Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Advances in Engineering Software 39 (2008) 505–513
ADVANCES IN ENGINEERING SOFTWARE
www.elsevier.com/locate/advengsoft

Structural reliability analysis using Monte Carlo simulation and neural networks

João B. Cardoso *, João R. de Almeida, José M. Dias, Pedro G. Coelho

Faculty of Science and Technology, New University of Lisbon, 2829-516 Caparica, Portugal

Received 10 September 2003; received in revised form 13 March 2007; accepted 30 March 2007
Available online 22 May 2007

Abstract

This paper examines a methodology for computing the probability of structural failure by combining neural networks (NN) and Monte Carlo simulation (MCS). MCS is a powerful tool, simple to implement and capable of solving a broad range of reliability problems. However, its use for evaluation of very low probabilities of failure implies a great number of structural analyses, which can become excessively time consuming. The proposed methodology makes use of the capability of a NN to approximate a function for reproducing structural behavior, allowing the computation of performance measures at a much lower cost. This approach seems very attractive, and its main challenge lies in the ability of a NN to approximate accurately complex structural response. In order to assess the validity of this methodology, a test function and two structural examples are presented and discussed. The second example is also used to show how this methodology can be used to perform reliability-based structural optimization.

Keywords: Reliability-based optimization; Structural reliability; Monte Carlo simulation; Neural networks

1. Introduction

The structural designer must verify, within a prescribed safety level, the serviceability and ultimate conditions, commonly expressed by the inequality: $S_d < R_d$, where S_d represents the action effect and R_d the resistance. The intrinsic random nature of material properties and actions is actually considered by the Eurocodes [1,2], which classify the methods available to deal with this randomness in three levels:

- Semi-probabilistic or level 1 methods, the most used in common practice, where the probability of failure is indirectly considered through the definition of characteristic values and the application of partial safety indexes.
- Approximate probabilistic or level 2 methods such as the first order or second order reliability methods (FORM/SORM) where the probability of failure is based on the reliability index β [3].
- Exact probabilistic or level 3 methods, where the probability of failure is computed from the joint probability distribution of the random variables associated with the actions and resistances.

Theory and methods for structural reliability have been developed substantially in the last few years and they are actually a useful tool for evaluating rationally the safety of complex structures or structures with unusual designs. Recent developments allow anticipating that their application will gradually increase, even in the case of common structures.

Monte Carlo simulation (MCS) is a simulation method of level 3 that presents the following characteristics: it can be applied to many practical problems, allowing the direct consideration of any type of probability distribution for the

* Corresponding author. Tel.: +351 212948567; fax: +351 212948531.
E-mail address: jbc@fct.unl.pt (J.B. Cardoso).

0965-9978/$ - see front matter
doi:10.1016/j.advengsoft.2007.03.015

random variables; it is able to compute the probability of failure with the desired precision; it is easy to implement.

However, despite the advantages it presents, the use of this method is not widespread in structural reliability because it is not efficient when compared to level 2 methods. In fact, MCS requires a great number of structural analyses, one for each sample of the set of random variables. The number of analyses needed to evaluate the probability of failure of a structure with a prescribed precision depends on the order of magnitude of that probability. As the values of the probability of failure associated to the ultimate limit states vary normally between 10^{-4} and 10^{-6}, the number of analyses to be performed for ensuring a 95% likelihood that the actual probability be within 5% of the computed one must be at least 1.6×10^7 to 1.6×10^9, according to Shooman [4].

These analyses are frequently performed with the help of finite elements codes. Therefore, the computation time can be prohibitively high, especially when the structure exhibits non-linear behavior or the numerical model is rather complex.

To eliminate this drawback, it is proposed here the use of neural networks (NN) to approximate structural response. Once properly trained, a NN allows the determination of the structural performances with a very small number of operations and at a fraction of the cost of the corresponding structural analysis. This methodology allows the application of MCS to practical cases of great complexity where the direct use of this method would not be feasible.

To illustrate the techniques proposed, three examples are presented, considering respectively a mathematical function and two different structures. In the first example, a non-linear analytical function is approximated by means of NN. In the second, a linear elastic steel frame is designed according to the Eurocodes 1 and 3 [1,2], and subsequently the probability of failure of the frame is computed by a Monte Carlo approach in which NN are used for reproducing the structural response. In the third, a genetic algorithm is used in combination with NN approximation and Monte Carlo method to perform reliability-based optimization of a steel truss.

2. Monte Carlo simulation

A reliability problem is normally formulated using a failure function, $g(X_1, X_2, \ldots, X_n)$, where $X_1, X_2, \ldots, X_n$ are random variables. Violation of the limit state is defined by the condition $g(X_1, X_2, \ldots, X_n) \leqslant 0$ and the probability of failure, p_f, is expressed by the following expression [5]:

$$p_f = P[g(X_1, X_2, \ldots, X_n) \leqslant 0] = \int\int_{g(X_1, X_2, \ldots, X_n) \leqslant 0} \cdots \int f_{X_1, X_2, \ldots, X_n}(x_1, x_2, \ldots, x_n)\, dx_1\, dx_2 \ldots dx_n \quad (1)$$

where $(x_1, x_2, \ldots, x_n)$ are values of the random variables and $f_{X_1, X_2, \ldots, X_n}(x_1, x_2, \ldots, x_n)$ is the joint probability density function.

The Monte Carlo method allows the determination of an estimate of the probability of failure, given by

$$\bar{p}_f = \frac{1}{N} \sum_{i=1}^{N} I(X_1, X_2, \ldots, X_n) \quad (2)$$

where $I(X_1, X_2, \ldots, X_n)$ is a function defined by

$$I(X_1, X_2, \ldots, X_n) = \begin{cases} 1 & \text{if } g(X_1, X_2, \ldots, X_n) \leqslant 0 \\ 0 & \text{if } g(X_1, X_2, \ldots, X_n) > 0 \end{cases} \quad (3)$$

According to (2), N independent sets of values $x_1, x_2, \ldots, x_n$ are obtained based on the probability distribution for each random variable and the failure function is computed for each sample. Using MCS, an estimate of the probability of structural failure is obtained by

$$\bar{p}_f = \frac{N_H}{N} \quad (4)$$

where N_H is the total number of cases where failure has occurred.

3. Neural networks

NN are numerical algorithms inspired in the functioning of biological neurons. This concept was introduced by McCulloch and Pitts [6], who proposed a mathematical model to simulate neuron behavior. Use of NN has become widespread in several fields of engineering, such as structural mechanics [7,8] and structural reliability [9]. Papadrakakis et al. [10] presented an approach in which a NN was associated to MCS in order to obtain the probability of structural failure. A similar approach is proposed herein.

The model adopted is shown in Fig. 1, representing the neuron m that receives an input signal vector $x = [x_i, \ldots, x_L]^T$ from L input channels. It then computes the weighted sum of the components of x, multiplying each component x_k by a coefficient w_{mk} reflecting the importance of the input channel k. The neuron m activation, a_m, is given by the expression

$$a_m = \sum_{k=1}^{L} w_{mk} x_k + b_m \quad (5)$$

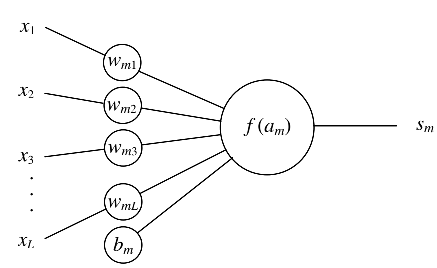

Fig. 1. Artificial neuron.