Artificial Neural Network Application for Structural Damage Diagnosis from Vibration Measurements

Calebe Paiva Gomes de Souza[1], **Paulo Roberto Gardel Kurka**[2], **Romulo Gonçalves Lins**[3] **and José Medeiros de Araújo Júnior**[4]

[1] Departamento de Estruturas/CT/UFPI, CEP: 64.049-550, Teresina-PI, Brasil, calebepaiva@ufpi.edu.br

[2] Departamento de Projeto Mecânico/FEM/UNICAMP, CEP: 13.083-970, Campinas-SP, Brasil, kurka@fem.unicamp.br

[3] Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas/UFABC, CEP: 09210580, Santo André-SP, Brasil, romulo.lins@ufabc.edu.br

[4] Engenharia Elétrica/CT/UFPI, CEP: 64.049-550, Teresina-PI, Brasil, jmjunior@ufpi.edu.br

Abstract: The structural damage diagnosis aims to verify, in the first stage, if damages exist in the structure and if they can compromise its operation. During the second stage, it is mandatory to localize the damage and determine its severity level. For developing an efficient diagnosis, one possibility is to use vibration measurements, because a damage modifies the modal parameters of structures. However, the efficient damage detection still remains a challenge, and then, modern techniques have been proposed in order to improve the initial diagnosis, such as the use of Artificial Neural Network (ANN). In this paper, a ANN-type Multi-Layer Perceptron (MLP) will be used, being that its architecture will be formed in three layers (input layer, hidden layer and output layer). Moreover, the samples to feed the MLP will be obtained from a numerical modeling of a simply supported beam by using Finite Element Method, which will supply dynamic response of the structure in two main cases: undamaged structure and with various damage scenarios. From the vibration response, some samples will be used for training the ANN by using backpropagation algorithm and Gradient Descent Method in a supervised learning, once for each damage will be known input data (modal parameters) and the expected results of damage location and its severity. Many ANN topologies are analyzed, varying both the number of hidden neurons and the activation function. The best topology will be obtained from the analysis of statistical parameters. In addition to the "tanh" activation function, the "ISRU" function, which has never been used to diagnosis of structures based on vibration measurements, will also be verified. After training procedure, the validation set will be inputted into the trained ANN in order to prove that it can detect, localize and quantify damages arbitrarily with accuracy and reliability.

Keywords: Vibration-based damage diagnosis, Pattern recognition and classification, Artificial Neural Network, Tangent Hyperbolic function, Inverse Square Root Unit function

INTRODUCTION

When the damage detection in large and complex structures is demanded, the most recommended method is that which detects global damages, from the analysis entire of structure. Among the possible methodologies, the detection of structural damage by vibration measurements has been widely used, since the basic premise is that the modal parameters (natural frequencies, modes of vibration, etc.) are directly connected to the physical properties (stiffness, mass and damping matrices) of the structure; that is, any damage occurring in the structure will change the physical parameters and consequently change the modal parameters (Doebling, Farrar and Prime, 1998).

Among the main modern techniques for detecting damage in structures, from the vibration measurements, Artificial Neural Networks (*ANNs*) have been highlighted by its capacity for identifying severe damages and also for its ability for pattern recognition (Tan et al., 2017), as well as for its power and high accuracy in classifying patterns (Lautour and Omenzetter, 2010). A special and well established type of *ANN* called of Multi-Layer Perceptron (MLP) have been commonly applied in damage identification of civil structures.

Mehrjoo et al. (2008) and Bakhary, Hao, and Deeks (2010) have explored damage detection algorithm by substructure evaluation, from the use of modal parameters as input into *ANNs* with single neural network and using standard back-propagation method (BP) added of momentum rate and learning coefficient rate. In addition, it used sigmoid activation function and the number of hidden layers and their hidden neurons was determined by trial and error method. Hakim, Razak, and Ravanfar (2015) presented technique developed for damage localization and severity classification in I-beam, being that the *ANN* training occurred from the clustering of specialist neural networks that are experts in each mode of vibration and traditional *BP*. The number of hidden layers and of hidden neurons was defined by absolute error. Tan et al. (2017) also made use of vibration measurements to structural damage diagnosis in I-beam from the modal strain energy based damage index (β) and use the Levenberg-Marquardt algorithm to train the neural network. Only one hidden layer was used and the number of hidden neurons was defined by the R coefficient and the Mean Square Error (MSE).

The *ANN* topology is defined, basically, by the amount of neurons that make up the layers and the activation functions applied. In order to determine the number of neurons in the hidden layer, several researches have presented different heuristics for this purpose (Arai, 1993), (Huang, 2003), (Tamura and Tateishi, 1997), (Jinchuan and Xinzhe, 2008), (Shibata and Ikeda, 2009), (Hunter et al., 2012) and (Gnana Sheela and Deepa, 2013). Although these methodologies are proposed, what can be said is that there is no consolidated and general definition of the number of hidden neurons to adopt in general cases. The nature of data sets have great relevance in such a definition. Thus, the trial and error method has been the most effective, from the choice of the appropriate stopping criterion, taking into consideration that small number of hidden neurons can lead to underfitting and many neurons can lead to overfitting (Panchal et al, 2011).

Regarding the activation functions, in the vast majority of searches with *ANNs* and damage identification in structures the most used functions are Hyperbolic Tangent (tanh) and Sigmoid (sig) functions. Carlile et al. (2017) said that among these functions the one that has the best performance is the *tanh*. In additon, he also presented a new function called Inverse Square Root Unit (ISRU) which can be more efficient than the *tanh* in about 3x to 6x. In several research related to the diagnosis of civil structures based on vibration measurements, no records of the application of this promising function were found.

In this work, non-damaged and damaged structures will be numerically modeled with Finite Element Method (FEM) that will provide the natural frequencies and mode shapes. Such results will be allocated to the input layer of the *ANN* and the expected results in terms of localization, and classification of the severity of the damage will be obtained in the output layer. In order to do so, the network architecture will be constituted, besides the input and output layers, by one hidden layer. Several topologies of the network will be studied, from the diversification of the number of neurons, in order to determine the optimum amount that implied better results. Also, the behavior of the *tahn* and *ISRU* functions will be analyzed in a critical way during the training of the *ANNs* that will occur by using *BP* and Gradient Descent Method (GDM).

MODAL ANALYSIS

In order to determine the dynamic properties of a structure by using *FEM* in non-damaged and damaged structures, under undamped free vibration, some algorithms have been implemented in Matlab®. In this case, the structures are under natural motion, as defined by Rayleigh (1877) *apud* Adhikari (2000), and the motion equation is determined by Eq. (1).

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{0}. \tag{1}$$

where: $\mathbf{x}(t)$ is the displacement vector whose the order is $n \times 1$; $\ddot{\mathbf{x}}(t)$ is the acceleration vector whose the order is $n \times 1$; $\mathbf{M}$ is a matrice, whose its order is n, related to the structural mass; $\mathbf{K}$ is a matrice, whose its order is n, related to structural rigidity.

The oscillation frequency and the displacement form are called as natural frequencies and mode shapes, respectively, being obtained from the motion expression, given by Eq. (2):

$$\left(\mathbf{M}^{-1}\mathbf{K} - \lambda \mathbf{I}\right)\phi = 0 \tag{2}$$

Disregarding the trivial solution, $\phi = \mathbf{0}$, can be perceived that the determinant of Eq. (2) generates polynomial function of order n, whose the solution implies in to find the follow ordenated roots (eigenvalue): $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. Furthermore, for each eigenvalue λ_i, it is associated an eigenvector $\phi_{\mathbf{i}}$. These two aforementioned entities are entitled as natural frequency and mode shapes of the system, respectively, and they will be the inputs of the proposed *ANN*, as discussed in the next Section.

ARTIFICIAL NEURAL NETWORK

In order to develop mechanisms that can help in the detection, localization and classification of damages in civil structures in service life, many mathematical and computational tools have been proposed for this purpose. Among such tools, the *ANNs* have obtained a prominent position among the several researches presented.

Multi-Layer Perceptron training process

ANNs are computational models created with the prerogative to simulate the functioning of biological neurons, such that their structures are equivalent. The knowledge and learning acquisition are important characteristics that *ANNs* must possess. In addition, Silva, Spatti and Flauzino (2016) and Haykin (2001) mention other relevant characteristics for *ANNs* are: adaptation by experience, generalization ability, organization of data and fault tolerance.

The *MLP* will be used and one of its main characteristics being the high distribution of data over the network. The high branching allows the samples to influence all the neurons of the hidden layer and the output layer. Thus, the process of distribution of the *MLP* data is presented in Fig. 1 as follows:

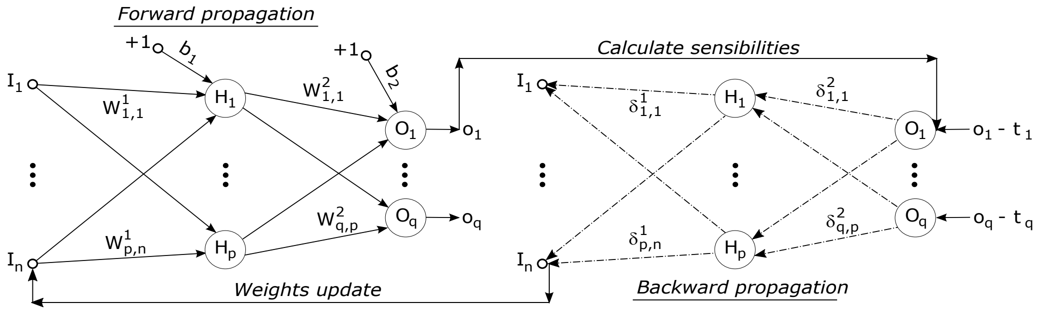


Figure 1: Multi-Layer Perceptron training process.

Mathematically, from the data collected and presented at the input layer, such data is weighted by the $W^{(1)}$ and conveniently summed up, resulting in the input, H_j^{in}, of each hidden layer neuron, according to Eq. (3). From this, the activation function f is applied, obtaining H_j^{out}, Eq. (4).

$$H_j^{in} = \sum_{i=1}^{n} W_{j,i}^{(1)} I_i + b_1 \therefore i = 1..n, j = 1..p \tag{3}$$

$$H_j^{out} = f\left(H_j^{in}\right) \tag{4}$$

where: I_i is the neuron i of the input vector; b_1 is the bias; H_j^{out} is the output of the neuron j of the hidden layer; $W_{j,i}^{(1)}$ is the matrix with weights between the hidden layer and the input layer; n is the amount of neurons in the input layer; p is the amount of neurons in the hidden layer.

In the next step, the outputs of the neurons of the hidden layer will be weighted by the $W^{(2)}$ and they will enter in each neuron of output layer by similar to previous way, Eq. (5) e Eq. (6).

$$O_k^{in} = \sum_{j=1}^{p} W_{k,j}^{(2)} H_j^{out} + b_2 \therefore j = 1..p, k = 1..q \tag{5}$$

$$O_k^{out} = f\left(O_k^{in}\right) \tag{6}$$

where: O_k^{out} is the output of the neuron k of the output layer; b_2 is the bias; $W_{k,j}^{(2)}$ is the matrix with the weights between the output layers and the hidden layer; q is the amount of neurons in the output layer. With the results obtained in the neurons of the output layer, it is possible to start the *BP* algorithm from the errors, in order to update the weights and bias. To do this, the Sum Square Error *(SSE)*, Eq. (7), is used to measure the difference between the output neuron results and the target values, T, for each sample s.

$$E = \sum_{k=1}^{q} E_k(s) = \sum_{k=1}^{q} \left(T_k(s) - O_k^{out}(s)\right)^2 \tag{7}$$

Obviously, when considering a training set consisting of s samples, the evaluation of the overall performance of this algorithm can be measured by Mean Square Error (MSE), labeled as E_m. From the calculation of E_m, Eq. (8), the error propagation occurs in order to update the synaptic weights and bias, Eq. (9) until Eq. (12). This procedure is performed by the mathematical technique *GDM*.

$$E_m = \frac{1}{s} E \tag{8}$$

$$W_{k,j}^{(2)} = W_{k,j}^{(2)} - \eta \cdot \delta^{(2)} \cdot H_j^{out} \tag{9}$$

$$b^{(2)} = b^{(2)} - \eta \cdot \delta^{(2)} \tag{10}$$

$$W_{j,i}^{(1)} = W_{j,i}^{(1)} - \eta \cdot \delta^{(1)} \cdot I_i \tag{11}$$

$$b^{(1)} = b^{(1)} - \eta \cdot \delta^{(1)} \tag{12}$$

where: η is the learning rate; $\delta^{(2)}$ e $\delta^{(1)}$ are the sensibilities defined in Eq. (13) and Eq. (14). Thus, for the network architecture in question, the sensibilities for activation functions are determined as follows:

$$\delta^{(2)} = -2 \cdot diag[f'(O_k^{out}(s))] \cdot (T_k(s) - O_k^{out}(s)) \quad (13)$$

$$\delta^{(1)} = diag[f'(H_j^{out}(s))] \cdot W_{k,j}^{(2)} \cdot \delta^{(2)} \quad (14)$$

where: $diag[\cdot]$ represents the diagonal matrix where the diagonal components are the first derivatives of the indicated functions.

As the weights and bias have been updated, the inputs and outputs of the neurons in the hidden layer are recalculated and, consequently, the inputs and outputs of the neurons of the output layer, resulting in new E_m. Therefore, considering that both forward propagation and backward propagation occur, the methodology used is called *BP*. For more details on all the procedures presented, see Hagan, Demuth and Beale (1996) and Hakim, Razak and Ravanfar (2015). The Appendix section shows a pseudocode with this algorithm.

Several mathematical tools can be applied to obtain better results. As we intend to carry out analysis by statistical parameters, all the above procedure will be repeated in several iterations, being that at each iteration all the samples will be reused several times (*epochs*) to obtain the optimal convergence or when the number of epochs is hit.

$$E_m(epoch) - E_m(epoch-1) \leq \varepsilon \quad (15)$$

From the minimum and maximum hit rates, statistical parameters (mean hit rate, standard deviation and variance), the choice of both the best activation function and the optimal number of hidden neurons will be developed. After determining the best neural network topology, the validation level will be measured from the *MSE*, but in only one iteraction, which will take into account the weights and bias already defined during the training and the outputs of the selected samples for classification.

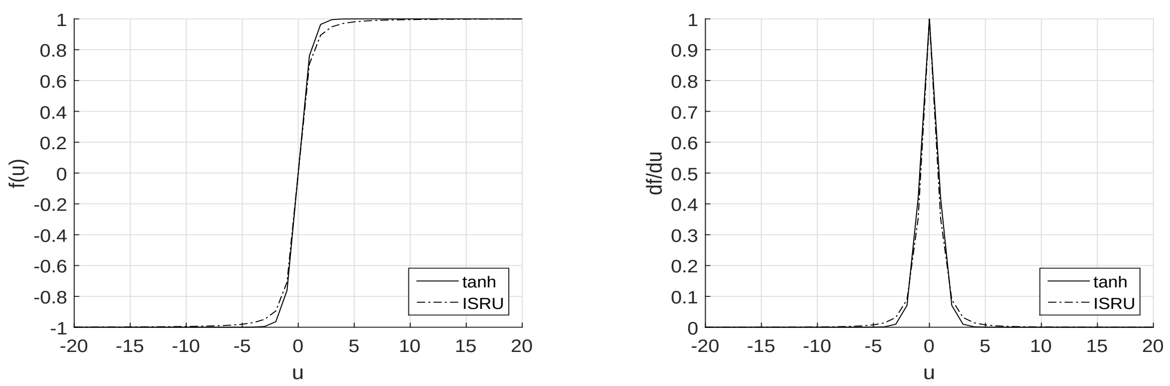
Activation functions

Especial topic of great importance to ensure a good performance of the network is related to activation functions that aim to limit the output of a neuron to a range of allowable values.

In this work, nonlinear and continuously differentiable activation functions will be used, according to Tab. 1 and Fig. 2. One of the most commonly used in pattern classification is the *tanh* function. Carlile et al. (2017) has proposed *ISRU* activation function that can be more efficient than that function and it has not still been used to detect, localize and classify damage based on vibration measurements.

Table 1: Activation functions and derivatives.

Activation function (label)	Expression	Derivative function
ISRU	$f(u) = u \cdot \left(1+u^2\right)^{-1}$	$f'(u) = \left(1+u^2\right)^{-\frac{3}{2}}$
tanh	$f(u) = (1-\exp(-2u)) \cdot (1+\exp(-2u))^{-1}$	$f'(u) = 1-(f(u))^2$



(a) Activation functions

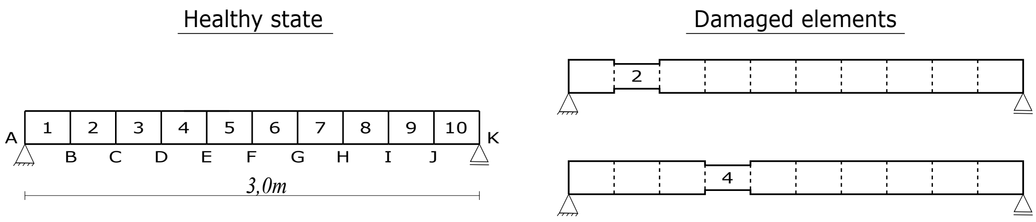
(b) Activation functions - first derivative.

Figure 2: Activation functions and their first derivatives.

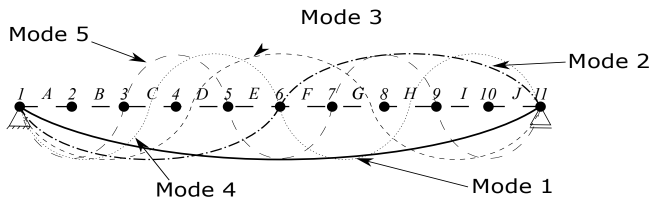
When visualizing these activation functions, Fig. 2a, it is possible notice that they have very similar behavior among each other. Furthermore, in the first derivative, Fig. 2b, the functions *tanh* and *ISRU* have the potential to achieve a higher convergence. All those activation functions were limited to range $[-1, 1]$.

CASE STUDY

In this work, a concrete beam was modeled in *FEM*, according to Fig. 3b, wherein the values in the 9 intermediate nodes (from *B* to *J*) for 5 modes of vibration, as well as their respective frequencies, were obtained for the non-damaged and the damaged structure according to the various imposed scenarios.



(a) The structure in three main stages: *healthy state*, *damaged element - 2*.



(b) Mode shapes for the first five modes of flexural vibration of the proposed beam.

Figure 3: Mode shapes 1 to 5 for *healthy state* and *damaged elements*.

Table 2 shows the damage parameters established for the beam and the Fig. 3a shows localization of each damaged element. Moreover, the longitudinal modulus of elasticity, **E**, for the material adopted is $24,000MPa$ and the beam cross-section has width (b_w) $15.0cm$ and height (h_w) $30.0cm$.

Table 2: Damage scenarios.

Scenarios	Damaged element	Reduction of "E"	Damage level	Number of classes
I	Healthy state	1.00**E**	0%	1
II - XXVI	2	0.98**E** : 0.02**E** : 0.50**E**	2.00% : 2.00% : 50.00%	25
XXVII - LI	4	0.98**E** : 0.02**E** : 0.50**E**	2.00% : 2.00% : 50.00%	25

The damage will be simulated by reducing **E** that will vary from 2.00% (0.98**E**) up to 50.00% (0.50**E**), Tab. 2. Thus, in each element, 25 damage levels will be simulated and modal parameters, obtained numerically from all damage scenarios, will feed into proposed *ANN*, as shown in Fig. 4. For each sample (damage scenario), the results will be outputted with the location, as well as the damage severity, which are previously known. At the end of the process, when all samples have fed the network and the error converges to the stipulated threshold value, the synaptic weights for the total set of training samples will be determined, reflecting that *ANN* will be trained. From this, the power of recognition and classification of trained *ANN* standards will be verified for the validation samples.

Although only one type of *ANN* topology composed of an input layer, a hidden layer and output layer is used, several architectures will be investigated from the variation of the number of neurons in the hidden layer and applying *tahn* and *ISRU* functions. The input layer will have 51 neurons, with frequencies and mode shapes of the undamaged and damaged structure, and the output layer will have two neurons that will provide the location and severity of the damage.

For each activation function and number of hidden neurons, the network will be compiled 50 times and in each iteration the training will occur respecting the limit of 50,000 epochs with the threshold value of the mean error of 0.0005. As this process will occur for each dimension of the hidden layer that will vary from 1 to 40 neurons, the results in each architecture will be discussed through statistical parameters. The learning rate for training is 0.01.

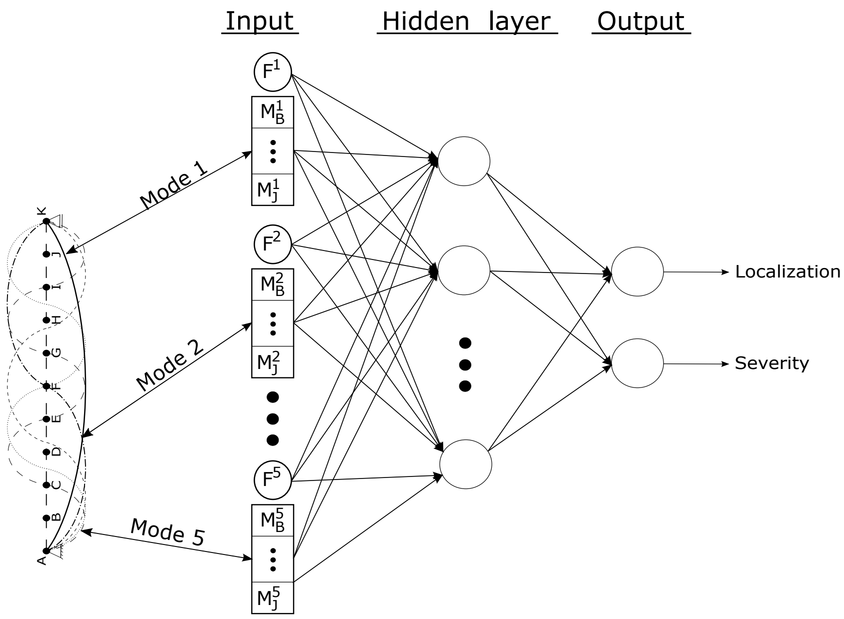
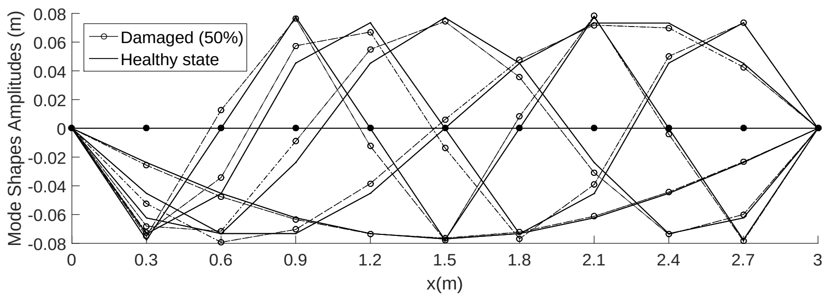


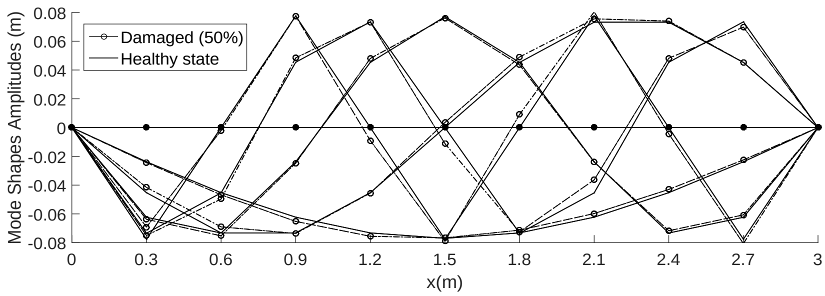
Figure 4: The adopted architecture of the proposed *ANN*.

NUMERICAL RESULTS AND DISCUSSION

The following results, Fig. 5, were obtained from numerical modeling processing by *FEM*, and they will be used as input values for training and classification of the proposed *ANN* architecture. The first five mode shapes (ϕ_1 to ϕ_5) were generated from *healthy state* and *damage scenarios*. Thus, both Fig. 5a and Fig. 5b show overlapping mode shapes with healthy state and damages of 50% in chosen elements. Therefore, given the enormous proximity of all curves, the challenge for neural networks to detect and classify the damage is notorious because each structural damage patterns imply such small changes in modal parameters.



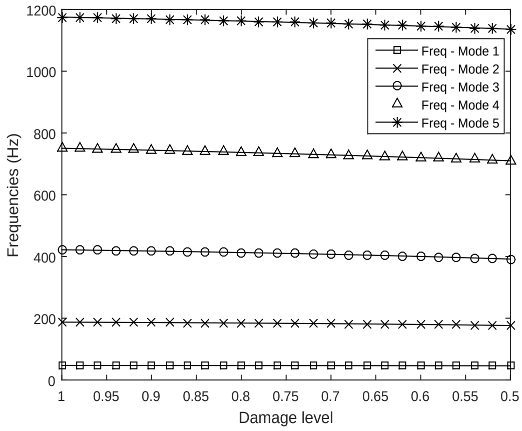
(a) Mode shapes 1 to 5 for *healthy state* and *maximum damage* in 2-element.



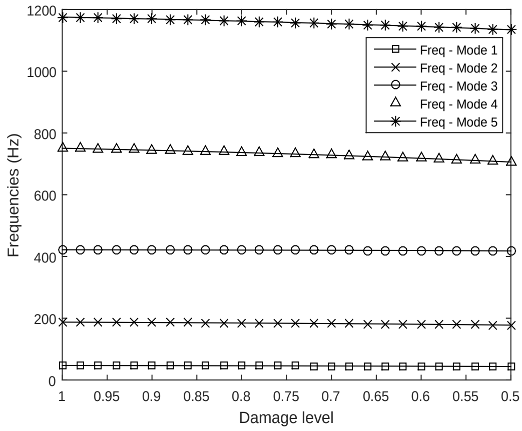
(b) Mode shapes 1 to 5 for *healthy state* and *maximum damage* in 4-element.

Figure 5: Mode shapes 1 to 5 for *healthy state* and *maximum damage*.

It is observed that the curves for the undamaged and damaged structures are very close. In addition, it is easy to intuit that the other damage scenarios in each element imply intermediate curves to those presented in Fig. 5. With the same interest, the frequencies are presented in Fig. 6, for each mode of vibration and in each damage scenario, it being noticeable that the magnitudes had small variations with the increase of the damage. Therefore, despite the availability of the above graphs, the diagnosis of the structure is not immediate through visual analysis of the results, being proposed the use of *ANNs* to detect, locate and classify the damage.



(a) Frequencies 1 to 5 for *healthy state* and *all damage scenarios* in 2-element.



(b) Frequencies 1 to 5 for *healthy state* and *all damage scenarios* in 4-element.

Figure 6: Frequencies 1 to 5 for *healthy state* and *all damage scenarios*.

In order to verify the best architecture of *ANN*, from the optimal number of neurons of the hidden layer and the activation function with better performance, it was chosen to perform the analysis through statistical parameters and 75% of the samples were used for training and 25% for classification or damage diagnosis. Therefore, of the 51 samples, 38 were used for training and 13 samples were selected for classification. This set of samples was rearranged at each iteration. The weights and biases were also initialized at each iteration and generated from random numbers using gaussian distributed.

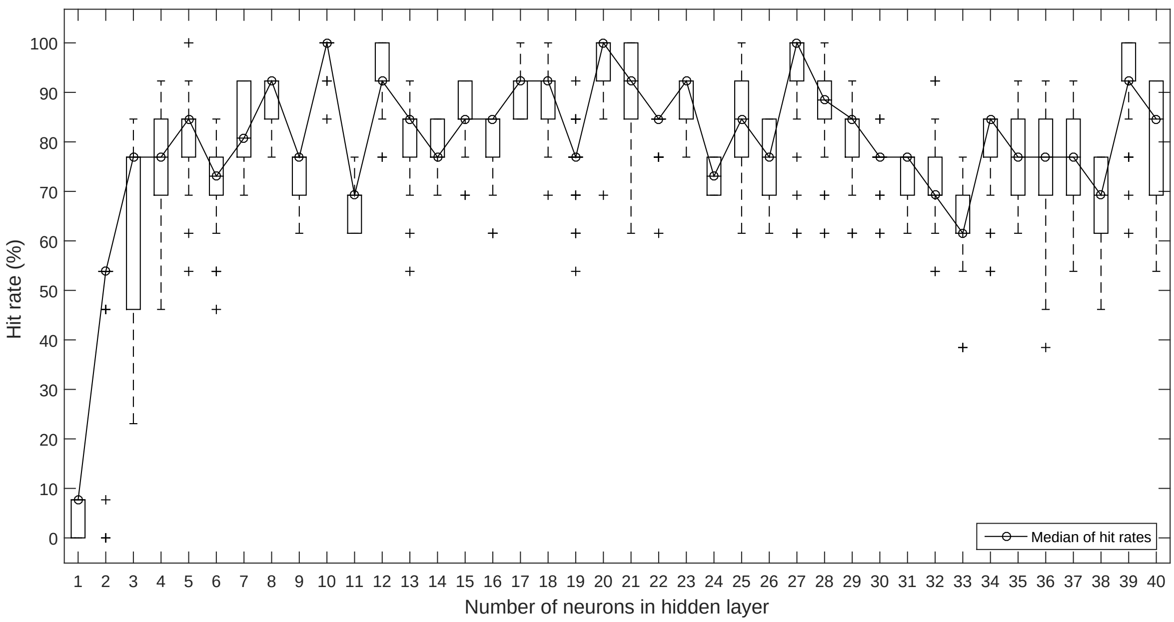


Figure 7: Hit rate by different number of neurons with *ISRU* activation function.

According to Fig. 7, using *ISRU* activation function, it can be seen that the architectures with 10, 20 and 27 neurons in the hidden layer obtained the maximum medians of 100% of hits, correctly identifying both the damaged element and the severity of the damage. In the architecture with 10 neurons, two outliers occurred, the smallest of which represents 84.6% hits. In the case of the second architecture with 20 neurons, a lower outlier with 69.2% hits occurred. In the third architecture, three outliers occurred, the smallest representing 61.5% hits. Therefore, knowing that the greatest number of neurons can lead to a high computational cost, the *ANN* with 10 neurons in the hidden layer stood out in terms of performance and excellent results and the one that presented the lowest variation in the hit rates. Here, we define ”hit” when the output neurons issue responses of both the damaged element and the damage severity simultaneously coinciding with the targets of the validation set.

From the results shown in Fig. 8, in cases whose activation function was used as *tanh*, the results in terms of median values were more stable than in the previous case, Fig. 7, obtaining practically the constant median of the order of 84.6%, for the cases with at least 3 neurons in the hidden layer. But in some architectures there were variations of median values up or down. However, it can be observed that, in relation to the architectures that resulted in higher median values, we can highlight the architectures with 17 and 32 neurons with 92.3% hits and the layer with 34 neurons that resulted in 96.2% of accurate diagnosis.

In the layer with 17 neurons, there was a lower outlier that represented 69.2% hits. In the second and third architectures, with 32 and 34 neurons respectively, the lower outliers were equal and of the order of 76.9%. Given these data, it is observed that choosing the architecture with 34 neurons could be more satisfactory in terms of hit rates, but it must be taken into account that the computational cost tends to become higher. Therefore, the first architecture aligns the appropriate hit rate with satisfactory computational cost.

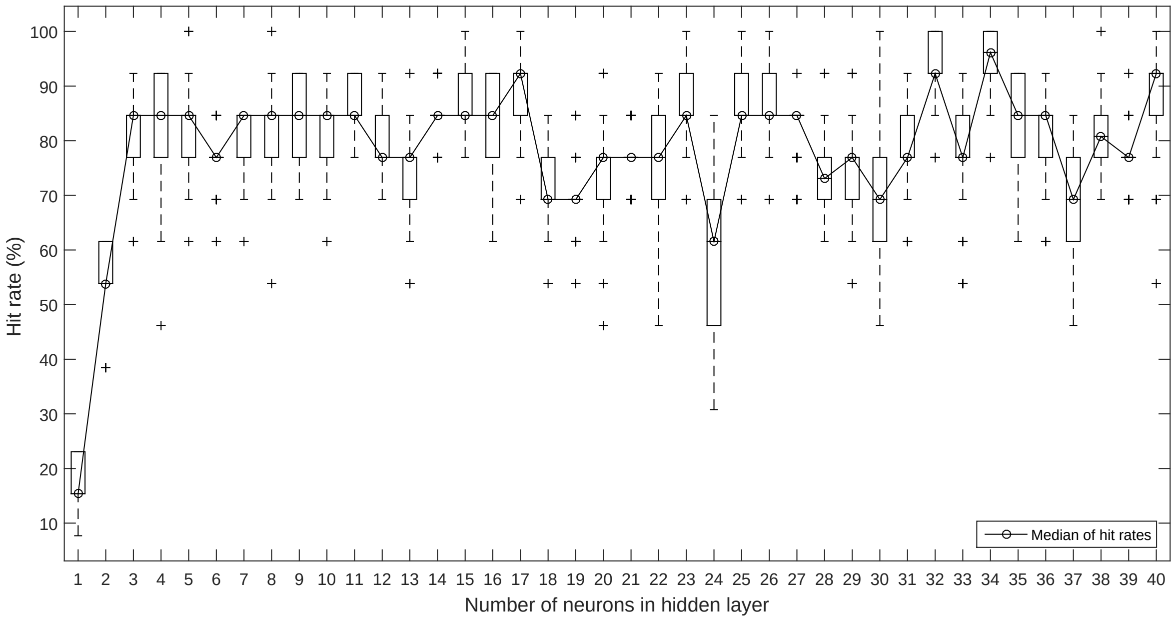


Figure 8: Hit rate by different number of neurons with *tanh* activation function.

Given the results obtained, it is possible to extract statistical parameters that support the choice of both the neural network architecture and the activation function, which together provide the best results and performance of the network, Tab. 3 and Tab. 4.

Table 3: Statistical analysis for *ISRU* activation function and three best ANN architectures.

Architecture	Minimum hit(%)	Maximum hit(%)	Mean hit(%)	Standard deviation	Variance	MSE
51-10-2	**84.6**	**100.0**	**99.1**	**0.38545**	**0.14857**	**0,000034**
51-20-2	69.3	**100.0**	96.5	0.81341	0.66163	0,000822
51-27-2	61.5	**100.0**	93.7	1.28873	1.66082	0.001442

The direct analysis of the *MSE* is not, in itself, satisfactory, considering that in all the selected architectures the values have been greatly reduced, demonstrating that the output value of the neural network and the real value are very close. Thus, by observing a set of the best statistical results, it was found that the architecture **51-10-2**, consisting of 10 neurons in the hidden layer, obtained the best performance from the application of the *ISRU* activation function.

Table 4: Statistical analysis for *tahn* activation function and three best ANN architectures.

Architecture	Minimum hit(%)	Maximum hit(%)	Mean hit(%)	Standard deviation	Variance	MSE
51-17-2	69.3	**100.0**	89.4	0.98747	0.97511	**0.000037**
51-32-2	**76.9**	**100.0**	93.3	0.91785	0.84245	0.000232
51-34-2	**76.9**	**100.0**	**94.9**	**0.77222**	**0.59633**	0.002563

CONCLUSIONS

It is known that the use of *ANNs* for damage detection in structures has been widely researched in the last three decades and the results are promissing. Thus, in order to contribute to this large research area, it is important that some concepts involved were rigorously analyzed. Therefore, this work had the goal of clarifying fundamental and inherent concepts to the *ANNs*, such as: what activation function and the number of neurons in the hidden layer result in better pattern recognition and classification for structural damage diagnosis, using *MLP* with *BP* and *GDM*.

The most well-known and applied activation function in *ANNs* problems and damage detection in structures is the *tahn* function which provides generally excellent results. The use of the *ISRU* activation function for similar damage detection problems in structures based on vibration measurements has not been found in the literature. However, when comparing the two activation functions, it was observed that the *ISRU* provided the best results regarding the structure diagnosis, and the best neural network architecture had 10 neurons in only one hidden layer. Therefore, the new application of the *ISRU* activation function has shown that its performance in detecting, locating and classifying the damage is quite promising and may be more exploited from now on.

Regarding the diagnosis based on vibration measurements of structures similar to the one proposed in the case study, the results showed that there was no need to increase the network with more than one hidden layer and with specialist networks, which could increase the computational cost.

ACKNOWLEDGMENTS

The authors would like to thank the financial support offered by Federal University of Piauí (UFPI) and to Professors Dr. José Menezes Júnior and Dr. Fábio Barbosa, both of the Department of Electrical Engineering at UFPI, for the fruitful explanations about Artificial Neural Networks.

REFERENCES

Adhikari, S., 2000, “Damping Models for Structural Vibration”, Ph.D. thesis, Cambridge University, Cambridge.

Arai, M., 1993, “Bounds on the Number of Hidden Units in Binary Valued Three-Layer Neural Networks”, Neural Networks, Vol.6, No. 6, pp. 855-860.

Bakhary, N., Hao, H., Deeks, A., 2010, “Structure Damage Detection Using Neural Network with Multi-Stage Substructuring”, Advances in Structural Engineering, Vol.13, pp. 95-110.

Carlile, B., Delamarter, G., Kinney, P., Marti, A. and Whitney, B., 2017, “Improving Deep Learning by Inverse Square Root Linear Units (ISRLUs)”, http://arxiv.org/abs/1710.09967.

Doebling, S.W., Farrar, C.R., Prime, M.B., 1998, “A Summary Review of Vibration-Based Damage Identification Methods”, Shock and Vibration Digest, Vol.30, pp. 91-105.

Lautour, O.R. de and Omenzetter, P., 2010, “Damage Classification and Estimation in Experimental Structures Using Time Series Analysis and Pattern Recognition”, Mechanical Systems and Signal Processing, Vol.24, No. 5, pp. 1556-1569.

Gnana Sheela, K. and Deepa, S.N., 2013, “Review on Methods to Fix Number of Hidden Neurons in Neural Networks”, Mathematical Problems in Engineering, Vol.2013, 11 pages.

Hagan, M.T., Demuth, H.B., Beale, M.H., 1996, “Neural Network Design”, Boston, MA, PWS Publishing.

Hakim, S., Razak, H.A. and Ravanfar, S., 2015, “Fault Diagnosis on Beam-Like Structures from Modal Parameters Using Artificial Neural Networks”, Measurement, Vol.76, No. Supplement C, pp. 45-61.

Haykin, S., 2001, “Redes Neurais: Princípios e Prática”, Ed. Bookman, Porto Alegre, RS, Brazil, 898 p.

Huang, G.B., 2003, “Learning Capability and Storage Capacity of Two-Hidden-Layer Feedforward Networks”, IEEE Transactions on Neural Networks, Vol.14, No. 2, pp. 274-281.

Hunter, D., YU, H., Pukish III, M.S., Kolbusz, J. and Wilamowski, B.M., 2012, “Selection of Proper Neural Network Sizes and Architectures: A Comparative Study”, IEEE Transactions on Industrial Informatics, Vol.8, No. 2, pp. 228-240.

Jinchuan, K. and Xinzhe, L., 2008, “Empirical Analysis of Optimal Hidden Neurons in Neural Network Modeling for Stock Prediction”, in Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Vol.2, pp. 828-832.

Mehrjoo, M., Khaji, N., Moharrami, H., Bahreininejad, A., 2008, “Damage Detection of Truss Bridge Joints Using Artificial Neural Networks”, Expert Systems with Applications, Vol.35, No. 3, pp. 1122-1131.

Panchal, G., Ganatra, A., Kosta, Y.P. and Panchal, D., 2011, "Behaviour Analysis of Multilayer Perceptronswith Multiple Hidden Neurons and Hidden Layers", International Journal of Computer Theory and Engineering. Vol.3, No. 2, pp. 332-337.

Silva, I.N. da, Spatti D.H., e Flauzino, R.A., 2016, "Redes Neurais Artificiais para Engenharia e Ciências Aplicadas: Curso Prático", Artliber Editora Ltda, 433 p., São Paulo, SP, Brasil.

Shibata, K. and Ikeda, Y., 2009, "Effect of Number of Hidden Neurons on Learning in Large-Scale Layered Neural Networks", in Proceedings of the ICROS-SICE International Joint Conference 2009 (ICCAS-SICE09), pp. 5008-5013.

Tamura, S. and Tateishi, M., 1997, "Capabilities of a Four-Layered Feedforward Neural Network: Four Layers Versus Three", IEEE Transactions on Neural Networks, Vol.8, No. 2, pp. 251-255.

Tan, Z.X., Thambiratnam, D.P., Chan, T.H.T. and Abdul Razak, H., 2017, "Detecting Damage in Steel Beams Using Modal Strain Energy Based Damage Index and Artificial Neural Network", Engineering Failure Analysis, Elsevier, Vol.79, pp. 253-262.

APPENDIX - TRAINING ALGORITHM - PSEUDOCODE

Algorithm 1 Pseudocode of MLP with backpropagation and GDM

Require: Get the sample set (frequencies and vibration modes) from MEF

Define the set of samples, p, for training
Associate each sample with the desired output vector (target - t_k)
Initialize the weights $(W^{(1)}_{j,i}, W^{(2)}_{k,j})$ with small random values and $sum \leftarrow 0$
Specify the learning rate (η), the desired precision (ε) and the number of epochs maximum $(nEpochs)$
for $epochs = 1 : nEpochs$ **do**
    for $q = 1 : length(Amostras)$ **do**
        $a_1 = f(\sum W^{(1)}_{j,i} \cdot I_i + b_1)$; *$I_i$ is the input values and "f" is activation function*
        $a_2 = f(\sum W^{(2)}_{k,j} \cdot a_1 + b_2)$
        $erro = t_k - o_k$; *o_k is the output values a_2*
        $\delta^{(2)} = -2 \cdot diag(a_2^{'}) \cdot (T_k - O_k)$;
        $\delta^{(1)} = diag(a_1^{'}) \cdot W^{(2)}_{k,j} \cdot \delta^{(2)}$
        $W^{(2)}_{k,j} = W^{(2)}_{k,j} - \eta \cdot \delta^{(2)} \cdot a_1$
        $b^{(2)} = b^{(2)} - \eta \cdot \delta^{(2)}$
        $W^{(1)}_{j,i} = W^{(1)}_{j,i} - \eta \cdot \delta^{(1)} \cdot p(q)$
        $b^{(1)} = b^{(1)} - \eta \cdot \delta^{(1)}$
        $sum \leftarrow (t_k - o_k)^2 + sum$
    end for
    $Emedio(epoch) = sum/length(Amostras)$
    if $abs(Emedio(epoch)) <= \varepsilon$ **then**
        break
    end if
end for

RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.