmail.com

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.06.034

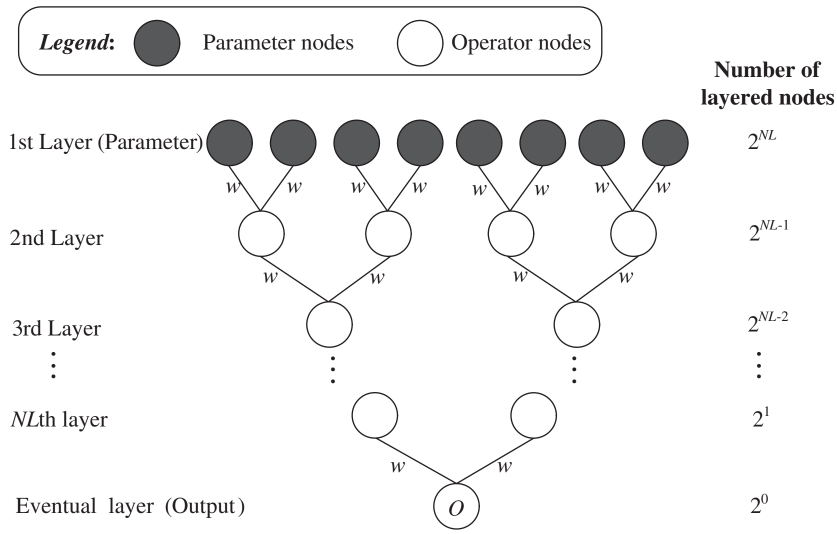

Fig. 1. Weighted operation structure.

where x_i^1 represents nodes in the first layer and i denotes a related node number; P_j is the jth input parameter; and NI is the number of inputs. Each x_i^1 node selects one attached P_j.

Layers from the second to the 'eventual' (i.e., the layer immediately following the NLth) use operation nodes to calculate values in top-down order (see Fig. 2). For two adjacent layers, x represents front nodes, which are treated as layer inputs, and y denotes back nodes, which are treated as layer outputs. A y is calculated by operations of two front x values. Operations involve two weights (w) and one function (F). Elements in GP and WOS are different in terms of connection weight. Number of scenarios for a GP element depends on number of function candidates and is, therefore, finite. However, WOS introduces two complex weights to balance nodal input influences. Number of scenarios for each WOS element is infinite. Such an improvement provides merits that include: (1) ability to search a wide territory range with an infinite number of combination variations and (2) the presence of weight coefficients throughout output formulas. Certainly, calculating both appropriate weights and functions for a WOS is more time consuming than for a GP, but worthy.

The layer after the NLth is called the "eventual layer" in the final output/formula. The node in the eventual layer is either an output node (O) or a back operation node (y). Therefore, parameter selections should be applied to the 2^{NL} nodes in the first layer. There are 2^{NL-1}, 2^{NL-2}, ..., and 2^0 operation nodes in the 2nd, 3rd, ..., and eventual layers, respectively. Every operation node y is operated by a set of defined functions to connect to two front nodal inputs of x_i and x_j with weights of w_i and w_j:

$$y = F(w_i, w_j, x_i, x_j) = \text{one of}\begin{cases} f_1 &= w_i x_i \\ f_2 &= (w_i x_i) + (w_j x_j) \\ f_3 &= (w_i x_i)(w_j x_j) \\ f_4 &= (w_i x_i)/(w_j x_j) \\ f_5 &= |w_i x_i|^{w_j x_j} \\ f_6 &= \sin(w_i x_i) \\ f_7 &= \cos(w_i x_i) \\ f_8 &= \exp(w_i x_i) \\ f_9 &= \log|w_i x_i| \\ \ldots & \ldots \quad \ldots \\ f_{NF} &= \frac{1}{\sin(w_i * x_i) + \cos(w_j * x_j)} \end{cases}. \tag{2}$$

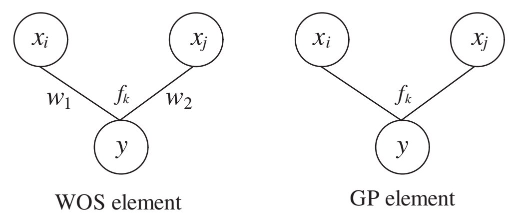

Fig. 2. Weighted operation structure and genetic programming elements.

This paper adopted the first nine functions in Eq. (2) for every F selection. The f_1 is designed to inherit the left-hand side front nodes with w_i scaling and it is a one-handed operator (use "C" to represent). f_2 indicates a weighted "+" operator and it is a two-handed operator. f_3, and f_4 are both two-handed operators related to "×" and "/" respectively. f_5 is a two-handed power("^") operator; f_6, f_7, f_8, and f_9 are all one-handed functional operators ("sin", "cos", "exp", and "log"). Generally, each adopted function should be unique. However, exceptions are permitted based on user requirements. The last function in Eq. (2) is an example of a case in which the user has confidence in the role of a lucky guess function. Although such function can be reproduced by combinations of the 4th, 5th, 1st, and 3rd equations, assigning such an appropriate function as a candidate can greatly improve convergence speed.

An answer O of a two-layered WOS can be represented as

$$O = y = F_1\{w_1, w_2,\ F_2(w_3, w_4, P_1, P_2),\ F_3(w_5, w_6, P_3, P_4)\}. \tag{3}$$

For instance, a WOS example output might be (see Fig. 3):

$$O = f_4\{0.1,\quad 0.2,\quad f_2(0.3,\quad -0.4,\quad P_1\quad P_2) \\ f_3(0.5\quad 0.6\quad 1\quad P_3)\} = \frac{0.3P_1 - 0.4P_2}{0.5P_4 \times 0.6P_3}. \tag{4}$$

In terms of GP, the output might be (see Fig. 3):

$$O = \frac{P_1 + P_2}{1.5 \times P_3}. \tag{5}$$

The coefficient term in Eq. (5) is produced by setting a coefficient for a parameter node corresponding to the unit parameter "1" in WOS. However, such a coefficient costs (wastes) a branch of the structure (tree). While this optimum coefficient exceeds the search domain, GP will identify other derivatives to cover this optimum coefficient. Numerically, a large coefficient in WOS may be reproduced by