Geoscience Frontiers 7 (2016) 33–44

Contents lists available at ScienceDirect

China University of Geosciences (Beijing)

Geoscience Frontiers

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

Research paper

State-of-the-art review of some artificial intelligence applications in pile foundations

Mohamed A. Shahin*

Department of Civil Engineering, Curtin University, Perth, WA 6845, Australia

A R T I C L E I N F O

Article history:
Received 11 August 2014
Received in revised form
17 October 2014
Accepted 26 October 2014
Available online 11 November 2014

Keywords:
Artificial intelligence
Pile foundations
Artificial neural networks
Genetic programming
Evolutionary polynomial regression

A B S T R A C T

Geotechnical engineering deals with materials (e.g. soil and rock) that, by their very nature, exhibit varied and uncertain behavior due to the imprecise physical processes associated with the formation of these materials. Modeling the behavior of such materials in geotechnical engineering applications is complex and sometimes beyond the ability of most traditional forms of physically-based engineering methods. Artificial intelligence (AI) is becoming more popular and particularly amenable to modeling the complex behavior of most geotechnical engineering applications because it has demonstrated superior predictive ability compared to traditional methods. This paper provides state-of-the-art review of some selected AI techniques and their applications in pile foundations, and presents the salient features associated with the modeling development of these AI techniques. The paper also discusses the strength and limitations of the selected AI techniques compared to other available modeling approaches.

1. Introduction

Over the last decade, artificial intelligence (AI) has been applied successfully to virtually every problem in geotechnical engineering. Examples of the available AI techniques are artificial neural networks (ANNs), genetic programming (GP), evolutionary polynomial regression (EPR), support vector machines (SVM), M5 model trees, and k-nearest neighbors (Elshorbagy et al., 2010). Of these, ANNs are by far the most commonly used AI technique in geotechnical engineering. More recently, GP and EPR have been frequently used in geotechnical engineering and have proved to be successful. The main focus of the current paper is on the use of ANNs, GP, and EPR in pile foundations.

The behavior of pile foundations in soils is complex, uncertain, and not yet entirely understood. This fact has encouraged many researchers to apply the AI techniques for prediction and modelling of the behavior of pile foundations, including the ultimate bearing capacity, settlement estimation, and load-settlement response. The objective of this paper is to provide an overview of the salient features relevant to the process and operation of ANNs, GP, and EPR, and to present a review of their applications to date in pile foundations. The paper also discusses most of the current challenges as well as future directions in relation to the use of AI techniques in geotechnical engineering prediction and modelling.

2. Overview of artificial intelligence

Artificial intelligence (AI) is a computational method that attempts to mimic, in a very simplistic way, the human cognition capability so as to solve engineering problems that have defied solution using conventional computational techniques (Flood, 2008). The essence of AI techniques in solving any engineering problem is to learn by examples of data inputs and outputs presented to them so that the subtle functional relationships among the data are captured, even if the underlying relationships are unknown or the physical meaning is difficult to explain. Thus, AI models are data-driven models that rely on the data alone to determine the structure and parameters that govern a phenomenon (or system), with less assumptions about the physical behavior of the system. This is in contrast to most physically-based models that use the first principles (e.g., physical laws) to derive the underlying relationships of the system, which usually justifiably simplified with many assumptions and require prior knowledge about the nature of the relationships among the data. This is one of

* Tel.: +61 8 9266 1822.
E-mail address: m.shahin@curtin.edu.au.
Peer-review under responsibility of China University of Geosciences (Beijing).

http://dx.doi.org/10.1016/j.gsf.2014.10.002
1674-9871/

the main benefits of AI techniques when compared to most physically-based empirical and statistical methods.

The AI modeling philosophy in attempting to capture the relationship between a historical set of model inputs and the corresponding outputs is similar to a number of conventional statistical models. For example, imagine a set of x-values and corresponding y-values in two-dimensional space, where $y = f(x)$. The objective is to find the unknown function f that relates the input variable x to the output variable y. In a linear regression statistical model, the function f can be obtained by changing the slope $\tan\phi$ and intercept β of the straight line in Fig. 1a, so that the error between the actual outputs and the outputs of the straight line is minimized. The same principle is used in AI models. Artificial intelligence can form the simple linear regression model by having one input and one output (Fig. 1b). Artificial intelligence uses available data to map between the system inputs and the corresponding outputs using machine learning by repeatedly presenting examples of the model inputs and outputs (training) in order to find the function $y = f(x)$ that minimizes the error between the historical (actual) outputs and the outputs predicted by the AI model.

If the relationship between x and y is non-linear, statistical regression analysis can be applied successfully only if prior knowledge of the nature of the non-linearity exists. On the contrary, this prior knowledge of the nature of the non-linearity is not required for AI models. In the real world, it is likely that complex and highly non-linear problems are encountered, and in such situations, traditional regression analyses are inadequate (Gardner and Dorling, 1998). In this section, a brief overview of three selected AI techniques (i.e., ANNs, GP, and EPR) is presented below.

2.1. Artificial neural networks

Artificial neural networks (ANNs) are a form of AI that attempt to mimic the function of the human brain and nervous system. Although the concept of ANNs was first introduced in 1943 (McCulloch and Pitts, 1943), research into applications of ANNs has blossomed since the introduction of the back-propagation training algorithm for feed-forward multi-layer perceptrons (MLPs) in 1986 (Rumelhart et al., 1986). Many authors have described the structure and operation of ANNs (e.g., Zurada, 1992; Fausett, 1994). Typically, the architecture of ANNs consists of a series of processing elements (PEs), or nodes, that are usually arranged in layers: an input layer, an output layer, and one or more hidden layers, as shown in Fig. 2.

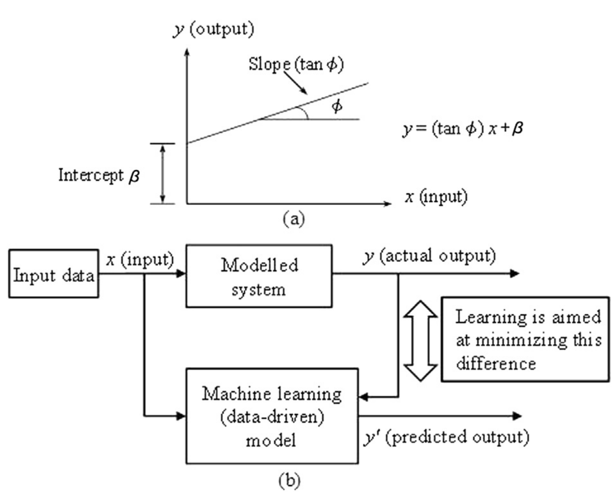

Figure 1. Linear regression versus artificial intelligence (AI) modeling: (a) linear regression modeling (after Shahin et al., 2001); (b) AI data-driven modeling (adapted from Solomatine and Ostfeld, 2008).

The input from each PE in the previous layer x_i is multiplied by an adjustable connection weight w_{ji}. At each PE, the weighted input signals are summed and a threshold value θ_j is added. This combined input I_j is then passed through a non-linear transfer function $f(.)$ to produce the output of the PE y_j. The output of one PE provides the input to the PEs in the next layer. This process is summarized in Eqs. (1) and (2), and illustrated in Fig. 2.

$$I_j = \sum w_{ji}x_i + \theta_j \quad \text{summation} \tag{1}$$

$$y_j = f\left(I_j\right) \quad \text{transfer} \tag{2}$$

The propagation of information in an ANN starts at the input layer, where the input data are presented. The network adjusts its weights on the presentation of a training data set and uses a learning rule to find a set of weights that produces the input/output mapping that has the smallest possible error. This process is called *learning* or *training*. Once the training of the model has successfully accomplished, the performance of the trained model needs to be validated using an independent validation set. The main steps involved in the development of an ANN, as suggested by Maier and Dandy (2000), are illustrated in Fig. 3 and discussed in some depth in Shahin (2013).

2.2. Genetic programming

Genetic programming (GP) is an extension of genetic algorithms (GA), which are evolutionary computing search (optimization) methods that are based on the principles of genetics and natural selection. In GA, some of the natural evolutionary mechanisms, such as reproduction, cross-over, and mutation, are usually implemented to solve function identification problems. GA was first introduced by Holland (1975) and developed by Goldberg (1989), whereas GP was invented by Cramer (1985) and further developed by Koza (1992). The difference between GA and GP is that GA is generally used to evolve the best values for a given set of model parameters (i.e., parameters optimization), whereas GP generates a structured representation for a set of input variables and corresponding outputs (i.e., modeling or programming).

Genetic programming manipulates and optimizes a population of computer models (or programs) proposed to solve a particular problem, so that the model that best fits the problem is obtained. A detailed description of GP can be found in many publications (e.g., Koza, 1992), and a brief overview is given herein. The modelling steps by GP start with the creation of an initial population of computer models (also called *chromosomes*) that are composed of two sets (i.e., a set of functions and a set of terminals) that are defined by the user to suit a certain problem. The functions and terminals are selected randomly and arranged in a tree-like structure to form a computer model that contains a root node, branches of functional nodes, and terminals, as shown by the typical example of GP tree representation in Fig. 4. The functions can contain basic mathematical operators (e.g., +, −, ×, /), Boolean logic functions (e.g., AND, OR, NOT), trigonometric functions (e.g., sin, cos), or any other user-defined functions. The terminals, on the other hand, may consist of numerical constants, logical constants, or variables.

Once a population of computer models has been created, each model is executed using available data for the problem at hand, and the model fitness is evaluated depending on how well it is able to solve the problem. For many problems, the model fitness is measured by the error between the output provided by the model