

NUEVAS ESTRATEGIAS DE ENTRENAMIENTO DE REDES NEURONALES Y COMPARACIÓN CON MÉTODOS ESTÁNDAR

Oscar Möller^a, Ricardo O. Foschi^b, Marcelo Rubinstein^a y Laura Quiroz^b

^a*Instituto de Mecánica Aplicada y Estructuras (IMAE), Universidad Nacional de Rosario, Riobamba y Beruti, 2000 Rosario, Argentina, moller@fceia.unr.edu.ar*

^b*Civil Engineering Department, University of British Columbia, 6250 Applied Sciences Lane, Vancouver, B.C., Canada V6T 1Z4, rowfal@civil.ubc.ca*

Palabras clave: Confiabilidad estructural, Redes neuronales, Entrenamiento, Optimización.

Resumen. En el análisis de la confiabilidad de sistemas estructurales no lineales con acciones dinámicas es necesario aplicar métodos de simulación para evaluar la probabilidad de falla, con técnicas de Monte Carlo, para varios niveles de desempeño de la estructura. Resulta conveniente aproximar la respuesta utilizando redes neuronales, las cuales deben ser “entrenadas” para minimizar las diferencias entre las predicciones de la red y los resultados del análisis estructural. Se propone una metodología de entrenamiento basada en técnicas de optimización mediante búsqueda aleatoria, evaluando el vector gradiente solo en la etapa de ajuste final. Los resultados muestran que es eficiente al compararla con el método estándar de backpropagation y con herramientas de software comercial.

1 INTRODUCCIÓN

La utilización de redes neuronales tiene importantes ventajas en el análisis de la probabilidad de falla o confiabilidad de estructuras solicitadas por acciones dinámicas.

En el caso de estructuras con acciones dinámicas aleatorias, por ejemplo acción de terremotos, es necesario considerar las incertidumbres presentes tanto en la demanda como en la capacidad del sistema. Se requiere el uso de métodos probabilísticos para evaluar la probabilidad de “falla” o de excedencia anual en cada nivel de desempeño considerado, dentro del marco del diseño sísmico basado en el desempeño (SEAOC Vision 2000, 1995; FEMA 273, 1997).

La formulación de los estados límites en cada nivel de desempeño requiere el cálculo de valores máximos de respuestas de la estructura en la duración del evento sísmico. Como el comportamiento estructural es fuertemente no lineal, es necesario recurrir al análisis dinámico no lineal paso a paso para obtener los parámetros de respuesta, con una importante demanda computacional que hace impracticable la estimación de la confiabilidad por simulación directa de Monte Carlo.

Además, no es posible establecer una relación explícita entre las variables de entrada (datos) y las respuestas dinámicas, sin embargo se pueden calcular resultados discretos para combinaciones específicas de las variables aleatorias y parámetros de diseño intervinientes. Estos resultados discretos se aproximan luego por una superficie de respuesta continua y explícita apropiadamente ajustada, que se utiliza como “sustituto” de la verdadera respuesta (Hurtado, 2004).

Del análisis de diferentes tipos de sustitutos (Möller et al., 2009a), las redes neuronales presentan una mejor aproximación, son más flexibles y adaptables, y en consecuencia son las de mayor utilización actual. Esta función explícita que aproxima la verdadera respuesta permite evaluar la probabilidad de falla por simulación de Monte Carlo con tiempos de computación razonables, como así también llevar adelante procesos de optimización con restricciones de confiabilidad (Möller et al., 2009b, c).

Las redes neuronales deben ser previamente “entrenadas”, que significa calcular los factores de peso que hagan mínima la diferencia entre los resultados disponibles del análisis dinámico no lineal y las predicciones de la red. La técnica de entrenamiento habitualmente utilizada es la denominada “backpropagation”, ver por ejemplo Zhang and Foschi (2004), técnica que aquí se denomina “estándar”.

En este trabajo se proponen nuevas técnicas de entrenamiento basadas en una selección aleatoria inicial, seguida de un proceso de optimización sin cálculo de gradientes, y combinado con un ajuste final utilizando recién en este paso el gradiente o el gradiente más la Hessiana, con variante de utilizar una matriz Hessiana simplificada.

Se presentan ejemplos de aplicación comparando los resultados obtenidos con la metodología propuesta, con el método estándar y con herramientas incorporadas en el software comercial Matlab.

2 FORMULACIÓN DE REDES NEURONALES

Las redes neuronales artificiales son dispositivos computacionales compuestos por una cantidad de unidades de procesamiento interconectadas. Cada una de ellas recibe información, realiza un cálculo simple y emite un resultado. La red en conjunto tiene la habilidad de

responder frente a “estímulos de entrada” (datos) y producir las respuestas correspondientes, además puede adaptarse a diferentes situaciones “aprendiendo de la experiencia” (entrenamiento). La expresión matemática genérica es

$$R(\mathbf{X}) \cong F(\mathbf{X}) = h \left(\sum_{j=1}^J W_{kj} h \left(\sum_{i=1}^N W_{ji} X_i + W_{j0} \right) + W_{k0} \right) \quad (1)$$

donde $R(\mathbf{X})$ es el “verdadero” valor de la respuesta obtenido con análisis dinámico no lineal para las variables X_i , $F(\mathbf{X})$ la aproximación con red neuronal, W_{kj} y W_{ji} son factores de peso a calcular, y $h(t)$ es la función no lineal de transferencia, que en este trabajo es la sigmoide:

$$h(t) = \frac{1.0}{(1 + \exp(-t))} \quad (2)$$

La arquitectura de la red es representada esquemáticamente en la fig.1, mostrando una capa de entrada con N neuronas conectadas a J neuronas de una capa intermedia. La información que recibe cada neurona j está formada por la contribución de cada una de las neuronas i de la capa de entrada, con valores X_i y los pesos W_{ji} , más un término independiente con W_{j0} . Esa información es procesada con la función de transferencia h , y el resultado es transmitido a la capa siguiente, de salida, utilizando los pesos W_{kj} más el término independiente con W_{k0} . La información que así llega a las neuronas de la capa de salida son procesadas nuevamente con la función h para producir los resultados finales en las neuronas K .

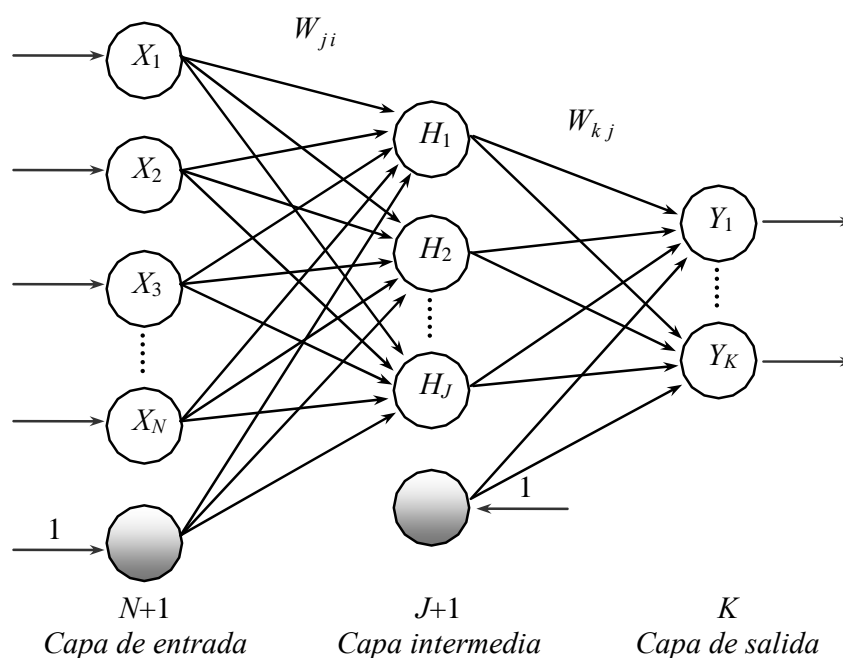


Figura 1: Red neuronal multicapa

En este trabajo se utilizan redes neuronales con una capa intermedia y con una sola neurona en la capa de salida ($Y = F(\mathbf{X})$). El número de neuronas en la capa intermedia puede variar desde una a un máximo controlado por el número de variables de entrada y el número de datos. Dada esa información el número óptimo de neuronas es calculado durante el entrenamiento.

El proceso de entrenamiento consiste en encontrar los pesos W que minimicen el error total entre las predicciones de la red y los resultados del análisis dinámico no lineal para diferentes combinaciones de las variables de entrada X_i .

3 MÉTODO DE ENTRENAMIENTO ESTÁNDAR: BACKPROPAGATION

Existen relaciones empíricas entre el número óptimo de neuronas de la capa intermedia u oculta y el número de variables de entrada, pero en general 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 entrenamiento.

El número máximo de neuronas, para poder realizar el cálculo de los factores de peso, está en función del número de variables de entrada N y del número de conjunto de datos NP

$$NEUMAX = \frac{0.8 NP - 1}{N + 2} \quad (3)$$

El proceso para encontrar el número óptimo de neuronas de la capa intermedia consiste en evaluar el error generalizado para diferente cantidad de neuronas desde 1 a $NEUMAX$, y adoptar la de menor error.

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 .

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

$$E = \frac{1}{2} \sum_p \sum_k (Y_k^p - T_k^p)^2 \quad (4)$$

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

$$W_{kj}(g+1) = W_{kj}(g) + \Delta W_{kj}(g) = W_{kj}(g) - \eta \frac{\partial E}{\partial W_{kj}} \quad (5)$$

donde g es el número de la iteración y η es la tasa de entrenamiento. Para compactar las expresiones se define:

$$I_j^p = \sum_{i=1}^N W_{ji} X_i^p + W_{j0} \quad H_j^p = h(I_j^p) \quad (6)$$

$$I_k^p = \sum_{j=1}^J W_{kj} H_j^p + W_{k0} \quad Y_k^p = h(I_k^p) \quad (7)$$

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

$$\Delta W_{kj} = \eta \sum_p (Y_k^p - T_k^p) h'(\mathbf{I}_k^p) H_j^p \quad (8)$$

Llamando:

$$\delta_k^p = (Y_k^p - T_k^p) h'(\mathbf{I}_k^p) \quad (9)$$

Entonces (8) resulta:

$$\Delta W_{kj} = \eta \sum_p \delta_k^p H_j^p \quad (10)$$

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

$$W_{ji}(g+1) = W_{ji}(g) + \Delta W_{ji}(g) = W_{ji}(g) - \eta \frac{\partial E}{\partial W_{ji}} \quad (11)$$

Usando la regla de derivación en cadena

$$\Delta W_{ji} = \eta \sum_p \sum_k (Y_k^p - T_k^p) h'(\mathbf{I}_k^p) W_{kj} h'(\mathbf{I}_j^p) X_i^p = \eta \sum_p \delta_j^p X_i^p \quad (12)$$

donde:

$$\delta_j^p = h'(\mathbf{I}_j^p) \sum_k \delta_k^p W_{kj} \quad (13)$$

La tasa de entrenamiento η 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 trabajo se utilizó $\eta = 0.01$.

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.

$$\Delta W(g) = -\eta \nabla E(g) + \alpha \Delta W(g-1) \quad (14)$$

donde g es el número de la iteración y α es el denominado momento. En este trabajo se utilizó $\alpha = 0.05$.

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:

$$g_{lim} = NITET \sqrt{NHID} \quad (15)$$

siendo $NITET$ una constante y $NHID$ el número de neuronas de la capa intermedia. En este trabajo se utilizó $NITET = 5000$.

Una vez determinada la arquitectura óptima se procede al entrenamiento final 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.

$$\max E_{rel} = \max (Y_k^P - T_k^P) / T_k^P \quad (16)$$

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, igual a 50 para este trabajo.

4 MÉTODOS DE ENTRENAMIENTO PROPUESTOS

Se propone una metodología con el objetivo de lograr un procedimiento más rápido y preciso que el estándar descrito en el punto anterior. Consiste en un conjunto de pasos que se describen a continuación, los cuales a su vez pueden ser combinados de diferentes maneras, dando origen a variantes que serán comparadas desde el punto de vista de la eficiencia.

4.1 Selección aleatoria inicial

Los datos de entrada $X_0(i,k)$ con $i = 1, N$ número de variables y $k = 1, NP$ número de muestras o conjunto de datos disponibles, son normalizados a $X(i,k)$ entre 0.01 y 0.99 para eliminar el problema de unidades y diferentes órdenes de magnitud entre ellos. Similarmente los resultados disponibles del análisis estructural $T_0(k)$, valores objetivo, también son normalizados a $T(k)$ entre 0.01 y 0.99, teniendo en cuenta que la función $h(t)$ de ec.(2) produce resultados límites entre 0 y 1.

Se seleccionan aleatoriamente muestras de valores de los factores de peso W con distribución uniforme entre -5 y 5. Estos límites son suficientemente amplios como para cubrir un amplio rango en las predicciones $Y(k)$ de la red. De todas las muestras analizadas, 5000 en este trabajo, se retienen los factores de peso W_0 correspondientes al mínimo error

$$E = \sum_{k=1}^{NP} (Y(k) - T(k))^2 \quad (17)$$

4.2 Optimización sin utilizar gradientes

Los NP datos disponibles se ordenan en función del error absoluto $Y(k) - T(k)$. El 80% con mayores errores se utiliza para entrenamiento de la red y el 20% restante para validación.

Los factores de peso W_0 se utilizan como primer “anclaje”, se calcula el error E_0 sobre el conjunto de datos de entrenamiento, y se comienza un proceso de optimización que consiste en seleccionar aleatoriamente valores de W en las proximidades de W_0 dentro de un radio $0.01 W_0$, calculando el error E con (17) sobre los datos de entrenamiento. Cuando se encuentra un caso para el cual el error $E < E_0$, ese punto W se transforma en el nuevo anclaje W_0 y se comienza otra búsqueda MS , con un máximo de 500 en este trabajo.

Si en una cantidad M de selecciones aleatorias alrededor del “anclaje”, 1000 en este trabajo, no se encuentra un caso con $E < E_0$, o si se agotan las 500 búsquedas MS , se calcula el error con (17) para todos los NP datos, se vuelven a reordenar en función del error absoluto y se repite todo el proceso.

Se realizan $NREP$ repeticiones del proceso, 100 en este trabajo, reteniéndose la repetición

que tenga menor error calculado con (17) para todos los NP datos (entrenamiento más validación). Esa es la solución “óptima” con este procedimiento.

Observar que solamente se calculan la predicciones de la red $Y(k)$ con ec.(1), sin necesidad de evaluar las derivadas.

4.3 Ajuste final utilizando derivadas

Las predicciones de la red $Y(k)$ son función de los factores de peso W , ver ec.(1). El error E entre las predicciones de la red y los valores objetivo se calcula con ec.(17). Las alternativas para mejorar los pesos W a partir de un conjunto W_0 consideradas son:

a) *Buscar los pesos W que hagan nulo el error E*

Se desarrolla en serie de Taylor el error E alrededor de E_0 evaluado para los pesos W_0

$$E = E_0 + \mathbf{G}_0^T (\mathbf{W} - \mathbf{W}_0) \quad (18)$$

donde las componentes del vector gradiente, que se calculan numéricamente, son

$$G_{0_i} = \left(\frac{\partial E}{\partial W_i} \right)_{W_0} \quad (19)$$

Se busca el vector $W - W_0$ en la dirección del gradiente, con signo menos para ir hacia el mínimo

$$(\mathbf{W} - \mathbf{W}_0) = -\lambda \mathbf{G}_0 \quad (20)$$

Reemplazando en (18) se obtiene

$$E = E_0 + \mathbf{G}_0^T (-\lambda \mathbf{G}_0) = 0 \quad \rightarrow \quad \lambda = \frac{E_0}{\mathbf{G}_0^T \mathbf{G}_0} \quad (21)$$

Se calcula entonces un nuevo valor de W con

$$\mathbf{W} = \mathbf{W}_0 - \frac{E_0}{\mathbf{G}_0^T \mathbf{G}_0} \mathbf{G}_0 \quad (22)$$

El proceso se aplica iterativamente hasta que el error comienza a aumentar, sin necesidad de resolver ningún sistema de ecuaciones. Para mejorar la convergencia es conveniente avanzar con una fracción de λ , en este trabajo 0.05.

b) *Buscar los pesos W que hagan nulo el gradiente G del error*

En este caso se busca anular el gradiente, lo que está asociado a un valor mínimo del error, ya que en general las predicciones de la red no serán exactamente iguales a los valores objetivos. Desarrollando en serie de Taylor el gradiente G alrededor de G_0 evaluado para los pesos W_0 , resulta

$$\mathbf{G} = \mathbf{G}_0 + \mathbf{H}_0 (\mathbf{W} - \mathbf{W}_0) \quad (23)$$

donde las componentes de la matriz Hessiana \mathbf{H}_0 , que al igual que el gradiente se calculan numéricamente, son

$$H_{0ij} = \left(\frac{\partial^2 E}{\partial W_i \partial W_j} \right)_{W_0} \quad (24)$$

Se busca el vector $W - W_0$ en la dirección del gradiente, con signo menos para ir hacia el mínimo

$$(\mathbf{W} - \mathbf{W}_0) = -\lambda \mathbf{G}_0 \quad (25)$$

Reemplazando en (23) y operando se obtiene

$$\begin{aligned} \mathbf{G} &= \mathbf{G}_0 + \mathbf{H}_0 (-\lambda \mathbf{G}_0) = 0 \\ \mathbf{G}_0 &= \lambda \mathbf{H}_0 \mathbf{G}_0 \\ \mathbf{G}_0^T \mathbf{G}_0 &= \lambda \mathbf{G}_0^T \mathbf{H}_0 \mathbf{G}_0 \rightarrow \lambda = \frac{\mathbf{G}_0^T \mathbf{G}_0}{\mathbf{G}_0^T \mathbf{H}_0 \mathbf{G}_0} \end{aligned} \quad (26)$$

Se calcula entonces un nuevo valor de W con

$$\mathbf{W} = \mathbf{W}_0 - \frac{\mathbf{G}_0^T \mathbf{G}_0}{\mathbf{G}_0^T \mathbf{H}_0 \mathbf{G}_0} \mathbf{G}_0 \quad (27)$$

El proceso se aplica iterativamente hasta que el error comienza a aumentar, sin necesidad de resolver ningún sistema de ecuaciones. Para mejorar la convergencia y no “pasarse” del mínimo es conveniente avanzar con una fracción de λ , en este trabajo 0.05.

c) *Combinación de alternativas*

Se investigan las siguientes combinaciones

ME = 0: Solamente se realiza el proceso de optimización (OPT) descrito en el punto 4.2.

ME = 1: A partir del W_0 obtenido al finalizar cada “repetición” del OPT, se busca mejorar la predicción de la red utilizando el gradiente, como se describió en el apartado a) de este punto.

ME = 2: A partir del W_0 obtenido al finalizar cada “repetición” del OPT, se busca mejorar la predicción de la red utilizando el gradiente y la Hessiana, como se describió en el apartado b) de este punto.

ME = 3: A partir del W_0 obtenido al finalizar cada “repetición” del OPT, se busca mejorar la predicción de la red utilizando el gradiente y una Hessiana simplificada en la cual solo se calculan los términos de la diagonal principal.

5 APLICACIONES NUMÉRICAS

5.1 Datos de la estructura

Se analizan los resultados del entrenamiento de redes neuronales que representan parámetros de respuesta de un pórtico de un vano y un piso solicitado por cargas permanentes y acción sísmica, cuya geometría se presenta en la fig.2.

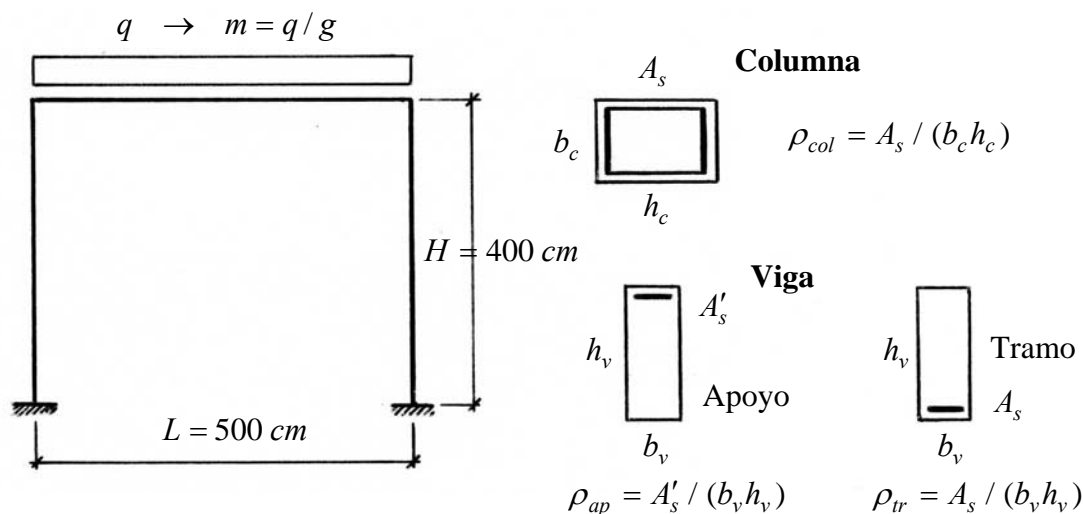


Figura 2: Geometría del pórtico

Se consideran como variables aleatorias la carga permanente (masa activa), las dimensiones y cuantías de acero de la viga y de las columnas, el confinamiento del hormigón, la aceleración máxima del suelo y la frecuencia central del filtro del suelo, estas dos últimas representando la aleatoriedad de la acción sísmica. Los límites inferior y superior de las variables, consideradas en este trabajo, se presentan en la tabla 1.

Variable	Descripción	Límite inferior	Límite superior
$X(1)$	m : masa repartida [$KN s^2 / cm^2$]	$0.215 \cdot 10^{-3}$	$0.714 \cdot 10^{-3}$
$X(2)$	b_v : ancho sección viga [cm]	15	30
$X(3)$	h_v : altura sección viga [cm]	40	70
$X(4)$	b_c : ancho sección columna [cm]	20	40
$X(5)$	h_c : altura sección columna [cm]	40	100
$X(6)$	ρ_{tr} : cuantía en tramo de viga	0.00298	0.01389
$X(7)$	ρ_{ap} : cuantía en apoyo de viga	0.00298	0.01389
$X(8)$	ρ_{col} : cuantía de columna	0.008	0.04286
$X(9)$	f_r / f'_{c0} : confinamiento hormigón	0	0.15
$X(10)$	a_G : aceleración máxima del suelo [cm/s^2]	25	1200
$X(11)$	f_g : frecuencia del suelo [Hz]	2	3

Tabla 1: Datos de las variables

5.2 Análisis dinámico no lineal

Aplicando la técnica de diseño de experimentos, ver Zhang (2003), se seleccionan 450 combinaciones de las variables dentro de los límites indicados en la tabla 1.

Para cada combinación se generan 10 sub-combinaciones teniendo en cuenta las siguientes variables aleatorias secundarias: (a) Los ángulos de fase de los acelerogramas artificiales generados como miembros de un proceso aleatorio, ver Möller (2001); (b) La resistencia del hormigón y del acero que generan variabilidad en los parámetros de las relaciones momento - curvatura de las secciones de la viga y columnas, ver Möller y otros (2006).

Se realizan los análisis dinámico no lineal con un modelo de elementos de barra para pórticos planos, ver Möller (2001), obteniéndose los siguientes parámetros de respuesta: desplazamiento horizontal máximo de piso $UMAX$, corte basal máximo $CORT$, aceleración máxima horizontal de piso $ACEL$, distorsión máxima de piso $DIST$, índice daño local máximo $DILO$, e índice de daño global de la estructura $DIES$.

Así, para cada uno de estos parámetros de respuesta R_i , en cada una de las 450 combinaciones, se calculan los valores medios y desvíos estándar, para $NS = 10$ casos correspondientes a las sub-combinaciones sobre los registros sísmicos y propiedades de histéresis, con:

$$\bar{R}_i = \frac{1}{NS} \sum_{k=1}^{NS} R_{ki} \quad \sigma_{R_i} = \sqrt{\frac{1}{NS-1} \sum_{k=1}^{NS} (R_{ki} - \bar{R}_i)^2} \quad (28)$$

Con estos resultados se ensamblan dos bases de 450 datos para cada parámetro de respuesta, que son usadas para entrenar dos redes neuronales: una para el valor medio y otra para el desvío estándar de la respuesta sobre las sub-combinaciones.

5.3 Análisis estático no lineal

Con el mismo modelo de elementos de barra para pórtico plano se realiza el push-over de cada una de las 450 combinaciones, para obtener el desplazamiento de fluencia del pórtico UY necesario para formular uno de los estados límites del nivel de desempeño operacional. Esa base de datos y resultados se utiliza para entrenar la red neuronal correspondiente que aproxime ese parámetro de respuesta estructural.

5.4 Probabilidad de falla. Índices de confiabilidad

En los niveles de desempeño “operacional”, “control de daños” y “límite de colapso” se analiza la probabilidad de excedencia para varios estados límites, ver los detalles en Möller y otros (2010). Estas probabilidades de falla son calculadas para 180 combinaciones de los parámetros de diseño, utilizando el método de Monte Carlo, ver Foschi et al.(2005).

Así, se obtienen las bases de datos y resultados de los índices de confiabilidad β_1 , β_2 , β_3 , con las cuales se entrenan otras tres redes neuronales.

5.5 Resultados del entrenamiento de las redes neuronales

Se entrenaron redes neuronales para aproximar cada parámetro de respuesta descrito en los puntos 5.2 a 5.4. Para ello se implementaron los algoritmos descritos en las secciones 3 y 4 en programas de cálculo numérico.

De las variantes consideradas en la metodología propuesta en este trabajo, se presentan los resultados solamente para el $ME = 1$, es decir que en cada “repetición”, luego de la optimización OPT , se mejora la solución con un ajuste utilizando el gradiente con la ec.(22).

Las variantes utilizando la matriz Hessiana no son más eficientes porque el tiempo de cálculo es bastante mayor y la precisión de los resultados prácticamente la misma. Se debe a

que con la optimización *OPT* ya se consigue un resultado muy cercano al que produce el error mínimo, y entonces con el uso del gradiente es suficiente para el ajuste “fino” final.

En la tabla 2 se presentan resultados del entrenamiento que permiten evaluar comparativamente la eficiencia entre el método clásico o estándar de backpropagation y la metodología propuesta. El prefijo *MV* indica valores medios, y el *SD* significa desvíos estándar. En las figs.3 a 5 se grafican algunas de las redes entrenadas con ambas metodologías y también los resultados obtenidos con las herramientas incorporadas en el software Matlab.

El grado de aproximación se analiza con el coeficiente de correlación lineal.

$$\rho_{YT} = \frac{\sigma_{YT}}{\sigma_Y \sigma_T} \quad \text{con} \quad \sigma_{YT} = \frac{1}{NP-1} \sum_{k=1}^{NP} (Y_k - \bar{Y})(T_k - \bar{T}) \quad (29)$$

Además, si el ajuste fuera perfecto, todos los resultados estarían sobre la recta a 45°. La dispersión se evalúa con el desvío estándar del error relativo, que es un coeficiente de variación promedio, dado por:

$$\sigma_{\varepsilon_r} = \sqrt{\frac{1}{NP-1} \sum_{k=1}^{NP} \left(\frac{Y_k - T_k}{Y_k} \right)^2} \quad (30)$$

donde Y_k es el valor calculado con la red, T_k es el valor obtenido con el análisis estructural y NP el número de casos analizados.

Parámetro	N	Backpropagation			Optimización $ME = 1$		
		ρ_{YT}	σ_{ε_r}	t [seg]	ρ_{YT}	σ_{ε_r}	t [seg]
<i>MVCORT</i>	11	0.9916	0.0766	8440	0.9928	0.1564	913
<i>SDCORT</i>	11	0.9768	0.1192	15700	0.9483	0.3023	895
<i>MVDILO</i>	11	0.9991	0.0851	6840	0.9949	0.5689	1054
<i>SDDILO</i>	11	0.9630	0.2265	15900	0.9767	0.6779	1010
<i>MVUMAX</i>	11	0.9971	0.0520	2980	0.9970	0.2130	842
<i>SDUMAX</i>	11	0.9949	0.0756	15900	0.9865	0.3517	746
<i>MVACEL</i>	11	0.9963	0.0441	5390	0.9952	0.1171	963
<i>SDACEL</i>	11	0.9882	0.0730	12500	0.9582	0.2269	1162
<i>MVDIES</i>	11	0.9920	0.1460	6830	0.9933	0.6106	949
<i>SDDIES</i>	11	0.9999	0.1589	8610	0.9997	0.6354	1050
<i>MVDIST</i>	11	0.9985	0.0443	3160	0.9959	0.2206	755
<i>SDDIST</i>	11	0.9948	0.1036	8600	0.9854	0.3559	1006
<i>UY</i>	9	0.9924	0.0226	8795	0.9869	0.3468	979
β_1	5	0.9996	0.0057	731	0.9991	0.0107	133
β_2	5	0.9987	0.0092	756	0.9920	0.0281	133
β_3	5	0.9918	0.0244	565	0.9875	0.0291	168

Tabla 2: Resultados del entrenamiento

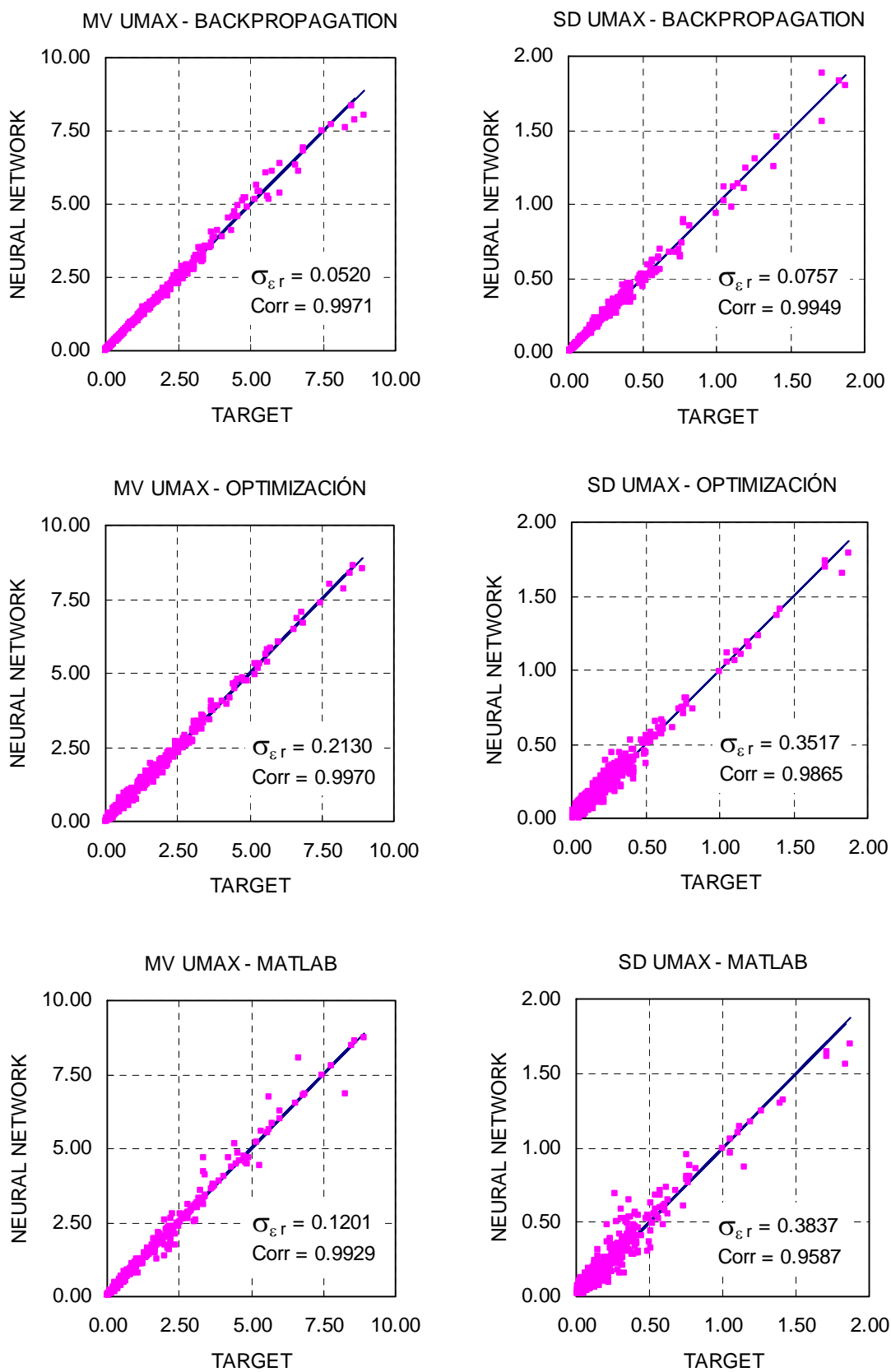


Figura 3: Resultados del entrenamiento para valor medio y desvío estándar de UMAX

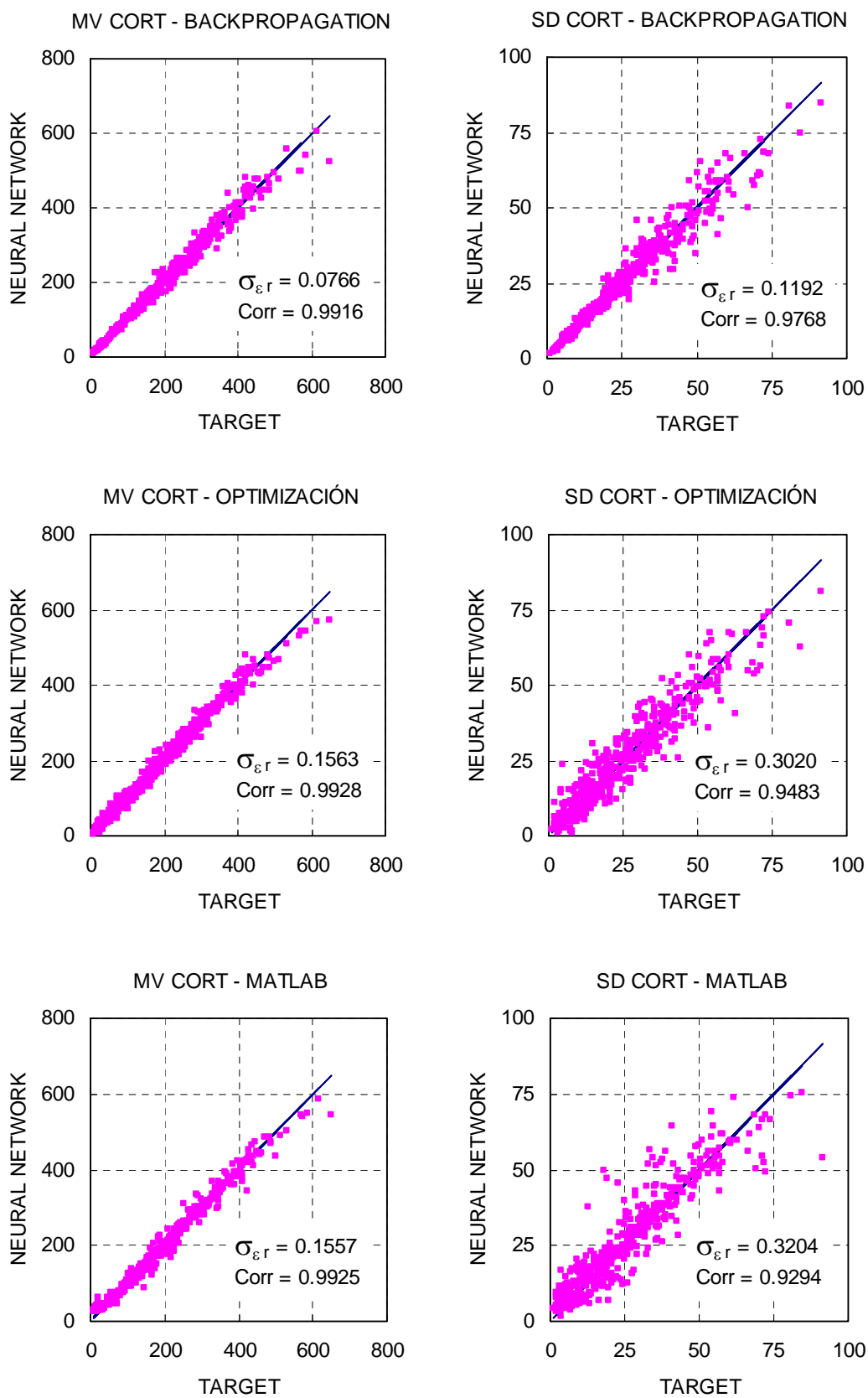


Figura 4: Resultados del entrenamiento para valor medio y desvío estándar de CORT

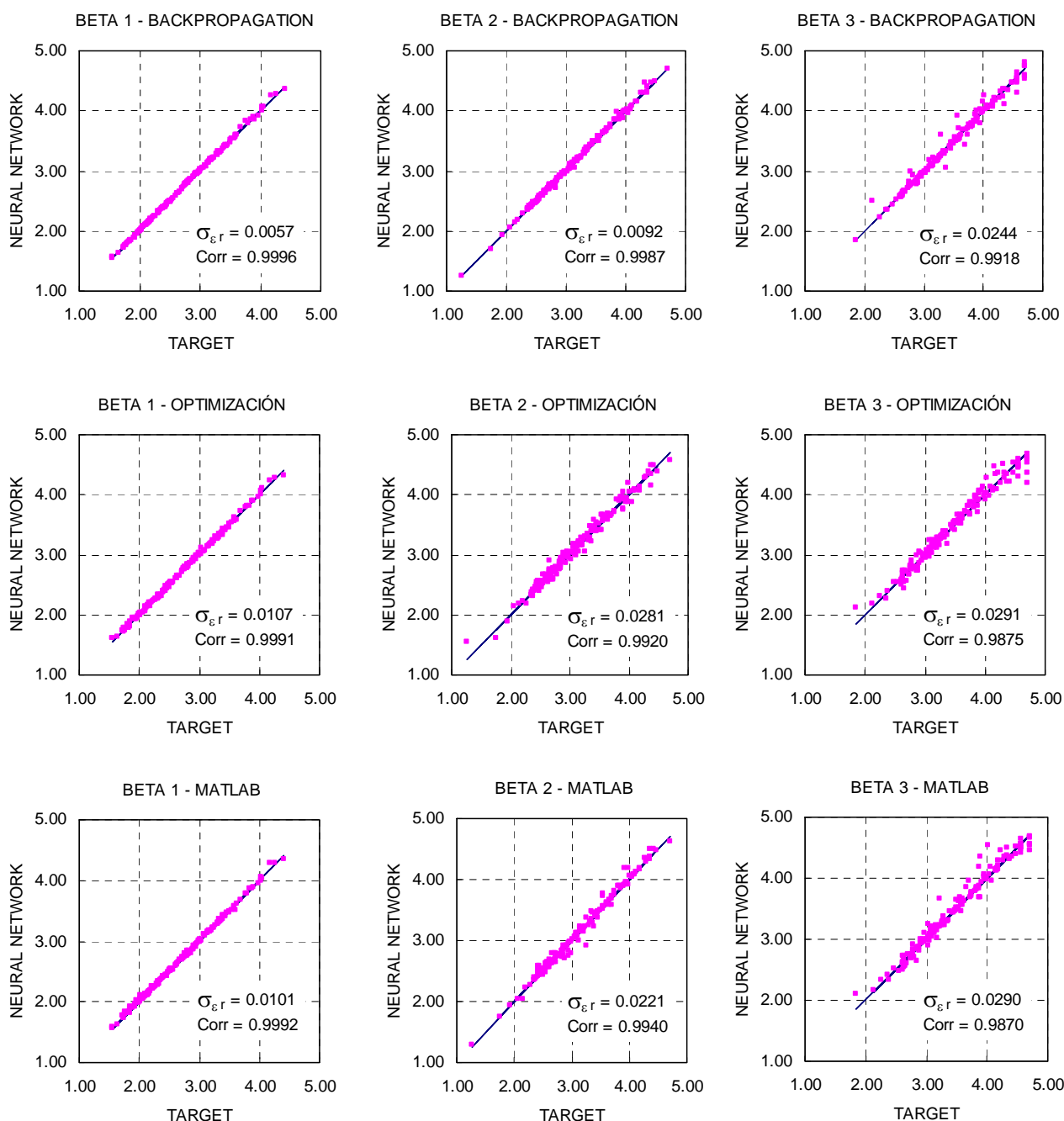


Figura 5: Resultados del entrenamiento para los índices de confiabilidad $\beta_1, \beta_2, \beta_3$

Como se observa en las figs.3 a 5 los resultados Y_k se distribuyen sin preferencias hacia un lado y otro de T_k , de forma tal que puede adoptarse una distribución normal. Luego el valor medio y el desvío estándar de los parámetros de respuesta del pórtico se pueden escribir con:

$$\bar{R}_i = \bar{Y}_i (1 + \sigma_{\epsilon_{ri}} X_N) \quad \sigma_{R_i} = \sigma_{Y_i} (1 + \sigma_{\epsilon_{ri}} X_N) \quad (31)$$

donde \bar{Y}_i, σ_{Y_i} son el valor medio y desvío estándar del parámetro i calculado con la red neuronal correspondiente, $\sigma_{\epsilon_{ri}}$ el desvío estándar del error relativo para el parámetro i (30), y X_N una variable aleatoria con distribución normal estándar (0,1). En la ec.(31) se muestra la

utilidad del desvío estándar del error relativo.

5.6 Análisis de resultados

Los coeficientes de correlación para todos los casos analizados son muy cercanos a la unidad (correlación perfecta), ver tabla 2, indicando que los tres procedimientos utilizados tienen la capacidad de entrenar satisfactoriamente las redes neuronales, es decir calcular los factores de peso, para obtener una suficiente aproximación en la representación de los parámetros de respuesta estructural.

En la mayoría de los casos se obtienen coeficientes de correlación ligeramente superiores con el método estándar de backpropagation, con una diferencia promedio de 0.4% respecto de la metodología de optimización propuesta, y una diferencia promedio de 1.4% con respecto a las herramientas de Matlab.

El desvío estándar del error relativo es mayor con el método de optimización que con backpropagation. Se debe a un mejor ajuste, es decir una menor dispersión, en los casos con resultados cercanos a cero, como se observa en las figs. 3 a 5. La mayor diferencia dividida por un valor pequeño produce un error relativo bastante superior. De todas maneras esta dispersión se tiene en cuenta, ec.(31), cuando se utilizan las redes neuronales como sustituto de la respuesta estructural en el cálculo de la probabilidad de falla por métodos de simulación como por ejemplo Monte Carlo. Los resultados de los casos también analizados con Matlab son muy similares a los del procedimiento de optimización.

El otro parámetro con que se evalúa la eficiencia del procedimiento de entrenamiento es el tiempo de cálculo. En la tabla 2 se observa claramente la ventaja del método propuesto de optimización, utilizando en todos los casos el número máximo de neuronas de ec.(3). En promedio sobre todos los casos estudiados, la relación de tiempos $OPT / BP = 0.15$. La importancia de este resultado se manifiesta en los procesos de optimización de estructuras, cuando en la búsqueda de la estructura de mínimo costo es necesario ampliar los límites de las variables y re-entrenar redes neuronales considerando resultados en la nueva zona agregada.

6 CONCLUSIONES

Para realizar evaluaciones de la confiabilidad de estructuras por métodos de simulación como Monte Carlo, es conveniente y necesario utilizar “sustitutos”, redes neuronales, que aproximen los parámetros de respuesta de la estructura.

Las redes neuronales tienen factores de peso que se deben calcular para minimizar la diferencia entre las predicciones de la red y resultados disponibles del análisis estructural para algunas combinaciones de las variables intervinientes.

Este proceso de “entrenamiento” es tradicionalmente realizado por el método estándar de backpropagation. En este trabajo se presenta otra metodología basada en técnicas de optimización.

La metodología propuesta consiste en una selección aleatoria inicial, seguida de un proceso iterativo de búsqueda del mínimo error calculando solamente las predicciones de la red, sin utilizar derivadas, y un ajuste final donde recién ahí se evalúa el vector gradiente.

Los ejemplos numéricos analizados muestran que

- La metodología propuesta es efectiva para entrenar redes neuronales para aproximar los parámetros de respuesta estructural.

- Los coeficientes de correlación son muy cercanos a la unidad (perfecta correlación) con valores numéricos solo 0.4% menores en promedio con respecto al método estándar, y 1.4% mayores con respecto a las herramientas de Matlab.
- El desvío estándar del error relativo es mayor con respecto al backpropagation, debido a mayores diferencias en los casos de resultados cercanos a cero. De todas maneras esa dispersión es tenida en cuenta en los cálculos probabilísticos.
- El tiempo de cálculo computacional es significativamente menor con respecto a la metodología estándar.

REFERENCIAS

- FEMA, NEHRP Guidelines for the seismic rehabilitation of buildings, Report 273, *Buildings Seismic Safety Council*, 1997.
- Foschi RO, Li H, Zhang J, Folz B, Yao F. Software RELAN: Reliability analysis. *Department of Civil Engineering, University of British Columbia*, Vancouver, Canada, 2005.
- Hurtado,J., *Structural Reliability – Statistical Learning Perspectives*, Lecture Notes in Applied and Computational Mechanics, Vol.17, Springer Verlag, 2004.
- 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.
- Möller,O., Foschi,R., Rubinstein,M., Quiroz,L., Momento-curvatura de secciones de hormigón armado sismorresistentes utilizando redes neuronales. *Mecánica Computacional* Vol. XXV, 2145-2162, AMCA, 2006.
- Möller,O., Foschi,R., Rubinstein,M., Quiroz,L. Seismic structural reliability using different nonlinear dynamic response surface approximations, *Structural Safety*, Vol.31, N°5, pp.432-442, ISSN 0167-4730, 2009.
- Möller,O., Foschi,R., Quiroz,L., Rubinstein,M. Performance-based seismic optimization implementing neural networks *Computational Structural Dynamics and Earthquake Engineering, Structures and Infrastructures Series*, Vol.II, Editores: Papadrakakis,M., Charnpis,D., Lagaros,N., Tsompanakis,Y. Series Editor: Frangopol,D.M. - Chapter 34, pp 547-564. Taylor & Francis Group - CRC Press – Balkema. ISBN 978-0-415-45261-8, 2009.
- Möller,O., Foschi,R., Quiroz,L., Rubinstein,M. Structural optimization for performance-based design in earthquake engineering: Applications of neural networks, *Structural Safety*, Vol.31, N°6, pp.490-499, ISSN: 0167-4730, 2009.
- Möller,O., Foschi,R., Rubinstein,M. Optimización, con requisitos de confiabilidad, a partir del diseño preliminar de pórticos sismorresistentes. *Mecánica Computacional* Vol. XXIX, en prensa, AMCA, 2010.
- SEAOC Vision 2000 Committee. Performance based seismic engineering of buildings. *Structural Engineers Association of California, Sacramento, California, U.S.A.*, 1995.
- Zhang,J. Performance-based seismic design using designed experiments and neural networks, *PhD. Thesis, Department of Civil Engineering, University of British Columbia*, Canada, 2003.
- 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.

AGRADECIMIENTOS

Este trabajo fue realizado con los aportes de los Proyectos de Investigación “Diseño y optimización de estructuras bajo acciones dinámicas aleatorias” 1ING 300 de la Universidad Nacional de Rosario, Argentina, y “Neural networks for reliability and performance-based design in earthquake engineering”, RGPIN 5882-04, University of British Columbia, Vancouver, Canada, supported by Canada’s Natural Science and Engineering Research Council.