XLIII CILAMCE
21-25, November, 2022
FOZ DO IGUAÇU - PR - BRAZIL

BEAM TRANSVERSAL AREA DIMENSIONS OPTIMIZATION USING ARTIFICIAL NEURAL NETWORKS

FERRO, Marco Aurelio Chaves[1], PERLINGEIRO, Mayra Soares Pereira Lima[1], CUNHA, Laís Brilhante da[2]

[1]*Dept. of Civil Engineering, Fluminense Federal University*
Rua Passo da Pátria, 156, São Domingos, ZIP-Code 24210-240, Niterói, RJ, Brazil
marcoferro@id.uff.br, mayraperlingeiro@id.uff.br
[2]*Dept. of Civil Engineering, Fluminense Federal University*
lais_cunha@id.uff.br

Abstract. The present work shows the analysis of a numerical experimental test that was performed using randomized and combined data to study the bending behavior of beams, through the deflection equation considering plane stresses in two different examples. The first is a cantilever beam with force concentrated at the free end and the second is a pinned-pinned beam with loading uniformly distributed along the span. The variables width, height, length, longitudinal modulus of elasticity, deflection, and loading were used as estimated parameters to calculate the ideal width and height dimensions for each beam and obtain a structural optimization considering the limits of deformation according to ABNT NBR 6118/2014. The data generation was generated in Excel spreadsheet format and worked in an Artificial Neural Networks in TensorFlow Python language, with six hidden layers. In addition, the functions 'mae', 'sgd' and 'loss' were used as optimizers or activation function in TensorFlow.

Keywords: Artificial Neural Networks, Structural Engineering, Python Language.

1 Introduction

The concepts of Artificial Intelligence (AI) are present in everyday life in different segments of society and industry through ANNs, Artificial Neural Networks (see Figure 1), mathematical and computational modeling that behaves similarly to human neurons, that is, they communicate with each other through neural synapses in order to achieve a given goal. From this perspective, the network receives data (input), analyzes it through layers (logical commands within the RNA methodology) and provides the results or analyzes previously programmed (output)[1].

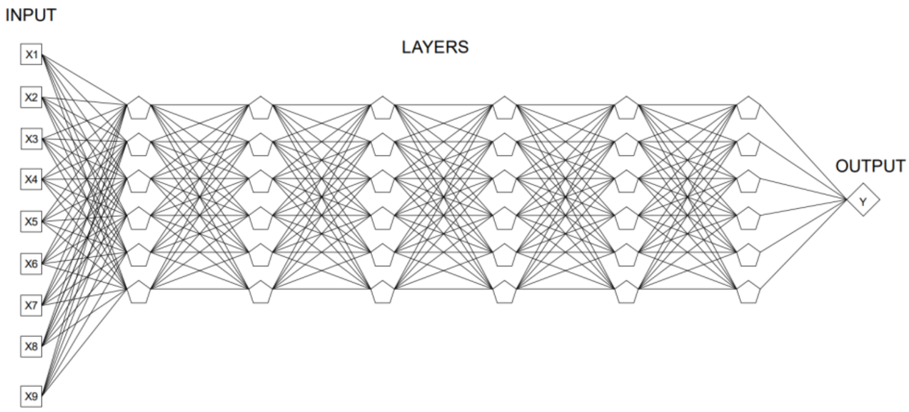

Figure 1. ANNs Illustration

2 Methodology

In this numerical test, the ANNs were used for the flexural analysis in two different cases of beams: the first one is a cantilever beam with a concentrated force at the free end and the second one is a pinned-pinned beam with uniform load along the span using the values of domain 3 of ABNT NBR 6118/2014 that is, the steel and the concrete act in consonance when reaching the flow (Carvalho)[1], as well as analyzing its results taking into account the strength characteristics of the beam and maximizing the use of this structural element so that the Network provides the ideal width and height values for each case. In this experiment, in both cases studied, values were arbitrated in spreadsheet format according to the table below using the 'random' function for each variable between the range 0 to 302 lines in Excel software. In parallel, the elasticity modulus of the concrete (which was calculated according to ABNT NBR 6118/2014 for fck from 20 to 50 MPa), moment of inertia (I in m^4), the Poisson coefficient (υ) and the shear modulus (G in GPa). As for the point load, it was arbitrated as a function of the elongation with the substitutions of the moment of inertia, shear modulus, elasticity modulus, length and height of the beam.

Table1. Range of parameters used in ANN

Range	fck (MPa)	E (GPa)	b (m)	h (m)	L (m)	υ	P (KN)
From	25	100	0.25	0.45	4	0.22	1
Until	50	200	0.35	0.55	5	0.35	5

Modulus of Elasticity as a function of the characteristic strenght of concrete to compression according to ABNT NBR 6118/2014: $E_{ci} = \alpha_E . 5600 . \sqrt{fck}$ for fck from 20 until 50 MPa and α_E = 1,0 for granite and gneiss.

Moment of Inertia:

$$I = \frac{b.h^3}{12} \tag{1}$$

Shear Modulus:

$$G = \frac{E}{2(1+\upsilon)} \tag{2}$$

Maximum deflection for a cantilever beam and a force P at the end:

$$\delta = -\left(\frac{PL^3}{3EI} + \frac{PL\mathrm{h}^2}{8GI}\right) \tag{3}$$

Maximum deflection for a pinned-pinned beam and a uniform load q:

$$\delta = -\left(\frac{\mathbf{5qL^4}}{\mathbf{384EI}} + \frac{\mathbf{3qL^2}}{\mathbf{20GA}}\right) \tag{4}$$

where A is the transversal section area.

Therefore, the variables were inserted into the stretching formula and the concentratred load (P) was a function of these variables. For cantilevered beams (fixed-free) the maximum deflection is $\frac{L}{150}$, according to ABNT NBR 6118/2014. For a supported beam (pinned-pinned) the maximum deflection is $\frac{L}{250}$, according to ABNT NBR 6118/2014. After creating the spreadsheet, it was saved in '.csv' format and imported into the Net to read the data and execute the program created through the Python/Tensorflow. In this experiment, 6 layers were used, followed by weights (W) that act similarly to the weighted average, parameterizing the function (Gad et al.)[2]. In addition, TensorFlow optimizers were used, specifically the 'mae', 'loss', and 'sgd' functions, a library that trains the network and recognizes patterns while the optimizers change patterns when necessary. For example, function 'mae' (Mean Absolute Error) is the tool responsible for calculating the absolute error of the function and mesaure the net performance. The ‘loss’ function is used to find out how much the predicted values deviated from the target value of the training data and to change the weights to mitigate losses. Finally, it is interesting to say that there is a consensus when defining that 'SGD' is a gradient descent optimizing function with the given stimulus. Furthermore, it is interesting to point out that other optimizers were tested, they are the 'mse' (Mean Squared Error), as the name already explains, the quadratic error is caused by the difference between the data obtained and the real data when square and calculates the simple arithmetic mean to obtain the model, and the 'adam' optimizer, which is an ideal approximation function for sparse data. One of the tools (hyperparameters) used in this training stage is the ReLu function, which behaves similar to linear regression, that is, a straight line is created in a positive quadrant, separating the data according to the training given and transforming the negative data into zero. ReLu is called as Activation Function in a Neural Network.

3 Results and Conclusions

Therefore, it can be seen from the graphs obtained in the Spyder IDE in Python Language, after reading the code, that the ANN training was successfully performed and it provided ideal dimension and error results for each analyzed case, as shown below:

3.1 Cantilever Beam

3.1.1 Height (h)

The parameters according to the table below were used as input values in the ANN code according to each case analyzed.

Table2. Entered values for cantilever beam target h

Input	fck (MPa)	b (m)	L (m)	υ	P (KN)	I (m^4)
Ex 1	26	0.29	4.53	0.32	5.90	3.81
Ex 2	47	0.29	4.76	0.24	4.32	4.02

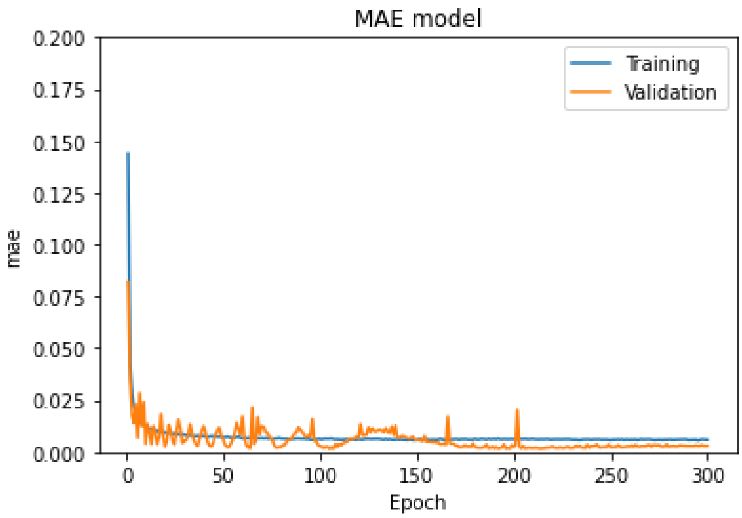

Figure 2. MAE for cantilever beam with target h

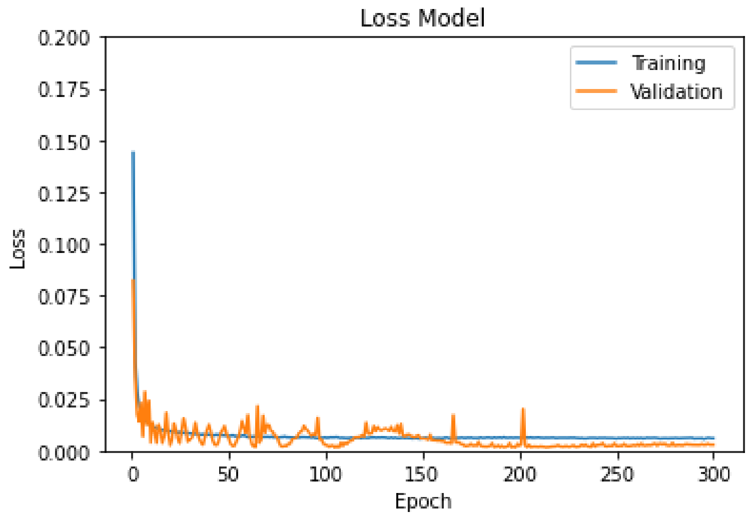

Figure 3. Loss for cantilever beam with target h

3.1.2 Base (b)

The parameters according to the table below were used as input values in the ANN code according to each case analyzed.

Table 3. Entered values for cantilever beam target b

Input	fck (MPa)	h (m)	L (m)	υ	P (KN)	I (m^4)
Ex 1	26	0.54	4.53	0.32	5.90	3.81
Ex 2	47	0.55	4.76	0.24	4.32	4.02

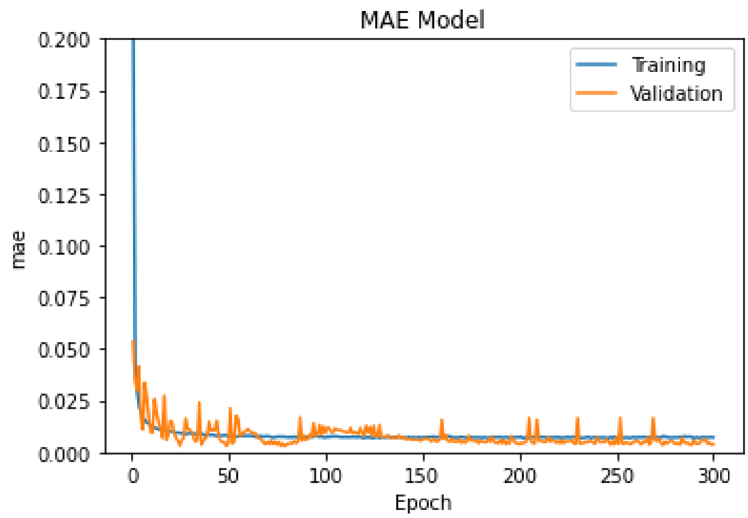

Figure 4. MAE for cantilever beam with target b

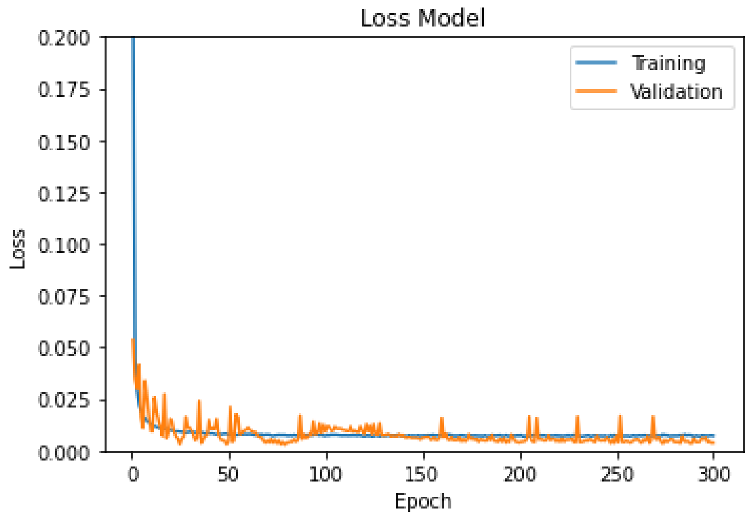

Figure 5. Loss for cantilever beam with target b

3.2 Pinned-Pinned Beam

3.2.1 Height (h)

The parameters according to the table below were used as input values in the ANN code according to each case analyzed.

Table 4. Entered values for pinned-pinned beam target h

Input	fck (MPa)	b (m)	L (m)	υ	P (KN)	I (m^4)
Ex 1	44	0.33	4.69	0.34	3.65	3.59
Ex 2	35	0.34	4.2	0.30	4.22	4.99

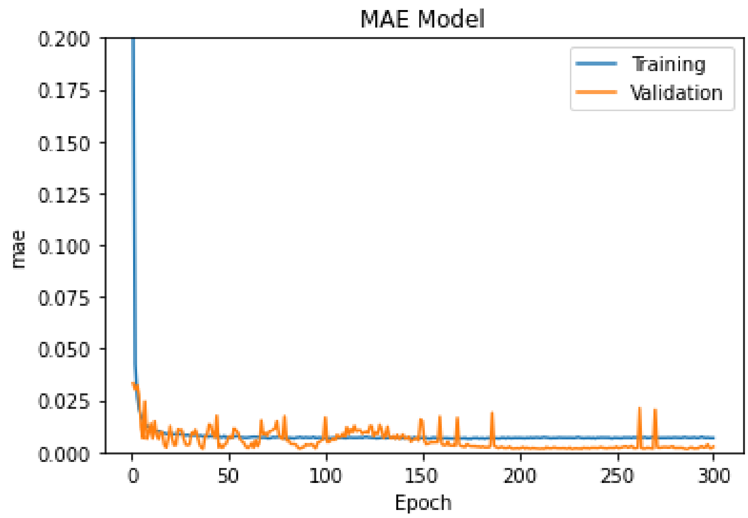

Figure 6. MAE for pinned-pinned beam with target h

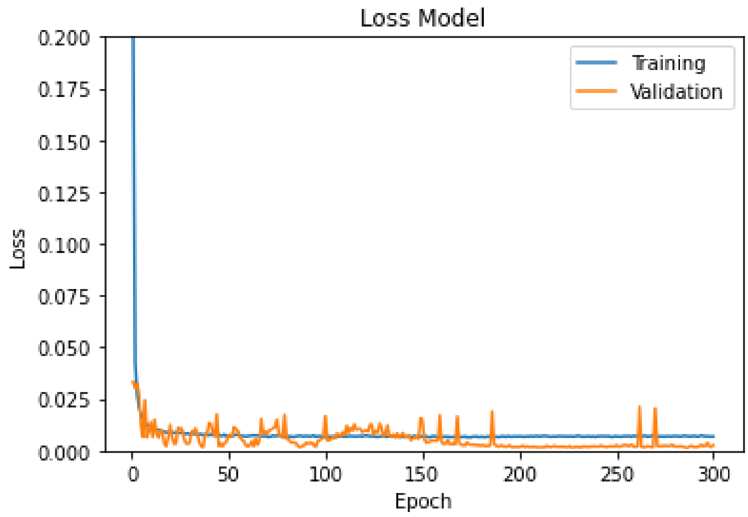

Figure 7. Loss for pinned-pinned beam with target h

3.2.2 Base (b)

The parameters according to the table below were used as input values in the ANN code according to each case analyzed.

Table 5. Entered values for pinned-pinned beam target b

Input	fck (MPa)	h (m)	L (m)	υ	P (KN)	I (m^4)
Ex 1	44	0.51	4.69	0.34	3.65	3.59
Ex 2	35	0.53	4.2	0.30	4.22	4.99

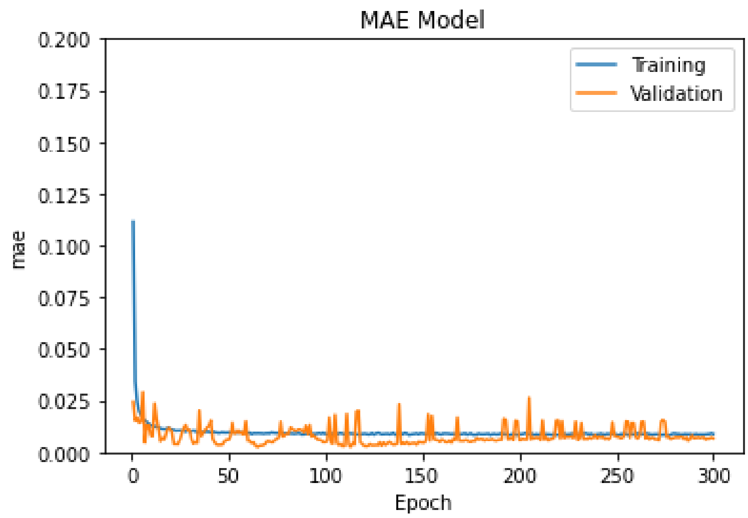

Figure 8. MAE for pinned-pinned beam with target b

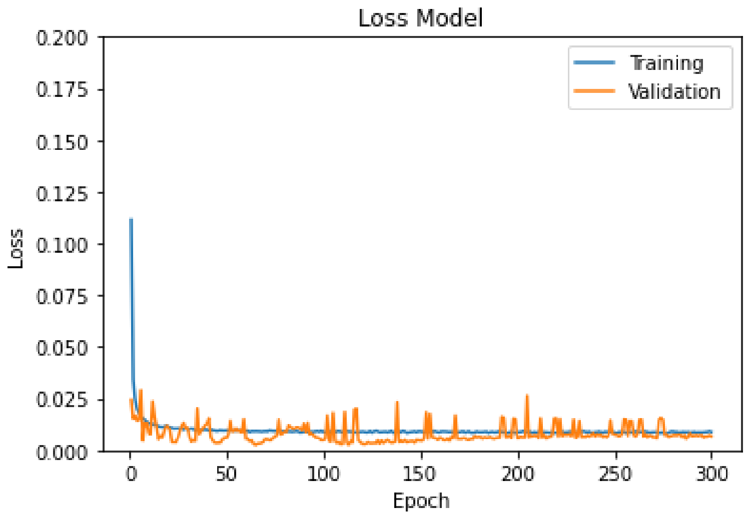

Figure 9. Loss for pinned-pinned beam with target b

Table 6. Results for Expected and Predicted Values and % of Error

Results		*Expected Values*	*Predicted Values*	*Error (in %)*
Cantilever (target h)	Ex 1	0.54	0.5389145	0.20102218
	Ex 2	0.55	0.54380304	2.0155473
Cantilever (target b)	Ex 1	0.29	0.2930587	-1.0547248
	Ex 2	0.29	0.29546383	-1.8840823
Pinned -Pinned (target h)	Ex 1	0.51	0.5102048	-0.04015717
	Ex 2	0.53	0.53460294	-0.86848474
Pinned -Pinned (target b)	Ex 1	0.33	0.3234616	1.9813396
	Ex 2	0.34	0.34002632	-0.00773984

References

__________________. ABNT NBR 6118/2014: design of concrete structures - procedure. Rio de Janeiro, 2014.

[1] Carvalho, Roberto Chust. Calculation and Detailing of Usual Reinforced Concrete Structures acording to NBR 6118:2014 / Roberto Chust Carvalho, Jasson Rodrigues de Figueiredo Filho. – 4.ed. – São Carlos: EDUFSCar, 2017.

[2] Gad, Ahmed Fawzy; Jarmouni, Fatima Ezzahra. Introduction to Deep Learning and Neural Networks with Python. Elsevier, 2021.