COBEM UBERLÂNDIA 2019

ABCM Associação Brasileira de Engenharia e Ciências Mecânicas

25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-0629
Localization of a harmonic excitation load in a clamped-clamped beam using finite element method and artificial neural network

Otávio Duarte Zotelli Boaventura
Daniel Henrique de Sousa Obata
Matheus Silva Proença
Amarildo Tabone Paschoalini
Department of Mechanical Engineering, São Paulo State University (Unesp), School of Engineering, Ilha Solteira, SP
otavio.boaventura@unesp.br, daniel.obata@unesp.br, matheus.proenca@unesp.br, amarildo.tabone@unesp.br

Abstract. *When designing, monitoring or evaluating the health of a structure its often critical to know which operational loads it is subjected to. However, sometimes, these loads cannot be measured directly, so estimating them using the structure response may be the only option. There are 3 categories of methods to do it: deterministic, probabilistic, and artificial intelligence-based. In this work the third kind of methods was used: an Artificial Neural Network (ANN) trained to find the position of a harmonic concentrated force acting on a beam clamped at both ends. Through the Finite Element Method, the beam's response was simulated, features were extracted from a few points, and used to train the ANN. After testing several configurations, the ANN was able to find the excitation position given attributes derived from velocity of any 4 nodes within a certain range, and mean absolute error of 0.2% was reached.*

Keywords: *Load Estimation, Force Reconstruction, Artificial Neural Network, Structure Health Monitoring.*

1. INTRODUCTION

Operational Load Estimation, or only Load Estimation for short, is an important issue often found in science and engineering when there is a need to monitor and assess damage progression on a structure. The problem consists of estimating amplitudes, frequencies, location or even probabilistic parameters of an excitation force based on the structure's response so one could refer to it as Force Reconstruction. It rises when the direct measurement of these external forces is impossible due to the unknown nature of it, accessibility problems or any other limitation concerning the force gauge, e.g. wind, seismic, traffic, explosion, and shock excitations (Klinkov and Fritzen, 2007).

Force reconstruction or loads estimation methods can be divided into 1. Deterministic methods: frequency or time-domain methods, 2. Stochastic methods: statistical models and 3. Artificial Intelligence (AI) methods (Uhl, 2007).

The first ones are based on physical models of the system and are often called inverse problems. The reason is, when estimating the loads, one is trying to find the inputs of the system (loads) given the outputs (behavior) and the model parameters. The results depend heavily on the accuracy of the inverse model identification and estimation of model parameters. Besides that, strongly nonlinear systems modeling is relatively inaccurate and difficult (Uhl, 2007). Besides that, this inverse problem may be ill-posed: it could have many possible solutions, a globally defined solution for all reasonable data doesn't exist, or the solution depends continuously on the given data. Regularization methods could be used to overcome these issues, but they also have their own limitations (Qiao *et al.*, 2019).

The second type of methods pursues statistical relations between the output and the input, which demands to measure them during operation of the structure. Although, as said, this cannot be performed during operation in some cases because of the problems with direct input measurements. A simple example is a regression model (Trujilloan and Busby, 1997). Other more complex methods such as Bayesian Inference (Zhang *et al.*, 2012) are shown by Sanchez and Benaroya (2014) to achieve better results than deterministic methods.

Lastly, AI methods include Artificial Neural Networks (ANN), Fuzzy algorithms, Evolutionary algorithms, and others. They require a training phase, in which measurements of the structure behavior are fed to the algorithm, thus it could establish appropriate relationships between inputs – the loads – and outputs – the structure's response. Those measurements can be obtained from real, physical system response, or be calculated via numerical simulations (Uhl, 2007).

Although the statistical methods show good results, AI is a hot topic today, presenting itself as a universal function approximation, even for strongly non-linear cases, without the need of any physical modeling (Cao *et al.*, 1998). An extensive list of successful applications of ANN's in engineering and structural reliability problems with detailed examples can be seen in Chojaczyk *et al.* (2015), indicating how versatile this method is. Breakthroughs in this field are coming up one after the other, and such discoveries have a great potential to improve Load Estimation. Nevertheless, this huge potential was explored in only a few works, mostly for aeronautical applications where a cantilever beam is used as a physical model or a more complex structure with a similar boundary condition (Cooper and DiMaio, 2018; Cao *et al.*, 1998).

A couple of other works for different applications and models may also be found, besides the already mentioned ones, but none using ANN's. For example, Chen *et al.* (2018) uses a time-domain approach combined with a Bayesian Regularization, and a simply supported beam to model a bridge, while (Feng *et al.*, 2015) used Bayesian Inference in a similar problem. Uhl (2007) used a spring-mass-damper to model a wheel-rail system and estimates the contact load employing Markov parameters. While Sun *et al.* (2015) modeled a truss with beam elements through a Finite Element Method (FEM).

Hence, the goal of this paper is not to focus on any specific engineering application, but rather use a simplified model to explore a different boundary condition. Therefore the problem of load estimation is explored using ANN for a dynamic load case: a beam clamped on both ends subjected to a concentrated harmonic force. In this work, a Python neural network library called Keras was adopted for the design and testing of a densely connected ANN (Chollet, 2015). The clamped-clamped beam is a suitable model for other range of applications such as micro mechanical transducers, diaphragms and civil structures (Bao, 2005).

2. METHODOLOGY

In its essence, ANN's were inspired by how biological neurons work. To approximate a function, the ANN's training process requires finding the parametric function (weights) of the links that connect the neurons. The simplest training method is the supervised kind. It starts by feeding the network with a set of training examples, and successively updating the weights iteratively using a Back Propagation (BP) learning algorithm. A process that resembles a human learning from mistakes and successes.

Before beginning the training process, it is required to choose an architecture (hyper-parameters), which consists in defining the number of layers, the number of neurons of each layer, and the initial values of weights, activation functions, and the BP algorithm. Figure 1 shows a diagram of a generic ANN. After receiving the input data from the left, each neuron in the first layers calculates a weighted sum of the inputs and adds a bias. The result is named 'activation state'. Then an 'activation function' is applied and the results are given as inputs to the next layer (hidden layers). Those layers repeat the process until the last layer is reached. After that, the error between the actual output and the network output, defined by a loss function, is calculated and must be reduced throughout the training. This is achieved by using a optimization method, a BP algorithm, based on the loss function's derivatives with respect to the weights of each layer. One can test several architectures, optimization algorithms, and loss functions to achieve the best results for a given training data set and also iterate over the same dataset. Each iteration is called an epoch (Cooper and DiMaio, 2018).

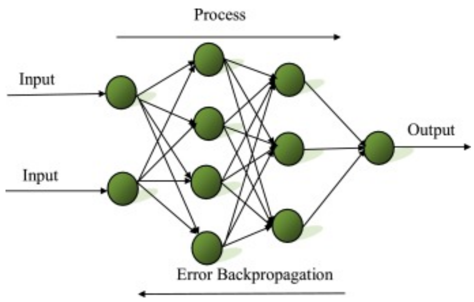

Figure 1. Densely connected ANN (Cooper and DiMaio, 2018).

To have high precision, an ANN must be trained with a reasonable amount of data. So, instead of performing a big number of experiments and to focus the efforts on the ANN, simulated data were used to train and test the model. A FEM software was used to generate the data needed, it discretizes a steel beam, with a rectangular cross-section, using elements with two degrees of freedom. Then it calculates displacements and velocities over time, the latter employing the Newmark integration method (Nishioka, 2007), for a harmonic force acting on one node.

At first, to generate one data example, a random node is chosen as the point where a harmonic force is applied to. Then the Root Mean Squared (RMS) value of velocity is calculated for each node time series, ignoring the transient part. Velocity was chosen as the measured response due to higher variations over several conditions, avoiding ill-posing the problem. After that, secondary features are calculated using the RMS values, which are: maximum and minimum values, the first four statistical moments – mean, standard deviation, kurtosis and skewness –, and the angular coefficient of a linear interpolation of the RMS values, sorted as the points of the position on the bar. The last feature is explained in Fig. 3. Different combination of these features was given as inputs to the network in order to test which one would result in the lowest error. The ANN output is always a prediction of which node the excitation was applied. This procedure to generate a data example is summarized in Fig. 2

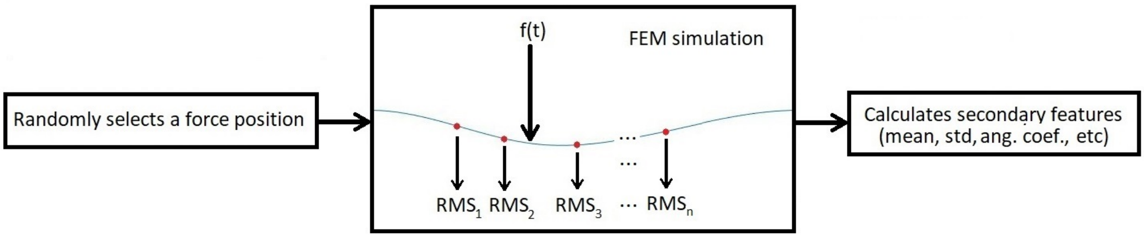

Figure 2. Data generation workflow.

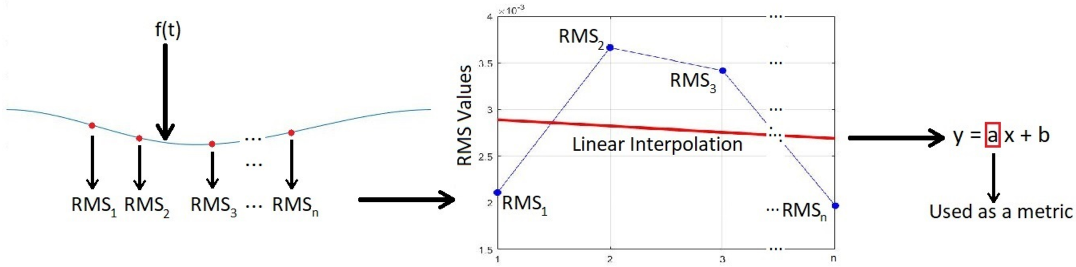

Figure 3. Angular coefficient feature.

The hyper-parameters of the ANN were explored using a Grid Search method (Bellman, 1961) searching for the configuration that would result in the lower Mean Absolute Error (MAE) for the force position. The training was performed on a laptop with an Intel Core i7, 16GB of RAM and an Nvidia Geforce MX150 graphics card. Each epoch took about three minutes.

3. RESULTS AND DISCUSSION

The beam's dimensions adopted were: 3mm thick, 19mm of width, and 50cm long, discretized as 50 elements, having 1cm of length each, resulting in 51 nodes. Training data were extracted at 11, 19, 27 and 35cm, see Fig. 4. The force position varied from 10 to 40cm, selecting randomly within this range for each training example, 10000 in total were generated.

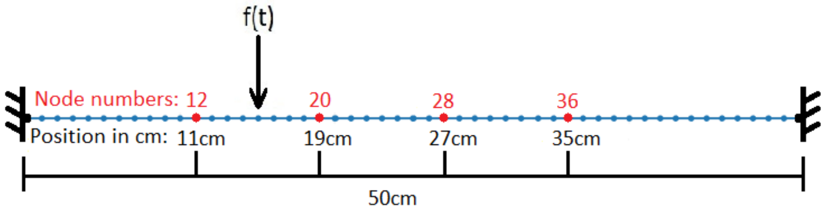

Figure 4. Discretized beam and nodes from which the features were extracted.

The ANN's configurations tested were: one to five hidden layers, 16 to 128 neuron number on each hidden layer, two types of loss function – Mean Squared Error (MSE) and MAE –, Adagrad (Duchi *et al.*, 2011) and Adam (Kingma and Ba, 2015) as optimization algorithms, and lastly, two activation functions were tested: Softmax and Rectified Linear Unit (ReLU) (Agarap, 2019). The number of iterations over the dataset was enough to make the algorithm to converge in all configurations.

After training, more data were generated for testing. Results can be seen in Fig. 5a), where the x coordinate is the true position values in terms of node number, as the y coordinates are the ANN's predictions. The dashed red line is a reference at which the predicted value would be equal to the real one. In Fig. 5b) we can see the error reduction over iterations. For these results, the hyper-parameters were 4 hidden layers with 64 neurons each, MAE as loss function, Adam was used as optimizer, ReLU as activation function. The inputs were the RMS values only and iterating until the stabilization of MSE. Figure 6 shows the results when Softmax was used as activation function, which also required changing the inputs to the four statistical moments, maximum and minimum RMS values.

The two configurations used to obtain the results from Figs. 5 and 6 converged to models that are capable of predicting the location of the input force with an MAE below 0.2cm. Nonetheless, the second configuration has a bigger error when the force is positioned after 37 cm. This configuration also used features derived from de RMS values taking longer to generate the data.

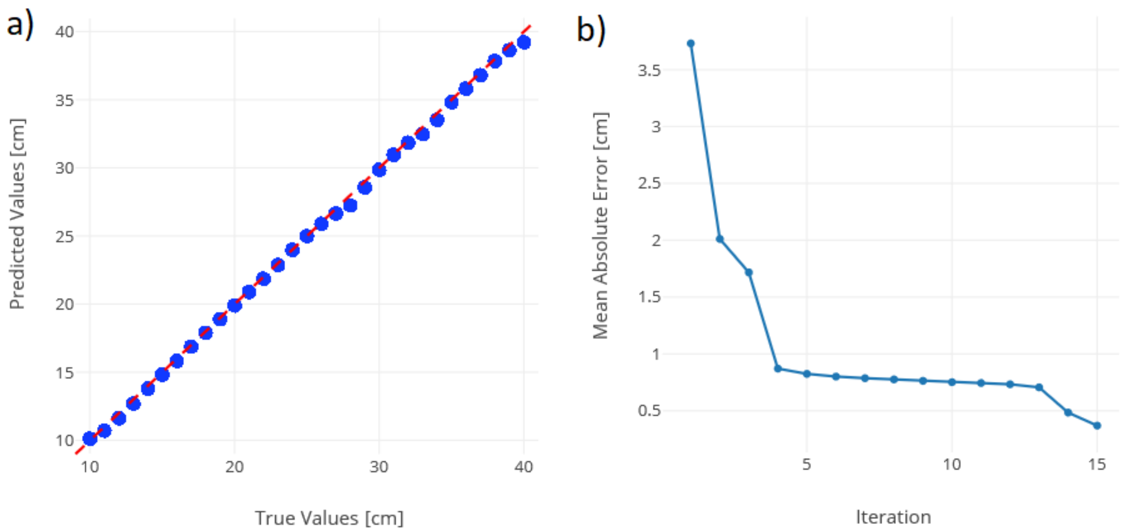

Figure 5. ReLU as activation function a) Predictions x Training outputs and b) Convergence.

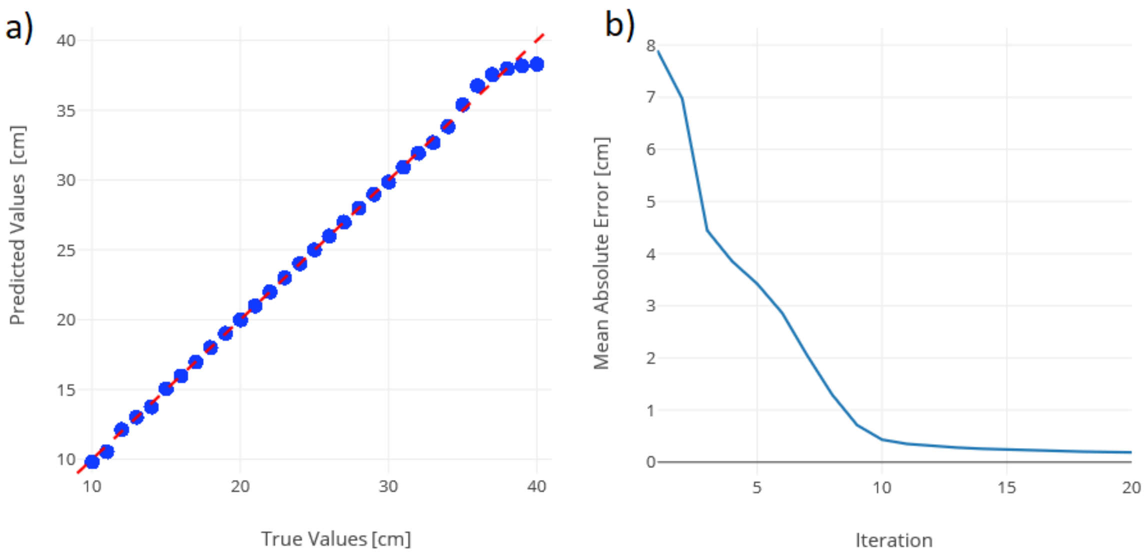

Figure 6. Softmax as activation function a) Predictions x Training outputs and b) Convergence.

For the same hyper-parameters, data from different sets of four nodes also converged to similar error levels, except when the distance between nodes used to collect input data was less than 5 nodes or 5cm, then the error was a lot larger. This may happen because the input values become too similar. Of course, when a different set of nodes were chosen, the ANN required retraining.

So in order exclude the requirement of always using the same set of nodes to derive the input data, i.e., modify the ANN so it could accept data from any random combination of nodes, a bigger data set was generated. A step was added to the workflow shown in Fig. 2: random selection of nodes to extract data from, but restricting them to be at least five nodes apart from each other. Then, using the first ANN architecture as a starting point, the ANN was trained using this new data set.

Besides that, an unusual feature is used in this case, explained in Fig. 3, to aid the ANN find out on which side of the beam the force was acting on, avoiding ill-posedness. When the excitation force is applied closer to the beam's left side, the angular coefficient is negative, if the force is acting on the right side, positive. Without it, the ANN often mispredicted the force position symmetrically. For example, if the force was applied on 30cm, 20cm from the closest clamp, the ANN would predict that the force was applied on 20cm, also 20cm from the closest clamp. One last adjustment was also needed: the normalization of the dataset, dividing each feature by its overall maximum value.

The configuration that resulted in the lowest error, approximately 0.1cm MAE, was the same used to obtain the first results shown in this work, only the inputs were changed this time: all the RMS values and the angular coefficient. Data were extracted from the same amount of nodes. Figure 7 shows the results.

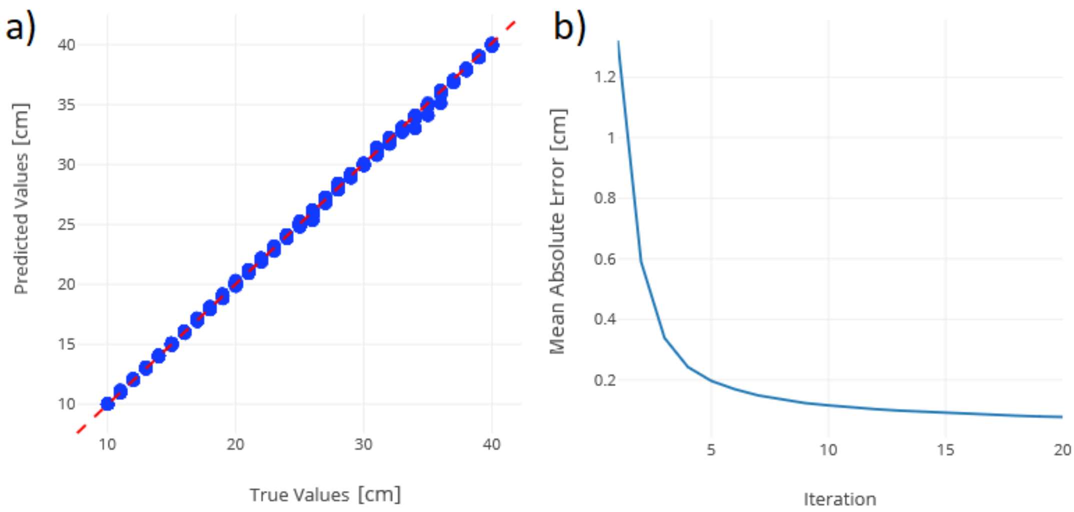

Figure 7. Results for a random combination of nodes used to extract data a) Predictions x Training outputs and b) Convergence.

It is important to point out that although the error is lower than the first two results, we can see in Fig. 7 that some points around 35cm deviate more from the red dashed line, that's because the used error metric is a mean of the absolute error between the predicted and the actual force position. So, although some predictions have a higher inaccuracy, since the majority present lower error, the MAE is lower.

4. CONCLUSION

An ANN was designed and trained to find the position where a harmonic exciting concentrated force was applied on a 50cm clamped-clamped beam. The training data were features derived from the RMS velocity of the structure response at few points, generated from a Finite Element Model. At first, we were able to find two configurations where the network inputs were derived from four fixed point of the beam. Then, one more generalizing network was developed and trained, using a larger and normalized dataset, and an unusual feature. The final result was an ANN able to find the position of the force given features of the structure response in any random four points, having a minimum distance between each other, and within the range of 10 to 40cm, the same range used to vary the force position. An MAE of 0.1cm or a 0.2% error was reached.

5. ACKNOWLEDGEMENTS

The authors thanks FEPISA for its financial support.

6. REFERENCES

Agarap, A.F.M., 2019. "Deep learning using rectified linear units (relu)". ArXiv. 17 Mar 2019 <https://arxiv.org/pdf/1803.08375.pdf>.

Bao, M., 2005. *Analysis and Design Principles of MEMS Devices: Chapter 2 - Mechanics of Beam and Diaphragm Structures*. Elsevier, 1st edition.

Bellman, R., 1961. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, New Jersey, 1st edition.

Cao, X., Sugiyama, Y. and Mitsui, Y., 1998. "Application of artificial neural networks to load identification". *Computers & Structures*, Vol. 69, pp. 63–78.

Chen, Z., Chan, T.H.T. and Nguyen, A., 2018. "Moving force identification based on modified preconditioned conjugate gradient method". *Journal of Sound and Vibration*, Vol. 423, pp. 100–117.

Chojaczyk, A.A., Teixeira, A.P., Neves, L.C., Cardoso, J.B. and Soares, C.G., 2015. "Review and application of artificial neural networks models in reliability analysis of steel structures". *Structural Safety*, Vol. 52, pp. 78–89.

Chollet, F., 2015. "Keras". GitHub repository. 19 Mar 2019 <https://github.com/fchollet/keras>.

Cooper, S.B. and DiMaio, D., 2018. "Static load estimation using artificial neural network: application on a wingrib".

Advances in Engineering Software, Vol. 125, pp. 113–125.

Duchi, J., Hazan, E. and Singer, Y., 2011. "Adaptive subgradient methods for online learning and stochastic optimization". *Journal of Machine Learning Research*.

Feng, D., Sun, H. and Feng, M.Q., 2015. "Simultaneous identification of bridge structural parameters and vehicle loads". *Computers & Structures*, Vol. 157, pp. 76–88.

Kingma, D.P. and Ba, J.L., 2015. "Adam: a method for stochastic optimization". *Published as a conference paper at ICLR*.

Klinkov, M. and Fritzen, C.P., 2007. "An updated comparison of the force reconstruction methods". *Key Engineering Materials*, Vol. 347, No. 461-466.

Nishioka, T., 2007. *Comprehensive Structural Integrity - Computational Aspects of Dynamic Fracture*, Vol. 3. Elsevier Science, 1st edition.

Qiao, B., Liu, J., Liu, J., Yang, Z. and Chen, X., 2019. "An enhanced sparse regularization method for impact force identification". *Mechanical Systems and Signal Processing*, Vol. 126, pp. 341–367.

Sanchez, J. and Benaroya, H., 2014. "Review of force reconstruction techniques". *Journal of Sound and Vibration*, Vol. Volume 333, No. Issue 14, pp. 2999–3018.

Sun, J.L.X., Han, X., Jiang, C. and DejieYu, 2015. "Dynamic load identification for stochastic structures based on gegenbauer polynomial approximation and regularization method". *Mechanical Systems and Signal Processing*, Vol. 56–57, pp. 35–54.

Trujilloan, D.M. and Busby, H.R., 1997. *Practical Inverse Analysis in Engineering*. CRC Press London, 1st edition.

Uhl, T., 2007. "The inverse identification problem and its technical application". *Arch Appl Mech*, Vol. 77, pp. 325–337.

Zhang, E., Antoni, J. and Feissel, P., 2012. "Bayesian force reconstruction with an uncertain model". *Journal of Sound and Vibration*, Vol. Volume 331, pp. 798–814.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.