



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA**

FACULTAD DE INGENIERÍA

**Uso de redes neuronales artificiales para estimar la
respuesta sísmica de sistemas estructurales**

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

INGENIERÍA CIVIL – ESTRUCTURAS

P R E S E N T A :

JUAN BOJORQUEZ MORA

TUTOR:

DRA. SONIA ELDA RUIZ GÓMEZ



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente:	Dr. Luis Esteva Maraboto
Secretario:	Dr. José Alberto Escobar Sánchez
Vocal:	Dra. Sonia Elda Ruiz Gómez
1 ^{er} . Suplente:	Dr. Jaime García Pérez
2 ^{do} . Suplente:	Dr. Juan José Pérez Gavilán Escalante

Lugar donde se realizó la tesis:

INSTITUTO DE INGENIERÍA, UNAM.

TUTOR DE TESIS:

DRA. SONIA ELDA RUIZ GÓMEZ

FIRMA

AGRADECIMIENTOS

Gracias a Dios por haberme puesto en este lugar llamado tierra y que por su poder me concedió este periodo de tiempo llamado vida a través de mis maravillosos padres Wildo y Ana Luisa. A quienes les debo tanto y han sabido guiarme por el buen camino de la vida. Gracias a su gran cariño, esfuerzo, dedicación, paciencia y bondad me es posible lograr una meta más en mi sendero.

A la persona que está detrás de esta investigación y que sin su apoyo no hubiese podido realizar este trabajo; a la Dra. Sonia E. Ruíz Gómez por su paciencia, amistad, consejos y apoyo brindados durante esta etapa de mi vida. Con su gran esfuerzo de todos los días motiva a las nuevas generaciones a superarse y llegar más lejos, es una grandiosa persona a quien respeto y admiro.

A los mejores hermanos que dios me dio Wildo, Edén, Omar y Wilnya que tanto quiero y aprecio, de quienes he aprendido tantas cosas y seguiré aprendiendo y en quienes encontré la motivación necesaria para llegar a este punto en mi vida y mucho más allá. Sé que si no fuera por su gran apoyo hoy no podría estar cumpliendo este sueño tan anhelado.

A mi abuelita que con su gran experiencia en la vida siempre está ahí para darme consejos y siempre esta apoyándome con lo que puede.

A la Universidad Nacional Autónoma de México y al Instituto de Ingeniería por brindarme un espacio de trabajo para concluir con estos estudios.

A CONACYT por brindarme el apoyo económico para poder realizar mis metas.

A los sinodales asignados por sus valiosos comentarios que enriquecieron este trabajo de tesis.

A mis compañeros de departamento Andrés José y Tirso, a todos mis amigos Danaé, Lorena, Paulina, Nathalie, Jonathan, Antonio, Dante, Tomás, David con quienes he compartido muchas experiencias, sé que encontraran su camino en la vida y llegaran muy lejos.

Índice

ÍNDICE

INTRODUCCIÓN

1. Capítulo 1. Introducción a la Redes Neuronales Artificiales	1
1.1. El modelo biológico	2
1.1.1. Aprendizaje	3
1.2. Estructura de un sistema neuronal artificial	3
1.3. Sistemas paralelos, distribuidos y adaptativos	5
1.4. El modelo de una Neurona Artificial	5
1.4.1. Modelo general de neurona artificial.....	6
1.4.2. Modelo estándar de neurona artificial.....	10
1.5. Tipos de neuronas.....	12
1.5.1. Neuronas Todo-Nada (dispositivos de tipo umbral.....	12
1.5.2. Neurona sigmoidea.....	12
1.5.3. Neurona estocástica	12
1.6. Arquitectura de una red neuronal artificial.....	13
1.7. Tipos de redes neuronales	14
1.7.1. Red neuronal monocapa.....	14
1.7.2. Red neuronal multicapa.....	14
1.7.3. Red neuronal recurrente	15
1.8. Definición formal de red neuronal	15
1.9. Modos de operación: recuerdo y aprendizaje.....	16
1.9.1. Fase de aprendizaje	16
1.9.1.1. Aprendizaje supervisado	17
1.9.1.2. Aprendizaje no supervisado o auto organizado.....	17
1.9.1.3. Aprendizaje híbrido.....	17
1.9.1.4. Aprendizaje reforzado	17
1.9.2. Fase de recuerdo (ejecución).....	18
1.10. Clasificación de los modelos de redes neuronales artificiales.....	18
1.10.1. Redes neuronales con conexión hacia adelante.....	19
1.10.1.1. Perceptrón.....	19
1.10.1.2. Adalina / Madalina	20
1.10.1.3. Perceptrón multicapa.....	20
1.11. Entrenamiento	21

Índice

1.11.1.	Para qué entrenar.....	21
1.11.2.	Cómo entrenar.....	21
1.12.	El algoritmo “Backpropagation”.....	22
1.12.1.	Deducción del algoritmo “Backpropagation”.....	22
1.13.	Modos de entrenamiento.....	26
1.13.1.	Modo secuencial.....	26
1.13.2.	Modo “batch”.....	27
1.14.	Aceleración del aprendizaje.....	27
1.15.	Generalización.....	28
2.	Capítulo 2. Primera Aplicación de RNA: Factores de Transformación.....	30
2.1.	Introducción.....	30
2.2.	Factores de transformación entre SMGL y SIGL con misma tasa anual de falla.....	30
2.2.1.	Objetivo de este capítulo.....	31
2.2.2.	Modelos de marcos estructurales.....	31
2.2.3.	Movimientos Sísmicos.....	33
2.2.3.1.	Escalamiento de sismos.....	34
2.2.4.	Curvas de peligro de demanda estructural.....	35
2.2.4.1.	Demandas de ductilidad global en función de la intensidad sísmica.....	35
2.2.4.2.	Curvas de vulnerabilidad sísmica.....	37
2.2.4.3.	Curvas de peligro de demanda de ductilidad global.....	38
2.2.4.4.	Curvas de peligro de distorsión máxima de entrepiso.....	39
2.2.4.5.	Curvas de peligro de demanda de ductilidad global y distorsión máxima de entre piso para SIGL equivalentes.....	39
2.2.5.	Factores de transformación.....	40
2.2.5.1.	Factores de transformación de ductilidad, FT_{μ}	40
2.2.5.2.	Factores de transformación de distorsiones máximas de entrepiso, FT_{γ}	41
2.3.	Funciones de transformación utilizando redes neuronales artificiales.....	42
2.3.1.	Modelo neuronal artificial.....	43
2.3.1.1.	Determinación de las entradas y salidas necesarias de la RNA.....	43
2.3.1.2.	Entrenamiento del modelo de RNA.....	43
2.3.1.3.	Arquitecturas elaboradas.....	46
2.3.2.	Resultados del entrenamiento y pruebas.....	47

Índice

2.4.	Metodología propuesta para encontrar las funciones de factores transformación.....	48
3.	Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales.....	52
3.1.	Introducción	52
3.2.	Registros utilizados	53
3.2.1.	Mecanismo de falla	55
3.2.2.	Tipo de suelo	56
3.3.	Espectros de respuesta en función de su ductilidad	56
3.4.	Modelo Neuronal Artificial.....	60
3.5.	Determinación de las entradas y salidas necesarias de la RNA	60
3.6.	Arquitecturas elaboradas	61
3.7.	Entrenamiento del modelo de RNA	62
3.8.	Resultados del entrenamiento.....	63
3.9.	Resultados de las pruebas.....	67
4.	Conclusiones.....	71

Bibliografía

INTRODUCCIÓN

El objetivo de este trabajo es introducir el concepto y herramientas básicas sobre Redes Neuronales Artificiales (RNA) para que cualquier persona interesada en conocer esta teoría y quiera aplicarla a la solución de un problema específico, esté en capacidad de decidir si este método es el mejor camino, y en tal caso, tenga las herramientas necesarias para escoger el tipo de Red Neuronal Artificial que permita dar solución a su interés particular.

Dando prioridad a las aplicaciones que pueda tener en el campo de la Ingeniería Civil, esta tesis se ha dividido en tres capítulos que progresivamente llevarán al lector a tener una idea clara de lo que es la teoría de las redes neuronales y cuáles son los requisitos para aplicarla.

En el capítulo 1 se encontrará un bosquejo general sobre el desarrollo de la teoría de las redes neuronales artificiales desde su nacimiento hasta los últimos aportes que se han hecho; se encontrará también un resumen especializado acerca del funcionamiento del cerebro en general y de cómo ocurre el proceso neuronal, para terminar haciendo la analogía que lleva del proceso biológico al modelo artificial. El primer capítulo es fundamental para el entendimiento de Redes Neuronales Artificiales, pues el lector descubrirá allí la importancia de esta teoría, además entrará al ambiente matemático y gráfico que le permitirá comprender con facilidad las secciones posteriores.

Los siguientes capítulos (2 y 3) son dedicados a aplicar los conocimientos adquiridos en el capítulo 1, específicamente a problemas de Ingeniería, cuya solución involucra procesos complejos que son enormemente simplificados al resolverse utilizando el método de las Redes Neuronales Artificiales. En el capítulo 2 se aplicarán las RNA para encontrar factores de transformación entre respuestas máximas, como son: distorsiones máximas de entrepiso y ductilidades globales, entre sistemas de múltiples grados de libertad (SMGL) y sus correspondientes sistemas de un grado de libertad (S1GL). En el capítulo 3 se propone un modelo de RNA para estimar espectros de respuesta inelásticos en función de: magnitud del registro M, mecanismo de falla, distancia de Joyner y Boore (DJB), velocidad de ondas de cortante Vs, periodo fundamental de la estructura To y ductilidad.

Finalmente, es oportuno señalar las ventajas y bondades de las RNA para la solución de los problemas propuestos, las cuales se mencionan en las Conclusiones.

Capitulo 1. Introducción a las Redes Neuronales Artificiales

La búsqueda interminable del hombre por encontrar nuevos caminos que faciliten su existencia, ha permitido el descubrimiento y desarrollo de máquinas calculadoras que ayuden a resolver de forma automática y rápida determinadas operaciones que resultan tediosas cuando se resuelven “a mano”.

Retornando un poco al pasado se puede considerar que el camino hacia la construcción de máquinas inteligentes inicia en la Segunda Guerra Mundial, cuando ya se disponía de los elementos electrónicos necesarios para construirlas. En 1946 se construyó la primera computadora electrónica, denominada ENIAC. Desde entonces los desarrollos en este campo han tenido un auge espectacular.

Hoy, el uso de las computadoras se ha convertido en parte integral de las actividades de los seres humanos; también en la Ingeniería Civil se emplean diferentes aplicaciones computacionales que ayudan a resolver problemas complejos. Aunque las computadoras se usan para modelar una gran variedad de problemas de Ingeniería, su potencial está siendo desperdiciado. La velocidad de procesamiento del cerebro es casi un millón de veces menor que la de las computadoras actuales, por lo que surge la pregunta ¿por qué pese al espectacular desarrollo de la electrónica e informática existen tareas que nuestras máquinas no ha podido resolver? particularmente tareas donde el problema no tenga una solución secuencial (programación tradicional) o que la información sea borrosa o incompleta, o simplemente que sea tanta la información de procesamiento que la computadora tardara miles de horas en procesarla.

Para ejemplificar este hecho, considere a una persona común a la cual se le pide que en una fotografía reconozca el rostro de un amigo. En la fotografía se encuentran cuatro personas y su amigo se encuentra de perfil, la persona tardaría aproximadamente un segundo en reconocer el rostro de su amigo, en cambio si a una computadora se le muestra la misma fotografía y se le ordena que reconozca el rostro de la misma persona tardaría horas en reconocerlo, si es que logra hacerlo. Como contrapartida, las máquinas construidas por el hombre llevan a cabo otro tipo de acciones, como pueden ser el cálculo y el razonamiento lógico, mucho más eficientemente que el cerebro.

Comprender la causa de este hecho es sencillo, basta con comparar la arquitectura de las computadoras y la estructura del cerebro. Los computadores actuales son máquinas “von Neumann”, en esencia una máquina de procesamiento que actúa ejecutando en serie una secuencia de instrucciones o programa, que almacena en su memoria. Por lo contrario, el cerebro, cuyas habilidades se quieren emular, no opera de acuerdo al marco descrito. El cerebro no es una arquitectura “von Neumann”, pues no está formado por un microprocesador, ni está constituido por unas cuantas CPU, sino que lo componen millones de procesadores elementales o neuronas, ampliamente interconectadas conformando redes de neuronas. La neurona es en realidad un pequeño procesador, sencillo, lento y poco fiable (a diferencia de los potentes procesadores actuales); sin embargo, en nuestro cerebro cohabitan unos cien mil millones de neuronas

operando en paralelo. Es aquí donde reside el origen de su poder de cómputo. Las neuronas individualmente son capaces de realizar procesamientos muy simples, aunque ampliamente interconectadas a través de la sinapsis y trabajando en paralelo pueden desarrollar una actividad global de procesamiento enorme.

Para comprender mejor las redes neuronales a continuación se describe su estructura biológica.

1.1 El modelo biológico

El cerebro es el elemento principal del sistema nervioso humano y está compuesto por un tipo especial de célula llamada neurona. Una neurona es una célula viva y como tal posee todos los elementos comunes de las células biológicas. A su vez, las neuronas tienen características propias que le permiten comunicarse entre ellas, lo que las diferencia del resto de las células biológicas. Se estima que el sistema nervioso contiene alrededor de cien mil millones de neuronas. Vistas al microscopio este tipo de células puede presentarse en múltiples formas, aunque muchas de ellas presentan un aspecto similar muy peculiar (ver Figura 1.1), con un cuerpo celular o soma (de entre 10 y 80 micras de longitud), del cual surge un denso árbol de ramificaciones (árbol dendrítico) compuesto por las dendritas, del cual parte una fibra tubular denominada axón (cuya longitud es alrededor de 100 micras), que también se ramifica en su extremo final para conectar con otras neuronas.

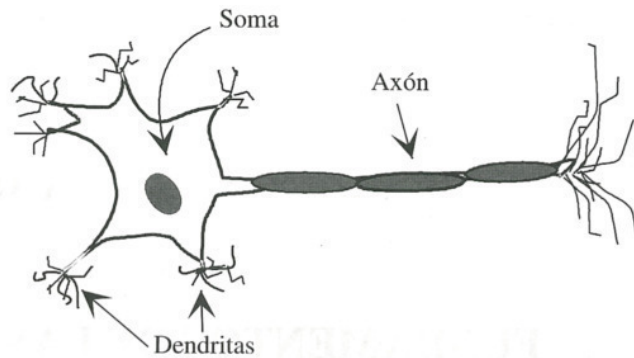


Figura 1.1 Estructura de una neurona biológica típica

Desde un punto de vista funcional las neuronas constituyen procesadores de información sencillos. Como todo sistema de este tipo, poseen un canal de entrada de información, las dendritas, un órgano de cómputo, el soma, y un canal de salida, el axón. En las interneuronas el axón envía la información a otras neuronas, mientras que en las neuronas motoras lo hace directamente al músculo. Existen neuronas receptoras o sensoras, que en vez de recibir la información de otras neuronas, la reciben directamente del exterior.

La unión entre dos neuronas se denomina sinapsis. En el tipo de sinapsis más común no existe un contacto físico entre las neuronas, sino que éstas permanecen separadas por un pequeño vacío de aproximadamente 0.2 micras. En relación a la sinapsis, se habla de la neurona presináptica (la que envía las señales), y la postsináptica (la que las recibe).

1.1.1 Aprendizaje

Las sinapsis son direccionales, es decir, la información fluye siempre en un único sentido. La intensidad de una sinapsis no es fija, sino que puede modificarse con base en la información proveniente del medio. De esta manera la estructura del cerebro no permanece fija sino que se va modificando por la formación de nuevas conexiones, ya sean excitadoras o inhibitoras, la destrucción de conexiones, la modificación de la intensidad de la sinapsis, o incluso por muerte neuronal. Esta plasticidad sináptica se supone que constituye, al menos en buena medida, el aprendizaje (se han encontrado evidencias experimentales de ello, Kandel 1992).

Durante el desarrollo de un ser vivo, el cerebro se modela, de forma de que existen muchas cualidades del individuo que no son innatas, sino que se adquieren por la influencia de la información que el medio externo proporciona a sus sensores. Existen diferentes formas de modelar el sistema nervioso: por el establecimiento de nuevas conexiones, ruptura de otras, modelado de las intensidades sinápticas (plasticidad) o incluso mediante muerte neuronal. Este tipo de acciones (en especial la modificación de intensidades sinápticas) son las que utilizan los sistemas neuronales artificiales para llevar a cabo el aprendizaje.

El esquema presentado en esta sección ha sido simplificado en buena medida, pues la realidad es mucho más compleja. El lector interesado en profundizar en los aspectos comentados puede leer, por ejemplo, Kandel 1999, Shepherd 1997, Arbib 1998. El sistema nervioso es lo suficientemente complejo como para que todavía quede mucho por descubrir y comprender.

1.2 Estructura de un sistema neuronal artificial

La Red Neuronal Artificial (RNA) imita la estructura del sistema nervioso, con la intención de construir sistemas de procesamiento de la información paralelos, distribuidos y adaptativos, que puedan presentar un cierto comportamiento “inteligente”.

Recordemos una idea ya comentada en la Introducción: pese al extraordinario desarrollo de la electrónica y las ciencias de la computación, ni el ordenador más potente puede llevar a cabo tareas tales como reconocer una mosca y atraparla al vuelo, que un sistema tan simple como el cerebro de la rana es capaz de llevar a cabo con eficiencia.

Como pudo apreciarse, el cerebro y una computadora convencional son mucho más diferentes de lo que suele suponerse cuando se habla de “cerebros electrónicos”. Recordemos que una computadora convencional es, en esencia, una máquina “von Neumann”, construida en torno a una única CPU o procesador, que ejecuta de modo secuencial un programa almacenado en memoria. Por el contrario, el cerebro no está compuesto por un único procesador, sino por miles de millones de ellos (neuronas), aunque muy elementales. Curiosamente, las neuronas son mucho más simples, lentas y menos fiables que una CPU, y a pesar de ello, existen problemas difícilmente abordables mediante un computador convencional, que el cerebro resuelve eficazmente.

Capítulo 1. Introducción a las Redes Neuronales Artificiales

Con base a lo anterior, la idea que subyace en los sistemas neuronales artificiales es que, para abordar el tipo de problemas que el cerebro resuelve con eficiencia, puede resultar conveniente construir sistemas que “copien” en cierto modo la estructura de las redes neuronales biológicas con el fin de alcanzar una funcionalidad similar.

Los elementos básicos de un sistema neuronal biológico son las neuronas, que se agrupan en conjuntos compuestos por millones de ellas organizadas en capas, constituyendo un sistema con funcionalidad propia. Un conjunto de estos subsistemas da lugar a un sistema global (el sistema nervioso, en el caso biológico). En la realización de un sistema neuronal artificial puede establecerse una estructura jerárquica similar. El elemento esencial de partida será la neurona artificial, que se organizará en capas; varias capas constituirán una red neuronal; y, por último, una red neuronal (o conjunto de ellas), junto con las interfaces de entrada y salida, constituirán el sistema global de proceso (Figura 1.2).

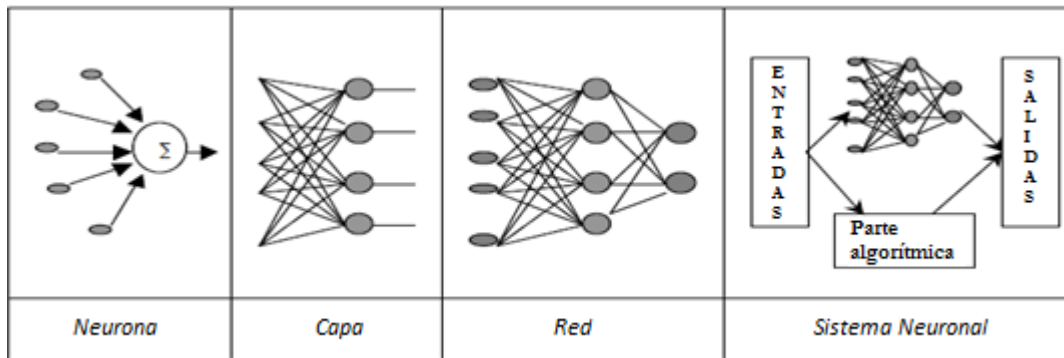


Figura 1.2 Estructura jerárquica de un sistema basado en RNA

Formalmente, y desde el punto de vista del grupo PDP (Parallel Distributed Processing Research Group, de la Universidad de California en San Diego, de D.E. Rumelhart y J.L. McClelland Rumelhart 1986, McClelland 1986), un sistema neuronal o conexionista, está compuesto por los siguientes elementos:

- Un conjunto de procesadores elementales o neuronas artificiales.
- Un patrón de conectividad o arquitectura.
- Una dinámica de activaciones.
- Una regla o dinámica de aprendizaje.
- El entorno donde opera.

Debido a que investigadores de numerosas áreas del conocimiento (neurobiológicos, psicólogos, matemáticos, físicos, ingenieros, etc.) trabajan en RNA, y a causa también de la relativa juventud de esta disciplina, coexisten diferentes terminologías y nomenclaturas. En este sentido, se están llevando a cabo esfuerzos encaminados a unificar conceptos, como los del comité de estandarización de la Neural Network Society del IEEE (“Institute of Electrical and Electronics Engineers”), o los de algunos investigadores (Fiesler 1994).

1.3 Sistemas paralelos, distribuidos y adaptativos

Los tres conceptos claves de los sistemas nerviosos que se pretenden emular en los artificiales, son: paralelismo de cálculo, memoria distribuida y adaptabilidad al entorno. De esta manera, se puede hablar de las redes neuronales como sistemas paralelos, distribuidos y adaptativos.

El *procesamiento paralelo* resulta esencial, como se deduce de un sencillo ejemplo. Un ordenador convencional tipo PC, que trabaja secuencialmente, instrucción a instrucción, emplearía varios minutos en realizar sobre una imagen compuesta por 256x256 píxeles una sencilla tarea de tratamiento en bajo nivel (acentuar contrastes, extraer contornos,...). Un sistema basado en un procesador digital de señales operando en paralelo emplearía del orden de 20 milisegundos en la misma tarea, puesto que cada uno podría operar en paralelo sobre diferentes sectores de la imagen. Por otra parte, el cerebro tarda aproximadamente este mismo tiempo en preprocesar una imagen compuesta por millones de píxeles, extraer sus rasgos característicos, analizarla, e interpretarla. Ningún sistema creado por el hombre es capaz de realizar algo semejante. La clave reside en que en este último caso los miles de millones de neuronas que intervienen en el proceso de visión (solamente en la retina, y sin contar el córtex cerebral, intervienen millones de ellas) están operando en paralelo sobre la totalidad de la imagen.

Otro concepto importante que aparece en el cerebro es el de *memoria distributiva*. Mientras que en una computadora la información ocupa posiciones de memoria bien definidas, en los sistemas neuronales se encuentra distribuida por las sinapsis de la red, de modo que si una sinapsis resulta dañada, no perdemos más que una parte muy pequeña de la información. Además, los sistemas neuronales biológicos son redundantes, de modo que muchas neuronas y sinapsis pueden realizar un papel similar; en definitiva, el sistema resulta tolerante a fallos (por ejemplo, cada día mueren miles de neuronas en nuestro cerebro, y sin embargo tienen que pasar muchos años para que se resientan nuestras capacidades).

El último concepto fundamental es el de *adaptabilidad*. Las RNA se adaptan fácilmente al entorno modificando sus sinapsis (y mediante otros mecanismos también), y aprender de la experiencia, pudiendo generalizar conceptos a partir de casos particulares. En el campo de las redes neuronales a esta propiedad se le llama generalización a partir de ejemplos.

1.4 El modelo de una neurona artificial

En esta sección se expone el modelo de neurona de las RNA. En primer lugar, se describe la estructura de una neurona artificial muy genérica. A continuación se muestra una versión simplificada de amplio uso en los modelos orientados a aplicaciones prácticas, que posee una estructura más próxima a la neurona tipo clásica.

Aunque el comportamiento de algunos sistemas neuronales biológicos sea lineal, como sucede en la retina del cangrejo *Limulus*, en general, la respuesta de las neuronas biológicas es de tipo **no lineal**, característica que se emula en las RNA. La formulación de la neurona artificial como dispositivo no lineal constituye una de sus características más destacables, y una de las que

proporciona un mayor interés en las RNA, pues el tratamiento de problemas altamente no lineales no suele ser fácil de abordar mediante técnicas convencionales.

1.4.1 Modelo general de neurona artificial

En esta sección se describe la estructura genérica de neurona artificial en el marco establecido por el grupo PDP (Rumelhart 1986, McClelland 1986).

Se denomina procesador elemental o neurona a un dispositivo simple de cálculo que, a partir de un vector de entrada procedente del exterior o de otras neuronas, proporciona una única respuesta o salida. Los elementos que constituyen la neurona de etiqueta i son los siguientes (véase la Figura 1.3):

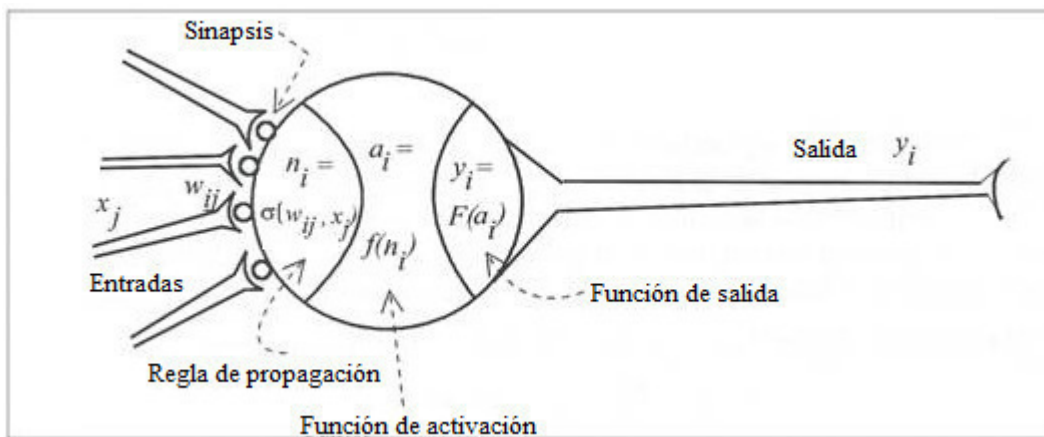


Figura 1.3 Modelo genérico de una neurona artificial

- Conjunto de **entradas**, $x_j(t)$, donde t representa el estado actual de la RNA.
- **Pesos sinápticos** de la neurona i , w_{ij} que representan la intensidad de interacción entre cada neurona presináptica j y la neurona postsináptica i .
- **Regla de propagación** $\sigma(w_{ij}, x_j(t))$, que proporciona el valor del potencial postsináptico $n_i(t) = \sigma(w_{ij}, x_j(t))$ de la neurona i en función de sus pesos y entradas.
- **Función de activación** $f_i(a_i(t-1), n_i(t))$, que proporciona el estado de activación actual $a_i(t) = f_i(a_i(t-1), n_i(t))$, de la neurona i , en función de su estado anterior $a_i(t-1)$ y de su potencial postsináptico actual.
- **Función de salida** $F_i(a_i(t))$, que proporciona la salida actual $y_i(t) = F_i(a_i(t))$, de la neurona i en función de su estado de activación.

Capítulo 1. Introducción a las Redes Neuronales Artificiales

De este modo, la operación de la neurona i puede expresarse como

$$y_i(t) = F_i(f_i[a_i(t-1), \sigma(w_{ij}, x_j(t))]) \quad (1.1)$$

Este modelo de neurona formal se inspira en la operación de la biología, en el sentido de integrar una serie de entradas y proporcionar cierta respuesta, que se propaga por el axón.

A continuación se describen con mayor detalle los conceptos básicos antes mencionados.

- **Entradas y salidas**

Las variables de entrada y salida pueden ser binarias (digitales) o continuas (analógicas), dependiendo del modelo de aplicación. Por ejemplo, un perceptrón multicapa o MLP (*Multilayer Perceptron*), el cual será abordado posteriormente, admite ambos tipos de señales. Por ejemplo, para tareas de clasificación poseería salidas digitales $\{0, +1\}$, mientras que para un problema de ajuste funcional de aplicación multivariable continua, se utilizarían salidas continuas pertenecientes a un cierto intervalo.

Dependiendo del tipo de salida, las neuronas suelen recibir nombres específicos (Müller 90). Las neuronas estándar cuya salida sólo puede tomar los valores 0 o 1 se suelen denominar genéricamente neuronas de tipo McCulloch-Pitts, mientras que aquellas que únicamente pueden tener por salidas -1 o 1 se suelen denominar neuronas tipo Ising. Si puede adoptar diversos valores discretos en la salida (por ejemplo -2, -1, 0, +1, +2), se dice que se trata de una neurona tipo Potts. En ocasiones, el intervalo de los valores que una neurona de salida continua puede proporcionar se suele limitar a un intervalo definido, por ejemplo, $(0, +1)$ o $(-1, +1)$.

- **Peso sináptico**

Define en este caso la intensidad entre la neurona presináptica j y la postsináptica i . Dada una entrada positiva (procedente de un sensor o simplemente la salida de otra neurona), si el peso es positivo, tenderá a excitar a la neurona postsináptica; si el peso es negativo, tenderá a inhibirla. Así se habla de sinapsis excitadoras (de peso positivo) e inhibidoras (de peso negativo). En caso de que el peso sea cero, no existe comunicación entre el par de neuronas. Mediante el ajuste de los pesos sinápticos la red es capaz de adaptarse a cualquier entorno y realizar una determinada tarea.

- **Regla de propagación**

La **regla de propagación** $\sigma(w_{ij}, x_j(t))$ permite obtener, a partir de las entradas y los pesos, el valor del potencial postsináptico n_i de la neurona

$$n_i(t) = \sigma(w_{ij}, x_j(t)) \quad (1.2)$$

La función más habitual es de tipo lineal, y se basa en la **suma ponderada** de las entradas con los pesos sinápticos

Capítulo 1. Introducción a las Redes Neuronales Artificiales

$$n_i(t) = \sum_j w_{ij}x_j \quad (1.3)$$

Formalmente también puede interpretarse como el producto escalar de los vectores de entrada y pesos

$$n_i(t) = \sum_j w_{ij}x_j = \mathbf{w}_i^T \cdot \mathbf{x} \quad (1.4)$$

- **Función de activación o de transferencia**

Proporciona el estado de activación actual, $a_i(t)$, de la neurona i en función de su estado anterior, $a_i(t-1)$, y de su potencial postsináptico actual. Es decir:

$$a_i(t) = f_i(a_i(t-1), n_i(t)) \quad (1.5)$$

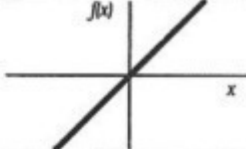
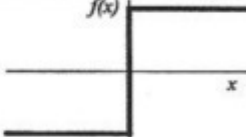
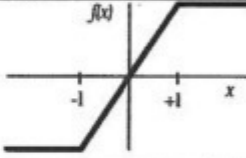
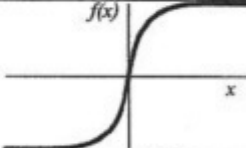
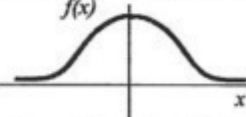
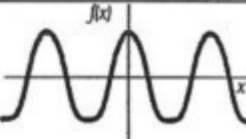
En muchos modelos de RNA se considera que el estado actual de la neurona no depende de su estado anterior, sino únicamente del actual, es decir:

$$a_i(t) = f_i(n_i(t)) \quad (1.6)$$

La función de activación $f(\cdot)$ se suele considerar determinista, y en la mayor parte de los modelos es monótona creciente y continua. Las formas $y = f(x)$ de las funciones de activación más empleadas en las RNA se muestran en la Tabla 1.1, donde x representa el potencial postsináptico, y el estado de activación.

En ocasiones los algoritmos de aprendizaje requieren que la función de activación cumpla con la condición de ser derivable. Las más empleadas en este caso son las funciones de tipo sigmoideo, como la de "Backpropagation" (BP, la cual será estudiada con detalle posteriormente). Otra función clásica es la Gaussiana, que se utiliza junto con reglas de propagación que involucran el cálculo de cuadrados de distancias (por ejemplo, la euclídea) entre los vectores de entradas y pesos. Alternativamente, en ocasiones se emplean funciones senoidales, en los casos que se requiere expresar explícitamente una periodicidad temporal.

Tabla 1.1 Funciones de activación más comunes

	Función	Intervalo	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal por tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Senoidal	$y = A\text{sen}(\omega x + \varphi)$	$[-1, +1]$	

• **Función de salida:**

Proporciona la salida actual, $y_k(t)$, de la neurona k en función de su estado de activación actual, $a_i(t)$. Muy frecuentemente la función de salida es simplemente la identidad $F(x) = x$, de modo que el estado de activación de la neurona se considera como la propia salida, es decir:

$$y_i(t) = F_i(a_i(t)) = a_i(t) \tag{1.7}$$

De manera general, la operación de la neurona i puede expresarse de la siguiente forma:

$$y_i(t) = F_i(f_i(a_i(t-1), \sigma_i(w_{ij}, x_j(t))))$$

(1.8)

A manera de ilustración, veamos cómo podría ser la interconexión entre dos neuronas. Para ello en la Figura 1.4 se presenta la interacción entre una neurona presináptica y una neurona postsináptica.

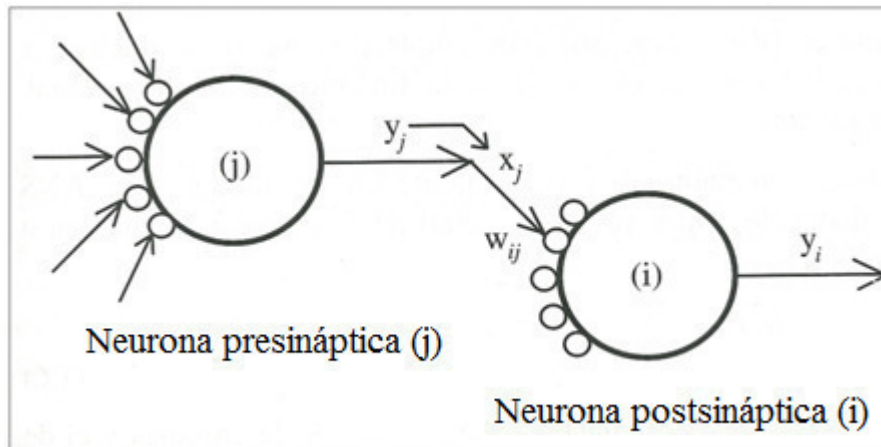


Figura 1.4 Interacción entre una neurona presináptica y otra postsináptica

1.4.2 Modelo estándar de neurona artificial

Considerando que la regla de propagación es la suma ponderada y que la función de salida es la identidad, la *neurona estándar* consiste en:

- Un conjunto de **entradas** $x_j(t)$.
- Unos **pesos sinápticos** w_{ij} asociados a las entradas.
- Una **regla de propagación** $n_i(t) = \sigma(w_{ij}, x_j(t))$. La más común suele ser $n_i(t) = \sum w_{ij} x_j$.
- Una **función de activación** $Y_i(t) = f_i(n_i(t))$ que representa simultáneamente la salida de la neurona y su estado de activación.

Todos estos elementos están resumidos en la Figura 1.5.

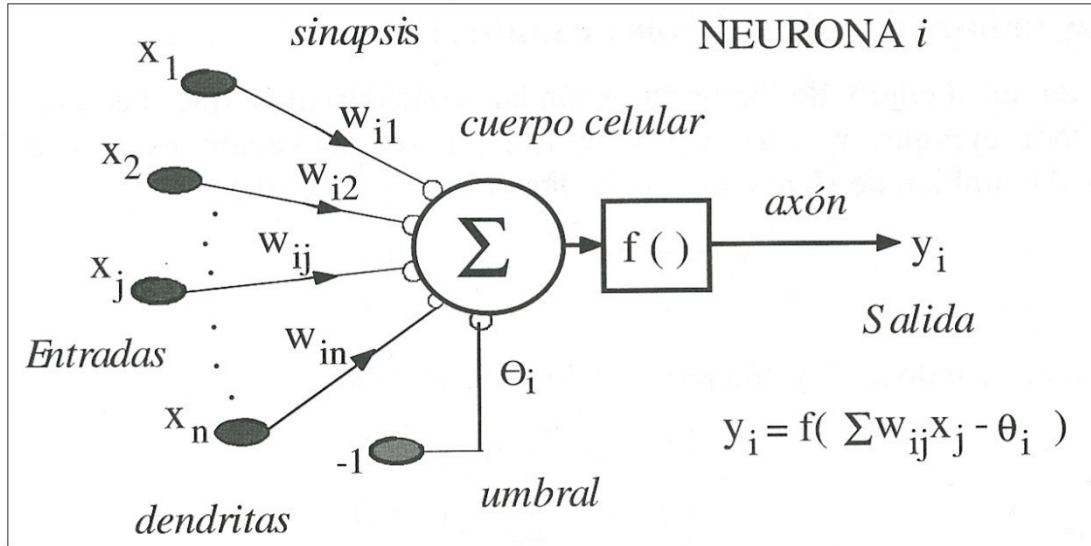


Figura 1.5 Modelo de neurona estándar

Con frecuencia se añade al conjunto de pesos de la neurona un parámetro adicional, θ_i , que denominaremos *umbral*¹, se resta del potencial postsináptico, por lo que el argumento de la función de activación queda

$$\sum_j w_{ij}x_j - \theta_i \quad (1.9)$$

Ahora bien, si los índices i y j comienzan en 0, y definiendo $w_{i0} = \theta_i$ y $x_0 = -1$ (constante), se puede obtener el comportamiento de la neurona a través de:

$$y_i(t) = f_i\left(\sum_{j=0}^n w_{ij}x_j\right) \quad (1.10)$$

Definida de esta manera la neurona estándar, basta con establecer la forma de la función de activación (Tabla 1.1) para determinarla por completo.

A continuación se muestran algunos de los tipos más usuales de neuronas.

¹Cantidad mínima de señal que ha de estar presente para ser registrada por un sistema.

1.5 Tipos de neuronas

1.5.1 Neuronas Todo-Nada (dispositivos de tipo umbral)

Considere que las entradas son digitales, por ejemplo $x_i = \{0,1\}$, y la función de activación es la función escalón $H(\cdot)$ (Heavside), definida entre 0 y 1. Entonces se tiene:

$$y_i(t) = H\left(\sum_j w_{ij}x_j - \theta_i\right)$$

Como $H(x) = 1$ cuando $x \geq 0$, y $H(x) = 0$ cuando $x < 0$, se obtiene:

$$y_i = \begin{cases} 1, & \sum w_{ij}x_j \geq \theta_i \\ 0, & \sum w_{ij}x_j < \theta_i \end{cases}$$

Es decir, si el potencial de membrana supera un valor umbral θ_i (*umbral de disparo*), entonces la neurona se activa, si no lo supera, la neurona no se activa. Este es el modelo de **neurona del Perceptrón original** o **dispositivo de tipo umbral**.

McCulloch y Pitts (1943) demostraron que se podía realizar cualquier función lógica mediante redes basadas en este modelo de neurona.

1.5.2 Neurona sigmoidea

En la neurona sigmoidea las entradas pueden ser tanto digitales como analógicas, mientras que las salidas, exclusivamente analógicas. Tomando como función de activación una sigmoidea (Tabla 1.1), se pueden definir las dos formas siguientes, en función del intervalo considerado:

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad , \quad y \in [0,1] \quad (1.11)$$

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \text{tgh}(x) \quad , \quad y \in [-1,1] \quad (1.12)$$

Este tipo de neuronas se utiliza comúnmente en el **perceptrón multicapa** debido a que este tipo de neuronas se pueden utilizar para resolver problemas no lineales.

1.5.3 Neurona estocástica

El modelo de neurona sigmoidea anterior puede interpretarse desde un punto de vista probabilístico, al utilizar neuronas con funciones de activación no deterministas, cuyas salidas se obtienen de forma probabilística y mediante mecanismos de aprendizaje estocástico. Las redes

neuronales probabilísticas (RNP) pueden ser utilizadas para problemas de clasificación. Cuando una entrada es presentada, la primera capa calcula las distancias desde el vector de entrada a los vectores de entrada entrenados y produce un vector cuyos elementos indican que tan cerca está la entrada respecto a la entrada entrenada. La segunda capa suma estas contribuciones para cada clase de entrada y produce con ellas un vector de salida con las probabilidades correspondientes. Finalmente, una función de transferencia en la salida de la segunda capa toma la máxima de estas probabilidades y produce un 1 para esa clase y cero para las demás.

1.6 Arquitectura de una red neuronal artificial

Una vez definido el tipo de neurona que se utilizará en un modelo de redes neuronales artificiales es necesario definir la topología de la misma.

La organización y disposición de las neuronas dentro de una red neuronal se denomina topología, y viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas.

En general, las neuronas se suelen agrupar en unidades estructurales denominadas *capas*. Dentro de una capa las neuronas pueden agruparse formando *grupos neuronales*. Dentro de una misma capa o agrupación las neuronas suelen ser del mismo tipo. El conjunto de una o más capas constituye una *red neuronal*. Podemos distinguir tres tipos de capas (ver Figura 1.6):

- *Capa de entrada*: compuesta por neuronas que reciben datos o señales procedentes del entorno.
- *Capa de salida*: aquella cuyas neuronas proporcionan la respuesta de la red neuronal.
- *Capa oculta*: aquella que no tiene una conexión directa con el entorno.

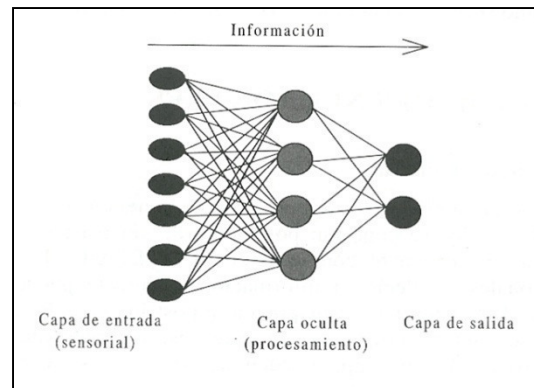


Figura 1.6 Capas de una red

Entre un par de neuronas de la red neuronal artificial pueden existir conexiones. Estas conexiones son las sinapsis, tienen asociadas un peso sináptico, y son direccionales. Cuando la conexión se establece entre dos neuronas de una misma capa se habla de conexiones laterales o conexiones intra-capas. Por el contrario, si la conexión se establece entre neuronas de distintas capas se la

denomina conexión inter-capa. Si la conexión se produce en el sentido inverso al de entrada-salida la conexión se llama recurrente o realimentada. Una red puede estar formada por una única capa de neuronas. En este caso se habla de redes monocapa, y las neuronas que conforman dicha capa cumplen la función de neuronas de entrada y salida simultáneamente. Cuando la red está compuesta por dos o más capas se denomina multicapa. A su vez, se habla de redes neuronales con conexión hacia delante (redes "feedforward") cuando las conexiones entre las distintas neuronas de la red siguen un único sentido, desde la entrada de la red hacia la salida de la misma. Cuando las conexiones pueden ser tanto hacia delante como hacia atrás entonces se tiene una red recurrente (redes "feedback"). A continuación se describen con mayor detalle los diferentes tipos de redes neuronales.

1.7 Tipos de redes neuronales

Una vez estudiados los elementos básicos de las redes neuronales, se pasará a enumerar las diferentes estructuras en las que dichos elementos se pueden asociar.

1.7.1 Red neuronal monocapa

Se considera que es la red neuronal más sencilla ya que se tiene una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan los diferentes cálculos. (ver Figura 1.7).

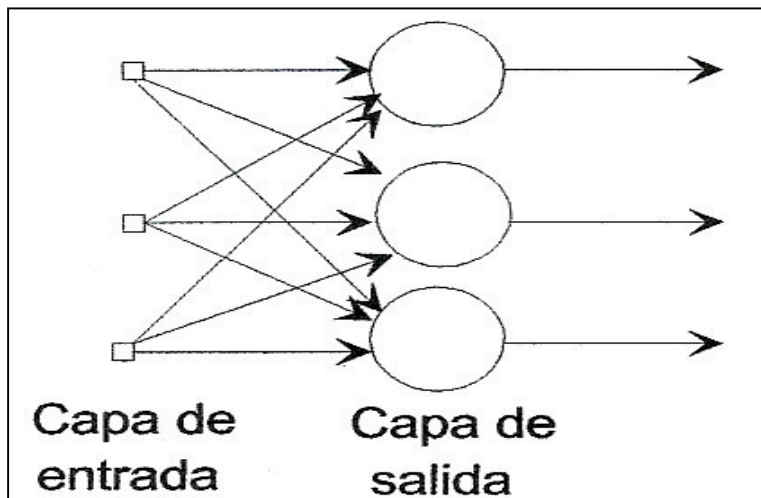


Figura 1.7. Red neuronal monocapa

1.7.2 Red neuronal multicapa

Es una generalización de la anterior. En la red neuronal multicapa existe un conjunto de capas intermedias entre la capa de entrada y la de salida (capas ocultas). Este tipo de red puede estar total o parcialmente conectada (ver Figura 1.8).

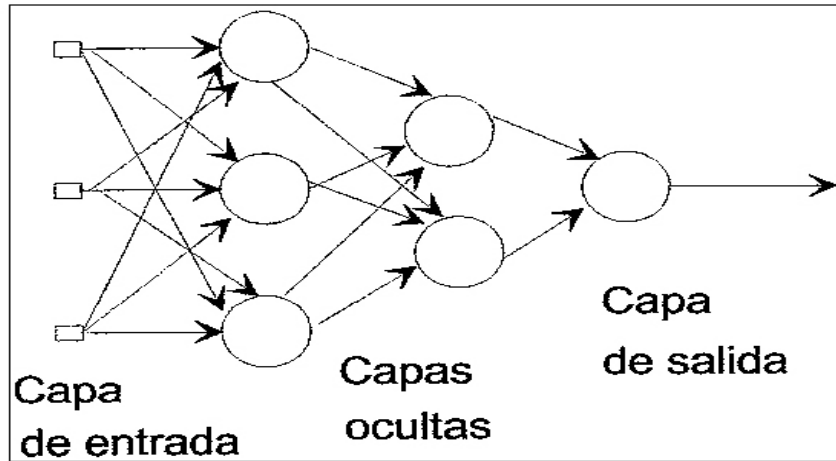


Figura 1.8 Red neuronal multicapa

1.7.3 Red neuronal recurrente

Este tipo de red se diferencia de las anteriores en la existencia de lazos de realimentación en la red. Estos lazos pueden ser entre neuronas de diferentes capas, neuronas de la misma capa o, entre una misma neurona. Esta estructura la hace especialmente adecuada para estudiar la dinámica de los sistemas no lineales (Figura 1.9).

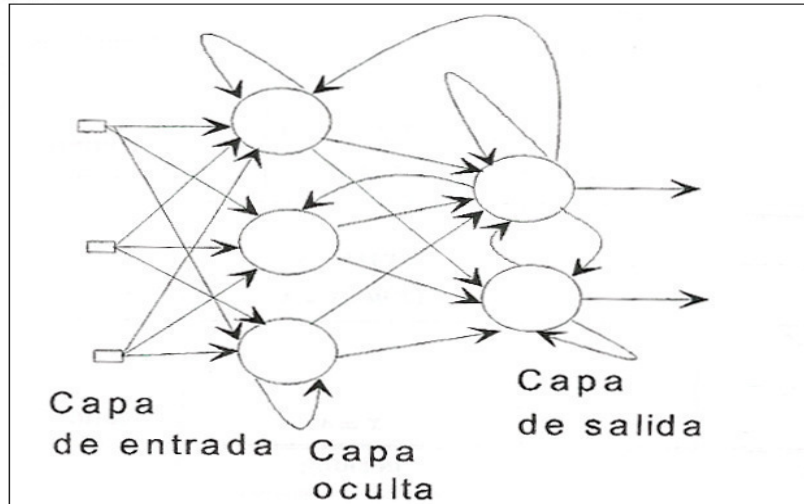


Figura 1.9 Red neuronal recurrente

1.8 Definición formal de red neuronal

Para obtener una definición de red se usa el concepto matemático *grafo*. A través de este término, se puede definir una red neuronal de la siguiente forma:

Una **red neuronal** es un grafo dirigido con las siguientes propiedades:

- 1) A cada nodo i se le asocia un variable de estado x_i .
- 2) A cada conexión (i, j) de los nodos i y j se le asocia un peso $w_{ij} \in R$.
- 3) A cada nodo i se le asocia un umbral θ_i .
- 4) Para cada nodo i se define una función $f_i(x_i, w_{ij}, \theta_i)$, que depende de los pesos de sus conexiones, del umbral y de los estados de los nodos j conectados a él. Esta función proporciona el nuevo estado del nodo.

1.9 Modos de operación: recuerdo y aprendizaje

1.9.1 Fase de aprendizaje

Puede definirse el *aprendizaje* como el proceso por el que se produce el ajuste de los parámetros libres de la red a partir de un proceso de estimulación por el entorno que rodea a la red. En la mayoría de los casos el aprendizaje consiste simplemente en determinar un conjunto de pesos sinápticos que permita a la red realizar correctamente el tipo de procesamiento deseado.

Al construir un sistema neuronal, se parte de un cierto modelo de neurona y de una determinada arquitectura de red, estableciéndose los pesos sinápticos iniciales como nulos o aleatorios. Para que la red resulte operativa es necesario entrenarla. El entrenamiento o aprendizaje se puede llevar a cabo mediante dos niveles:

1) A través del modelado de las sinapsis; que consiste en modificar los pesos sinápticos siguiendo una cierta regla de aprendizaje, construida normalmente a partir de la optimización de una función de error o coste, que mide la eficacia actual de la operación de la red.

Si $w_{ij}(t)$ es el peso que conecta la neurona presináptica j con la postsináptica i en la instante t , el algoritmo de aprendizaje, en función de las señales que en el instante t llegan procedentes del entorno, proporcionará el valor $\Delta w_{ij}(t)$ que da la modificación que se debe incorporar en dicho peso, el cual quedará actualizado de la forma:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (1.13)$$

El proceso de aprendizaje es usualmente iterativo, se actualizan los pesos de la manera anterior (ecuación 1.13), una y otra vez, hasta que la red neuronal alcanza el rendimiento deseado.

2) A través de la creación o destrucción de neuronas. En esta alternativa se lleva a cabo una modificación de la propia arquitectura de la red.

Los tipos de aprendizaje que pueden distinguirse son:

- Supervisado.
- No supervisado o Auto organizado.
- Híbrido.
- Reforzado.

Los algoritmos de aprendizaje se basan usualmente en métodos numéricos iterativos que tratan de reducir una función de coste, lo que puede dar lugar a veces a problemas en la convergencia del algoritmo. En un sentido riguroso, la convergencia es una manera de comprobar si una determinada arquitectura, junto a su regla de aprendizaje, es capaz de resolver cierto problema.

En el proceso de aprendizaje es importante distinguir entre el nivel de error alcanzado al final de la fase de aprendizaje para el conjunto de datos de entrenamiento, y el error que la red ya entrenada comete ante patrones no utilizados en el aprendizaje, lo cual mide la capacidad de *generalización* de la red. Interesa más una buena generalización que un error muy pequeño en el entrenamiento, pues ello indicará que la red ha capturado la topografía de la memoria (“mapping”) subyacente en los datos.

1.9.1.1 Aprendizaje supervisado

En el aprendizaje supervisado se presenta a la red un conjunto de patrones, junto con la salida deseada u objetivo, e iterativamente ésta ajusta sus pesos hasta que la salida tiende a ser la deseada, para ello se utiliza información detallada del error que se comete en cada paso. De este modo, la red es capaz de estimar relaciones entrada/salida sin necesidad de proponer una cierta forma funcional de partida. Es decir, si $e[W]$ es una función que representa el error esperado de la operación de la red, expresado en función de sus pesos sinápticos W , se pretende estimar una cierta función multivariable $f: R^n \rightarrow R^m$ a partir de muestras (x, y) tomadas aleatoriamente por medio de la minimización iterativa de $e[W]$, mediante aproximación estocástica (las técnicas de aproximación estocástica estiman valores esperados a partir de cantidades aleatorias observadas).

1.9.1.2 Aprendizaje no supervisado o auto organizado

El aprendizaje no supervisado se puede describir genéricamente como la estimación de la función densidad de probabilidad $P(x)$ que describe la distribución de patrones $x \in R^n$ (espacio de entrada).

En este tipo de aprendizaje se presentan a la red multitud de patrones sin adjuntar la respuesta que deseamos. La red, por medio de la regla de aprendizaje, estima $P(x)$, a partir de lo cual podemos reconocer regularidades en el conjunto de entradas, extraer rasgos o agrupar patrones según su similitud (“clustering”).

1.9.1.3 Aprendizaje híbrido

En este caso existen en la red los dos tipos de aprendizaje básicos, supervisado y auto organizado, normalmente en distintas capas de neuronas.

1.9.1.4 Aprendizaje reforzado

Se sitúa entre el aprendizaje supervisado y el auto organizado. Como en el primero, se emplea información sobre el error cometido, pero en este caso existe una única señal de error que representa un índice global del rendimiento de la red (solamente le indicamos lo bien o lo mal que está actuando). Como en el caso del no supervisado, no se suministra explícitamente la salida deseada.

1.9.2 Fase de recuerdo (ejecución)

Una vez que el sistema ha sido entrenado, el aprendizaje se desconecta, por lo que los pesos y la estructura quedan fijos, estando la red neuronal ya dispuesta para procesar datos.

En las redes unidireccionales, ante un patrón de entrada, las neuronas responden proporcionando directamente la salida del sistema. Al no existir bucles de realimentación, no existe ningún problema en relación con su estabilidad. Por el contrario, las redes con realimentación son sistemas dinámicos no lineales, que requieren ciertas condiciones para que su respuesta acabe convergiendo a un estado estable o punto fijo. Existe una serie de teoremas generales que establecen las condiciones que aseguran la estabilidad de la respuesta en una amplia gama de redes neuronales en ciertas condiciones.

Para demostrar la estabilidad del sistema, dichos teoremas se basan en el *método de Lyapunov*. Dicho método constituye una manera accesible de estudiar la estabilidad de un sistema dinámico. Consiste en tratar de encontrar una función del sistema, que disminuya siempre en su operación, entonces el sistema es estable.

1.10 Clasificación de los modelos de redes neuronales artificiales

Dependiendo del modelo de neurona concreto que se utilice, de la arquitectura o topología de conexión, y del algoritmo de aprendizaje, surgirán distintos modelos de redes neuronales como se muestra en la Figura 1.10 (Brío y Sanz 2002)

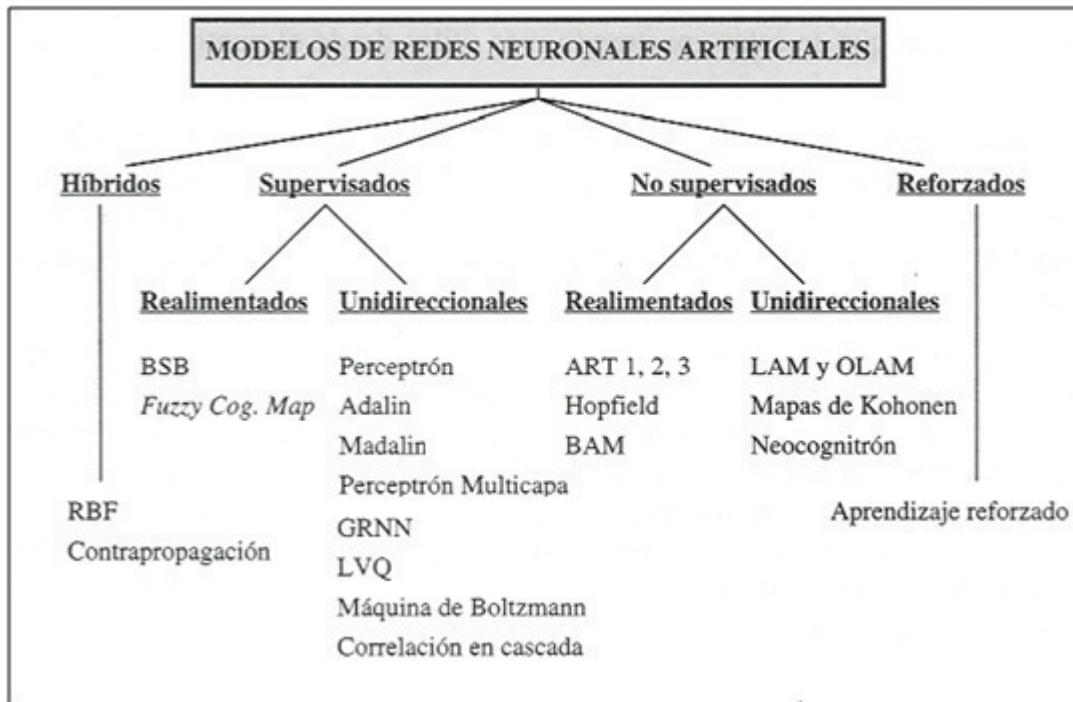


Figura 1.10 Clasificación de los RNA por el tipo de aprendizaje y la arquitectura (Brío y Sanz, 2002)

1.10.1 Redes neuronales con conexión hacia adelante

En esta tesis se tratan particularmente las redes neuronales artificiales con conexión hacia adelante. Este tipo de red, que se caracteriza por su organización en capas y conexiones estrictamente hacia adelante. Utiliza algoritmos de entrenamiento del tipo supervisado.

Este grupo de red es el más utilizado en aplicaciones prácticas que utilicen redes neuronales, obteniéndose muy buenos resultados fundamentalmente como clasificadores de patrones y estimadores de funciones. Dentro de este grupo de redes neuronales se encuentra el perceptrón, la red Adaline/Madaline, y al perceptrón multicapa.

1.10.1.1 Perceptrón

Este modelo tiene gran importancia histórica ya que fue el primer modelo en poseer un mecanismo de entrenamiento que permite determinar automáticamente los pesos sinápticos que clasifican correctamente a un conjunto de patrones a partir de un conjunto de ejemplos (Rosenblatt 1962; Hertz 1991; Principe 2000). La arquitectura del perceptrón está compuesta por dos capas de neuronas, una de entrada y una de salida. La capa de entrada es la que recibe la información proveniente del exterior y la transmite a las neuronas sin realizar ningún tipo de operación sobre la señal de entrada. En general la información entrante es binaria. La función de activación de las neuronas de un perceptrón es del tipo escalón (ver sección 1.5.1), dando de esta manera sólo salidas binarias. Cada neurona de salida del perceptrón representa a una clase. Una neurona de salida responde con 1 si el vector de entrada pertenece a la clase a la que representa y responde con 0 en caso contrario. La operación de un perceptrón con n neuronas de entrada y m neuronas de salidas puede resumirse de la siguiente manera:

$$y_i(t) = f\left(\sum_{j=1}^n w_{ij}x_j - \theta_i\right) \quad (1.14)$$

El algoritmo de entrenamiento del perceptrón se encuentra dentro de los denominados algoritmos por corrección de errores. Este tipo de algoritmos ajustan los pesos de manera proporcional a la diferencia entre la salida actual proporcionada por la red y la salida objetivo, con el fin de minimizar el error producido por la red. Se puede demostrar que este método de entrenamiento converge siempre en un tiempo finito y con independencia de los pesos de partida, siempre que la función a representar sea linealmente separable (es decir, cuyas regiones de decisión pueden ser separadas mediante una única condición lineal o hiperplano es decir, una línea recta si trabajamos en dos dimensiones). El principal problema de este método de entrenamiento es que cuando la función a representar no es linealmente separable el proceso de entrenamiento oscilará y nunca alcanzará la solución. Las funciones no separables linealmente no pueden ser representadas por un perceptrón.

1.10.1.2 Adalina / Madalina

Otro de los modelos que tienen gran importancia es la red neuronal Adalina. La topología de la red Adalina es similar a la del perceptrón sólo que en este caso la función de salida de las neuronas es lineal. Dado que las señales de entrada pueden ser continuas, la red Adalina es un dispositivo de entrada/salida analógica (continua), a diferencia del perceptrón que es un dispositivo de entrada/salida digital (binaria). La operación de una red Adalina con n neuronas de entrada y m neuronas de salidas puede resumirse de la siguiente manera:

$$y_i(t) = \sum_{j=1}^n w_{ij} x_j - \theta_i \quad (1.15)$$

La principal diferencia entre la red Adalina y el perceptrón consiste en la regla de aprendizaje que utilizan. En el caso de la red Adalina implementa como método de aprendizaje la regla de Widrow-Hoff, también conocida como regla LMS ("least mean squares", mínimos cuadrados), que realiza una actualización continua de los pesos sinápticos de acuerdo con la contribución de cada neurona sobre el error total de la red. Este método produce un conjunto de pesos sinápticos óptimos desde el punto de vista de mínimos cuadrados (un conjunto de pesos que minimiza el error cuadrático que comete la red), y en caso de que los vectores de entrada sean linealmente independientes produce una asociación perfecta entre entradas-salidas.

Existe una versión multicapa de la Adalina denominada Madalina (múltiples Adalinas) que consiste en una red neuronal con neuronas similares a las de la Adalina pero que contiene capas de neuronas ocultas.

1.10.1.3 Perceptrón multicapa

El perceptrón multicapa es una extensión del perceptrón simple. La topología de un perceptrón multicapa está definida por un conjunto de capas ocultas, una capa de entrada y una de salida. No existen restricciones sobre la función de activación, aunque en general se suelen utilizar funciones sigmoideas. La operación de un perceptrón multicapa con una única capa oculta puede resumirse de la siguiente manera:

$$z_k = \sum_j w'_{kj} y_j - \theta'_k = \sum_j w'_{kj} f\left(\sum_i w_{ji} x_i - \theta'_j\right) - \theta'_k \quad (1.16)$$

Este modelo es el más utilizado en la actualidad. El espectro de aplicaciones del perceptrón multicapa es muy amplio, lo que hace muy difícil enumerar todas sus aplicaciones más relevantes. Enseguida se mencionan algunas aplicaciones en Ingeniería Civil.

*Predicción de la aceleración máxima del terreno (Günaydin y Günaydin 2008)

*Estimación de espectros de respuesta elásticos (Alcántara et al. 2009)

*Optimación basada en confiabilidad y algoritmos evolutivos (Hurtado y Álvarez 2002)

*Optimación del costo estructural dada una confiabilidad mínima (Möller et al. 2009)

*Estimación de la probabilidad de falla de una estructura (Cardoso et al. 2009)

La popularidad de este modelo de redes neuronales no se debe únicamente al éxito obtenido en aplicaciones prácticas del mismo. Existen demostraciones teóricas que permiten explicar el éxito de dichas aplicaciones. Funahashi (1989) demuestra que un perceptrón multicapa cuya función de activación sea no constante, acotada y monótona creciente es un buen aproximador universal de funciones. Hornik et al (1989) llegan a un resultado similar utilizando funciones de activación sigmoideas, no necesariamente continuas.

En esta tesis se utiliza este modelo (perceptrón multicapa) en el cual se utiliza el algoritmo "Backpropagation" para su entrenamiento.

1.11 Entrenamiento

1.11.1 Para qué entrenar

El objetivo que persigue el entrenamiento de una red neuronal es básicamente establecer valores para el vector de pesos con los cuales el error cometido al evaluar los ejemplos de entrenamiento sea mínimo.

Una vez calculados estos pesos, la red estará lista para ser probada con otros patrones de prueba con los cuales no ha sido entrenada. El objetivo de esta nueva comprobación es probar cómo se comporta la red cuando las entradas son distintas a las que usa para el entrenamiento.

Al entrenar una red neuronal con unos determinados ejemplos de entrenamiento e intentar minimizar el error a valores muy pequeños, se corre el riesgo de especializar demasiado la red, la cual se comportará de una manera óptima con los ejemplos con los cuales ha sido entrenada; sin embargo, para ejemplos con los que no se ha entrenado, se pueden llegar a producir errores considerables.

Todo esto lleva a la conclusión de que hay que minimizar el error, pero no hay que intentar "rizar el rizo", ya que la especialización conlleva una pérdida de generalización.

Si este método de validación falla, debemos optar por la opción de volver a entrenar la red desde el principio, pero con unos pesos iniciales aleatorios distintos a los del entrenamiento anterior.

Si los resultados son correctos, la red estará lista para funcionar, y podremos asegurar que para nuevas entradas desconocidas para la red, ésta dará resultados *correctos*.

1.11.2 Cómo entrenar

El entrenamiento de las redes neuronales se realiza mediante algoritmos de entrenamiento que se

basan siempre en intentar buscar los pesos de las neuronas que ofrezcan mejores resultados.

Antes de iniciar a entrenar la red, lo primero que se debe hacer es definir la condición de parada (cuando la red obtiene resultados satisfactorios). Esta condición puede ocurrir por los siguientes motivos:

- Se ha alcanzado una cuota de error que se considere suficientemente pequeña.
- Se ha llegado a un número máximo de iteraciones que se ha definido como tope para el entrenamiento.
- Se ha llegado a un punto de saturación en el que, por más que se entrene, ya no es posible reducir el error.

1.12 El algoritmo “Backpropagation”

El algoritmo “Backpropagation” es el método de entrenamiento más utilizado en redes con *conexión hacia adelante*. Es un método de aprendizaje supervisado en el que se distinguen claramente dos fases: primero se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias. Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido, se ajustan los errores de los pesos sinápticos de cada neurona.

1.12.1 Dedución del algoritmo “Backpropagation”

En 1969 Arthur E. Bryson y Yu-Chi Ho describieron el algoritmo “Backpropagation”, pero fue hasta 1974 que a través de los trabajos de Paul Werbos el algoritmo se utilizó en el campo de las redes neuronales artificiales.

Para deducir el algoritmo “Backpropagation” aquí se utiliza una red sencilla de tres capas como la que se ilustra en la Figura 1.11. La deducción matemática de este procedimiento se realizará para una red con una capa de entrada, una capa oculta y una capa de salida. Después se generalizará el procedimiento para redes que tengan más de una capa oculta.

Para iniciar el entrenamiento se presenta a la red un patrón de entrenamiento, el cual tiene i componentes como se describe en la ecuación 1.17.

$$x = \begin{pmatrix} 1 \\ 2 \\ 3 \\ \cdot \\ i \end{pmatrix} \quad (1.17)$$

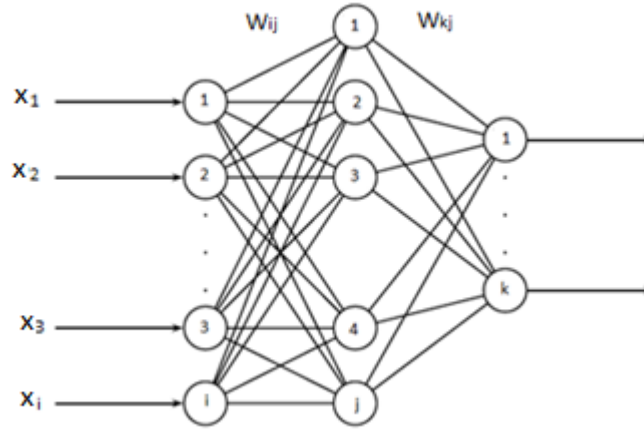


Figura 1.11 Red sencilla de tres capas

Sea

- i : número de componentes del vector de entrada.
- j : número de neuronas de la capa oculta.
- k : número de neuronas de la capa de salida.

Cuando se presenta a la red un patrón de entrenamiento, este se propaga por la red a través de las conexiones existentes produciendo una entrada neta n en cada una de las neuronas de la capa siguiente. La entrada neta a la neurona j de la siguiente capa debido a la presencia de un patrón de entrenamiento en la entrada está dada por la ecuación 1.18.

$$n_j^o = \sum_i w_{ji}^o x_i + \theta_j^o \quad (1.18)$$

w_{ji}^o .- Peso que une la componente i de la entrada con la neurona j de la capa oculta

x_i .- Vector de entrenamiento

θ_j^o .- Umbral de la neurona j de la capa oculta

Donde el superíndice (o) representa la capa a la que pertenece cada parámetro, en este caso la capa oculta.

Cada una de las neuronas de la capa oculta tiene una salida y_j^o que está dada por la ecuación 1.19.

$$y_j^o = f^o \left(\sum_i w_{ji}^o x_i + \theta_j^o \right) \quad (1.19)$$

f^o .- Función de transferencia de la capa oculta

Capítulo 1. Introducción a las Redes Neuronales Artificiales

Las salidas y^o_j de las neuronas de la capa oculta son las entradas a los pesos de conexión de la capa de salida, $y^o_j \Rightarrow n^s_j$. Este comportamiento está descrito por la ecuación 1.20.

$$n^s_k = \sum_i w^s_{kj} y^o_j + \theta^s_k \quad (1.20)$$

w^s_{kj} .- Peso que une la neurona j de la capa oculta con la neurona k de la capa de salida

y^o_j .- Salida de la neurona j de la capa oculta

θ^s_k .- Umbral de la neurona k de la capa de salida

n^s_k .- Entrada neta a la neurona k de la capa de salida

(s).- Representa la capa a la que pertenece cada parámetro; capa de salida

La red produce una salida final descrita por la ecuación (1.21)

$$y^s_k = f^s(n^s_k) \quad (1.21)$$

f^s .- Función de transferencia de la capa de salida

Remplazando la ecuación (1.20) en la ecuación (1.21) se obtiene la salida de la red en función de la entrada neta y de los pesos de conexión con la última capa oculta (ecuación 1.22).

$$y^s_k = f^s\left(\sum_i w^s_{kj} y^o_j + \theta^s_k\right) \quad (1.22)$$

La salida de la red y^s_k se compara con la salida deseada para calcular el error en cada unidad de salida

$$\delta_k = (t_k - y^s_k) \quad (1.23)$$

t_k .- Salida conocida

El error debido a cada patrón p propagado por la red está dado por la ecuación (1.24)

$$ep^2 = \frac{1}{2} \sum_k (\delta_k)^2 \quad (1.24)$$

ep^2 .- Error medio cuadrático para cada patrón de entrenamiento p

δ_k .- Error de la neurona k de la capa de salida

Este proceso se repite para el número total de patrones de entrenamiento (r). Para un proceso exitoso el objetivo del algoritmo es actualizar todos los pesos y ganancias de la red minimizando el error medio cuadrático total, descrito por la ecuación (1.25).

$$e^2 = \sum_{p=1}^r ep^2 \quad (1.25)$$

e^2 .- Error total en el proceso de aprendizaje en una iteración después de haber presentado a la red r patrones de entrenamiento

El error de una red neuronal en función de sus pesos genera un espacio de n dimensiones, donde n es el número de pesos de conexión de la red, al evaluar el gradiente del error en un punto de esta superficie se obtendrá la dirección en la cual la función del error tendrá mayor crecimiento. Como el objetivo del proceso de aprendizaje es minimizar el error debe tomarse la dirección negativa para obtener el mayor decremento del error y de esta forma su minimización. Con esta condición es posible actualizar la matriz de pesos de la red (ecuación 1.26).

$$w_{k+1} = w_k - \nabla ep^2 \quad (1.26)$$

∇ep^2 .- Es el gradiente negativo de ep^2 , se calcula como la derivada del error respecto a todos los pesos de la red

En la capa de salida el gradiente del error con respecto a los pesos es:

$$\frac{\partial ep^2}{\partial w_{kj}^s} = \frac{\partial ep^2}{\partial n_k^s} \frac{\partial n_k^s}{\partial w_{kj}^s} \quad (1.27)$$

En la capa oculta el gradiente del error con respecto a los pesos es:

$$\frac{\partial ep^2}{\partial w_{ji}^o} = \frac{\partial ep^2}{\partial n_j^o} \frac{\partial n_j^o}{\partial w_{ji}^o} \quad (1.28)$$

El error para cada neurona en función de su estado de activación se denomina sensibilidad y está dado por la ecuación 1.29.

$$S_i = \frac{\partial ep^2}{\partial n_i} \quad (1.29)$$

Al remplazar la ecuación 1.29 en 1.27 y 1.28, y derivando se tiene:

$$\frac{\partial ep^2}{\partial w_{kj}^s} = \frac{\partial ep^2}{\partial n_k^s} \frac{\partial n_k^s}{\partial w_{kj}^s} = S_k a_j^o \quad (1.30)$$

$$\frac{\partial ep^2}{\partial w_{ji}^o} = \frac{\partial ep^2}{\partial n_j^o} \frac{\partial n_j^o}{\partial w_{ji}^o} = S_i x_i \quad (1.31)$$

El cálculo de la sensibilidad para las unidades ocultas se realiza mediante:

$$S_k = \frac{\partial ep^2}{\partial n_k^s} = \frac{\partial ep^2}{\partial \delta_k} \frac{\partial \delta_k}{\partial y_k^s} \frac{\partial y_k^s}{\partial n_k^s} = (-1) \delta_k f'(n_k^s) \quad (1.32)$$

Al remplazar la ecuación 1.32 en 1.30 se tiene:

$$\frac{\partial ep^2}{\partial w_{kj}^s} = (-1)\delta_k f'(n_k^s)y_j^o \Rightarrow -\nabla ep^2 = \delta_k f'(n_k^s)y_j^o \quad (1.33)$$

Una vez conocido el gradiente del error es posible actualizar los pesos de la red. Al remplazar la ecuación 1.33 en 1.26 se tiene:

$$w_{k+1} = w_k + \delta_k f'(n_k^s)y_j^o \quad (1.34)$$

La ecuación 1.34 representa como los pesos se actualizan de tal manera que el error se minimice. Nótese que esta ecuación es solo válida para los pesos de la capa oculta.

Si se aplica la misma metodología para las unidades ocultas, en general la sensibilidad quedaría representada por:

$$S_k = -f'(n_k^s)(t_k - y_k^s), \text{ para la capa de salida} \quad (1.35)$$

$$S^o = -f'(n^o)(w^{o+1})^T S^{o+1}, \text{ para las capas ocultas} \quad (1.36)$$

1.13 Modos de entrenamiento

Durante la aplicación del algoritmo "Backpropagation", el aprendizaje se produce mediante la presentación sucesiva de un conjunto de entrenamiento. Cada presentación completa al perceptrón multicapa del conjunto de entrenamiento se denomina "epoch". Así, el proceso de aprendizaje se repite "epoch" tras "epoch" hasta que los pesos sinápticos se estabilizan y el desempeño de la red converge a un valor aceptable.

La forma en que se actualizan los pesos sinápticos da lugar a dos modos de entrenamiento distintos, cada uno con sus ventajas y desventajas.

1.13.1 Modo secuencial

En este modo de entrenamiento la actualización de los pesos sinápticos se produce tras la presentación de cada ejemplo de entrenamiento, también es conocido como modo por patrón. Si un conjunto de entrenamientos posee N ejemplos, el modo secuencial de entrenamiento tiene como resultado N correcciones de pesos sinápticos durante cada "epoch".

1.13.2 Modo "batch"

En este modo de entrenamiento la actualización de los pesos sinápticos se produce una única vez, tras la presentación de todo el conjunto de entrenamiento. Para cada "epoch" se calcula el error medio cuadrático (1.24) producido por la red. La variación de los peso sinápticos, para un conjunto

de entrenamiento de N ejemplos, se puede calcular a partir de las ecuaciones (1.27), (1.28) y (1.29) como:

$$\Delta w_{ji} = -\alpha \frac{\partial e}{\partial w_{ji}} = -\frac{\alpha}{N} \sum_{n=1}^N e_j \frac{\partial e_j}{\partial w_{ji}} \quad (1.37)$$

Después, la derivada se puede definir de la manera que se definió antes.

Si los patrones de entrenamiento se presentan a la red de manera aleatoria, el modo de entrenamiento secuencial convierte a la búsqueda en el espacio de pesos en estocástica por naturaleza, y disminuye la probabilidad de que el algoritmo "Backpropagation" quede atrapado en un mínimo local; sin embargo, la naturaleza estocástica del modo de entrenamiento secuencial dificulta el establecimiento de condiciones teóricas para la convergencia del algoritmo.

Por su parte, el uso del modo de entrenamiento "batch" provee una estimación precisa del vector gradiente, garantizando de esta manera la convergencia hacia el mínimo global.

1.14 Aceleración del aprendizaje

El algoritmo "Backpropagation" ha sido utilizado con éxito en gran cantidad de aplicaciones; sin embargo, el éxito y la velocidad de convergencia de este mecanismo de entrenamiento tienen un alto grado de dependencia de la configuración del mismo. Por ello se han realizado una serie de métodos que permiten mejorar significativamente el desempeño del algoritmo.

- **Modo de actualización.** La actualización secuencial es más rápida computacionalmente y demanda menos recursos que la actualización "batch". Esto es especialmente cierto cuando el conjunto de datos es grande y altamente redundante.
- **Conjunto de datos.** La calidad del conjunto de datos es un factor muy importante y se debe tener en cuenta. Cada caso debe cumplir con las siguientes dos premisas:
 - Minimizar el error de entrenamiento
 - Maximizar la información
- **Presentación de los casos.** El modo en que se presentan los casos a resolver es otro factor importante. La aleatorización del orden en que se presentan los ejemplos en los distintos "epochs" evita que los resultados se vean distorsionados por el orden de los casos.
- **Valores objetivos.** La selección de los valores objetivos debe hacerse de acuerdo con la función de activación seleccionada. Una técnica que permite acelerar el tiempo de aprendizaje es desplazar el valor objetivo del valor máximo de la función. El algoritmo "Backpropagation" tiende a saturar las neuronas ocultas cuando el valor objetivo es igual al máximo de la función de

activación. Cuando esto sucede se produce el efecto de parálisis, lo que produce un aumento en el tiempo total de entrenamiento.

- **Normalización de las entradas.** Si bien el algoritmo “Backpropagation” no exige que los valores de entrada a la red se encuentren normalizados, esta es una buena técnica para acelerar los tiempos de entrenamiento (Bishop 1996). La normalización de las entradas debe realizarse de manera tal que el valor medio de la misma se encuentre cercano a cero.

1.15 Generalización

Una vez que se finaliza la fase de aprendizaje, la red puede utilizarse para realizar la tarea para la que fue entrenada. Una de las principales ventajas que posee este modelo es que la red aprende la relación existente entre los datos, adquiriendo la capacidad de generalizar conceptos. De esta manera, una red neuronal puede tratar con información que no le fue presentada durante de la fase de entrenamiento.

Cuando se evalúa una red neuronal no sólo es importante evaluar si la red ha sido capaz de aprender los patrones de entrenamiento. Es imprescindible también evaluar el comportamiento de la red ante patrones nunca antes vistos. Esta característica de las redes neuronales se la conoce como capacidad de generalización y se adquiere durante la fase de entrenamiento. Es necesario que durante el proceso de aprendizaje la red extraiga las características de las muestras, para poder luego responder correctamente a nuevos patrones.

De lo dicho anteriormente surge la necesidad de evaluar durante la fase de entrenamiento dos tipos de errores. El error de aprendizaje, que indica la calidad de la respuesta de la red a los patrones de entrenamiento, y el error de generalización, que indica la calidad de la respuesta de la red a patrones nunca antes vistos. Para poder obtener una medida de ambos errores es necesario dividir el conjunto de datos disponibles en dos, el conjunto de datos de entrenamiento, y el conjunto de datos de evaluación. El primero se utiliza durante la fase de entrenamiento para que la red pueda extraer las características de los mismos y, mediante el ajuste de sus pesos sinápticos, la red logre una representación interna de la función. El conjunto de evaluación se utiliza para evaluar la capacidad de generalización de la red.

La causa más común de la pérdida de capacidad de generalización es el sobreaprendizaje. Esto sucede cuando la cantidad de ciclos de entrenamientos tiende a ser muy alta. Se observa que la respuesta de la red a los patrones de entrenamiento es muy buena mientras que la respuesta a nuevos patrones tiende a ser muy pobre. Al aumentar el número de ciclos la red tiende a sobreajustar la respuesta a los patrones de entrenamiento, a expensas de una menor capacidad de generalización. La Figura 1.12 muestra gráficamente la situación de lo dicho anteriormente. En la Figura 1.12 se observa que en un determinado punto la red comienza a perder capacidad de generalización como consecuencia del sobreaprendizaje de los patrones de entrenamiento.

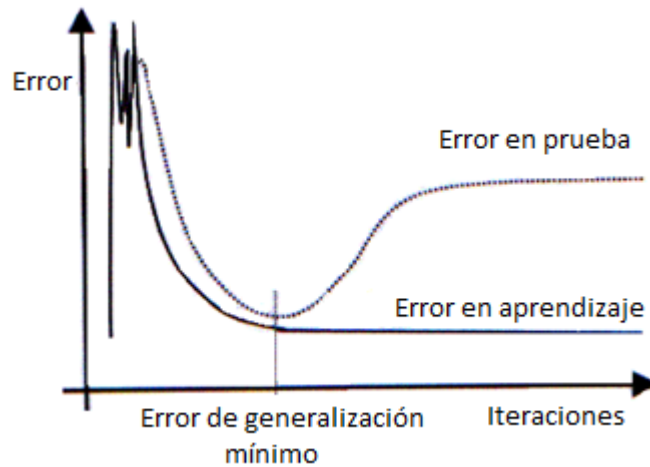


Figura 1.12 Errores en función del número de iteraciones

En la Figura 1.13 se muestra una situación más real del mismo caso. A medida que transcurre el proceso de aprendizaje se obtienen varios mínimos sobre el conjunto de evaluación. Existen diversas técnicas de parada temprana (“early stopping”) aunque en la mayoría de los casos se deja que el proceso de aprendizaje avance hasta alcanzar una cota de error razonable, guardando periódicamente las distintas configuraciones intermedias, para luego seleccionar la de menor error de evaluación.

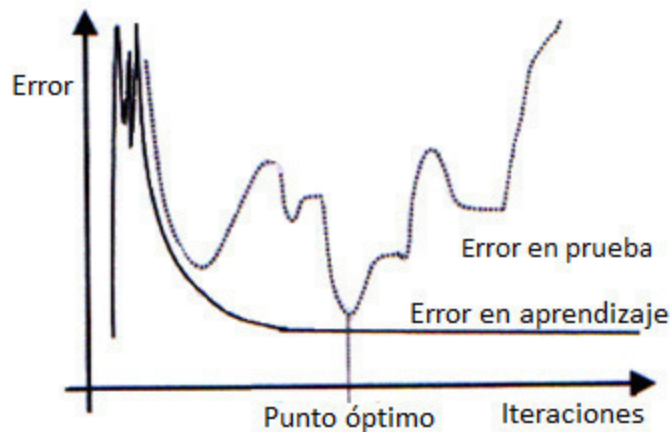


Figura 1.13 Generalización. Situación real

En ocasiones la pérdida de capacidad de generalización se produce por el uso excesivo de neuronas ocultas en la red neuronal. Esto hace que la red tienda a ajustar con mucha exactitud los patrones de entrenamiento, evitando que la red extraiga las características del conjunto. Este problema se ve agravado cuando los patrones de entrenamiento poseen ruido, ya que la red ajusta también el ruido de los mismos.

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

2.1 Introducción

En el primer capítulo se estudió con detalle la teoría y desarrollo de las Redes Neuronales Artificiales (RNA). El capítulo 2 se dedica a aplicar los conocimientos adquiridos en el capítulo anterior. Las bondades y ventajas que las RNA ofrecen en la solución de problemas específicos son muchas, se pueden abordar problemas de gran esfuerzo computacional, y utilizar las RNA como una herramienta de predicción.

En este capítulo se aplicarán las RNA para encontrar factores de transformación entre respuestas máximas, como son: distorsiones máximas de entrepiso y ductilidades globales, entre sistemas estructurales de múltiples grados de libertad (SMGL) y sus correspondientes sistemas de un grado de libertad (S1GL) equivalentes, garantizando una misma tasa anual de excedencia entre ambos sistemas. Utilizando el modelo de RNA es posible encontrar funciones de transformación entre SMGL y S1GL con una aproximación aceptable o factores de transformación específicos para un sistema dado, con una significativa reducción en el esfuerzo computacional.

La formulación de nuevos códigos de diseño sísmico que consideren la confiabilidad estructural depende de modelos práctico-aplicables que estimen la tasa anual de falla de sistemas estructurales utilizados comúnmente. La fácil aplicación de los códigos de diseño es uno de sus principales requisitos. Estimar la confiabilidad para casos típicos de Ingeniería no es una tarea fácil ni práctica, por lo que es necesario utilizar otras alternativas.

En casi todos los reglamentos del mundo el diseño estructural se basa en confiabilidades que no son pre-establecidas, y se concentran en satisfacer restricciones deterministas como las distorsiones máximas de entrepiso.

El Reglamento de Construcciones del Distrito Federal no establece condiciones para que un diseño estructural tenga un valor de confiabilidad prescrito, por lo que las estructuras construidas en el Distrito Federal tienen tasas anuales de falla inciertas. La tendencia es que en el futuro las normas se orienten no solo a garantizar un nivel de desempeño satisfactorio, sino también un nivel de confiabilidad adecuado en las estructuras sismo-resistentes, ante distintos estados límite.

2.2 Factores de transformación entre SMGL y S1GL con misma tasa anual de falla

En esta parte de la tesis se obtienen factores de transformación para estimar la respuesta (en términos de ductilidades y distorsiones máximas de entrepiso) de sistemas estructurales de múltiples grados de libertad (SMGL) a partir de la respuesta de S1GL equivalentes, garantizando una misma tasa anual de falla de ambos sistemas. Primero, se presentan los factores de

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

transformación correspondientes a varios edificios de acero, de cuatro a catorce niveles de altura, diseñados con el Reglamento de Construcción del Distrito Federal (RCDF 2004). Los sistemas estructurales se sujetan a la acción de nueve movimientos sísmicos registrados en terreno blando del Valle de México. La estimación de factores de transformación requiere de un gran número de análisis dinámicos no lineales. Aquí se utiliza un nuevo enfoque para obtener estos factores mediante un modelo de RNA. El modelo demuestra que el uso de la teoría de la RNA puede utilizarse para obtener estos factores de transformación con suficiente aproximación, reduciendo significativamente el esfuerzo computacional, ya que se necesitan pocos análisis dinámicos no lineales. Los resultados obtenidos del modelo de RNA se resumen en funciones aplicables a la estimación de factores de transformación para una misma tasa anual de excedencia de cierta respuesta estructural. Los factores de transformación que aquí se proponen podrían incluirse en la nueva generación de criterios de diseño sísmico basado en confiabilidad estructural.

2.2.1 Objetivo de este capítulo

El objetivo de este capítulo es obtener factores de transformación (FT) que relacionen la respuesta en términos de ductilidad global y de distorsión máxima de entrepiso (que son los parámetros más utilizados para representar la respuesta de las estructuras sujetas a la acción de los sismos), de manera que se garantice la misma tasa anual de excedencia entre SMGL y sus correspondientes S1GL. Para alcanzar este objetivo se utilizan modelos de RNA.

2.2.2 Modelos de marcos estructurales

En este trabajo se analizan cinco marcos estructurales de acero de 4, 6, 8, 10 y 14 niveles diseñados con el Reglamento de Construcciones del Distrito Federal (2004). Para todos los marcos se utilizó un factor de comportamiento sísmico de $Q=3$ en el diseño. Como se muestra en la Figura 2.1, los marcos tienen tres crujías de 8m cada una, y una altura de entrepiso de 3.5m. Se usó acero A-36 en las vigas y columnas de los marcos. Las características geométricas globales de los marcos se muestran en la Figura 2.1.

Para cada marco estructural se obtuvo un S1GL equivalente que representa sus características dinámicas. La manera en que se obtuvieron los S1GL equivalentes fue haciendo que fueran iguales tanto el periodo de vibración, como el cortante basal a la fluencia y el porcentaje de amortiguamiento crítico de los marcos estructurales de acero y los S1GL (5% del amortiguamiento crítico). Tanto el cortante basal resistente como el desplazamiento de fluencia de cada marco se obtuvieron mediante un análisis de empuje lateral estático no-lineal (“push-over”) utilizando el programa RUAUMOKO (Carr 2000) (ver Figura 2.2). Para el análisis, se consideró un modelo histerético bilineal con 3% de rigidez de post-fluencia.

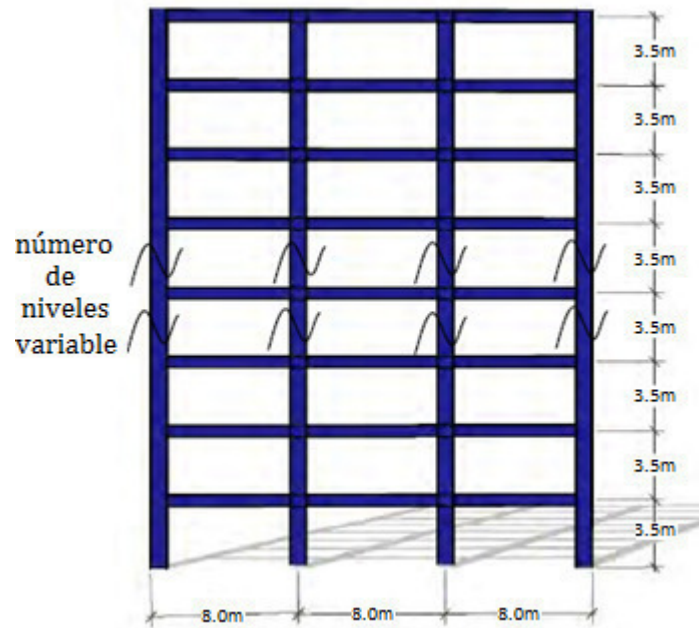


Figura 2.1 Geometría de los marcos utilizados

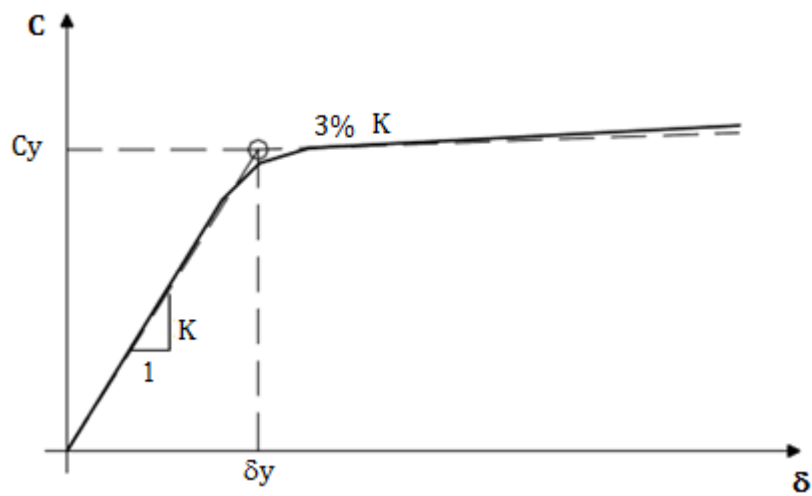


Figura 2.2 Gráfica coeficiente basal vs desplazamiento de azotea. Modelo de comportamiento bilineal

Las características dinámicas más importantes de los marcos se resumen en la Tabla 2.1, en donde T_0 representa el periodo fundamental de vibración de la estructura, C_y el coeficiente sísmico resistente, y δ_y el desplazamiento global de fluencia.

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

Tabla 2.1 Periodo fundamental de vibración (T_0), coeficiente sísmico resistente (C_y) y desplazamiento de fluencia ($\delta\gamma$), correspondientes a los marcos analizados

Marco	T_0 (s)	C_y	$\delta\gamma$ (m)
M4	0.9	0.48	0.13
M6	1.07	0.42	0.17
M8	1.2	0.38	0.19
M10	1.37	0.36	0.23
M14	1.91	0.25	0.3

2.2.3 Movimientos sísmicos

Se utilizaron nueve movimientos sísmicos registrados en la zona IIIb del Valle de México cuya ubicación se ilustra en la Figura 2.3. Dichos registros corresponden al sismo de Colima de 1995 que tuvo una magnitud de 7.3. La Figura 2.4 muestra los espectros de respuesta elásticos (para 5% de amortiguamiento crítico) de los nueve registros utilizados en este estudio. En la Tabla 2.2 se muestran los datos correspondientes a los registros utilizados.

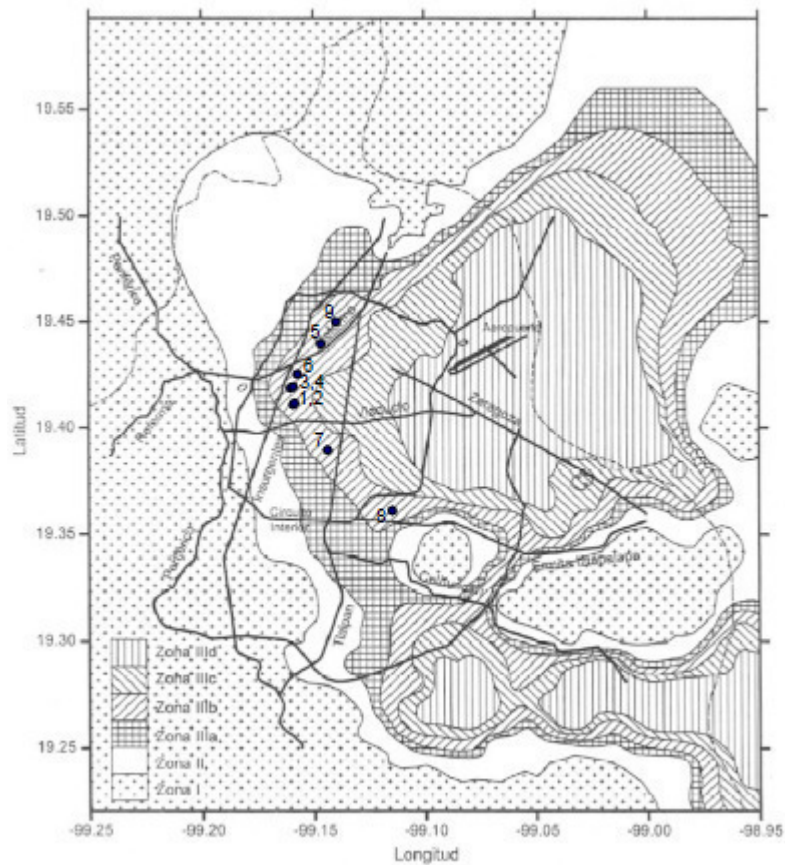


Figura 2.3 Ubicación de los registros sísmicos en el Valle de México

Tabla 2.2 Registros sísmicos utilizados en este estudio

Registro	Clave de Estación	Institución encargada	Coordenadas de la Estación		Nombre de la Estación	Aceleración máxima del suelo (cm/s ²)
			Latitud	Longitud		
1	CJ03	FICA	19.41	99.157	CU Juárez	14.4
2	CJ04	IIUNAM	19.41	99.157	Centro urbano Presidente Juárez	15.7
3	CO56	CIRES	19.422	99.159	Córdoba	24.9
4	EJCL	IIUNAM	19.423	99.159	Edif. Jalapa	12.3
5	GA62	CIRES	19.439	99.14	Ganbaldi	11
6	LI58	CIRES	19.426	99.157	Liverpool	17.6
7	PE10	CIRES	19.39	99.132	Plutarco Elias Calles	19.2
8	SP51	CIRES	19.366	99.119	Sector Popular	12.9
9	TL08	CIRES	19.45	99.134	Tlatelolco	10.1

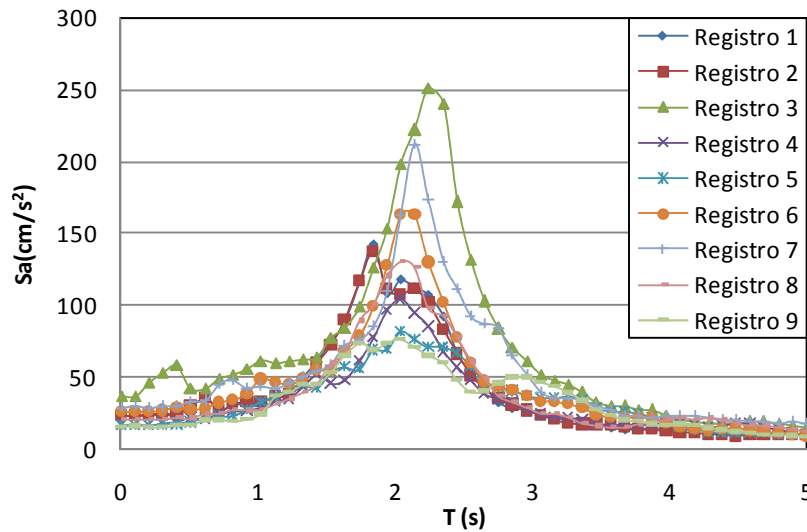


Figura 2.4 Espectros de respuesta elástica de los registros sísmicos

2.2.3.1 Escalamiento de sismos

Con el fin de contar con diferentes intensidades sísmicas, los registros se multiplicaron por diferentes factores de escala. Dichos factores están dados por la relación que existe entre la pseudoaceleración correspondiente al periodo fundamental de vibración del sistema en estudio y la pseudoaceleración correspondiente a la intensidad a la que se desea escalar el movimiento (Shome y Cornell 1999, Chan et al. 2005). El factor se multiplica por la historia completa de aceleraciones del registro en estudio. En la Figura 2.5 se muestran los espectros elásticos correspondientes a un marco estructural con periodo de vibración fundamental $T_0 = 1.20s$ (marco M8), escalados a una pseudoaceleración $S_a = 500$ gals. Puede observarse que todos los espectros elásticos coinciden en el punto correspondiente al periodo $T = 1.2s$, la forma espectral es muy similar al ser escalados para el periodo fundamental del marco; las mismas observaciones son

válidas para el caso de los otros periodos correspondientes al periodo fundamental de los marcos estudiados.

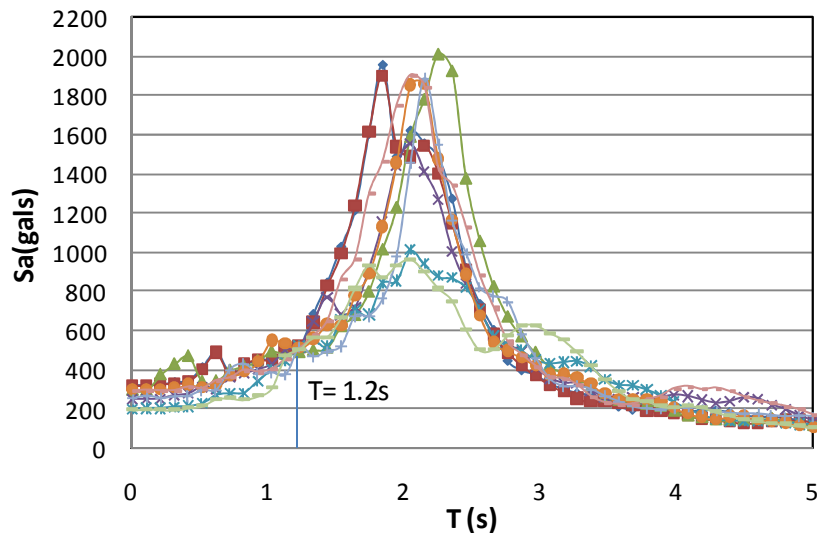


Figura 2.5 Espectros elásticos de los registros escalados a $S_a = 500$ gals

2.2.4 Curvas de peligro de demanda estructural

En esta sección se presenta la metodología a seguir para evaluar la confiabilidad estructural, medida a través de las curva de peligro de demanda de los edificios, a partir de sus curvas de vulnerabilidad sísmica y de las curvas de peligro sísmico del sitio de interés.

Se analizan las respuestas estructurales correspondientes tanto a ductilidades globales máximas como a las distorsiones máximas de entrepiso.

2.2.4.1 Demandas de ductilidad global en función de la intensidad sísmica

Para estimar las ductilidades globales máximas fue necesario obtener, primero, los desplazamientos máximos de azotea por cada uno de los movimientos sísmicos indicados en la Tabla 2.2, escalados desde una intensidad mínima hasta 1500 gals, a cada 100 gals. Para obtener las ductilidades globales máximas los desplazamientos globales máximos se dividieron entre los desplazamientos de fluencia (δ_y) correspondientes a cada marco (ver Tabla 2.1).

Se consideró que la distribución de ductilidades globales máximas correspondiente a un mismo nivel de intensidad presenta una función lognormal (Shome y Cornell 1999). Para cada nivel de intensidad se calculó la mediana (D_μ) y la desviación del logaritmo natural de la ductilidad global ($\sigma_{n\mu}$) en función de la intensidad, correspondiente a cada marco. Los resultados se muestran en las Figuras 2.6 a 2.10.

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

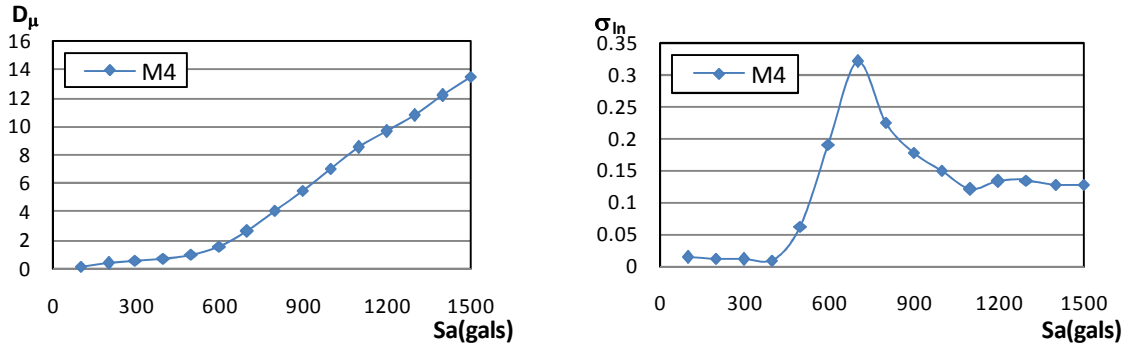


Figura 2.6 Medianas y desviaciones logarítmicas de las ductilidades para el marco M4

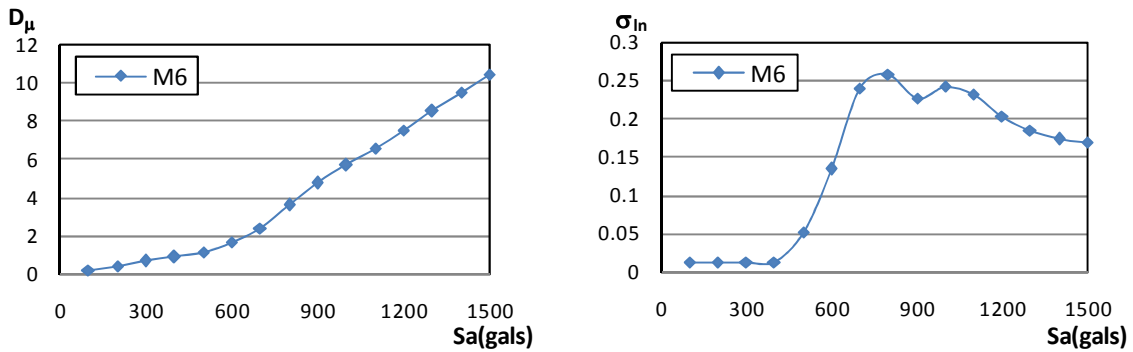


Figura 2.7 Medianas y desviaciones logarítmicas de las ductilidades para el marco M6

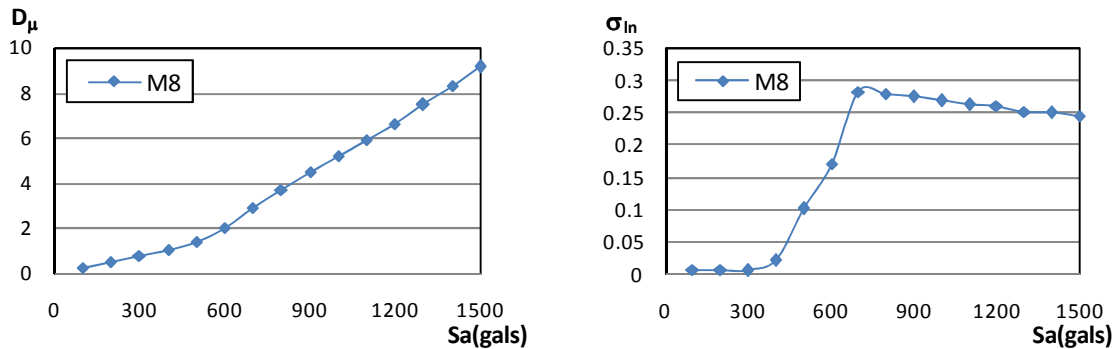


Figura 2.8 Medianas y desviaciones logarítmicas de las ductilidades para el marco M8

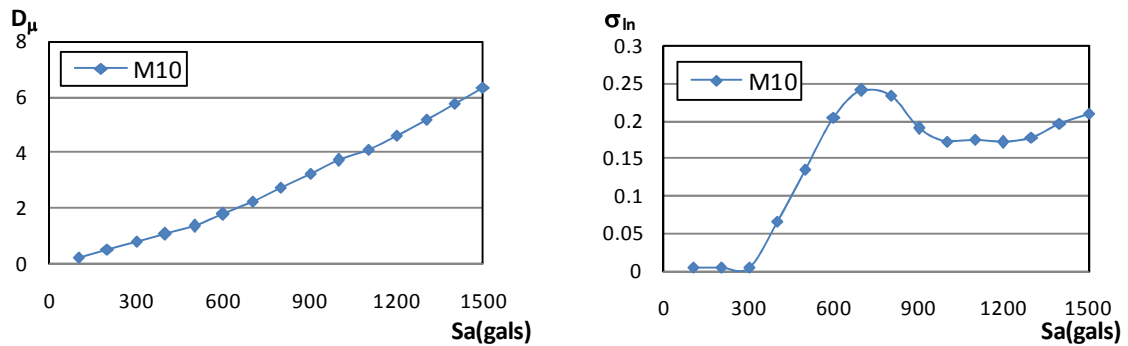


Figura 2.9 Medianas y desviaciones logarítmicas de las ductilidades para el marco M10

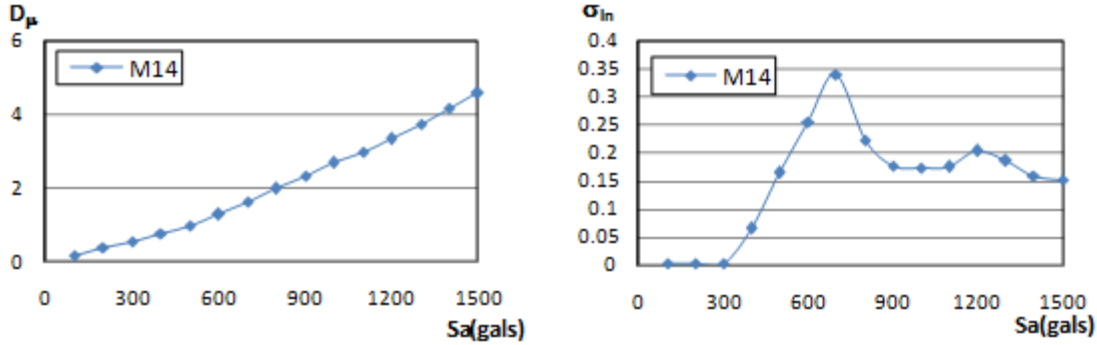


Figura 2.10 Medianas y desviaciones logarítmicas de las ductilidades para el marco M14

En las gráficas que aparecen a la izquierda de las Figuras 2.6 a 2.10 se observa que el valor de las medianas va incrementando proporcionalmente a la intensidad del movimiento. La variación entre el nivel de respuesta de los marcos estudiados, para una misma intensidad (S_a), es tal que la respuesta va disminuyendo conforme se incrementa la altura de los edificios. Esto es, para un mismo nivel de intensidad, se desarrollan mayores ductilidades en el edificio de cuatro niveles que en el edificio de 14 niveles.

En las gráficas que aparecen a la derecha de las Figuras 2.6 a 2.10 se advierte que para intensidades pequeñas (donde las estructuras permanecen dentro de su intervalo de comportamiento lineal) todos los sismos demandan aproximadamente el mismo nivel de respuesta global, por lo que es casi nula su desviación estándar ($\sigma_{ln\mu}$); sin embargo, a medida que la intensidad (S_a) se incrementa, los edificios presentan un comportamiento inelástico lineal y los valores de ductilidad muestran mayor dispersión, para el mismo nivel de intensidad.

2.2.4.2 Curvas de vulnerabilidad sísmica

Primero se obtienen curvas de peligro de demanda de ductilidad máxima global y, posteriormente, las de desplazamientos máximos de entrepiso. En ambos casos, se supuso que la respuesta máxima para cierta intensidad tiene una función de densidad de probabilidad lognormal (Shome y Cornell 1999).

Sobre esta suposición, se obtiene un conjunto de puntos que definen la probabilidad (P) de que la respuesta máxima μ_{max} sea mayor que cierto valor μ_g , dada una intensidad S_a

$$P(\mu_{max} \geq \mu_g | S_a) \tag{2.1}$$

Posteriormente, se ajustan los parámetros de la siguiente expresión que define las curvas de vulnerabilidad sísmica

$$P(\mu_{max} \geq \mu_g | S_a) = 1 - 0.1 \left(\frac{S_a/g}{a} \right)^b \tag{2.2}$$

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

Las curvas de vulnerabilidad correspondientes a ductilidades globales que se obtuvieron con este procedimiento, indican que a medida que crece el número de niveles, la separación entre dichas curvas se incrementa. Debido a lo anterior, la probabilidad de exceder cierta ductilidad (μ) para niveles altos de intensidad sísmica es mayor para el edificio de cuatro niveles que para el de diez. Como ejemplo de lo anterior, la Tabla 2.3 presenta la probabilidad de exceder una ductilidad igual a 3, correspondiente a cada uno de los edificios en estudio, dada una intensidad $S_a = 800$ gals.

Tabla 2.3 Probabilidad de exceder $\mu = 3$ con un nivel de intensidad $S_a = 800$ gals

Marco	$P(\mu_{max} \geq \mu_g S_a)$
M4	0.92
M6	0.77
M8	0.45
M10	0.35
M14	0.17

2.2.4.3 Curvas de peligro de demanda de ductilidad global

La curva de peligro de demanda para cada uno de los cinco marcos se obtuvo utilizando la siguiente expresión (Esteva 1967, Cornell 1968)

$$v_\mu(\mu) = \int \left| \frac{dv(S_a)}{d(S_a)} \right| P(\mu_{max} \geq \mu | S_a) d(S_a) \quad (2.3)$$

donde

- μ_{max} representa el parámetro considerado.
- $v_\mu(\mu)$ es el número de veces por año que se excede el parámetro μ
- S_a es la pseudoaceleración.
- $P(\mu_{max} \geq \mu | S_a)$ (curva de vulnerabilidad) representa la probabilidad condicional de que el valor μ_{max} en una estructura exceda μ , dada una intensidad.
- $v(S_a)$ es el número de veces por año que ocurre una intensidad mayor que S_a . Este parámetro representa la curva de peligro sísmico del sitio de interés. En este estudio, se utilizaron las curvas de peligro sísmico para el sitio SCT (Alamilla, 2001).

La Figura 2.11 muestra las curvas de peligro de demanda de ductilidad global correspondientes a los cinco marcos estudiados.

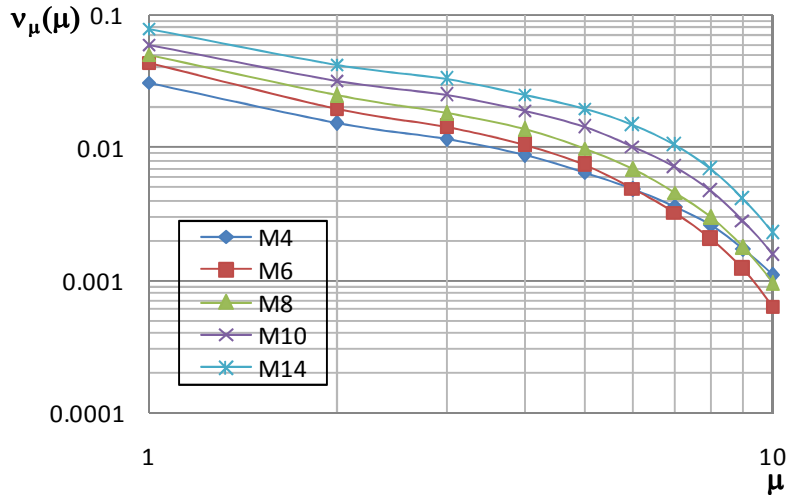


Figura 2.11 Curvas de peligro de demanda de ductilidad global

2.2.4.4 Curvas de peligro de distorsión máxima de entrepiso

Con base en la ec. 2.3, también se obtuvieron curvas de peligro de distorsión máxima de entrepiso (γ) de los cinco marcos de acero (en este caso el parámetro μ se sustituye por γ en la ec. 2.3). La Figura 2.12 ilustra las curvas correspondientes a los cinco marcos. Se observa que la tasa de excedencia anual para un valor de $\gamma = 0.003$, se encuentra entre 0.07 y 0.13 (que corresponden a periodos de retorno T_R entre 8 y 14 años). Por otro lado, la correspondiente tasa de excedencia anual para un valor de $\gamma = 0.03$ se encuentra entre 0.00055 y 0.0029 (o sea $360 < T_R < 1800$ años).

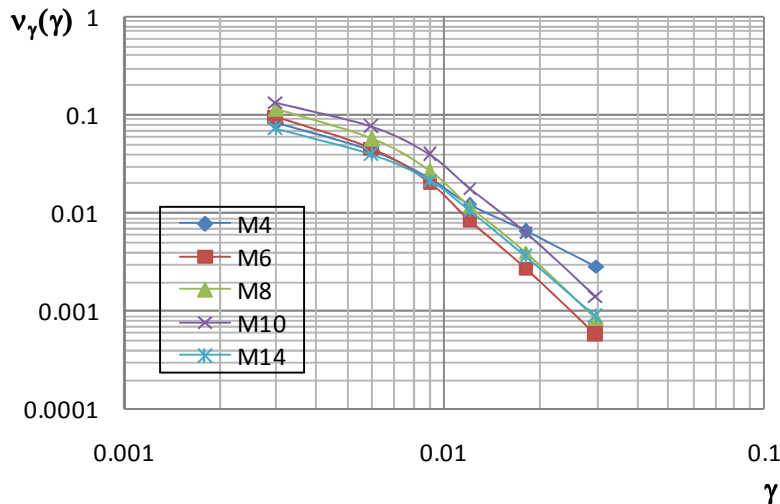


Figura 2.12 Curvas de peligro de distorsión máxima de entrepiso

2.2.4.5 Curvas de peligro de demanda de ductilidad global y distorsión máxima de entrepiso para S1GL equivalentes

Por otro lado, se obtuvieron curvas de peligro (ductilidad global y distorsión máxima de entrepiso) para los sistemas equivalentes de un grado de libertad (S1GL) de los cinco marcos de acero en

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

estudio. Las condiciones que se establecen para los (S1GL) son que el coeficiente sísmico de fluencia C_y , su periodo fundamental de vibración T y su porcentaje de amortiguamiento crítico ξ sean los mismos que los de los edificios de acero de múltiples grados de libertad (SMGL). Con las cuales se obtienen los factores de transformación que se definen a continuación.

2.2.5 Factores de transformación

La probabilidad anual de excedencia de un sistema de MGDL puede estimarse si se conoce la probabilidad anual de excedencia de su S1GL equivalente. Para ello, es necesario conocer la relación entre las respuestas de ambos sistemas estructurales para una misma tasa de excedencia v_0 , que en este trabajo se denominan factores de transformación FT . Estos toman en cuenta la diferencia entre ambos sistemas estructurales (ver Figura 2.13), dada la misma probabilidad de excedencia.

En estudios previos, se han calculado funciones de transformación que relacionen las respuestas máximas entre marcos estructurales de MGDL y S1GL en términos de distintos parámetros de excedencia (Inoue y Cornell 1991, Esteva *et al.* 2005, Bojórquez *et al.* 2005) para una misma tasa de excedencia; sin embargo, hasta ahora no se ha obtenido ninguna función de transformación que relacione las respuestas en términos de distorsiones máximas de entrepiso y que sean aplicables a edificios de mediana altura ubicados en la zona blanda del valle de México. Tampoco se han utilizado RNA para este tipo de estudios.

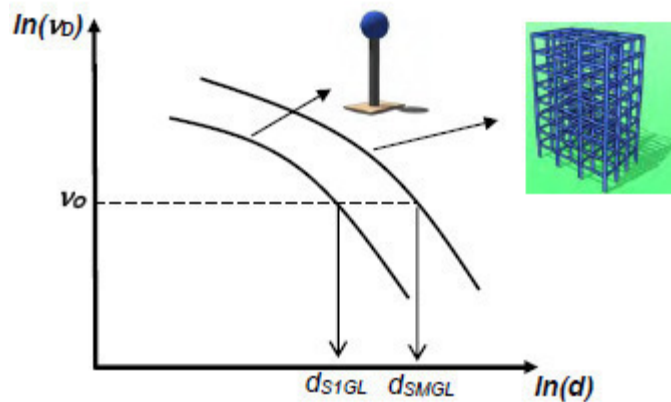


Figura 2.13 Ejemplo de curvas de peligro de demanda estructural para un SMGL y su correspondiente S1GL equivalente

2.2.5.1 Factores de transformación de ductilidad, $FT\mu$

Debido a que la demanda de ductilidad en una estructura real puede diferir de la de un S1GL equivalente, es deseable considerar las incertidumbres en la respuesta de una estructura real de múltiples grados de libertad y la de un S1GL; es decir se trata de representar las características de dicha estructura, mediante un factor de transformación apropiado.

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

El factor de transformación de ductilidades se obtiene mediante la siguiente relación;

$$FT_{\mu} = \frac{\mu_{SMGL}(\nu_0)}{\mu_{S1GL}(\nu_0)} \quad (2.4)$$

donde

- FT_{μ} es el factor correspondiente a la ductilidad global
- $\mu_{SMGL}(\nu_0)$ es la ductilidad global en el marco estructural de acero asociada a una tasa anual de excedencia ν_0
- $\mu_{S1GL}(\nu_0)$ es la ductilidad en el S1GL equivalente correspondiente a una tasa anual de excedencia ν_0

En la Figura 2.14 se observan los factores de transformación de ductilidad obtenidos para los cinco marcos y sus S1GL equivalentes analizados en este trabajo. El eje vertical indica el valor del factor de transformación definido como la relación entre la ductilidad desarrollada en un marco estructural de acero, dividido entre la ductilidad global desarrollada por su respectivo S1GL equivalente, para un mismo nivel de tasa anual de excedencia. La Figura 2.14 muestra que los valores son muy cercanos a la unidad. Además, los valores obtenidos son similares sin importar el número de niveles de los marcos estructurales de acero.

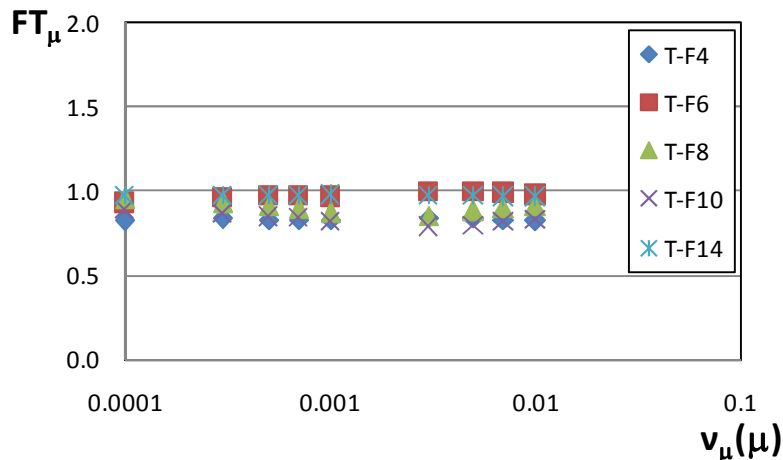


Figura 2.14 Factores de transformación de ductilidad FT_{μ} , asociados a distintas tasas de excedencia

2.2.5.2 Factores de transformación de distorsiones máximas de entrepiso, FT_{γ}

Enseguida se presentan los factores de transformación de distorsión máxima de entrepiso correspondientes a los marcos estructurales de acero en estudio. Estos factores se determinaron como la relación entre las curvas de peligro de distorsión máxima de entrepiso de los marcos y las correspondientes a las distorsiones del S1GL (distorsión global máxima). Esta última se obtuvo

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

como la relación entre el desplazamiento máximo dividido entre la altura del marco. Los resultados para diferentes valores de tasas de excedencia se ilustran en la Figura 2.15.

En la Figura 2.15 se observa que los valores de factores de transformación de distorsión máxima de entrepiso son alrededor de 1.5. Esto implica que se obtuvieron mayores distorsiones máximas de entrepiso en el SMGL para todo el intervalo de las tasas anuales de excedencia estudiado. *Por lo tanto, utilizar las distorsiones obtenidas de un S1GL (suponiendo distribución lineal en la altura) para representar las distorsiones en marcos estructurales de acero puede llevar a diseños inadecuados.* Esta observación es muy importante por ser una de las respuestas estructurales más utilizadas por los códigos de diseño sísmico para fomentar el desempeño estructural y no estructural de edificios sismo-resistentes.

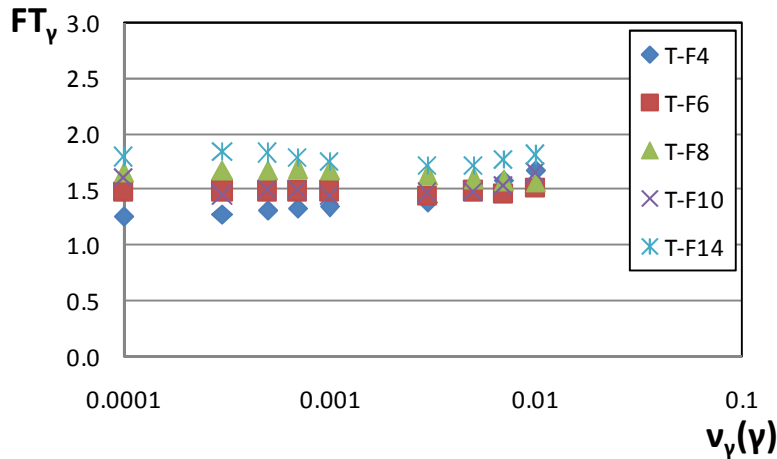


Figura 2.15 Factores de transformación de distorsión máxima de entrepiso FT_{γ} asociados a distintas tasas de excedencia

2.3 Funciones de transformación utilizando redes neuronales artificiales

En esta sección se aplica una nueva metodología basada en RNA que puede usarse para elaborar nuevas reglas de diseño basados en confiabilidad. Ello facilita el esfuerzo computacional obteniendo resultados con una aproximación aceptable. Las bondades que ofrecen los modelos simplificados para representar la respuesta de problemas de complejos son innumerables, tal es el caso de S1GL para representar SMGL. Con este objetivo en mente, esta parte del estudio se enfocará a encontrar funciones de transformación entre los SMGL y S1GL que se han venido analizando.

Primero se obtendrá un modelo de RNA que represente estos factores de transformación, para cada uno de los casos estudiados: FT de ductilidad, y FT de distorsiones máximas de entrepiso, como función de los cuatro parámetros, (mismos que se encontró que representan la respuesta del problema en cuestión). Una vez ajustado el modelo de RNA óptimo se obtendrán funciones de transformación que dependan únicamente de estos cuatro parámetros.

2.3.1 Modelo neuronal artificial

El modelo neuronal artificial surge de la necesidad de ahorro de tiempo computacional para la estimación de la respuesta de diferentes modelos de edificios, existe evidencia de que las RNA pueden utilizarse para la optimización estructural basada en confiabilidad (Papadrakakis 2002), utilizando específicamente el modelo Perceptrón multicapa (“Multi Layer Percetron”) con alimentación hacia adelante, por este motivo se ha seleccionado un modelo de RNA MLP (“feedforward”), para los dos casos estudiados en esta tesis (capítulos 2 y 3).

2.3.1.1 Determinación de las entradas y salidas necesarias de la RNA

Una parte fundamental para el uso correcto de las RNA es la adecuada selección de los parámetros que la representen. Para el caso en estudio se estableció que dichos parámetros son: 1) periodo de la estructura, T_0 , 2) tasa de excedencia, v , 3) coeficiente sísmico resistente, C_y y 4) desplazamiento de fluencia, δy . Estos parámetros son los necesarios para encontrar los FT entre S1GL y SMGL. De acuerdo a lo anterior, se establece que los nodos (neuronas) de entrada al modelo neuronal quedan definidos por los valores de T_0 , v , C_y y δy , y el nodo de salida queda representado por FT. Los valores de T_0 , v , C_y y δy son los obtenidos de los edificios estudiados en la sección anterior (Tabla 2.1). Los valores para la salida de la red son los FT encontrados en la sección 2.2.5 (Figuras 2.14 y 2.15). El éxito de la aproximación funcional hecha con la RNA depende principalmente de la calidad de los datos de entrenamiento. Los intervalos de entrenamiento establecidos para este modelo se muestran en la Tabla 2.3.

Tabla 2.3 Intervalos de valores de las entradas a la red

Entrada	Intervalo
T_0	0.9 a 1.91
v	0.01 a 0.0001
C_y	0.25 a 0.48
δy	0.13 a 0.3

2.3.1.2 Entrenamiento del modelo de RNA

La arquitectura de las redes está basada en una capa de entrada, una capa oculta y una capa de salida. Para el entrenamiento del modelo neuronal que se propone, la capa de entrada quedó constituida por cuatro neuronas que representan a T_0 , v , C_y y δy , y la capa de salida por una sola neurona que representa el FT. Por lo que el conjunto de entrenamiento quedó compuesto por la matriz de datos que se presenta en la Tabla 2.4.

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

v	To	Cy	Δy	Salida		v	To	Cy	Δy	Salida
0.01	0.9	0.48	0.13	0.82		0.01	0.9	0.48	0.13	1.67
0.007	0.9	0.48	0.13	0.83		0.007	0.9	0.48	0.13	1.58
0.005	0.9	0.48	0.13	0.84		0.005	0.9	0.48	0.13	1.5
0.003	0.9	0.48	0.13	0.84		0.003	0.9	0.48	0.13	1.39
0.001	0.9	0.48	0.13	0.83		0.001	0.9	0.48	0.13	1.34
0.0007	0.9	0.48	0.13	0.83		0.0007	0.9	0.48	0.13	1.33
0.0005	0.9	0.48	0.13	0.82		0.0005	0.9	0.48	0.13	1.31
0.0003	0.9	0.48	0.13	0.83		0.0003	0.9	0.48	0.13	1.28
0.0001	0.9	0.48	0.13	0.82		0.0001	0.9	0.48	0.13	1.25
0.01	1.07	0.42	0.17	0.98		0.01	1.07	0.42	0.17	1.51
0.007	1.07	0.42	0.17	0.99		0.007	1.07	0.42	0.17	1.46
0.005	1.07	0.42	0.17	0.99		0.005	1.07	0.42	0.17	1.48
0.003	1.07	0.42	0.17	0.99		0.003	1.07	0.42	0.17	1.44
0.001	1.07	0.42	0.17	0.97		0.001	1.07	0.42	0.17	1.49
0.0007	1.07	0.42	0.17	0.97		0.0007	1.07	0.42	0.17	1.49
0.0005	1.07	0.42	0.17	0.97		0.0005	1.07	0.42	0.17	1.48
0.0003	1.07	0.42	0.17	0.96		0.0003	1.07	0.42	0.17	1.48
0.0001	1.07	0.42	0.17	0.93		0.0001	1.07	0.42	0.17	1.48
0.01	1.91	0.25	0.3	0.97		0.01	1.91	0.25	0.3	1.81
0.007	1.91	0.25	0.3	0.97		0.007	1.91	0.25	0.3	1.76
0.005	1.91	0.25	0.3	0.97		0.005	1.91	0.25	0.3	1.71
0.003	1.91	0.25	0.3	0.97		0.003	1.91	0.25	0.3	1.71
0.001	1.91	0.25	0.3	0.98		0.001	1.91	0.25	0.3	1.74
0.0007	1.91	0.25	0.3	0.97		0.0007	1.91	0.25	0.3	1.78
0.0005	1.91	0.25	0.3	0.97		0.0005	1.91	0.25	0.3	1.82
0.0003	1.91	0.25	0.3	0.97		0.0003	1.91	0.25	0.3	1.83
0.0001	1.91	0.25	0.3	0.97		0.0001	1.91	0.25	0.3	1.79
0.01	1.37	0.36	0.23	0.83		0.01	1.37	0.36	0.23	1.65
0.007	1.37	0.36	0.23	0.81		0.007	1.37	0.36	0.23	1.53
0.005	1.37	0.36	0.23	0.79		0.005	1.37	0.36	0.23	1.47
0.003	1.37	0.36	0.23	0.79		0.003	1.37	0.36	0.23	1.46
0.001	1.37	0.36	0.23	0.81		0.001	1.37	0.36	0.23	1.44
0.0007	1.37	0.36	0.23	0.84		0.0007	1.37	0.36	0.23	1.5
0.0005	1.37	0.36	0.23	0.84		0.0005	1.37	0.36	0.23	1.5
0.0003	1.37	0.36	0.23	0.86		0.0003	1.37	0.36	0.23	1.45
0.0001	1.37	0.36	0.23	0.88		0.0001	1.37	0.36	0.23	1.6

a) Ductilidad máxima global

b) Distorsiones máximas de entrepiso

Tabla 2.4 Pares de entrenamiento en el modelo RNA

En la Tabla 2.4 se presentan los pares de entrada y salida utilizados para entrenar los modelos de RNA. Para el entrenamiento se utilizó el algoritmo “Backpropagation”. La función de transferencia en la capa oculta fue sigmoidea, y para la capa de salida, la función lineal (Tabla 1.1).

Se elaboró un programa en Visual Basic 6 para entrenar a la red (ver Figura 2.16). El entrenamiento se llevó a cabo de la siguiente manera: se inicializaron de manera aleatoria los pesos y umbrales de las conexiones entre las neuronas de cada capa. Teniendo estos pesos se procedió a presentar a la red un par de entrenamiento el cual generó una respuesta de la red, esta se compara con la respuesta esperada. Con el error encontrado, y utilizando el gradiente de la función de error, se actualizaron los pesos entre las conexiones neuronales. Después se presentó a la red nuevos pares de entrenamiento de manera aleatoria, hasta que el error se considera suficientemente pequeño o se hubiese alcanzado el número de iteraciones previamente establecido.

Nótese que en los pares de entrenamiento no se presentó a la red los pares correspondientes al marco M8, ya que este sirvió para comprobar el funcionamiento de la misma. Es importante

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

mencionar que las entradas y salidas de la red nunca fueron normalizadas (es decir no se utilizó la normalización de entradas y salidas como método de aceleración del aprendizaje de la red).

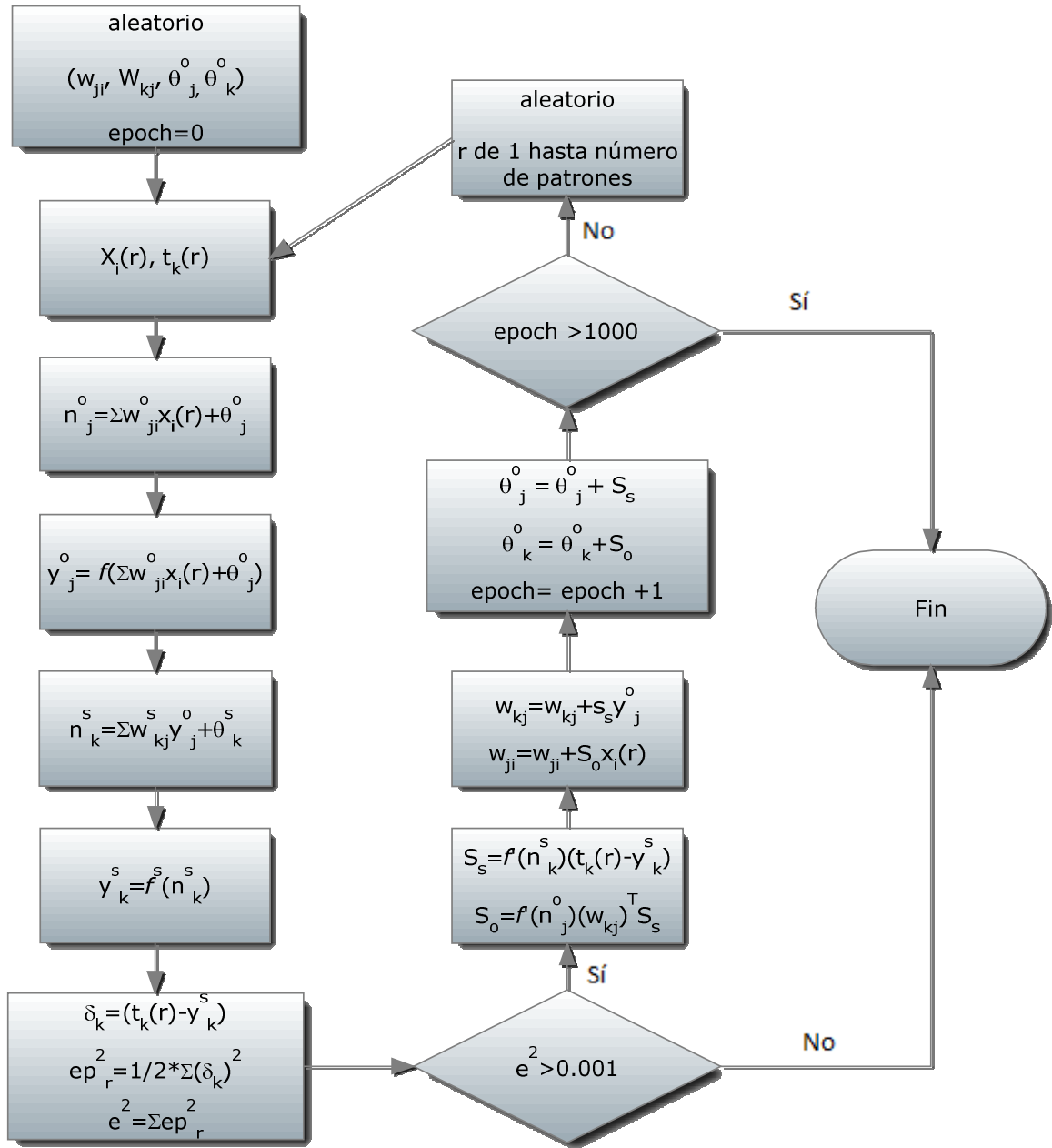


Figura 2.16. Diagrama de flujo del programa elaborado en esta tesis

2.3.1.3 Arquitecturas elaboradas

La selección de la adecuada arquitectura del modelo de RNA está basada en la modificación de la misma con base en los resultados obtenidos para los diferentes vectores de entrenamiento, en los modelos que aquí se presentan solo se utilizó una capa oculta, ya que se encontró que si se modificaba el número de capas ocultas, el error no disminuía significativamente. El número de neuronas que tiene la capa oculta fue modificado en varias ocasiones con base en el error obtenido de los vectores de entrenamiento. Los mejores resultados se presentan en las Tablas 2.5 y 2.6. Los valores incluidos en la columna denominada arreglo RNA tienen el formato ExOxS en el cual "E" indica el número de neuronas de la capa de entrada, "O" se refiere a la capa oculta (número de neuronas); y "S" indica el número de nodos de la capa de salida.

Tabla 2.5 Arquitecturas de las RNA's (Ductilidades globales)

Arreglo RNA ExOxS	Iteraciones	Error total
4X10x1	65	0.086
4X20x1	39	0.033
4X30x1	75	0.058
4X40x1	102	0.011
4X50x1	163	0.037
4X60x1	97	0.075
4X70x1	82	0.024

Tabla 2.6 Arquitecturas de las RNA's (Distorsiones máximas de entre piso)

Arreglo RNA ExOxS	Iteraciones	Error total
4X10x1	73	0.088
4X20x1	28	0.048
4X30x1	211	0.078
4X40x1	2181	0.022
4X50x1	142	0.038
4X60x1	56	0.077
4X70x1	135	0.028

En las Tablas 2.5 y 2.6 se puede ver que la arquitectura óptima en función del error total (error en el entrenamiento más el error en la prueba) es la que presenta el arreglo 4x40x1. Es decir para ambos casos esta arquitectura resultó ser la óptima: cuatro neuronas en la capa de entrada, 40 neuronas en la capa oculta, y una neurona en la capa de salida.

2.3.2 Resultados del entrenamiento y pruebas

En el entrenamiento de los modelos de RNA queda un error de estimación implícito el cual se puede observar en las Figuras 2.17 y 2.18. Los resultados de la etapa de entrenamiento muestran una excelente correlación con errores en estimación no mayores a 4% y 3% para los casos de distorsiones de entrepiso, y de ductilidades globales, respectivamente.

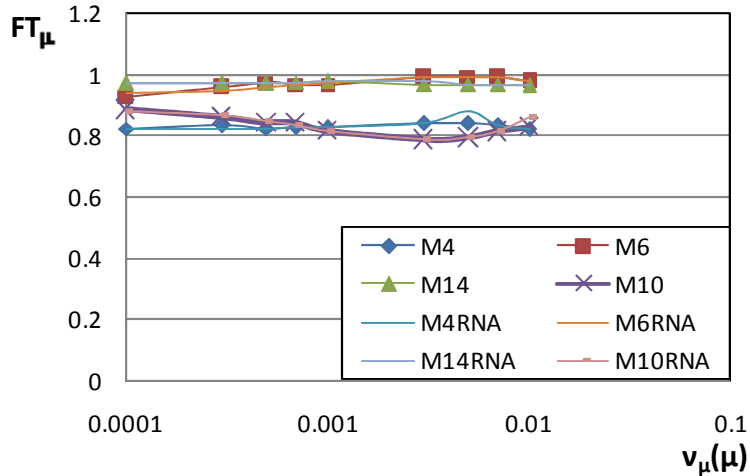


Figura 2.17 Error de estimación en entrenamiento (FT μ)

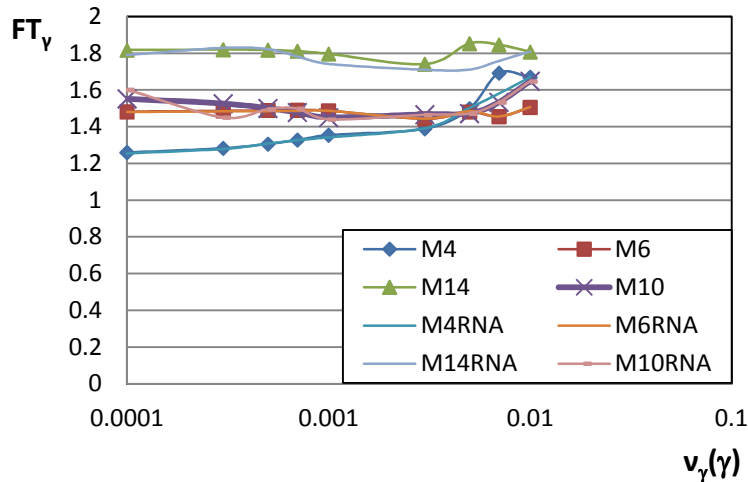


Figura 2.18 Error de estimación en entrenamiento (FT γ)

Los resultados del error en la fase de prueba se muestran en las Figuras 2.19 y 2.20. Como es de esperarse, los resultados obtenidos en la fase de prueba no alcanzan los niveles encontrados durante el entrenamiento, no obstante se aprecia un aceptable grado de similitud entre los FT obtenidos mediante RNA y los obtenidos del análisis con un error máximo del 10%. Se hace la aclaración que el modelo (M8) presentando en la fase de prueba nunca fue utilizado para entrenar a la red.

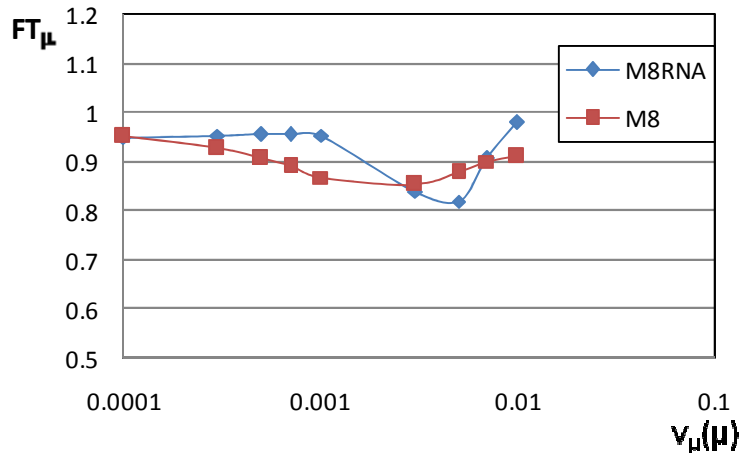


Figura 2.19 Error de estimación en prueba (FT_{μ})

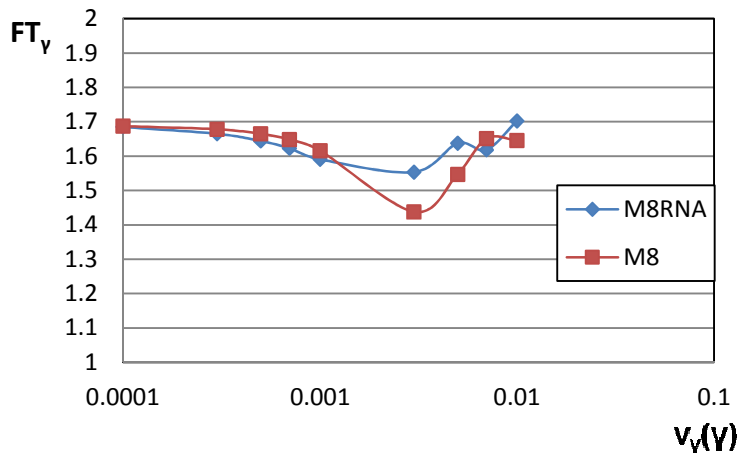


Figura 2.20 Error de estimación en prueba (FT_{γ})

2.4 Metodología propuesta para encontrar funciones de factores transformación

En la sección anterior se comprobó que el uso de la teoría de las RNA es aplicable para la estimación de FT, generalizando la respuesta a partir de algunos ejemplos. En esta sección se presenta la metodología para encontrar funciones de FT correspondientes a estructuras desplantadas en una determinada zona. Se presentan funciones para estimar los FT, específicamente para el caso de distorsiones máximas de entrepiso, bajo la condición que los sistemas estructurales modelados como SMGL y sus correspondientes S1GL tienen la misma tasa anual de excedencia ($v = 0.008$). Las funciones se considerarán dependientes únicamente de C_{γ} , δ_{γ} y T_0 . Estas funciones podrán aplicarse a cualquier sistema estructural que se encuentre dentro del intervalo considerado.

Para encontrar estas funciones primero se tienen que encontrar los FT_{γ} para estructuras reales (SMGL) diseñadas en una determinada zona (en este caso la zona IIIb del valle de México). Con los valores de FT_{γ} de estructuras reales, se entrena un modelo de RNA que pueda estimar la

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

respuesta de estructuras nunca antes presentadas a la red. Con esta ventaja, se puede hacer la simulación de muchas estructuras en un esfuerzo mucho menor al que se necesitaría si se analizaran de manera convencional. Una vez que se tiene un modelo de RNA funcionando de manera aceptable, se procede a analizar un conjunto de estructuras nunca antes presentadas a la red con el fin de establecer las funciones válidas para una amplia gama de estructuras.

Para encontrar las funciones de transformación en esta sub-sección se entrenó de nuevo un modelo de RNA incluyendo al marco M8 (ya que fue necesario incrementar el número de estructuras en los pares de entrenamiento a la red). Los valores de los intervalos utilizados para los parámetros se muestran en la Tabla 2.7.

Tabla 2.7 Nuevo intervalo de valores para la entrada a la red

Entrada	Intervalo
To	0.8 a 1.2
Cy	0.1 a 0.49
δy	0.1 a 0.5
v	0.008

El incremento de los intervalos de cada parámetro de entrada es el siguiente; para To se utilizó un incremento de 0.1, para Cy un incremento de 0.05, y por último el incremento de δy fue de 0.01. Este incremento tan pequeño es debido a que se pretende obtener una función de transformación para diferentes intervalos de δy , que esté en función de Cy, ya que este parámetro es uno de los más utilizados en los reglamentos del mundo.

Se presentó a la red un total de 1800 estructuras simuladas (estructuras de las cuales no se les conoce su geometría, sus propiedades de materiales, su masa ni su rigidez; pero sí se conocen sus propiedades dinámicas To, Cy y δy). Esto con el fin de obtener reglas generales de transformación (que pudieran aplicarse en los reglamentos de diseño). En la Figura 2.21 se muestra la variación del $FT\gamma$ en función de Cy, para $0.2 \leq \delta y \leq 0.24$

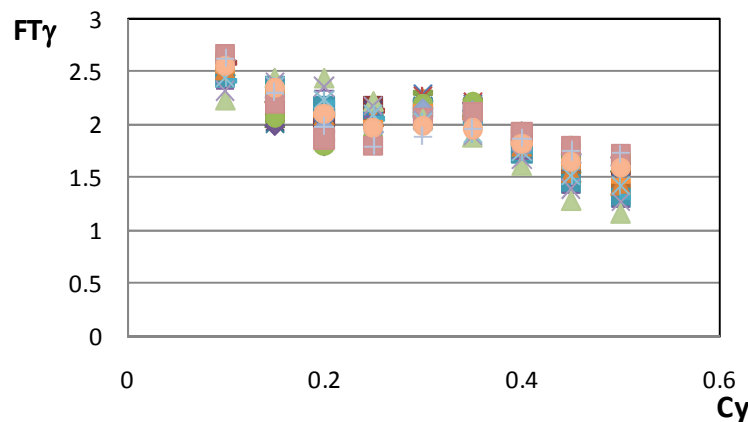


Figura 2.21 $FT\gamma$ en función de Cy ($0.2 \leq \delta y \leq 0.24$)

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

En la Figura 2.21 se observa que los valores FT_{γ} van disminuyendo de 2.5 a 1.5 aproximadamente, conforme se incrementa el valor de C_{γ} ; es decir, los valores de FT_{γ} son inversamente proporcionales a C_{γ} . Cada uno de los puntos de esta gráfica representa una estructura simulada dentro de un intervalo de δ_{γ} entre 0.2 a 0.24. Se estimó el valor para cada uno de los valores de C_{γ} . Los resultados de las medias para cada uno de los valores de C_{γ} , considerando diferentes intervalos de δ_{γ} , se muestran en la Figura 2.22.

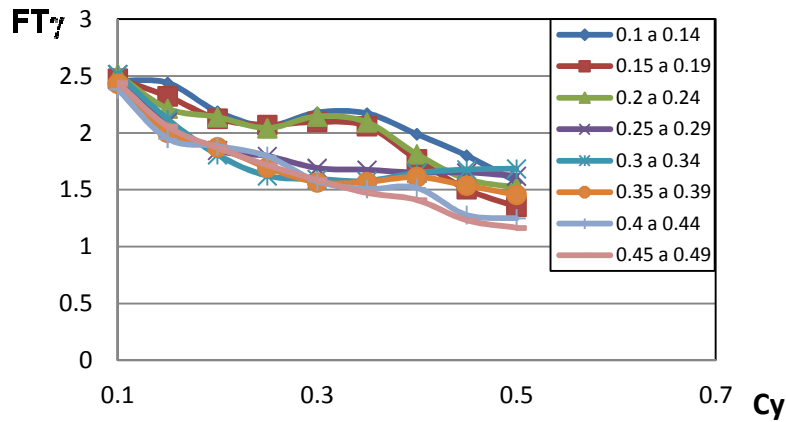


Figura 2.22 Media de los valores FT_{γ} en función de C_{γ} ($0.1 \leq \delta_{\gamma} \leq 0.49$)

En la Figura 2.22 se puede observar que los FT_{γ} presentan dos tendencias que corresponden a diferentes intervalos de δ_{γ} (entre 0.1 y 0.24, y entre 0.25 y 0.49). Se observa que se pueden formular de manera muy simplificada dos ecuaciones para cada uno de los intervalos mencionados anteriormente. Los resultados del ajuste de curvas para las dos ecuaciones se muestran en la Figura 2.23.

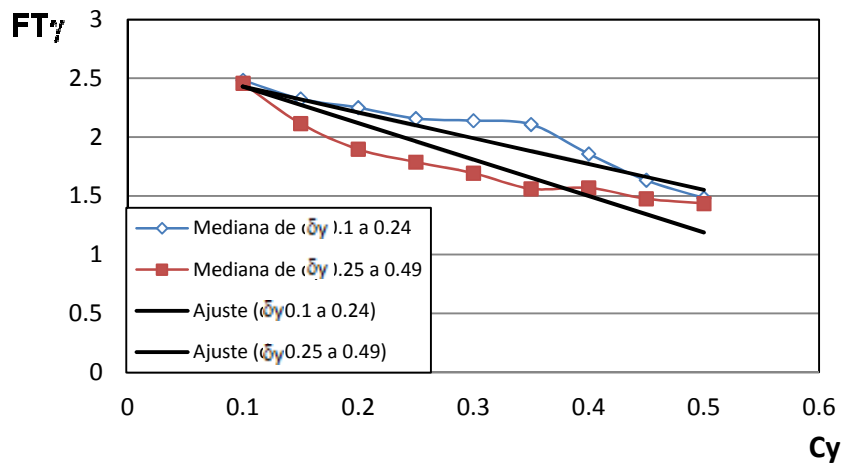


Figura 2.23 Media de los valores FT_{γ} en función de C_{γ} , para $0.2 \leq \delta_{\gamma} \leq 0.24$ y para $0.1 \leq \delta_{\gamma} \leq 0.49$

Capítulo 2. Primera aplicación de RNA: Factores de Transformación

En la Figura 2.23 se muestra el ajuste de curvas mediante mínimos cuadrados, utilizando una función lineal en la aproximación para ambos casos. El resultado se resume en las ecuaciones 2.5 y 2.6. Es importante mencionar que estas ecuaciones son válidas solo para un intervalo de periodo de vibración estructural de $0.8 \leq T_0 \leq 1.2s$.

$$FT\gamma = -2.2Cy + 2.65 \quad (0.1 \leq \delta y \leq 0.24) \quad (2.5)$$

$$FT\gamma = -3.1Cy + 2.65 \quad (0.25 \leq \delta y \leq 0.49) \quad (2.6)$$

La metodología aquí propuesta se podría seguir para distintas zonas e intervalos de parámetros de interés, para finalmente aplicarse en reglas de diseño sísmico de estructuras basadas en confiabilidad.

Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

3.1 Introducción

En este capítulo se presenta un modelo de RNA para estimar espectros de respuesta inelásticos en función de magnitud del evento M , mecanismo de falla MF, distancia de Joyner y Boore (JyB), velocidad de ondas de cortante V_s , periodo fundamental de la estructura T_0 , y ductilidad del sistema. El modelo se aplica a varios registros de aceleraciones tomados de diferentes zonas sísmicas del mundo obtenidos de la base de datos "Next Generation Attenuation NGA". De los resultados obtenidos se aprecia una concordancia aceptable entre los valores reales y los obtenidos mediante las RNA. El estudio muestra que las RNA son una valiosa herramienta para la estimación de espectros de respuesta inelásticos.

El espectro de respuesta es una de las herramientas más útiles en la Ingeniería Sísmica, el concepto de espectro de respuesta se usó por primera vez por Benioff en 1934, posteriormente por Housner 1941 y Biot (1941). Numerosos estudios han intentado establecer metodologías para la estimación de espectros de respuesta; sin embargo, la estimación de espectros de respuesta inelásticos requiere de análisis complejos, por lo que los métodos tradicionales difícilmente pueden obtener un error de aproximación aceptable si se utilizan pocos registros.

Existen diferentes modelos de RNA. En este capítulo se seleccionó un modelo de RNA "feedforward" entrenada mediante el algoritmo "Backpropagation". Esto es debido a que con este modelo se han obtenido resultado aceptables para problemas de naturaleza similar. Por ejemplo Alves (2006) utilizó las redes neuronales en la predicción de eventos sísmicos. Lee y Han (2002) utilizaron cinco modelos de RNA para generar sismos artificiales y espectros de respuesta. Kerth y Ting (2005) entrenaron un modelo "Backpropagation" para estimar la aceleración máxima del suelo, en función de la distancia al epicentro, profundidad del hipocentro y de la magnitud del sismo, para el diseño de las líneas de alta velocidad de ferrocarriles en Taiwan. Utilizando un modelo de red neuronal se pudieron predecir las réplicas sísmicas después de un gran terremoto (Barrile et al. 2006), por otro lado se han desarrollado trabajos para estimar la aceleración máxima del suelo (Arjun y Kumar, 2009; Günaydın y Günaydın 2008). Es importante mencionar que muchas de las investigaciones han utilizado una función de transferencia sigmoidea en las capas ocultas y una función lineal en la capa de salida obteniendo resultados satisfactorios. Por este motivo aquí se seleccionan estas mismas funciones de transferencia.

3.2 Registros utilizados

Se utilizaron un total de cincuenta registros de aceleraciones obtenidos de la base de datos NGA, correspondientes a sismos que han ocurrido en todo el mundo. Los registros utilizados para este estudio corresponden a sismos con distancias de Joyner y Boore (1993) (JyB) comprendidas entre 18 y 194Km, y magnitudes (M) comprendidas entre 5.9 y 7.7. La variación de la magnitud con la distancia de Joyner y Boore se muestra en la Figura 3.1.

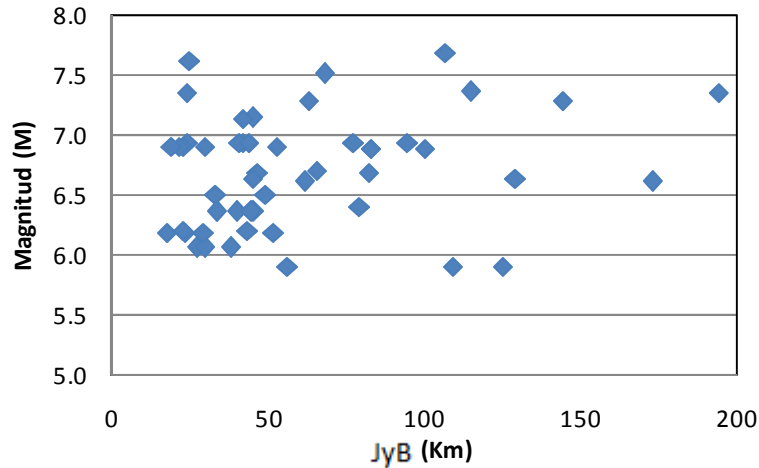


Figura 3.1 Magnitud contra distancia de Joyner y Boore

En las Tablas 3.1 a 3.5 se listan los registros sísmicos utilizados para este estudio. En dichas tablas la primer columna se refiere a el número de registro, la segunda columna corresponde a la magnitud del evento, la tercer columna se refiere a la distancia de (JyB), y la última columna corresponde a la velocidad de ondas de cortante del sitio. El tipo de falla y de suelo (columnas 3 y 5) se definen más adelante.

Tabla 3.1 Registros de aceleraciones de la Zona A

Registro	Magnitud	Falla	Distancia(km)	Suelo	Vs(m/s)
1	6.61	2	61.79	A	1735.0
2	6.90	1	52.93	A	1562.2
3	6.36	2	40.01	A	1676.1
4	6.88	1	100.22	A	1559.6
5	6.19	0	23.23	A	1862.2
6	6.06	3	38.22	A	1684.9
7	6.93	3	41.68	A	1815.3
8	7.28	0	144.13	A	1946.0
9	6.69	2	46.65	A	1821.7
10	7.68	0	106.71	A	1659.6

Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

Tabla 3.2 Registros de aceleraciones de la Zona B

Registro	Magnitud	Falla	Distancia(km)	Suelo	Vs(m/s)
1	6.63	0	129.11	B	1328.6
2	6.50	2	49.13	B	1274.4
3	6.88	1	82.60	B	1274.4
4	6.06	3	27.21	B	1112.4
5	7.14	0	45.16	B	1413.0
6	7.13	0	41.82	B	814.3
7	6.36	2	45.49	B	1315.0
8	6.36	2	33.42	B	1015.6
9	6.88	1	83.00	B	1274.4
10	6.06	3	29.56	B	1015.6

Tabla 3.3 Registros de aceleraciones de la Zona C

Registro	Magnitud	Falla	Distancia(km)	Suelo	Vs(m/s)
1	6.61	2	173.16	C	388.5
2	7.35	2	24.07	C	368.2
3	6.90	1	29.79	C	532.7
4	6.20	1	22.68	C	530.0
5	6.36	2	44.82	C	438.5
6	6.70	0	65.67	C	548.6
7	6.19	0	29.35	C	738.5
8	6.93	3	40.85	C	449.6
9	7.28	0	62.98	C	645.4
10	6.40	1	79.33	C	598.6

Tabla 3.4 Registros de aceleraciones de la Zona D

Registro	Magnitud	Falla	Distancia(km)	Suelo	Vs(m/s)
1	7.36	2	114.62	D	253.2
2	6.19	0	17.64	D	327.1
3	6.63	0	45.12	D	180.8
4	6.50	2	33.32	D	319.6
5	7.35	2	193.91	D	219.6
6	6.20	1	43.50	D	359.6
7	6.93	3	24.27	D	191.8
8	6.69	2	82.03	D	296.6
9	7.62	3	24.98	D	188.1
10	6.90	1	22.54	D	320.0

Tabla 3.5 Registros de aceleraciones de la Zona E

Registro	Magnitud	Falla	Distancia(km)	Suelo	Vs(m/s)
1	6.90	0	21.35	E	151.2
2	7.51	0	68.09	E	175.0
3	6.90	0	19.14	E	164.0
4	6.93	3	77.32	E	155.1
5	6.93	3	94.56	E	169.7
6	6.93	3	43.77	E	116.4
7	6.19	0	51.68	E	146.0
8	5.90	2	56.17	E	176.0
9	5.90	2	109.02	E	173.0
10	5.90	2	124.87	E	156.0

3.2.1 Mecanismo de falla

Los registros de aceleraciones utilizados fueron generados por cuatro tipos de falla diferentes. Estos se muestran en la Figura 3.2. Se utilizaros cuatros tipos de falla en el estudio: falla longitudinal (0), Normal (1), Inversa (2) y Oblicua (3). Es importante mencionar que en la selección de los registros de aceleraciones se consideraron un promedio de 12 registros para cada tipo de falla, con el fin de que la RNA se pudiera generalizar a partir de pocos registros y reducir el error en la estimación.

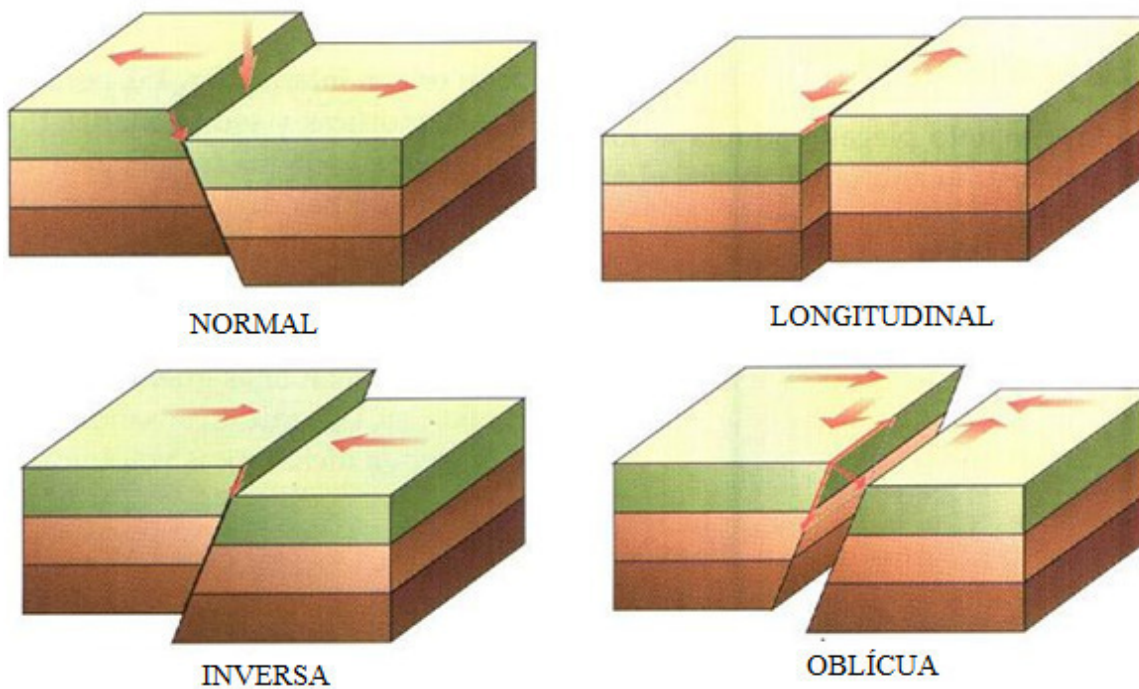


Figura 3.2 Tipos de fallas

Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

3.2.1 Tipo de suelo

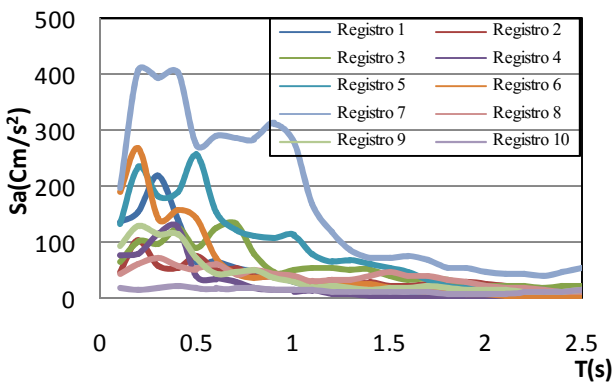
Para este estudio se consideraron cinco tipos de suelo en función de la velocidad de ondas de cortante. En la Tabla 3.6 se presenta el tipo de suelo en función de la velocidad de ondas de cortante (V_s) según (NGA).

Tabla 3.6 Tipos de suelo utilizados

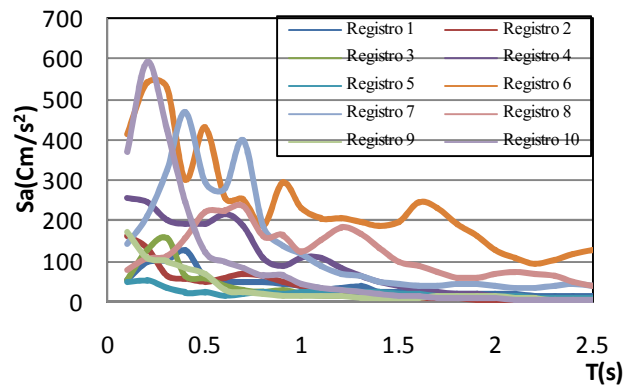
Zona	Condiciones del sitio	Velocidad (V_s en m/s)
A	Roca Dura	> 1500
B	Roca	760 - 1500
C	Roca Blanda	360 - 760
D	Roca profunda	180 - 360
E	Arcilla	< 180

3.3 Espectros de respuesta en función de su ductilidad (D_u)

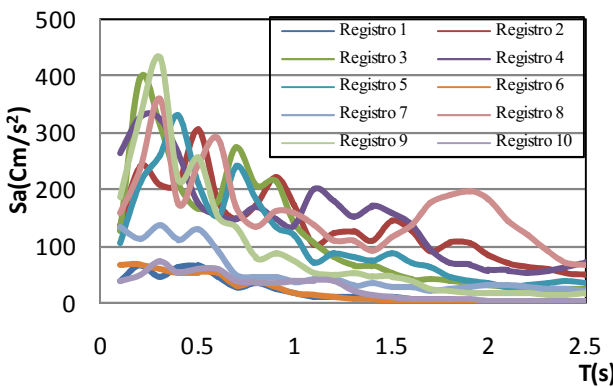
Los espectros de respuesta asociados a cada una de las zonas se muestran en las Figuras 3.3 a 3.6. Estos corresponden a ductilidades $D_u = 1, 2, 4$ y 6 , respectivamente.



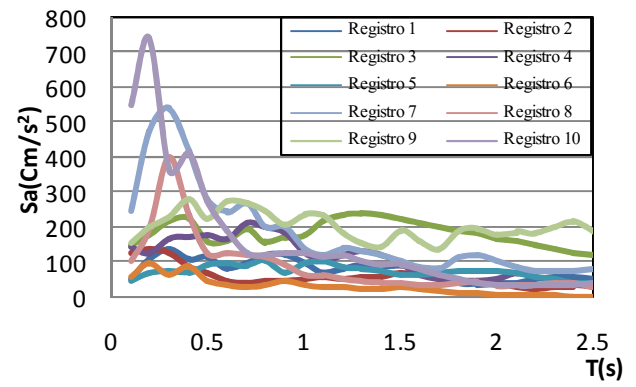
Zona A



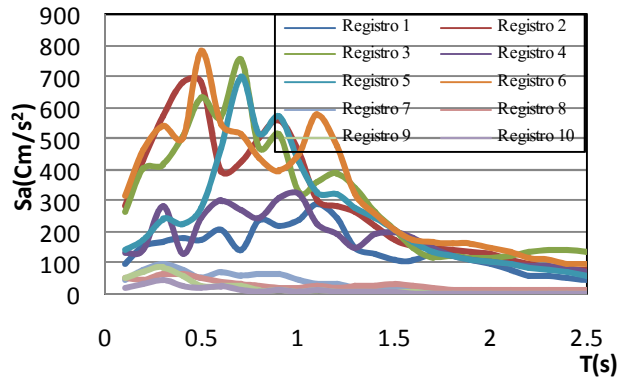
Zona B



Zona C

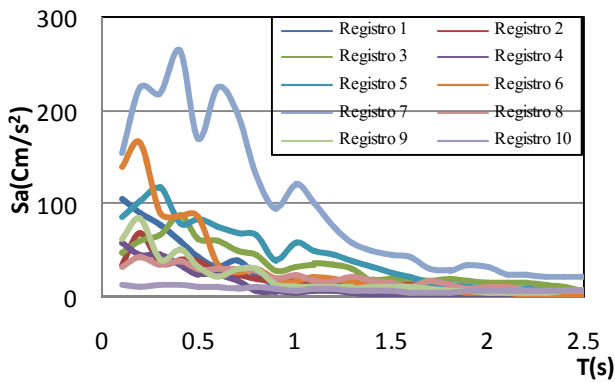


Zona D

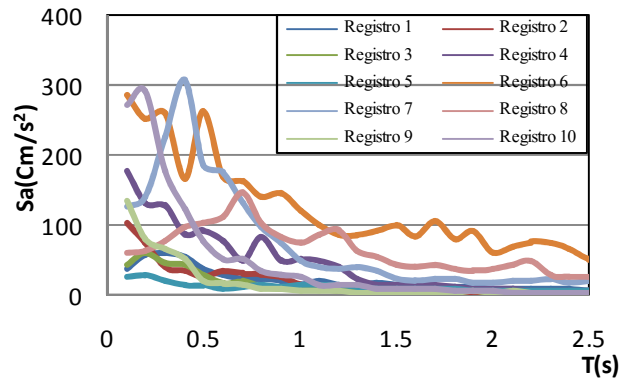


Zona E

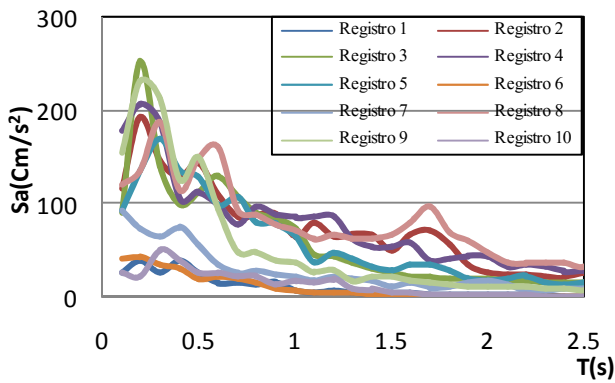
Figura 3.3 Espectros de respuesta para las diferentes zonas. $Du = 1$



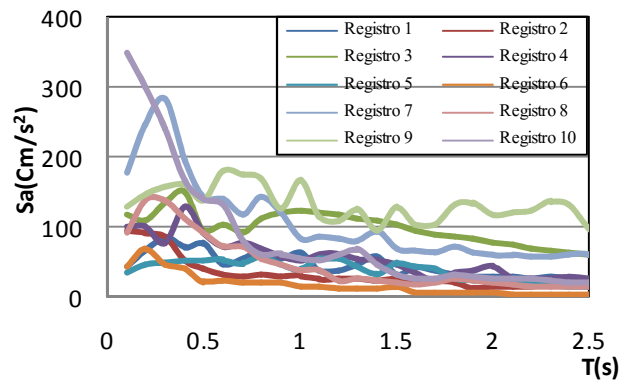
Zona A



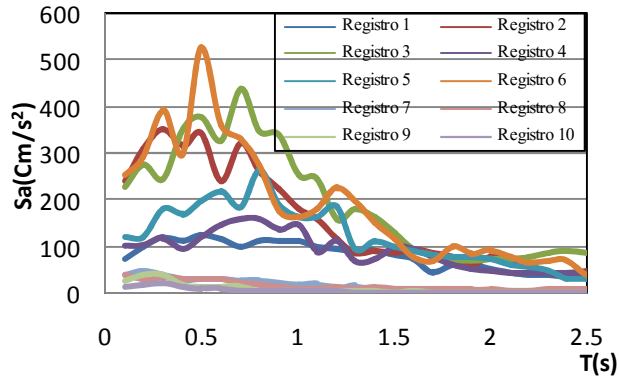
Zona B



Zona C

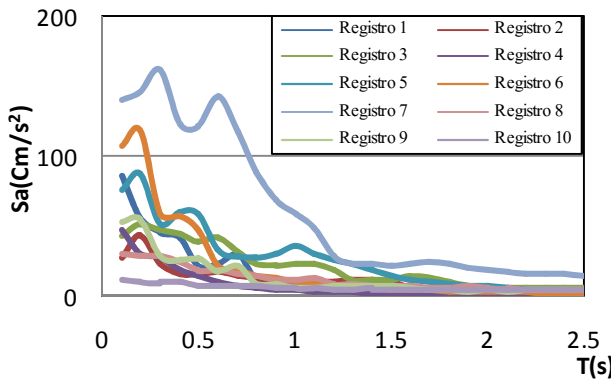


Zona D

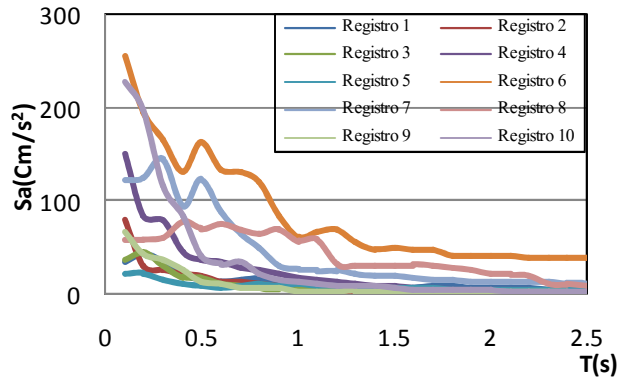


Zona E

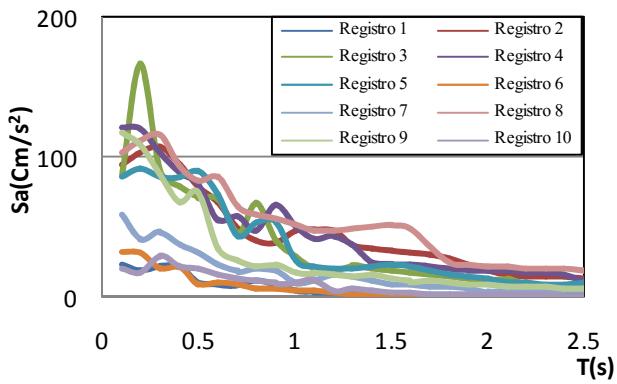
Figura 3.4 Espectros de respuesta para las diferentes zonas. $Du = 2$



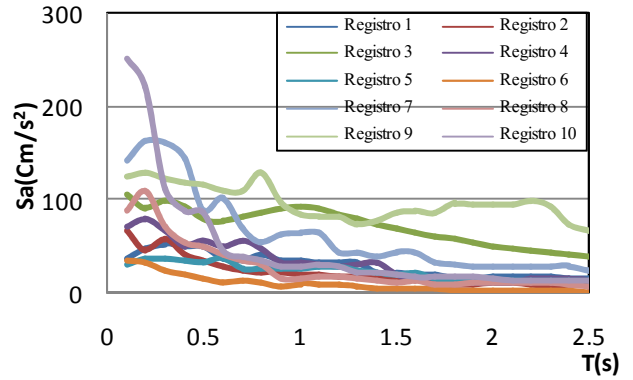
Zona A



Zona B



Zona C



Zona D

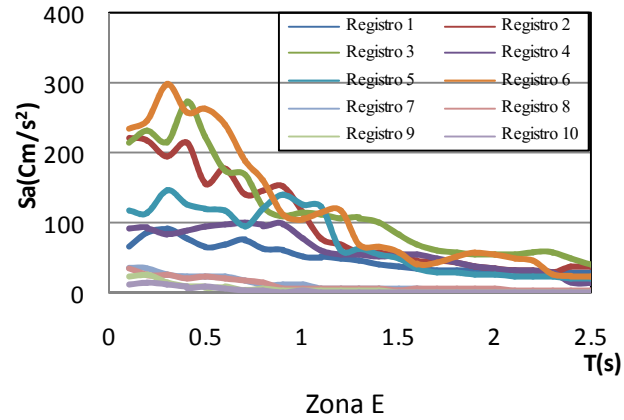
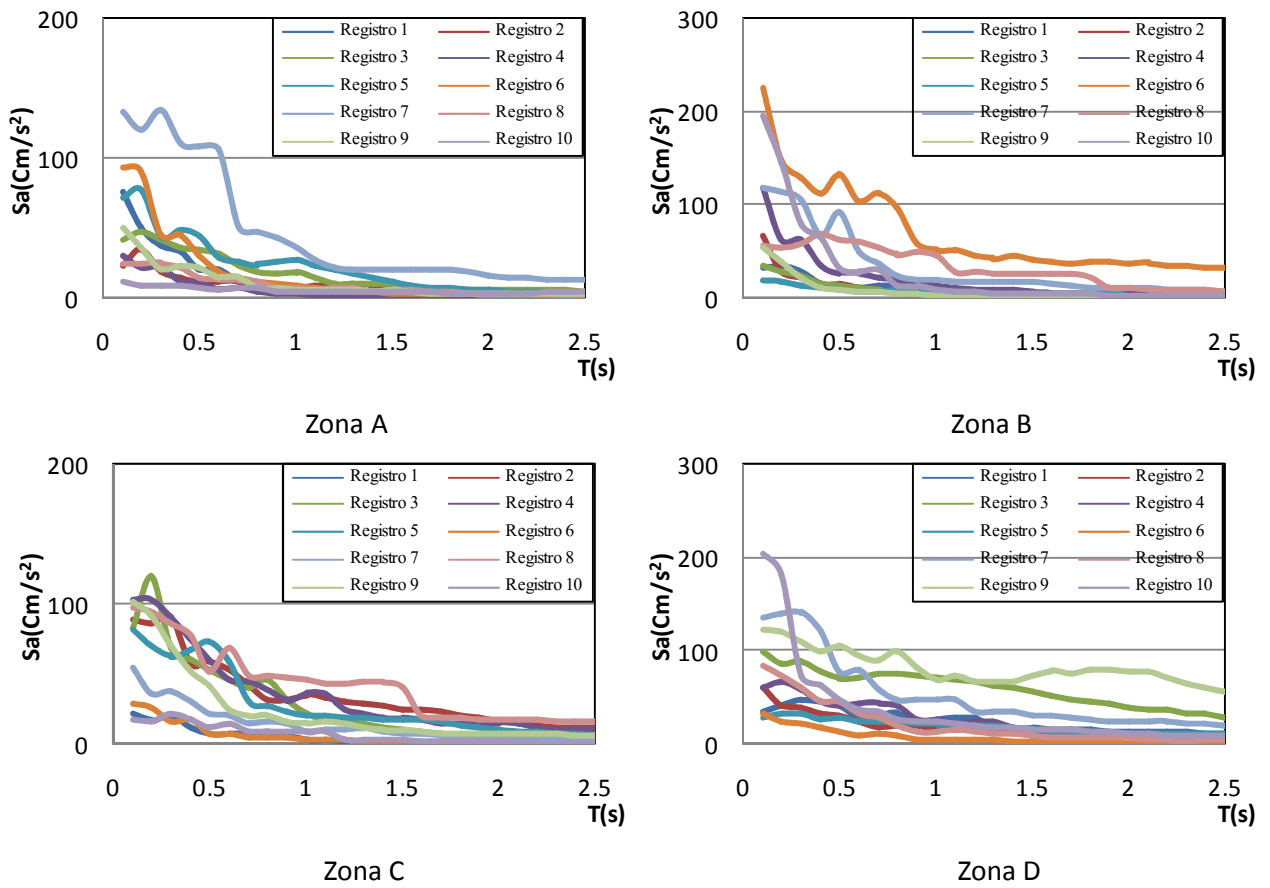


Figura 3.5 Espectros de respuesta para las diferentes zonas. $D_u = 4$



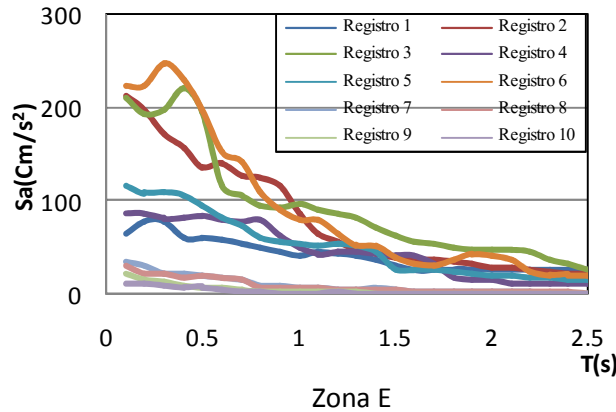


Figura 3.6 Espectros de respuesta para las diferentes zonas. $D_u = 6$

3.4 Modelo Neuronal Artificial

El objetivo de esta parte del estudio es implementar redes neuronales artificiales para la estimación de espectros de respuesta inelásticos utilizando el Perceptron Multicapa (“Backpropagation”). Para lograr este objetivo primero se deben seleccionar los parámetros adecuados para el entrenamiento, y en segundo lugar se debe encontrar la arquitectura óptima de la red. La limitación más importante de las RNA es que su eficiencia depende del algoritmo de entrenamiento y de la arquitectura de la red. Desafortunadamente no existe una guía para determinar estas características de la red. Solo utilizando el procedimiento de prueba y error se puede tratar de buscar la red óptima; sin embargo, este método no garantiza que la red sea óptima. Para que el algoritmo “Backpropagation” proporcione buenos resultados es necesario que el conjunto de valores que representan el problema incluya información de todo el intervalo de salida. La apropiada selección de los parámetros de entrenamiento es un factor importante en el entrenamiento de las redes neuronales artificiales.

El espectro de respuesta puede ser visualizado gráficamente como un elemento que integra la respuesta máxima de un sistema de un grado de libertad con cierto nivel de amortiguamiento y ductilidad cuando está expuesto a una excitación en la base. Esta respuesta puede ser definida mediante diferentes parámetros (desplazamientos, velocidades etc.). En este estudio se utilizará la pseudoaceleración como parámetro para el espectro de respuesta, por lo que el parámetro de la salida de la red será un valor de la pseudoaceleración.

3.5 Determinación de las entradas y salidas necesarias de la RNA

La adecuada selección de los parámetros que representen a una RNA es fundamental para lograr el correcto funcionamiento de la red. En esta parte del estudio se establece que dichos parámetros para representar la respuesta en términos de pseudoaceleraciones son: magnitud (M), mecanismo de falla (MF), distancia (JyB), velocidad de ondas de cortante (Vs), periodo de vibración del sistema (T_o) y su ductilidad (D_u). De acuerdo a lo anterior las entradas a la red quedan

Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

definidas por los valores de M, MF, JyB, Vs y To y Du. El nodo de salida queda representado por la pseudoaceleración (Sa). Los valores para los parámetros de entrada se mostraron en las Tablas 3.1 a 3.5, los valores de la pseudoaceleración se pueden observar en las Figuras 3.3 a 3.6. Un resumen de los intervalos de de los valores de entrada se muestra en la Tabla 3.7.

Tabla 3.7 Rango de intervalos para las entradas a la red

Entrada	Intervalo
M	5.9 a 7.68
MF	4 tipos
JyB	18 a 194 Km
Vs	116 a 1946 m/s
To	0 a 3 seg
Du	1,2 ,4 y 6

3.6 Arquitecturas elaboradas

Se probaron alrededor de 300 arquitecturas con el fin de obtener el modelo de RNA óptimo. En la Tabla 3.8 se muestran algunos de los modelos que se utilizaron. En dicha tabla se puede observar que los modelos con más de una capa oculta presentan un error total menor en el entrenamiento que los modelos con una capa oculta; sin embargo, estos modelos en la fase de pruebas generaban un error total mucho mayor a los modelos con una capa oculta, esto es debido a que el modelo se sobreentrenaba y la red podía aproximarse con mucha eficiencia a los espectros en el entrenamiento, pero no era así para la fase prueba; es decir, la red no se podía generalizar. Resultó que la arquitectura óptima para este problema es una red neuronal multicapa con alimentación hacia adelante. La selección de la arquitectura óptima de una RNA no es una tarea fácil ya que es necesario probar un gran número de arquitecturas para lograr establecer una como óptima. Actualmente existen métodos de entrenamiento de RNA basados en algoritmos evolutivos que son capaces de encontrar la arquitectura óptima de manera más eficiente que el juicio de un ser humano. En esta tesis, bajo el juicio del autor en función del error y las pruebas, resultó que la arquitectura del modelo óptimo está constituida por una capa de entrada, una capa oculta y una capa de salida; con la topología siguiente: $6 \times 100 \times 1$, es decir 6 neuronas en la capa de entrada 100 neuronas en la capa oculta y una neurona en la capa de salida. El formato para los arreglos de las arquitecturas mostradas en la Tabla 3.8 es el siguiente: $E \times O_1 \times O_2 \times \dots \times O_i \times S$. Donde O_i representa las i capas ocultas de la red.

Tabla 3.8 Arquitecturas elaboradas de RNA's

Arreglo	Iteraciones	Error Total
6X50X1	73	0.13
6X75X1	56	0.12
6X100X1	165	0.082
6X125X1	196	0.093
6X30X60X1	338	0.095
6X50X60X1	498	0.087
6X90X60X1	657	0.084
6X100X50X1	865	0.077
6X30X60X30X1	1823	0.069
6X50X50X50X1	1435	0.073
6X60X90X60X1	1748	0.064
6X100X100X100X1	2181	0.057

3.7 Entrenamiento del modelo de RNA

Para el entrenamiento se utilizó el algoritmo "Backpropagation". Las funciones de transferencia en las capas de la red fueron: sigmoidea para la capa oculta, y función lineal para la capa de salida. El entrenamiento se llevó a cabo de manera similar al entrenamiento que se presentó en el capítulo anterior. Durante el entrenamiento se observó que las RNA tenían problemas para estimar los espectros de respuesta inelásticos por lo que fue necesario probar alrededor de 300 arquitecturas. Se programó un código que incluye rutinas de Matlab Neural Network Tool Box para entrenar y probar los modelos de RNA.

La selección de los parámetros de entrenamiento fue tal que las RNA tuvieran una capacidad predictiva ante eventos no utilizados durante el entrenamiento. En las Figura 3.7 se muestra la regresión en el entrenamiento de las redes, donde se compara el valor estimado por la red y el valor real. En las Figuras 3.7 a y b se presenta el modelo 6x100x1 y el 6x100x100x100x1, respectivamente, se observa que para el caso b el error en la estimación en el entrenamiento es ligeramente menor que para el caso a; sin embargo, durante la fase de entrenamientos el error producido por la red del caso b era mucho mayor que la red del caso a, esto es debido a un sobreaprendizaje (ver sección 1.15). Se observó el problema de sobreaprendizaje en todas las redes con dos o más capas ocultas.

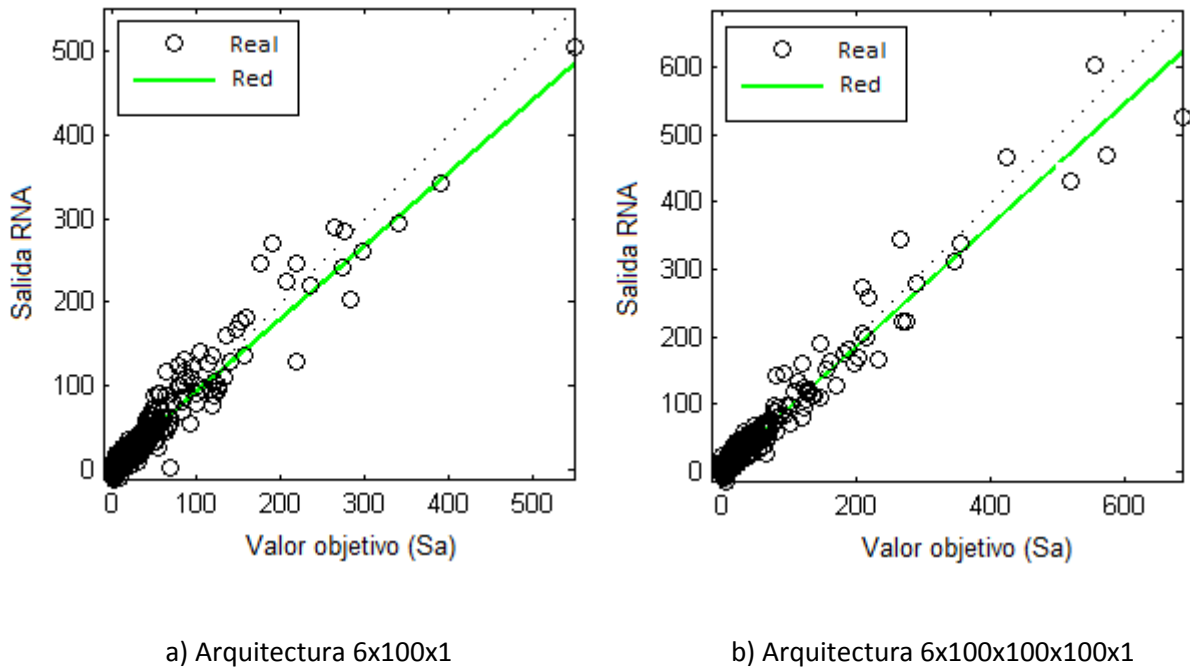
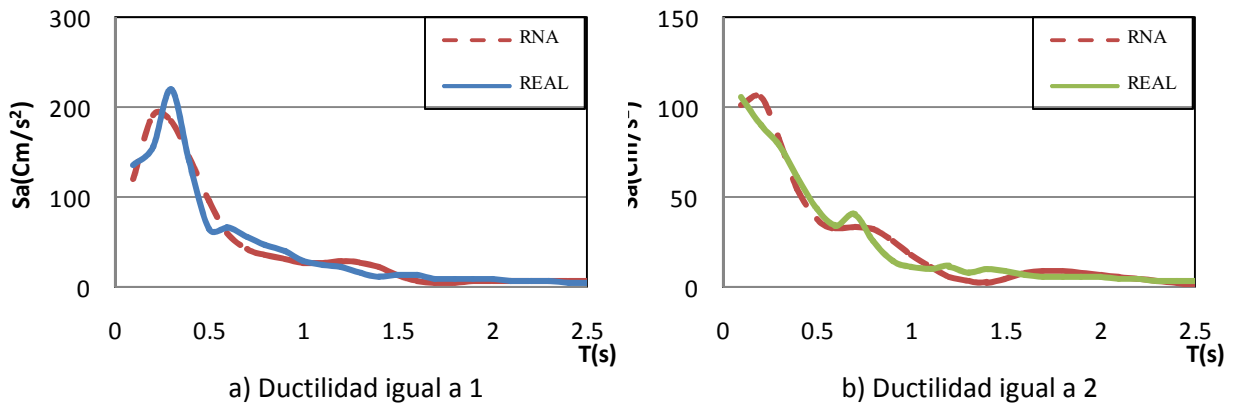


Figura 3.7 Regresión en el entrenamiento de las redes

3.8 Resultados del entrenamiento

En las Figuras 3.8 a 3.12 se comparan los espectros de respuestas obtenidos durante el entrenamiento (línea discontinua) y los espectros reales (línea continua). Los resultados de la etapa de entrenamiento muestran una correlación aceptable, debido a que el error máximo entre los espectros de respuesta reales y utilizando RNA fue de 15%. Se observa que la forma espectral presenta una clara definición en todo el intervalo de periodos. Solo se presentan los espectros correspondientes a un registro sísmicos correspondiente a cada una de las zonas en estudio.



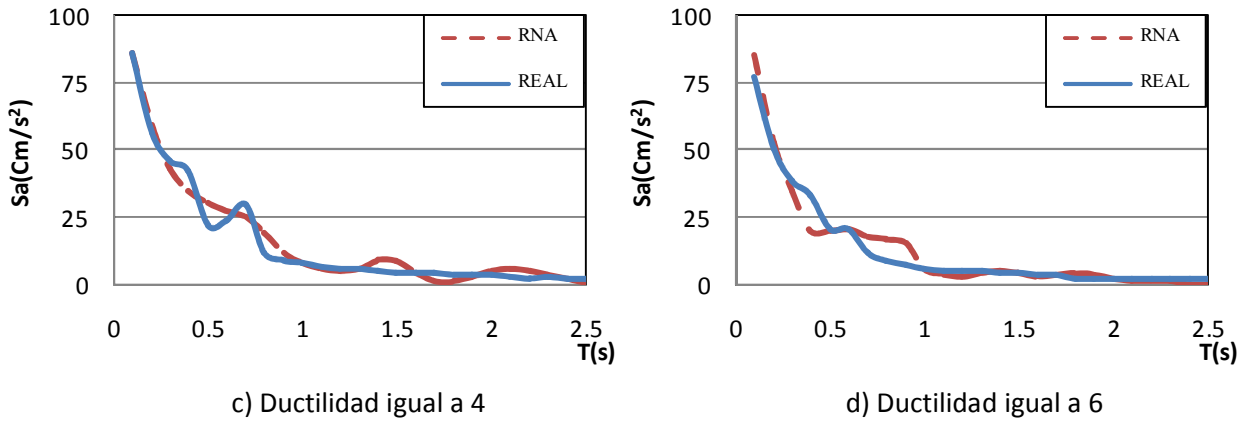


Figura 3.8 Espectros de respuesta obtenidos en la etapa de entrenamiento (Zona A, registro 1)

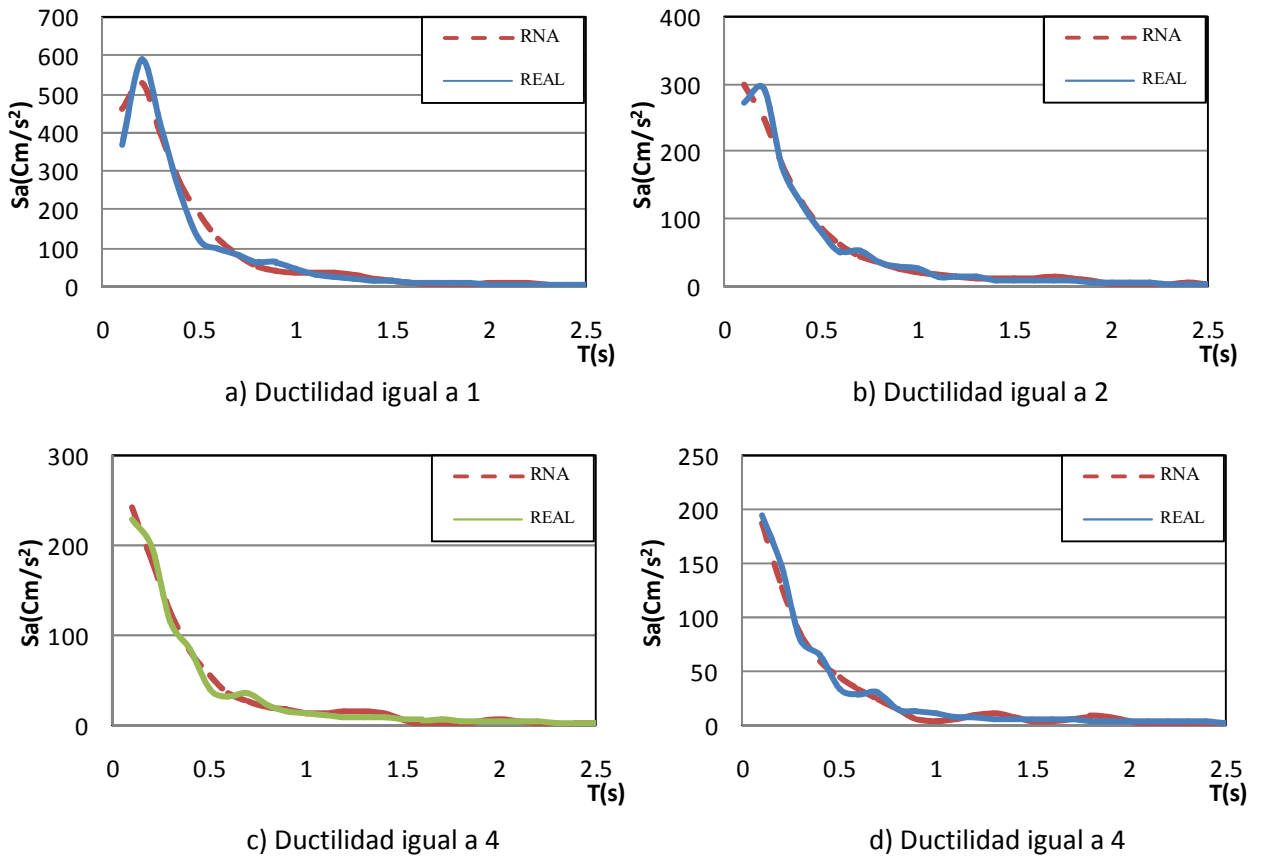


Figura 3.9 Espectros de respuesta obtenidos en la etapa de entrenamiento (Zona B, registro 10)

Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

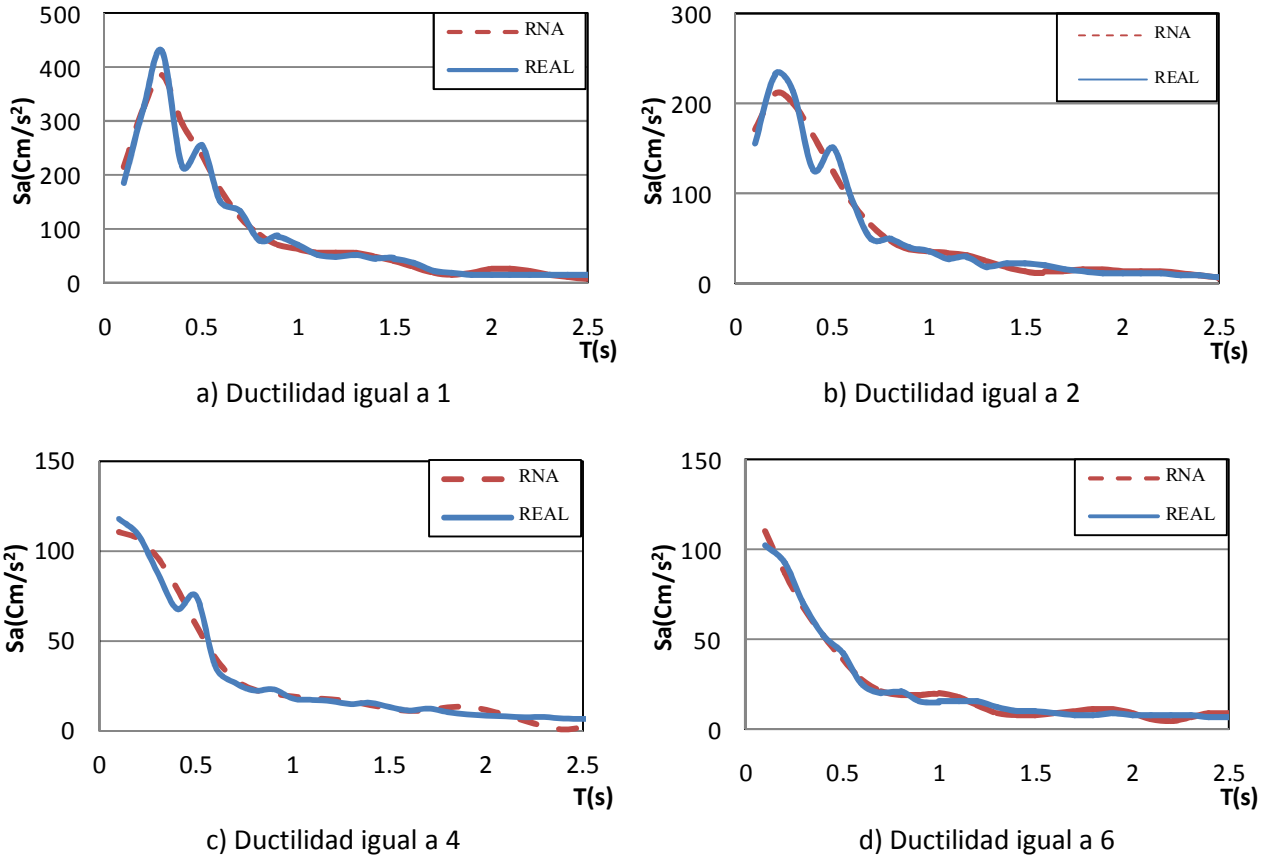
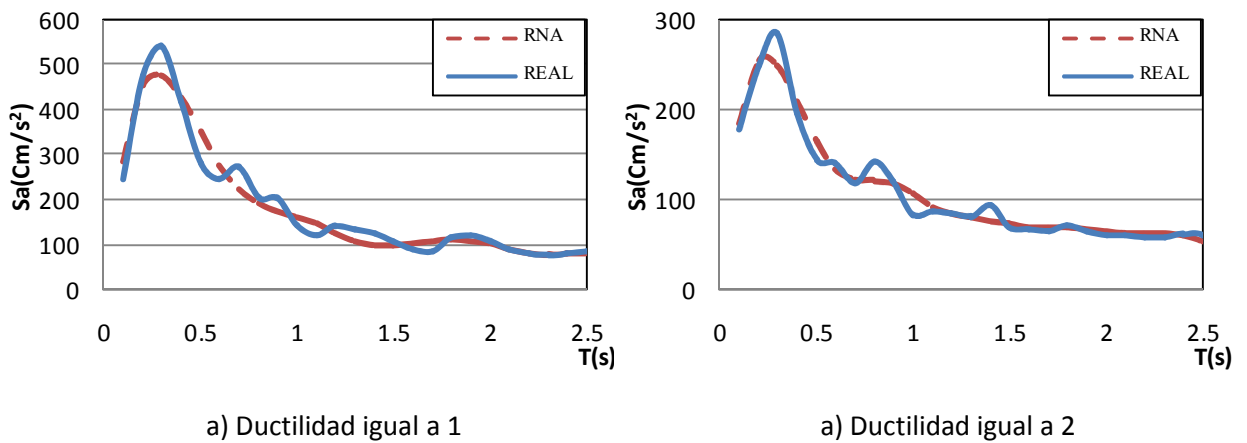


Figura 3.10 Espectros de respuesta obtenidos en la etapa de entrenamiento (Zona C, registro 9)



Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

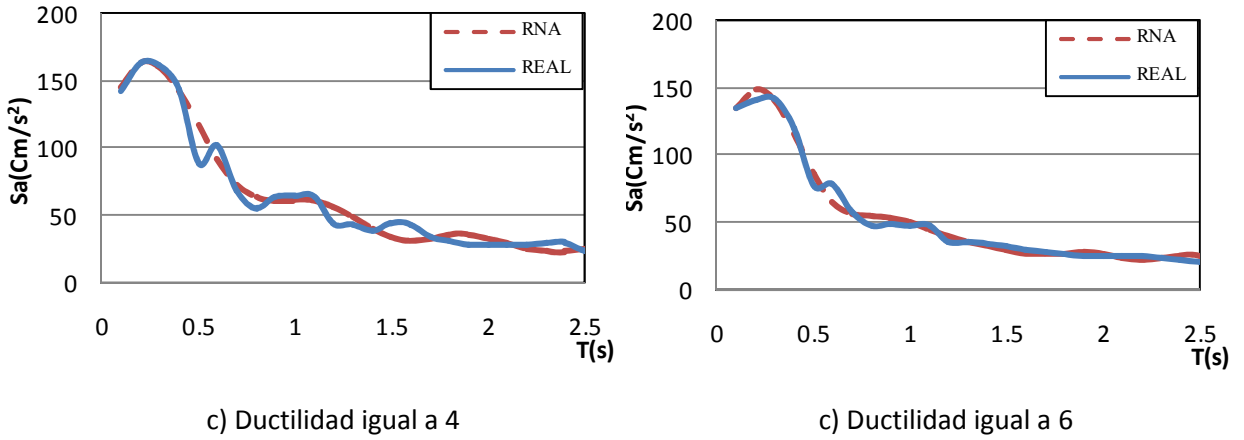


Figura 3.11 Espectros de respuesta obtenidos en la etapa de entrenamiento (Zona D, registro 7)

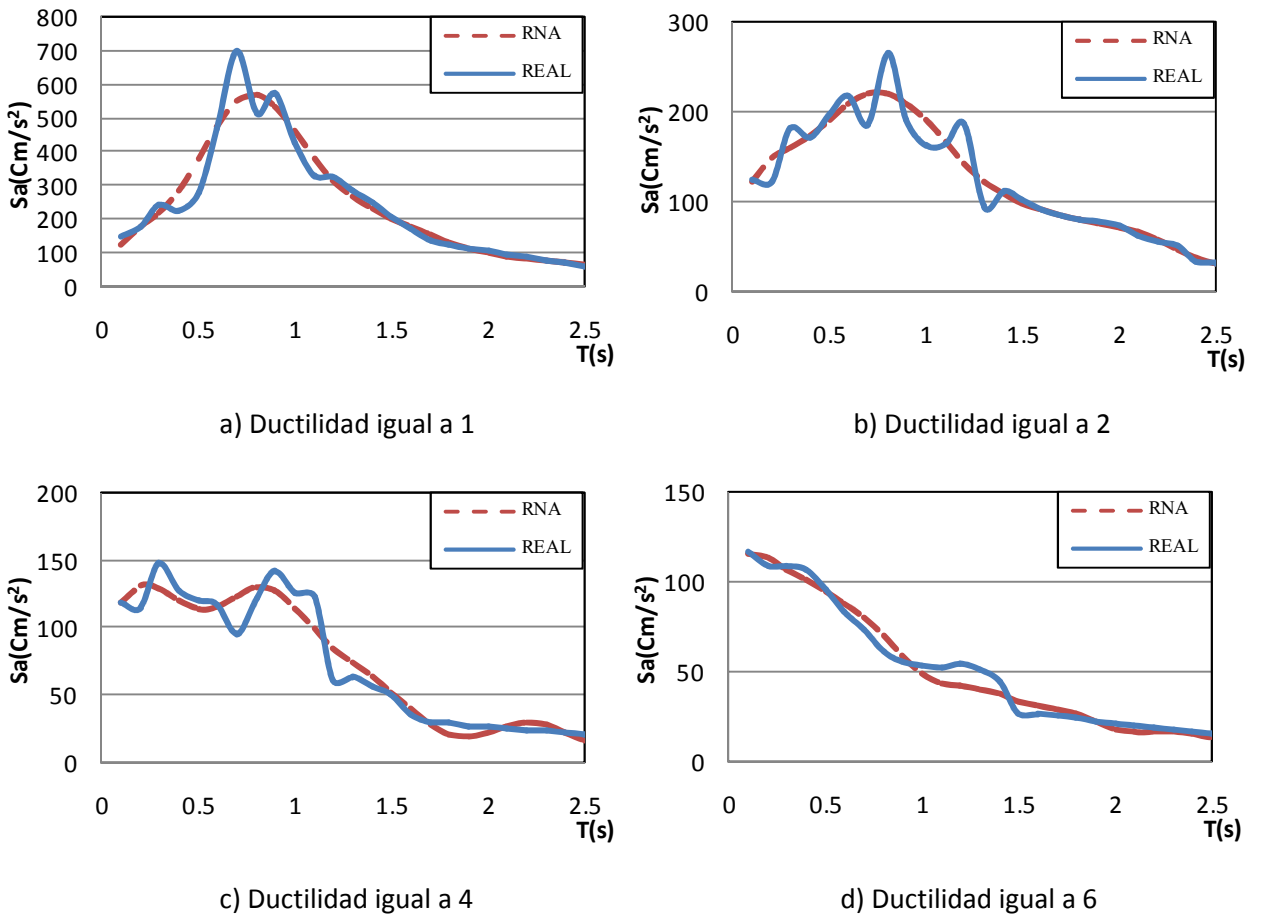


Figura 3.12 Espectros de respuesta obtenidos en la etapa de entrenamiento (Zona E, registro 5)

3.9 Resultados de las pruebas

Los resultados de las pruebas se muestran en las Figuras 3.13 a 3.17. Se observa que en la etapa de prueba los resultados obtenidos con RNA no alcanzan los niveles obtenidos durante la etapa de entrenamiento; no obstante, los resultados presentan una buena correlación con los espectros reales. Se observa que el modelo de RNA conserva la forma espectral en todos los casos. El error medio cuadrático entre el modelo de RNA y los espectros reales es de 12%. Se presentan resultados para cada una de las zonas utilizando únicamente un registro para cada una de ellas. Se encontraron resultados similares para todos los registros analizados.

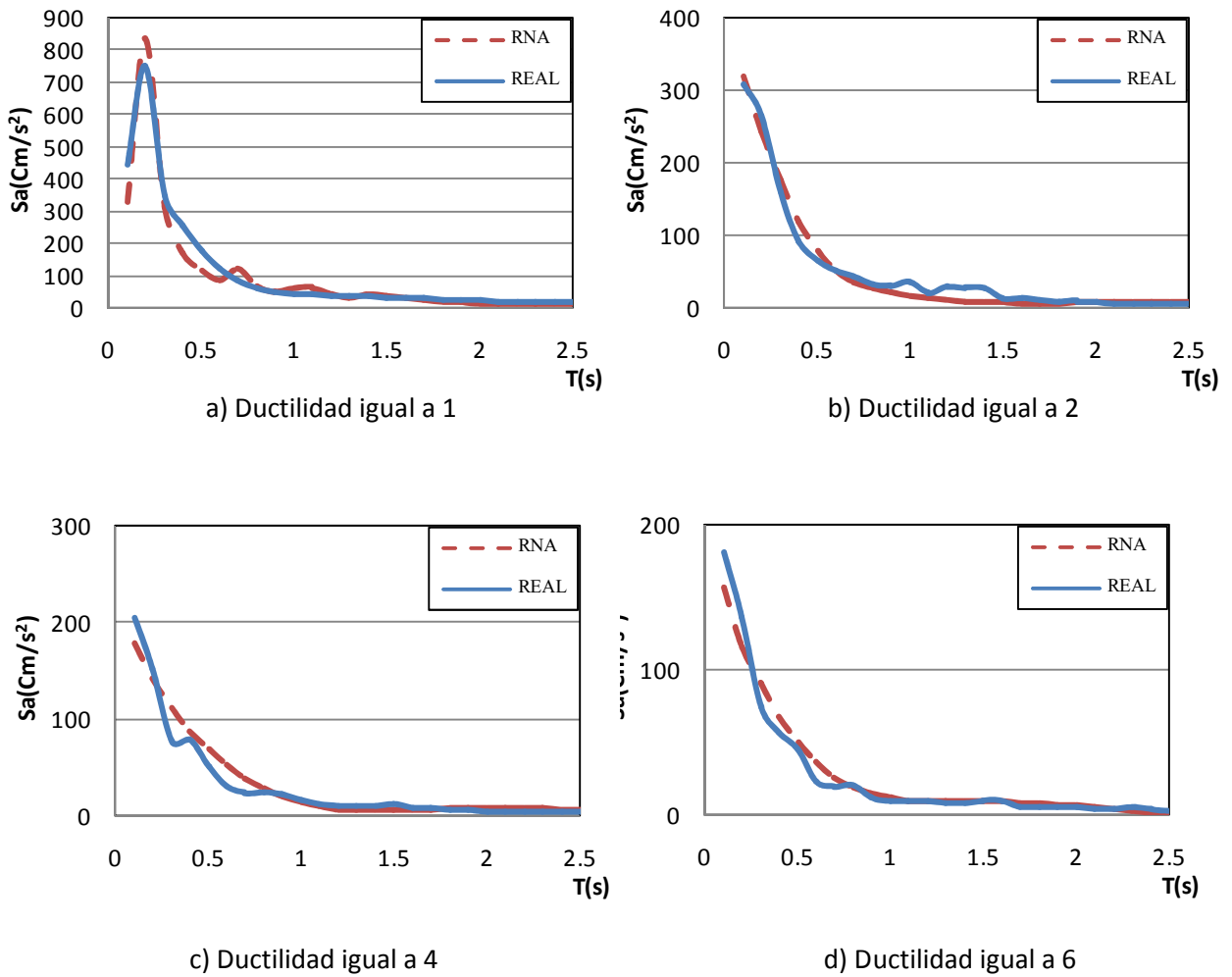


Figura 3.13 Espectros de respuesta obtenidos en la etapa de pruebas para la Zona A

Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

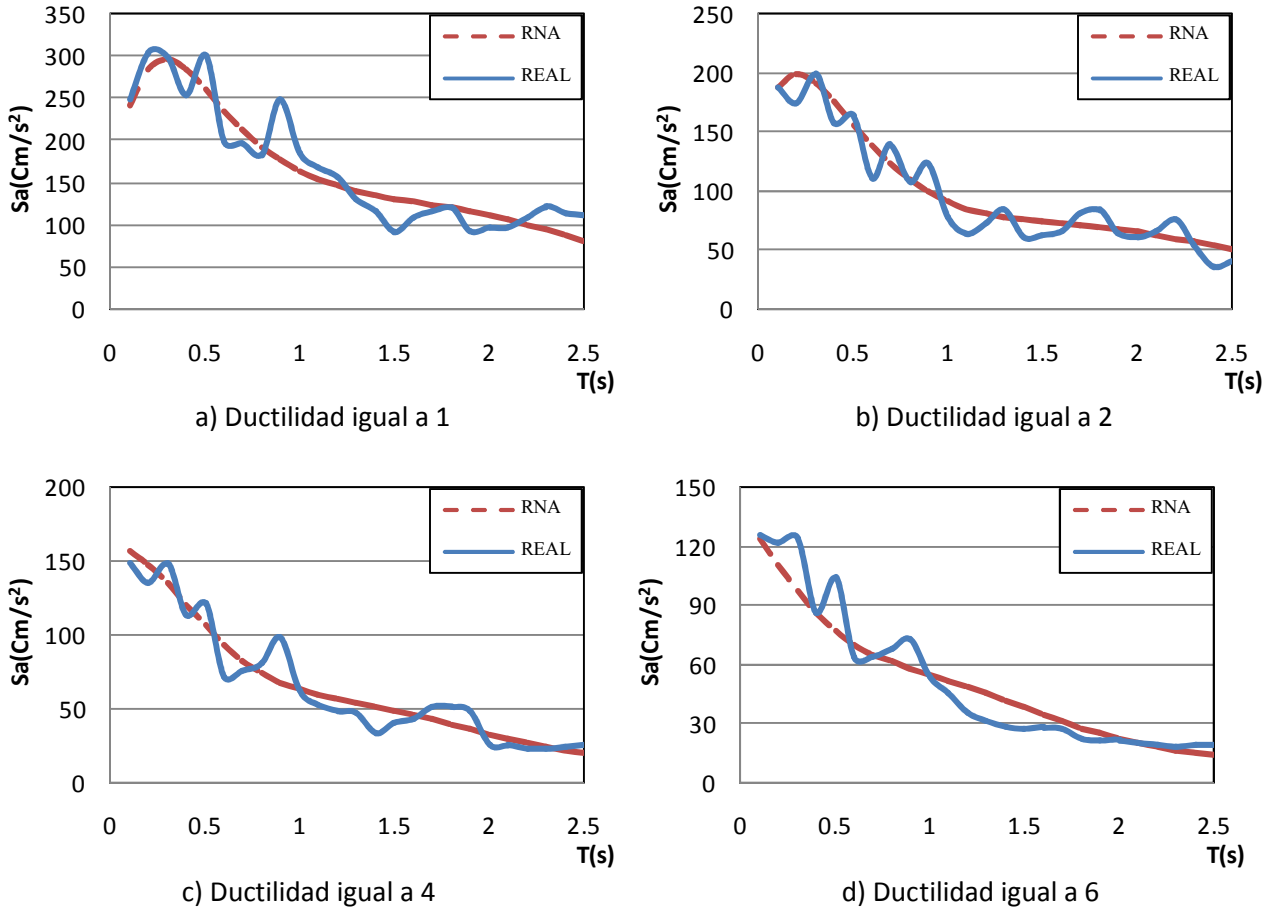
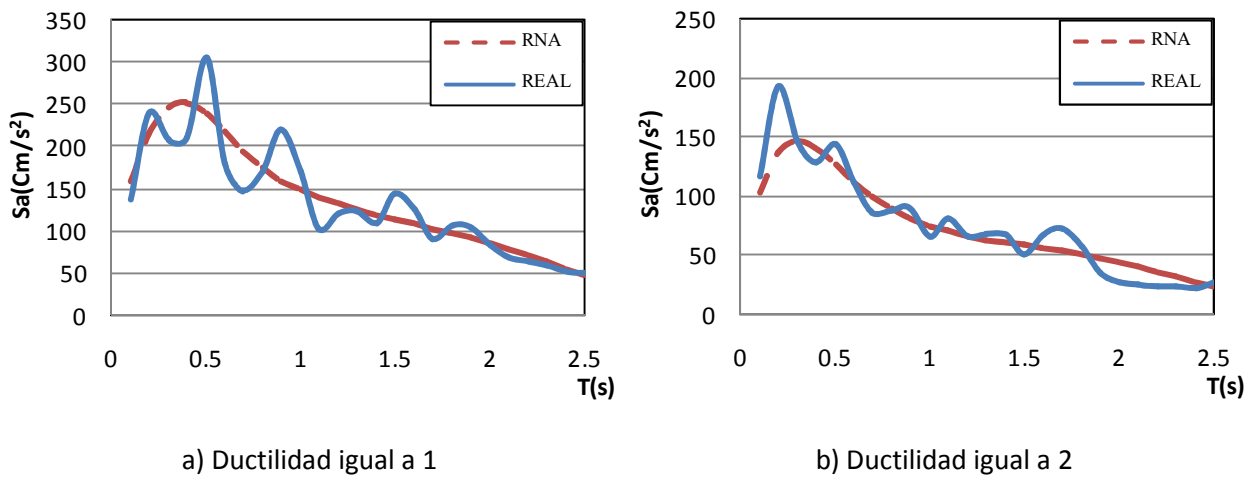
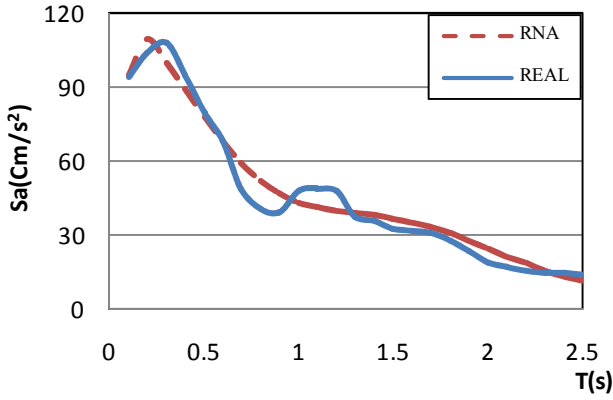
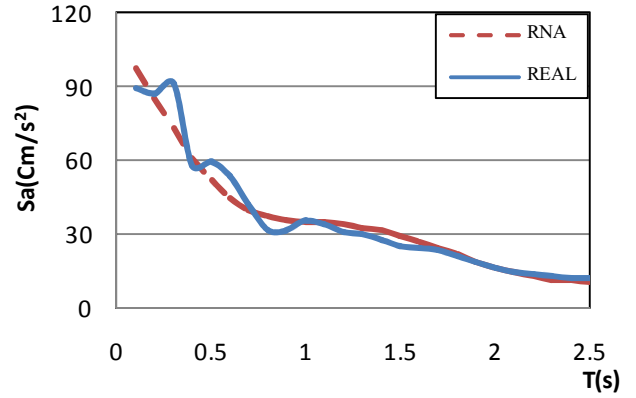


Figura 3.14 Espectros de respuesta obtenidos en la etapa de pruebas para la Zona B



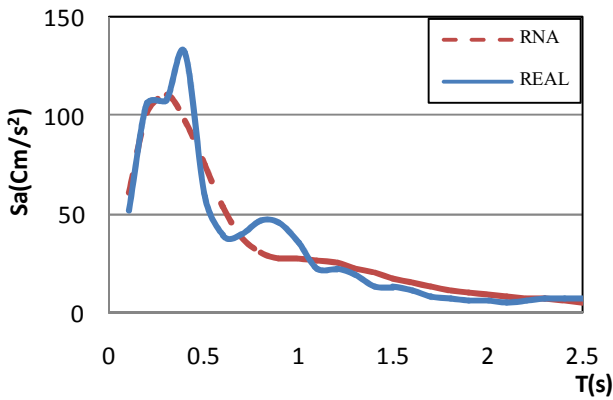


c) Ductilidad igual a 4

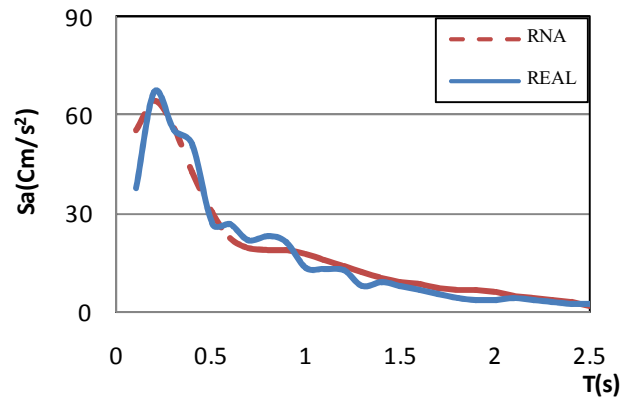


d) Ductilidad igual a 6

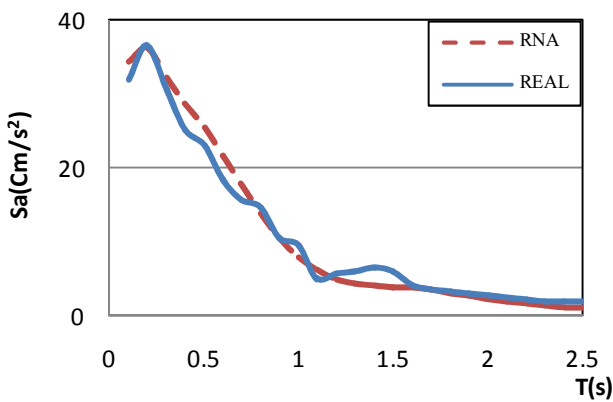
Figura 3.15 Espectros de respuesta obtenidos en la etapa de pruebas para la Zona C



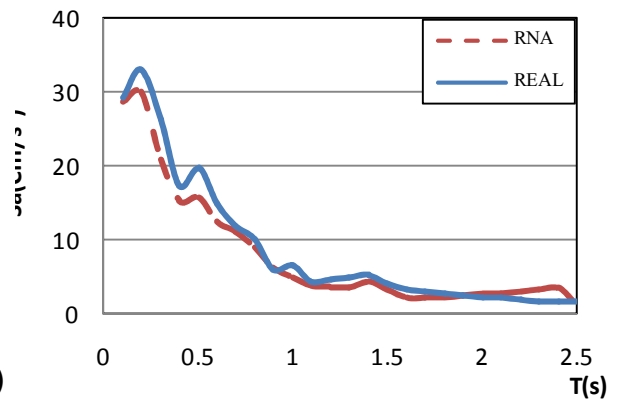
a) Ductilidad igual a 1



b) Ductilidad igual a 2



c) Ductilidad igual a 4



d) Ductilidad igual a 6

Figura 3.16 Espectros de respuesta obtenidos en la etapa de pruebas para la Zona D

Capítulo 3. Segunda aplicación de RNA: Espectros de Respuesta No-lineales

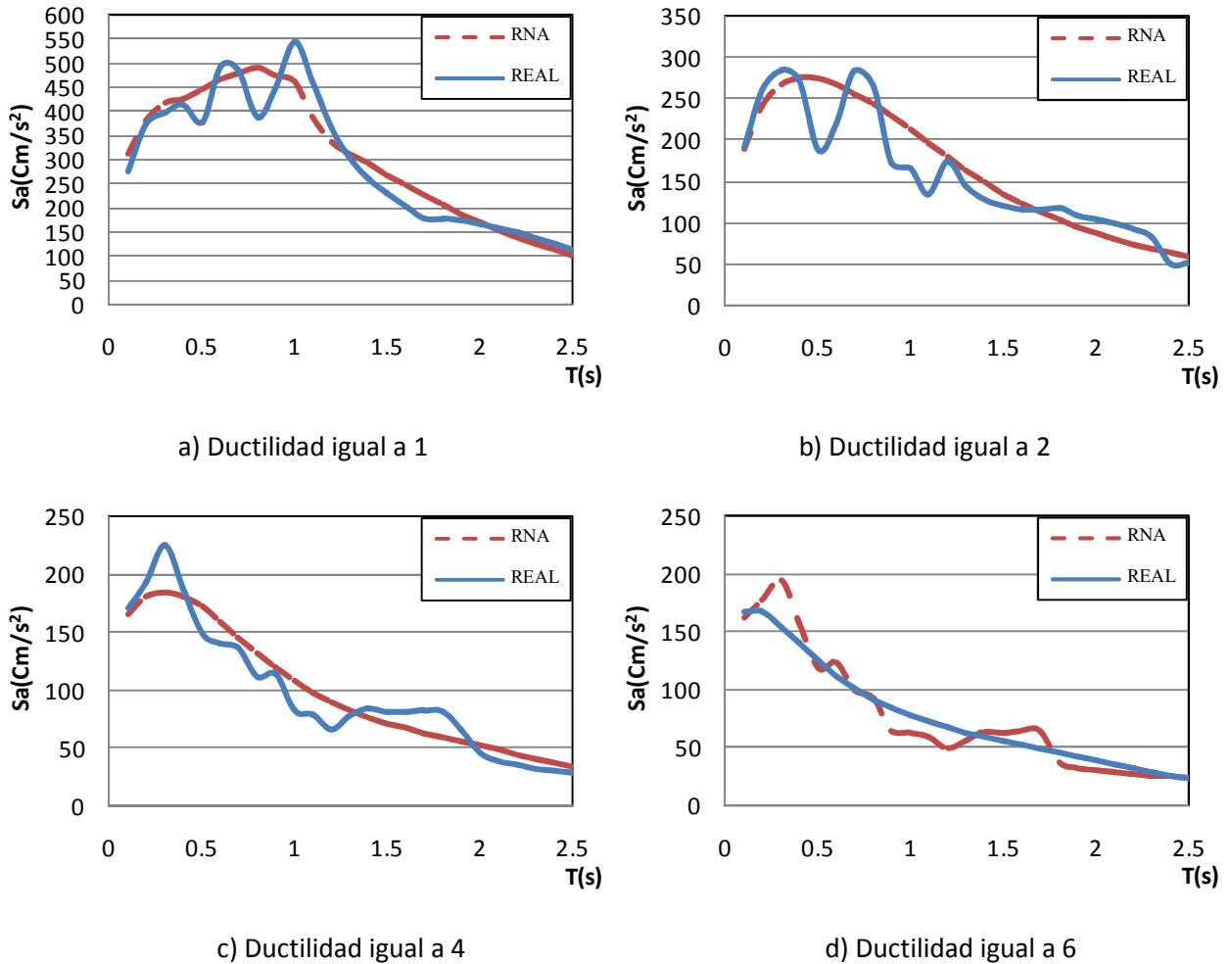


Figura 3.17 Espectros de respuesta obtenidos en la etapa de pruebas para la Zona E

Los resultados del modelo de RNA para estimar la respuesta inelásticas de S1GL sometidos a diferentes registros de aceleraciones son aceptables. Se observa que el error en la estimación crece conforme se incrementa la ductilidad. Al estimar espectros de respuesta en diferentes zonas se observa que la red tiene más dificultades para estimarlos en la zona E que en la zona A, es decir que en general el utilizar el modelo de RNA para estimar espectros de respuesta en suelos blandos genera más error que para suelos intermedios o duros. Se concluye que el modelo de RNA conserva la forma espectral de todos los espectros de respuesta.

CONCLUSIONES

El objetivo principal de esta tesis es comprobar y demostrar la eficiencia de las Redes Neuronales Artificiales (RNA) en algunos casos de aplicación en Ingeniería Civil. Se presentaron modelos de RNA's para la solución de dos problemas específicos: primero se aplicaron para encontrar factores de transformación entre respuestas máximas (distorsiones máximas de entrepiso y ductilidades globales) entre sistemas de múltiples grados de libertad (SMGL) y sus correspondientes sistemas de un grado de libertad (S1GL), garantizando una misma tasa anual de excedencia entre ambos sistemas. En segundo lugar, se aplicó un modelo de RNA para estimar espectros de respuesta sísmica inelástica.

De la primera aplicación se concluye que los factores de transformación de ductilidad $FT\mu$, resultaron cercanos a la unidad. El modelo de RNA presentó resultados prácticamente igual a los reales con errores no mayores al 8% en la fase de pruebas. Para el caso de los factores de transformación de distorsiones máximas de entrepiso ($FT\gamma$) se obtuvieron valores alrededor de 1.5, el modelo de RNA logró estimar estos factores de manera satisfactoria ya que el error máximo en las prueba fue menor que 10%.

Además, se presentó una metodología para encontrar funciones para estimar los $FT\gamma$ considerando que los sistemas estructurales modelados como SMGL y sus correspondientes S1GL tienen la misma tasa anual de excedencia. A partir de los $FT\gamma$ reales y utilizando un modelo de RNA se logró estimar la respuesta de un gran número de estructuras simuladas, con lo que se establecieron dos ecuaciones para encontrar estos factores dependientes únicamente de el coeficiente sísmico Cy , el desplazamiento de fluencia Sy y el periodo fundamental de vibración To . Esta metodología se podría seguir para establecer nuevos criterios de diseño sísmico ya que el esfuerzo computacional es reducido en gran medida si se utilizan RNA.

Del segundo ejemplo de aplicación de las RNA se concluye que el modelo de RNA para estimar espectros de respuesta inelásticos tuvo una aceptable concordancia entre los espectros reales y los obtenidos mediante las RNA. Se observó que las RNA tienen más facilidad para estimar los espectros de respuesta elásticos, ya que estos presentaron el menor error. Para el caso de los espectros inelásticos el error se fue incrementando conforme se incrementaba la ductilidad, aun así la estimación de espectros inelásticos mediante el modelo de RNA es aceptable ya que se conserva la forma espectral entre los espectros reales y los espectros estimados mediante RNA.

Es importante mencionar que para lograr un buen funcionamiento del modelo de RNA es esencial seleccionar de manera adecuada la arquitectura, su modo y algoritmos de aprendizaje, así como también los parámetros que la representen.

AGRADECIMIENTOS

Se agradece el apoyo recibido de la DGAPA-UNAM dentro del proyecto PAPIIT-IN107011-3

BIBLIOGRAFÍA

Alamilla, J. (2001), "*Criterios de diseño sísmico basados en confiabilidad para estructuras aperticadas*", Tesis para obtener el grado de Maestro en Ingeniería, DEPMI-UNAM.

Alcántara L., Ovando E. y Macías M. A. (2009), "*Estimación de espectros de respuesta en la ciudad de Puebla utilizando redes neuronales artificiales*", XVII Congreso Nacional de Ingeniería Sísmica, Puebla, Puebla.

Alves E. I. (2006), "*Earthquake forecasting using neural networks: results and future work*", *Nonlinear Dynamics*, Vol. 44, No. 1-4, pp 341-349.

Arbib M. A. (1998), "*The handbook of brain theory and neural networks*", MIT Press.

Arjun C. R. y Kumar A. (2009), "*Artificial neural network-based estimation of peak ground acceleration*", *Journal of Earthquake Technology*, Vol. 46, No. 1, pp 19-28.

Barrile V., Cacciola V., D'Amico, S., Greco A., Morabito F. C. y Parrillo F. (2006), "*Radial basis function neural networks to foresee aftershocks in seismic sequences related to large earthquakes*", *Proceedings of the 13th International Conference on Neural Information Processing*, vol. 4233 of *Lecture Notes in Computer Science*, pp 909-916.

Benioff H. (1934), "*The physical evaluation on seismic destructiveness*", *Bulletin of the Seismological Society of America*, Vol. 24, No.2, pp 398-403.

Biot M. A. (1941), "*A mechanical analyzer for the prediction of earthquake stresses*", *Bulletin of the Seismological Society of America*, Vol. 31, No. 2, pp 151-171.

Bishop C. M. (1996), "*Neural network for pattern recognition*", Clarendon Press, Oxford, pp. 482.

Bojórquez E, Ruiz S.E. y Esteva L. (2005), "*Funciones de transformación de respuestas máximas entre marcos estructurales y sus correspondientes S1GL con probabilidades de falla similares*", XV Congreso Nacional de Ingeniería Sísmica (CD), Ciudad de México.

Brío B. y Sanz A. (2002), "*Redes neuronales y sistemas difusos*", segunda edición, Alfaomega.

Bryson A. E. y Ho Yu-Chi. (1969), "*Applied optimal control: optimization, estimation, and control*", Blaisdell Publishing Company or Xerox College Publishing. pp 481.

Cardoso J.B., Almeida J.R., Dias J.M. y Coelho P.G. (2007), "*Structural reliability analysis using Monte Carlo simulation and neural networks*", *Advances in Engineering Software*, Vol. 39, No. 6, pp 505-513.

Carr A. (2000), "*RUAUMOKO*", inelastic dynamic analysis program, University of Canterbury, Department of Civil Engineering.

Chan S, Ruiz S.E. y Montiel M.A. (2005), "*Escalamiento de acelerogramas y número mínimo de registros requeridos para el análisis de estructuras*", Revista de Ingeniería Sísmica, Vol. 72, pp 1-24.

Chopra A. K. (2001), "*Dynamics of structures, theory and applications to earthquake engineering*", second edition, prentice-hall, Inc., ISBN 0-13-086973-2, Upper Saddle River, New Jersey.

Cornell C.A. (1968), "*Engineering seismic risk analysis*", Bulletin of the Seismological Society of America, Vol. 58, No. 5, pp 1583-1606.

Demuth H., Beale M. y Hagan M. (2009) "*Neural network toolbox: for use with Matlab, Mathworks*", Natick, Mass, USA.

Elkordy M., Chang K. y Lee G. (1993), "*Neural networks trained by analytically simulated damage states*", Journal of Computing in Civil Engineering. Vol. 7, No. 2, pp 130-145.

Esteva L. (1976), "*Seismicity*", Capítulo 6, Seismic risk and engineering decision, Elsevier, pp 179-224.

Esteva L., Ruiz S.E. y Rivera J. (2005), "*Reliability and performance-based seismic design of structures with energy-dissipating devices*", 9th World Seminar on Seismic Isolation, Energy Dissipation and Active Vibration Control of Structures, Kobe, Japón.

Fiesler, E. (1994), "*Neural network clasification and formalization*", Computer Standards and Interfaces, Vol. 16, special issue on Neural Networks Standards, J. Fulcher (edt), Elsevier.

Funahashi K. I. (1989), "*On the approximate realization of continuous mappings by neural networks*", Neural Networks, Vol. 2, pp 183-192.

Günaydın K. y Günaydın A. (2008), "*Peak ground acceleration prediction by artificial neural networks for northwestern Turkey*", Mathematical Problems in Engineering. Vol 2008.

Haykin S. (1999), "*Neural networks: A comprehensive foundation*", (2nd ed.). Upper Saddle River, New Jersey: Prentice Hall.

Hertz J., Krogh A. y Palmer R. (1991), "*Introduction to the theory of neural computation*", Addison-Wesley.

Hornik K., Stichcombe M. y White H. (1989) "*Multilayer feedforward networks are universal approximators*", Neural Networks, 2, pp 359-366.

Housner G. W. (1941), "*An investigation of the effects of earthquakes on buildings*", tesis doctoral, California Institute of Technology, Pasadena.

Hurtado J. y Álvarez D. (2002), "*Optimación basada en confiabilidad por medio de redes neuronales y algoritmos evolutivos*", Revista de Métodos Numéricos para el Cálculo y Diseño en Ingeniería, Vol 18, No. 4, pp 573-593.

Inoue T. y Cornell C.A. (1991), "*Seismic hazard analysis of MDOF structures*", ICASP 6, Ciudad de México, Vol. 1, pp 437-444.

Joyner W.B. y Boore D.M. (1993) "*Methods for regression analysis of strong-motion data*", Bulletin of the Seismological Society of America, Vol.83, No.2, pp.469-487.

Lee S. C. y Han S. W. (2002), "*Neural-network-based models for generating artificial earthquakes and response spectra*", *Computers and Structures*, Vol. 80, No. 20-21, pp 1627–1638.

Kandel E. R. y Hawkins R. D. (1992), "*Bases biológicas del aprendizaje y de la individualidad*", Investigación y Ciencia, Vol. 58.

Kandel E.R., Schwartz T.W y Jessel, T. M. (1999), "*Principles of neural science*", 4a edición, McGraw-Hill".

Kerh T. y Ting S. B. (2005), "*Neural network estimation of ground peak acceleration at stations along Taiwan high-speed rail system*", *Engineering Applications of Artificial Intelligence*, Vol. 18, No. 7, pp. 857–866.

Marsi S., Nakamura M., Chassiakos A. y Caughey T. (1996), "*Neuronal network approach to detection of changes in structural parameters*", *Journal of Engineering Mechanics, ASCE*, Vol. 122, No. 4, pp 350-360.

McClelland J. L. y Rumelhart D. E. (1986), "*Parallel distributed processing Vol 2: psychological and biological models*", MIT Press.

McCulloch W.S. y Pitts W. (1991), "*A logical calculus of the ideas immanent in nervous activity*", *Bulletin of Mathematical Biophysics*.

Möller O., Foschi R.O., Quiroz L.M. y Rubinstein M. (2009), "*Structural optimization for performance-based design in earthquake engineering: Applications of neural networks*", *Structural Safety*, Vol. 31, No. 6, pp 490-499.

Moravec H. (1988), "*Mind children. The future of robot and human intelligence*", Harvard University Press.

Mukherjee A. (1997), "*Self-Organizing neural networks for identification of natural modes*", *Journal of Computing in Civil Engineering, ASCE*, Vol. 11, No. 1, 74-77.

Müller B. y Reinhardt J. (1990), "*Neuronal networks. An introduction*", Springer-Verlag.

Nakamura M., Masri S., Chassiakos A. y Caughey T. (1998), "*Method for non-parametric damage detection through the use of neural networks*", *Earthquake Engineering and Structural Dynamics*, John Wiley & Sons, Ltd., pp 997-1010.

Papadrakakis M. y Lagaros D. L. (2002), "*Reliability-based structural optimization using neural networks and Monte Carlo simulation*", *Computer Methods in Applied Mechanics and Engineering*, 191, pp 3451-3507.

Principe J. C., Euliano N. R. y Lefebvre W. C. (2000), "*Neural and adaptative systems. Fundamentals through simulations*", John Wiley.

Reglamento de Construcciones para el Distrito Federal (2004), Gaceta Oficial del Gobierno del Distrito Federal, 29 de enero de 2004.

Rumelhart D.E. y McClelland J.L. (1986), "*Parallel distributed processing . Vol 1: foundations*", MIT Press.

Rosenblatt F. (1962), "*Principles of neurodynamics*", Spartan Books, New York.

Shepherd G. M. (1997), "*The synaptic organization of the brain*". 4a edición, Oxford University Press.

Shome N. y Cornell C. A. (1999), "*Probabilistic seismic demand analysis of nonlinear structures*", *Reliability of Marine Structures Program, Report No. RMS-35*, Dept. of Civil Eng., Stanford University, Stanford, CA.

Werbos P. J. (1974), "*Beyond regression: new tools for prediction and analysis in the behavioral sciences*", Tesis doctoral, Harvard University.

Zapico J., Worden, K. y Molina F. (2000), "*Structural amage assessment using neural networks*", Conference on System Identification & Structural Health Monitoring 2000, pp 387-396.