



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**

**APLICACIÓN DE REDES NEURONALES  
ARTIFICIALES PARA EL DISEÑO SÍSMICO  
DE EDIFICIOS**

**TESIS**

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO CIVIL

**PRESENTA**

MANUEL CASTELLANOS CANTÚ



MÉXICO, D.F

2013



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

## **Agradecimientos**

Deseo expresar mis agradecimientos a la Facultad de Ingeniería de la Universidad Autónoma de México por el gran trabajo que realizó con mi formación de Ingeniero Civil. Así mismo, agradezco a la división de Ingenierías Civil y Geomática y a todos los profesores que me ayudaron a forjarme como ingeniero competente.

Por otro lado quiero agradecer a Sonia E. Ruiz Gómez y a Juan Bojórquez Mora por el gran apoyo que me dieron en definir y desarrollar el tema de esta tesis. Agradezco al Instituto de Ingeniería por permitirme desarrollar este trabajo en sus instalaciones.

Por último, agradezco a la Dirección General de Asuntos del Personal Académico (DGAPA) y a su programa de apoyo a proyectos de investigación e innovación tecnológica (PAPIIT) el cual me otorgó una beca para realizar éste trabajo bajo el proyecto con número IN107011.

## Índice:

<b>AGRADECIMIENTOS.....</b>	<b>1</b>
<b>INTRODUCCIÓN .....</b>	<b>5</b>
<b>1 JUSTIFICACIÓN DEL USO DE REDES NEURONALES ARTIFICIALES (RNA) PARA EL DISEÑO SÍSMICO.....</b>	<b>6</b>
1.1 Metodología convencional de diseño sísmico .....	6
1.2 Aplicación de las RNA al diseño sísmico.....	7
1.3 Ventajas de esta aplicación .....	7
<b>2 CONCEPTOS BÁSICOS DE LAS REDES NEURONALES ARTIFICIALES.....</b>	<b>9</b>
2.1 Redes Neuronales Artificiales .....	9
2.1.1 Generación de una RNA .....	10
2.1.2 Elementos de las Redes Neuronales Artificiales .....	10
2.2 Arquitectura de las RNA.....	13
2.2.1 Número de capas .....	13
2.2.2 Tipo de conexión .....	15
2.2.3 Grado de conexión .....	16
2.3 Entrenamiento y Operación .....	17
2.3.1 Tipos de entrenamiento .....	17
2.3.2 Modelos neuronales.....	18
2.3.3 Perceptrón.....	19
2.3.4 Perceptrón multicapa .....	21
2.3.5 “Backpropagation”.....	22
2.3.6 Normalización de las entradas.....	27
2.4 Ejemplo de una RNA.....	29

2.4.1	Datos .....	29
2.4.2	Entrenamiento, validación y comprobación .....	31
2.4.3	Operación.....	33
<b>3</b>	<b>OBJETIVO .....</b>	<b>36</b>
3.1	Objetivo primario .....	36
3.2	Objetivo secundario .....	36
<b>4</b>	<b>METODOLOGÍA Y SU APLICACIÓN .....</b>	<b>37</b>
4.1	Definición de la estructura general de la red .....	37
4.1.1	Entradas .....	37
4.1.1.1	Salidas.....	40
4.1.2	Arquitectura de la RNA para diseño sísmico .....	43
4.2	Generación de los modelos de entrenamiento .....	44
4.2.1	Análisis de cargas y combinaciones de cargas .....	45
4.2.2	Aspecto sísmico .....	46
4.2.3	Diseño de los elementos de concreto reforzado.....	50
4.2.4	Espacio de entrenamiento .....	53
4.3	Entrenamiento de la RNA .....	56
4.3.1	Determinación del número de neuronas en la capa oculta .....	56
4.3.2	Resultados de entrenamiento para la RNA elegida.....	59
<b>5</b>	<b>COMPROBACIÓN DEL DISEÑO HECHOS CON MÉTODOS CONVENCIONALES Y ALTERNATIVAMENTE, CON LA RNA .....</b>	<b>65</b>
5.1	Datos del edificio a diseñar con la RNA.....	65
5.2	Operación de la red.....	65
5.2.1	Interpretación de los resultados de la red.....	68
5.3	Análisis estructural y sísmico .....	69

5.3.1	Solicitaciones de servicio .....	70
5.3.2	Solicitaciones de resistencia .....	71
5.4	Resistencia de los elementos estructurales.....	71
5.4.1	Vigas .....	72
5.4.2	Columnas .....	72
<b>6</b>	<b>CONCLUSIONES .....</b>	<b>77</b>
	<b>REFERENCIAS .....</b>	<b>79</b>
	<b>ANEXOS .....</b>	<b>81</b>
A)	Diagrama de bloques para la generación de modelos de entrenamiento .....	81
B)	Diagrama de bloques para la operación de la red en Matlab .....	82

## Introducción

El uso de Redes Neuronales Artificiales (RNA) en el campo de ingeniería sísmica se ha extendido en los últimos años [21, 22, 23 y 24]. Esto se debe al gran avance que ha tenido la teoría de las RNA en las últimas décadas. Serrano presenta una reseña histórica de la teoría de las RNA [3], en ésta se muestra como fue evolucionando la teoría y las diferentes arquitecturas hasta que se pudieron aplicar a problemas de ingeniería.

En México, Bojórquez realizó una investigación de las RNA para la estimación de espectros de respuesta inelásticos [2], sin embargo, la mayoría de los trabajos comprenden aplicaciones muy específicas y ninguna dirigida a los diseñadores de la práctica.

Una de las posibles aplicaciones de esta herramienta es el diseño de estructuras que presentan una geometría y estructuración típicas, como por ejemplo, torres de transmisión, plataformas marinas tipo "jacket", silos, tanques elevados, etc., cuya estructuración es básicamente la misma pero las dimensiones de otros diseños pueden ser diferentes.

Los edificios modernos en general son edificios irregulares que no presentan una estructuración típica, por lo que su diseño debe hacerse de manera individual; sin embargo, existen casos en los que la estructuración no varía demasiado de un edificio a otro, como por ejemplo, escuelas, edificios multifamiliares de interés social, etc. En estos casos es posible entrenar una RNA (o utilizar una red entrenada) que sea útil para realizar el pre-diseño de las estructuras.

En el presente trabajo se aplican las RNA para el diseño sísmico de edificios de geometría regular, de concreto reforzado, de hasta 10 niveles, ubicados en la zona IIIb del Valle de México. El diseño de los edificios cumple con las especificaciones del Reglamento de Construcciones para el Distrito Federal (RCDF-2004).

# 1 Justificación del uso de Redes Neuronales Artificiales (RNA) para el Diseño Sísmico

En este capítulo se compara la metodología utilizada actualmente con la alternativa de utilizar RNA para el diseño sísmico de edificios. Al final del capítulo se comentan las ventajas de utilizar las RNA.

## 1.1 Metodología convencional de diseño sísmico

Hoy en día el diseño sísmico estructural se apoya en herramientas informáticas las cuales trabajan con un modelo virtual de la estructura en el cual se representa la geometría del sistema estructural y las características de los materiales. Dicho modelo es propuesto por el usuario y demanda del conocimiento de cargas y de los materiales con los que se va a diseñar la estructura, además, es necesario conocer las dimensiones de los elementos estructurales.

Una vez cargado el modelo, el programa realiza el análisis estructural obteniendo la respuesta del edificio. Estas respuestas son las fuerzas y desplazamientos a las que se somete el edificio, y se les llama “solicitaciones”. En algunos casos los programas de cómputo tienen integradas las ecuaciones propuestas en los códigos de diseño para el cálculo de la resistencia, en otros casos, el usuario debe de hacer el cálculo de las resistencias por su cuenta. De esta forma se puede hacer la revisión propuesta en la Ecuación 1.1 y verificar que todos los elementos cumplan con las condiciones de resistencia.

$$Resistencia \geq Solicitacion$$

Ecuación 1.1

En el caso de las solicitaciones debido a las acciones sísmicas, es necesario realizar un análisis sísmico con el modelo. Muchos de los programas de análisis estructural tienen la opción de realizar un análisis dinámico modal, con el cual se determinan los modos de vibración de la estructura; es decir, la frecuencia a la cual oscilará el edificio al ser sometida a un movimiento del suelo. Las acciones que le corresponden a la estructura se determinan con base en un espectro de diseño, el cual asigna la aceleración de las masas en el sistema estructural (y por lo tanto la fuerza de inercia) en función del periodo de vibración. El espectro depende del lugar de desplante y por lo tanto, del periodo de vibración dominante en el suelo. Con dichas acciones el programa puede calcular las solicitaciones por sismo. Los códigos de diseño establecen los límites máximos a los que se puede someter la estructura. Por lo general estos límites están en función de las distorsiones de entrepiso máximas o de desplazamientos máximos, y es necesario compararlos con las solicitaciones de tal forma que satisfagan la Ecuación 1.2.

$$Distorsión\ máxima\ tolerable \geq Distorsión\ máxima\ demandada$$

Ecuación 1.2

Los paquetes de cómputo para realizar diseño sísmico tienen una gran capacidad de refinamiento. El número de opciones diferentes con las que se puede modelar la estructura, y los resultados arrojados por los modelos son prácticamente incontables. Esto es una ventaja para proyectos innovadores y



atrevidos que se realizan de vez en cuando y en los que trabajan ingenieros con muchos años de experiencia; sin embargo, en un proyecto de edificación “común” que se realiza más seguido, este nivel de complejidad puede desacelerar el proceso de diseño, y en el peor de los casos, puede llegar a generar errores por falta de conocimiento de los paquetes de diseño.

## 1.2 Aplicación de las RNA al diseño sísmico

Como alternativa a los paquetes de diseño convencionales, se propone en este trabajo aprovechar la teoría de las RNA para el diseño sísmico de edificios típicos.

Como se verá en el capítulo siguiente, las Redes Neuronales Artificiales (RNA) hacen uso de información disponible, bases de datos, para poder entregar resultados durante un proceso que se denomina de **aprendizaje**. La base de datos puede ser generada especialmente para la red o puede tomarse una base de datos existente. Una red tiene la capacidad de entregar resultados en la etapa de post-aprendizaje.

En este trabajo se demuestra que es posible generar una RNA que entregue resultados de diseño estructural en menor tiempo que un programa convencional de diseño, y que además pueda ser utilizado por personas sin experiencia en diseño. Para generar la red, es necesario recopilar una base de datos que contenga información acerca del diseño estructural de varios edificios.

La red propuesta solo puede diseñar un tipo de edificios que cumplen con ciertas características: material, región sísmica, uso del edificio, etcétera y con geometrías aceptadas por la red. Al interconectar varias RNA individuales especializadas en un tipo de edificio es posible generar una estructura de redes neuronales las cuales permitan diseñar una gran gamma de edificios con diferentes geometrías, usos, y regiones sísmicas; de tal forma que el diseño estructural para edificaciones “comunes” sea más rápido, sencillo y posible de realizarse con una herramienta de esta naturaleza.

Para que los diseños que arroja una red cumplan con los requisitos de estados límites establecidos en los códigos, se debe de entrenar con una base de datos de edificios que cumplen con estas especificaciones. Así, los resultados de la red son una especie de interpolación basada en los edificios de la base de datos. Además, es posible entrenar la red con diseños tales que el costo de los materiales y mano de obra sean óptimos, de tal forma que los diseños que arroje la red también cumplirán con estas características.

## 1.3 Ventajas de esta aplicación

Como se observa, existen varias ventajas de las RNA para diseño de edificios típicos ya que se pueden desarrollar redes que cumplan con cualquier condición deseada, siempre y cuando la base de datos cumpla con las mismas condiciones. Al lograr una RNA que entregue buenos resultados, los diseños se

pueden hacer en un tiempo muy corto, además cualquier persona que sepa utilizar una computadora podría realizar el diseño.

Además de las ventajas antes mencionadas: rapidez, facilidad de uso y que se pueden generar varias redes para cubrir un gran espacio de diseño, existen otro tipo de ventajas menos obvias. En primer lugar, el diseño es mucho menos costoso debido a que se ahorra tiempo, además el tiempo extra se podría dedicar a otro tipo de detalles como conexiones o especificaciones de otro tipo.

Al utilizar una base de datos, con una gran variedad de diseños, es posible la continua actualización de esta base de datos, con diseños de edificios con materiales de última generación y con técnicas a la vanguardia. Dicha base de datos puede ser actualizada por una empresa o instituto y ser distribuida a los despachos de diseño de toda la región, de tal forma que se pueda llevar un control de las RNA en la región y evitar que se cometan errores. Al final, todo el ahorro de tiempo y dinero se transfiere a productividad y desarrollo en el país de aplicación.

La mayor ventaja de poder aplicar la teoría de las RNA para el diseño de edificios, es tener un diseño más homólogo, que cumpla con estándares de seguridad establecidos, y que proteja la vida de los habitantes de los edificios, de tal forma que acelere el desarrollo y se eviten desastres con pérdidas de vidas humanas o dinero.

## 2 Conceptos Básicos de las Redes Neuronales Artificiales

### 2.1 Redes Neuronales Artificiales

Dentro del campo de la Inteligencia Artificial se han tratado de crear máquinas que reproduzcan las funciones del cerebro de los humanos. Las neuronas del cerebro, en comparación con la unidad de procesamiento de una computadora (CPU), procesan la información significativamente más lento; sin embargo, surge la duda del porqué teniendo unidades de procesamiento tan veloces como el CPU, las computadoras todavía no son capaces de realizar tareas relativamente sencillas para nosotros como el reconocimiento de patrones o el aprendizaje por experiencia.

Comparando las características del cerebro con las de un CPU, podemos resaltar el gran número de interconexiones que existen entre neuronas. Estas interconexiones permiten el proceso de información en paralelo, ejecutando tareas que demandan una gran cantidad de cálculos en menor tiempo que un CPU. Un gran ejemplo es el reconocimiento de una imagen; los humanos podemos distinguir rápidamente los objetos o la cara de una persona en una imagen, sin embargo una computadora se puede tomar mucho tiempo para reconocer el objeto o la persona [2].

Además, es de destacar la energía necesaria para operar. Una computadora que trabaja por horas, necesita de grandes cantidades de energía y de una unidad de enfriamiento para no sobrecalentar el sistema. Se estima que el cerebro humano usa 15 kilocalorías por hora y que un CPU (quad-core), demanda la misma cantidad de energía en 20 minutos [9].

Debido a esto, es atractivo el estudio de un modelo que emule el procesamiento del cerebro, el cual se ha desarrollado a partir de la teoría de las Redes Neuronales Artificiales (RNA). Las RNA son un sistema de procesamiento de información que a partir de la unidad básica, la neurona artificial, intentan adoptar las características del cerebro. Entre las características que intentan adoptar las RNA se pueden mencionar las siguientes:

- **Sistemas no secuenciales o lineales:** Una computadora convencional trabaja secuencialmente, es decir, ejecuta comandos uno tras otro. En cambio una RNA puede ejecutar operaciones en paralelo, lo cual le permite adaptarse a funciones no lineales.
- **Adaptabilidad:** Las RNA son sistemas que pueden adecuarse a las características de un problema. Esto lo hacen modificando la intensidad de la señal entre neuronas. A dicha intensidad se le denomina peso sináptico, y se hablará de éste más adelante.
- **Memoria distributiva:** Las variables que definen a una red están distribuidas a lo largo de toda la red. Esto permite que en dado caso que una neurona se dañe, el rendimiento de la red no será afectado significativamente.

- Generalización: Las RNA tienen la capacidad de generalizar un problema, es decir, aunque la red haya sido diseñada para casos específicos de un problema, la misma tendrá la capacidad de aproximar los resultados para casos más generales [3].

### 2.1.1 Generación de una RNA

Para generar una RNA hay que pasar por un proceso de tres etapas. La primera etapa es la definición del problema el cual se quiere resolver con la RNA, es importante tener identificadas las variables de entrada y las variables de salida. En esta misma etapa se debe de elegir la arquitectura más adecuada para el problema, en el capítulo 2.2 se hablará detalladamente de los tipos de arquitectura.

La segunda etapa consiste en entrenar la red para que trabaje adecuadamente. El entrenamiento es un proceso iterativo el cual pretende que las salidas de la red converjan a resultados en orden de magnitud aceptables. Existen gran variedad de algoritmos de entrenamiento dependiendo de la aplicación y del tipo de entrenamiento. En el capítulo 2.3 se explica detalladamente en qué consiste el entrenamiento y los diferentes algoritmos.

La última etapa es la operación de la red. En esta etapa se puede alimentar a la red con valores de entrada del problema y la red generará una señal de resultados. En este punto la red funciona como una herramienta que puede resolver el problema para el cual se diseñó.

### 2.1.2 Elementos de las Redes Neuronales Artificiales

Los elementos de una RNA se pueden clasificar en microestructura, mesoestructura y macroestructura dependiendo del nivel en el cual procesan la información. La microestructura son aquellos elementos que componen una neurona. La mesoestructura son las distintas arquitecturas que pueden formar un conjunto de neuronas, y las macroestructuras son conjuntos de RNA que pueden resolver grandes problemas [2,3].

Todos los componentes se pueden representar de manera algebraica, con variables y operadores de tal forma que en conjunto forman un modelo matemático estructurado. Es posible simular dicho modelo matemático en una computadora ordinaria, lo que nos permite darle aplicación a la teoría de las RNA.

La microestructura de las RNA son neuronas artificiales, estas se pueden descomponer en los siguientes componentes:

**Señal de entrada:** La señal de entrada de una neurona es la información que se procesa para generar una señal de salida. Por lo general esta señal se puede representar con un vector columna  $P$ , de  $q$  componentes.

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_q \end{bmatrix} \quad \text{Ecuación 2.1}$$

**Pesos sinápticos y umbral:** Los pesos sinápticos ( $W$ ) y el umbral ( $b$ ) son valores que afectan la señal de entrada siguiendo una regla de propagación. El valor de los pesos y el umbral son las variables que cambian en la etapa de entrenamiento hasta converger a un valor óptimo, de tal forma que en la etapa de operación, estos valores son constantes. Cada neurona tiene un vector de pesos del mismo tamaño que el vector de entrada, y un valor escalar para el umbral.

Pesos sinápticos y umbral:

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_q \end{bmatrix} \quad \text{Ecuación 2.2}$$

Umbral:

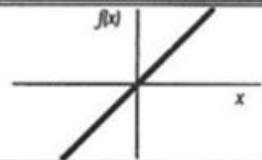
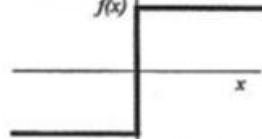
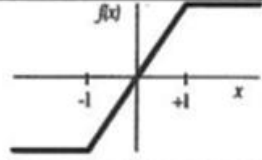

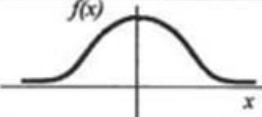
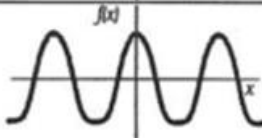
$$b$$

**Regla de Propagación:** La regla de propagación es una función en términos de la señal de entrada, los pesos sinápticos y el umbral. La regla de propagación transforma el vector de entrada en un solo valor escalar, el cual se denomina valor de propagación  $n$ . Existen varias formas para esta función; sin embargo, lo más común es utilizar una transformación lineal de la siguiente forma:

$$n = \sum_{i=1}^q w_i p_i + b \quad \text{Ecuación 2.3}$$

**Función de activación o de transferencia:** Esta función limita la amplitud de la señal de salida de la neurona, obligando a que todas las neuronas en la red trabajen dentro de un mismo intervalo. En el siguiente cuadro (Tabla 2.1) se muestran algunas de las funciones de activación más comunes.

Tabla 2.1: Funciones de activación

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

**Señal de salida:** La señal de salida de la neurona es un valor escalar resultado de la función de activación la cual está en términos del valor de propagación.

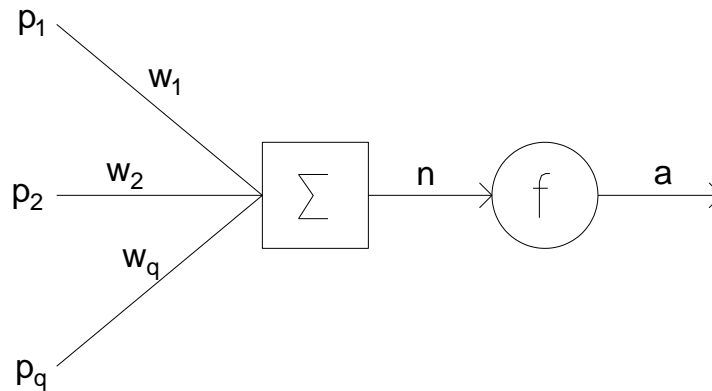
$$a = f(n) = f\left(\sum_{i=1}^q w_i p_i + b\right)$$

Ecuación 2.4

donde:

- $a$ : Señal de salida de una neurona
- $f$ : Función de activación
- $n$ : Valor de propagación
- $w$ : Peso sináptico
- $p$ : Señal de entrada
- $b$ : Umbral

La Figura 2.1 muestra el modelo de neurona artificial con todos los componentes antes mencionados. Hay que recordar que la señal siempre fluye en una dirección (ver Figura 2.1). La dirección de flujo de la información es de izquierda a derecha.



**Figura 2.1: Modelo de neurona artificial**

## 2.2 Arquitectura de las RNA

Un conjunto de neuronas interconectadas conforman una red. Existe una gran variedad de formas en las que se pueden organizar y conectar las neuronas, a cada una de estas formas se le llama topología o arquitectura.

Existen diversos criterios para clasificar las arquitecturas neuronales de acuerdo con sus características. Se encuentra la clasificación según el **número de capas**, según el **tipo de conexión** y por último, según el **grado de conexión**. A continuación se explicara cada uno de estos criterios.

### 2.2.1 Número de capas

Las neuronas se organizan en unidades estructurales con el nombre de capas, las neuronas de una misma capa procesan información en paralelo. Cada capa de neuronas procesa información en distintos niveles, de tal forma que la información fluye de las capas inferiores a las capas superiores. Una red puede tener tantas capas como sea necesario.

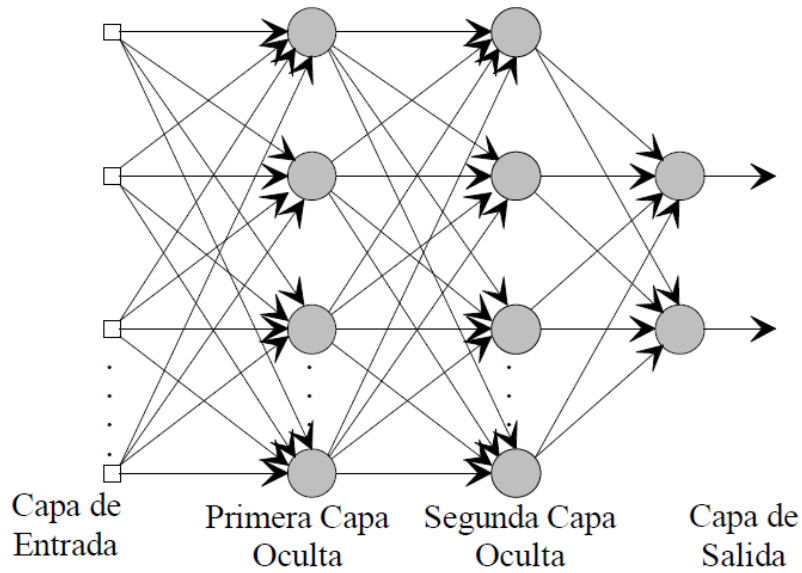


Figura 2.2: Jerarquía de capas en una red

Las capas con las que interactúa un usuario son las capas de entrada y de salida. La capa de entrada trabaja con la información con la que dispone el usuario y la capa de salida entrega la información que busca el usuario. Las capas con las cuales el usuario no tiene interacción se le denominan capas ocultas, por lo general son las capas intermedias de una red.

La red más sencilla tiene una estructura **monocapa** (Figura 2.3), esto quiere decir que solo cuenta con una sola capa en la que procesa la información. Después están las redes **multicapas** (Figura 2.4) las cuales contienen una o más capas ocultas. [2]

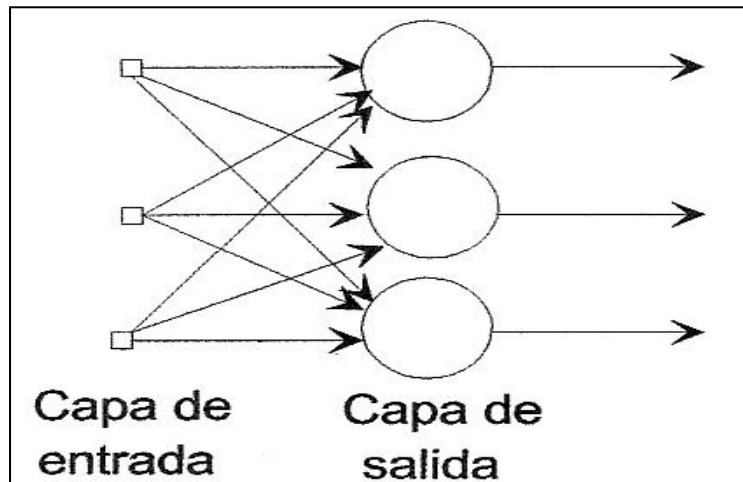


Figura 2.3: Red monocapa



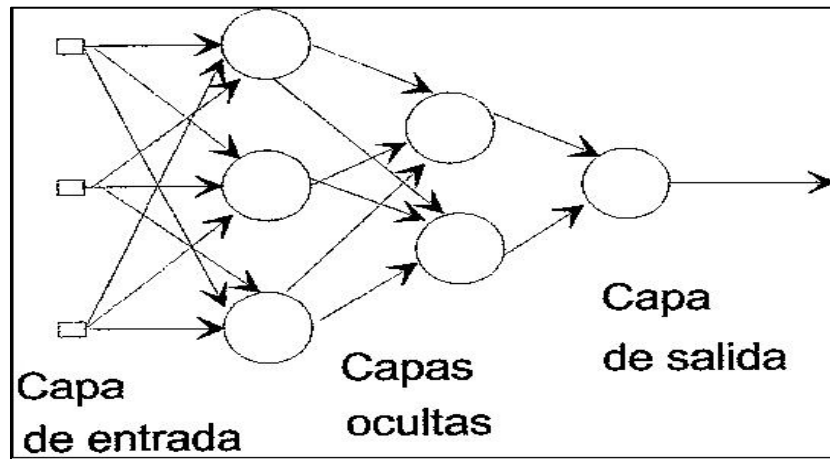


Figura 2.4: Red multicapa

### 2.2.2 Tipo de conexión

Al igual que las neuronas biológicas, las conexiones entre dos neuronas artificiales se realizan de manera direccional, esto quiere decir que la información solo fluye en un sentido. Debido a esta propiedad, surge la clasificación según el tipo de conexión.

Cuando la conexión entre dos neuronas se establece en la misma capa, se le denomina conexión lateral o intra-capa. A las conexiones establecidas entre dos neuronas de distintas capas se le denomina conexión inter-capa o laterales.

Las redes que solo cuentan con conexiones hacia adelante, es decir, conexiones entre neuronas de capas inferiores a neuronas de capas superiores se les denomina **no recurrentes** (Figura 2.5). A este tipo de redes también se les llama de *"feedforward"* o de propagación hacia adelante debido al sentido que lleva la información. Este tipo de redes son las más desarrolladas hasta ahora en aplicaciones prácticas.

El segundo tipo son las redes **recurrentes** (Figura 2.6) las cuales consisten de redes multicapa con conexiones hacia atrás, es decir, conexiones entre neuronas de una capa superior hacia una capa inferior, o redes con conexiones laterales. [2]

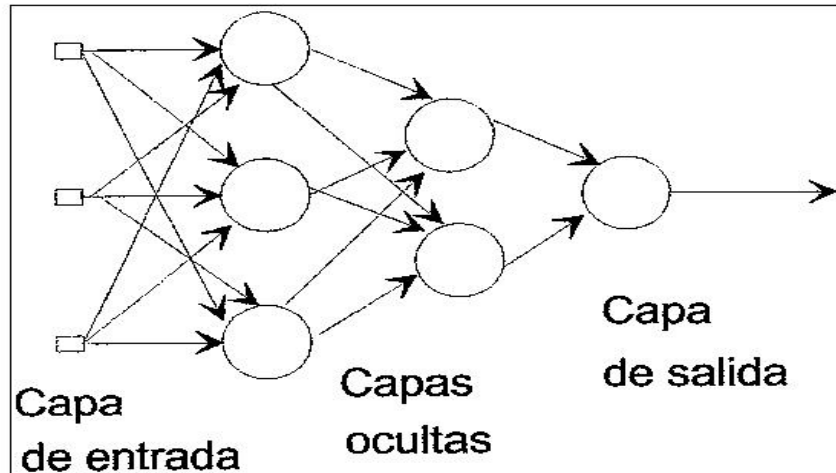


Figura 2.5: Red No recurrente

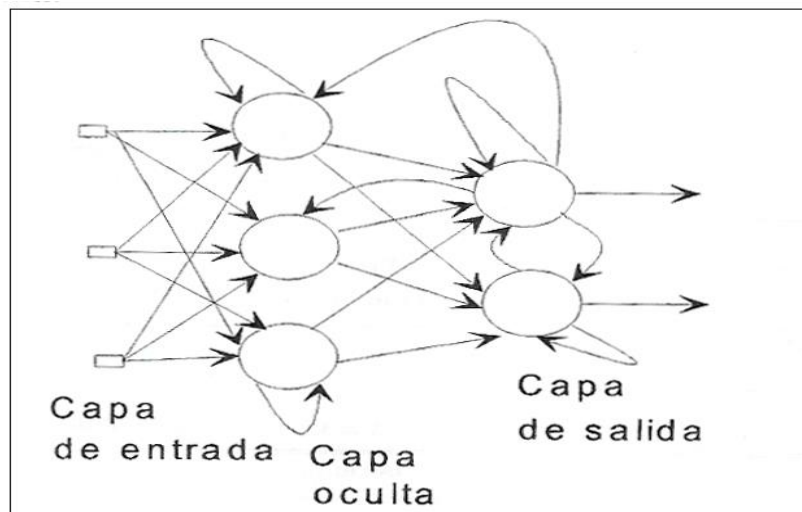


Figura 2.6: Red recurrente

### 2.2.3 Grado de conexión

No existe una regla la cual establezca que una neurona debe de estar conectada con todas las neuronas de la capa siguiente. Esto lleva a que podamos clasificar las redes en **totalmente conectadas** (Figura 2.7) y en **parcialmente conectadas** (Figura 2.8). Las primeras obedecen dicha regla y las segundas tienen la libertad de omitir conexiones entre neuronas. [3]

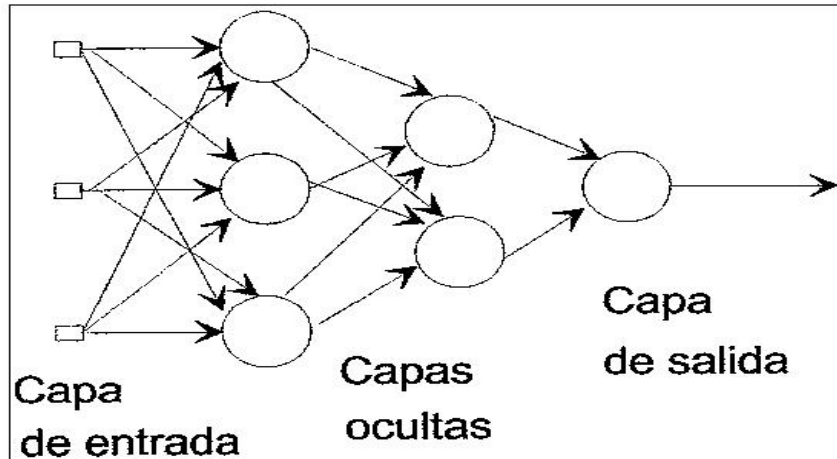


Figura 2.7: Red totalmente conectada

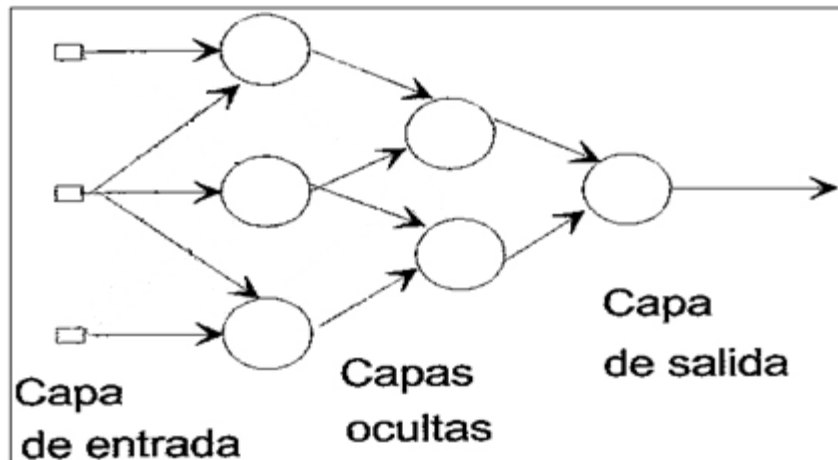


Figura 2.8: Red parcialmente conectada

## 2.3 Entrenamiento y Operación

### 2.3.1 Tipos de entrenamiento

Para que una red pueda operar y entregar buenos resultados, primero hay que entrenarla, es decir, llevarla a un estado de comportamiento deseado. Como se comentó anteriormente el entrenamiento consiste en el ajuste del vector de pesos y del umbral de cada neurona, de tal forma que el resultado que arroja la red debe de seguir un patrón deseado. A la etapa de entrenamiento también se le llama de aprendizaje debido a la semejanza que este tiene con el aprendizaje de un animal.

En el entrenamiento hay que seguir una serie de pasos que ajustan los parámetros de la red. A esta serie de pasos se le llama algoritmo de entrenamiento o regla de aprendizaje. Existen dos tipos de algoritmos de entrenamiento básicos: el supervisado y el no supervisado. De estos,

derivan otros entrenamientos como el híbrido y el reforzado; Sin embargo, en este trabajo solo nos enfocaremos al entrenamiento supervisado y el no supervisado.

El **entrenamiento supervisado** consiste en presentarle a la red un conjunto de entradas y salidas de las cuales a cada entrada le corresponde una salida. A este conjunto de entradas y salidas se le llama patrón de entrenamiento. La red debe procesar la información de entrada y generar una nueva señal de salida. La nueva señal generada se compara con la salida presentada en el patrón, de tal forma que el algoritmo genera una señal de error con la cual modifica o actualiza los parámetros de la red.

En el **entrenamiento no supervisado**, a la red solo se le presentan patrones de entrada. Consiste en determinar la distribución de probabilidad de los patrones presentados; dado esto, la red necesita un gran número de patrones de entrada. La red, a partir del algoritmo de entrenamiento reconoce las regularidades en el conjunto de entrada y permite clasificarlos según similitudes. Este tipo de entrenamiento es útil para la agrupación de datos [2].

### 2.3.2 Modelos neuronales

Dentro de la corta historia de las RNA se han estudiado una variedad de **modelos neuronales**. Los modelos neuronales obedecen una arquitectura y un algoritmo de entrenamiento específico. La elección de un modelo neuronal depende del tipo de problema que se quiere resolver, así existen modelos útiles para el ajuste de funciones y modelos útiles para la clasificación de datos. Debido a esto, cada modelo tiene diferentes enfoques y está limitado a ciertas condiciones, por lo que es común que para resolver un problema totalmente nuevo en el campo de las RNA se tenga que desarrollar un modelo nuevo.

Existen diversas maneras de clasificar los modelos neuronales, arquitectura, regla de aprendizaje, función de activación o valores de entrada. La Universidad Tecnológica de Pereira [4] presenta el siguiente árbol de clasificación, donde se presentan algunos de los modelos de RNA más comunes.

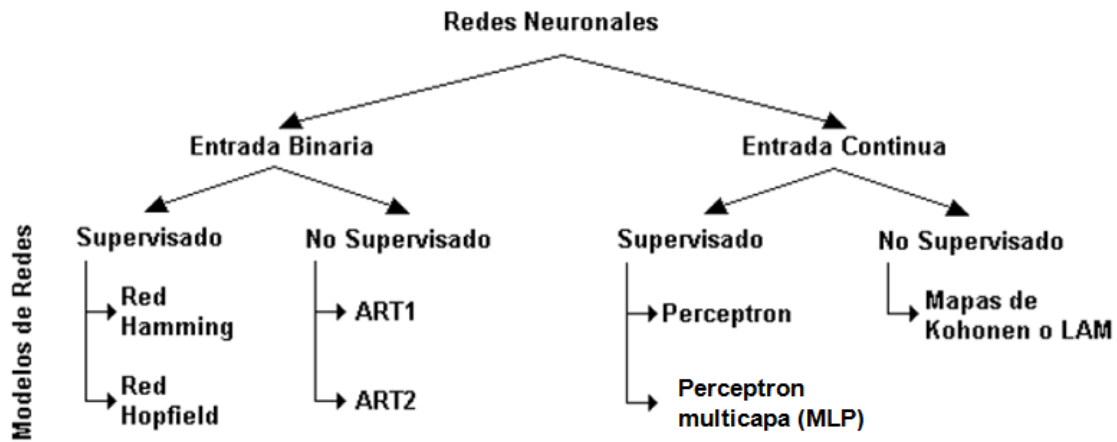


Figura 2.9: Clasificación de los modelos neuronales [4]

Este trabajo se enfocará en el perceptrón multicapa o MLP por sus siglas en inglés. Para entender el MLP es necesario estudiar primero el perceptrón.

El **perceptrón** es de gran importancia en el campo de las RNA debido a que fue una de las primeras redes que se entrenaron con reglas de aprendizaje supervisadas. Además, ayuda conocer su funcionamiento ya que de aquí se derivó el algoritmo “backpropagation”, del cual se hablará más adelante.

### 2.3.3 Perceptrón

El perceptrón tiene una arquitectura monocapa de una neurona. La función de activación que se utiliza en la neurona es la función escalón signo (ver Tabla 2.1), de tal forma que las salidas tienen una forma binaria. La Figura 2.10 muestra el esquema general de un perceptrón.

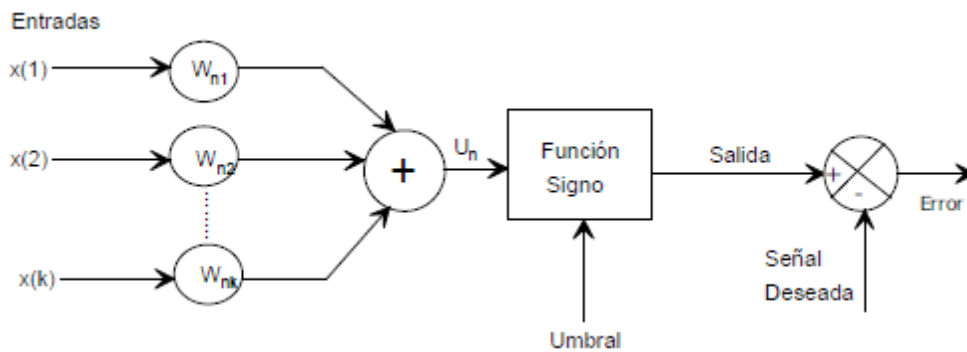


Figura 2.10: Esquema general del Perceptrón

La regla de aprendizaje del perceptrón, es de tipo supervisada de tal forma que se necesita de un conjunto de datos de entrada, y sus respectivas salidas deseadas. A continuación se explica a detalle el algoritmo de entrenamiento del perceptrón [4].

Con la Ecuación 2.4 se genera la salida de la red, dándole valores aleatorios a los pesos y al umbral. Una vez que se genera una señal de salida, ésta se compara con la señal deseada. Esta comparación da lugar a una señal de error, la cual permite actualizar los pesos.

$$\Delta W_t = (t - a)P \quad \text{Ecuación 2.5}$$

donde:

$\Delta W_t$ : Señal de error de pesos en el ciclo  $t$  subíndice  
 $P$ : Patrón de entrenamiento  
 $t$ : Señal deseada para el patrón de entrenamiento  $P$

Los pesos de red se actualizan con la siguiente expresión:

$$W_{t+1} = W_t + \Delta W_t \quad \text{Ecuación 2.6}$$

donde:

$W_t$ : Matriz de peso de cualquier neurona artificial  
 $W_{t+1}$ : Matriz de pesos corregidos

Se deben de realizar tantos ciclos  $t$  como sean necesarios hasta que los pesos en dos ciclos consecutivos no cambien, es decir, sean constantes. Cuando se llega a pesos constantes se dice que la red ha sido entrenada. Una vez entrenada, se puede operar el perceptrón con la Ecuación 2.4 y utilizando cualquier entrada. Para distinguir entre una red en entrenamiento y una en operación, generalmente los valores de entrada  $P$  cambian por la variable  $X$ .

Del estudio del perceptrón se demostró que siempre que existan patrones linealmente separables, el perceptrón llega a una solución del problema [6]. Un problema linealmente separable es cuando existe una recta que separa los elementos de dos clases diferentes (ver Figura 2.11).

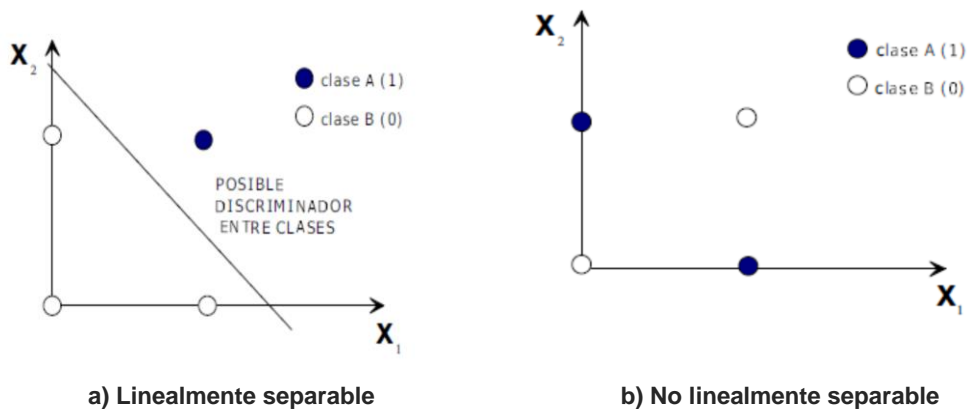


Figura 2.11: Diferentes tipos de problema

Debido a que el perceptrón no tiene la capacidad para resolver problemas que no son linealmente separables (como el de la Figura 2.11), se debe recurrir a otros modelos neuronales como el perceptrón multicapa.

### 2.3.4 Perceptrón multicapa

El Perceptrón multicapa (MLP) es un modelo neuronal con una arquitectura “feedforward”, con la característica de utilizar funciones de activación sigmoidea en las capas ocultas, y función identidad en la capa de salida (ver Tabla 2.1). Este modelo permite el uso de cualquier número de capas ocultas, siempre y cuando haya más de una.

El MLP utiliza el algoritmo de entrenamiento “backpropagation”, el cual es similar al algoritmo que utiliza el perceptrón pero con la variante de que la señal de error se calcula con el gradiente del error medio cuadrático (MSE). La razón de utilizar funciones sigmoideas e identidad es porque estas funciones son derivables y por lo tanto es posible la obtención del gradiente [2]. A continuación se muestra la forma de estas funciones y su respectiva derivada:

Tabla 2.2: Función de transferencia sigmoidea y función identidad

Función	Forma	Derivada
<i>Logsig</i>	$f(n) = \frac{1}{1 + e^{-n}}$	$f'(n) = f(n)(1 - f(n))$
<i>Tansig</i>	$f(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	$f'(n) = 1 - (f(n))^2$
<i>Identidad</i>	$f(n) = n$	$f'(n) = 1$

En la siguiente sección se explicará el algoritmo “backpropagation” y se desarrollarán las expresiones matemáticas que el algoritmo requiere.

### 2.3.5 “Backpropagation”

Al igual que el algoritmo que se utiliza en el perceptrón, el algoritmo “backpropagation” tiene dos etapas, una etapa de propagación hacia adelante donde se genera una señal de salida y otra etapa de propagación hacia atrás donde se compara la salida deseada con la salida generada y se actualizan los parámetros de la red.

Para iniciar el entrenamiento hay que asignar una matriz de pesos y un umbral aleatorios para cada neurona, después, conforme el algoritmo va avanzando, estos valores se modificaran para que converjan a una solución.

Para poder entrenar la red es necesario conocer el número de capas ocultas y el número de neuronas en cada capa. Además, también se requiere de los patrones de entrenamiento y sus respectivas salidas. A continuación se muestran ambos requerimientos:

La Figura 2.12 muestra una red con tres capas de propagación hacia adelante.

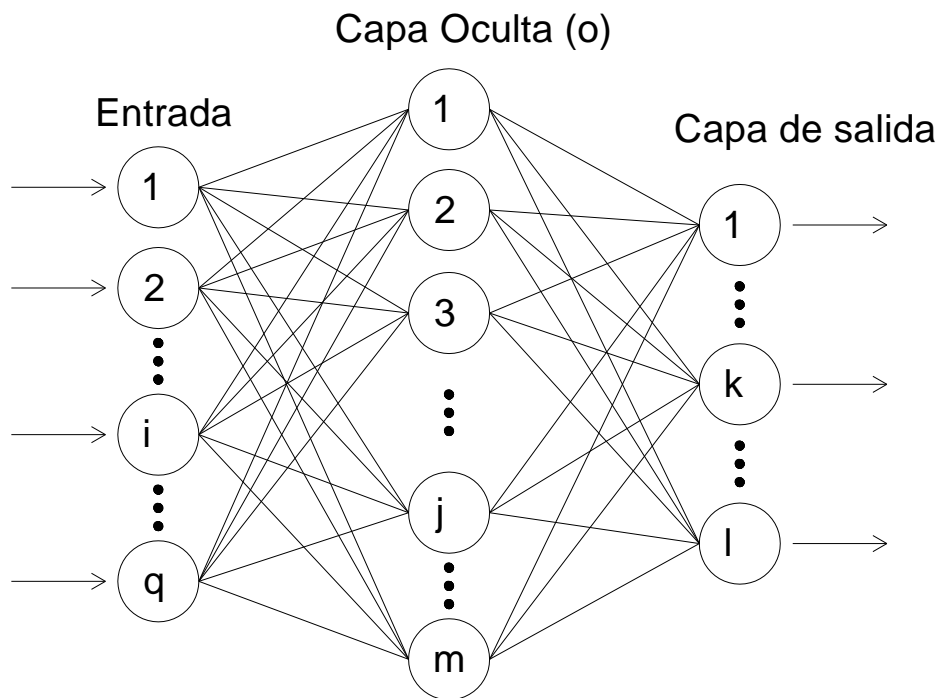


Figura 2.12: Red de propagacion hacia delante de tres capas



donde:

$q$ : Número de componentes del vector de entrada

$m$ : Número de neuronas de la capa oculta

$l$ : Número de neuronas en la capa de salida

Además, se tiene una serie de entradas y sus respectivas salidas.

$$\{P_1, T_1\}, \{P_2, T_2\}, \dots, \{P_r, T_r\} \quad \text{Ecuación 2.7}$$

donde:

$P_r$ : Patrón de entrenamiento, el tamaño de este vector es igual a  $q$

$T_r$ : Salida deseada correspondiente patrón de entrenamiento  $P_r$ , el tamaño de este vector es  $l$

$r$ : Tamaño del espacio de entrenamiento

Ahora se procede a entrenar la red. El primer paso es asignar valores aleatorios a los pesos y al umbral de cada neurona, para poder distinguir cada uno de estos se utilizaran los subíndices como los indicadores de cada neurona y los superíndices como indicadores de capa:

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

$w_{lj}^s$ : Peso que une la componente  $j$  de la capa oculta con la neurona  $l$  de la capa de salida

$b_j^o$ : Umbral de la neurona  $j$  de la capa oculta

$b_l^s$ : Umbral de la neurona  $l$  de la capa de salida

Para realizar la **propagación hacia adelante** es necesario transmitir la información por cada una de las capas hasta llegar a la capa de salida. En este caso solo tenemos dos capas, así que la señal de salida de cada capa se calcula con las ecuaciones dadas por:

Señal de salida de la capa oculta:

$$a_j^o = f(n_j^o) = f^o \left( \sum_{i=1}^q w_{ji}^o p_i + b_j^o \right) \quad \text{Ecuación 2.8}$$

Señal de salida final de la red:

$$a_k^s = f(n_k^s) = f^s \left( \sum_{j=1}^m w_{kj}^s a_j^o + b_k^s \right) \quad \text{Ecuación 2.9}$$

donde:

$f^o$ : Función de transferencia de la capa oculta (función Tansig o función Logsig)

$f^s$ : Función de transferencia de la capa de salida (función Identidad)

Una vez calculado el vector de salida de la red  $A^s$ , se compara con la salida deseada  $T$  y se calcula el error medio cuadrático con la expresión:

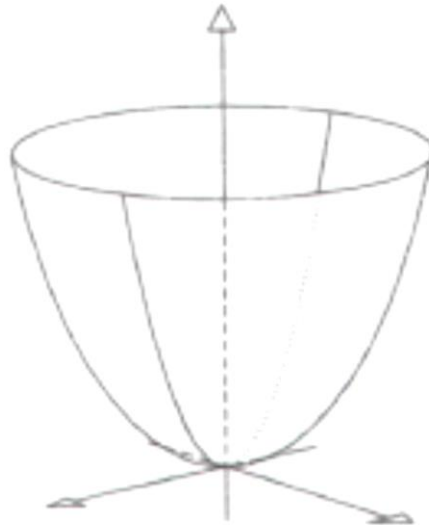
$$ep^2 = \frac{1}{2} \sum_{k=1}^l (\delta_k)^2 = \frac{1}{2} \sum_{k=1}^l (t_k - a_k^s)^2 \quad \text{Ecuación 2.10}$$

donde:

$ep^2$ : Error medio cuadrático para cada patrón de entrada  $p$

$\delta_k$ : Error en la neurona  $k$  de la capa de salida con  $l$  neuronas

El objetivo del algoritmo “backpropagation” es minimizar el error medio cuadrático para que la señal generada y la señal deseada sean iguales. Si se desarrolla el error medio cuadrático en función de los pesos sinápticos, se puede observar que la expresión resultante se asemeja a la de un hiperparaboloide, esto es, un paraboloide en un espacio mayor de tres dimensiones (Figura 2.13).



**Figura 2.13: Superficie del error medio cuadrático**

Debido a la geometría de la superficie, esta tiene un mínimo global (Figura 2.13). El punto donde se encuentra el mínimo es el de interés dado a que es la solución al problema. Para poder minimizar el error es necesario calcular el gradiente negativo del error medio cuadrático respecto a los pesos

sinápticos. De tal forma que los pesos y los umbrales se actualizan con la formula general de la regla delta:

$$\xi_{t+1} = \xi_t + \alpha \nabla e p^2 \quad \text{Ecuación 2.11}$$

donde:

$\xi_{t+1}$ : Parámetro actualizado  
 $\xi_t$ : Parámetro que se actualiza  
 $\nabla e p^2$ : Gradiente del error medio cuadrático  
 $\alpha$ : Constante de adaptación

El gradiente negativo  $-\nabla e p^2$ , indica la dirección donde se encuentra el mínimo global; sin embargo, no la distancia a la que se encuentra, para esto se introduce  $\alpha$ , una constante de adaptación o rata de aprendizaje que ayuda a acelerar el proceso de convergencia [4].

Para la determinar el gradiente respecto a los pesos sinápticos se tiene que recurrir a conceptos del cálculo vectorial. En este trabajo no se presenta la deducción de las expresiones para determinar dicho gradiente, en Bojórquez (2011) se encuentran más detalles al respecto [2].

Durante la **propagación hacia atrás**, primero hay que actualizar los pesos y los umbrales de las neuronas en la capa de salida, una vez hecho esto, se procede a actualizar los pesos y los umbrales de la capa oculta.

La expresión para determinar el gradiente negativo de la capa de salida es:

$$-\nabla e p^2 = -\frac{\partial e p^2}{\partial w_{kj}^s} = (t_k - a_k^s) \times f'^s(n_k^s) \times a_j^o \quad \text{Ecuación 2.12}$$

Para facilitar los cálculos, a los primeros dos términos de la expresión anterior se le llama sensibilidad de salida  $S^s$ , de tal forma que la expresión queda de la siguiente manera:

$$-\nabla e p^2 = -\frac{\partial e p^2}{\partial w_{kj}^s} = s_k^s a_j^o \quad \text{Ecuación 2.13}$$

$$s_k^s = (t_k - a_k^s) \times f'^s(n_k^s) \quad \text{Ecuación 2.14}$$

donde:

$s_k^s$ : Sensibilidad de salida para la neurona  $k$  de la capa de salida.

Sustituyendo la Ecuación 2.13 en la Ecuación 2.11 se obtiene la siguiente expresión para actualizar los pesos y los umbrales de la capa de salida:

$$(w_{kj})_{t+1} = (w_{kj})_t - \alpha s_k^s a_j^o \quad \text{Ecuación 2.15}$$

$$(b_k)_{t+1} = (b_k)_t - \alpha s_k^s \quad \text{Ecuación 2.16}$$

La expresión para determinar el gradiente negativo en la capa oculta es:

$$-\nabla_{ep^2} = -\frac{\partial ep^2}{\partial w_{ji}^o} = \sum_{k=1}^1 s_k^s \times f'^o(n_j^o) \times w_{kj}^s \times p_i \quad \text{Ecuación 2.17}$$

De igual manera, para simplificar la expresión, se le denominó sensibilidad de la capa oculta  $s_j^o$ , a los primeros dos términos, quedando la Ecuación 2.17 de la siguiente forma:

$$-\nabla_{ep^2} = -\frac{\partial ep^2}{\partial w_{ji}^o} = s_j^o \times w_{kj}^s \times p_i \quad \text{Ecuación 2.18}$$

$$s_j^o = \sum_{k=1}^1 s_k^s \times f'^o(n_j^o) \quad \text{Ecuación 2.19}$$

donde:

$s_j^o$ : Sensitividad de la capa oculta para la neurona  $j$  de la capa oculta.

Sustituyendo la Ecuación 2.18 en la Ecuación 2.11 se obtiene la siguiente expresión para actualizar los pesos y los umbrales de la capa oculta [4]:

$$(w_{ji})_{t+1} = (w_{ji})_t - \alpha s_j^o p_i \quad \text{Ecuación 2.20}$$

$$(b_j)_{t+1} = (b_j)_t - \alpha s_j^o \quad \text{Ecuación 2.21}$$

Una vez que se actualizan los parámetros de todas las neuronas se completa un ciclo de entrenamiento. El primer ciclo de entrenamiento debe de hacerse con el patrón de entrenamiento  $\{P_1, T_1\}$ . El siguiente ciclo de entrenamiento debe de realizarse con el siguiente patrón  $\{P_2, T_2\}$ , y así

sucesivamente. Una vez que se utilizan todos los patrones de entrenamiento y se actualizan los parámetros de la red  $r$  veces, se ha completado una **época**.

Se deben de completar tantas épocas como sean necesarios, de tal forma que el error calculado con la Ecuación 2.10 sea aceptable. El número de épocas requeridas esta implícitamente definido por la constante de adaptación  $\alpha$ , ya que esta define el tamaño del paso que se da en cada ciclo; sin embargo, si se utiliza una constante de adaptación muy alta, se corre el riesgo de sobrepasar el error mínimo, y jamás aproximarse al mínimo global.

Debido a que es posible que se requieran de varias épocas para llegar a un error aceptable, Bishop [7] propuso una variante para calcular el error y poder realizar las épocas en menor tiempo y menor costo computacional. La expresión que propuso es:

$$ep^2 = \frac{1}{2r} \sum_{u=1}^r \sum_{k=1}^l [(\delta_k)^2]_u \quad \text{Ecuación 2.22}$$

El entrenamiento realizado con la expresión Ecuación 2.10 se le llama entrenamiento en línea, debido a que el error medio cuadrático se calcula ciclo por ciclo hasta completar  $r$  ciclos. En cambio, al entrenamiento realizado con la expresión Ecuación 2.22 se le llama “*batch*” debido a que el error medio cuadrático se calcula en bloque con todos los patrones de entrenamiento presentados a la red.

En ambos casos, la muestra de entrenamiento tiene que ser lo suficientemente grande para envolver el problema, y los patrones de entrenamiento deben de estar bien repartidos a lo largo de todo el espacio del problema. Si no se cumple con ambas características, se puede incurrir a problemas de falta de generalización, en donde la red entrenada solo puede resolver una parte del problema y no engloba todos los casos posibles.

### 2.3.6 Normalización de las entradas

Las entradas para un problema dado pueden representar cualquier característica física o propiedad del problema. A manera de ejemplo, supóngase una red que determine el momento resistente de una viga de concreto reforzado. Naturalmente, las entradas de la red serían la geometría de la viga, el área de acero de refuerzo y las características de los materiales. Las unidades que se utilizan en la práctica para representar estas características son centímetros, centímetros cuadradas y kilogramos sobre centímetros cuadrados. Dada la diferencia en el orden de magnitud de los datos con los que trabajaría la red, ésta puede presentarse problemas de convergencia. Para evitar este problema, es recomendable la normalización de los datos de entrada.

La normalización de los datos de entrada consiste en realizar una transformación, de tal forma que el orden de magnitud de los datos de entrada se mantenga dentro de un límite establecido. Existen diversas maneras de llevar a cabo esta transformación, la más común es una transformación lineal, en donde el valor mínimo de la muestra se asocia a un mínimo general y el valor máximo de la muestra a un máximo general [8].

Para un patrón de entrenamiento:

$$P_r = \begin{bmatrix} p_1 \\ \vdots \\ p_i \end{bmatrix}$$

$$p_i^* = \left[ \frac{p_i - p_{min}}{p_{max} - p_{min}} \right] [p_{max}^* - p_{min}^*] + p_{min}^* \quad \text{Ecuación 2.23}$$

donde:

$p_i$ : Valor de entrada

$p_i^*$ : Valor normalizado para la entrada  $p_i$

$p_{min}$ : Valor mínimo de la entrada  $p_i$  para todos los patrones de entrenamiento  $r$

$p_{max}$ : Valor máximo de la entrada  $p_i$  para todos los patrones de entrenamiento  $r$

$p_{min}^*$ : Valor mínimo normalizado, asociado a  $p_{min}$

$p_{max}^*$ : Valor máximo normalizado, asociado a  $p_{max}$

Por lo general el rango de normalización se utiliza entre menos uno y uno, es decir  $p_{min}^* = -1$  y  $p_{max}^* = 1$ , quedando la expresión anterior como:

$$p_i^* = 2 \left[ \frac{p_i - p_{min}}{p_{max} - p_{min}} \right] - 1 \quad \text{Ecuación 2.24}$$

Si la a la red se le presenta entradas normalizadas, entonces entregará salidas normalizadas. Para transformar las salidas a valores dentro del intervalo de magnitud deseado, entonces se aplica la transformación inversa, o sea, escalar el valor normalizado. Para la Ecuación 2.24 la transformación inversa que se aplica a las salidas de la red es:

$$a_k^s = \frac{(a_k^{s*} + 1)(t_{max} - t_{min})}{2} + t_{min} \quad \text{Ecuación 2.25}$$

donde:

$a_k^{s*}$ : Valor de salida de la red

$a_k^s$ : Valor escalado

$t_{max}$ : Valor máximo de la salida deseada  $t_k$  para todos los patrones de entrenamiento  $r$

$t_{min}$ : Valor mínimo de la salida deseada  $t_k$  para todos los patrones de entrenamiento  $r$

## 2.4 Ejemplo de una RNA

En esta sección se presentará un ejemplo de una RNA entrenada para ajustarse a la función trigonométrica coseno (Ecuación 2.5). Para entrenar la red se utilizó la herramienta de Matlab [27] (nntool), la cual utiliza el algoritmo de entrenamiento Levenberg-Marquardt. Dicho algoritmo tiene la misma estructura que el “backpropagation” pero con pequeñas variantes que mejoran la velocidad de convergencia. Para conocer más acerca de este algoritmo se puede referir al texto de la Universidad de Pereira [4].

### 2.4.1 Datos

La red elegida para este ejemplo sigue el modelo de MLP, con una capa oculta de 4 neuronas y función de transferencia tansig. La Figura 2.14 muestra el esquema de la red propuesta.

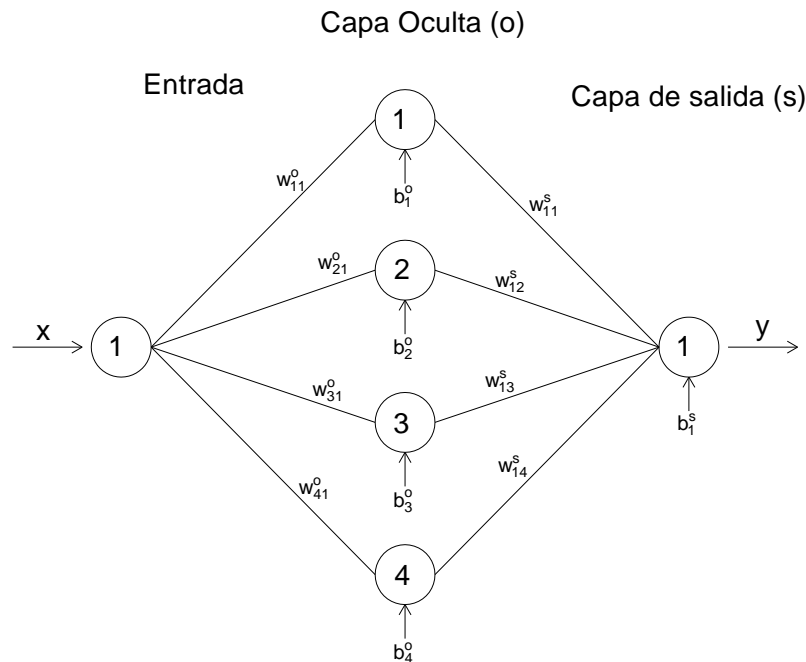


Figura 2.14: Modelo de red propuesto

Debido a que la función coseno es una función escalar de una variable, la red solo tiene una neurona en la capa de entrada y una neurona en la capa de salida. La forma de la función coseno se muestra en la Ecuación 2.26 y la gráfica de la función en la Figura 2.15.

$$y = \cos x$$

Ecuación 2.26

donde:

$x$ : Ángulo expresado en radianes.

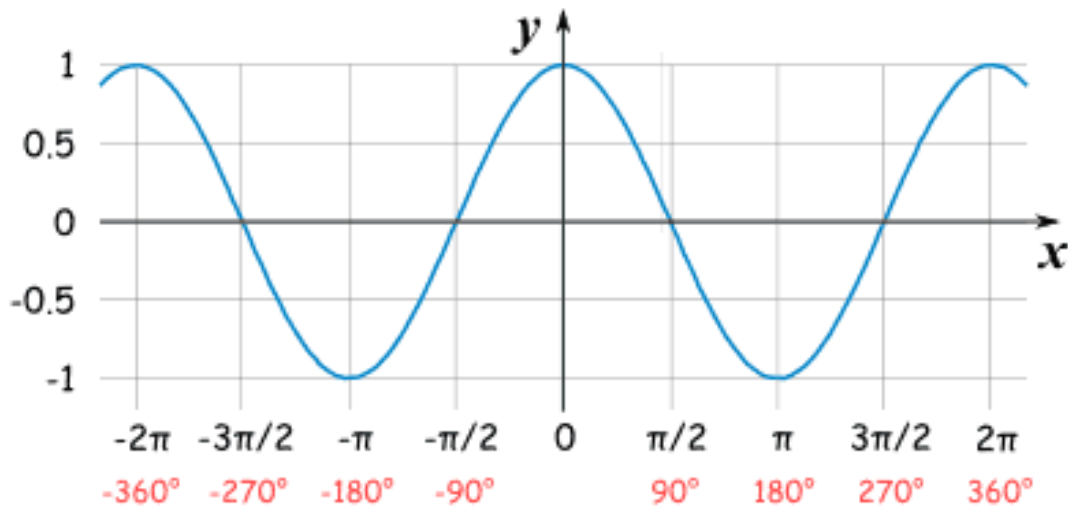


Figura 2.15: Gráfica de la función coseno

Para entrenar la red se han elegido 21 patrones de entrenamiento, repartidos uniformemente entre  $-2\pi$  y  $2\pi$ . De tal forma que la muestra englobe el espacio de entrenamiento, y la red pueda generalizar la función. En la Tabla 2.3 se muestran los patrones de entrenamiento elegidos.

Tabla 2.3: Patrones de entrenamiento

r	1	2	3	4	5	6	7	8	9	10
P	-6.283	-5.655	-5.027	-4.398	-3.770	-3.142	-2.513	-1.885	-1.257	-0.628
T	1.000	0.809	0.309	-0.309	-0.809	-1.000	-0.809	-0.309	0.309	0.809

11	12	13	14	15	16	17	18	19	20	21
0.000	0.628	1.257	1.885	2.513	3.142	3.770	4.398	5.027	5.655	6.283
1.000	0.809	0.309	-0.309	-0.809	-1.000	-0.809	-0.309	0.309	0.809	1.000

La elección de los patrones de entrenamiento puede ser de cualquier forma siempre y cuando existan suficientes patrones para representar el problema y éstos tengan una distribución más o menos uniforme. En el caso de ser un problema discreto se puede acotar el problema y seleccionar un porcentaje significativo de patrones del problema. Para un problema continuo, como es éste caso, se divide el problema en partes iguales hasta obtener suficientes patrones de entrenamiento. Entre mayor es el número de patrones de entrenamiento mayor va a ser la precisión del resultado.

La manera en la que Matlab [27] realiza el entrenamiento es de tipo "Batch" o en bloque, es decir que utiliza la Ecuación 2.22 para calcular el error medio cuadrático de todos los patrones de entrenamiento. Así, actualiza los pesos época a época tomando en cuenta la contribución del error de todos y cada uno de los patrones de entrenamiento.



Para determinar rendimiento de la red una vez entrenada, Matlab asigna un porcentaje de los patrones de entrenamiento para validación y comprobación (Validation, Testing). Esto quiere decir que los patrones asignados para validación y comprobación no se utilizan en el entrenamiento.

Los patrones de validación los utiliza para determinar el número de iteraciones o épocas que realizará. De esta manera Matlab puede detener el algoritmo cuando se llega a un punto óptimo, es decir, cuando el error de entrenamiento y el de validación es muy parecido. [5]

Los patrones de comprobación permiten al usuario revisar el rendimiento que la red tiene para patrones jamás presentados a la misma. Debido a que al entrenar la red se inician los pesos y los umbrales de manera aleatoria, cada vez que se entrena la red el rendimiento cambia. Así, la comprobación juega un papel importante para que el usuario pueda decidir si el entrenamiento fue satisfactorio o comenzará un nuevo entrenamiento que entregue mejores resultados.

En este ejemplo se asignaron 17 patrones de entrenamiento, 3 de validación, y 3 de comprobación. Lo que equivale a un porcentaje del 70%, 15%, 15% respectivamente. La manera en la que Matlab elige que patrones utilizar para cada etapa; entrenamiento, validación y comprobación, es aleatoria. Con lo anterior se tiene todo lo necesario para comenzar a entrenar la red.

#### **2.4.2 Entrenamiento, validación y comprobación**

Una vez ingresados los datos al programa de computadora, se realizó el entrenamiento. El entrenamiento se realizó en 36 épocas, es decir se le presentaron 36 veces a la red los 17 patrones de entrenamiento y se actualizaron los pesos y los umbrales 36 veces. Para cada época se calculó el error medio cuadrático (MSE) con la Ecuación 2.22. La Figura 2.16 muestra el comportamiento de la red respecto al MSE calculado en cada época.

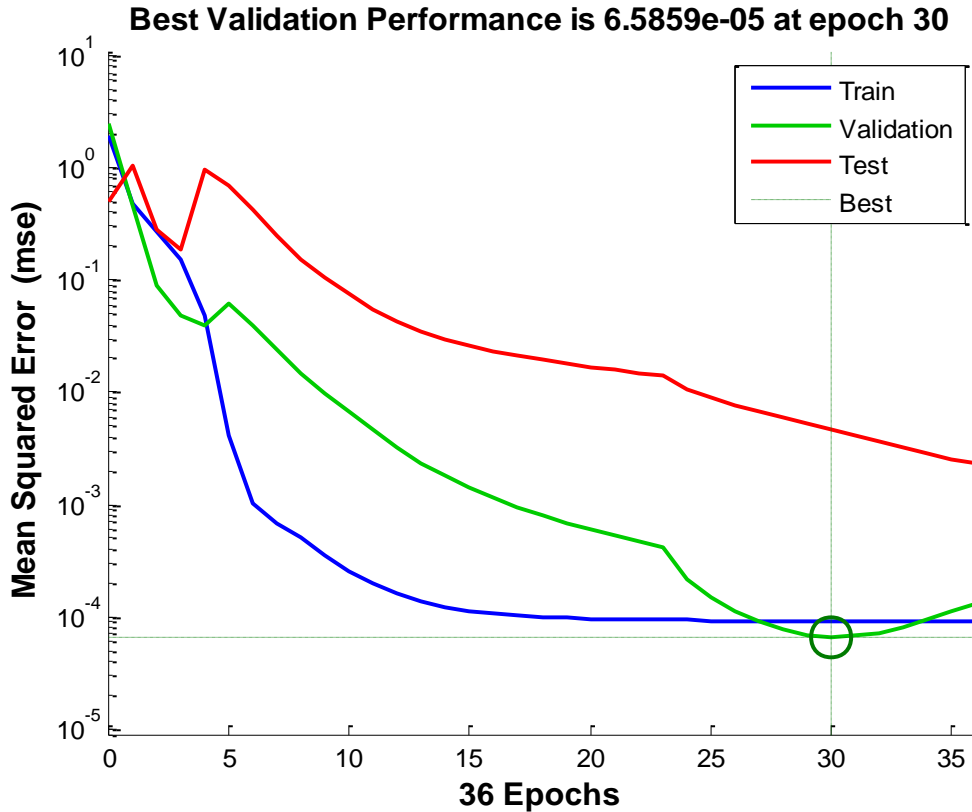


Figura 2.16: Comportamiento durante el entrenamiento

El valor de los pesos y los umbrales que se tomaron fueron los obtenidos en la actualización número 30, debido a que es en esta época fue en la que mejor rendimiento de validación se obtuvo, con un error de validación de  $6.58 \times 10^{-6}$ . El algoritmo se detuvo en la época 35 debido a que el error de validación no disminuyó en las seis épocas posteriores a la época 30.

A continuación se muestran las matrices de pesos y de umbrales de cada capa obtenidas en la actualización de la época 30.

Capa oculta:

$$W^o = \begin{bmatrix} 5.1306 \\ 4.3062 \\ -4.0410 \\ 4.1601 \end{bmatrix} \quad \text{Ecuación 2.27}$$

$$B^o = \begin{bmatrix} -3.7238 \\ -1.0377 \\ -1.0456 \\ 3.0182 \end{bmatrix} \quad \text{Ecuación 2.28}$$

Capa de salida:

$$W^s = [1.3522 \quad -1.6425 \quad -1.8110 \quad -1.6471] \quad \text{Ecuación 2.29}$$

$$B^s = 1.2872 \quad \text{Ecuación 2.30}$$

En la Figura 2.17 se puede observar el ajuste que se obtuvo para los parámetros de entrenamiento, validación y comprobación. En ningún caso el error calculado  $\delta_k$  excede de  $\pm 0.15$ , de hecho la mayoría de los puntos se encuentran en un rango de error  $\pm 0.02$ . En la misma figura se puede observar que el ajuste es casi perfecto y la curva obtenida es muy parecida a la del coseno de la Figura 2.15.

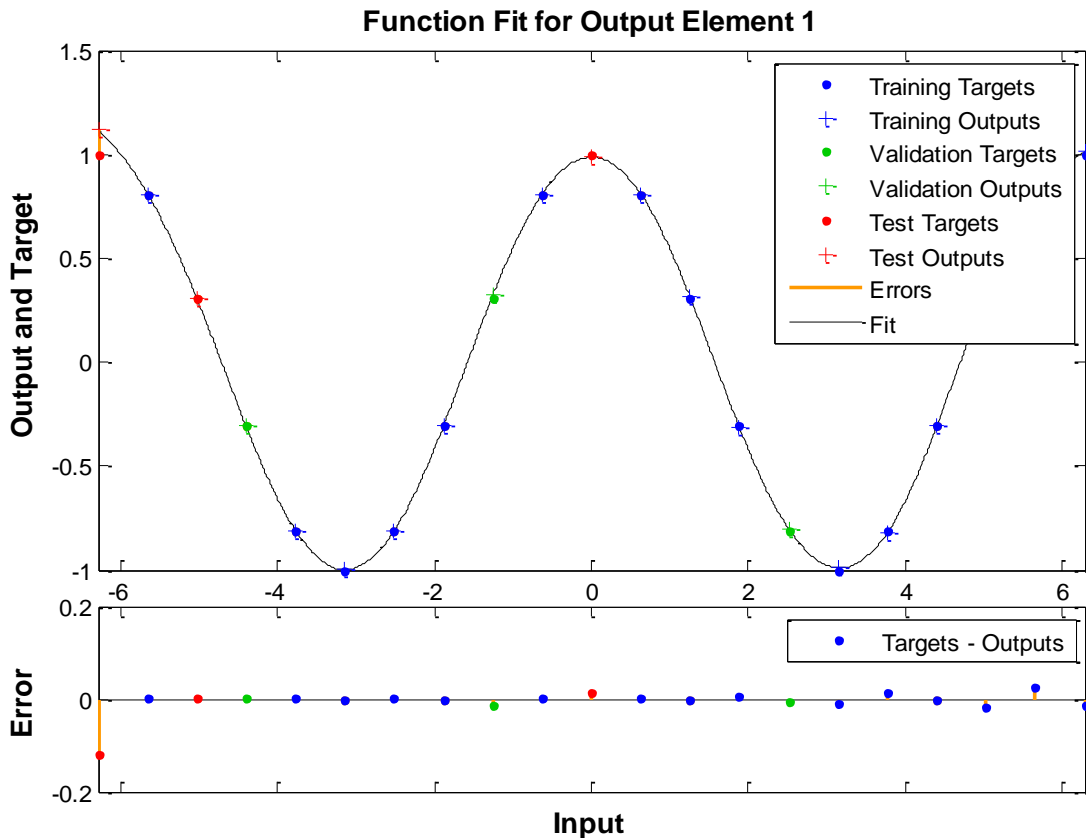


Figura 2.17: Ajuste de la RNA a la función coseno

### 2.4.3 Operación

Para operar la red obtenida con Matlab, solo hace falta utilizar el comando  $y = sim(net, x)$ , en donde  $y$  es la salida,  $net$  es la red creada y  $x$  es parámetro de entrada a la red. En este ejemplo trabajaremos con las matrices de pesos y los umbrales para ilustrar la operación de la red. Para operar la red con un valor de entrada  $x$  igual a  $60^\circ$  se tiene:

$$x = 60^\circ = \frac{1}{3}\pi \quad \text{Ecuación 2.31}$$

Normalizando el valor de entrada con la Ecuación 2.24 obtenemos:

$$x^* = 2 \left[ \frac{\frac{1}{3}\pi + 2\pi}{2\pi + 2\pi} \right] - 1 = \frac{1}{6} \quad \text{Ecuación 2.32}$$

Para calcular el vector de propagación de la primera capa se aplica la Ecuación 2.3 de forma matricial. La forma matricial permite propagar la señal automáticamente en todas las neuronas de una capa. Hay que recordar que cada renglón de las matrices representa una neurona. Sustituyendo las Ecuaciones 2.27, 2.28 y 2.32 Ecuación 2.28:

$$N^o = W^o X^* + B^o = \begin{bmatrix} 5.1306 \\ 4.3062 \\ -4.0410 \\ 4.1601 \end{bmatrix} \frac{1}{6} + \begin{bmatrix} -3.7238 \\ -1.0377 \\ -1.0456 \\ 3.0182 \end{bmatrix} = \begin{bmatrix} -2.8687 \\ -0.3200 \\ -1.7193 \\ 3.7115 \end{bmatrix} \quad \text{Ecuación 2.33}$$

Aplicando la Ecuación 2.4 obtenemos la salida de la capa oculta. La función de transferencia de la capa oculta es la función *tansig*.

$$A^o = \tanh(N^o) = \begin{bmatrix} -0.9936 \\ -0.3095 \\ -0.9378 \\ 0.9988 \end{bmatrix} \quad \text{Ecuación 2.34}$$

Al igual que en la capa oculta, la capa de salida requiere del cálculo del vector de propagación el cual está dado por:

$$N^s = W^s A^o + B^s = [1.3522 \quad -1.6425 \quad -1.8110 \quad -1.6471] \begin{bmatrix} -0.9936 \\ -0.3095 \\ -0.9378 \\ 0.9988 \end{bmatrix} + 1.2872 = 0.5053$$

Ecuación 2.35

La función de transferencia de la capa de salida es la función *identidad* lo que quiere decir que la salida de la red es:

$$A^{s*} = N^s = 0.5053 \quad \text{Ecuación 2.36}$$

El superíndice \*, indica que la salida de la red está normalizada. Para transformar esta salida a un valor escalado, se aplica la Ecuación 2.25; sin embargo, como se puede ver en la Tabla 2.3,  $t_{min}$  y  $t_{max}$  corresponden a -1 y 1 respectivamente. Lo que lleva al resultado final:

$$A^{s*} = A^s = 0.5053 \qquad \text{Ecuación 2.37}$$

Como se observa, se obtuvo un resultado aceptable ya que sustituyendo las ecuaciones Ecuación 2.31 en la Ecuación 2.26 se obtiene:

$$\text{Cos}\left(\frac{1}{3}\pi\right) = 0.5 \qquad \text{Ecuación 2.38}$$

### **3 Objetivo**

#### **3.1 Objetivo primario**

Determinar una manera en la que se puede aplicar la teoría de las Redes Neuronales Artificiales (RNA) al diseño sísmico de edificios.

Generar una RNA que tenga la capacidad de realizar el diseño sismorresistente de edificios de concreto reforzado. Dicho diseño debe de cumplir con los lineamientos de Reglamento de Construcciones para el Distrito Federal (RCDF-2004) y sobretodo con las características de marcos dúctiles especificadas en ese reglamento.

Realizar un ejemplo comparativo de un edificio diseñado con métodos convencionales de diseño y, alternativamente, con la RNA generada.

#### **3.2 Objetivo secundario**

Este trabajo tiene como objetivo secundario demostrar la utilidad de las RNA para el diseño de edificios sismorresistentes. En el trabajo se intenta encontrar una herramienta la cual facilite el diseño de edificios pudiéndolo hacer en menor tiempo con respecto al de un diseño convencional. El fin de la RNA es comprobar la versatilidad e utilidad de esta herramienta en el diseño sísmico de edificios.

Este trabajo pretende acercar a los Ingenieros Civiles a la teoría de las RNA, dándoles una breve introducción a los conceptos básicos y las suficientes herramientas para poder generar, entrenar y operar una RNA.

Pretende que se genere interés en los Ingenieros Civiles, y se desarrolle investigación y tecnología para que en un futuro se utilicen las RNA como una herramienta de diseño.

## 4 Metodología y su Aplicación

En este capítulo se detalla el proceso que se siguió para generar y entrenar la RNA. Recordando el capítulo 2, para poder iniciar el proceso de generación de la RNA, hay que comenzar por definir las variables del problema, después se debe de elegir una arquitectura y un algoritmo de entrenamiento capaces de resolver el problema. Al final se debe de entrenar la red para obtener la matriz de pesos y los umbrales de cada neurona. En este caso en particular, para el entrenamiento, se tuvo que realizar una base de datos que contiene las entradas y las salidas con las que se entrenó la red. Dichas entradas y salidas, corresponden a edificios diseñados siguiendo los parámetros que se explican en el siguiente subcapítulo 4.1. A los edificios diseñados se les llamó modelos de entrenamiento.

### 4.1 Definición de la estructura general de la red

A muy grandes rasgos, el diseño consiste en dar un resultado, a partir de un conjunto de datos. Por ejemplo, para una viga, dado un momento flexionante se debe dimensionar la viga y se debe definir la cantidad de acero para que resista el momento. De igual manera, para un edificio del cual solo se conoce su configuración arquitectónica o geometría, se deben de dimensionar todos los elementos estructurales para cumplir con los límites de resistencia y servicio. Con esta idea se determina la estructura de la RNA para diseño sísmico, la cual va a necesitar de datos de entrada que consisten en la geometría del edificio y de los datos de salida que son las dimensiones de los elementos estructurales, así como la cantidad de acero necesaria.

#### 4.1.1 Entradas

En el estudio de las RNA se observa que el entrenamiento supervisado es el más adecuado para el ajuste de funciones, o sea, dado un conjunto de entradas y salidas, se entrena una red que tiene un comportamiento deseado. En nuestro problema de diseño, existen una gran cantidad de posibles variables de entrada con las que se puede generar la red. Dentro de las más obvias se tiene la siguiente lista:

- Número de niveles
- Altura de entresijos
- Número de crujeas
- Características de los materiales
- Carga Viva
- Carga Muerta
- Zona Sísmica

Para poder demostrar el uso de las RNA se fijó como constantes algunos de los valores antes mencionados. Esto implica que la RNA solo trabajará dentro del rango de estos valores constantes, es decir, que los parámetros del entrenamiento y operación deberán de elegirse dentro del rango especificado. De lo contrario se correrá el riesgo de obtener resultados incorrectos. La Tabla 4.1 muestra las variables constantes del problema y sus valores.

Tabla 4.1: Variables constantes

Variables	Valor constante para la RNA
Material de construcción	Concreto reforzado
Resistencia del concreto	$f'c= 250 \text{ kg/cm}^2$
Fluencia del acero	$f_y= 4200 \text{ kg/cm}^2$
Zona Sísmica	IIIb de la Ciudad de México
Factor de respuesta sísmica	$Q=3$
Tipo de construcción	Oficinas

Una vez que se fijaron todas las variables de diseño presentadas, también se fijó una geometría general con la cual se realizarán los diseños de entrenamiento. Esta geometría limita la planta de los edificios a ser cuadrada o rectangular, y a que las columnas estén alineadas en ejes perpendiculares equidistantes. La ventaja de utilizar esta geometría es que cumple en parte con los requisitos del RCDF-2004 [13] de un edificio regular, y permite utilizar un factor  $Q = 3$ . Otra de las características de las plantas utilizadas es que no cuenta con vigas secundarias, las columnas centrales siempre tienen una sección rectangular y las columnas de esquina tienen una relación  $l_x/l_y$  que es igual a la relación  $L_x/L_y$  de toda la planta. En la Figura 4.1 se muestra una planta ejemplo de tres crujiás en la dirección X, y dos crujiás en la dirección Y.

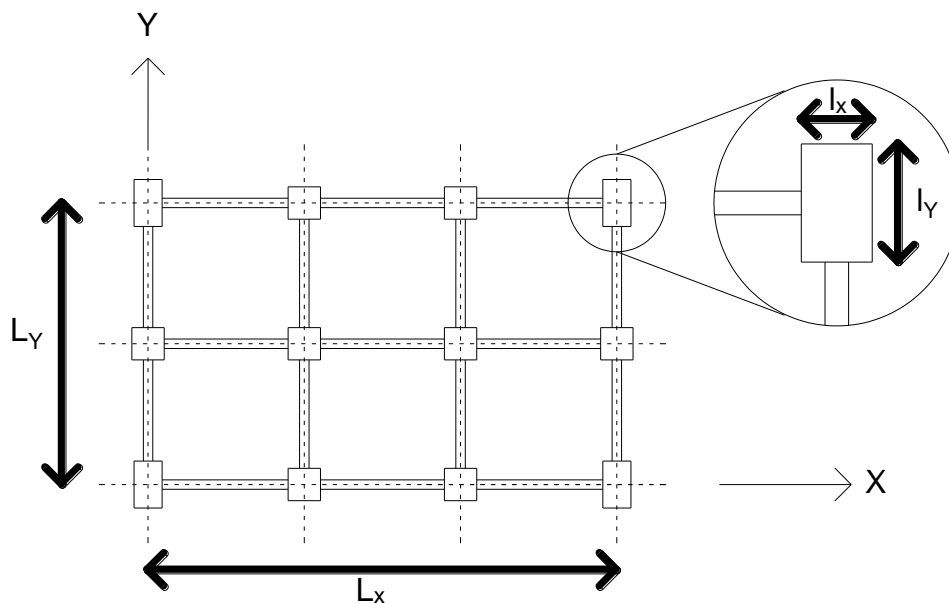


Figura 4.1: Planta ejemplo 3x2 crujiás



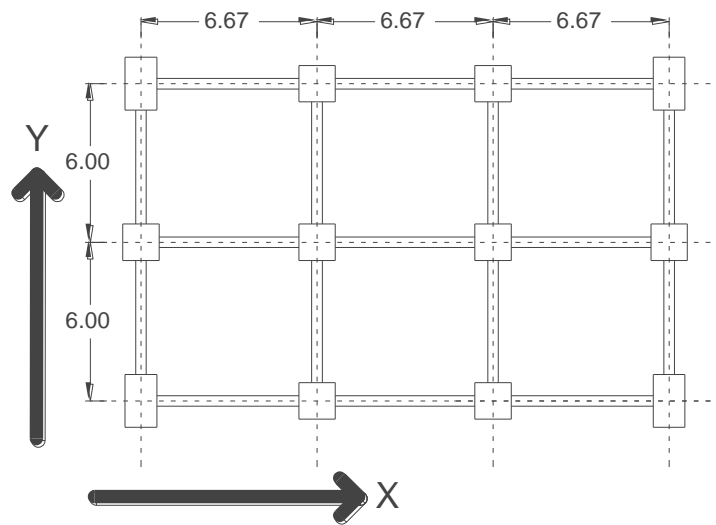
Los datos de la geometría presentada son los valores de entrada de los modelos de entrenamiento. Dichos datos se pueden presentar en un vector de tal forma que cada vector representa un edificio. La forma del vector de entrada se muestra en la Tabla 4.2.

**Tabla 4.2: Vector de entrada**

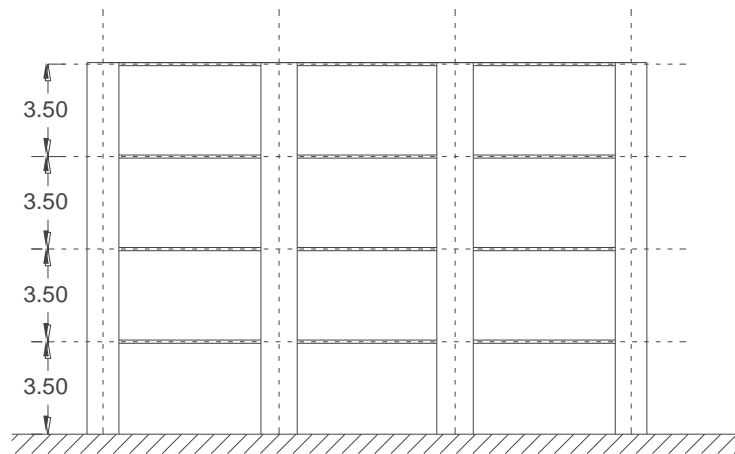
<b>Forma del vector de entrada:</b>	
<b>Datos de diseño</b>	<b>Rangos del vector</b>
Número de crujías en la dirección <i>X</i>	De 3 hasta 5
Número de crujías en la dirección <i>Y</i>	De 3 hasta 5
Número de niveles	De 4 hasta 10
Claros en la dirección <i>X</i>	De 5 metros hasta 10 metros
Claros en la dirección <i>Y</i>	De 5 metros hasta 10 metros
Altura de entrepiso	De 3 metros hasta 4 metros

Como se puede observar en la Tabla 4.2 los datos de entrada tienen un rango dentro del cual podrá operar la red. El rango se eligió de esta manera ya que comprende un espacio de edificios medianos y bajos de geometría regular. En la Ecuación 4.1 se muestra un vector de entrada y en la Figura 4.2 un esquema de un edificio representado por el vector.

$$\text{Vector de entrada} = \begin{bmatrix} 3 \\ 2 \\ 4 \\ 6.67 \\ 6 \\ 3.5 \end{bmatrix} \qquad \text{Ecuación 4.1}$$



PLANTA DEL EDIFICIO



ELEVACIÓN DIRECCIÓN X

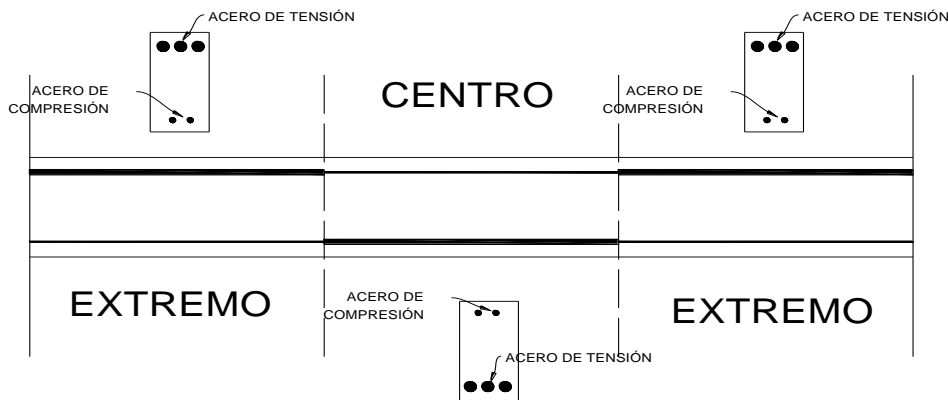
Figura 4.2: Edificio representado por la Ecuación 4.1

#### 4.1.1. Salidas

Los valores de salida son las dimensiones y la cuantía de acero que cada elemento estructural tiene. Los elementos estructurales que se dimensionaron son vigas y columnas. Para cada uno de estos existen algunos parámetros que los definen, estos parámetros se encuentran en la Tabla 4.3 para vigas y en la Tabla 4.4 para columnas.

Tabla 4.3: Parámetros de las vigas

VIGAS	
Parámetro	Unidades
$b$ : Base	Centímetros
$h$ : Peralte	Centímetros
$\rho_{e_c}$ : Cuantía de acero de compresión (extremos)	Adimensional
$\rho_{e_t}$ : Cuantía de acero de tensión (extremos)	Adimensional
$\rho_{c_c}$ : Cuantía de acero de compresión (centro)	Adimensional
$\rho_{c_t}$ : Cuantía de acero de tensión (centro)	Adimensional

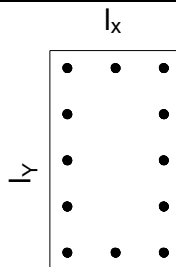


## VIGAS

Figura 4.3: Secciones de una viga

Tabla 4.4: Parámetros de las columnas

COLUMNAS	
Parámetro	Unidades
$l_x$ : Lado dirección $X$	Centímetros
$l_y$ : Lado dirección $Y$	Centímetros
$\rho$ : Cuantía de acero	Adimensional



## COLUMNAS

Figura 4.4: Sección de una columna

Los elementos estructurales de un edificio cambian dependiendo de su ubicación. Por ejemplo una viga en la dirección  $X$  puede cambiar con respecto a una viga en la dirección  $Y$ . En el problema, se define que para cada edificio se tienen 8 elementos estructurales diferentes (4 vigas, 4 columnas) representados por 36 valores:

- $Vp_x$ : Vigas perimetrales en dirección  $X$
- $Vp_y$ : Vigas perimetrales en dirección  $Y$
- $Vc_x$ : Vigas centrales en dirección  $X$
- $Vc_y$ : Vigas centrales en dirección  $Y$
- $Ce_s$ : Columnas superiores de esquina
- $Ce_i$ : Columnas inferiores de esquina
- $Cc_s$ : Columnas superiores centrales
- $Cc_i$ : Columnas inferiores centrales

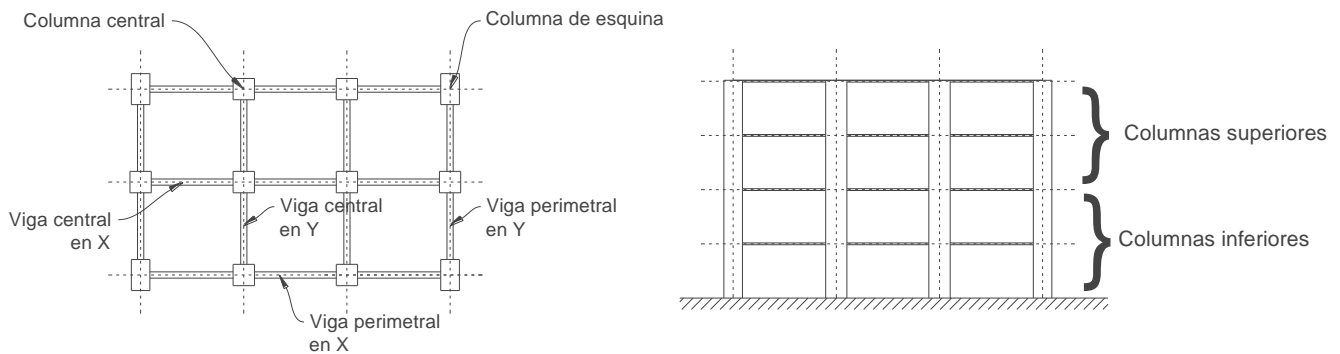


Figura 4.5: Ubicación de los elementos de un edificio

En los edificios las vigas de todas las plantas se repitieron, pero las columnas cambiaron conforme al nivel de la planta. La frontera entre las columnas superiores e inferiores se hizo de tal forma que los edificios con un número par de niveles tienen la misma cantidad de plantas con columnas superiores que inferiores, en cambio, los edificios con un número impar de niveles tienen una planta menos de columnas superiores.

Con todo lo anterior se arma un vector de salida para un edificio, este tendría 36 valores y se ve de la siguiente manera:

$$\text{Vector de salida} = \begin{bmatrix} Vp_x \\ Vp_y \\ Vc_x \\ Vc_y \\ Ce_s \\ Ce_i \\ Cc_s \\ Cc_i \end{bmatrix} \quad \text{Ecuación 4.2}$$

En donde:

$$Vigas = \begin{bmatrix} b \\ h \\ \rho e_t \\ \rho e_c \\ \rho c_t \\ \rho c_c \end{bmatrix} \qquad Columnas = \begin{bmatrix} l_x \\ l_y \\ \rho \end{bmatrix}$$

#### 4.1.2 Arquitectura de la RNA para diseño sísmico

En el capítulo 2 se explicó que para resolver este problema se utiliza el modelo del perceptrón multicapa (MLP) ya que es el más común para aproximar funciones. Utiliza una arquitectura “*feedforward*” con una o más capas ocultas con función sigmoidea, y la capa de salida con función identidad. También utiliza el algoritmo de entrenamiento “*backpropagation*” o alguna de sus variantes. En este caso contiene una capa oculta, la cual tendrá entre 20 y 30 neuronas. Para determinar el número exacto de neuronas en la capa oculta se entrenarán varias redes, y se utilizará la que mejor rendimiento tenga (ver sub-capítulo 4.3.1).

Por lo tanto, la arquitectura de la red será como la que se muestra en la Figura 4.6. Tendrá 6 entradas, de 20 a 30 neuronas en la capa oculta (este intervalo se explica en la sección 4.3) y 36 neuronas en la capa de salida. Las entradas y las salidas de la RNA tienen unidades, y aunque estas unidades no son relevantes para el procesamiento de información por la red, sí lo son para el entrenamiento y la operación.

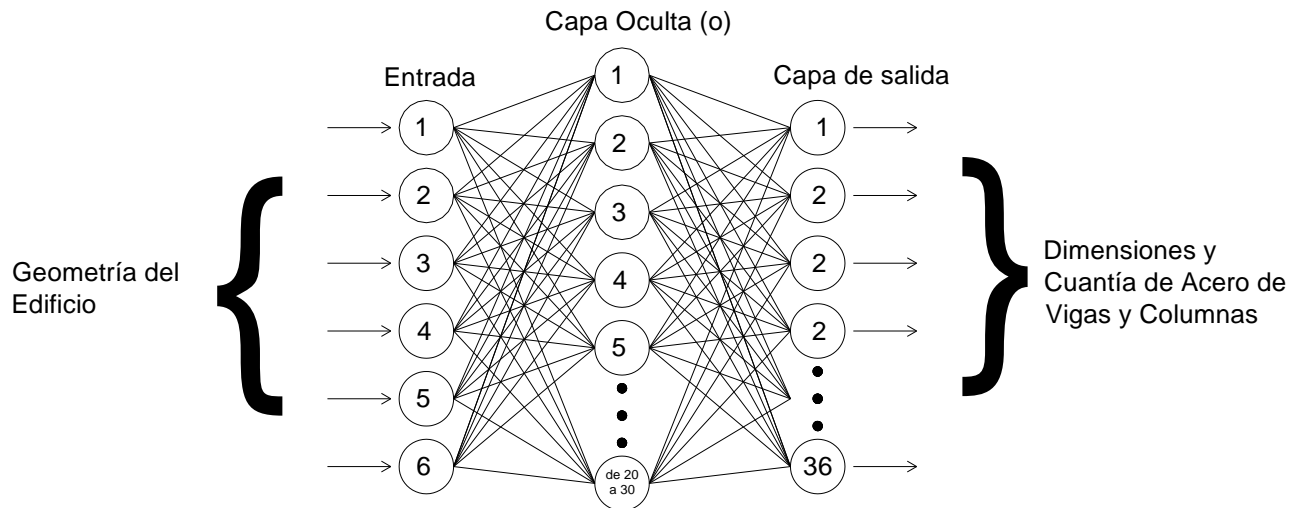


Figura 4.6: Arquitectura de la RNA para diseño sísmico

Conociendo la arquitectura de la RNA se puede también conocer las variables que se necesitan en el entrenamiento. Estas variables son los vectores y matrices de umbrales, y pesos de cada capa.

$W^o$ : Vector de pesos de la capa oculta  
 $B^o$ : Vector de umbrales de la capa oculta  
 $W^s$ : Vector de pesos de la capa de salida  
 $B^s$ : Vector de umbrales de la capa de salida

El tamaño de estas variables depende del número de neuronas en la capa oculta, de tal forma que el tamaño de las matrices de pesos tienen el mismo número de renglones que las neuronas de la capa y el mismo número de columnas que las neuronas de la capa inmediata anterior. El tamaño del vector de umbrales es simplemente el tamaño de la capa. La Tabla 4.5 muestra el rango de tamaño en el que fluctuarán estos vectores.

**Tabla 4.5: Tamaño de las matrices de pesos y umbrales**

Variable	Tamaño	Tipo de Variable
$W^o$	$[CO, 6]$	Matriz
$B^o$	$[CO, 1]$	Vector columna
$W^s$	$[36, CO]$	Matriz
$B^s$	$[36, 1]$	Vector columna

donde:

$CO$ : es el número de neuronas en la capa oculta (entre 20 y 30).

## 4.2 Generación de los modelos de entrenamiento

En el entrenamiento de tipo supervisado es necesario tener patrones de entrenamiento (entrada y salida) que abarquen un espacio global de tal forma que el problema pueda generalizarse. En esta sección se explica cómo se obtuvieron los patrones de entrenamiento y cómo se determinó el espacio que abarcan. Debido a que el número de modelos de entrenamiento que se obtuvieron es de 131 (ver sub-capítulo 4.2.4), no es posible presentar en el presente trabajo toda la base de datos, por lo que solo se presentará una porción de esta.

Para poder generar la base de datos con diseños de edificios se tuvo que determinar de qué manera se obtendrían los diseños. Se decidió que se utilizaría el programa ETABS [28] para realizar el análisis estructural y el análisis sísmico. Después los resultados de ETABS se importarían a un código realizado en Matlab (ver Anexo A), el cual diseñaría las vigas y las columnas siguiendo los lineamientos del RCDF-2004 [12, 13, 14 y 15]. De esta manera se generó una base de datos de 133 edificios en un tiempo relativamente corto. Cada edificio se representó con 6 parámetros de entrada y 36 parámetros de salida como se explica en el capítulo 4.1.

En todo este proceso se tomaron algunas consideraciones de diseño, necesarias para poder utilizar el programa ETABS y para poder generar el código de Matlab. A continuación se muestran estas

consideraciones referentes al análisis de cargas y combinaciones de cargas, aspecto sísmico, de concreto reforzado, y por último, se presenta el diagrama de bloques para la generación de modelos de entrenamiento (Ver anexo A).

#### 4.2.1 Análisis de cargas y combinaciones de cargas

Para realizar el análisis estructural de los modelos de entrenamiento fue necesario establecer las cargas con las que trabajarían los modelos. Todos los modelos generados se simularon con las mismas cargas para tener un patrón de comportamiento estable.

Nótese en la Tabla 4.1 que la red solo podrá dar resultados para edificios destinados a oficinas. Utilizando el RCDF-2004 [12] se realizó un análisis de cargas para este tipo de construcción y se llegó a los resultados que se muestran en la Tabla 4.6, los cuales se utilizaron para el diseño de los modelos de entrenamiento.

Tabla 4.6: Análisis de cargas para oficinas

Carga	Valor
Viva máxima (niveles intermedios)	250 kg/m <sup>2</sup>
Viva máxima (azotea)	100 kg/m <sup>2</sup>
Viva instantánea (niveles intermedios)	180 kg/m <sup>2</sup>
Viva instantánea (azotea)	70 kg/m <sup>2</sup>
Muerta de elementos no estructurales*	230 kg/m <sup>2</sup>
Muerta de elementos estructurales**	Peso del concreto reforzado

\* Los elementos estructurales toman en cuenta los muros, las instalaciones, los acabados, y todos aquellos elementos que no forman parte del marco estructural.

\*\* El programa ETABS realiza el cálculo del peso de los elementos estructurales, columnas, vigas y losas, como la multiplicación del peso volumétrico del concreto por el volumen.

Las combinaciones y factores de carga que se establecieron también corresponden a los estipulados en las NTC-CA [14]. En estas normas se determinan las combinaciones de carga de acuerdo a la naturaleza de las acciones:

Combinación con carga gravitacional:

$$1.4 CM + 1.4 CV_m \quad \text{Ecuación 4.3}$$

Combinación con cargas accidentales:

$$1.1 CM + 1.1 CV_a + 1.1 S \quad \text{Ecuación 4.4}$$

Combinación de servicio:

$$CM + CV_a + S$$

Ecuación 4.5

donde:

$CM$ : Carga Muerta (Estructura + acabados e instalaciones)

$CV_m$ : Carga Viva máxima

$CV_a$ : Carga Viva instantánea (reducida)

$S$ : Acciones sísmicas

#### 4.2.2 Aspecto sísmico

Para poder analizar los modelos de entrenamiento sometidos a las acciones por sismo se realizó un **análisis modal espectral**. Es decir, un análisis dinámico de cada modelo para determinar el periodo de los primeros modos de vibración, y a partir de un espectro de diseño, se determinan las aceleraciones y las fuerzas que actúan sobre la estructura. Para la superposición de acciones (cortante y desplazamientos) se utilizó el criterio de Rosenblueth como lo establece el RCDF-2004 [13]:

$$R_R = \left( \sum_{i=1}^N R_i \right)^{1/2}$$

Ecuación 4.6

donde:

$R_R$ : Respuesta total del sistema (desplazamiento, fuerza cortante, etc.)

$R_i$ : Respuesta para el modo de vibración  $i$ .

$N$ : Número total de modos de vibración calculados, 10 en este trabajo.

##### 4.2.2.1 Espectro de diseño

Como se puede observar, en la Tabla 4.1 se estableció que el edificio se ubicaría en la zona sísmica IIIb lo que implica que solo se trabajará con el espectro de diseño, correspondiente a este sitio. El Apéndice A de las NTC-DS [13] permite calcular espectros de diseño respecto al periodo dominante del suelo  $T_S$ . Para la zona en cuestión, IIIb, el periodo dominante del suelo es de 2 segundos. Ignorando la interacción suelo-estructura, el espectro de diseño calculado quedó como se indica en la Figura 4.7.



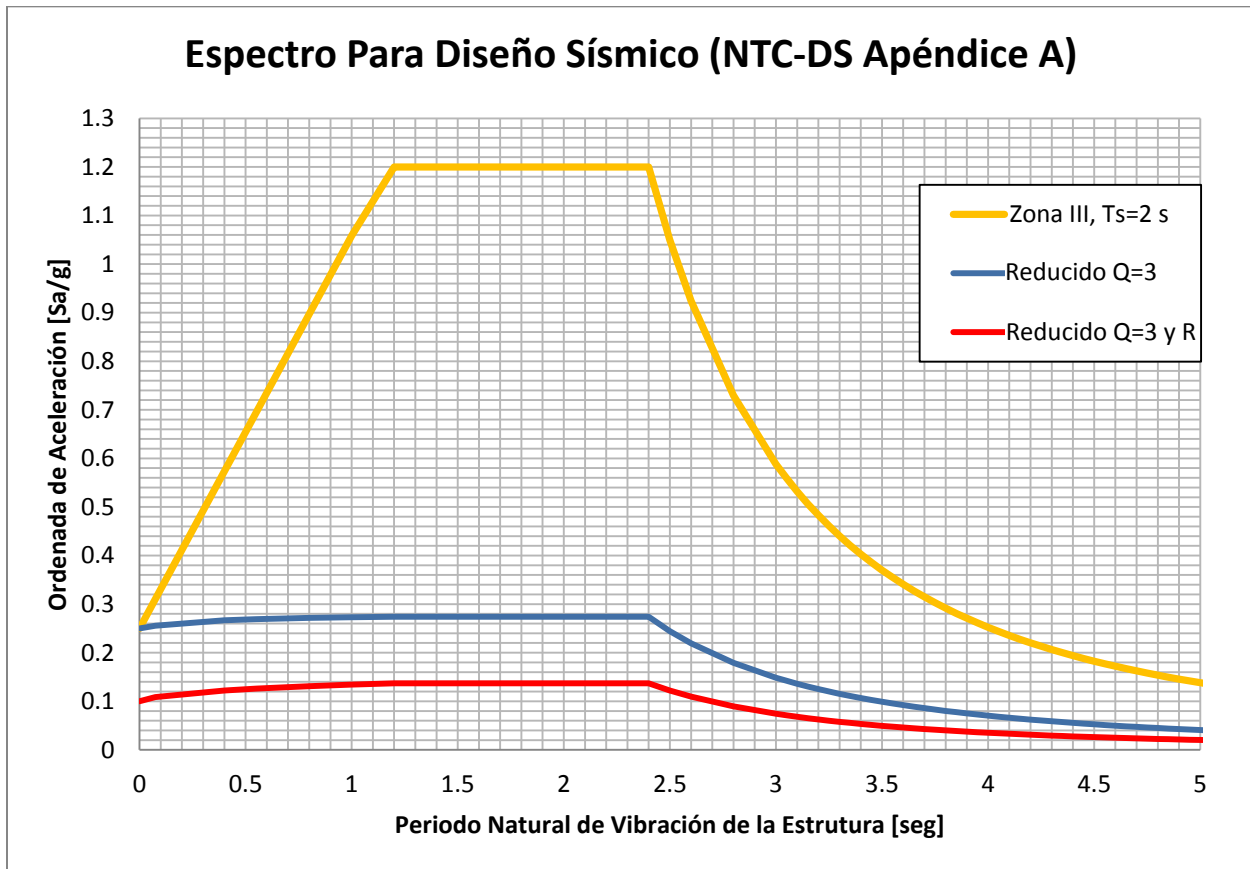


Figura 4.7: Espectro para diseño de los modelos de entrenamiento

En la Figura 4.7 se encuentra el espectro de diseño no reducido (amarillo), también se muestra la gráfica del espectro de diseño reducido por ductilidad (azul) y reducido además, por sobre-resistencia (rojo). Es este último el espectro con el que se analizaron los modelos en ETABS. Como se puede observar la ordenada máxima que acepta el espectro reducido (rojo) es 0.13.

#### 4.2.2.2 Efectos bidireccionales

Para el análisis sísmico en dos direcciones perpendiculares  $X$  y  $Y$ , se tomaron las combinaciones establecidas por el RCDF-2004. Se establece que se tome el 100% de las fuerzas sísmicas en la dirección principal, y el 30% de la fuerza sísmica en la dirección perpendicular. Se tomaron en cuenta cuatro direcciones principales,  $(+X)$ ,  $(-X)$ ,  $(+Y)$  y  $(-Y)$ , de esta manera las combinaciones accidentales definidas en la Ecuación 4.4 quedan de la siguiente manera:

Tabla 4.7: Combinaciones de carga

	$1.1 CM + 1.1 CV_a + 1.1 S_x + 0.33 S_y$
<i>Dirección X</i>	$1.1 CM + 1.1 CV_a + 1.1 S_x - 0.33 S_y$
	$1.1 CM + 1.1 CV_a - 1.1 S_x + 0.33 S_y$
	$1.1 CM + 1.1 CV_a - 1.1 S_x - 0.33 S_y$
	$1.1 CM + 1.1 CV_a + 0.33 S_x + 1.1 S_y$
<i>Dirección Y</i>	$1.1 CM + 1.1 CV_a + 0.33 S_x - 1.1 S_y$
	$1.1 CM + 1.1 CV_a - 0.33 S_x + 1.1 S_y$
	$1.1 CM + 1.1 CV_a - 0.33 S_x - 1.1 S_y$

#### 4.2.2.3 Marco Dúctil

Otra de las consideraciones que se tomó en cuenta para el entrenamiento y la operación fue el uso de un factor de respuesta sísmico  $Q = 3$ . Esto implica que todos los diseños realizados para entrenar la red deben de cumplir con las consideraciones de marcos dúctiles especificadas en las NTC para el diseño de estructuras de concreto (NTC-DEC) [15]. Además los diseños que se generaron con la red, también deberán cumplir con las características de marcos dúctiles. Las consideraciones que se tomaron en cuenta en este aspecto fueron las siguientes:

##### Vigas:

- La relación entre el peralte y el ancho no debe exceder de 3.0
- El ancho de la viga no será menor de 25cm
- En toda la sección se dispondrá de acero tanto en el lecho superior como en el inferior y constará cuando menos de dos barras corridas del número 4
- El momento resistente positivo en el paño de la unión viga-columna no será menor que la mitad del momento resistente negativo; ni el resistente positivo, será menor que la cuarta parte del máximo momento resistente que tenga en los extremos.

Columnas:

- La dimensión transversal mínima no será menor que 30 cm
- La relación entre la menor dimensión transversal y la dimensión transversal perpendicular no deben ser menor que 0.4
- La resistencias a flexión de las columnas en un nudo deben satisfacer la Ecuación 4.7:

$$\sum M_e \geq 1.5 \sum M_g \quad \text{Ecuación 4.7}$$

donde:

$M_e$ : Momentos resistentes de las columnas que llegan al nudo

$M_g$ : Momentos resistentes de las vigas que llegan al nudo

#### 4.2.2.4 *Losas y condiciones de apoyos*

Debido a que las losas en los edificios contribuyen considerablemente en las rigideces de entrepiso, se decidió que a los modelos realizados en ETABS se les agregaría el efecto producido por las losas. Las losas en los modelos, no se diseñaron, es decir, no se les asignó un porcentaje de acero ni se dimensionaron, en cambio solo se incluyeron al modelo por cuestiones de rigidez en el análisis sísmico.

Para determinar el espesor de la losa se recurrió a trabajos anteriores en donde se diseñaron edificios similares [16]. El espesor de las losas de concreto que se utilizaron en los modelos corresponde a 11 cm para edificios de 4 y 5 niveles, 12 cm para edificios de 6 y 7 niveles, y 13 cm para edificios de entre 8 y 10 niveles.

Hay que tener en cuenta que aunque la RNA no entrega resultados de dimensiones y cantidad de acero en las losas, es conveniente utilizar estos espesores en edificios diseñados con la red para que su comportamiento sea realista.

De igual manera, a todos los modelos realizados en ETABS se les asignaron condiciones de apoyo en la base de empotramiento, es decir, todas las columnas del primer nivel están empotradas en el suelo.

#### 4.2.2.5 *Estado límite de servicio*

Una vez realizado el análisis dinámico de la estructura y calculados los desplazamientos de la misma es necesario revisar si esta pasa los estados límites de servicio recomendados por las NTC-DS [13]. El apéndice A de estas normas establece que se revisarán los desplazamientos para evitar el daño de los elementos no estructurales, así como para garantizar la seguridad contra el colapso. Los dos límites en

cuestión se comparan con la distorsión máximas de entrepiso (*“Drift”*), es decir la relación del desplazamiento entre la altura de entrepiso. En la Tabla 4.8 se muestran los límites que se utilizaron en este trabajo.

Tabla 4.8: Estado límite de servicio

Estado límite de servicio	Máxima distorsión de entrepiso permitida
<b>Daño de elemento no estructurales</b>	0.004
<b>Seguridad contra colapso</b>	0.030

Para revisar los límites no basta con compararlos con la respuesta del análisis dinámico, hay que multiplicar las distorsiones del análisis por un factor que toma en cuenta la respuesta elasto-plástica de los materiales. La razón de utilizar este factor es porque el análisis modal solo se hace dentro del rango elástico de los materiales, y la ductilidad y sobre-resistencia se toman en cuenta en el espectro. Al multiplicar la respuesta por este factor se tiene una mejor aproximación a la respuesta real de la estructura. Las Ecuaciones 4.8 y 4.9, muestran estos factores de amplificación para cada estado límite.

Daño de elementos no estructurales:

$$Q'R/7 \qquad \text{Ecuación 4.8}$$

Seguridad contra colapso:

$$QR \qquad \text{Ecuación 4.9}$$

donde:

*Q*: Coeficiente sísmico

*Q'*: Coeficiente sísmico en función del periodo

*R*: Sobre resistencia en función del periodo

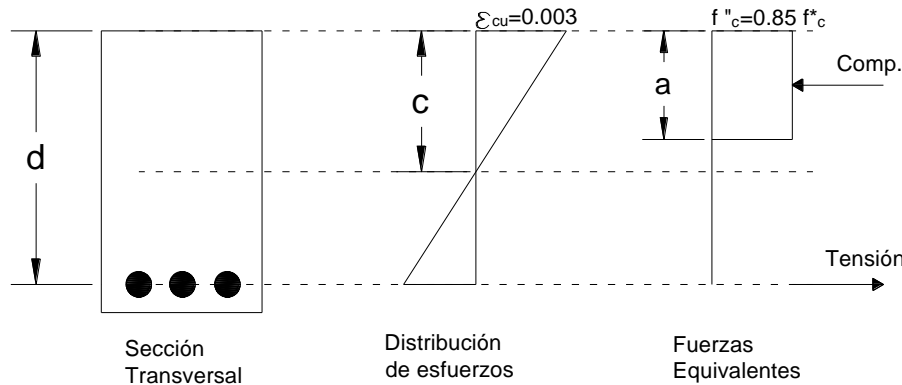
#### 4.2.3 Diseño de los elementos de concreto reforzado

El diseño de los elementos vigas y columnas se realizó con la ayuda de un código escrito en Matlab. El código toma los resultados del análisis estructural y separa las máximas solicitaciones para cada elemento estructural del edificio (ver Figura 4.5 y Ecuación 4.2). Con las máximas solicitaciones dimensiona los elementos estructurales para que resistan las fuerzas y los momentos actuantes. El diagrama de flujo de este código se encuentra en el Anexo A de este trabajo. Las fórmulas de diseño y

los factores de reducción de resistencia, utilizados en el código de Matlab se tomaron del RCDF-2004 [15]. Algunas de las fórmulas y consideraciones utilizadas para escribir el código se muestran enseguida.

### Vigas

Para el diseño de vigas, además de las consideraciones de marcos dúctiles, se tomaron en cuenta las hipótesis del bloque equivalente de esfuerzos (ver Figura 4.8).



**Figura 4.8: Hipótesis utilizada para el diseño de vigas**

Las NTC-DEC [15] establecen que se debe de diseñar de tal manera que el área máxima de acero no debe de exceder del 75% del área correspondiente a la falla balanceada. El factor de reducción que se utilizó fue el que establecen las NTC-DEC [15] de 0.9.

Todas las vigas diseñadas son doblemente reforzadas, y tienen dos configuraciones de acero de refuerzo dependiendo de la dirección del momento resistente (ver Figura 4.3). El recubrimiento del refuerzo tanto de tensión como de compresión se tomó de 4 cm, sin importar las dimensiones de la viga ni la cantidad de acero. Esta consideración se tomó debido a que la resistencia de las vigas no cambia considerablemente si aumenta o disminuye el recubrimiento y un recubrimiento de 4 cm permite utilizar varillas de cualquier diámetro comercial como refuerzo.

### Columnas

Al igual que las vigas, las columnas se diseñaron con las hipótesis de la Figura 4.8; sin embargo, en las columnas se debe tomar en cuenta el efecto de flexión biaxial. Para el tipo de secciones con las que se trabajaron (prismáticas), las NTC-DEC [15] permiten utilizar la fórmula de Bresler (Ecuación 4.10). Esta fórmula sirve para calcular los valores máximos de la carga axial que actúa con excentricidades  $e_x$  y  $e_y$ .

$$\frac{1}{P_R} = \frac{1}{P_{R_x}} + \frac{1}{P_{R_y}} - \frac{1}{P_{R0}}$$

**Ecuación 4.10**

donde:

$P_R$ : Carga normal resistente de diseño, aplicada con las excentricidades  $e_x$  y  $e_y$

$P_{Rx}$ : Carga normal resistente de diseño, aplicada con una excentricidad  $e_x$  en un plano de simetría

$P_{Ry}$ : Carga normal resistente de diseño, aplicada con una excentricidad  $e_y$  en un plano de simetría

$P_{R0}$ : Carga axial resistente de diseño, suponiendo  $e_x = e_y = 0$

Las excentricidades  $e_x$  y  $e_y$  se calcularon como la relación del momento flexionante de cada dirección entre la carga axial que recibe la columna. Dichas excentricidades nunca se tomaron menor al 5% de la dimensión del lado en la dirección cuestión o que 20 mm.

El factor de reducción que se utilizó fue el que establecen las NTC-DEC [15] para núcleos no confinados, cuando el elemento falla a tensión el factor es de 0.8, y cuando falla a compresión es de 0.7.

El acero de refuerzo en las columnas se encuentra distribuido uniformemente en el perímetro de la columna. Para poder realizar los diseños de las columnas se tuvo que fijar una configuración de los lechos de acero para cada tipo de columna, inferior y superior (ver Figura 4.9). Ambas configuraciones utilizan un recubrimiento de 5 cm para el acero de refuerzo. Estas configuraciones se tomaron iguales para todos los modelos para facilitar el diseño; sin embargo, en realidad dependen del área de acero y número de varillas. Al igual que el recubrimiento, si se modifica el número de lechos de acero la resistencia de la columna no cambiaría considerablemente.

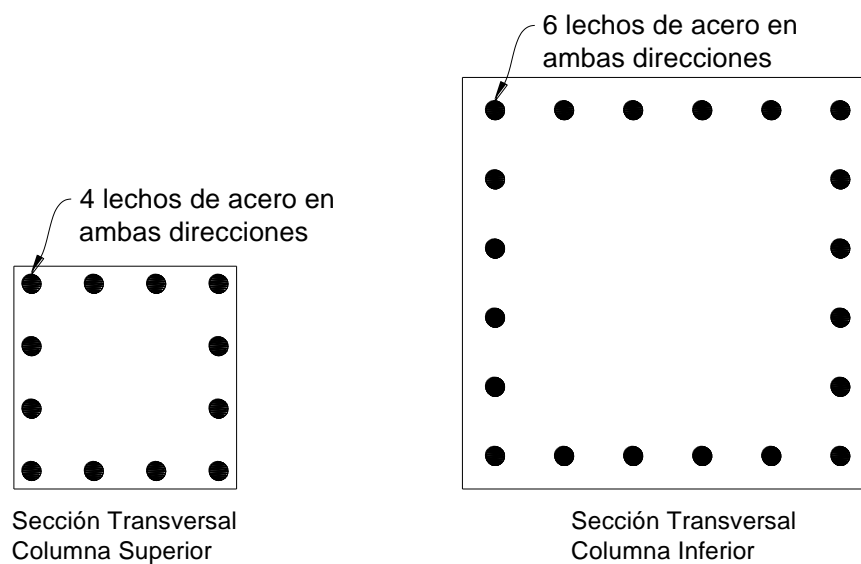


Figura 4.9: Distribución de acero para cada tipo de columna

Adicionalmente el código de Matlab también toma las consideraciones de las NTC-DEC [15] sobre marcos dúctiles que se presentan en el capítulo 4.2.2 sobre el aspecto sísmico.

#### 4.2.4 Espacio de entrenamiento

La RNA debe de tener capacidad de generalización del problema, es decir, dado cualquier ejemplo dentro del espacio de diseño, aunque el ejemplo no haya sido un patrón de entrenamiento, ésta pueda entregar un diseño que cumpla con los lineamientos del RCDF-2004. Para garantizar lo anterior el espacio de entrenamiento debe de ser lo más amplio posible y debe de estar bien distribuido en el espacio de diseño. Se determinó que si se cubría un 5% del espacio de diseño con los modelos de entrenamiento, sería suficiente para obtener buenos resultados. Para llegar al tamaño adecuado del espacio de entrenamiento procedió como sigue:

##### 4.2.4.1 *Tamaño del espacio de diseño*

En la Tabla 4.2 se muestra el espacio de diseño considerado en este estudio. Nótese que la mitad de las variables son discretas (número de niveles y crujías), y la otra mitad son continuas (largo del claro y altura de entrepiso), esto significa que el espacio de diseño tiene un número infinito de combinaciones; Sin embargo, para poder tener una idea del tamaño del espacio de entrenamiento de los modelos que se pueden diseñar con la RNA, se tomaron largos de los claros a cada metro y lo mismo con la altura de entrepiso, Tabla 4.9.

Tabla 4.9: Espacio de diseño

ESPACIO DE DISEÑO		
Variable de diseño	Rangos de valores	Tamaño de la variable
Número de Crujías en la dirección $X$	3, 4, 5	3
Número de Crujías en la dirección $Y$	3, 4, 5	3
Número de niveles	4, 5, 6, 7, 8, 9, 10	7
Claros en la dirección $X$ [m]	6, 7, 8, 9, 10	5
Claros en la dirección $Y$ [m]	6, 7, 8, 9, 10	5
Altura de entrepiso [m]	3, 4	2

Nótese en la Tabla 4.9 que las crujías y los claros están en función de la dirección  $X$  y  $Y$ . Esto significa que el espacio de diseño se puede reducir debido a que un edificio con dimensiones iguales pero orientadas en la otra perpendicularmente daría los mismos resultados. Por ejemplo, un edificio con 3 crujías en  $X$  y 5 crujías en  $Y$ , tiene los mismos resultados que un edificio con 5 crujías en  $X$  y 3 crujías en  $Y$ , siempre y cuando los demás valores se mantengan constantes. La Ecuación 4.11 se utiliza para calcular este tipo de combinaciones, la cual se le denomina combinaciones con repetición [18].

$$NC = \frac{(n + r - 1)!}{r!(n - 1)!} \quad \text{Ecuación 4.11}$$

donde:

$n$ : Número de valores que se pueden elegir

$r$ : Cantidad de valores que se eligen

$NC$ : Número de combinaciones. No importa el orden y se pueden repetir los valores

En este caso particular, para el número de crujías se adopta  $n = 3$ , y para el largo de los claros  $n = 5$ . En ambos casos  $r = 2$  ya que se tienen dos direcciones. Al aplicar la fórmula obtenemos 6 combinaciones para el primer caso y 15 combinaciones para el segundo caso.

Para calcular tamaño del espacio de diseño basta con multiplicar el tamaño de todas las variables entre sí.

*Tamaño del Espacio de Diseño = Crujías  $\times$  Niveles  $\times$  Claros  $\times$  Alturas de entrepiso*

$$\text{Tamaño del Espacio de Diseño} = 6 \times 7 \times 15 \times 2 = 1,260$$

**Ecuación 4.12**

#### 4.2.4.2 *Distribución del espacio de entrenamiento (modelos de entrenamiento)*

Conociendo el tamaño del espacio de diseño (discreto) de la RNA, podemos determinar que el espacio de entrenamiento tiene que tener cuando menos el 10% de éste, es decir, 126 modelos de entrenamiento. La base de datos recopilada en el trabajo fue de 131 modelos de entrenamiento, o sea 10.4 %. La manera en la que se distribuyen los modelos de entrenamiento dentro del espacio de diseño es importante para garantizar la generalización de la RNA.

Para obtener una distribución uniforme de los modelos de diseño sobre el espacio de diseño, se siguieron las siguientes reglas:

- Se tomaron modelos con (3,3), (5,3), (4,4), (5,5) crujías en  $(X, Y)$ .
- Los edificios de (3,3) y (3,5) crujías se modelaron con todas la combinaciones posibles de claros de 6, 8 y 10 metros, siempre tomando el claro más largo en la dirección  $X$ .
- Los edificios de (4,4) y (5,5) crujías se modelaron con todas la combinaciones posibles de claros de 5, 7 y 9 metros, siempre tomando el claro más largo en la dirección  $X$ .
- De todos los modelos anteriores se modelaron todos los niveles y alturas de entrepiso 3 y 4m.

Al final se obtuvo una base de datos de 131 edificios con la cual fue posible realizar el entrenamiento de la RNA. La Tabla 4.10 y Tabla 4.11 muestran una porción de la base de datos recopilada, la cual sigue la forma de los vectores de entrada (Ecuación 4.1) y de salida (Ecuación 4.2).



Tabla 4.11: Extracto de la base de datos recopilada (salidas)

Tabla 4.10: Extracto de la base de datos recopilada (entradas)

Modelo	#1	#2	#3
ENTRADAS	3	3	3
	3	3	3
	10	9	8
	10	10	10
	10	10	10
	4	4	4

Modelo #1

3 crujías en X

3 crujías en Y

10 Niveles

Claros en X = 10 m

Claros en Y = 10 m

Altura de entrepiso = 4 m

Modelo	#1	#2	#3	
	40	40	35	$Vp_x = \begin{bmatrix} b \\ h \\ \rho e_t \\ \rho e_c \\ \rho c_c \\ \rho c_t \end{bmatrix}$
	120	115	105	
	0.0070	0.0073	0.0055	
	0.0147	0.0132	0.0116	
	0.0036	0.0032	0.0029	
	0.0092	0.0070	0.0052	
	40	40	35	$Vp_y$
	120	115	105	
	0.0070	0.0073	0.0055	
	0.0147	0.0132	0.0116	
	0.0036	0.0032	0.0029	
	0.0092	0.0070	0.0052	
	30	30	25	$Vc_x$
	90	90	75	
	0.011	0.009	0.012	
	0.021	0.019	0.025	
	0.005	0.005	0.006	
	0.013	0.012	0.017	
	30	30	25	$Vc_y$
	90	90	75	
	0.011	0.009	0.012	
	0.021	0.019	0.025	
	0.005	0.005	0.006	
	0.013	0.012	0.017	
	180	165	155	$Ce_s = \begin{bmatrix} l_x \\ l_y \\ \rho \end{bmatrix}$
	180	165	155	
	0.0048	0.0053	0.0042	
	180	165	155	
	180	165	155	
	0.0048	0.0053	0.0042	
	185	165	155	$Ce_i$
	185	165	155	
	0.004938	0.004858	0.004589	$Cc_s$
	185	165	155	
	185	165	155	$Cc_i$
	0.0133	0.0143	0.0106	

### 4.3 Entrenamiento de la RNA

La Figura 4.6 del capítulo 4.1.2 muestra la arquitectura que va a utilizar la RNA, en este momento no se sabe cuál va a ser el número de neuronas que se utilizará en la capa oculta. Con un número muy bajo de neuronas la RNA no podrá aproximarse a los resultados que se intentan reproducir, un número muy alto de neuronas y la aproximación será tan exacta, que la red no podrá reproducir otros resultados aparte de los utilizados para el entrenamiento, es decir, pierde la capacidad de generalización.

En este capítulo se muestran los resultados obtenidos en la búsqueda de la mejor RNA (entre 20 y 30 neuronas en la capa oculta), y posteriormente se muestra la RNA que se eligió, así como su rendimiento durante el entrenamiento y el error obtenido para todos los modelos de entrenamiento. Además se presentan los vectores y matrices de pesos, y umbrales de la red elegida.

#### 4.3.1 Determinación del número de neuronas en la capa oculta

La razón por la que se determinó el rango de número de neuronas en la capa oculta de 20 a 30 neuronas fue porque se hicieron entrenamientos desde 5 hasta 50 neuronas en la capa oculta, elevando el número de neuronas de 5 en 5, y fue en el rango de 20 a 30 neuronas donde se obtuvo el mejor rendimiento. Posteriormente se profundizó la búsqueda y se realizaron cinco entrenamientos para cada caso; 21, 22, 23 y así sucesivamente hasta 30 neuronas en la capa oculta para obtener la red con mejor rendimiento.

Al igual que el ejemplo que se realizó en el capítulo 2.4 se utilizó el programa de Matlab y en particular su herramienta “*Neural Network Train Tool*” (nntraintool) para entrenar la RNA de diseño sísmico y obtener los vectores y matrices de umbrales y pesos. Esta herramienta utiliza el algoritmo de entrenamiento Levenberg-Marquardt el cual tiene una forma similar al algoritmo “backpropagation” pero con algunas variantes. Lo único que requiere esta herramienta para realizar el entrenamiento son los parámetros de entrenamiento, la arquitectura incluyendo el número de neuronas en la capa oculta y, por último el porcentaje de patrones de entrenamiento que se van a asignar a validación y comprobación.

Los patrones de entrenamiento asignados para validación y comprobación (Validation y Testing) los utiliza Matlab para detener el algoritmo en el primer caso, y para obtener el rendimiento de la red entrenada, en el segundo caso. En este trabajo se utilizó el 15% del total de los patrones de entrenamiento para validación y el 15% para comprobación en todos los entrenamientos realizados.

En este caso se definió la red con mejor rendimiento aquella red que tenga el menor error medio cuadrático, MSE (ver Ecuación 2.10), obtenido del promedio del MSE de entrenamiento, validación y comprobación:

$$MSE_{Tot} = \frac{MSE_{Training} + MSE_{Val} + MSE_{Test}}{3} \quad \text{Ecuación 4.13}$$

donde:

$MSE_{Tot}$ : Error medio cuadrático total

$MSE_{Training}$ : Error medio cuadrático de entrenamiento

$MSE_{Val}$ : Error medio cuadrático de validación

$MSE_{Test}$ : Error medio cuadrático de comprobación

Definiendo esto se realizaron cinco entrenamientos para cada arquitectura diferente, desde 20 neuronas hasta 30 neuronas en la capa oculta. Los resultados obtenidos para cada arquitectura no fueron muy diferentes; sin embargo, fue gracias a estos entrenamientos que se pudo obtener el mejor tipo de red (menor  $MSE_{Tot}$ ). En total se realizaron 55 entrenamientos los cuales se muestran en la Tabla 4.12.

Tabla 4.12: Resultados de entrenamiento

Neuronas	Índice	Error Medio Cuadrático (MSE)			
		Train	Val	Test	Total
20	1	3.10	12.23	15.82	<b>10.38</b>
20	2	2.63	16.30	14.81	<b>11.25</b>
20	3	3.87	11.90	11.74	<b>9.17</b>
20	4	6.45	15.09	12.97	<b>11.50</b>
20	5	5.01	10.50	13.32	<b>9.61</b>
21	6	4.56	9.60	13.86	<b>9.34</b>
21	7	3.61	13.39	15.24	<b>10.75</b>
21	8	2.98	8.94	11.98	<b>7.97</b>
21	9	4.46	17.40	16.94	<b>12.93</b>
21	10	4.33	20.24	12.59	<b>12.39</b>
22	11	2.89	14.22	17.51	<b>11.54</b>
22	12	1.71	17.09	18.79	<b>12.53</b>
22	13	3.37	14.39	18.15	<b>11.97</b>
22	14	1.85	9.32	17.55	<b>9.57</b>
22	15	3.75	14.86	12.37	<b>10.33</b>
23	16	2.09	17.63	25.12	<b>14.95</b>
23	17	3.54	21.09	47.80	<b>24.14</b>
23	18	3.74	17.53	14.17	<b>11.81</b>
23	19	5.00	15.60	29.57	<b>16.72</b>
23	20	5.23	14.73	29.67	<b>16.54</b>
24	21	3.23	14.85	15.99	<b>11.36</b>
24	22	3.75	13.37	16.41	<b>11.18</b>
24	23	3.98	15.59	17.57	<b>12.38</b>
24	24	3.55	16.97	16.36	<b>12.30</b>
24	25	2.49	14.57	20.76	<b>12.61</b>
25	26	3.00	12.68	19.80	<b>11.82</b>
25	27	5.75	13.97	16.49	<b>12.07</b>
25	28	4.04	12.22	17.15	<b>11.14</b>
25	29	3.16	15.94	28.99	<b>16.03</b>
25	30	3.54	13.89	18.81	<b>12.08</b>

← Mínimo  $MSE_{Tot}$

		Error Medio Cuadrático (MSE)			
Neuronas	Índice	Train	Val	Test	Total
26	31	4.27	9.75	25.32	<b>13.11</b>
26	32	6.23	17.10	17.44	<b>13.59</b>
26	33	10.56	22.77	20.39	<b>17.91</b>
26	34	4.16	17.85	32.63	<b>18.21</b>
26	35	6.63	17.76	19.35	<b>14.58</b>
27	36	3.00	12.22	28.66	<b>14.62</b>
27	37	2.46	19.50	19.35	<b>13.77</b>
27	38	2.40	15.04	20.50	<b>12.65</b>
27	39	4.97	15.71	13.82	<b>11.50</b>
27	40	4.42	13.16	14.74	<b>10.77</b>
28	41	1.44	9.90	34.55	<b>15.30</b>
28	42	4.32	28.17	23.80	<b>18.76</b>
28	43	1.49	17.99	23.49	<b>14.33</b>
28	44	15.51	28.40	26.07	<b>23.33</b>
28	45	12.32	23.42	41.84	<b>25.86</b>
29	46	8.57	14.88	30.68	<b>18.04</b>
29	47	2.79	23.33	16.89	<b>14.34</b>
29	48	1.91	26.73	14.57	<b>14.40</b>
29	49	19.09	31.39	36.31	<b>28.93</b>
29	50	2.50	19.61	23.85	<b>15.32</b>
30	51	1.18	30.41	21.45	<b>17.68</b>
30	52	2.67	18.60	18.32	<b>13.20</b>
30	53	2.51	12.87	16.38	<b>10.59</b>
30	54	4.50	34.31	28.17	<b>22.33</b>
30	55	4.09	17.74	32.78	<b>18.20</b>

Como se puede observar en la tabla anterior, el mejor resultado obtenido fue para una red de 21 neuronas en la capa oculta (Índice 7). Ésta RNA, con  $MSE_{tot} = 7.97$  es la que se eligió, y de la cual se presentarán más resultados en la siguiente sección.

Por lo tanto, la arquitectura de la red elegida es de 6:21:36, es decir, 6 entradas, 21 neuronas en la capa oculta, y 36 neuronas en la capa de salida, y utiliza función de transferencia *tansig* (ver Tabla 2.1) en la capa oculta y función de transferencia identidad en la capa de salida. La Figura 4.10 muestra la arquitectura de la red elegida.

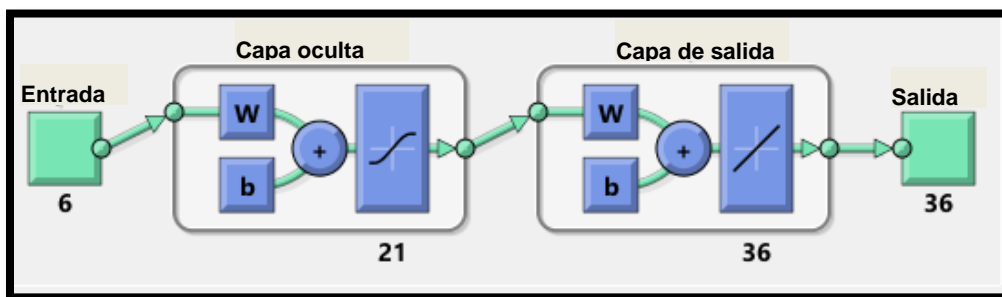


Figura 4.10: Arquitectura de la red elegida

Conocido el número de neuronas en la capa ocultas de la red, es posible conocer el tamaño de nuestras variables, umbrales y pesos. Recordando, en la Tabla 4.5 se mostró el tamaño de las variables en función del número de neuronas en la capa oculta,  $C0$ . Sustituyendo  $C0$  por 21 neuronas, obtenemos la Tabla 4.13:

**Tabla 4.13: Tamaño de matrices de pesos y umbrales de la RNA**

<b>Variable</b>	<b>Tamaño</b>	<b>Tipo de Variable</b>
$W^o$	[21, 6]	Matriz
$B^o$	[21,1]	Vector columna
$W^s$	[36, 21]	Matriz
$B^s$	[36,1]	Vector columna

#### 4.3.2 Resultados de entrenamiento para la RNA elegida

Los resultados de entrenamiento de una RNA muestran la manera en la que se llegó a las matrices de pesos y umbrales. En general, entrenar una red de 6:21:36 con Matlab no lleva mucho tiempo (aproximadamente 40 segundos en una computadora personal). El programa muestra resultados durante todo el proceso.

Uno de los resultados más relevantes es el número de épocas (ver capítulo 2.4) y el rendimiento en cada una. Para este caso se pudo minimizar el MSE en solo 13 épocas. Obteniendo el mejor rendimiento de validación en la época número 7. Estos resultados se presentan en una gráfica de MSE contra época, en la Figura 4.11.

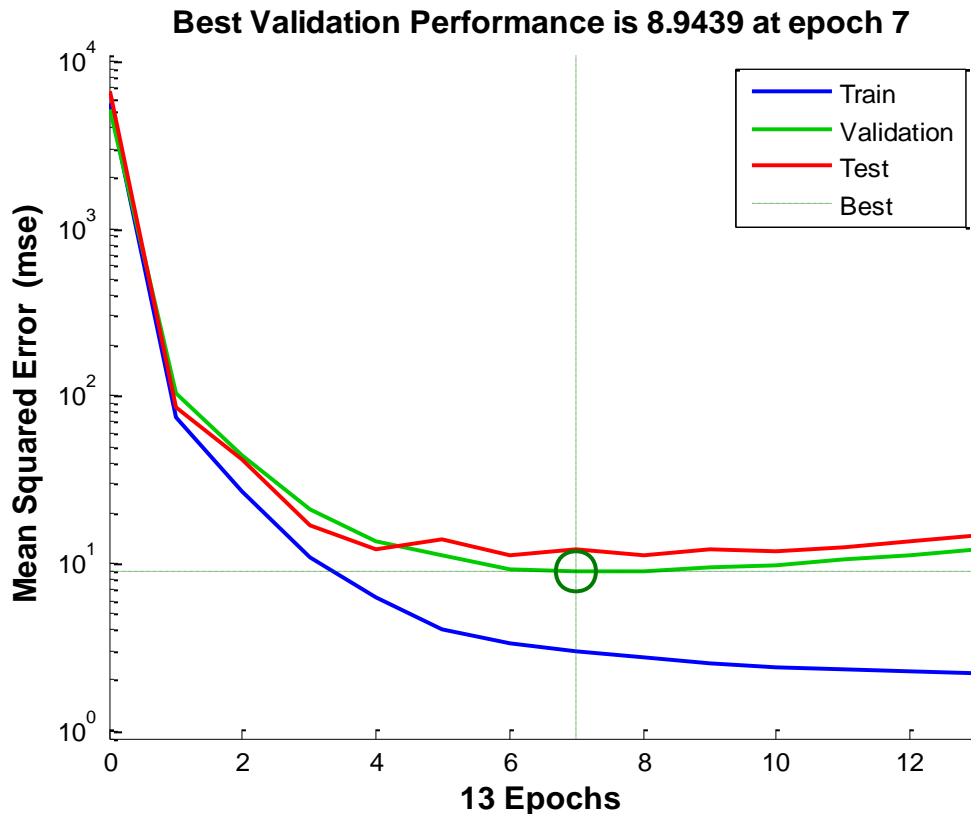


Figura 4.11: Comportamiento durante el entrenamiento

Matlab obtiene las matrices de pesos y umbrales extrayéndolas de la época con menor MSE de validación. En este caso fue la actualización de pesos y umbrales de la época 7 de donde se obtuvieron éstas matrices. La arquitectura y las matrices de pesos y umbrales son el resultado de este trabajo, ya que son las únicas variables que se necesita para operar la RNA. Como se vio en el ejemplo del capítulo 2.4 se puede operar la red ya sea simulándola en Matlab o de manera matricial, o sea, no es necesario tener Matlab para operar la red.

En las Tablas 4.14 y 4.15 se muestran los vectores y matrices de umbrales y pesos resultados de del entrenamiento de la RNA para diseño sísmico. Con estas matrices es posible operar la red, es decir, es posible obtener el diseño de un edificio (vector de salida) introduciendo las características del edificio (vector de entrada). La manera de operar la red se puede hacer como se hizo en el ejemplo del capítulo 2.4 ó, como se verá más adelante, en el capítulo 5.2.

En el caso de las matrices de pesos ( $W$ ) el número de columnas está determinado por las entradas de la capa, en cambio el número de filas se determina por el número de neuronas en la capa. Por otro lado los vectores de umbrales ( $B$ ) tienen el mismo número de filas que neuronas en la capa.

# Capa Oculta

$B^o =$

FILA/ COL	1
1	-2.4060
2	2.3860
3	-2.1787
4	1.3251
5	1.6750
6	-0.6187
7	-0.4640
8	0.7569
9	-0.2125
10	-0.0431
11	-1.1550
12	-0.3819
13	-0.6142
14	1.1382
15	-1.0387
16	1.2310
17	-1.9410
18	-0.4589
19	-1.3888
20	-2.1198
21	2.9963

$W^o =$

FILA/ COL	1	2	3	4	5	6
1	0.9039	-0.9025	1.3772	0.6968	2.4254	0.4320
2	-0.1773	0.2462	-1.0728	-0.6986	0.7001	1.6530
3	0.7117	-1.1811	0.4303	0.1124	0.3312	0.1509
4	-0.2215	-1.2664	-0.9632	-0.5374	-0.7073	-0.3074
5	0.3109	-1.1451	-0.4453	-1.0497	0.7627	-0.5670
6	0.2396	-1.1760	-0.5421	0.1885	1.5277	-0.0739
7	0.3779	0.7768	-0.7787	-0.1022	0.0124	-0.1124
8	-1.7049	0.9285	0.0228	-1.0033	-1.2601	0.0891
9	-0.2199	0.1877	0.3236	0.2755	-1.3676	-0.2746
10	0.8887	0.2810	-0.9664	0.1017	-0.9795	-0.0082
11	-0.2687	-0.4119	-1.5034	0.0795	0.0161	1.8682
12	0.1574	-0.0148	0.7423	-0.8362	0.7658	0.1108
13	-0.1724	-0.6109	-0.6695	0.0670	0.0560	-0.8726
14	1.0870	-0.8324	-0.3587	0.5304	1.3992	1.0831
15	-1.9328	0.0897	-0.4977	0.7476	-1.1260	2.1916
16	0.4477	1.2010	1.1745	-0.3336	1.0113	0.3482
17	-0.1447	-1.9954	-1.2857	-0.1365	-0.3221	-0.3051
18	-0.6835	-0.9621	-0.1854	-0.4194	0.9233	0.0073
19	-0.5520	-0.0924	-2.5700	-1.3570	1.0105	0.2932
20	-0.2754	1.6291	-0.6466	1.5308	1.7757	-1.0150
21	1.6154	-0.9089	1.1601	0.5825	-0.2629	1.7139

Tabla 4.14: Matrices de umbrales ( $B^o$ ) y pesos ( $W^o$ ) de la capa oculta

## Capa de Salida

$B^S =$

$W^S =$

FILA/ COL	1	FILA/ COL	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	-0.6487	1	0.3637	0.3300	-0.4265	-0.0685	-0.2293	-0.0026	-0.2219	-0.4769	0.2548	-0.1044	-0.3213	0.0951	-0.3536	-0.0119	0.0889	-0.3646	-0.4387	0.0803	0.0572	-0.2440	-0.5064
2	-0.9587	2	0.1154	-0.0275	-0.2671	-0.1610	-0.1258	-0.1526	-0.5114	-0.2178	-0.2190	0.1039	-0.3360	-0.2371	-0.4882	0.1134	-0.0015	-0.2596	-0.5532	0.3081	-0.0265	-0.1794	-0.0233
3	-0.9750	3	-0.2944	-0.7993	0.2380	-0.6556	0.7043	-0.7522	-0.3935	-0.9375	0.4471	0.4310	0.8688	-0.3956	0.1239	-0.6498	-0.6479	-0.9192	0.8119	0.3439	0.9291	0.7020	-0.6768
4	0.7160	4	0.6906	-0.4190	-0.8889	0.4077	0.4010	0.3985	-0.7637	-0.0734	-0.5394	-0.5762	-0.8985	-0.2765	-0.1205	-0.2753	0.8922	0.8328	0.1293	0.4220	-0.4210	-0.0005	-0.0318
5	-0.7435	5	-0.0511	-0.0694	0.9281	-0.5757	-0.4630	-0.3509	-0.1501	0.5932	-0.3201	-0.3342	0.6521	-0.1008	-0.1087	-0.8520	0.1635	0.6419	-0.2657	-0.9486	0.3760	0.8666	0.2632
6	-0.8622	6	-0.9264	-0.8530	0.4334	0.9057	-0.3076	0.9827	-0.4760	-0.3047	-0.6643	0.4023	0.4139	-0.3552	-0.6114	0.0022	-0.7083	-0.2550	-0.4699	-0.7763	-0.2386	0.6258	0.1209
7	-0.3393	7	0.2905	0.2679	-0.1739	-0.1700	-0.2358	-0.2586	-0.1164	-0.5425	0.1965	-0.1332	-0.2221	0.0641	-0.2385	-0.0468	0.0338	-0.3805	-0.4715	0.3482	0.0363	-0.3149	-0.4786
8	-0.7783	8	0.1040	0.0226	-0.0752	-0.2691	-0.1309	-0.3447	-0.4769	-0.3060	-0.1053	0.0569	-0.1198	-0.2524	-0.2129	0.0416	-0.0560	-0.2280	-0.5878	0.4185	-0.0735	-0.1964	0.1438
9	0.7169	9	0.4047	0.3377	-0.6497	0.0355	-0.8215	0.9100	-0.5967	0.6561	0.9447	0.5948	0.8689	0.4695	0.8514	0.6633	-0.8908	-0.0677	0.4832	0.6393	0.5148	-0.0044	-0.7405
10	0.9257	10	-0.0919	0.7976	-0.5855	-0.1586	0.1882	0.7851	0.2509	-0.2879	0.6292	0.8416	-0.1435	-0.4515	0.0347	0.4123	-0.3769	-0.9054	-0.9023	0.3632	-0.2103	0.1405	0.1011
11	0.9935	11	-0.1712	-0.3942	-0.4499	0.7071	0.6197	0.7005	-0.9237	-0.3213	-0.0534	-0.3109	0.4223	-0.0350	-0.9375	-0.5661	0.1177	-0.5533	0.6750	0.7105	0.1107	0.1934	-0.3158
12	0.3681	12	-0.5054	-0.7031	-0.0920	-0.8674	0.4087	0.9111	0.9705	0.3390	-0.0040	0.7888	-0.4346	0.9052	-0.5419	0.9014	-0.7976	-0.1779	0.8954	0.6032	0.3675	-0.3560	0.1069
13	-0.5219	13	-0.0204	0.1209	0.6594	-0.2238	0.0756	0.2614	0.2409	0.0998	0.0181	-0.2135	-0.0748	-0.0771	-0.1965	-0.0768	-0.0467	-0.3857	-0.3659	0.0416	-0.0263	-0.1089	0.0799
14	-0.5833	14	0.0120	0.0236	0.4474	-0.3582	0.1457	0.1897	-0.2476	0.3485	-0.2957	-0.2067	-0.1947	-0.2902	-0.1722	0.0206	0.0693	-0.1186	-0.1859	-0.1582	-0.0846	0.0341	0.3121
15	-0.4603	15	-0.0994	0.6137	0.4465	0.2447	-0.6879	0.2578	-0.9061	0.7849	-0.3515	-0.8043	-0.3779	0.1745	-0.7856	-0.4052	0.1408	-0.4589	0.9359	-0.8831	-0.0578	-0.2258	0.3460
16	-0.5969	16	-0.3198	0.5725	-0.8109	0.7135	-0.2213	0.6371	-0.7494	0.4822	-0.5792	0.6777	-0.1413	0.9254	-0.3143	0.6037	0.8009	-0.7313	-0.5791	-0.7786	0.9635	-0.1622	0.7837
17	-0.1824	17	0.8006	0.6002	0.6357	0.2518	0.0316	0.4053	0.9265	-0.3049	0.0253	-0.0543	-0.9340	-0.0733	0.9503	0.9021	0.5782	0.7613	-0.2553	0.9403	-0.5836	-0.8965	-0.8516
18	0.5041	18	0.4430	0.7479	0.9285	-0.5968	0.5973	-0.2040	-0.8173	0.9775	0.4421	-0.0164	0.6079	-0.7119	-0.9475	-0.2488	0.9185	-0.3309	-0.4233	-0.6007	-0.3304	0.5822	0.7404
19	-0.2805	19	0.0148	0.0902	0.7433	-0.3739	0.1765	0.3255	0.2407	0.1943	0.0584	-0.1210	0.0088	0.0630	0.0172	-0.0510	0.0133	-0.3183	-0.3634	-0.0164	0.0007	-0.1006	0.0717
20	-0.9010	20	0.0125	0.0204	0.2891	-0.3388	0.2087	0.2362	-0.1859	0.2524	-0.1543	-0.1452	-0.2122	-0.2677	-0.2297	0.1811	-0.0506	0.2895	-0.0235	0.0382	0.0211	-0.0914	0.2213
21	-0.8703	21	-0.6602	-0.0312	0.0995	0.4131	-0.0744	-0.6859	-0.8065	0.3726	-0.9340	-0.9959	-0.9367	0.3600	-0.9426	0.9157	-0.6330	0.6000	-0.7276	0.8884	-0.0442	0.3070	0.6810
22	-0.7738	22	0.4939	-0.3926	-0.5816	-0.0748	-0.0398	0.0605	0.1417	0.6621	-0.8447	-0.3697	0.8747	-0.5289	0.4245	-0.9710	0.3178	0.2254	-0.9344	-0.9912	0.1451	0.4306	0.3043
23	-0.5790	23	0.3031	-0.9961	-0.3121	0.4586	0.7501	-0.1726	-0.3092	-0.2046	0.8201	0.0875	-0.4507	0.8919	-0.1245	0.3819	0.9980	-0.3845	0.7634	0.1035	-0.4821	-0.2416	-0.3754
24	0.3798	24	0.9802	0.4699	0.4990	0.5942	0.1969	0.1045	0.2122	0.4310	0.6018	0.8223	-0.5892	-0.1281	-0.7871	-0.9726	-0.7996	-0.8263	0.3502	0.6639	-0.1696	0.2119	0.1442
25	-0.8234	25	0.1931	-0.0545	-0.6111	-0.2781	0.2057	-0.2533	-0.1725	-0.1598	-0.2107	0.0182	-0.1596	-0.0396	-0.3129	0.1697	-0.0198	-0.1853	-0.3351	0.3279	-0.0954	-0.1418	-0.1827
26	-0.5021	26	0.1603	-0.1183	0.2766	-0.2133	0.0161	-0.2444	-0.5063	-0.2120	-0.0742	0.2088	-0.3214	-0.5783	-0.5691	0.1780	0.0201	-0.3830	-0.4198	0.2140	-0.0704	-0.1028	-0.0776
27	-0.5740	27	-0.3076	0.9445	0.3288	0.5429	0.8459	-0.8317	0.6832	-0.7887	0.7157	-0.9087	0.7434	0.2824	-0.6894	0.1216	-0.0744	0.2894	0.6525	0.6061	0.0772	0.9990	-0.5033
28	-0.8234	28	0.1931	-0.0544	-0.6111	-0.2781	0.2057	-0.2533	-0.1726	-0.1598	-0.2105	0.0182	-0.1596	-0.0396	-0.3129	0.1697	-0.0199	-0.1854	-0.3352	0.3278	-0.0953	-0.1418	-0.1827
29	-0.5021	29	0.1603	-0.1184	0.2766	-0.2133	0.0161	-0.2444	-0.5063	-0.2120	-0.0743	0.2088	-0.3215	-0.5784	-0.5692	0.1780	0.0201	-0.3830	-0.4198	0.2140	-0.0705	-0.1028	-0.0776
30	-0.6031	30	0.8193	0.9756	0.9956	0.4638	0.8446	-0.0757	0.0155	0.7305	0.2471	-0.4913	-0.6113	0.0998	-0.2630	-0.0479	0.9241	0.1706	-0.7349	0.9475	-0.3110	-0.9265	0.5763
31	-0.1220	31	0.1079	-0.1227	0.1647	-0.2981	-0.0030	-0.2871	-0.1647	-0.1287	-0.2821	0.0362	-0.2907	-0.2061	-0.4491	0.1035	0.0724	-0.3351	-0.3516	0.4032	-0.1328	-0.0751	-0.0517
32	-0.1219	32	0.1079	-0.1227	0.1647	-0.2981	-0.0031	-0.2871	-0.1646	-0.1287	-0.2822	0.0363	-0.2908	-0.2061	-0.4492	0.1035	0.0725	-0.3352	-0.3516	0.4032	-0.1328	-0.0751	-0.0518
33	-0.2518	33	-0.2564	0.1655	-0.1034	-0.5056	-0.2199	-0.2664	-0.6476	-0.7488	0.5501	0.6951	0.9978	-0.5858	0.0606	-0.4952	0.0085	-0.0113	0.6633	-0.8628	0.1312	-0.7193	-0.2341
34	-0.0974	34	0.1116	-0.1282	0.1694	-0.2988	-0.0037	-0.2889	-0.1586	-0.1317	-0.2814	0.0488	-0.2946	-0.1997	-0.4567	0.1061	0.0765	-0.3530	-0.3645	0.4175	-0.1326	-0.0745	-0.0642
35	-0.0973	35	0.1115	-0.1284	0.1695	-0.2989	-0.0037	-0.2890	-0.1585	-0.1316	-0.2815	0.0488	-0.2946	-0.1997	-0.4566	0.1062	0.0765	-0.3529	-0.3644	0.4176	-0.1327	-0.0745	-0.0641
36	0.1005	36	0.9500	0.7618	-0.3278	0.1341	0.2347	0.7693	-0.2140	-0.6506	-0.1563	0.4656	0.7718	-0.8325	-0.9495	0.4128	-0.0884	-0.8772	0.4028	0.6987	-0.3684	-0.9293	0.8941

Tabla 4.15: Matrices de umbrales ( $B^0$ ) y pesos ( $W^0$ ) de la capa de salida



Otro de los resultados que arroja Matlab es relación que tienen las salidas de la red, con el objetivo deseado (salida de los modelos de entrenamiento). Esta comparación de las grafica es de tal forma que el eje de las abscisas es el objetivo (“*Target*”), y el eje de las ordenadas es el resultado de la red. De tal forma que la red perfecta coincidiría con una recta a 45° ,  $Y = T$ . En la Figura 4.12 se muestra este ajuste para el entrenamiento, la validación, la comprobación y el total (azul, rojo, verde y gris respectivamente ídem anterior colores). Se puede observar que el ajuste tiene una precisión aceptable ya que la correlación,  $R$  es mayor a 95% en todos los casos. El número de datos graficados es el total de salidas con las que se entrenó la red, es decir,  $131 \times 36 = 4,716$  datos.

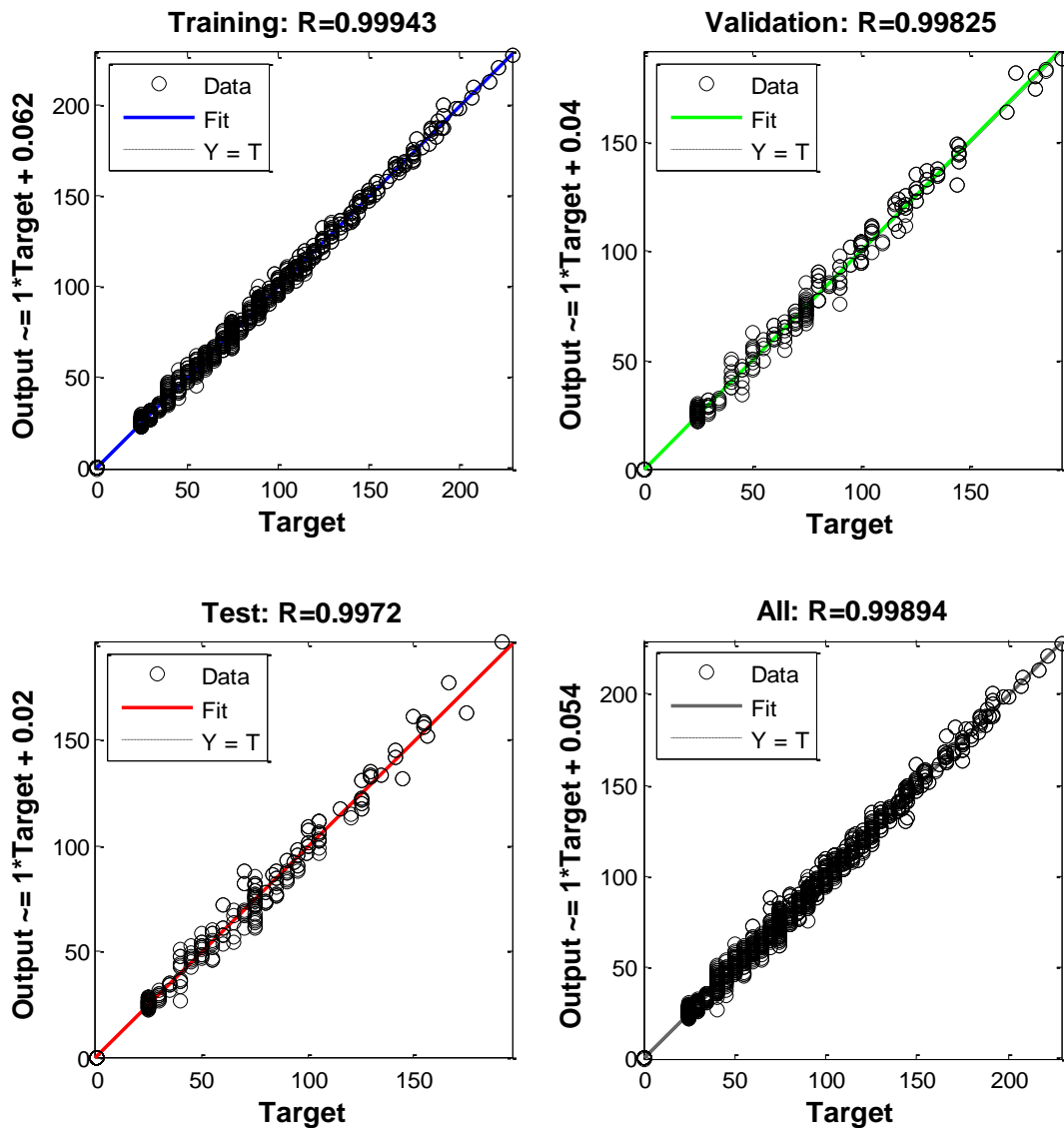


Figura 4.12: Ajuste de datos de entrenamiento (azul), validación (verde), comprobación (rojo), y total (gris)

El histograma de errores es una gráfica que permite conocer el rango de error con el que trabaja la red. Este histograma, al igual que el ajuste de la figura anterior, compara los resultados de la red con el objetivo deseado, obteniendo un error [5]. A diferencia con el MSE, el error calculado en el histograma de errores no está elevado al cuadrado, es simplemente la resta del objetivo menos la salida de la red, de tal forma que es posible obtener valores negativos. En la Figura 4.13 se muestra este histograma, y se puede observar que de los 4,716 valores calculados la gran mayoría se encuentran en un rango de error entre -3 y 3, lo que permite observar el ajuste que la red obtuvo con los patrones de entrenamiento. Además, en la misma figura se puede observar que el error máximo es de 13.23 y el mínimo es de -17.35 pero representan un porcentaje muy pequeño de todo el conjunto.

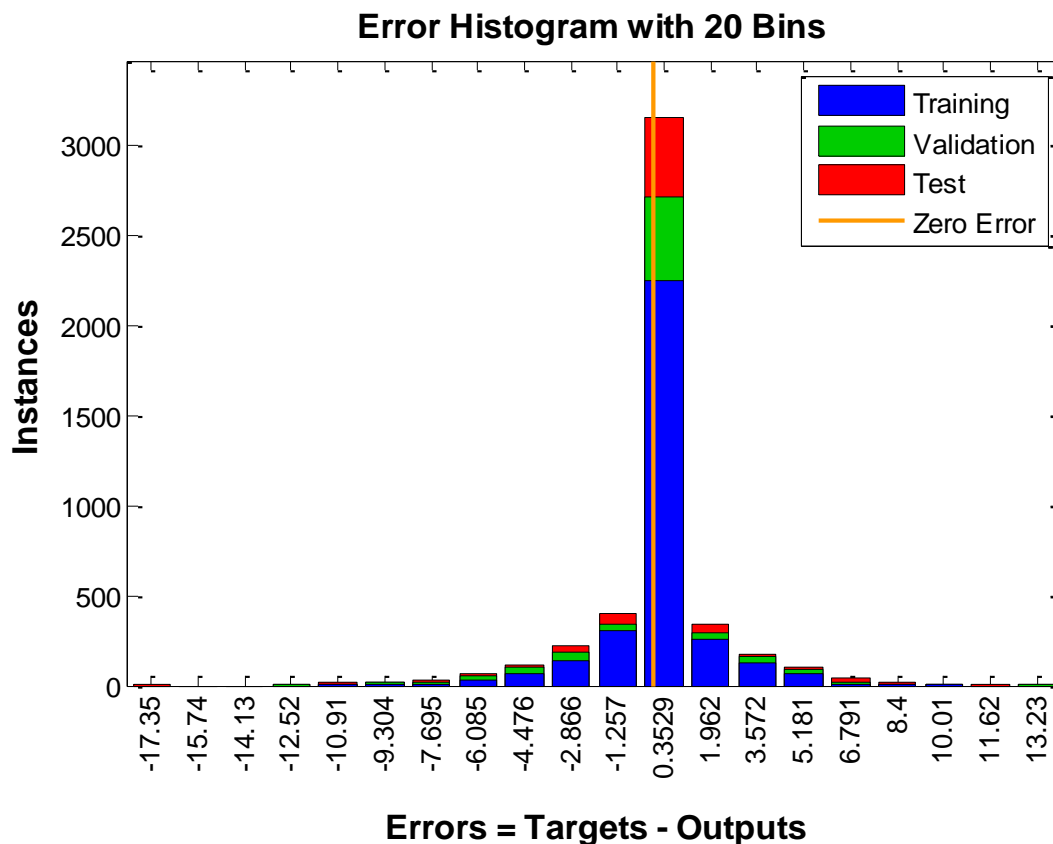


Figura 4.13: Histograma de error, objetivo - resultados

Con estos resultados podría ser suficiente para evaluar el rendimiento de la RNA obtenida. Estos resultados permiten visualizar el ajuste de la red a este problema y el nivel de errores que la red obtiene. La red aquí presentada está lista para operarse, es decir, para diseñar cualquier edificio que se encuentre dentro del espacio de diseño (capítulo 4.1).

Para ilustrar esto, en el siguiente capítulo se hará, a modo de ejemplo, una comparación de un edificio diseñado con métodos convencionales y alternativamente con la RNA aquí propuesta. Esta comparación permite observar la aplicación de las RNA al diseño sísmico.

## 5 Comprobación del diseño hechos con métodos convencionales y alternativamente, con la RNA

En este capítulo se pone a prueba un diseño realizado con la RNA. En primer lugar se realiza el diseño de un edificio con la RNA, después el diseño se modela en ETABS para realizar el análisis estructural y revisar si las distorsiones máximas se encuentran dentro de los valores permisibles, por último se hace un análisis de los resultados obtenidos.

### 5.1 Datos del edificio a diseñar con la RNA

Para generar un diseño con la red (operación de la red) en primer lugar hay que saber la geometría del edificio que se desea diseñar. Para este caso se ha elegido un edificio con 4 crujiás en  $X$ , 3 crujiás en  $Y$ , 5 niveles, claros en ambas direcciones de 6.5 m y altura de entrepiso de 3.5 m. Dicho edificio se puede representar con el vector de entrada de la Ecuación 5.1 con el cual se operara la red. Además esta geometría cumple con las características del espacio de diseño de la red (Tabla 4.2). También, dicho edificio cumple con las características de la Tabla 4.1. Es importante observar que el ejemplo aquí propuesto jamás fue presentado a la red en su etapa de entrenamiento.

$$Edificio = \begin{bmatrix} 4 \\ 3 \\ 5 \\ 6.5 \\ 6.5 \\ 3.5 \end{bmatrix} \quad \text{Ecuación 5.1}$$

### 5.2 Operación de la red

Como se explicó en el capítulo 2.4.3 es posible operar la red directamente con las matrices de pesos y umbrales (Tabla 4.14 y Tabla 4.15) o, si se tiene la red programada en Matlab, con el comando de la Ecuación 5.2.

$$Y = sim(net, X) \quad \text{Ecuación 5.2}$$

donde:

$Y$ : es el vector de salida de la red.  
 $net$ : La red programada en Matlab.  
 $X$ : es el vector de entrada de la red.

La manera como se opera la red con la Matlab (Ecuación 5.2) se explica en el diagrama de bloques del Anexo B. Como se puede observar, la ventaja de utilizar un paquete de matemáticas como Matlab para operar una red es que el resultado es directo y no es necesario normalizar los vectores como se tuvo que hacer en el ejemplo del capítulo 2.4. En este caso se realizó la simulación de la red entrenada con el vector *Edificio* (Ecuación 5.1) y los resultados que se obtienen son los de la Ecuación 5.3.

$$Vp_x = \begin{bmatrix} 21.5279 \\ 44.8271 \\ 0.0030 \\ 0.0247 \\ -0.0170 \\ 0.0095 \end{bmatrix}$$

$$Vp_y = \begin{bmatrix} 22.1683 \\ 47.5811 \\ 0.0451 \\ 0.0758 \\ 0.0317 \\ 0.0490 \end{bmatrix}$$

$$\text{Vector de salida} = \begin{bmatrix} Vp_x \\ Vp_y \\ Vc_x \\ Vc_y \\ Ce_s \\ Ce_i \\ Cc_s \\ Cc_i \end{bmatrix}$$

**Ecuación 5.3**

$$Vc_x = \begin{bmatrix} 23.8110 \\ 53.3543 \\ 0.0203 \\ 0.0803 \\ 0.0010 \\ 0.0161 \end{bmatrix}$$

donde:

$Vp_x$ : Vigas perimetrales en dirección  $X$  (Tabla 4.3).

$Vp_y$ : Vigas perimetrales en dirección  $Y$  (Tabla 4.3).

$Vc_x$ : Vigas centrales en dirección  $X$  (Tabla 4.3).

$Vc_y$ : Vigas centrales en dirección  $Y$  (Tabla 4.3).

$Ce_s$ : Columnas de esquina superiores (Tabla 4.4).

$Ce_i$ : Columnas de esquina inferiores (Tabla 4.4).

$Cc_s$ : Columnas centrales superiores (Tabla 4.4).

$Cc_i$ : Columnas centrales inferiores (Tabla 4.4).

$$Vc_y = \begin{bmatrix} 24.4899 \\ 52.6237 \\ 0.0175 \\ -0.0187 \\ 0.0034 \\ 0.0307 \end{bmatrix}$$

$$Ce_s = \begin{bmatrix} 65.8880 \\ 82.9147 \\ 0.0074 \end{bmatrix}$$

$$Ce_i = \begin{bmatrix} 65.8926 \\ 82.9126 \\ 0.0140 \end{bmatrix}$$

$$Cc_s = \begin{bmatrix} 69.6612 \\ 69.6576 \\ 0.0109 \end{bmatrix}$$

$$Cc_i = \begin{bmatrix} 69.6223 \\ 69.6264 \\ 0.0758 \end{bmatrix}$$

### 5.2.1 Interpretación de los resultados de la red

Como se puede observar en los resultados de la Ecuación 5.3 existen valores que no pueden ser posibles, por ejemplo, para  $Vp_x$  encontró que la cuantía de acero de compresión en el centro ( $\rho_{c_c}$ ) tiene signo negativo. Esto se debe a que la red tiene un margen de error y es tan poca la cantidad de acero necesaria en esta sección que el margen de error abarca una parte negativa. Para tomar en cuenta este margen de error esto se interpreta como que la viga en esa sección necesita el mínimo de acero en compresión. En el caso de la  $Vc_y$  el error es mayor, debido a que el acero en el lecho de compresiones es mayor que en el de tensiones. Para interpretar el resultado se tienen que ambos lechos tienen la misma cuantía de acero.

Además, las dimensiones de los elementos con los que trabaja la red no son números enteros, esto no es práctico en la construcción, lo que se hace es redondear estas dimensiones (en el caso de  $Vp_x$ ,  $Vp_y$ ,  $Vc_x$  y  $Vc_y$  al múltiplo de cinco más cercano). Interpretando los resultados con los criterios antes mencionados se obtienen los siguientes resultados (Ecuación 5.4):

$$\begin{array}{ccc}
 Vp_x = \begin{bmatrix} 25 \\ 45 \\ 0.0030 \\ 0.0247 \\ 0.0023 \\ 0.0095 \end{bmatrix} & Vc_x = \begin{bmatrix} 25 \\ 55 \\ 0.0203 \\ 0.0803 \\ 0.0020 \\ 0.0161 \end{bmatrix} & Ce_s = \begin{bmatrix} 65 \\ 83 \\ 0.0074 \end{bmatrix} \\
 \\
 Vp_y = \begin{bmatrix} 25 \\ 50 \\ 0.0451 \\ 0.0758 \\ 0.0317 \\ 0.0490 \end{bmatrix} & Vc_y = \begin{bmatrix} 25 \\ 55 \\ 0.0175 \\ 0.0175 \\ 0.0034 \\ 0.0307 \end{bmatrix} & Ce_i = \begin{bmatrix} 65 \\ 83 \\ 0.0140 \end{bmatrix} \\
 \\
 & & Cc_s = \begin{bmatrix} 70 \\ 70 \\ 0.0109 \end{bmatrix} \\
 & & Cc_i = \begin{bmatrix} 70 \\ 70 \\ 0.0758 \end{bmatrix}
 \end{array}$$

Ecuación 5.4

Para realizar la revisión de resistencia de este edificio hay que convertir las cuantías de acero de las secciones anteriores en áreas (cm<sup>2</sup>). Al realizar esto se puede calcular la resistencia de cada elemento para poderlo comparar con los resultados del análisis estructural. Las secciones calculadas por la red quedan como se muestra en la Ecuación 5.5.

$$\begin{aligned}
 Vp_x &= \begin{bmatrix} 25 \text{ cm} \\ 45 \text{ cm} \\ 3.32 \text{ cm}^2 \\ 27.77 \text{ cm}^2 \\ 2.59 \text{ cm}^2 \\ 10.66 \text{ cm}^2 \end{bmatrix} & Vc_x &= \begin{bmatrix} 25 \text{ cm} \\ 55 \text{ cm} \\ 27.88 \text{ cm}^2 \\ 110.41 \text{ cm}^2 \\ 2.75 \text{ cm}^2 \\ 22.16 \text{ cm}^2 \end{bmatrix} & Ce_s &= \begin{bmatrix} 65 \text{ cm} \\ 83 \text{ cm} \\ 39.71 \text{ cm}^2 \end{bmatrix} \\
 \\ \\
 Vp_y &= \begin{bmatrix} 25 \text{ cm} \\ 50 \text{ cm} \\ 56.32 \text{ cm}^2 \\ 106.77 \text{ cm}^2 \\ 84.62 \text{ cm}^2 \\ 103.70 \text{ cm}^2 \end{bmatrix} & Vc_y &= \begin{bmatrix} 25 \text{ cm} \\ 55 \text{ cm} \\ 24.08 \text{ cm}^2 \\ 24.08 \text{ cm}^2 \\ 4.70 \text{ cm}^2 \\ 42.17 \text{ cm}^2 \end{bmatrix} & Ce_i &= \begin{bmatrix} 65 \text{ cm} \\ 83 \text{ cm} \\ 75.56 \text{ cm}^2 \end{bmatrix} \\
 \\ \\
 & & & & Cc_s &= \begin{bmatrix} 70 \text{ cm} \\ 70 \text{ cm} \\ 53.60 \text{ cm}^2 \end{bmatrix} \\
 & & & & Cc_i &= \begin{bmatrix} 70 \text{ cm} \\ 70 \text{ cm} \\ 371.47 \text{ cm}^2 \end{bmatrix}
 \end{aligned}$$

Ecuación 5.5

### 5.3 Análisis estructural y sísmico

Conociendo los datos geométricos del edificio (Ecuación 5.1) y las dimensiones de los elementos estructurales (Ecuación 5.5) es posible modelar el edificio en el programa de ETABS para hacer el análisis estructural y el análisis sísmico. Con el primero podremos comparar la resistencia del edificio a las solicitaciones de resistencia, y con el segundo podremos definir si el diseño realizado con la RNA pasa los límites de servicio y de colapso.

En la Figura 5.1 se observa el edificio modelado en ETABS: se observa a) la planta, b) una elevación y c) la vista en tercera dimensión.

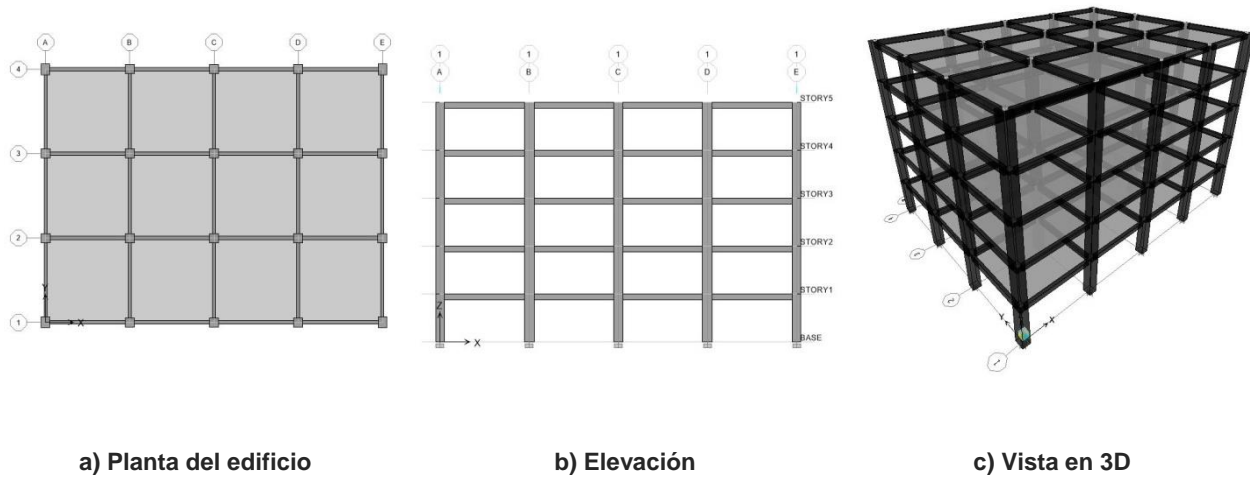


Figura 5.1: Modelo del edificio en ETABS

El modelo se realizó de acuerdo con el análisis de cargas, las combinaciones y factores de cargas y el espectro que se mencionan en las secciones 4.2.1 y 4.2.2.

### 5.3.1 Solicitaciones de servicio

Del análisis dinámico se obtuvo que el periodo fundamental de la estructura es:  $T = 0.88 \text{ seg}$ . Además, también se obtuvo la máxima distorsión de entrepiso (Drift) en cada nivel. Dicha distorsión se muestran en la Tabla 5.1:

Tabla 5.1: Distorsiones de entrepiso obtenidas

Nivel	<i>Drift X</i>	<i>Drift Y</i>
5	0.0017	0.0016
4	0.0023	0.0021
3	0.0028	0.0025
2	0.0026	0.0024
1	0.0013	0.0012

En la Tabla 5.1 se puede observar que la máxima distorsión de todo el edificio se encuentra en el nivel 3 en la dirección  $X$  y es de 0.0028; sin embargo, estas distorsiones se calcularon con la teoría elástica, por lo tanto hay que escalar dichas distorsiones con los factores del RCDF-2004 [13]: Ecuación 4.8 para daños de elementos no estructurales, y con la Ecuación 4.9 para seguridad contra colapso. Los resultados de las distorsiones máximas, escaladas por su respectivo factor se muestran en la Tabla 5.2.



Tabla 5.2: Distorsiones máximas de entrepiso permitidas

Caso	Máxima distorsión escalada obtenida del análisis
<b>Daños elementos no estructurales</b>	0.0029
<b>Seguridad contra colapso</b>	0.0172

Al comparar los valores de la Tabla 5.2 con los de la Tabla 4.8, en la cual se muestran los límites permitidos para cada caso, se observa que en ninguno de los casos se superan estos límites, lo que indica que el diseño tiene un buen comportamiento ante solicitaciones sísmicas. Es importante notar que si se quiere obtener un comportamiento mejor, alguna de las dos distorsiones se encontraría en el límite. En este caso la distorsión más cercana al límite es la de daños a elementos estructurales, lo que indica que el diseño todavía se puede mejorar en cuanto a deformaciones máximas.

### 5.3.2 Solicitaciones de resistencia

Debido a que los resultados del análisis estructural es muy extenso no se mostrarán todos los resultados en este trabajo, solo se mostrará el máximo valor de la solicitación para cada uno de los ocho elementos estructurales de los que está conformado el edificio. En la Tabla 5.3 se muestran los momentos que deberán resistir las vigas y en la Tabla 5.4 se muestran la fuerza axial y los momentos que deben resistir las columnas.

Tabla 5.3: Solicitaciones para las vigas

Tipo de viga	Momento en los extremos [T-m]	Momento en el centro [T-m]
$Vp_x$	-17.315	8.966
$Vp_y$	-21.83	13.553
$Vc_x$	-28.93	12.903
$Vc_y$	-29.197	12.986

Tabla 5.4: Solicitaciones para las columnas

Tipo de columna	Fuerza axial [T]	Momento en X [T-m]	Momento en Y [T-m]
$Ce_s$	-37.11	25.97	32.7
$Ce_i$	-98.97	25.97	32.7
$Cc_s$	-103.83	25.97	32.7
$Cc_i$	-273.77	25.97	32.7

## 5.4 Resistencia de los elementos estructurales

Con los datos del análisis y de las secciones de cada elemento geométrico es posible hacer la comparación de la resistencia contra la solicitación, de tal forma que se puede evaluar el comportamiento del edificio.

### 5.4.1 Vigas

Con la teoría de la flexión y las dimensiones y cantidades de acero obtenidas con la red (Ecuación 5.5) se calculó la resistencia de cada una de las vigas del edificio. La Tabla 5.5 muestra la resistencia de cada uno de estos elementos estructurales. Dichos momentos resistentes ya están afectados por el factor de resistencia que establece el RCDF-2004 [15].

Tabla 5.5: Resistencia de las vigas

Tipo de viga	Momento resistente en los extremos [T-m]	Momento resistente en el centro [T-m]
$Vp_x$	-29.43	14.86
$Vp_y$	-121.78	148.07
$Vc_x$	-91.96	35.36
$Vc_y$	-42.93	70.24

Como se puede observar en la Tabla 5.5 todos los momentos resistentes superan a los momentos solicitados por la estructura (Tabla 5.3). Esto quiere decir que los diseños son adecuados. La gran mayoría de los momentos resistentes superan las solicitaciones en un rango muy amplio, por ejemplo, el momento resistente de la viga  $Vp_x$  es 6 veces mayor que la solicitud. Esto se podría interpretar como una necesidad sísmica para limitar las distorsiones de entrepiso; sin embargo, como se vio en la sección 5.3.1 los límites de servicio no están optimizados, lo que indica que las secciones y las cuantías de acero todavía tienen un rango en el cual se pueden mejorar.

Es obvio que en el diseño de las vigas la distorsión de entrepiso fue la que rigió las dimensiones y las cuantías.

### 5.4.2 Columnas

Para el cálculo de la resistencia de las columnas se utilizó la Ecuación 4.10, la cual requiere que se calculen los diagramas de interacción de las columnas en ambas direcciones y de ahí se obtenga la resistencia a carga axial en cada dirección para la excentricidad máxima a la que va a ser sometida la columna. De la Figura 5.2 a la Figura 5.7 se muestran dichos diagramas de interacción.

Las excentricidades se calculan con ayuda de la Tabla 5.4 como  $M/P$ , y son las siguientes:

Tabla 5.6: Excentricidad de las columnas

Tipo de Columna	Excentricidad en X [m]	Excentricidad en Y [m]
$Ce_s$	0.700	0.881
$Ce_t$	0.262	0.330
$Cc_s$	0.250	0.315
$Cc_t$	0.095	0.119

Con estas excentricidades se pueden calcular los términos  $P_{Rx}$  y  $P_{Ry}$  con los diagramas de interacción, trazando una recta del origen que pase con todos los puntos de la misma excentricidad. En la Tabla 5.7 se muestran los resultados obtenidos para cada columna utilizando la Ecuación 4.10.  $P_R$  es la carga axial máxima que puede aceptar la columna.

**Tabla 5.7: Resistencia de las columnas**

<b>Tipo de Columna</b>	$P_{R0}$ [T]	$P_{Rx}$ [T]	$P_{Ry}$ [T]	$P_R$ [T]
$Ce_s$	867.1	83.80	87.89	45.1
$Ce_i$	987.6	396.55	408.58	252.7
$Cc_s$	846.5	368.91	305.30	208.1
$Cc_i$	1914.5	1387.67	1292.39	1028.7

Como se puede observar en la tabla anterior, ninguna de las columnas excede la sollicitación, lo que indica que el diseño realizado es aceptable. Al igual que en las vigas, en el diseño de columnas el diseño se vio regido por las distorsiones de entrepiso.

## Columna de esquina

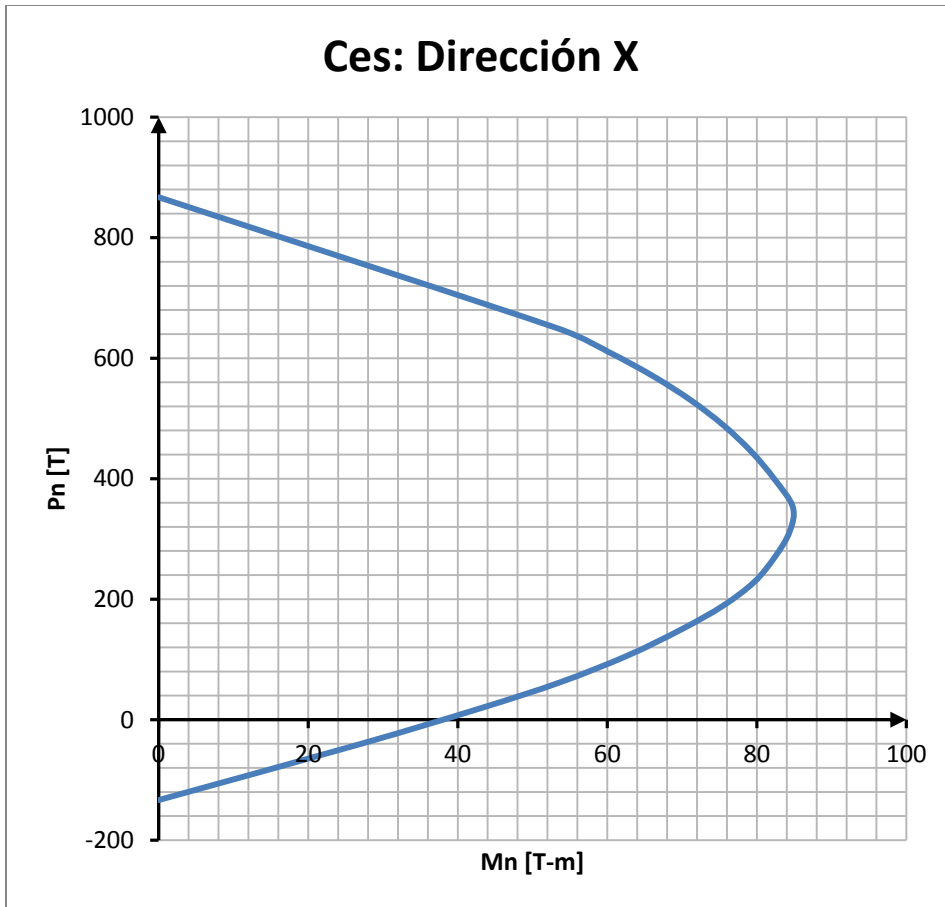


Figura 5.2: Diagrama de interacción columna de esquina superior

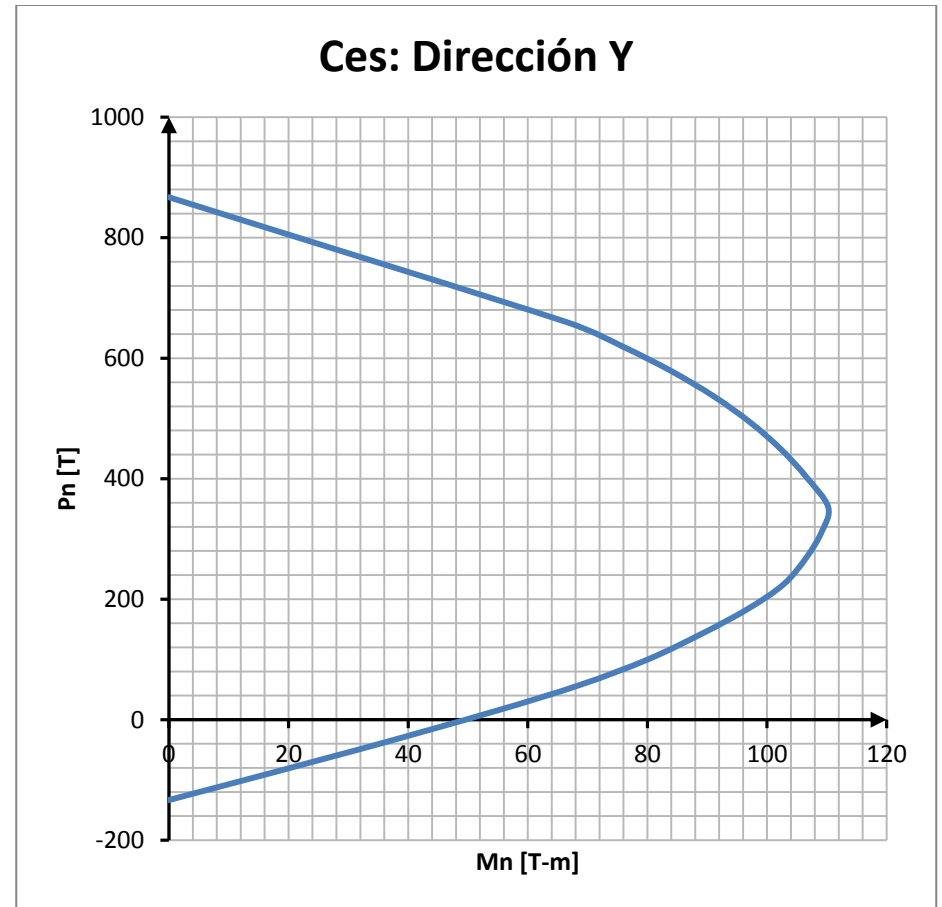


Figura 5.3: Diagrama de interacción columna de esquina superior

## Columna de esquina

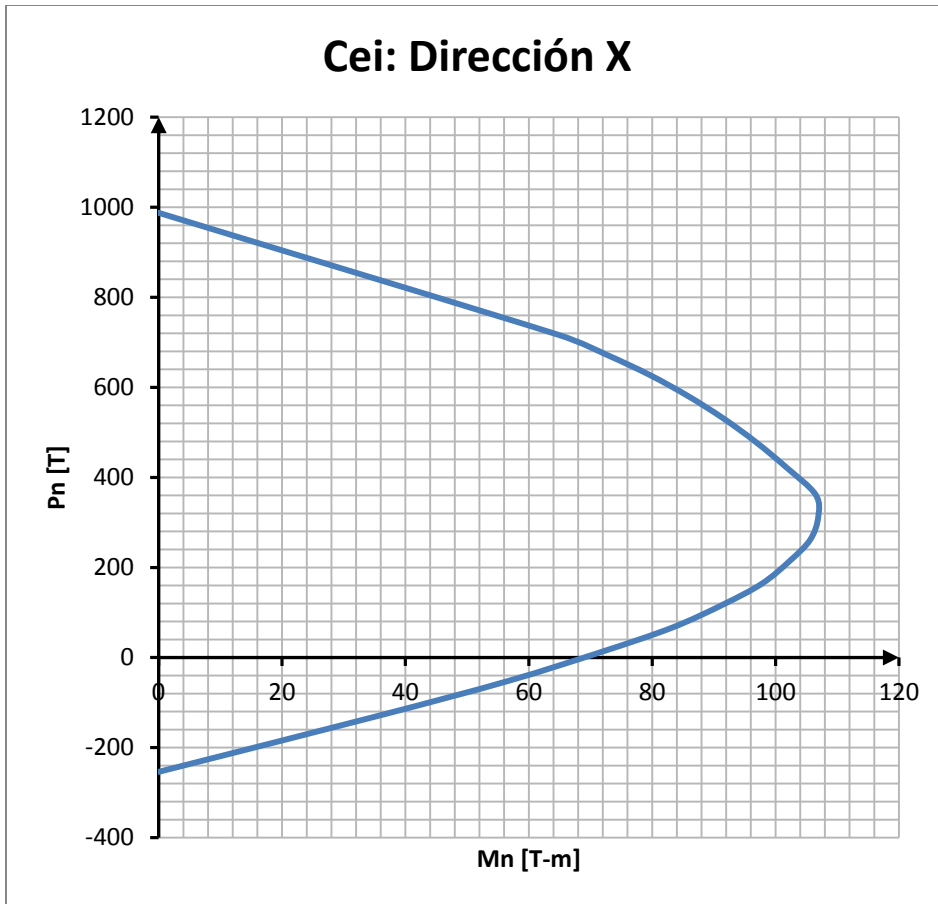


Figura 5.4: Diagrama de interacción columna de esquina inferior

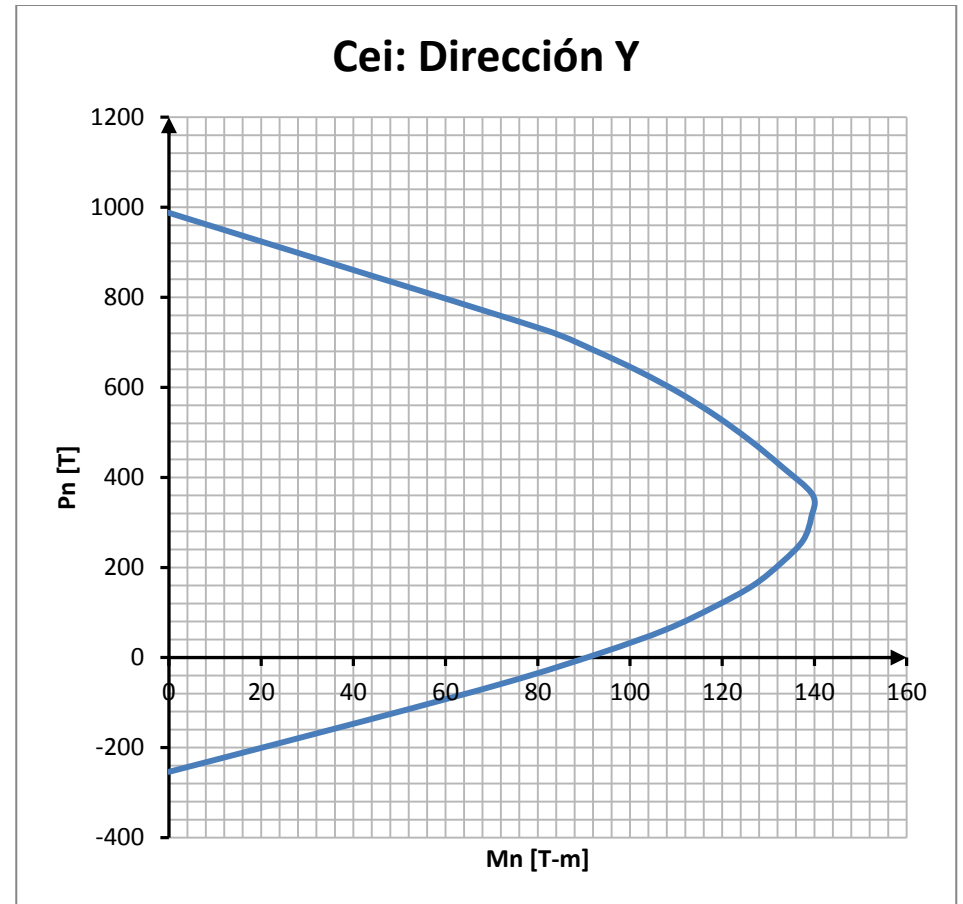


Figura 5.5: Diagrama de interacción columna de esquina inferior

## Columna central

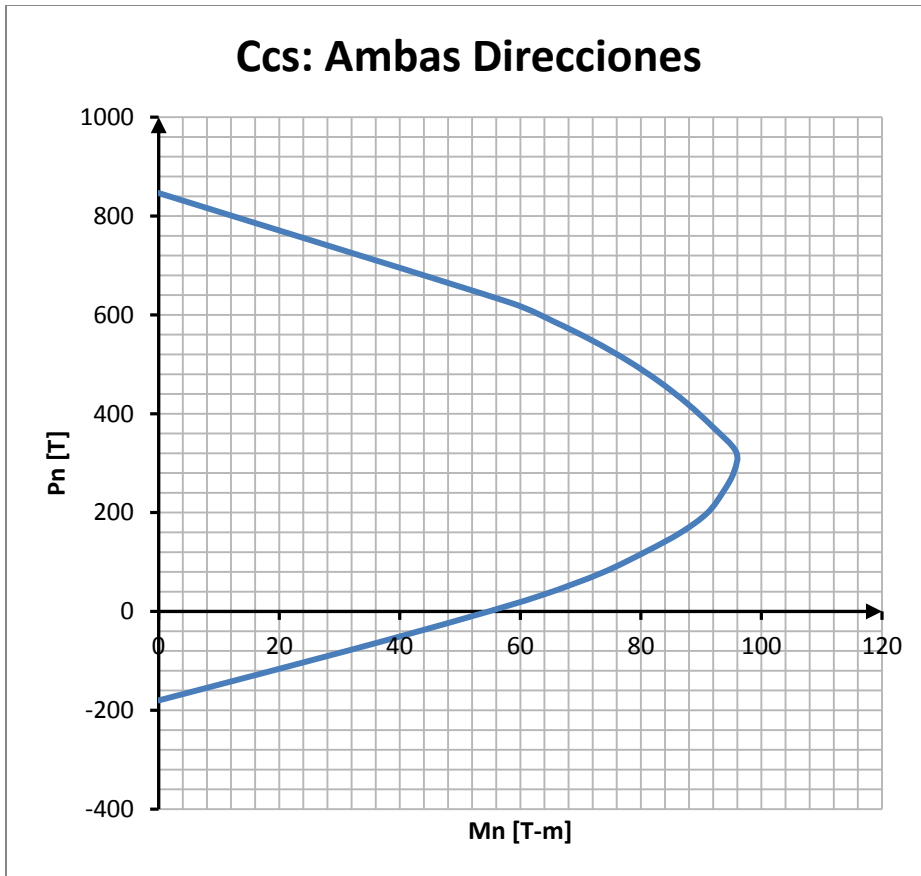


Figura 5.6: Diagrama de interacción columna central superior

## Columna central

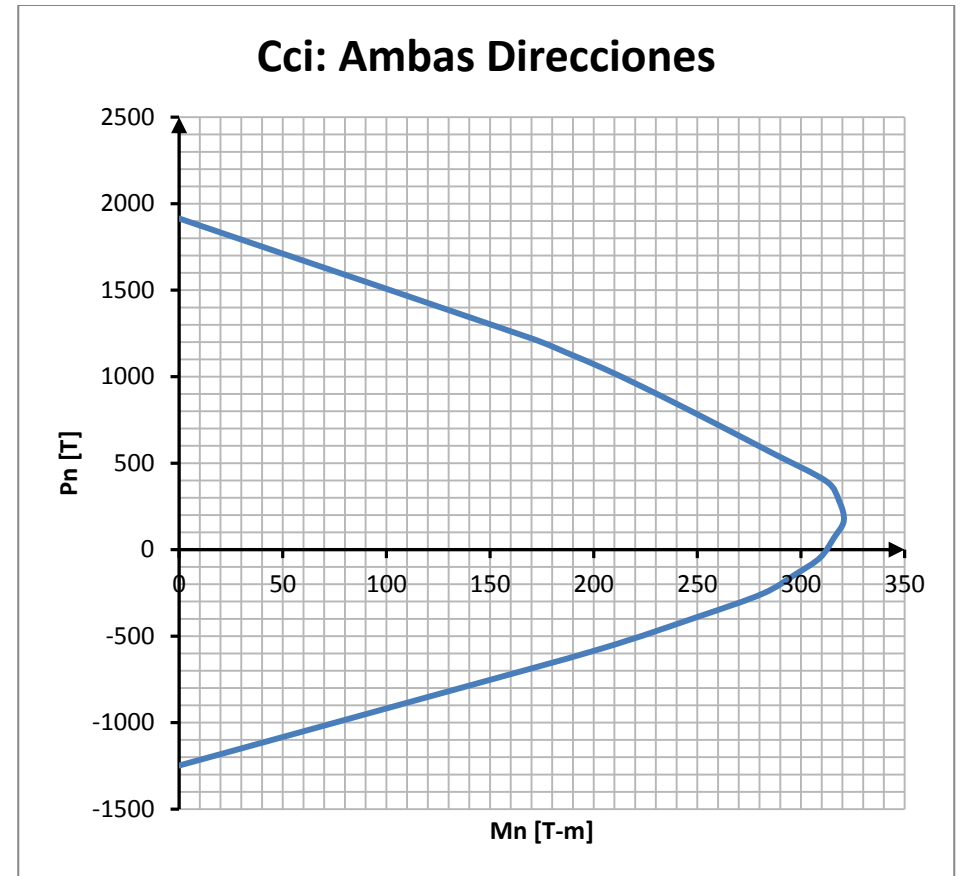


Figura 5.7: Diagrama de interacción columna central inferior

## 6 Conclusiones

Se concluye que las RNA tienen la capacidad para poder aproximar el resultado de cualquier operación con la ayuda de un conjunto de datos de salidas de entradas. Esta característica se puede aprovechar en el campo de la ingeniería estructural para realizar diseños de edificios típicos resistentes a sismos.

Los resultados obtenidos durante la etapa de entrenamiento de la red son satisfactorios, esto se debe a que los errores encontrados son menores que la unidad para la gran mayoría de ejemplos. Además también se encontró que la correlación de los resultados de la red entrenada con respecto al objetivo es del orden de 0.99 (ver Figura 4.12). De esto se concluye que existe la posibilidad de generar una red capaz de diseñar edificios en zonas sísmicas los cuales cumplan con los requisitos del código de diseño con la cual se diseña la red (en este caso RCDF-2004).

Es importante mencionar las limitaciones que tiene la red, como por ejemplo que está sujeta a la forma en la que se diseñaron los edificios, así como también los posibles errores de aproximación. Los resultados obtenidos con la red están lejos de ser óptimos aun así este tipo de herramienta sirve para dar un primer tanto al diseño estructural. Además cabe recalcar que una red con una base de datos (optimizada) y que además sea gigante obtendría resultados mejores.

En el diseño realizado con la red entrenada, se encontró que los resultados arrojados por la red no fueron tan directos como se esperaba, se realizó una interpretación de los resultados de tal manera que pudieran representar la geometría y cuantía de acero de las secciones de una manera correcta. Dicha desigualdad se refleja principalmente en las cuantías de acero de las vigas y columnas arrojadas por la red, y se atribuye a que la base de datos recopilada durante el proceso de entrenamiento asigna cuantías de acero mínimas en la mayoría de las secciones, con la intención de minimizar los costos directos de material. Para mejorar el rendimiento de una red de esta naturaleza se tendría que recopilar una base de datos mayor y revisar cada diseño uno por uno de manera manual, para que tengan la tendencia deseada con las cuantías de acero deseado.

La manera de selección de la red entre todas las redes entrenadas fue la que menor error total tuviera (Tabla 4.12), esto llevó a que el error en las cuantías de acero sea de un porcentaje mayor que en las dimensiones de las secciones debido a las unidades que utilizan en cada parámetros. Un mejor proceso de selección sería el de elegir la red con menor error en la cuantía de acero, ya que esta variable es más sensible que las dimensiones de las secciones.

Debido a los alcances de este trabajo solo se realizó una red para un tipo de edificio y una zona sísmica en el DF; sin embargo, sería posible comenzar a generar una serie de redes para otras zonas sísmicas del DF (zona-I, zona-II, zona-IIIa, etc.) e interconectar dichas redes en una estructura que permita

realizar diseños sísmicos de edificios sin importar la zona. De la misma manera, se tendrían que entrenar distintas redes para poder otros tipos de estructuración de edificios.

Como se comentó en Capítulo 1 la visión a largo plazo de las RNA para el diseño sísmico es que en algún punto de la historia sea posible realizar diseños para un área o una región específica con este tipo de herramientas. Para lograr esto es necesario que se siga mejorando el modelo utilizado en este trabajo.



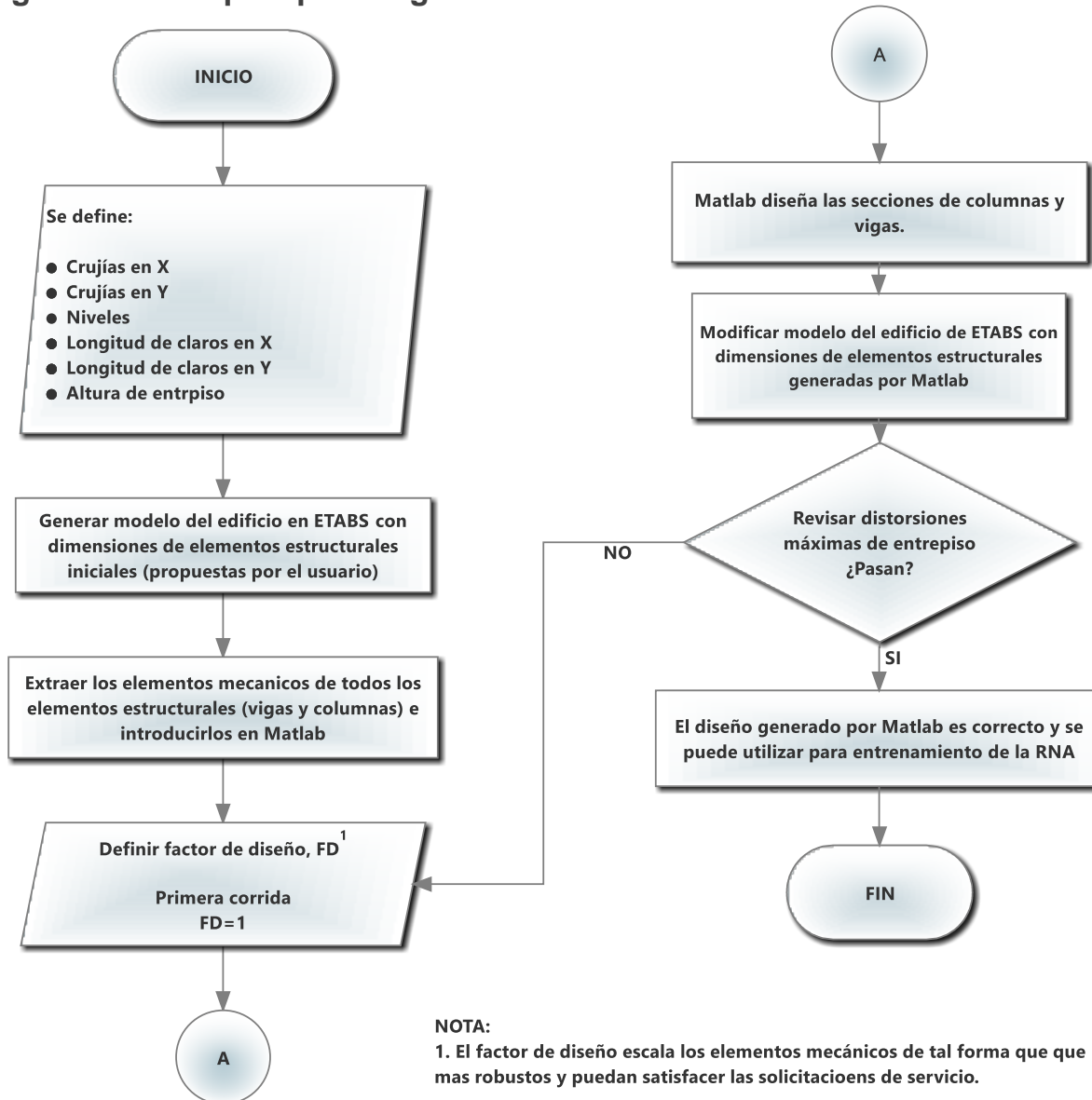
## Referencias

1. Meli R. (2011), "*Diseño estructural*", Editorial Limusa.
2. Bojórquez J. (2011), "*Uso de redes neuronales artificiales para estimar la respuesta sísmica de sistemas estructurales*", tesis de Maestría, Instituto de Ingeniería, UNAM, México.
3. Serrano A., Soria E., y Martín J. (2009-2010), "*Redes neuronales artificiales.*" Escola Técnica Superior d'Enginyeria.
4. Acosta M., Salazar H., y Zuluaga C. (2000), "*Tutorial de redes neuronales*", Universidad Tecnológica de Pereira.
5. Demuth, H., Beale M., y Hagan, M. (1992), "*Neural Network Toolbox™ 6.*" User Guide.
6. Kung, S. (1993) "*Digital neural networks*" Prentice-Hall.
7. Bishop, C. (1996) "*Neural networks for pattern recognition*". Clarendon Press, Oxford.
8. Kennedy, R., Van Roy, B., Reed, C. y Lippman R. (1998) "*Solving data mining problems through pattern recognition*" Prentice-Hall.
9. <http://www.quora.com/Human-Brain/How-is-the-human-brain-so-energy-efficient>, (Mayo 2013).
10. González y Robles (2010), "*Aspectos fundamentales del concreto reforzado*", Editorial Limusa.
11. CSI, SAP2000 V. 8 (2002), "*Integrated finite element analysis and design of structures basic analysis reference manual*", Computers and Structures Inc., Berkeley, California. USA.
12. Diario Oficial de la Federación (1993), "Reglamento de construcciones para el Distrito Federal.", México, DF.
13. Gaceta Oficial del Distrito Federal (2004), "*Normas técnicas complementarias para diseño por sismo*", México, DF.
14. Gaceta Oficial del Distrito Federal (2004), "*Normas técnicas complementarias sobre criterios y acciones para el diseño estructural de las edificaciones*", México, DF.
15. Gaceta Oficial del Distrito Federal (2004), "Normas técnicas complementarias para diseño y construcción de estructuras de concreto", México, DF.
16. Rangel Aguilar G. (2011), "*Influencia en la confiabilidad sísmica estructural de las incertidumbres en las propiedades mecánicas de los materiales y en las cargas vivas máximas*", tesis de Maestría, Instituto de Ingeniería, UNAM, México.
17. Gómez Tremari R., (1982) "*Fundamentos y métodos prácticos de diseño sísmico*", Colegio de Ingenieros Civiles del estado de Jalisco, Guadalajara, México
18. Montgomery C. D. y Runger C. G. (2007), "*Probabilidad y estadística aplicadas a la ingeniería*", McGraw-Hill, México DF.
19. Gomes, Wellison J. de S., y André B. (2012), "*Global structural optimization considering expected consequences of failure and using ANN surrogates*", Computers & Structures.

20. Hurtado J. E. y Alvarez D. A. (2000), "*Reliability assessment of structural systems using neural networks.*" Proc. European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS.
21. Möller O., Foschi R., Rubinstein M. y Quiroz L. (2006), "*Momento-curvatura de secciones de hormigón armado sismorresistentes utilizando redes neuronales.*" Mecánica Computacional 25: 2145-2162.
22. Möller O., Foschi R., Rubinstein M. y Quiroz L. (2008) "*Optimización de pórticos con acciones sísmicas: diferentes estrategias numéricas utilizando redes neuronales.*" Mecánica Computacional 27: 2583-2603.
23. Zhang, J., y Foschi R. (2004), "*Performance-based design and seismic reliability analysis using designed experiments and neural networks*" Probabilistic engineering mechanics 19.3: 259-267.
24. Möller O., Luchesi L., Foschi R. y Rubinstein M. (2005), "*Redes neuronales aplicadas a dinámica de estructuras.*" Mecánica Computacional, AMCA 24: 797-812.
25. Chau, K. (2007), "*Reliability and performance-based design by artificial neural network.*" Advances in Engineering Software 38.3: 145-149.
26. Cardoso J., Almeida J., Dias J. y Coelho P. (2008), "Structural reliability analysis using Monte Carlo simulation and neural networks." *Advances in Engineering Software* 39.6: 505-513.
27. The MathWorks Inc. (2012) "Matlab" R2012b (8.0.0.783)
28. Computers and Structures Inc. (2010) "ETABS" Nonlinear Version 9.7.0

## Anexos

### A) Diagrama de bloques para la generación de modelos de entrenamiento



### B) Diagrama de bloques para la operación de la red en Matlab

