XLI CILAMCE
XLI Ibero-Latin American Congress on Computational Methods in Engineering
16-19 November, 2020 FOZ DO IGUAÇU PARANÁ - BRAZIL

A Machine Learning-based Constitutive Model for Nonlinear Analysis via Finite Element Method

Álefe F. Figueiredo[1], Saulo S. Castro[1], Roque L. S. Pitangueira[1], Samir S. Saliba[1]

[1]*Dept. of Structures Engineering, Federal University of Minas Gerais*
Av. Antônio Carlos, 6627, Pampulha, Belo Horizonte, 31270-901, Minas Gerais, Brazil
alefefreitas_11@hotmail.com, saullo9@yahoo.com.br, roque@dees.ufmg.br, samirsaliba@yahoo.com

Abstract. This paper addresses a machine learning technique in the context of constitutive modelling. Since it has been proven that multilayer perceptrons with the backprogapation algorithm are capable of approximating any class of functions, studies have been developed with the objective of using it as approximation functions for the nonlinear behaviour of complex material media. This is only possible because neural networks have a powerful adaptability, capability of learning and generalizability. In this context, a multilayer perceptron is trained with stress-strain results from a nonlinear analysis via finite element method with Mazars material in order to develop a neural network-based constitutive model. This implementation is carried out with the help of a recognized machine learning package in order to obtain more accurate results. To validate the proposed constitutive model, the results obtained through the multilayer perceptron are compared with the ones of the finite element numerical analysis.

Keywords: Machine Learning; Neural Networks; Multilayer Perceptron; Constitutive Models; Nonlinear Analysis; Finite Element Method; Neural Network-Based Constitutive Models.

1 Introduction

In the context of the 4.0 Industry, *Artificial Intelligence* (AI) stands out. AI is a system's ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation [1]. In order to achieve this technological innovation, it is used an algorithm called *Machine Learning* (ML), that allows a system to learn automatically by previous experiences. The most applied technique of ML is known as *Neural Networks* (NN). NN are computational models inspired by our understanding of the biological structure of neurons and the internal operation of human brain [2].

NN has enabled studies in order to represent behavior of complex engineering materials, such as concrete, since it was proved that *Multilayer Feedforward Neural Network*, also known as *Multilayer Perceptron* (MLP), was able to approximate virtually any function of interest to any desired degree of accuracy, and therefore, established as a class of universal approximators [3].

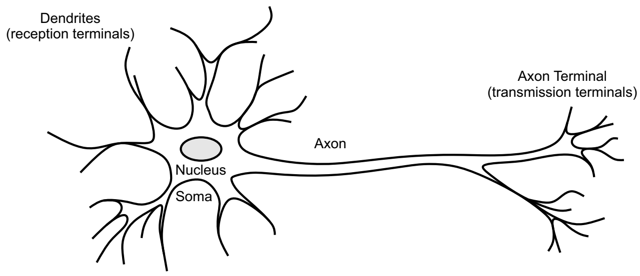
The basic strategy for developing a *Neural Network-based Constitutive Model* (NNCM) is to train a MLP with the *backpropagation learning algorithm* on the *stress-strain* results from experiments and synthetic data from numerical simulations. Then, it could be incorporated into a finite element software as an alternative to the procedural representations of complex material behavior currently used in structure nonlinear analysis [4].

A NNCM draws attention because of its learning and generalization capability and due to its ability to later have its knowledge expanded with new training data, in order to represent unknown materials' behavior.

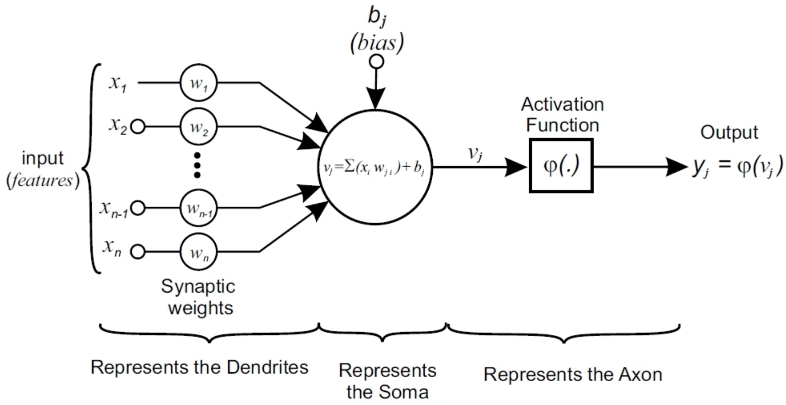
2 Multilayer Perceptrons

A NN is a nonlinear dynamic system formed by a large number of highly interconnected processing units. The processing units are known as neurons, because of their similarities with the human brain connections (see figure 1a). Each neuron receives an input information from the neurons to which it is connected, computes its nonlinear activation level and transmit that activation to other processing units [4]. Figure 1b shows a nonlinear model of a neuron idealized by [5].

The linear combination that happens in a neuron j can be rewritten according to the following equation:



(a) Biologic Neuron



(b) Nonlinear model of a neuron labeled j proposed by [5]

Figure 1. Neuron Model

$$v_j = \sum_{i=1}^{m} w_{ji} x_i + b_j \tag{1}$$

where w_{ji} is the synaptic weight, or strength, and the first subscript refers to the neuron that receives the message and the second subscript to the connection from which the signal came; v_j is the induced local field, or activity, which is the sum of the input signal x_i, applied in the input layer, weighted by its respective strength w_{ji}; and b_j is the bias.

A MLP consist of an input layer, one or more hidden layers, and an output layer. Furthermore, each connection processes a synaptic weight, that stores the knowledge of the computational model and works as a filter between correlated neurons.

The learning rule applied in a MLP is the backpropagation algorithm. This supervised learning process occurs in two steps: the forward pass, where the input signal is sent through the network until its output layer and error signals calculated; and the backward pass, that is responsible for the modification of the synaptic weights based on the calculation of this error signals. This method has as main objective to minimize the synaptic weight vector w in order to find the vector that best correlates the network output values and the target values for these outputs, through an unconstrained optimization technique known as gradient descent.

To facilitate mathematical manipulation for the backpropagation process, it was necessary to perform a small adaptation in the model shown in Figure 1b. This adaptation consists in including in the input an additional value equal to 1 so that the associated synaptic weight plays the role of the bias. Considering that the i, j e k indexes refer to distinct neurons in the network so that the signals propagates in the network from left to right, Equation 1 can be rewritten for a neuron k in the middle of the network, in the n_{th} iteration as

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n) y_j(n) \tag{2}$$

The function signal of neuron k is calculated as

$$y_k(n) = \varphi_k(v_k(n)) \tag{3}$$

where φ_k is neuron's k activation function.

Based on the gradient descent method, and in the derivation of the mean squared error as the loss function with respect to the synaptic weights, the variation in synaptic weights from one iteration to the next can be evaluated as

$$\Delta w_{kj}(n) = \eta \delta_k(n) y_j(n) \tag{4}$$

where the *local gradient* $\delta_k(n)$ can be distinguished in two different cases, depending on whether the network neuron k is in the output layer or in the hidden layers. For the former, it can be defined as

$$\delta_k(n) = e_k(n) \varphi_k^{'}(v_k(n)) \tag{5}$$

and for the later, as

$$\delta_k(n) = \varphi_k^{'}(v_k(n)) \sum \delta_l(n) w_{lk}(n) \tag{6}$$

where l is the neurons that are directly connected with neuron k in the subsequent layer.

The error of a neuron from the output layer of a NN can be established as

$$e_k(n) = d_k(n) - y_k(n) \tag{7}$$

where y_k is the k_{th} component of a predicted vector and d_k is the k_{th} correspondent component of the expected answer vector.

The training of a NN happens through an iterative process which involves the propagation of the training set and the adjustments of the synaptic weights. Each training set is composed by training examples and a complete presentation of the entire training set through the network is defined as an *epoch*. After a NN is sufficient trained, or in other words, has learned the patterns of the desired data, the knowledge is stored in the synaptic weights and used in a testing process, that is needed to validate its generalization capability.

However, deciding the best moment to stop the training is not easy because if the NN is trained for too many epochs, the model will present high accuracy in the training phase but low accuracy in the testing phase, that is, the model will learn patterns from the training data that will not generalize well the test data. This phenomenon is known as *overfitting*. *Underfitting* is the opposite of overfitting and occurs when accuracy's improvement is still possible but it was not achieved.

In order to overcome the issue of overfitting and ensure the network efficiency, it is often used a technique named *k-fold cross-validation*. The basic strategy is to separate randomly the available data set into a training sample and a test sample, and further divide the training sample into two disjoint subsets: an estimation subset, and a validation subset [6]. In this approach, the training test is performed k different times, each time using a subgroup as the validation set, while the others as the training set. At the end of the training, the error is averaged as the mean of errors of each one of the k cycles.

In summary, the MLP with the backpropagation algorithm proceeds as follows: First, the synaptic weights are randomly defined and the feedforward process initiates with the input layer receiving the training data. Second, the k neurons in each layer receives an output from its connections and performs the calculations of the induced local field $v_k(n)$ and after the signal function $y_k(n)$. Then, the output values are sent to others neurons through the outgoing connections and this process continues until the output layer, when the neurons at the output layer computes its errors $e_k(n)$. After that, the backward pass back propagating the error through the network initiates, with the synaptic weights adjustments. Finally, the iterative process continues until a defined convergence criteria is achieved.

3 Nonlinear Finite Element Analysis

A nonlinear finite-element analysis is an incremental-iterative process used to model structures with complex material behavior. The *finite element method* (FEM) discretizes the problem and interpolates some physical quantities from nodal values. After the FEM was established, several more elaborate material models have been studied and developed in order to formulate constitutive models that best apply to specific practical cases. In this context, NNCM presents itself as a great alternative to conventional models because it can represent any type of material in a structure, as long as the network is properly trained with adequate data.

4 Neural Network-based Constitutive Model

A *neural network-based constitutive model* (NNCM) aims to represent the behavior of the material in the structure through a neural network algorithm. The basic strategy for developing a NNCM is to train a MLP on the *stress-strain* dataset from experimental tests or from synthetic data obtained through numerical analysis of established models. Equation 8 represents a general notation used in engineering applications with neural networks.

$$\{output\} = \mathbf{NN}(\{input\} : \{architecture\}) \tag{8}$$

In this context, the NN model utilised to predict stress-strain relationships more adequate for usage in finite element analysis is the *strain-controlled model*, where along the stress-strain state, the strain increment is the input and the stress increment is the output. Moreover, in the case of nonlinear materials, such as concrete, a portion of the stress-strain history is included in the input [2]. Figure 2 illustrates the idea of a NNCM. Equation 9 represents a model of one history point.

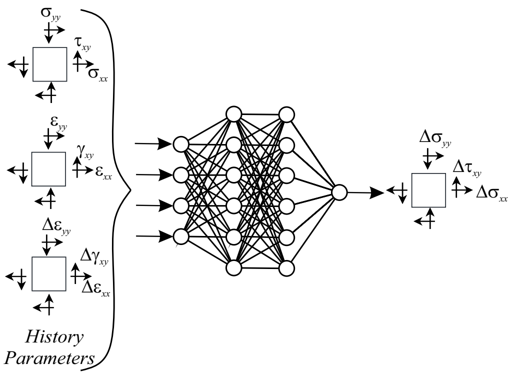


Figure 2. A neural network-based constitutive model

$$\{\Delta\sigma\} = \mathbf{NN}(\{\Delta\varepsilon\}, \{\sigma\}_n, \{\varepsilon\}_n, \{\sigma\}_{n-1}, \{\varepsilon\}_{n-1} : | \; | \; |) \tag{9}$$

5 Application

In order to develop a NNCM, it was used a MLP from the scikit-learn machine learning library for the Python programming language. The proposed network can be represented by Equation 10. The architecture argument shows that it was idealized 15 neurons in the input layer, 3 neurons in the output layer and three hidden layers with 160 neurons each. Besides that, for the backpropagation algorithm, it was used the ReLu [1] as the activation function.

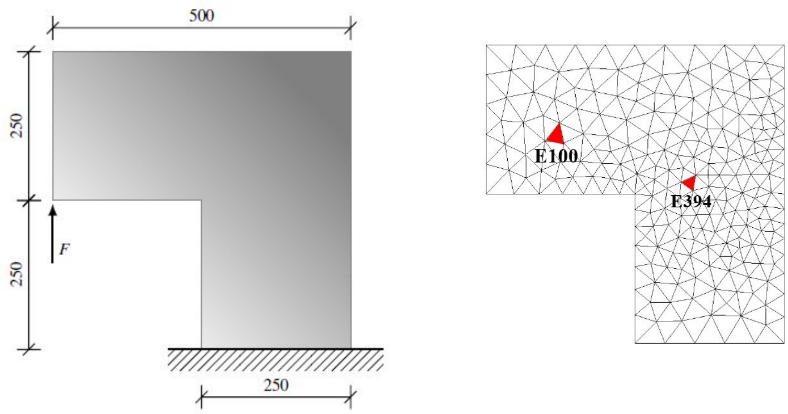
$$\{\Delta\sigma\} = \mathbf{NN}(\{\Delta\varepsilon\}, \{\sigma\}_n, \{\varepsilon\}_n, \{\sigma\}_{n-1}, \{\varepsilon\}_{n-1} : 15 \mid 160 \mid 160 \mid 160 \mid 3) \tag{10}$$

The NN was trained with the k-fold cross validation technique, with k = 5. The training data used was the stress-strain states calculated at the integration points from a nonlinear finite-element analysis on a L-shaped panel with thickness of 100 mm subjected to a vertical force F = 7000 N. Figure 3a illustrates the described problem. Figure 3b shows the finite element discretization. 1159 elements were used for training and only 4 for testing, including the two highlighted.

To solve the system of nonlinear equations was used the Newton-Rhapson process with the direct displacement control method Fuina [7] and the Mazars scalar damage model Mazars [8] . The material parameters adopted in the numerical simulation are: Young's modulus $E = 25850$ MPa and Poisson's ratio $\nu = 0.18$. The Mazars scalar damage model parameters adopted for the exponential damage law are: $\alpha = 0.95$, $\beta = 1100$ and $K_0 = 1.12 \times 10^{-4}$. A vertical displacement increment equal to 5.0×10^{-3} mm was applied at the node of load application, with a tolerance for the convergence in displacement of 1×10^{-4}. Then, the equilibrium path was monitored and further compared with the experiments results from [9].

It can be seen from Figure 4 that the constitutive model used in the numerical simulation presents a good representativeness about the real behavior of the material in the structure. So, it is reasonable to assume that even if the training data are synthetic, obtained from a numerical simulation, it could contain valuable information about the behavior of the material in the context of a structural analysis.

[1] Rectified Linear Unit Function



(a) Geometry (measured in mm) (b) Discretization - FEM

Figure 3. L-shaped panel

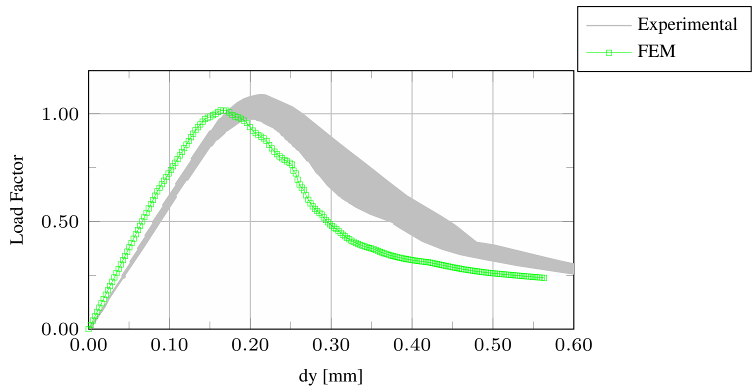


Figure 4. L-shaped panel - Equilibrium path

After the training, the MLP was tested on the stress-strain states at the integration points from the elements highlighted in Figure 3b, for the purpose of checking its generalization capability. Then, in each step, from the current stress-strain state, the previous stress-strain state and strains increment, the MLP predicted stress increments. Figure 5 presents the global stress-strain (σ_{xx}-ε_{xx}) history for the E100 and Figure 6 presents the local stress-strain (σ_1-ε_1) history for E394, according to the orientation of the main strains, both showing the network performance compared with FE results.

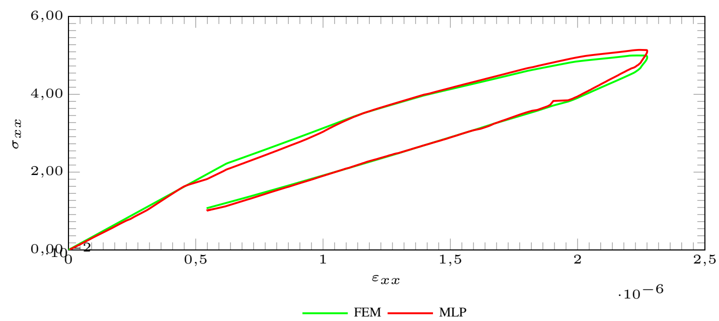


Figure 5. Global stress-strain (σ_{xx}-ε_{xx}) history for E100

It can be observed that there was a satisfactory agreement between the curves obtained through FEM and the curve obtained through the MLP. The network was able to learn precisely the unloading situation of the E100 and the softening behavior of the E394, that are unknown points for the NNCM, that is, not used for the training process.

Then, the same NNCM was used to predict the stress states at the integration points of a three point bending

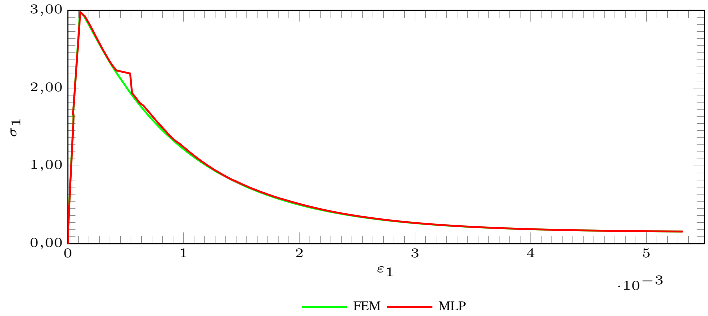


Figure 6. Local stress-strain (σ_1-ε_1) history for E394

test, a different structure never seen before by the network but that was simulated by FEM with the same Mazars constitutive properties. The problem is presented in Figure 7. The beam has a thickness of 50mm and is subjected to a central load of F = 800N. Once again, it was used Newton-Rhapson method, but with the displacement control method on the vertical direction of the load application point with a vertical displacement increment equal to -5.0×10^{-3} m and with a tolerance for the convergence in displacement of 1×10^{-4}.

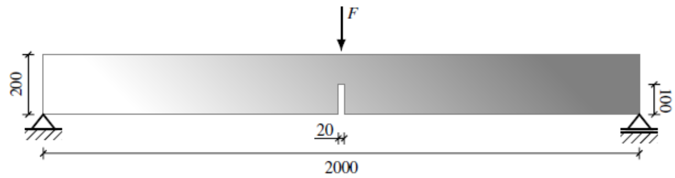


Figure 7. Three-point bending test (measures in mm)

Figure 8 shows the local stress-strain state at the integraiton point of the element E516. This element is located at the tip of the crack and it is under tension. It can be noticed that, compared with the results from the nonlinear finite element analysis, the MLP was able to predict with great accuracy the stress history of the element E516 of the simulated structure.

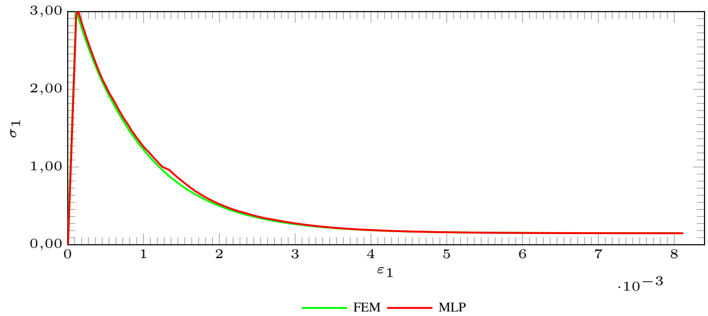


Figure 8. Local stress-strain (σ_1-ε_1) field from E516

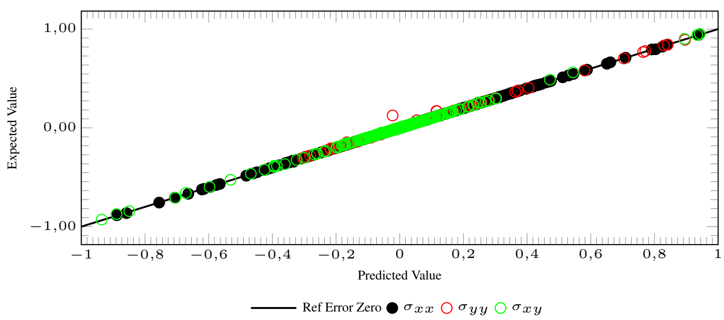


Figure 9. Errors Evaluation

In order to evaluate the predictive capacity of the NNCM in the entire structure, an error graph was plotted (see figure 9). 1000 stress-strain states of the finite element mesh were randomly chosen throughout the nonlinear analysis of the structure, the stresses were predicted by the MLP and compared with the results obtained via FEM. Once again, there is an excellent agreement in the results.

6 Conclusions

This paper presented a NNCM built through the training of a MLP on stress-strain states from a nonlinear finite element analysis of a L-Shaped panel modeled with Mazars constitutive properties. It was observed that the MLP was able to learn Mazars constitutive model and generalize this representation to a three point bending beam with the same material. This approach presents the advantage of not needing the finite element method to calculate the stresses at the Gauss integration points.

Despite the network great performance in learning the Mazars material behavior and generalizing it for other structure, the same network will not be able to predict another material type behavior, unless trained with representative data from this new material. Still, it should be pointed out that the knowledge already acquired by the network can be later expanded with new dataset, due to NN powerful flexibility and adaptivity.

This ongoing study involves yet the understanding of what features should be sufficient to train the network in order to represent the behavior of different material types. Besides that, although conventional constitutive models are well accepted as representative of the behavior of concrete in different practical cases, a point that draws attention is the possibility of training the network based on experimental results of material tests. However, it is known that the amount of experimental information becomes a limiter. To circumvent this problem, it is intended to use a learning technique in order to use load-displacement curves in training inside a nonlinear analysis so that the network is self-learning. Once the network is properly trained, it is intended to incorporate it into a finite element code on the INSANE system with the aim of substitute conventional constitutive models in nonlinear finite element analysis.

Acknowledgements. The authors gratefully acknowledge the support from the Brazilian research agencies CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais; grant PPM-00747-18), and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico; grant 309515/2017-3).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] Kaplan, A. & Haenlein, M., 2019. Siri, siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, vol. 62, pp. 15–25.

[2] Ghaboussi, J., Wu, X., & Kaklauskas, G., 1999. Neural network material modelling. *Statyba*, pp. 250–257.

[3] Hornik, K., Stinchcomebe, M., & White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, vol. 2, pp. 359–366.

[4] Ghaboussi, J., Garrett Jr, J. H., & Wu, X., 1991. Knowledge-based modeling of material behavior with neural networks. *Journal of Engineering Mechanics*, vol. 117.

[5] McCulloch, W. S. & Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133.

[6] Haykin, S., 2009. *Neural networks and learning machines*. Pearson.

[7] Fuina, J. S., 2004. Métodos de controle de deformações para análise não-linear de estruturas. Master's thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.

[8] Mazars, J., 1984. *Application de le Mécanique de l'endommagement au comportement non lineaire et à la rupture du béton de Structure*. PhD thesis, Université Pierre et Marie Curie - Laboratoire de Mécanique et Technologie, Paris, France.

[9] Winkler, B., Hofstetter, G., & Lehar, H., 2004. Application of a constitutive model for concrete to the analysis of a precast segmental tunnel lining. *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 28, pp. 797–819.