

## REDES NEURONALES APLICADAS A DINÁMICA DE ESTRUCTURAS

Oscar Möller<sup>\*</sup>, Lucas Luchesi<sup>\*</sup>, Ricardo O. Foschi<sup>†</sup>, Marcelo Rubinstein<sup>\*</sup>

<sup>\*</sup> Instituto de Mecánica Aplicada y Estructuras (IMAE), Universidad Nacional de Rosario, Riobamba y Berutti, 2000 Rosario, Argentina.  
e-mail: moller@fceia.unr.edu.ar

<sup>†</sup> Civil Engineering Department, University of British Columbia  
6250 Applied Sciences Lane, Vancouver, B.C., Canada V6T 1Z4.  
e-mail: rowfal@civil.ubc.ca

**Palabras clave:** Confiabilidad estructural, Redes neuronales, Dinámica no lineal.

**Resumen.** *En el análisis de la confiabilidad de sistemas estructurales no lineales bajo acciones dinámicas, donde es necesario aplicar métodos de simulación, resulta muy conveniente y efectivo aproximar la respuesta del sistema con un sustituto. Para ello se discute la aplicación de teorías de aprendizaje estadístico, en particular el algoritmo de redes neuronales. Mediante un proceso iterativo de entrenamiento se optimiza la arquitectura de la red y los factores de peso. La aplicación a una estructura de tres pisos muestra la muy buena precisión que se logra con esta técnica y la facilidad para evaluar probabilidades de falla.*

**Abstract.** *For the reliability analysis of non linear structural systems under dynamic actions, where simulation methods are needed to apply, it is convenient and effective to approximate the response of the system with a solver surrogate. It is discussed the applicability of statistical learning techniques, in particular neural network algorithm. Through an iterative training process the architecture of the net and the weight factors are optimized. An example of three story system shows the very good precision reached by this technique and the facility for failure probability calculations.*

## 1 INTRODUCCIÓN

En el diseño de estructuras se busca que el comportamiento sea satisfactorio durante el tiempo de la vida útil. Para acciones sísmicas el problema es dinámico, es decir la demanda y la respuesta estructural son variables con el tiempo, y dentro del marco de la “Ingeniería Sísmica basada en el desempeño”<sup>1,2</sup> se definen varios “niveles de desempeño” como requisitos de diseño.

Las importantes incertidumbres presentes en las variables que gobiernan el problema hacen que la formulación se realice en términos probabilísticos, evaluando la probabilidad de falla, o de excedencia anual, de cada nivel de desempeño o estado límite.

Los métodos para llevar adelante el análisis de la confiabilidad estructural se pueden dividir en dos grupos<sup>3,4</sup>:

a) Métodos de *segundo momento y transformación*: basados en expansión en serie de Taylor de la función de falla  $G(\mathbf{X})$  alrededor del punto de diseño (punto de falla más probable) que se obtiene durante el proceso de solución. Estos métodos se los conoce como FORM y SORM: First and Second Order Reliability Meted, respectivamente.

Necesitan del cálculo de la primera (FORM) y segunda (SORM) derivada de la  $G(\mathbf{X})$ , lo cual es un inconveniente porque en general la función de falla es implícita en las variables básicas  $\mathbf{X}$ , y se debe recurrir a diferencias finitas. Significa una importante cantidad de evaluaciones de  $G(\mathbf{X})$ , que crece con la dimensión de  $\mathbf{X}$ , es decir de análisis dinámico no lineal con método de elementos finitos, y problemas numéricos de precisión por las irregularidades locales.

b) Métodos de *simulación* tipo Monte Carlo: son completamente generales, no necesitan de derivación numérica ni cálculo del punto de diseño, pero la tarea computacional es enorme, a veces impracticable. Por ejemplo para probabilidades de falla del orden de  $10^{-3}$  son necesarias  $10^5$  a  $10^6$  simulaciones con análisis dinámico no lineal. La cantidad de simulaciones se puede disminuir con las técnicas de reducción de variancia, como los métodos de Importance sampling o Adaptive sampling, pero deben incorporar información sobre la ubicación del punto de diseño que a priori es desconocido, generándose procesos iterativos.

Un camino para superar estas dificultades es encontrar un *sustituto* del análisis completo por elementos finitos<sup>5</sup>, tal que con muy poca cantidad de análisis dinámico no lineal se puedan optimizar los parámetros del sustituto. Luego en el análisis de confiabilidad, cada vez que se requiera calcular la respuesta estructural  $R(\mathbf{X})$  se la aproxima con el sustituto  $F(\mathbf{X})$ , que es explícito y para su evaluación prácticamente no se consume tiempo.

En este sentido se ha empleado el *método de la superficie de respuesta*<sup>6,7</sup> con la siguiente expresión genérica:

$$R(X) \cong F(\mathbf{X}) = a + \sum_{i=1}^N b_i X_i + \sum_{i=1}^N c_i X_i^2 = \sum_{i=1}^N w_i h_i(X) \quad (1)$$

donde  $h_i(\mathbf{X})$  son funciones de potencia de las coordenadas y  $w_i$  parámetros que se

optimizan a partir de conocer al menos  $2N+1$  valores de  $R(\mathbf{X})$ .

El problema es la ubicación de los puntos para evaluar  $R(\mathbf{X})$  y éstos deben estar próximos al punto de diseño para aproximar correctamente la función  $F(\mathbf{X})$  en la zona de mayor contribución a la probabilidad de falla. Se recurre a procesos iterativos, o a valoraciones globales de  $F(\mathbf{X})$  haciendo cumplir requisitos del problema<sup>8</sup>.

Esta técnica tiene características de (i) Funciones  $h_i(\mathbf{X})$  son no adaptables al no depender de los puntos elegidos para definir  $F$ . La responsabilidad de la aproximación es únicamente de los pesos  $w_i$ , y entonces el modelo es muy sensible a la ubicación de dichos puntos que determinan los valores de  $w_i$ ; (ii) Funciones  $h_i(\mathbf{X})$  son no flexibles debido a que tienen efecto sobre todo el dominio y los errores que pueda haber en alguna evaluación de  $R(\mathbf{X})$  afectan a todo la aproximación.

En este trabajo se analizan nuevas alternativas como es la aplicación de *Teorías de aprendizaje estadístico*<sup>5</sup> y en particular el algoritmo de *Redes neuronales*<sup>5,9,10,11</sup>, cuyo nombre se debe a la similitud con el funcionamiento de estructuras biológicas y de ahí el nombre también utilizado de *Inteligencia artificial*.

La expresión genérica es

$$R(X) \cong F(\mathbf{X}) = h \left( \sum_{k=0}^m w_k h \left( \sum_{i=1}^N w_{ki} X_i \right) \right) \quad (2)$$

donde  $h(t) = 1 / (1 + \exp(-t))$  es una función no lineal llamada sigmoide.

Se mejoran las características del método de la superficie de respuesta ya que el modelo de redes neuronales es (i) Adaptable: los parámetros de las funciones son determinados por los puntos utilizados en la optimización de la red; (ii) Flexible: la región activa de cada función base  $h_i(t)$  es limitada, de manera que valores de  $X$  alejados de su zona de influencia no producen resultados de significación. Así los errores o diferencias en algún punto no se difunden; (iii) Capacidad para *aprender* de la experiencia y ejemplos: la red se entrena a partir de un conjunto de datos y resultados conocidos del análisis completo del sistema.

Entre los varios algoritmos desarrollados para entrenar redes neuronales conocidos como técnicas de retropropagación<sup>5,9</sup>, en este trabajo se discute el algoritmo del descenso del gradiente con la regla de incrementos con tasa de aprendizaje y momento variables. También se analiza la arquitectura óptima de la red, la valoración de errores y los esquemas iterativos.

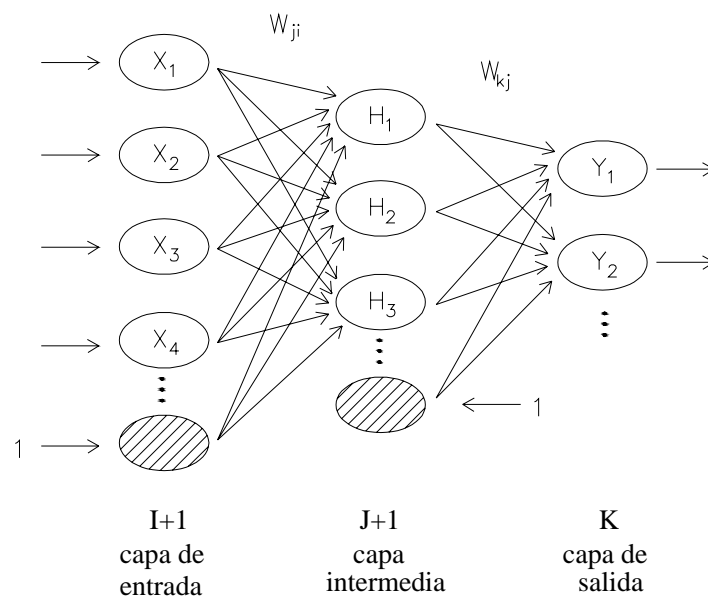
La discusión se realiza a través de un sistema estructural con comportamiento no lineal, de tres pisos bajo acciones dinámicas de terremotos, encontrándose interesantes resultados.

## 2 REDES NEURONALES

Las redes neuronales artificiales son dispositivos computacionales compuestos por una cantidad de unidades de procesamiento interconectadas. Cada una de ellas retiene información localmente y es capaz de realizar simples cálculos. La red en conjunto tiene la habilidad de responder frente a estímulos de entrada y producir las respuestas correspondientes, además puede adaptarse a diferentes situaciones aprendiendo de la experiencia.

### 2.1 Aproximación de la respuesta del sistema

Existen muchos algoritmos de redes neuronales artificiales entre los cuales se encuentra el de “Red Neuronal Multicapa”. Este comprende los siguientes elementos básicos: una capa de neuronas de entrada, una o varias capas de neuronas intermedias u ocultas y una capa de neuronas de salida. Son las capas ocultas las que cumplen un rol fundamental en el entrenamiento de la red. Gracias a las mismas la red puede modelar relaciones no lineales entre las variables de entrada y de salida. En la figura 1 se muestra un esquema de la arquitectura de la red implementada en este trabajo.



**Figura 1:** Red neuronal multicapa

Las neuronas de la capa de entrada sólo reciben información, es decir los datos de entrada del problema, y la transmiten a las neuronas de la capa oculta. Estas procesan esta información y obtienen resultados parciales que luego son transmitidos a las neuronas de la capa siguiente, de salida. Estas últimas son las que producen las predicciones de la red.

Esta transferencia de información de neuronas de capas adyacentes se realiza a través de

unos coeficientes de peso  $w$  y de una función de transferencia  $h(x)$ . Si en la figura 1 se considera que el número de neuronas en la capa de entrada, oculta y de salida son  $I+1$ ,  $J+1$  y  $K$  respectivamente; y que  $X_i^p$  es el  $p$ -ésimo dato de la  $i$ -ésima neurona de la capa de entrada,  $I_j^p$  el  $p$ -ésimo dato de la  $j$ -ésima neurona de la capa oculta,  $H_j^p$  el  $p$ -ésimo resultado de la  $j$ -ésima neurona de la capa oculta,  $I_k^p$  el  $p$ -ésimo dato de la  $k$ -ésima neurona de la capa de salida, e  $Y_k^p$  el  $p$ -ésimo resultado de la  $k$ -ésima neurona de la capa de salida, luego resultan las siguientes expresiones:

$$I_j^p = \sum_i^I w_{ji} X_i^p + w_{j0} \quad H_j^p = h(I_j^p) \quad (3)$$

$$I_k^p = \sum_j^J w_{kj} H_j^p + w_{k0} \quad Y_k^p = h(I_k^p) \quad (4)$$

donde  $w_{ji}$  son los factores de peso que conectan la  $i$ -ésima neurona de la capa de entrada en la  $j$ -ésima neurona de la capa oculta,  $w_{kj}$  son los factores de peso que conectan las neuronas  $j$ -ésima con la  $k$ -ésima y  $h(\cdot)$  es la función de transferencia dada por:

$$h(x) = \frac{1.0}{(1 + \exp(-x))} \quad (5)$$

La red es entonces entrenada presentándole una base de datos y resultados disponibles, optimizando los factores de peso  $w_{ji}$ ,  $w_{kj}$ . De acuerdo al principio de minimización del riesgo empírico logra minimizar la diferencia entre los resultados obtenidos con análisis estructural y los obtenidos con la red. Luego de ser entrenada, con los factores de peso ya optimizados, la red puede ser usada para predicciones.

## 2.2 Entrenamiento de la red

El primer paso es definir la arquitectura de la red, es decir el número de capas ocultas y el número de neuronas por capa. En este trabajo se utiliza una sola capa intermedia y una capa de salida con una única neurona, es decir que cada red solo se utiliza para obtener un único resultado a partir de la información de entrada. El número de neuronas de la capa de entrada resulta de la cantidad de variables de entrada mas una neurona, que puede tener un valor máximo de 50.

Para seleccionar las combinaciones de los valores de las variables de entrada que se utilizarán para entrenar la red, se aplica la teoría de diseño de experimentos. Además, durante el entrenamiento, los coeficientes de peso  $w$  se optimizan proporcionalmente a unas derivadas que tienden a valores pequeños cuando el valor absoluto de los coeficientes crece, por lo que se escalan las variables de entrada a un rango de actuación entre  $-1.0$  y  $1.0$  para agilizar el entrenamiento.

$$U = -1.0 + 2.0 \frac{(X - X_l)}{(X_u - X_l)} \quad (6)$$

donde  $U$  representa a los valores normalizados de las variables de entrada  $X$ , y  $X_l$  y  $X_u$  son los límites inferior y superior de las mismas

La normalización de las variables de salida depende de la función de transferencia utilizada  $h(\cdot)$ , la cual tiene un límite inferior de 0 y uno superior de 1 (función sigmoidea). En principio se utiliza una transformación no lineal debido a que es más conveniente cuando los resultados están próximos entre sí y luego se escalan entre 0.1 y 0.9, no entre 0 y 1 debido a que para esos valores no están definidos los logaritmos.

$$S = \ln(Y) \quad V = 0.1 + 0.8 \frac{(S - \ln Y_l)}{(\ln Y_u - \ln Y_l)} \quad (7)$$

donde  $V$  es el valor normalizado de la variable de salida  $Y$ ,  $Y_l$  e  $Y_u$  son los límites inferior y superior de las mismas

En caso que las variables de salida puedan asumir valores menores o iguales que 0 se adiciona un término dentro del logaritmo para trabajar siempre con argumentos positivos en forma equivalente.

$$S = \ln(|Y_l| + 3 + Y) \quad (8)$$

El número óptimo de la capa oculta debe determinarse de acuerdo a cada problema en particular. En general una red con pocas neuronas en la capa intermedia no logra capturar las relaciones entre los datos y los resultados, mientras que una con muchas neuronas tiende a no poder generalizar para datos distintos a los de entrenamientos. El número óptimo de neuronas en esta capa se determina como se indica a continuación.

Se define un número máximo de neuronas en función del número de variables de entrada y un mínimo igual a 1. El número máximo de neuronas es de suma importancia en el tiempo de ejecución del entrenamiento de la red. Existen relaciones empíricas entre el número óptimo de neuronas y el número de variables de entrada, pero en general deben verificarse para cada problema en particular.

Se van generando distintas arquitecturas, cada una con número de neuronas en la capa intermedia creciente de a una, entre los límites inferior y superior.

En cada arquitectura el total de la muestra de datos y resultados se divide en 5 conjuntos, de los cuales 4 se utilizan para entrenar la red, es decir el 80%, y el restante, 20%, para la validación de los resultados obtenidos. Se realizan 5 pruebas, en las que se combinan los grupos de entrenamiento y de validación de a uno por vez, es decir para el primer caso el primer grupo para validación y los demás para entrenamiento, y así sucesivamente para los 5 grupos. En cada una de estas pruebas se determina el error generalizado (total), el cual se minimiza utilizando un algoritmo iterativo de retropropagación de errores. Es un método de gradiente descendente en el cual los factores de peso  $w$  se van optimizando iterativamente (aprendizaje) en forma proporcional a la diferencia entre los valores objetivos  $T_k$  (resultados disponibles) y las predicciones de la red  $Y_k$ .

Se elige finalmente la arquitectura de la red, es decir el número óptimo de neuronas en la

capa intermedia, aquella que resulte con menor error generalizado.

Este error generalizado, sobre las muestras de entrenamiento y de validación, se calcula:

$$E = \frac{1}{2} \sum_p \sum_k (T_k^p - Y_k^p) \quad (9)$$

Luego para minimizarlo se debe cumplir para los coeficientes de peso que conectan la capa oculta con la capa de salida:

$$w_{kj}(g+1) = w_{kj}(g) + \Delta w_{kj}(g) = w_{kj}(g) - \eta \frac{\partial E}{\partial w_{kj}} \quad (10)$$

donde  $g$  es el número de la iteración y  $\eta$  es la tasa de entrenamiento.

Usando la regla de derivación en cadena, se puede escribir:

$$\Delta w_{kj} = \eta \sum_p (T_k^p - Y_k^p) h'(I_k^p) H_j^p \quad (11)$$

Llamando:

$$\delta_k^p = (T_k^p - Y_k^p) h'(I_k^p) \quad (12)$$

Entonces (11) resulta:

$$\Delta w_{kj} = \eta \sum_p \delta_k^p H_j^p \quad (13)$$

Para las neurona de la capa oculta, para las cuales no hay valores objetivos de salida, se propaga el error hacia atrás según el mismo criterio.

$$w_{ji}(g+1) = w_{ji}(g) + \Delta w_{ji}(g) = w_{ji}(g) - \eta \frac{\partial E}{\partial w_{ji}} \quad (14)$$

Usando la regla de derivación en cadena

$$\Delta w_{ji} = \eta \sum_p \sum_k (T_k^p - Y_k^p) h'(I_k^p) w_{kj} h'(I_j^p) X_i^p = \eta \sum_p \delta_j^p X_i^p \quad (15)$$

donde:

$$\delta_j^p = h'(I_j^p) \sum_k \delta_k^p w_{kj} \quad (16)$$

La tasa de entrenamiento  $\eta$  depende de cada problema, en general si es demasiado alta la red podrá no converger nunca al resultado para algunas muestras, si en cambio es demasiado baja la convergencia será muy lenta. En este método de retropropagación el entrenamiento es rápido al principio pero luego se hace lento en una región plana de la superficie del error. Para mejorar esta situación se fijan inicialmente los coeficientes de peso a valores pequeños y se agrega un segundo término de momento.

$$\Delta w(g) = -\eta \nabla E(g) + \alpha \Delta w(g-1) \quad (17)$$

donde  $g$  es el número de la iteración y  $\alpha$  es el denominado momento.

El número de iteraciones límite  $g_{lim}$  para determinar el mínimo error generalizado en la selección del número óptimo de neuronas de la capa oculta es:

$$g_{lim} = NITET \sqrt{NHID} \quad (18)$$

siendo  $NITET$  una constante y  $NHID$  el número de neuronas de la capa intermedia.

Una vez determinado la arquitectura óptima se procede al entrenamiento de la red. En primer lugar se divide de la misma forma la muestra en 5 subconjuntos y se utiliza el mismo algoritmo de retropropagación del error para actualizar los coeficientes de peso  $w$ . Se calcula el error relativo de entrenamiento y de validación para cada elemento de la muestra y se obtiene el mayor de todos.

$$\max E_{rel} = \max (T_k^p - Y_k^p) / Y_k^p \quad (19)$$

En caso que este error máximo no sea inferior a un error límite de 10% se reordenan los grupos que se utilizaban para entrenar y para validar, ubicando a las muestras con mayores errores relativos en el conjunto de entrenamiento y a las que tenían errores menores en el de validación. Se vuelven a actualizar los coeficientes de peso  $w$  hasta que se cumpla la condición del error máximo del 10% o hasta un número de ciclos límite (para este trabajo 50).

Respecto al cálculo de errores relativos  $E_{rel}$  se plantea el problema de calcularlos en variables originales o cuando ya han sido transformadas y escaladas. Debido a que son errores relativos se pueden tener dos situaciones diferentes, si la muestra posee valores objetivos iguales a cero, es conveniente calcular los errores en variables transformadas. Otra posibilidad es evaluar el error en variables originales y cuando se detecten aquellos casos con valores objetivos nulos omitir el cálculo del error. Si en cambio no aparecen resultados nulos es lo mismo calcularlos de una u otra forma.

Por otro lado el número total de ciclos de entrenamiento tiene gran incidencia en el tiempo de ejecución del programa debido a que por cada ciclo deben realizarse todas las iteraciones  $g_{lim}$  del algoritmo de retropropagación del error.

Cuando se hayan cumplido una de las condiciones para finalizar los ciclos se ejecuta finalmente un último entrenamiento donde se actualizan los coeficientes de peso  $w$  por última vez pero con un número de iteraciones  $10 g_{lim}$  (en general), y luego se determinan algunos parámetros estadísticos de la red como ser promedio de los errores, desvío estándar, etc.

Por último siempre es conveniente una última prueba para evaluar el poder de generalización de la red. Esto se logra con una nueva muestra de datos y resultados obtenidos con programa de análisis de estructuras, y comparándolos con los resultados o predicciones dadas por la red para esas mismas variables de entrada.



### 3 DISEÑO DE EXPERIMENTOS

Desde que la red “aprende” de las muestras presentadas a la misma, la generación de un número suficiente de muestras de entrenamiento es muy importante.

Estas muestras deben cubrir todo el espacio de diseño desde el límite inferior al superior de cada variable. El problema de seleccionar aleatoriamente las muestras de las variables de diseño puede ser tratado con la teoría de diseño de experimentos<sup>9</sup>.

Uno de los métodos de diseño es el de la grilla basada en el diseño óptimo. Se comienza por escalar las variables a combinar al espacio [0;1] de la siguiente manera:

$$x_{ki} = \frac{X_{ki} - X_i^l}{(X_i^u - X_i^l)} \quad (20)$$

donde  $x_{ki}$  :  $k$ -ésima muestra de la  $i$ -ésima variable en el espacio [0;1]

$X_i^l$  : límite inferior de la  $i$ -ésima variable

$X_i^u$  : límite superior de la  $i$ -ésima variable

$X_k^i$  :  $k$ -ésima muestra de la  $i$ -ésima variable en el espacio de diseño

Es importante para este fin determinar los límites inferior y superior de cada variable acorde al problema particular. En este trabajo se consideró un rango de variación dado por  $\mu \pm 3\sigma$ , siendo  $\mu$  valor medio y  $\sigma$  el desvío standard.

Para cada variable la distancia entre el límite inferior y superior se divide en sectores, por lo que se van generando hipercubos de caras rectangulares en el espacio normalizado. Además se define el número de muestras en cada hipercubo. En caso de que alguna variable sea de mayor importancia que otras, se les asigna mayor cantidad de sectores. Resulta así un número total de combinaciones dado por el número de variables, de sectores por variable y de muestras en cada hipercubo.

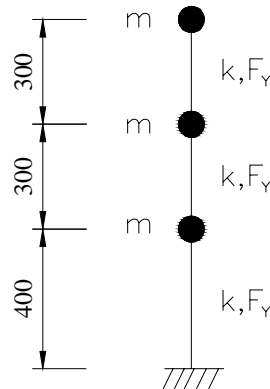
Con una secuencia de números aleatorios se generan las combinaciones de valores de las variables cubriendo los diferentes hipercubos. Este proceso se repite varias veces, calculándose en cada caso la mínima distancia entre las muestras generadas. Se elige finalmente el caso con distancia mínima mayor, evitando el aglutinamiento de muestras.

Por último las variables son escaladas al espacio de diseño. Estas transformaciones de variables se deben fundamentalmente a que las variables pueden tener distintas dimensiones, y calcular las distancias en el espacio original puede distorsionar la distribución de las muestras.

## 4 APLICACIONES

### 4.1 Datos de la estructura

Para el entrenamiento de la red se utiliza un sistema estructural con comportamiento no lineal, de tres pisos bajo acciones dinámicas de terremotos. Se considera un grado de libertad por piso y la masa concentrada por piso. En la figura 2 se muestra el sistema utilizado.



**Figura 2:** Sistema analizado

Como variables intervinientes en el problema se consideraron la masa  $m$  por piso, la rigidez a corte  $k$  de cada nivel con su resistencia  $F_y$ , y la aceleración pico del suelo  $a_G$ . Se utilizó el registro del acelerograma de Kobe (1995), el cual fue escalado para los diferentes valores de  $a_G$ . En la tabla 1 se resumen los datos de las variables.

**Tabla 1:** Datos de las variables

Descripción	Promedio	Límite inferior	Límite superior	Nro. de sectores
$m$ : masa por piso	37.5	26.3	48.8	3
$k$ : rigidez	50	35	65	3
$F_y$ : resistencia	23.6	16.5	30.7	3
$a_G$ : aceleración pico	300	150	900	5

### 4.2 Casos considerados

Utilizando la técnica de diseño de experimentos descrita en el punto 3, se generaron 405 combinaciones de las variables considerando distintos sectores en cada una de ellas.

Para cada una de las 405 combinaciones se obtuvo el desplazamiento máximo del último piso, la distorsión de piso máxima y el corte basal con un programa de análisis dinámico no

lineal con el método de los elementos finitos. Además se agregaron nuevas combinaciones con aceleraciones del suelo nulas para contar también con valores objetivos nulos.

Se entrenaron distintas redes, una para cada uno de los resultados, para distintas situaciones. Se consideraron las siguientes variantes:

- Número máximo de neuronas de la capa intermedia (NEUMAX): 20 y 30
- Número de ciclos de entrenamiento (NITES): 15 y 50.
- Constante del número de iteraciones para minimización del error (NITET): 5000 y 50000.

Para verificar la aproximación de la respuesta de la red se realizaron nuevas combinaciones de las variables dentro del rango de actuación de las mismas y se calcularon para cada una de estas el desplazamiento máximo del último piso, la distorsión máxima y el corte basal con el mismo programa de análisis no lineal como se explicó anteriormente. Luego utilizando las redes entrenadas se obtuvieron las predicciones de las mismas. Se detallan a continuación algunas de las situaciones estudiadas.

### 4.3 Resultados

Se analiza la variación del error generalizado (total) respecto a las variantes consideradas. En la figura 3 se presentan como ejemplo los resultados referidos a la red neuronal que representa el desplazamiento máximo del último piso, para las siguientes situaciones:

- Caso 1 : NITES = 15, NITET = 5000
- Caso 2 : NITES = 50, NITET = 50000

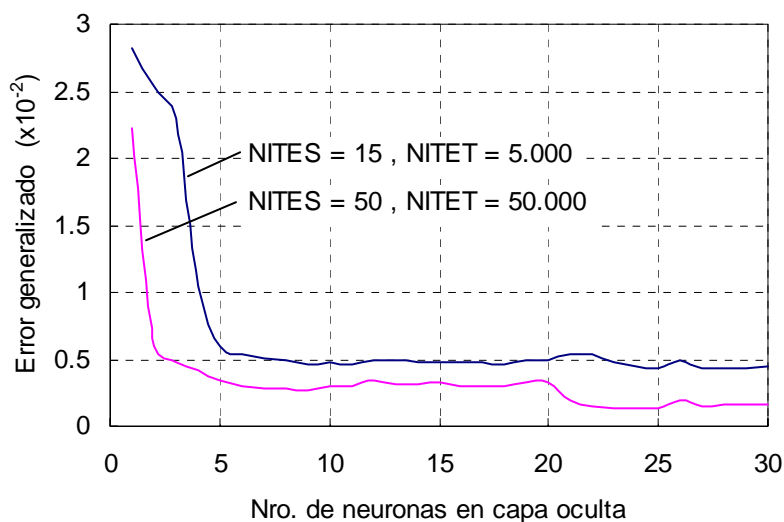


Figura 3: Variación del error generalizado

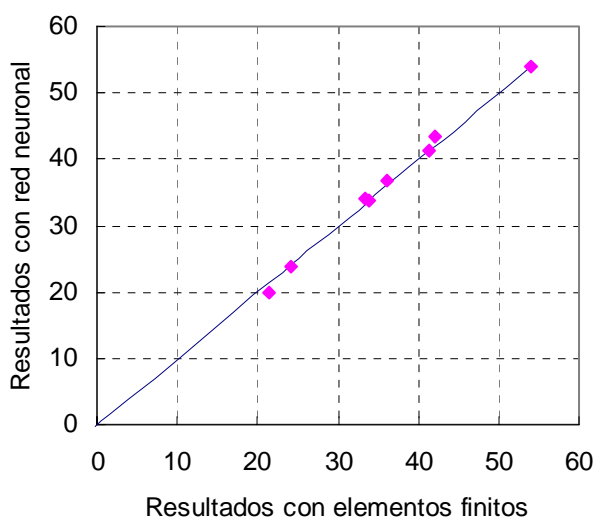
Se observa que es mucho más efectivo agregar mayor cantidad de iteraciones que hacer más grande el número máximo de neuronas, además que para mayor número de neuronas en la capa oculta se consume mucho más tiempo durante el entrenamiento. Es decir que aumentando el número máximo de neuronas de 20 a 30 no resulta en una significativa reducción del error, habiéndose obtenido en este caso un número óptimo de 25 neuronas.

Con las iteraciones y el número de neuronas máximo antes propuestas se obtienen errores generalizados muy bajos para esta aplicación, por lo que se produce una buena aproximación para todas las redes.

Para verificar la precisión de las redes neuronales ya entrenadas se presentan las siguientes comparaciones:

a) *Corte máximo en la base*: en la figura 4 se muestran los resultados calculados con análisis dinámico no lineal por elementos finitos y los obtenidos con la red neuronal entrenada para un número de neuronas máximo NEUMAX = 20 y con menor cantidad de iteraciones, es decir con NITES = 15 y NITET = 5000.

La red en este caso fue entrenada incluyendo valores objetivos nulos y el cálculo de errores relativos se realizó en el espacio de las variables originales, no calculando el error para el caso de valores objetivos nulos. El número óptimo de neuronas (NHID) resultó de 20 mientras que el error máximo generalizado de  $0.239 \times 10^{-2}$ .



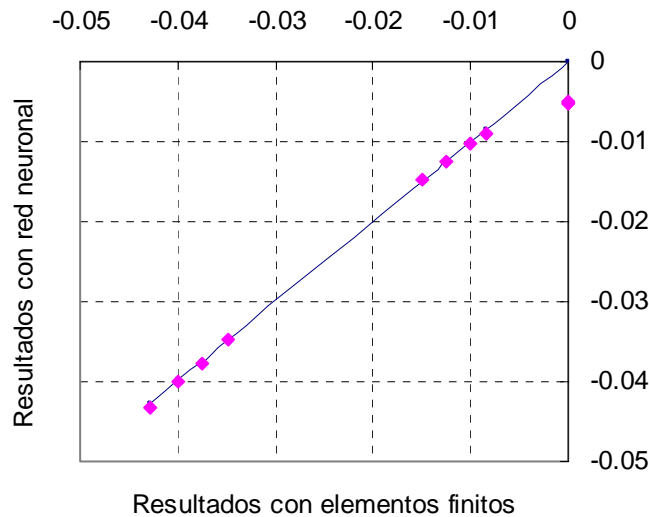
**Figura 4:** Precisión de la predicción con red neuronal para el corte máximo en la base

Se observa que la red aproxima satisfactoriamente con un error relativo máximo con respecto a resultados con elementos finitos de 6.25%. No se tomaron combinaciones para reacciones nulas.

Además al contar con pocas iteraciones el número óptimo de neuronas resultó igual al

máximo, por lo que se concluye que la red no fue optimizada al máximo.

b) *Distorsión máxima de piso*: en la figura 5 se muestran los resultados calculados con análisis dinámico no lineal por elementos finitos y los obtenidos con la red neuronal entrenada para un número de neuronas máximo NEUMAX = 30 y con menor cantidad de iteraciones, es decir con NITES = 15 y NITET = 5000.



**Figura 5:** Precisión de la predicción con red neuronal para la distorsión máxima de piso

No se consideraron valores objetivos nulos en la base de datos de entrenamiento, y el error relativo se calcula en variables transformadas, obteniéndose un número óptimo de neuronas de 20, mientras que el máximo error generalizado es  $0.302 \times 10^{-2}$ . El número óptimo de neuronas resultó menor que el máximo por lo que puede hablarse de una red bien entrenada.

Se observa además que al no ser entrenada con objetivos nulos no se obtienen buenos resultados para combinaciones de variables que den distorsiones nulas, es decir fuera del rango de actuación de la red.

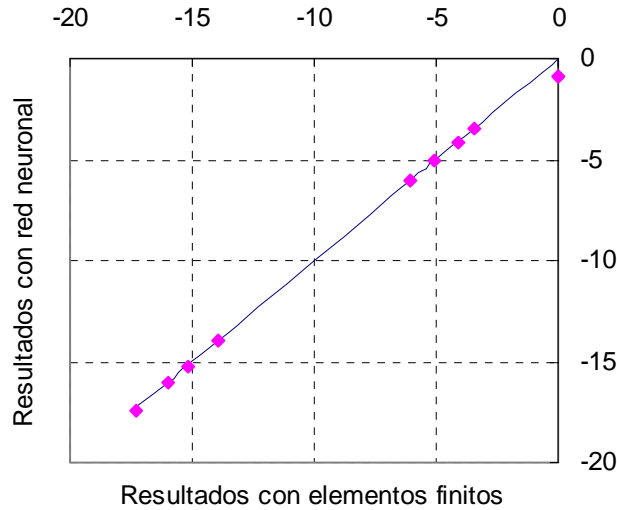
En este caso también se observa que la red aproxima muy bien los resultados calculados con elementos finitos, con un error relativo máximo de 8.01%.

c) *Desplazamiento máximo en el último piso*: en la figura 6 se muestran los resultados calculados con análisis dinámico no lineal por elementos finitos y los obtenidos con la red neuronal entrenada para un número de neuronas máximo NEUMAX = 30 y con mayor cantidad de iteraciones, es decir con NITES = 50 y NITET = 50000.

Para el entrenamiento de la red se incluyeron en este caso valores objetivos nulos, resultando un número óptimo de 24 neuronas y un error máximo generalizado igual a  $0.134 \times 10^{-2}$ .

El error relativo máximo entre las predicciones de la red y los resultados del análisis

dinámico no lineal por elementos finitos resultó sólo de 0.75%. Entre las muestras se contaron con combinaciones con desplazamientos nulos.



**Figura 6:** Precisión de predicción con red neuronal para el desplazamiento máximo del último piso

#### 4.4 Aplicación al cálculo de la probabilidad de falla

Como aplicación de las redes neuronales a la evaluación de la confiabilidad estructural, se calcula la probabilidad de excedencia de la distorsión de piso límite, si ocurre un terremoto.

La función de falla se expresa con

$$G(X) = RLIM - R(X) = 0.035 - DISTM(X) \quad (21)$$

siendo  $RLIM = 0.035$  la distorsión de piso límite y  $DISTM(X)$  la distorsión de piso máxima, respuesta estructural, función de las variables aleatorias básicas cuyos datos se presentan en la tabla 2.

**Tabla 2:** Datos de las variables aleatorias

Variable	Descripción	Tipo	$\bar{X}$	$\sigma_X$
$X(1)$	$m$ : masa por piso	Normal	37.5	3.75
$X(2)$	$k$ : rigidez	Normal	50.0	5.00
$X(3)$	$F_Y$ : resistencia	Normal	23.6	2.36
$X(4)$	$a_G$ : aceleración pico	Lognormal	384.2	117.3

Se utilizan las técnicas de simulación de Monte Carlo y de Importance Sampling. Para cada simulación, la evaluación de  $\text{DISTM}(\mathbf{X})$  se realiza con la red neuronal correspondiente. Los resultados de probabilidad de falla, índice de confiabilidad y tiempo de cálculo en PC se indican en la tabla 3.

**Tabla 3:** Resultados

Método	Nro. Simulac.	$P_f$	$\beta$	Tiempo
Monte Carlo	N = 1000000	$0.15312 \cdot 10^{-1}$	2.162	36 seg.
Importance Sampling	N = 2000	$0.17338 \cdot 10^{-1}$	2.112	1 seg.

Cada análisis dinámico no lineal completo por el método de elementos finitos tiene una duración promedio de 0.73 seg. Luego, si en cada simulación se haría este tipo de análisis, el tiempo total estimado sería de 730000 seg., es decir 8.45 días para Monte Carlo con N = 1000000, y de 1460 seg. para Importance Sampling con N = 2000.

Se observa así la enorme ventaja de usar redes neuronales como sustituto del análisis por elementos finitos.

## 5 CONCLUSIONES

- En el análisis de la confiabilidad estructural es muy conveniente aproximar la respuesta del sistema con un sustituto del análisis completo por elementos finitos. Se estudió la aplicación de teorías de aprendizaje estadístico y en particular el algoritmo de redes neuronales.
- Esta técnica permite establecer la relación funcional no lineal entre la respuesta del sistema y las variables de entrada (datos) mediante simples operaciones matemáticas. Para ello se optimiza la arquitectura de la red y los valores de ciertos factores de peso a través de métodos iterativos basados en el descenso del gradiente, que minimizan la diferencia entre los resultados del análisis estructural y los obtenidos con la red.
- Se aplicó a una estructura de 3 pisos, de comportamiento no lineal, con acción dinámica de terremotos. Se observa que es más efectivo aumentar el número de iteraciones en la optimización de los factores de peso que aumentar el número de neuronas en la capa intermedia u oculta.
- En la verificación de las redes neuronales sobre casos que no participaron de su entrenamiento, y dentro del rango de las variables, se observa un excelente acuerdo con los resultados calculados por elementos finitos.
- En la evaluación de la probabilidad de falla se muestra que es posible aplicar el método de Monte Carlo con un millón de simulaciones, utilizando en cada simulación la red neuronal para aproximar la respuesta.

## 6 REFERENCIAS

- [1] SEAOC Vision 2000 Committee, “Performance based seismic engineering of buildings”. *Structural Engineers Association of California, Sacramento, California, U.S.A.*, (1995).
- [2] FEMA “NEHRP Guidelines for the seismic rehabilitation of buildings”, Report 273, *Buildings Seismic Safety Council* (1997).
- [3] Melchers,R.E., *Structural Reliability - Analysis and Prediction*, John Wiley and Sons (1987).
- [4] Thoft Christensen,P., Baker,M.J., *Structural Reliability - Theory and Applications*, Springer Verlag (1982).
- [5] Hurtado, J., *Structural Reliability – Statistical Learning Perspectives*, Lecture Notes in Applied and Computational Mechanics, Vol.17, Springer Verlag (2004).
- [6] Möller, O., “Metodología para evaluación de la probabilidad de falla de estructuras sismorresistentes y calibración de códigos”, *Tesis de Doctorado en Ingeniería*, Universidad Nacional de Rosario (2001).
- [7] Möller, O., Foschi, R. “Reliability evaluation in seismic design: a response surface methodology”, *Earthquake Spectra*, **19** (3), 579-603 (2003).
- [8] Möller, O., Rubinstein, M. “Confiabilidad de sistemas estructurales bajo acciones sísmicas”, *Mecánica Computacional*, AMCA, Vol. XXIII, 499-514 (2004).
- [9] Zhang, J. “Performance-based seismic design using designed experiments and neural networks”, *PhD. Thesis*, Department of Civil Engineering, University of British Columbia, Canada (2003).
- [10] Foschi, R., Li, H., Zhang, J. “Reliability and performance-based design: a computational approach and applications”, *Structural safety*, **24**, 205-218 (2002).
- [11] Zhang, J., Foschi, R.O., “Performance-based design and seismic reliability analysis using designed experiment and neural networks”, *Probabilistic Engineering Mechanics*, **19**, 259-267 (2004).

## 7 AGRADECIMIENTOS

Este trabajo fue realizado con los aportes de los Proyectos de Investigación “Análisis dinámico, confiabilidad y control de sistemas estructurales”, ING 40, de la Universidad Nacional de Rosario, Argentina, y “Natural networks for reliability and performance-based design in earthquake engineering”, Natural Science Engineering Research Council: RGPIN 5882-04, University of British Columbia, Vancouver, Canada.