Advances in Engineering Software 95 (2016) 68–81

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Artificial neural networks in the calibration of nonlinear mechanical models

Tomáš Mareš, Eliška Janouchová, Anna Kučerová*

Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7, 166 29, Prague 6, Czech Republic

ARTICLE INFO

Article history:
Received 27 July 2015
Revised 14 December 2015
Accepted 27 January 2016
Available online 1 March 2016

Keywords:
Artificial neural network
Multi-layer perceptron
Parameter identification
Principal component analysis
Sensitivity analysis
Affinity hydration model
Concrete

ABSTRACT

Rapid development in numerical modelling of materials and the complexity of new models increase quickly together with their computational demands. Despite the growing performance of modern computers and clusters, calibration of such models from noisy experimental data remains a nontrivial and often computationally intensive task. Layered neural networks provide a robust and efficient technique for overcoming the time-consuming simulations of calibrated models. The potential advantages of neural networks include simple implementation and high versatility in approximating nonlinear relationships. Therefore, there were several approaches proposed in literature for accelerating the calibration of nonlinear models by neural networks. This contribution reviews and compares three possible strategies based on approximating (i) the model response, (ii) the inverse relationship between the model response and its parameters and (iii) an error function quantifying how well the model fits the data. The advantages and drawbacks of particular strategies are demonstrated with the calibration of four parameters of an affinity hydration model from simulated data as well as from experimental measurements. The affinity hydration model is highly nonlinear but computationally cheap, thus allowing its calibration without any approximation and better quantification of results obtained by the examined calibration strategies. This paper can be viewed as a guide for engineers to help them develop an appropriate strategy for their particular calibration problems.

1. Introduction

Development in numerical modelling allows for the description of complex phenomena in material or structural behaviour. The resulting models are, however, often highly nonlinear and defined by many parameters which have to be estimated so as to properly describe the investigated system and its behaviour. The aim of model calibration is thus to rediscover unknown parameters, knowing the experimentally obtained response of a system to given excitations. The principal difficulty of model calibration lies in the fact that while the numerical model of an experiment represents a well-defined mapping from input (model, material, structural, or other parameters) to output (structural response), there is no guarantee that an inverse relationship even exists.

The most widely used approach to parameter identification is usually an error minimisation technique, where the distance between parameterised model predictions and observed data is minimised [1]. Since the inverse relationship (mapping of model outputs to inputs) is often ill-posed, the error minimisation technique leads to a difficult optimisation problem that is highly nonlinear and multi-modal. Therefore, the choice of an appropriate identification strategy is not trivial.

Another approach intensively developed during the last decade is based on Bayesian updating of uncertainty in parameters' description [2,3]. Uncertainty in observations is expressed by a corresponding probability distribution and employed for estimating the so-called posterior probabilistic description of identified parameters together with prior expert knowledge about the parameter values [4,5]. The unknown parameters are thus modelled as random variables originally endowed with prior expert-based probability density functions which are then updated using the observations to the posterior density functions. While the error minimisation techniques lead to a single point estimate of the parameters' values, the result of Bayesian inference is a probability distribution that summarises all available information about the parameters. Another very important advantage of Bayesian inference consists in treating the inverse problem as a well-posed problem in an expanded stochastic space.

Despite progress in uncertainty quantification methods [6,7], the additional information provided by Bayesian inference is

* Corresponding author. Tel.: +420 2 2435 5326; fax: +420 2 2431 0775.
E-mail addresses: marestom87@gmail.com (T. Mareš), eliska.janouchova@fsv.cvut.cz (E. Janouchová), anicka@cml.fsv.cvut.cz (A. Kučerová).

http://dx.doi.org/10.1016/j.advengsoft.2016.01.017
0965-9978/

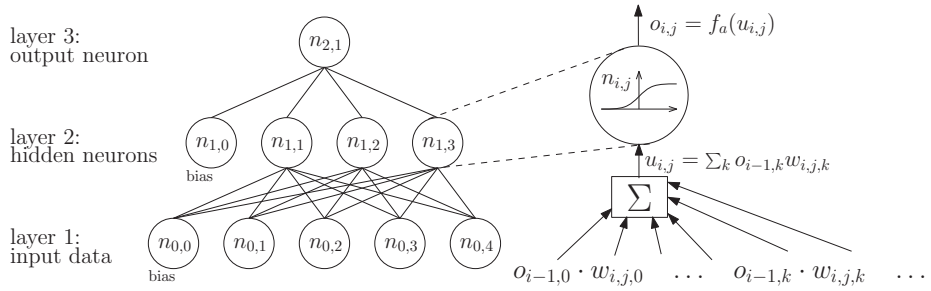

Fig. 1. Architecture of a multi-layer perceptron.

generally related to more time-consuming computations. In many situations, the single point estimate approach remains the only feasible one, and the development of efficient tools suitable for this strategy is still a current topic. Within the last several decades, much attention was devoted to so-called intelligent methods of information processing, particulary soft computing methods such as artificial neural networks (ANNs), evolutionary algorithms or fuzzy systems [8]. A review of soft computing methods for parameter identification can be found e.g. in [9]. In this paper, we focus on applications of ANNs in the single point approach to parameter identification. In particular, we elaborate upon our previous work presented in [10,11] with the goal of presenting a detailed and comprehensive comparison of three different strategies for using ANNs in parameter identification problems.

The next section briefly recall the basics of ANNs. The classification of the different applications of ANNs in calibration problems is introduced in Section 3 and a description of an illustrative example — affinity hydration model for concrete — follows in Section 4. In the context of this particular example, the calibration strategies are then described in detail in five sections, beginning with training data preparation and sensitivity analysis in Section 5. Neural network inputs and outputs in particular strategies are discussed in Section 6, and training with topology determination is described in Section 7. Verification and validation on simulated and experimental data are summarised in Sections 8 and 9, respectively. Finally, results are discussed in Section 10.

2. Artificial neural networks

Artificial neural networks (ANNs) [12,13] are powerful computational systems consisting of many simple processing elements — so-called neurons — connected together to perform tasks analogous to biological brains. Their main feature is the ability to change their behaviour based on the external information that flows through an ANN during the learning (training) phase.

A particular type of ANN is the so-called feed-forward neural network which consists of neurons organised into layers, where outputs from one layer are used as inputs into the following layer, see Fig. 1. There are no cycles or loops in the network, no feedback connections. The most frequently used example is the multilayer perceptron (MLP) with a sigmoid transfer function and a gradient descent method of training called the back-propagation learning algorithm. In practical usage, MLPs are known for their ability to approximate nonlinear relationships and therefore, when speaking about an ANN, the MLP is considered in the following text.

The input layer represents a vector of input parameters which are directly the outputs of the input layer. The outputs $o_{i-1,k}$ of the $(i-1)$th layer are multiplied by a vector of constants $w_{i,j,k}$, the so-called synaptic weights, summarized and used as inputs $u_{i,j}$ into the jth neuron in the following ith layer. Elements in the hidden and output layers — neurons — are defined by an activation function $f_a(u_{i,j})$ which is applied to the input and produces the output value of the jth neuron in the ith layer, i.e.

$$o_{i,j} = f_a\left(u_{i,j}\right) \quad \text{where} \quad u_{i,j} = \sum_{k=0}^{K}\left(o_{i-1,k} w_{i,j,k}\right). \tag{1}$$

The synaptic weights $w_{i,j,k}$ are parameters of an ANN to be determined during the training process. K is the number of neurons in the $i-1$ layer. The type of activation function is usually chosen in accordance with the type of function to be approximated. In the case of continuous problems, the sigmoid activation function given as

$$o_{i,j} = f_a\left(u_{i,j}\right) = \frac{1}{1+e^{-u_{i,j}}} \tag{2}$$

is the most common choice.

One bias neuron is usually added to the input and hidden layers. It does not contain an activation function and only has a constant value. Its role is to enable the shift in the value of a sum over the outputs of its neighbouring neurons before this sum enters as the input into the neurons in the following layer. The value of biases is determined by the training process together with the synaptic weights.

Despite the popularity of ANNs there are very few published recommendations for selecting a particular ANN architecture. The authors, e.g. in [14,15], show that an ANN with any of a wide variety of continuous nonlinear hidden layer activation functions and one hidden layer having an arbitrarily large number of units suffices for a "universal approximation" property. Therefore, we limit our numerical experiments to such a case. The number of units in the input and the output layers is usually given by the particular problem under consideration, but there is no theory yet which specifies the number of units in the hidden layer. On the one hand, too few hidden units leads to large prediction errors. On the other hand, a large number of hidden units may cause so-called overfitting, where the ANN provides precise outputs for training samples but fails when it encounters previously unseen samples. In such situations, an ANN tries to fit the training data despite increasing oscillations in the intermediate space.

To overcome this problem, a model selection technique [16] has to be applied in order to perform a guided choice of an ANN topology. Recent approaches encompass e.g. growing-pruning methods (see e.g. [17]) or more complex techniques designed for optimising ANN topology, such as meta-learning [18,19]. Here, we employ a simple and general strategy to evaluate a particular ANN topology: cross-validation, which does not involve any probabilistic assumptions or dependencies on an identification problem. The idea of cross-validation is based on repeating evaluation of the prediction error obtained by individual ANNs for a chosen subset of training data and selecting the ANN with the smallest averaged prediction errors. Comparing to a well-known model validation on some independent set of data, the advantage of cross-validation