

## **Detección de daño en vigas utilizando redes neuronales artificiales y parámetros dinámicos**

### **Damage detection in beams by using artificial neural networks and dynamical parameters**

*Jesús D. Villalba<sup>1,\*</sup>, Ivan D. Gomez<sup>2</sup>, José E. Laier<sup>1</sup>*

<sup>1</sup>Departamento de Ingeniería de Estructuras. Escuela de Ingeniería de São Carlos. Universidad de São Paulo. Av. Trabalhador Sãocarlense 400, CEP 13566-590. São Carlos, SP, Brasil.

<sup>2</sup>Escuela de Ingeniería Civil. Universidad Industrial de Santander. Kra 9 N.º. 25-56. Bucaramanga, Colombia.

(Recibido el 15 de octubre de 2010. Aceptado el 29 de mayo de 2012)

#### **Resumen**

En este trabajo se presenta una red neuronal *perceptron* multicapa combinada con el método Nelder-Mead Simplex para detectar daño en vigas. Los parámetros de entrada a la red se basan en frecuencias naturales y flexibilidad modal. Se considera que solo una cantidad específica de modos fueron identificados y que se dispone de mediciones en grados de libertad verticales. La confiabilidad de la metodología propuesta se evalúa a partir de escenarios de daño aleatorios y de la definición de 3 tipos de error que la red puede cometer durante el proceso de detección del daño. Los resultados muestran que la metodología puede determinar confiablemente los escenarios de daño buscados. Sin embargo, su aplicación a vigas de gran tamaño puede verse limitada por el elevado costo computacional asociado al entrenamiento de la red.

----- *Palabras clave:* Detección de daños, redes neuronales, parámetros dinámicos

#### **Abstract**

In this paper is presented a multilayer perceptron neural network combined with the Nelder-Mead Simplex method to detect damage in multiple support beams. The input parameters are based on natural frequencies and modal flexibility. It was considered that only a number of modes were available and

---

\* Autor de correspondencia: teléfono: 55 + 16 33 73 94 85, correo electrónico: villalba@sc.usp.br (J. Villalba)

that only vertical degrees of freedom were measured. The reliability of the proposed methodology is assessed from the generation of random damages scenarios and the definition of three types of errors, which can be found during the damage identification process. Results show that the methodology can reliably determine the damage scenarios. However, its application to large beams may be limited by the high computational cost of training the neural network.

----- **Keywords:** Damage detection, neural networks, dynamical parameter.

## Introducción

Las estructuras civiles son sistemas susceptibles al deterioro ya sea por el uso, el envejecimiento, sobrecargas, fatiga o el acontecimiento de eventos extremos. Por lo tanto, la evaluación regular o continua de su estado a través de un sistema de detección de daños es una tarea imprescindible en situaciones donde se encuentren comprometidas la seguridad y la economía. Dichos sistemas contribuyen en la toma de decisiones sobre la necesidad de reforzar o reparar las estructuras reduciendo al mínimo la posibilidad de daños mayores y protegiendo la integridad de la estructura y de sus posibles usuarios.

En la literatura especializada se pueden encontrar diferentes metodologías para la detección de daños en estructuras civiles en forma no destructiva. Un grupo de esas metodologías consigue evaluar el daño sólo en forma local y consisten en inspecciones visuales o en la aplicación de técnicas que emplean acústica, ultra sonido, campos magnéticos, rayos-X, efectos térmicos, entre otros. A este tipo de metodologías se asocian dos grandes desventajas: la necesidad de que se conozca de antemano la localización del daño y que deba ser posible acceder a ellos [1]. Por otro lado, existen metodologías que permiten determinar en forma global el estado de la estructura, y entre ellas se encuentran aquellas que utilizan redes neuronales y datos de entrada tomados a partir de características dinámicas de la estructura.

La utilización de redes neuronales para detectar daño ha cobrado importancia en la comunidad

científica, encontrándose una gran cantidad de trabajos en el área. Chang et al [2] utilizaron una red neuronal iterativa y formas modales. Los casos de entrada se generaron utilizando el concepto de vectores ortogonales, el cual permite tener una muestra representativa de los posibles escenarios de daño y mantener al mínimo los casos de entrenamientos necesarios. Ni et al [3] propusieron detectar daño en un proceso de dos etapas. Primero, un conjunto de elementos probablemente dañados es encontrado a partir de vectores de entrada basados en parámetros dinámicos dependientes únicamente de la localización del daño. Después, una nueva red es entrenada con parámetros dinámicos que toman en cuenta la extensión del daño, pero solo de aquellos elementos localizados en la etapa 1. Fang et al [4] utilizaron funciones de respuesta en frecuencia como datos de entrada a una red neuronal *perceptron* multicapa. Se propuso el algoritmo *tunable steepest descent* para entrenar la red, el cual está equipado con una heurística que permite el control de la tasa de aprendizaje y produjo mejores resultados que diversos algoritmos reportados en la literatura. Lee y Kim [5] detectan la presencia de daño en forma estadística mediante un índice de anomalía obtenido a partir de respuestas de aceleración y deformaciones dinámicas. El índice de anomalía es utilizado junto con una red neuronal para localizar el daño. El desempeño de la red se evaluó a partir de ensayos experimentales mediante una tasa de identificación errada (*misclassification rate*).

En este trabajo se implementa una técnica de detección de daños a través del análisis de las características de la respuesta dinámica de la

estructura y de la utilización de redes neuronales. Primero, se realiza una breve descripción de los conceptos relacionados a las redes neuronales y después se presenta una descripción detallada de la metodología propuesta. El desempeño de la red se evalúa a partir de escenarios de daño simulados, considerando que pocos elementos estructurales presentan daño y que la extensión del daño no es severa.

### Teoría de redes neuronales

Una red neuronal es una herramienta matemática que establece una emulación del procesamiento de la información que realiza biológicamente el cerebro humano. En general, las redes neuronales son un conjunto de unidades idénticas, computacionalmente simples, llamadas neuronas. Estas unidades pueden tener conexiones de entradas, tanto del exterior, como de otras neuronas dentro de la red. A su vez, cada neurona puede tener salidas a otras neuronas de la red y/o al exterior. Cada conexión a las neuronas tiene asociado una resistencia usualmente modificable, la cual es llamada peso. Los pesos pueden ser positivos (excitadores) o negativos (inhibitorios). Además, cada neurona tiene un valor de sesgo (*bias*) que también es modificable. La salida de cada neurona depende tanto del *bias* como de la suma ponderada de todas sus entradas, transformado a través de una simple función lineal o no lineal, llamada función de activación. En las figuras 1 y 2 se muestran una representación esquemática de una neurona y diferentes funciones de activación, respectivamente.

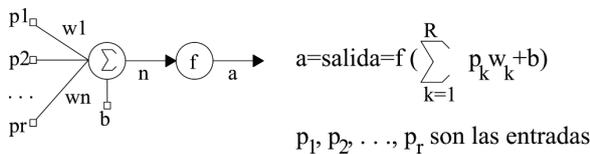


Figura 1 Esquema de una red neuronal

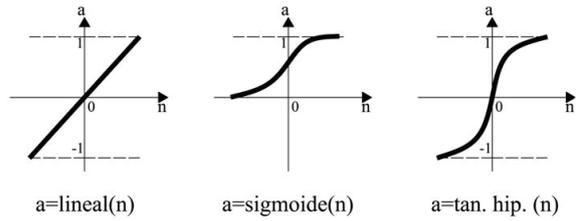


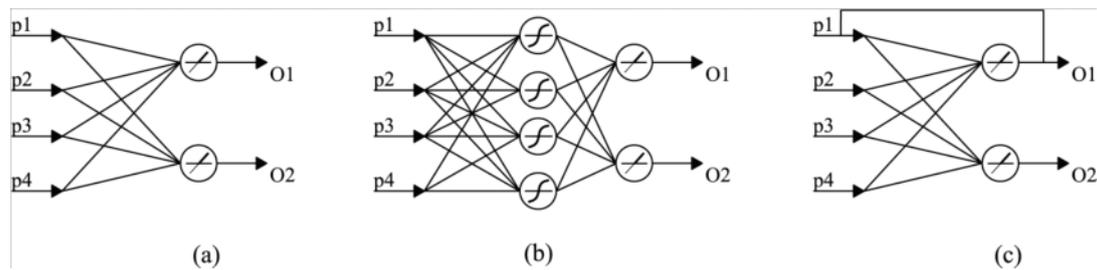
Figura 2 Funciones de activación

### Tipos de conexión

Las redes neuronales son construidas utilizando tres tipos de conexiones: intra-capas, entre-capas y conexiones recurrentes. La conexión intra-capa, la cual es llamada conexión lateral o conexión cruzada, son *links* entre neuronas de la misma capa de la red. Las conexiones entre-capas son *links* entre neuronas en diferentes capas. Las conexiones recurrentes proporcionan un link de auto-retroalimentación (*self-feedback*) a las neuronas. En las conexiones entre-capas, el flujo de la información se da de dos formas: alimentación hacia adelante (*feedforward*) o retroalimentación (*feedback*).

### Arquitectura de la red

En las redes neuronales se pueden distinguir tres tipos de estructuras: redes *feedforward* de una sola capa, redes *feedforward* multicapas y redes recurrentes, como se muestra en la figura 3. El tipo más común de redes *feedforward* de una sola capa es el *perceptron*. La configuración de las redes *feedforward* multicapas se caracteriza por un flujo de información de las entradas a las salidas sin *feedback* en las capas. Es conocido el teorema de Cybenko [6] que establece que “una red neuronal *feedforward* con suficiente cantidad de neuronas ocultas, con funciones de transferencia continuas y diferenciables pueden aproximar cualquier función continua sobre un intervalo cerrado”. En una red multicapas deberán ser definidos los vectores de entrada a la red, los parámetros de salida, el número de capas intermedias u ocultas, y el número de neuronas por capa.



**Figura 3** Arquitecturas de redes neuronales: a) única capa, b) multi-capa y c) recurrente

### Entrenamiento de la red

Para poder aplicar una red neuronal a una tarea específica se requiere que los pesos de las conexiones y de los *bias* sean determinados, lo cual es realizado mediante un aprendizaje iterativo, donde cada iteración recibe el nombre de época. En el aprendizaje supervisado, la regla de aprendizaje está compuesta de un conjunto de ejemplos, conocido como el conjunto de entrenamiento, compuesto por suficientes entradas y salidas (objetivos) que permita describir el modelo dinámico. Luego, las entradas se aplican a la red y las salidas de la red son comparadas con los objetivos. El objetivo es reducir el error que se presenta entre el objetivo y la salida de la red. Este proceso se lleva a cabo iterativamente hasta que la red alcance la respuesta deseada o alcance un estado de equilibrio.

### Modelado de las estructuras inicial y con daño

El estado inicial de la estructura es representado por un modelo de elementos finitos (MEF), el cual permite que sean obtenidos los parámetros dinámicos de dicha condición. El modelo que representa la condición con daño es obtenido a partir de la actualización del modelo inicial. En este trabajo se considera que el daño afecta únicamente la matriz de rigidez en la forma de una reducción en el módulo de elasticidad del elemento con daño [7]. Si el elemento  $j$  presenta daño, su nuevo módulo de elasticidad,  $E_{d_j}$ , puede ser calculado en función del módulo de elasticidad original,  $E_j$ , como

$$E_{d_j} = E_j \times (1 - \beta_j) \quad (1)$$

con  $\beta_j$  el factor de reducción de elasticidad calculado para el elemento  $j$ . La Ecuación (1) es aplicada para todos los elementos en la estructura, tal que si  $\beta_j$  asume un valor de 0 significa que el elemento no presenta daño y un valor de 1 representa la pérdida total del elemento.

Las matrices de rigidez de la estructura para las condiciones inicial y actual pueden ser calculadas a partir de la contribución de la rigidez de cada uno de los elementos, así:

$$K_{inic} = \sum_{j=1}^{nelem} k_j(E_j, I_j, L_j) \quad (2)$$

$$K_d = \sum_{j=1}^{nelem} k_j(E_{d_j}, I_j, L_j) \quad (3)$$

donde  $nelem$  es el número de elementos de la estructura,  $k_j$ ,  $I_j$  y  $L_j$  son la matriz de rigidez global, la inercia y la longitud del elemento  $j$ , respectivamente. *inic* se refiere a la condición inicial y *d* a la condición con daño. La matriz de masa se considera constante y se calcula así:

$$M = \sum_{j=1}^{nelem} m_j(m_j, a_j, \rho_j) \quad (4)$$

con  $m_j$ ,  $a_j$ ,  $\rho_j$  siendo la matriz de masa global, el área de la sección y la densidad del material del elemento  $j$ .

Finalmente, los parámetros dinámicos de las estructuras inicial y con daño podrán ser determinados a partir de:

$$(K_{inc} - \omega_{inc}^2 M) \phi_{ini} = 0 \quad (5)$$

$$(K_d - \omega_d^2 M) \phi_d = 0 \quad (6)$$

donde  $\omega$  corresponde a frecuencia natural y  $\phi$  a la forma modal.

## Red neuronal artificial propuesta

### Arquitectura de la Red

En este trabajo se implementó una red neural artificial de tipo *Perceptron* Multicapa en MATLAB utilizando el entrenamiento *Scaled conjugate gradient backpropagation*. Este programa se escogió debido a que dispone de un *toolbox* de redes neuronales donde la red y el algoritmo de entrenamiento ya se encuentran programados. El usuario debe suministrar los vectores de entrada y salida de la red junto con una serie de parámetros, como: el número de capas ocultas y la cantidad de neuronas en cada capa. Después de realizar algunos entrenamientos preliminares, el número de capas ocultas fue definido como dos y el número de neuronas en cada capa como el 80% del tamaño del vector de valores de entrada, *VVE*. Dicho vector está formado por parámetros dinámicos y se describe a continuación:

$$VVE = [W_1 \ W_2 \ \dots \ W_{nm} \ F_1 \ F_2 \ \dots \ F_{nglv}] \quad (7)$$

donde  $nm$  es la cantidad de frecuencias naturales y formas modales medidas y  $nglv$  es el número de grados de libertad verticales. Los términos  $W$  y  $F$  son calculados como:

$$W_j = \frac{\omega_{j,ANN} - \omega_{j,SD}}{\omega_{j,SD}} \quad (8)$$

$$F_i = \frac{f_{ii,ANN} - f_{ii,SD}}{f_{ii,SD}} \quad (9)$$

donde  $\omega_j$  es la  $j$ -ava frecuencia natural y  $f_{ii}$  es la posición  $i$  en la diagonal de la matriz de flexibilidad modal referida a los grados de libertad verticales. Los subíndices *ANN* y *SD* representan valores obtenidos a partir de la red neuronal artificial y del modelo de elementos finito de la estructura sin daño, respectivamente. La matriz de flexibilidad puede ser obtenida a partir de las frecuencias naturales y las formas modales como:

$$[f] = \sum_{j=1}^{nm} \frac{1}{\omega_j} \{\phi\}_j \{\phi\}_j^t \quad (10)$$

donde  $\phi_j$  es la  $j$ -ava forma modal.

Por otro lado, los parámetros de salida corresponden al factor de reducción del módulo de elasticidad de cada elemento en la estructura, y cuyo tamaño es igual al número de elementos en que fue dividido el MEF de la estructura.

En la figura 4 se puede observar la configuración de la red para una viga simplemente apoyada con 4 elementos y en la cual se tiene información de los dos primeros modos de vibración. Por lo tanto, esta red tendrá 5 valores de entrada (2 frecuencias naturales y 3 valores de flexibilidad modal), 4 neuronas en cada capa oculta y 4 valores de salida correspondiendo al factor de reducción de elasticidad de cada uno de los 4 elementos.

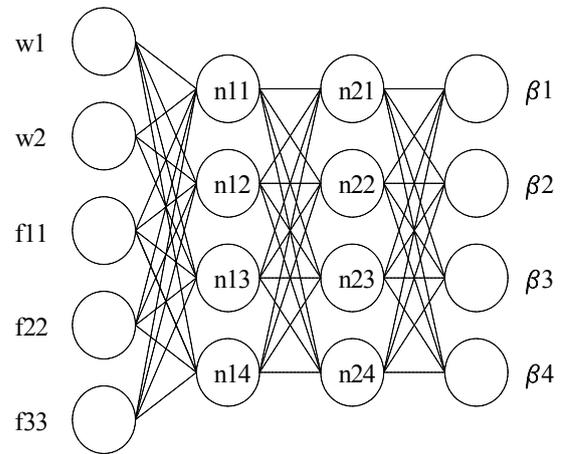
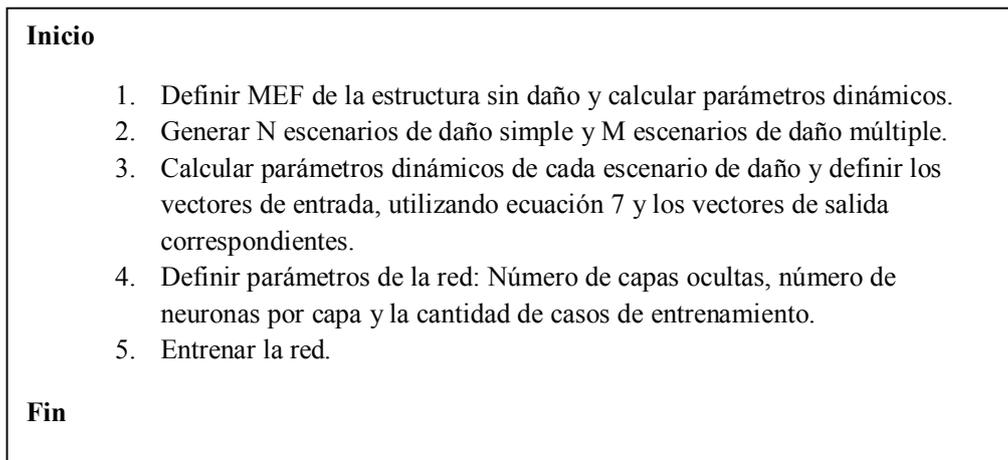


Figura 4 Arquitectura de la red neuronal

### **Definición de las etapas previas al entrenamiento de la red**

El entrenamiento de la red es realizado siguiendo el procedimiento mostrado en la figura 5. Primero, se define un MEF el cual servirá como representación del estado sin daño de la estructura. En el segundo paso son simulados los diversos escenarios de daño, los cuales son clasificados como simples o múltiples en función del número de elementos dañados que presenten. Los escenarios de daño simple son formados por un único elemento dañado para el cual se varía uniformemente la cantidad de daño entre los límite máximo y mínimo predefinidos. Lo anterior, es realizado para cada elemento en la estructura y considerando que el rango de daño sobre el cual trabajará la red neuronal es de 0,05 a 0,55. Para la viga mencionada anteriormente un caso de daño simple corresponde, por ejemplo, a daño de 0,2 en el elemento 2. Para la creación de los escenarios de daño múltiple se genera un número aleatorio entre 2 y 4 (máximo

número permisible de elementos dañados) el cual definirá el número de elementos dañados que tendrá dicho escenario. Los elementos y sus extensiones de daño son determinados en forma aleatoria también. Un ejemplo de un escenario de daño múltiple corresponde a daño en los elementos 1, 3, y 4 con factores de reducción de elasticidad de 0,1, 0,3 y 0,15, respectivamente. La determinación del número de escenarios de daño necesarios para entrenar la red será realizada en función del desempeño de la red y será discutida más adelante. En el paso 3, las frecuencias naturales y las formas modales de las estructuras configuradas por la red son calculadas con ayuda de la Ecuación (6). A partir de estos valores los vectores de entrada pueden ser creados utilizando las Ecuaciones (7) a (10). En el paso 4 son definidas las características de la red neuronal y en el paso final es realizado el proceso de entrenamiento de la red, la cual podrá iterar hasta alcanzar un número de épocas máxima igual a 20000 o hasta alcanzar un nivel de error deseado.



**Figura 5** Pasos previos al entrenamiento de la red

#### *Aplicación*

Un número pre-determinado de escenarios de daño serán simulados para cada una de las vigas analizadas con el objetivo de demostrar el desempeño de la red. La generación de estos escenarios sigue el mismo procedimiento utilizado

para crear los casos de entrenamiento correspondientes a daño múltiple, pero en este caso el número de elementos dañados puede variar entre 1 y 4. La aplicación de esta metodología, consiste en generar los vectores de entrada a partir de los escenarios de daño simulados, sustituyendo los términos relacionados con la red neuronal en

las Ecuaciones 8 y 9 por aquellos obtenidos para evaluar la red.

Los resultados obtenidos por la red serán mejorados a partir de la implementación del método de optimización no lineal Nelder-Mead Simplex, NMS, y cuya configuración de parámetros fue aquella utilizada en [8]. Las variables siendo optimizadas corresponden a aquellos elementos en los cuales la red determinó una extensión de daño mayor a 0,03. La función objetivo utilizada es la siguiente:

$$Max G = \sum_{j=1}^{nm} \frac{200}{1 + |\omega_{j,NMS\_ANN} - \omega_{j,D}|} \quad (11)$$

donde *NMS\_ANN* se refiere a la solución encontrada después de aplicar el NMS.

El desempeño de la metodología propuesta será analizado en relación a tres tipos de error:

- Error Tipo 1: Debido a la diferencia entre el valor de la extensión de daño calculada por la metodología y el valor de daño de un elemento que realmente presenta daño.
- Error Tipo 2: Se presenta cuando la metodología determina un valor de daño para un elemento que realmente no se encuentra dañado.

- Error Tipo 3: Se presenta cuando no se consigue determinar que un elemento se encuentra dañado.

Tres estadísticas son realizadas en función de cada uno de los tipos de error, siendo cada tipo de error, a su vez, dividido según el grado del error. Cada escenario de daño será clasificado por el mayor grado de error que presente para un tipo de error dado.

## Ejemplos numéricos

### Descripción de las vigas analizadas

En este trabajo se analizan 8 estructuras de tipo viga (figura 6), las cuales presentan diferente número de elementos finitos (entre 8 y 40). Cada elemento en las diferentes vigas tiene módulo de elasticidad =200E9 N/m<sup>2</sup>, densidad =78E2 kg/m<sup>3</sup>, área= 1E-2 m<sup>2</sup> e inercia= 5E-4 m<sup>2</sup>. Un total de 200 escenarios de daño son simulados para cada una de las vigas con el objetivo de determinar la capacidad de la metodología propuesta para detectar daño. Finalmente, se considera que es posible tener información sobre las primeras 8 formas modales para todas las vigas y que se desea evaluar daño entre 0,05 e 0,55.

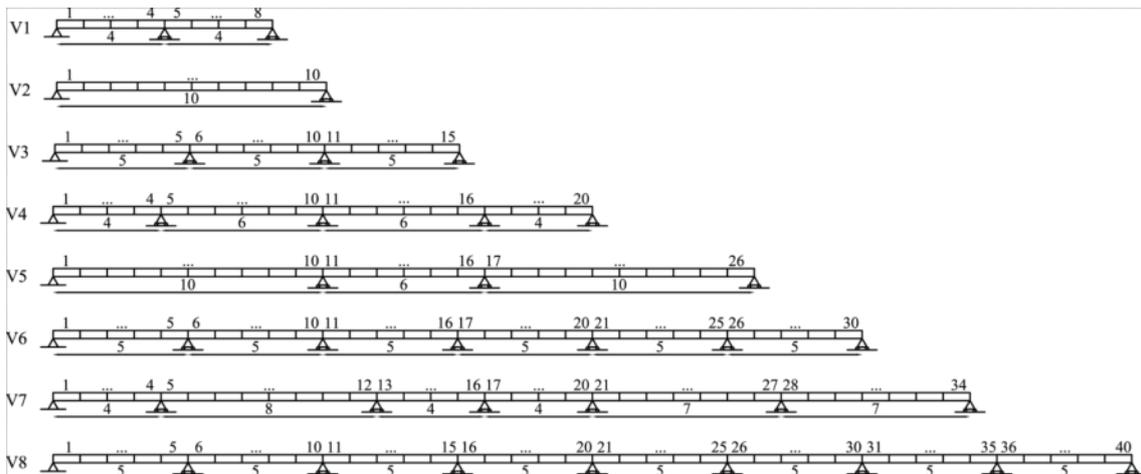


Figura 6 Vigas analizadas

### Aplicación de la metodología a casos puntuales

La metodología propuesta es aplicada en la detección de escenarios de daño simple y múltiple en la viga V5, ver figuras 7, 8 y 9, con el objetivo

de mostrar una solución típica encontrada. En los ejemplos analizados se observa que la red consigue detectar los elementos realmente dañados con una extensión de daño próxima de la real, así como la presencia de una serie de elementos falsamente identificados con bajos niveles de daño.

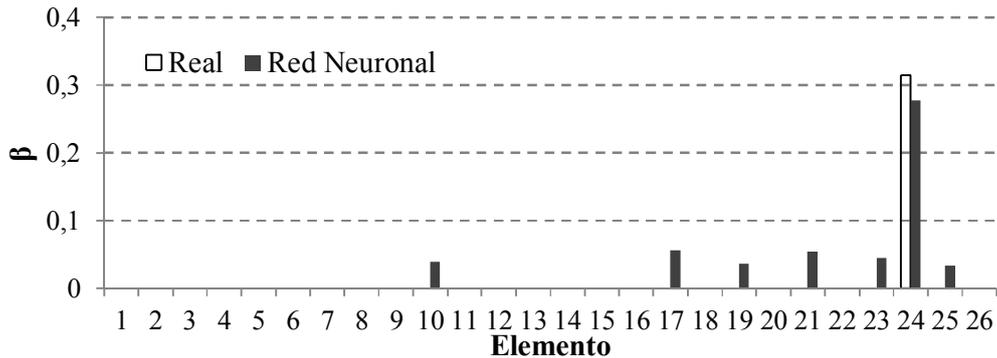


Figura 7 Caso1: daño en un único elemento

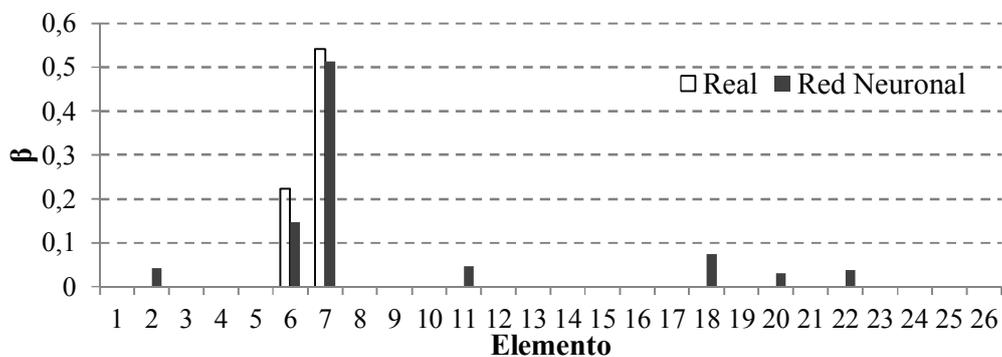


Figura 8 Caso 2: daño en dos elementos contiguos

En las siguientes secciones los resultados mostrados son referidos a los 200 escenarios

simulados para cada una de las vigas analizadas.

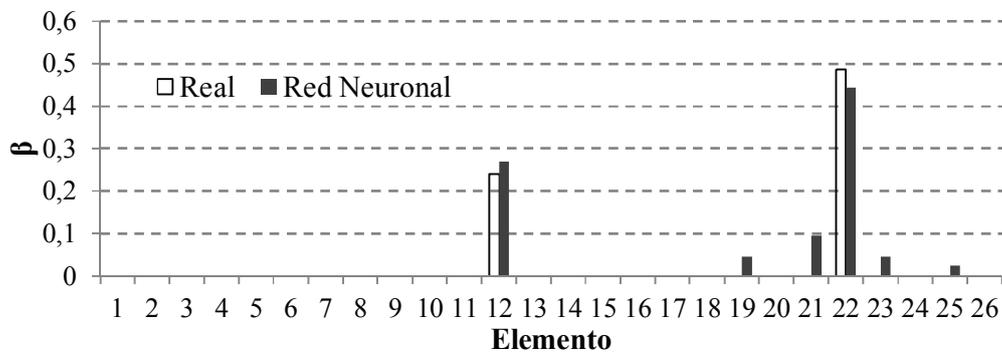


Figura 9 Caso 3: daño en dos elementos separados

**Desempeño de la metodología: Cálculo del Error Tipo 1**

Un factor que influye en el desempeño de la red neuronal es el número de casos de entrenamiento utilizado y su origen. En este trabajo se consideró que ese conjunto estaba conformado por vectores de entrada provenientes de escenarios de daño simple y múltiple. Para el primer caso, los ejemplos fueron creados utilizando un elemento por vez y dividiendo el rango de daño en 25 partes, así el total de casos puede ser calculado como:

$$NES = 25 \times \# Elem \quad (12)$$

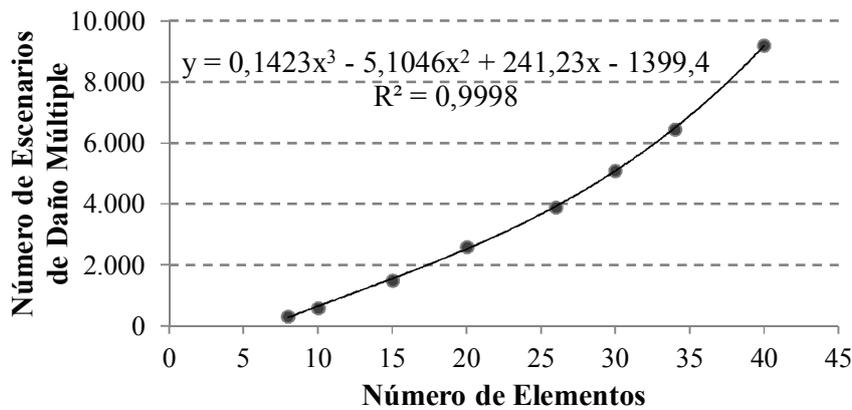
donde *NES* es el número de casos de entrenamiento provenientes de escenarios de daño simples y *#Elem* es la cantidad de elementos en la estructura.

Por otro lado, la determinación del número necesario de escenarios de daño múltiples fue realizada en forma iterativa. Diferentes cantidades de escenarios fueron utilizadas hasta garantizar que por lo menos en el 80% de los casos simulados

el error máximo en el cálculo de la extensión del daño fuera menor a 0,05. En la tabla 1 se presenta el número de escenarios de daño utilizado para cada una de las vigas estudiadas y en la figura 10 se muestra la variación del número de casos de entrenamiento de daño múltiple en función del número de elementos en la estructura.

**Tabla 1** Cantidad de escenarios utilizados para el entrenamiento de la red

Viga	Número de casos		
	Simple	Múltiple	Total
V1	200	320	520
V2	250	600	850
V3	375	1.500	1.875
V4	500	2.600	3.100
V5	650	3.900	4.550
V6	750	5.100	5.850
V7	850	6.460	7.310
V8	1.000	9.200	10.200



**Figura 10** Relación entre el número de escenarios de daño múltiple y el número de elementos de la viga

A partir de los resultados encontrados en la figura 10, se derivó una expresión que permite calcular, en forma aproximada, el número de

casos de entrenamiento múltiples, *NEM*, bajo las condiciones utilizadas en este trabajo (cantidad de modos disponibles, rango de valores de daño y número de elementos dañados máximo):

$$NEM = 0.1423 \times \# Elem^3 - 5.1046 \times \# Elem^2 + 241,23 \times Elem - 1399.4 \quad (13)$$

Finalmente, la cantidad total de casos de entrenamiento, *NET*, viene dada por:

$$NET = NES + NEM \quad (14)$$

La cantidad de escenarios presentados en la tabla 1, producen los valores de Error Tipo 1 presentados en la tabla 2. Los resultados se muestran en forma de porcentaje del número de casos en los cuales una determinada diferencia entre el valor de daño real y el calculado,  $\Delta\beta$ , es encontrada, siendo los resultados en paréntesis

aquellos encontrados por la red sin la aplicación del NMS. Es importante anotar que  $\Delta\beta$  es calculado para cada elemento en la estructura, siendo el grado de error correspondiente al escenario estudiado definido en función del elemento que presente el peor  $\Delta\beta$ . En general se observó que para todas las vigas la aplicación del NMS mejoró el resultado de la red, siendo bastante notorio para las vigas V5 y V7. Sin embargo, el NMS no consiguió mejorar los resultados de aquellos escenarios que presentaban el peor grado de error.

**Tabla 2** Porcentaje de casos simulados en los cuales se presenta el error tipo 1

<i>Viga</i>	$\Delta\beta < 0.02$	$0.02 \leq \Delta\beta < 0.05$	$0.05 \leq \Delta\beta < 0.1$	$\Delta\beta \geq 0.1$
V1	73(72)	13(14)	8(8)	6(6)
V2	63(60)	18(21)	13(12)	6(7)
V3	84(78)	11(17)	3(3)	2(2)
V4	83(80)	10(13)	3(3)	4(4)
V5	73(50)	7(30)	13(12)	7(8)
V6	84(78)	10(15)	5(6)	1(1)
V7	83(68)	7(22)	7(7)	3(3)
V8	84(78)	11(17)	3(3)	2(2)

**Desempeño de la metodología: Cálculo del Error Tipo II y III**

En las tablas 3 y 4 se muestra el desempeño de la metodología en relación a los errores Tipo 2, elementos falsamente identificados como dañados, y Tipo 3, elementos dañados que no son identificados. En la medida en que la estructura crece la tendencia del Error Tipo 2 es crecer en el grado de error; sin embargo,

solo se encontraron elementos erróneamente clasificados como dañados con una extensión de daño mayor a 0,2 en un 3% de los casos aproximadamente. En relación al Error Tipo 3 la cantidad de casos en los cuales un elemento con daño menor a 0,1 no es identificado se mantuvo más o menos constante, con un promedio de 14,5%. En general, en menos del 3% de los casos no se determinó la presencia de un elemento dañado con más de 0,1 de daño.

**Tabla 3** Porcentaje de casos simulados en los cuales se presenta el error tipo 2

<i>Viga</i>	$\Delta\beta < 0.05$	$0.05 \leq \Delta\beta < 0.1$	$0.1 \leq \Delta\beta < 0.2$	$\Delta\beta \geq 0.2$
V1	70	19	8	3
V2	56	34	10	0

<b>Viga</b>	<b><math>\Delta\beta &lt; 0.05</math></b>	<b><math>0.05 \leq \Delta\beta &lt; 0.1</math></b>	<b><math>0.1 \leq \Delta\beta &lt; 0.2</math></b>	<b><math>\Delta\beta \geq 0.2</math></b>
V3	75	19	6	0
V4	51	26	20	3
V5	19	51	30	0
V6	23	43	31	3
V7	18	56	26	0
V8	14	36	48	2

**Tabla 4** Porcentaje de casos simulados en los cuales se presenta el error tipo 3

<b>Viga</b>	<b><math>\Delta\beta &lt; 0.1</math></b>	<b><math>\Delta\beta \geq 0.1</math></b>
V1	10	6
V2	11	2
V3	7	2
V4	12	3
V5	15	2
V6	17	1
V7	27	2
V8	17	2

**Relación entre la cantidad de modos medidos y el desempeño de la metodología propuesta**

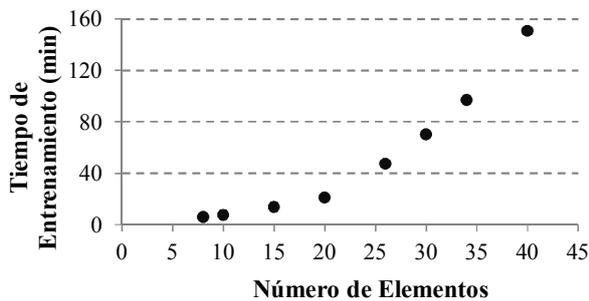
La viga V3 se utiliza para observar el efecto del número de modos medidos en el desempeño de la metodología propuesta. Los resultados mostrados en la tabla 5 indican que a partir de 6 modos la red presenta un desempeño constante, con un error en el cálculo de la extensión del daño, de los elementos con daño, menor a 0,02 en por lo menos el 85% de los casos. Siendo así, cuando se desee aplicar la metodología propuesta se deberán realizar simulaciones numéricas que permitan determinar la cantidad de modos óptima necesaria para obtener un desempeño determinado. Es importante anotar que este valor se encuentra limitado superiormente por el número de modos que pueden ser medidos experimentalmente.

**Tabla 5** Desempeño de la metodología propuesta en función del número de modos utilizados: Error tipo 1

<b>Modos</b>	<b><math>\Delta\beta &lt; 0.02</math></b>	<b><math>0.02 \leq \Delta\beta &lt; 0.05</math></b>	<b><math>0.05 \leq \Delta\beta &lt; 0.1</math></b>	<b><math>\Delta\beta \geq 0.1</math></b>
2	62	11	16	11
4	77	14	5	4
6	86	6	5	3
8	85	10	3	2
10	88	5	5	2
12	86	9	4	1

### Determinación del tiempo de entrenamiento

En la figura 11 se muestra como varía el tiempo de entrenamiento, tomados en un PC Intel® Core (TM) 2 Quad de 2.67 Ghz, en función del tamaño de la estructura. Dicha variación presenta un comportamiento no lineal debido al incremento no lineal de los casos de entrenamiento en relación al tamaño de la estructura. Este hecho limita la aplicación de redes neuronales para detectar daño en estructuras con muchos elementos debido a que se requiere de computadores de gran desempeño que puedan manejar eficientemente la cantidad de información involucrada.



**Figura 11** Tiempo de entrenamiento de la red en función del número de elementos

### Conclusiones

En este trabajo se presentó una metodología que combina parámetros dinámicos, redes neuronales (*Perceptron* multicapa) y una técnica de optimización (Nelder-Mead Simplex) para determinar la presencia de daño en estructuras de tipo viga. Un total de 200 escenarios de daño fueron simulados para comprobar la confiabilidad de la metodología propuesta. Los resultados muestran que el error máximo en la determinación de la extensión del daño es inferior a 0,05 en el 80% de los casos y que solo en un 3% de los casos simulados se presentaba un elemento falsamente identificado como dañado cuya extensión de daño fuera mayor a 0,2. Así mismo, en un promedio de 2% de los casos simulados

se encontró que la metodología no conseguía identificar un elemento que tuviera daño mayor a 0,1.

En relación a la configuración y definición de la metodología podemos decir lo siguiente:

1. Los vectores de entrada se conformaron a partir de frecuencias naturales y flexibilidades modales, siendo considerado que unas pocas formas modales fueron medidas únicamente en los grados de libertad verticales. Se observó que el número de formas modales disponibles puede influenciar el resultado encontrado por la metodología propuesta.
2. La definición de la cantidad de capas ocultas y el número de neuronas por capa se realizó a partir de diferentes simulaciones. Se propuso una ecuación para calcular el número de casos de entrenamiento en vigas cuyos MEF tengan hasta 40 elementos, la cual busca ayudar al usuario a estimar un valor aproximado de los casos de entrenamiento pero sin que represente una regla fija. Diferentes tipos de entrenamiento se pueden utilizar en el aprendizaje de la red, lo cual podría resultar en un mejor desempeño de la metodología propuesta.
3. El algoritmo de optimización puede ser cambiado por otros, tal como algoritmos genéticos, con el fin de obtener un mejor proceso de refinamiento de la solución.

### Agradecimientos

Los autores desean reconocer al *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (Brasil) por el apoyo financiero, el cual permitió el desarrollo de este trabajo.

### Referencias

1. S. Doebling, C. Farrar, M. Prime. "A Summary Review of Vibration-Based Damage Identification Methods". *The shock and vibration digest*. Vol. 30. 1998. pp. 91-105.
2. C. Chang, T. Chang, Y. Xu, M. Wang. "Structural Damage Detection Using an Iterative Neural Network".

- J. of Intelligent Material Systems and Structures*. Vol. 11. 2000. pp. 32-42.
3. Y. Ni, B. Wang, J. Ko. "Constructing Input Vectors to Neural Networks for Structural Damage Identification". *Smart Materials and Structures*. Vol. 11. 2002. pp. 825-833.
  4. X. Fang, H. Luo, J. Tang. "Structural Damage Detection using neural network with learning rate improvement". *Computers and Structures*. Vol. 83. 2005. pp. 2150-2161.
  5. J. Lee, S. Kim. "Structural Damage Detection in the Frequency Domain using Neural Networks". *J. of Intelligent Material Systems and Structures*. Vol. 18. 2007. pp. 785-792.
  6. G. Cybenko. "Approximation by superposition of a Sigmoidal Function". *Mathematics of Control, Signals and Systems*. Vol. 2. 1989. pp. 303-314.
  7. B. Koh, S. Dyke. "Structural health monitoring for flexible bridge structures using correlation and sensitivity of modal data". *Computers and Structures*. Vol. 85. 2007. pp. 117-130.
  8. O. Begambre. *Algoritmo Híbrido para a Avaliação da Integridade Estrutural: Uma Abordagem Heurística*. Tesis Doctoral. Universidad de São Paulo. São Paulo (Brasil). 2007. pp 152.