

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



**APLICACIÓN DE REDES NEURONALES
CELULARES A UN PROBLEMA DE INGENIERÍA**

POR

LIC. MANUEL FRANCISCO SAUCEDA SAUCEDA

TESIS

**EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS DE LA
INGENIERÍA ELÉCTRICA CON ESPECIALIDAD EN CONTROL**

CIUDAD UNIVERSITARIA

FEBRERO DE 2011

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



**APLICACIÓN DE REDES NEURONALES
CELULARES A UN PROBLEMA DE INGENIERÍA**

**POR
LIC. MANUEL FRANCISCO SAUCEDA SAUCEDA**

TESIS

**EN OPCION AL GRADO DE MAESTRO EN CIENCIAS DE LA
INGENIERÍA ELÉCTRICA CON ESPECIALIDAD EN CONTROL**

CIUDAD UNIVERSITARIA

FEBRERO DE 2011

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «**Aplicación de redes neuronales a un problema de ingeniería** », realizada por el alumno **Manuel Francisco Saucedá Saucedá**, con número de matrícula 0101430, sea aceptada para su defensa como opción al grado de Maestro en Ciencias de la Ingeniería Eléctrica con especialidad en Control.

El Comité de Tesis



Dr. Oscar Leonel Chacón Mondragón

Asesor



Dr. Ernesto Vázquez Martínez

Revisor



Dr. José Paz Pérez Padrón

Revisor

Vo. Bo.



Dr. Moisés Hinojosa Rivera

División de Estudios de Postgrado

San Nicolás de los Garza, Nuevo León, Enero de 2011

Aplicación de Redes Neuronales Celulares a un Problema de Ingeniería

Alumno: Manuel Francisco Saucedá Saucedá

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica de la U.A.N.L.

División de Estudios de Posgrado.

Director de Tesis: Dr. Oscar L. Chacón Mondragón

19 de enero de 2011

Índice general

1. INTRODUCCIÓN	1
1.1. Objetivos	2
1.2. Aportaciones	3
1.3. Organización de la Tesis	4
2. REDES NEURONALES ARTIFICIALES	5
2.1. Panorama Histórico	6
2.2. Definición de Redes Neuronales Artificiales	10
2.3. Elementos de Redes Neuronales Artificiales	14
2.3.1. Unidades de Proceso: La Neurona Artificial	14
2.3.2. Estado de Activación	15
2.3.3. Función de Salida o de Transformación	17
2.3.4. Regla de Propagación	20
2.3.5. Regla de Aprendizaje	21
2.3.6. Representación Vectorial	21
2.4. Características de las Redes Neuronales	23
2.4.1. Topología de las Redes Neuronales	23
2.4.2. Patrón de Conectividad	24

2.4.3.	Mecanismo de Aprendizaje	26
2.4.4.	Ventajas de las Redes Neuronales	36
3.	REDES NEURONALES CELULARES	37
3.1.	Definición y Características de Redes Neuronales Celulares	38
3.2.	Arquitectura de Redes Neuronales Celulares	39
3.2.1.	Rango Dinámico de Redes Neuronales Celulares	44
3.2.2.	Estabilidad de Redes Neuronales Celulares	45
4.	DISEÑO PARA REDES NEURONALES BASADO EN EL ALGORITMO DEL PERCEPTRÓN	55
4.1.	Antecedentes	57
4.2.	Preliminares	60
4.3.	Un Nuevo Algoritmo de Diseño	62
4.3.1.	Redes Neuronales con Restricciones en los Elementos de la Diagonal de la Matriz de Conexión	68
4.3.2.	Redes Neuronales con Restricciones de Simetría y Escasez Sobre la Estructura de Interconexión	73
5.	DISEÑO Y APLICACIÓN DE UNA RED NEURONAL CELULAR A UN PROBLEMA DE INGENIERÍA	80
5.1.	Diseño de una Red Neuronal Celular Basada en el Uso de Memorias Asociativas Para Aplicarla en Reconocimiento de Patrones	81
5.2.	Aplicación de Redes Neuronales Celulares a Reconocimiento de Patrones.	96
	Referencias Bibliográficas	102

DEDICO ESTE TRABAJO DE TESIS:

A la memoria de mis amados padres Sr. José Saucedo Fuentes y Sra. Victoria Saucedo de Saucedo. Gracias por su amor incondicional y por enseñarme a ver lo positivo de la vida les amaré siempre.

A mi amada esposa Laura Amparo Garza González y a mi querido hijo Jesús Francisco por su amor, motivación, paciencia y tolerancia.

A mis queridas hermanas: M. Amelia e I. Cristina Saucedo Saucedo por su cariño.
A toda mi familia.

Agradecimientos

Deseo agradecer de una manera muy especial a mi asesor de tesis:

Al Dr. Oscar L. Chacón Mondragón por su disposición y apoyo.

A los Profesores Investigadores del grupo de Control Automático y de Ingeniería Eléctrica de F.I.M.E. de la U.A.N.L. por su apoyo.

A los directivos de la Universidad Autónoma de Nuevo León por su apoyo y confiar en mi.

A la M. A. Patricia Martínez Moreno, directora de la Facultad de Ciencias Físico-Matemáticas, por su ayuda y amistad.

A la Lic. Rosa Alicia Gámez, por su amistad e inapreciable ayuda.

Que Dios los bendiga a todos.

Resumen

Esta tesis presenta los siguientes resultados principales:

El diseño de una red neuronal celular basado en memorias asociativas que permite la clasificación de un conjunto de datos, que consisten en medidas de señales. Este diseño combina la teoría de redes neuronales artificiales con retroalimentación y la de redes neuronales celulares. Lo que permite realizar memorias asociativas para reconocer patrones.

Dado que los datos originales no satisfacen las condiciones para ser clasificados por dicha red neuronal, fue necesario encontrar una transformación apropiada del plano complejo para modificarlos, de manera que permita que los datos originales modificados por esta transformación puedan ser clasificados por la red neuronal celular diseñada.

Capítulo 1

INTRODUCCIÓN

Existe actualmente una tendencia a establecer un nuevo campo de las ciencias de la computación que integraría los diferentes métodos de resolución de problemas que no pueden ser descritos fácilmente mediante un enfoque algorítmico tradicional. Estos métodos, de una forma u otra, tienen su origen en la emulación, más o menos inteligente, del comportamiento de los sistemas biológicos. Algunos autores ya han acuñado distintos términos para referirse a este nuevo tipo de computación. Los términos más usados son: Computación cognitiva, Computación del mundo real, Computación suave y Brainware.

Se trata de una nueva forma de computación que es capaz de manejar las imprecisiones e incertidumbres que aparecen cuando se trata de resolver problemas relacionados con el mundo real (reconocimiento de formas, toma de decisiones, etc.), ofreciendo soluciones robustas y de fácil implementación. Para ello dispone de un conjunto de metodologías como son la lógica borrosa, las redes neuronales, los algoritmos genéticos, el razonamiento aproximado, la teoría del caos y la teoría del aprendizaje.

Con las redes neuronales artificiales (una de estas formas de computación) se intenta expresar la solución de problemas complejos, no como una secuencia de pasos, sino como

la evolución de unos sistemas de computación inspirados en el funcionamiento del cerebro humano y dotados, por tanto, de cierta “inteligencia”. Estos sistemas de computación no son sino la combinación de una gran cantidad de elementos simples de proceso (neuronas) interconectados que operando de forma masivamente paralela, consiguen resolver problemas relacionados con el reconocimiento de formas o patrones, predicción, codificación, clasificación, control y optimización. De lo anterior, resulta claro que muchos de los problemas de ingeniería requieren este nuevo tipo de computación, para su adecuada solución, porque tratan con fenómenos complejos no lineales; de ahí que se haya elegido su uso para resolver particularmente nuestro problema.

Se describen y/o se definen en detalle los siguientes términos y conceptos: redes neuronales, redes neuronales celulares, neuronas, sinapsis, axón, aprendizaje, algoritmo de aprendizaje, convergencia de algoritmos y entrenamiento de redes neuronales, así como, su topología (arquitectura) y sus aplicaciones. Se analiza, también, un concepto muy importante en análisis de sistemas que es el de estabilidad. Se propone un diseño para una red neuronal celular basado en una aplicación a memoria asociativa para redes neuronales con retroalimentación, éste, a su vez, se basa en el algoritmo de entrenamiento de perceptrones.

1.1. Objetivos

El objetivo principal de esta tesis es el diseño de una red neuronal celular que permita la clasificación de medidas de señales usando memorias asociativas. Este diseño combina la teoría de redes neuronales artificiales con la metodología de redes neuronales celulares, lo que permite realizar memorias asociativas para reconocer patrones. Para poder aplicar el método propuesto se requiere que los patrones de entrenamiento sean vectores binarios

bipolares y que los datos a clasificar estén asociados a ellos de modo que si X es un vector binario n -dimensional (patrón de entrenamiento con componentes 1 y -1) y Y es un vector n -dimensional que representa a un dato a clasificar, entonces Y está relacionado con X si $Y \in C(X)$. Los datos a clasificar pertenecen a un conjunto de medidas de señales eléctricas de una línea de transmisión de un sistema eléctrico de potencia. Estas señales se encuentran en estado de falla o en estado de oscilación, el problema consiste en tomar la medida de una señal y determinar a que estado de la señal corresponde, es decir, se quiere saber si la señal se encuentra en estado de falla o en estado de oscilación. Para lograrlo se aplicarán técnicas de reconocimiento de patrones por medio de redes neuronales celulares.

1.2. Aportaciones

La principal aportación consiste en el diseño de una red neuronal celular que realiza clasificación de patrones, en este caso, determinar el tipo al que pertenecen las medidas de un conjunto de medidas de señales eléctricas.

En la resolución del problema del diseño de la red neuronal celular se tomó en cuenta la metodología empleada por Derong Liu para redes neuronales con retroalimentación, la teoría de redes neuronales artificiales y la de redes neuronales celulares.

Puesto que las medidas de la señal dada (datos obtenidos), que están representadas como números complejos, no satisfacen los requisitos impuestos por la red diseñada, fué necesario modificarlos para aplicar el método propuesto. Se realizó una transformación del plano complejo Z al plano complejo W que permite que los datos correspondientes a fallas pertenezcan a una de las clases y los correspondientes a oscilación a la otra clase. Finalmente, aplicando la metodología propuesta para entrenar la red neuronal celular, se logró obtener a qué tipo de estado corresponde una medida de señal dada.

En la resolución del problema de determinar o encontrar la transformación apropiada que permita la clasificación de los datos se recurrió a consultar textos de variable compleja, de manera que la aportación consiste en la aplicación de una red neuronal celular diseñada en base a memorias asociativas para clasificar patrones.

1.3. Organización de la Tesis

En el Capítulo 2 se presenta un panorama histórico donde se describe el origen y desarrollo de las redes neuronales artificiales, se describe la estructura de una red neuronal artificial; se mencionan algunas características de las redes neuronales, tales como: la topología, el mecanismo de aprendizaje y la manera en que se asocian las informaciones de entrada y salida de las redes.

En el Capítulo 3 se definen las redes neuronales celulares (RNC), se describen sus fundamentos analíticos y se incluyen algunos de los resultados existentes con relación a ellas.

En el Capítulo 4 se presenta el desarrollo de un nuevo algoritmo de diseño para memorias asociativas basado en el algoritmo de entrenamiento del perceptrón.

En el Capítulo 5 se diseña una red neuronal celular, tomando en cuenta la definición de RNC y las restricciones que deben satisfacerse para la misma [7] y [9]. Luego se utiliza la metodología empleada por Derong Liu y Zanjun Lu [16] que aparece en el capítulo 4 para poder usarla en reconocimiento de patrones para clasificar un conjunto de señales obtenidas de un nodo de una red eléctrica.

Finalmente, se exponen las conclusiones y comentarios sobre los resultados obtenidos al aplicar los diseños de redes neuronales propuestos. También se dan algunas perspectivas generales sobre el trabajo futuro.

Capítulo 2

REDES NEURONALES ARTIFICIALES

En este capítulo se presentan los conceptos básicos necesarios para el estudio y análisis de los temas expuestos en los capítulos siguientes. Se da un panorama general de lo que son las redes neuronales artificiales, su definición, sus componentes, su estructura o topología, su funcionamiento, y algunas clasificaciones y aplicaciones. El capítulo está organizado de la siguiente manera: en la sección 2.1 se da un panorama histórico donde se describe el origen y desarrollo de las redes neuronales artificiales; también se mencionan a los pioneros e investigadores que han contribuido a su creación y desarrollo, así como a su implementación.

En la sección 2.2 se definen las redes neuronales artificiales; en la sección 2.3 se especifican los elementos y componentes de las redes neuronales; en la sección 2.4 se describe la estructura de una red neuronal artificial.

En la sección 2.5 se mencionan algunas características de las redes neuronales, tales como las topologías de las redes neuronales artificiales, el mecanismo de aprendizaje, sus

diferentes tipos y la manera en que se asocian las informaciones de entrada y salida de las redes neuronales; en las secciones 2.6 y 2.7 se mencionan las ventajas y aplicaciones de las redes neuronales artificiales, respectivamente.

2.1. Panorama Histórico

Los trabajos sobre redes neuronales artificiales han sido motivados desde sus inicios reconociendo que el cerebro calcula (y procesa información) en una forma totalmente diferente a los computadores digitales. Los esfuerzos por entender el cerebro se deben mucho a los trabajos de Ramón y Cajal (1911), quien introdujo la idea de que las neuronas son las componentes estructurales del cerebro.

Alan Turing (1936) fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación; sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren Mc Culloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes en 1943, crearon una teoría acerca de las formas de trabajar de las neuronas. Ellos modelaron una red neuronal simple mediante circuitos eléctricos [29].

En 1949 Donald Hebb escribe el libro: La organización del comportamiento, en el que establece una conexión entre Psicología y Fisiología [4]. Por su parte, Frank Rosenblatt comenzó el desarrollo del perceptrón en 1957. El perceptrón es la más antigua red neuronal y se usa de varias formas en la aplicación para reconocer patrones [17].

Bernard Widrow y Marcial Hoff, de Standford desarrollaron en 1959 el modelo ADALINE (ADaptive LINear Elements) [3]. Esta fue la primera red neuronal aplicada a un problema real (filtros adaptivos para eliminar ecos en las líneas telefónicas) [32].

Uno de los principales investigadores de redes neuronales desde los años sesentas hasta

nuestros días es Stephen Grossberg (Universidad de Boston). A partir de su conocimiento fisiológico, estudió los mecanismos de la percepción y de la memoria. Ha escrito numerosos libros y en 1967 realizó una red Avalancha para realizar actividades como reconocimiento del habla y aprendizaje del movimiento de los brazos de un robot [28].

En 1969 surgieron numerosas críticas que frenaron, hasta 1982, el crecimiento que estaban experimentando las investigaciones sobre redes neuronales. Marvin Minsky y Seymour Papert, del Instituto Tecnológico de Massachusetts publicaron un libro, *Perceptrons*, que además de contener un análisis matemático detallado del perceptrón, consideraban que la extensión a perceptrones multinivel era completamente estéril; las limitaciones eran importantes, muchas investigaciones giraron hacia la inteligencia artificial [30].

A pesar de esto, algunos investigadores continuaron sus investigaciones. James Anderson, desarrolló en 1977 un Asociador Lineal que consistía en unos elementos integradores lineales (Neuronas) que sumaban sus entradas. Posteriormente diseñó una extensión del Asociador lineal, llamada Brain-State-in-a-Box [33]. También, en 1977 Theuvo Kohonen desarrolló un modelo similar al de Anderson, pero independientemente [23]. En 1980 Kunihiko Fukushima desarrolló el Neocognitrón, un modelo de red neuronal para el reconocimiento de patrones visuales.

En 1982 coincidieron numerosos eventos que hicieron resurgir el interés por las redes neuronales. Por un lado, John Hopfield presentó su trabajo sobre redes neuronales en la Academia Nacional de Ciencias; en éste, describe con claridad y rigor matemático una red a la que ha dado su nombre, que es una variación del Asociador Lineal, pero, además, mostró cómo tales redes pueden trabajar y qué pueden hacer. Por otro lado, se celebró la US-Japan Joint Conference on Cooperative/ Competitive Neural Networks y Fujitsu comenzó el desarrollo en computadoras pensantes para aplicaciones en robótica

[26].

Posteriormente, en 1985, el Instituto Americano de Física comenzó lo que ha sido la reunión anual Neural Network for Computing. En 1987, el IEEE celebró la primera conferencia internacional sobre redes neuronales. En el mismo año se formó la International Neural Network Society (INNS) bajo la iniciativa de Grossberg en E.U. Kohonen en Finlandia y Amari en Japón. En 1988, de la unión de la IEEE y la INNS surgió la International Joint Confernce on Neural Network (IJCNN) que produjo, en 1989, 430 artículos, 63 de los cuales enfocados a una aplicación. En 1990 incluyó una hora de concierto de música realizada por redes neuronales. También merece una referencia especial la reunión anual Neural Information Procesing Systems (NCPS) celebrada en Denver, Colorado desde 1987, y que probablemente represente el nivel más alto de calidad desde el punto de vista científico.

Después de esto, se han desarrollado otros tipos de redes neuronales y algoritmos de entrenamiento para nuevas aplicaciones. En seguida se presentan algunos casos relacionados con el perceptrón y sus generalizaciones.

Los perceptrones universales (PU) son una generalización del perceptrón de Rosenblatt en los cuales se pueden implementar funciones Booleanas. Para implementar estas funciones los PU toman diferentes clases de estructuras topológicas simples en las que cada una de ellas contiene a lo más una capa oculta con una cantidad mínima de neuronas [6].

Otra aplicación inspirada por el concepto de sistema de secuencias del ADN en biología, ha desarrollado un nuevo algoritmo llamado aprendizaje como el del ADN, el cual es capaz de entrenar rápidamente una red neuronal con unas funciones booleanas preescritas. Este algoritmo tiene algunas ventajas tales como: corridas rápidas y robustez. Además no necesita la propiedad de convergencia.

Por otra parte, los procesos mentales como la percepción, cognición, instintos, emociones y habilidades cognitivas superiores, necesarias para lograr un pensamiento abstracto, son considerados como paradigmas de modelos neuronales.

Los mecanismos matemáticos fundamentales para tratar con procesos, que van de lo vago o difuso a lo concreto o categórico, son llamados lógica dinámica y relaciona a estos con una descripción psicológica de la mente. Dichos procesos corresponden al funcionamiento mental de entendimiento Aristotélico [10].

Recientemente, mediciones obtenidas por imágenes de resonancia magnética del funcionamiento de la imaginación confirman que este proceso es un mecanismo de percepción neuronal.

En lo que toca a reconocimiento de patrones, la tarea más difícil es, quizás, el reconocimiento de patrones dinámicos. Basándose en la teoría de aprendizaje determinístico se propone el siguiente procedimiento para lograrlo: 1) los patrones dinámicos variantes en el tiempo pueden representarse como invariantes en el tiempo y espacialmente distribuidos de manera que el aprendizaje determinístico se realice; 2) se da una definición para características similares de patrones dinámicos basada en sistemas dinámicos; 3) se da un mecanismo de reconocimiento rápido de patrones dinámicos por el cual un patrón dinámico de prueba es reconocido como similar a un patrón dinámico de entrenamiento si el estado de sincronización se logra de acuerdo a una clase de paridad interna en el sistema dinámico. Los errores de sincronización pueden ser considerados como la medida de similaridad entre los patrones de entrenamiento y los de prueba [14].

Sin embargo, también se sabe que el comportamiento dinámico es muy lento para perceptrones multicapa, quedando frecuentemente atrapados por el fenómeno 'plateau' debido a las singularidades en el espacio de parámetros de los perceptrones, en el que se trazan las trayectorias de aprendizaje [25].

En lo que se refiere a la convergencia de los pesos del perceptrón se tiene que si existe un conjunto no vacío de pesos iniciales tales que el conjunto de los pesos del perceptrón están acotados, entonces los pesos del perceptrón están acotados para todos los pesos iniciales [20].

Por último, se consideran dos aspectos relacionados con dos decisiones: los criterios para determinar cuando parar la red y los reentrenamientos de la red antes de calcular la elevación de costos.

Los resultados experimentales indican que el incremento de los costos computacionales asociados con el reentrenamiento de la red con algunas características removidas temporalmente antes de calcular la elevación de costos es recompensada con una mejora significativa [66].

2.2. Definición de Redes Neuronales Artificiales

El Modelo Biológico

La teoría y modelado de redes neuronales artificiales está inspirada en la estructura y funcionamiento de los sistemas nerviosos, donde la neurona es el elemento fundamental. El cerebro tiene una gran estructura y la habilidad de construir sus propias reglas a través de lo que usualmente llamamos experiencia. Sin duda, la experiencia se construye con los años. La etapa más drástica del desarrollo del cerebro humano, tiene lugar en los primeros dos años de su vida; pero el desarrollo continúa más allá de esa etapa. En esta primera etapa, se forman cerca de un millón de sinapsis por segundo. Las **sinapsis** son estructuras elementales y unidades funcionales que median las interacciones entre neuronas. La clase más común de sinapsis es una sinapsis química, que opera como sigue: un proceso sináptico libera una substancia transmisora que se distribuye cruzando las

uniones sinápticas entre neuronas y entonces activa un proceso postsináptico. Así, una sinapsis convierte una señal eléctrica presináptica en una señal química y la devuelve en una señal eléctrica postsináptica. En descripciones tradicionales de redes neuronales se asume que una sinapsis es una conexión simple que impone excitación o inhibición, pero, no ambas sobre la neurona receptiva.

Existen neuronas de diferentes formas, tamaños y longitudes. Estos atributos son importantes para determinar la función y utilidad de la neurona. La clasificación de estas células en tipos normalizados ha sido realizada por muchos anatomistas.

Una **neurona** es una célula viva y como tal, contiene los mismos elementos que forman parte de todas las células biológicas; además, contienen elementos característicos que las diferencian. En general, una neurona consta de un cuerpo celular más o menos esférico, de 5 a 10 micras de diámetro, del que salen una rama principal, el **axón**, y varias ramas más cortas, llamadas **dendritas** [5]. A su vez, el axón puede producir ramas en torno a su punto de arranque y con frecuencia se ramifica extensamente cerca de su extremo (ver figura 1).

Una de las características que diferencian a las neuronas del resto de las células vivas, es su capacidad de comunicarse. En términos generales, las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular las combina e integra y emite señales de salida. El axón transporta esas señales a las terminales axonómicas, que se encargan de distribuir información a un nuevo conjunto de neuronas. Por lo general, una neurona recibe información de miles de otras neuronas y a su vez, envía información a miles de neuronas más. Se calcula que en el cerebro humano existen del orden de 10^{15} conexiones [31].

Casi todas las neuronas reciben entradas procedentes de sinapsis excitadoras e inhibitoras; en cada instante, algunas de ellas estarán activas y otras estarán en reposo. La

suma de los efectos excitadores e inhibidores determina si la célula será o no estimulada, es decir, si emitirá o no un tren de impulsos y a qué velocidad.

El desarrollo neuronal es sinónimo de cerebro plástico: la **plasticidad** permite el desarrollo del sistema nervioso para adaptarse a su ambiente circundante. En un cerebro adulto la plasticidad se consigue por dos mecanismos: la creación de nuevas conexiones sinápticas entre neuronas y la modificación de las sinapsis existentes. Los axones (líneas de transmisión) y las dendritas (zonas receptoras) constituyen dos tipos de filamentos de celdas que pueden distinguirse sobre una base morfológica: un axón tiene una superficie suave, pocas ramas y gran longitud, mientras que una dendrita tiene una superficie irregular y más ramas [1] (ver figura 1).

Para establecer una similitud directa entre la actividad sináptica y la analogía con las redes neuronales artificiales, vamos a fijar los siguientes aspectos: las señales que llegan a las sinapsis son las **entradas** a la neurona; éstas son ponderadas (atenuadas o amplificadas) a través de un parámetro, denominado peso, asociado a la sinapsis correspondiente. Estas señales de entrada pueden excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo). El efecto es la suma de las entradas ponderadas. Si la suma es mayor o igual que el umbral de la neurona, entonces la neurona se activa. Esta es una situación todo o nada; cada neurona se activa o no se activa, la facilidad de transmisión de señales se altera mediante la actividad del sistema nervioso. Las sinapsis son susceptibles a la fatiga, deficiencia de oxígeno y algunos agentes tales como los analgésicos, otros eventos pueden incrementar el grado de activación [31]. Esta habilidad de ajustar señales es un **mecanismo de aprendizaje**. La plasticidad parece ser esencial en el funcionamiento de las neuronas como unidades de procesamiento de información en el cerebro humano, así también, lo es con redes neuronales de neuronas artificiales. En su forma más general, una red neuronal artificial es una máquina que está diseñada para modelar la forma en

que el cerebro realiza una tarea o función de interés; la red usualmente se implementa usando componentes electrónicos o simulada en un computador digital.

Una red neuronal artificial se puede definir de distintas maneras, algunas de estas definiciones son las siguientes:

- Una nueva forma de computación, inspirada en modelos biológicos.
- Un sistema de computación hecho por un gran número de elementos de proceso simples, muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas [34].
- Son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptables) y con organización jerárquica; las cuales, intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico [24].

Se puede dar la siguiente definición de red neuronal vista como una máquina adaptable.

Definición 2.1

Una **red neuronal artificial** es un procesador distribuido masivamente en paralelo, que almacena conocimiento de la experiencia haciéndolo disponible para usarlo.

Esta se asemeja al cerebro en dos aspectos:

- 1) El conocimiento es adquirido por la red a través de un proceso de aprendizaje.
- 2) Existen intensas conexiones entre neuronas conocidas como pesos sinápticos que son usadas para almacenar el conocimiento.

El procedimiento usado para realizar el proceso de aprendizaje se llama **algoritmo de aprendizaje**, cuya función es modificar los pesos sinápticos de la red de una manera ordenada, así como lograr el objetivo diseñado.

2.3. Elementos de Redes Neuronales Artificiales

En seguida se describen los componentes más importantes de una red neuronal artificial los cuales son:

- Unidades de procesamiento (la neurona artificial)
- Estado de activación de cada neurona
- Función de transformación o de salida
- Regla de propagación
- Regla de aprendizaje
- Patrón de conectividad entre neuronas

2.3.1. Unidades de Proceso: La Neurona Artificial

Cualquier modelo de red neuronal consta de dispositivos elementales de proceso: las **neuronas**. A partir de ellas, se pueden generar representaciones específicas, de tal forma que un estado conjunto de ellas puede significar una letra, un número o cualquier otro objeto.

En cualquier sistema que se esté modelando, es útil caracterizar tres tipos de **unidades**: entradas, salidas y unidades ocultas. Las **unidades de entrada** reciben señales desde el entorno; estas entradas (que son a la vez entradas a la red) pueden ser señales provenientes de sensores que tomarán la información de entrada. Dicha información se transmite a ciertos elementos internos que se ocupan de su procesado. Es en las sinapsis y neuronas correspondientes a este segundo nivel donde se genera cualquier tipo de representación interna de la información. Puesto que no tienen relación directa

con la información de entrada ni con la de salida, estos elementos se denominan **unidades ocultas**, las unidades ocultas son aquellas entradas y salidas que se encuentran dentro del sistema, es decir, no tienen contacto con el exterior. Una vez finalizado el período de procesado, la información llega a las **unidades de salida**, cuya misión es dar la respuesta del sistema; estas señales pueden controlar directamente otros sistemas.

2.3.2. Estado de Activación

Adicionalmente al conjunto de unidades, la representación necesita los **estados** del sistema en un tiempo t . Esto se especifica por un vector de N números reales $A(t)$, que representa el **estado de activación** del conjunto de unidades de procesamiento. Cada elemento del vector representa la activación de una unidad en el tiempo t . La **activación** de una unidad U_i , en el tiempo t se designa por $a_i(t)$; es decir:

$$A(t) = (a_1(t), a_2(t), \dots, a_i(t), \dots, a_N(t))$$

El procesamiento que realiza la red se ve como la evolución de un patrón de activación en el conjunto de unidades que lo componen a través del tiempo.

Una función de activación, F , determina el nuevo **estado de activación** $a_i(t + 1)$ de la neurona, teniendo en cuenta la entrada total calculada y el anterior estado de activación $a_i(t)$; asociado a cada unidad, existe una función de salida, f_i , que transforma el estado actual de activación en una señal de salida, y_i . Dicha señal es enviada a través de los canales de comunicación unidireccionales a otras unidades de la red; en estos canales la señal se modifica de acuerdo con la sinapsis (el peso, w_j) asociada a cada uno de ellos según una determinada regla. Las señales moduladas que han llegado a la unidad j -ésima se combinan entre ellas, generando así la entrada total, $Net_j = \sum_{i=1}^n w_{ji} * y_i$.

La dinámica que rige la actualización de los estados de las unidades (evolución de

la red) puede ser de dos tipos: modo asíncrono y modo síncrono. En el primer caso, las neuronas evalúan su estado continuamente, según les va llegando información y lo hacen de forma independiente. En el caso síncrono, la información también llega de forma continua, pero los cambios se realizan simultáneamente, como si existiera un reloj interno que decidiera cuándo deben cambiar su estado. Los sistemas biológicos quedan probablemente entre ambas posibilidades. Si se tienen N unidades (neuronas), podemos ordenarlas arbitrariamente y designar la j -ésima unidad como U_j . Su trabajo es simple y único y consiste en recibir las entradas de las células vecinas y calcular un valor de salida, el cual es enviado a todas las células restantes.

Todas las neuronas que componen la red se hallan en cierto estado. En una visión simplificada, podemos decir que hay dos posibles estados, reposo y excitado, a los que denominaremos **estados de activación**, y a cada uno de los cuales se le asigna un valor. Los valores de activación pueden ser continuos o discretos. Además, pueden ser limitados o ilimitados. Si son discretos, suelen tomar un conjunto pequeño de valores o bien valores binarios. En notación binaria, un estado activo se indicaría por un 1, y se caracteriza por la emisión de un impulso por parte de la neurona (potencial de acción), mientras que un estado pasivo se indicaría por un 0 y significaría que la neurona está en reposo. En otros modelos se considera un conjunto continuo de estados de activación, en lugar de sólo dos estados, en cuyo caso se les asigna un valor entre $[0,1]$ o en el intervalo $[-1,1]$, generalmente siguiendo una función sigmoïdal.

2.3.3. Función de Salida o de Transformación

Entre las unidades o neuronas que forman una red neural artificial existe un conjunto de conexiones que unen unas a otras. Cada unidad transmite señales a aquellas que están conectadas con su salida. Asociada con cada unidad U_i hay una función de salida $f_i(a_i(t))$, que transforma el estado actual de activación $a_i(t)$, en una señal de salida $y_i(t)$, es decir:

$$y_i(t) = f_i(a_i(t))$$

El vector que contiene las salidas de todas las neuronas en un instante t es:

$$Y(t) = (f_1(a_1(t)), f_2(a_2(t)), \dots, f_i(a_i(t)), \dots, f_N(a_N(t)))$$

En algunos modelos, esta salida es igual al nivel de activación de la unidad, en cuyo caso la función f_i es la función identidad, $f_i(a_i(t)) = a_i(t)$. A menudo, f_i es de tipo sigmoidal, y suele ser la misma para todas las unidades.

Existen cuatro **funciones de transformación** típicas que determinan distintos tipos de neuronas:

- Función escalón
- Función lineal y mixta
- Función Sigmoidal
- Función gaussiana

Las funciones mixta y sigmoidal son las más apropiadas cuando queremos como salida información analógica. Veamos con más detalle las distintas funciones.

Función escalón

La función de transformación escalón dada por la siguiente ecuación:

$$\mathbf{f}(t) = \left\{ \begin{array}{l} 1, \text{ si } t > 0 \\ 0, \text{ si } t \leq 0 \end{array} \right\}$$

La **función escalón** o umbral únicamente se utiliza cuando las salidas de la red son binarias (dos posibles valores) (ver figura 2). La salida de una neurona se activa (la activación es 1), sólo cuando el estado de activación es mayor o igual que cierto valor umbral θ_i (la función puede estar desplazada sobre los ejes); si es menor, la activación es 0 o -1. Por otro lado, las redes formadas por este tipo de neuronas son fáciles de implementar en hardware, pero a menudo sus capacidades están limitadas.

Funciones lineal y mixta

La **función lineal** o identidad equivale a no aplicar función de salida, se usa muy poco. La función lineal o identidad responde a la expresión $f(t) = t$ y su gráfica se muestra en la figura 3.

La **función mixta** (escalón y lineal) dada por:

$$\mathbf{f}(t) = \left\{ \begin{array}{l} 1, \text{ si } t > c \\ t, \text{ si } -c \leq t \leq c \\ -1, \text{ si } t < -c \end{array} \right\}$$

toma los siguientes valores, la activación se define como 0 (o -1) si la suma de las señales de entrada es menor que un límite inferior. Si dicha suma es mayor o igual que el límite superior, entonces la activación es 1. Si la suma de entrada está comprendida entre ambos límites, superior e inferior, entonces la activación se define como una función lineal de la suma de las señales de entrada. La representación de la función de activación aparece en la figura 4.

La función sigmoïdal

Cualquier función definida simplemente en un intervalo de posibles valores de entrada que puede ser $(-\infty, \infty)$, con un incremento monotónico y que tenga ambos límites superiores e inferiores (por ejemplo, las funciones sigmoïdal o arcotangente), podrá realizar la función de activación o de transformación de forma satisfactoria. Con **la función sigmoïdal**, para la mayoría de los valores del estímulo de entrada (variable independiente), el valor dado por la función es cercano a uno de los valores asintóticos. Esto hace que en la mayoría de los casos, el valor de salida esté comprendido en la zona alta o baja del sigmoïde cuya ecuación es:

$$f(t) = \frac{1}{1 + e^{-t}}$$

De hecho, cuando la pendiente es elevada, esta función tiende a la función escalón. Sin embargo, la importancia de la función sigmoïdal (o cualquier otra función similar) es que su derivada es siempre positiva y cercana a cero para los valores grandes positivos o negativos; además, toma su valor máximo cuando t es 0. Esto hace que se puedan utilizar las reglas de aprendizaje definidas para las funciones escalón, con la ventaja, respecto a esta función, de que la derivada está definida en todo el intervalo (dominio), mientras que, en la función escalón la derivada en el punto de transición no existe y esto no ayuda a los métodos de aprendizaje en los cuales se usan derivadas. La gráfica de esta función se muestra en la figura 5.

Función de transformación gaussiana

En las funciones gaussianas los centros y anchuras de las gráficas de estas funciones pueden ser ajustados, lo cual las hace más adaptables que las funciones sigmoïdales. Por ejemplo, ciertos mapeos que suelen requerir dos niveles ocultos (neuronas en la red que se

encuentran entre las de entrada y las de salida) cuando se utilizan neuronas con funciones de transformación sigmoideas; algunas veces se pueden realizar con un solo nivel en redes con neuronas de función gaussiana, la cual está definida como:

$$f(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(t-\mu)^2/2\sigma^2}$$

donde μ es la media y σ^2 es la varianza de la distribución normal (gaussiana). La gráfica aparece en la figura 6.

2.3.4. Regla de Propagación

Una neurona recibe un conjunto de señales que le dan información del estado de activación de todas las neuronas con las que se encuentra conectada. Cada conexión (sinapsis) entre la neurona i y la neurona j está ponderada por un peso W_{ji} . Normalmente, como simplificación, se considera que el efecto de cada señal es aditivo, de tal forma que la entrada neta que recibe una neurona (potencial postsináptico) Net_j es la suma del producto de cada señal individual por el valor de la sinapsis que conecta ambas neuronas.

Esta regla muestra el procedimiento a seguir para combinar los valores de entrada a una unidad con los pesos de las conexiones que llegan a esa unidad y es conocida como **regla de propagación**.

Suele utilizarse una matriz W con todos los pesos w_{ji} que reflejan la influencia que sobre la neurona j tiene la neurona i . W es un conjunto de elementos positivos, negativos o nulos. Si w_{ji} , es positivo, indica que la interacción entre las neuronas i y j es excitadora; es decir, siempre que la neurona i esté activada, la neurona j recibirá una señal de la neurona i que tenderá a activarla. Si w_{ji} , es negativo, la sinapsis será inhibitoria. En este caso, si i está activada, enviará una señal a j que tenderá a desactivar a ésta. Finalmente,

si $w_{ji} = 0$, se supone que no hay conexión entre ambas.

2.3.5. Regla de Aprendizaje

Existen muchas definiciones del **aprendizaje**, una de ellas es: la modificación del comportamiento inducido por la interacción con el entorno y como resultado de experiencias conducente al establecimiento de nuevos modelos de respuesta a estímulos externos. Esta definición fue enunciada muchos años antes de que surgieran las redes neuronales; sin embargo, puede ser aplicada también a los procesos de aprendizaje de estos sistemas.

En el caso de las redes neuronales artificiales, se puede considerar que el conocimiento se encuentra representado en los pesos de las conexiones entre neuronas, todo **proceso de aprendizaje** implica cierto número de cambios en estas conexiones. Al igual que el funcionamiento de una red depende del número de neuronas de las que disponga y de cómo estén conectadas entre sí, cada modelo dispone de su o sus propias técnicas de aprendizaje.

Por último, falta considerar lo relativo al patrón de conectividad entre neuronas que, por conveniencia, será tratado en la siguiente sección.

2.3.6. Representación Vectorial

En ciertos modelos de redes neuronales, se utiliza la **forma vectorial** como herramienta de representación de algunas magnitudes. Si consideramos una red formada por varias capas de neuronas idénticas, se pueden considerar las salidas de cierta capa de n unidades como un vector n -dimensional

$$Y = (y_1, y_2, \dots, y_n)$$

Si este vector n -dimensional de salida representa los valores de entrada de todas las unidades de una capa m -dimensional, cada una de las unidades de esta capa poseerán pesos asociados a las conexiones procedentes de la capa anterior. Por tanto, hay m vectores de pesos n -dimensionales asociados a la capa m .

El vector de pesos de la j -ésima unidad tendrá la forma:

$$y_j = (y_{j1}, y_{j2}, \dots, y_{jn})$$

La entrada neta de la j -ésima unidad se puede escribir en forma de producto escalar del vector de entradas por el vector de pesos. Cuando los vectores tienen igual dimensión, este producto se define como la suma de los productos de los componentes correspondientes a ambos vectores:

$$Net_j = \sum_{i=1}^N w_{ji} * y_i$$

en donde N representa el número de conexiones de la j -ésima unidad. La ventaja de la notación vectorial, es que la anterior ecuación se puede escribir de la forma:

$$Net_j = W * Y$$

Según esta notación, utilizaremos vectores de entrada, vectores de salida y vectores de peso, aunque hay que tener presente que se reserva la notación vectorial para los casos en los que resulta especialmente adecuada.

2.4. Características de las Redes Neuronales

Existen cuatro aspectos que caracterizan una red neuronal: su topología, tipo de asociación realizada entre la información de entrada y de salida (patrón de conectividad), el mecanismo de aprendizaje, y por último, la forma de representación de estas informaciones.

2.4.1. Topología de las Redes Neuronales

Como se adelantó en el apartado 2.3, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

Cuando se realiza una clasificación de las redes en términos topológicos, se suele distinguir entre las redes con una sola capa o nivel de neuronas y las redes con múltiples capas (2, 3, etc.). En los siguientes apartados se analizarán ambos tipos de redes.

Se conoce como **capa** a un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino (que puede ser, en ambos casos, otra capa de neuronas) [31].

Redes monocapa

En las redes monocapa, como la red de Hopfield y la red Brain-State-In-A-Box, se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red. También, pueden existir conexiones autorrecurrentes (salida de una neurona conectada a su propia entrada), aunque en algún modelo como el de Hopfield, esta recurrencia no se utiliza.

Redes multicapa

Las redes multicapa son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas (2, 3, etc.). En estos casos, una forma para distinguir la capa a la que pertenece una neurona, consistiría en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Normalmente, todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a las entrada de la red y envían las señales de salida a una capa posterior, más cercana a la salida de la red. A estas conexiones se les denomina **conexiones hacia adelante** un ejemplo se muestra en la figura 7.

Sin embargo, en un gran número de estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de las capas anteriores, a estas conexiones se les denomina **conexiones hacia atrás**.

Estas dos posibilidades permiten distinguir entre dos tipos de redes con múltiples capas: las redes con conexiones hacia adelante y las redes que disponen de conexiones tanto hacia adelante como hacia atrás.

2.4.2. Patrón de Conectividad

Redes con conexiones hacia adelante

En estas redes, todas las señales neuronales se propagan hacia adelante a través de las capas de la red. No existen conexiones hacia atrás (ninguna salida de neuronas de una capa i se aplica a la entrada de neuronas de capas $i-1, i-2, \dots$), y normalmente tampoco autorrecurrentes (salida de una neurona aplicada a su propia entrada), ni laterales (salida de una neurona aplicada a la entrada de neuronas de la misma capa), excepto en el caso de los modelos de red propuestos por Kohonen denominados **Learning Vector**

Quantizer (LVQ) y **Topology Preserving Map** (TPM), en las que existen unas conexiones implícitas muy particulares entre las neuronas de la capa de salida.

Las redes más conocidas son: **Perceptrón**, **Adaline** y **Madaline**. Todas ellas son especialmente útiles en aplicaciones de reconocimiento o clasificación de patrones [2].

Redes con conexiones hacia adelante y hacia atrás

En este tipo de redes circula información tanto hacia adelante (forward) como hacia atrás (backward) durante el funcionamiento de la red. Para que esto sea posible, existen conexiones hacia adelante y conexiones hacia atrás entre las neuronas.

En general, excepto el caso particular de las redes Cognitrón y Neocognitrón, suelen ser bicapa (dos capas), existiendo por tanto dos conjuntos de pesos: los correspondientes a las conexiones hacia adelante de la primera capa (capa de entrada) hacia la segunda (capa de salida) y los de las conexiones hacia atrás de la segunda a la primera. Los valores de los pesos de estos dos tipos de conexiones no tienen por qué coincidir, siendo diferentes en la mayor parte de los casos.

Este tipo de estructura (bicapa) es particularmente adecuada para realizar una asociación de una información o patrón de entrada (en la primera capa) con otra información o patrón de salida en la segunda capa (lo cual se conoce como heteroasociación), aunque también pueden ser utilizadas para la clasificación de patrones.

Algunas redes de este tipo tienen un funcionamiento basado en lo que se conoce como resonancia, de tal forma que las informaciones en la primera y segunda capas interactúan entre sí hasta que alcanzan un estado estable. Este funcionamiento permite un mejor acceso a las informaciones almacenadas en la red.

Los dos modelos de red hacia adelante y hacia atrás de dos capas más conocidos son la red **ART** (Adaptive Resonance Theory) y la red **BAM** (Bidirectional Associative Memory). También en este grupo de redes existen algunas que tienen conexiones laterales entre

neuronas de la misma capa. Estas conexiones se diseñan como conexiones excitadoras (con peso positivo), permitiendo la cooperación entre neuronas, o como inhibidoras (con peso negativo), estableciéndose una competición entre las neuronas correspondientes. Una red de este tipo que, además, dispone de conexiones autorrecurrentes es la denominada **CABAM** (Competitive Adaptive Bidirectional Associative Memory).

Finalmente, hay que comentar la existencia de un tipo de red feedforward / feed-back multicapa muy particular, denominada **Neocognitrón**, en la que las neuronas se disponen en planos superpuestos (capas bidimensionales), lo cual permite que puedan eliminarse las variaciones geométricas (tamaños, giros, desplazamientos) o distorsiones que presenten las informaciones o patrones de entrada a la red.

2.4.3. Mecanismo de Aprendizaje

El **aprendizaje** es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas. En los sistemas biológicos existe una continua creación y destrucción de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero; de la misma forma, una conexión se destruye cuando su peso pasa a ser cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables.

$$\frac{dW_{ji}}{dt} = 0$$

Un aspecto importante respecto al aprendizaje en las redes neuronales es el conocer cómo se modifican los valores de los pesos; es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información.

Estos criterios determinan lo que se conoce como la regla de aprendizaje de la red. De forma general, se suelen considerar dos tipos de reglas: las que responden a lo que habitualmente se conoce como aprendizaje supervisado, y las correspondientes a un aprendizaje no supervisado.

La diferencia fundamental entre ambos tipos estriba en la existencia o no de un agente externo (supervisor) que controle el proceso de aprendizaje de la red. Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual o si el aprendizaje supone la desconexión de la red, es decir, su inhabilitación hasta que el proceso termine. En el primer caso, se trataría de un aprendizaje en línea, mientras que el segundo es lo que se conoce como aprendizaje fuera de línea. Cuando el aprendizaje es fuera de línea, se distingue entre una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de test o prueba que serán utilizados en la correspondiente fase. En las redes con aprendizaje fuera de línea, los pesos de las conexiones permanecen fijos después que termina la etapa de entrenamiento de la red. Debido precisamente a su carácter estático, estos sistemas no presentan problemas de estabilidad en su funcionamiento. En las redes con aprendizaje en línea no se distingue entre fase de entrenamiento y de operación, de tal forma que los pesos varían dinámicamente siempre que se presente una nueva

información al sistema. En este tipo de redes, debido al carácter dinámico de las mismas, el estudio de la estabilidad suele ser un aspecto fundamental de estudio.

Redes con Aprendizaje Supervisado

El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor comprueba la salida de la red y en el caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada.

En este tipo de aprendizaje se suelen considerar, a su vez, tres formas de llevarlo a cabo que dan lugar a los siguientes aprendizajes supervisados:

- Aprendizaje por corrección de error
- Aprendizaje por refuerzo
- Aprendizaje estocástico

Aprendizaje por Corrección de Error Este aprendizaje consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error cometido en la salida.

Una regla o **algoritmo simple de aprendizaje por corrección de error** podría ser el siguiente:

$$\Delta W_{ji} = \rho * y_i(d_j - y_j)$$

Siendo:

ΔW_{ji} : variación en el peso de la conexión entre neuronas i y j .

$\Delta W_{ji} = W_{ji}$ actual $- W_{ji}$ anterior.

y_i : valor de la salida de la neurona i .

d_j : valor de la salida deseado para la neurona j .

ρ : factor de aprendizaje ($0 \leq \rho \leq 1$) que regula la velocidad del aprendizaje.

Un ejemplo de este tipo de algoritmo lo constituye la regla de aprendizaje del **Perceptrón**, utilizada en el entrenamiento de la red del mismo nombre que desarrolló Rosenblatt en 1958 [17]. Sin embargo, existen otros algoritmos más evolucionados que éste, que presenta algunas limitaciones, como el no considerar la magnitud del error global cometido durante el proceso completo de aprendizaje de la red, considerando únicamente los errores individuales (locales) correspondientes al aprendizaje de cada información por separado.

Un algoritmo muy conocido que mejora el del Perceptrón y permite un aprendizaje más rápido y un campo de aplicación más amplio es el propuesto por Widrow y Hoff en 1960, denominado regla delta o regla del mínimo error cuadrado (Least-Mean-Squared Error), también conocida como regla de Widrow-Hoff. El cual se aplicó en las redes desarrolladas por los mismos autores, conocidas como **Adaline**, con una única neurona de salida, y **Madaline** (Multiple Adaline), con varias neuronas de salida.

Widrow y Hoff definieron una función que permitía cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanta más información se tenga sobre el error cometido, más rápido se puede aprender.

Este error global se expresa de la siguiente forma:

$$Error_{global} = \frac{1}{2p} \sum_{k=1}^p \sum_{j=1}^N (y_j^k - d_j^k)^2$$

Siendo:

N : Número de neuronas de salida (en el caso de ADALINE $N = 1$).

p : Número de informaciones que debe aprender la red.

$\frac{1}{2} \sum_{j=1}^N (y_j^k - d_j^k)^2$: El error cometido en el aprendizaje de la información k -ésima.

Por tanto, de lo que se trata es de encontrar unos pesos para las conexiones de la red que minimicen esta función de error. Para ello, el ajuste de los pesos de las conexiones de la red se puede hacer de forma proporcional a la variación relativa del error que se obtiene al variar el peso correspondiente:

$$\Delta w_{ji} = k \frac{\partial(\text{Error}_{global})}{\partial(w_{ji})}$$

Mediante este procedimiento, se llegan a obtener un conjunto de pesos con los que se consigue minimizar el error medio.

Otro algoritmo de aprendizaje por corrección de error lo constituye el denominado **regla delta generalizada** o **algoritmo de retropropagación del error** (error back-propagation), se trata de una generalización de la regla delta para aplicarla a redes con conexiones hacia adelante (feedforward) con capas o niveles internos u ocultos de neuronas que no tienen relación con el exterior. Son redes con capa de entrada, capas ocultas y capa de salida.

Estas redes multicapa pueden utilizarse en muchas más aplicaciones que las ya conocidas (Perceptrón, Adaline y Madaline), pero su proceso de aprendizaje es mucho más lento, debido a que durante el mismo se debe explorar el espacio de posibles formas de utilización de las neuronas de las capas ocultas, es decir, se debe establecer cuál va a ser su papel en el funcionamiento de la red.

Las bases de este nuevo método de aprendizaje (al que Rumelhart, Hinton y Williams llamaron backpropagation) fueron sentadas, por diferentes investigadores [40, 38, 36, 32]

que propusieron soluciones al problema del entrenamiento de redes multicapa de forma independiente y sin conocimiento de la existencia de otros trabajos paralelos.

Existe, también, una versión recurrente del algoritmo backpropagation que se suele utilizar en redes multicapa que presentan conexiones recurrentes con el fin de que estas redes aprendan la naturaleza temporal de algunos datos [46, 47, 45].

Para concluir con los algoritmos por corrección de error, hay que mencionar que también se utilizan en algunas redes monocapa con conexiones laterales y autorrecurrentes como es el caso de la red Brain-State-In-A-Box (BSB), introducida por Anderson, Silverstein Ritz y Jones en 1977 [33]. Aunque en una primera fase el aprendizaje de esta red es sin supervisión, se suelen refinar los valores de los pesos de las conexiones mediante un aprendizaje por corrección de error basado en una adaptación de la regla delta de Widrow-Hoff.

Aprendizaje por Refuerzo Se trata de un aprendizaje supervisado, más lento que el anterior, que se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado, es decir, de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada.

En el aprendizaje por refuerzo la función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito = +1 o fracaso = -1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades. Se podría decir que en este tipo de aprendizaje, la función del supervisor se asemeja más a la de un crítico (que opina sobre la respuesta de la red) que a la de un maestro (que indica a la red la respuesta concreta que debe generar), como ocurría en el caso de supervisión por corrección de error.

Un ejemplo de algoritmo por refuerzo lo constituye el presentado por Narendra y Thathacher en 1974 [44] denominado Linear Reward-Penalty o L_{R-P} (algoritmo lineal con recompensa y penalización). Este algoritmo ha sido ampliado por Barto y Anandan, quienes en 1985 [42] desarrollaron el denominado Associative Reward-Penalty o A_{R-P} (algoritmo asociativo con recompensa y penalización), que se aplica en redes con conexiones hacia adelante de dos capas cuyas neuronas de salida presentan una función de activación estocástica.

Otro algoritmo por refuerzo es el conocido como Adaptive Heuristic Critic, introducido por Barto, Sutton y Anderson en 1983 [43], que se utiliza en redes feedforward de tres capas especialmente diseñadas para que una parte de la red sea capaz de generar un valor interno de refuerzo que es aplicado a las neuronas de salida de la red.

Aprendizaje Estocástico Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

En el aprendizaje estocástico se suele hacer una analogía en términos termodinámicos, asociando la red neuronal con un sólido que tiene cierto estado energético. En el caso de la red, la energía de la misma representaría el grado de estabilidad de la red, de tal forma que el estado de mínima energía correspondería a una situación en la que los pesos de las conexiones consiguen que su funcionamiento sea el que más se ajusta al objetivo deseado.

Según lo anterior, el aprendizaje consistiría en realizar un cambio aleatorio de los valores de los pesos y determinar la energía de la red (habitualmente la función de energía es una función denominada de Lyapunov). Si la energía es menor después del cambio, es decir, si el comportamiento de la red se acerca al deseado, se acepta el cambio. Si, por el

contrario, la energía no es menor, se aceptaría el cambio en función de una determinada y preestablecida distribución de probabilidades.

Una red conocida como máquina de Boltzmann utiliza este tipo de aprendizaje, ideada por Hinton, Ackley y Sejnowski en 1984 [35], que lo combina con el aprendizaje Hebbiano (se describirá en el siguiente apartado) o con aprendizaje por corrección de error (como la regla delta). La red máquina de Boltzmann es una red con diferentes topologías alternativas, pero siempre con unas neuronas ocultas que permiten, mediante un ajuste probabilístico, introducir un ruido que va decreciendo durante el proceso de aprendizaje para escapar de los mínimos relativos (locales) de la función de energía favoreciendo la búsqueda del mínimo global.

El procedimiento de utilizar ruido para escapar de mínimos locales suele denominarse **simulated annealing** (temple simulado) y su combinación con la asignación probabilística mediante la capa oculta es lo que se conoce como aprendizaje estocástico. El término *simulated annealing* proviene del símil termodinámico antes mencionado. La idea es asemejar la red con un sólido físico que inicialmente presenta una alta temperatura (ruido) y que se va enfriando gradualmente hasta alcanzar el equilibrio térmico (mínima energía).

Existe otra red basada en este tipo de aprendizaje, denominada *Cauchy Machine*, desarrollada por Szu en 1986 [48], que es un refinamiento de la anterior y que utiliza un procedimiento más rápido de búsqueda del mínimo global y una función de probabilidad diferente (la distribución de probabilidad de Cauchy frente a la de Boltzmann, utilizada en la red anterior).

Redes con Aprendizaje no Supervisado

Las redes con aprendizaje no supervisado (también conocido como auto-supervisado) no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas.

La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; por ello, suele decirse que estas redes son capaces de auto organizarse.

Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten en su entrada. Puesto que no hay un supervisor que indique a la red la respuesta que debe generar ante una entrada concreta, cabría preguntarse precisamente por lo qué la red genera en estos casos. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado.

En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado). En otro caso, podría realizar una clusterización (clustering) o establecimiento de categorías, indicando la salida de la red a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de correlaciones entre las informaciones presentadas. Una variación de esta categorización es el prototipado. En este caso, la red obtiene ejemplares o prototipos representantes de las clases a las que pertenecen las informaciones de entrada.

También el aprendizaje sin supervisión permite realizar una codificación de los datos de entrada, generando a la salida una versión codificada de la entrada, con menos bits, pero manteniendo la información relevante de los datos.

Finalmente, algunas redes con aprendizaje no supervisado, lo que realizan es un mapeo de características (feature mapping), obteniéndose en la neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada. De tal forma que si se presentan a la red informaciones similares, siempre serán

afectadas neuronas de salida próximas entre sí, en la misma zona del mapa.

En cuanto a los algoritmos de aprendizaje no supervisado, en general se suelen considerar dos tipos que dan lugar a los siguientes aprendizajes: Aprendizaje hebbiano y Aprendizaje competitivo y cooperativo, de los cuales se describe únicamente al primero.

Aprendizaje Hebbiano Este tipo de aprendizaje se basa en el siguiente postulado formulado por Donald O. Hebb en 1949 [4]: Cuando un axón de una celda A está suficientemente cerca como para conseguir excitar una celda B y repetida o persistentemente toma parte en su activación, algún proceso de crecimiento o cambio metabólico tiene lugar en una o ambas celdas, de tal forma que la eficiencia de A , cuando la celda a activar es B , aumenta. Por celda, Hebb entiende un conjunto de neuronas fuertemente conexas a través de una estructura compleja. La eficiencia podría identificarse con la intensidad o magnitud de la conexión, es decir, que el aprendizaje hebbiano consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación (multiplicación en el caso de valores binarios +1 y -1) de los valores de activación (salidas) de las dos neuronas conectadas:

$$\Delta W_{ji} = y_i \cdot y_j$$

Esta expresión responde a la idea de Hebb, puesto que si las dos unidades son activas (positivas), se produce un reforzamiento de la conexión. Por el contrario, cuando una es activa y la otra pasiva (negativa), se produce un debilitamiento de la conexión. Se trata de una regla de aprendizaje no supervisado, pues la modificación de los pesos se realiza en función de los estados (salidas) de las neuronas obtenidos tras la presentación de cierto estímulo (información de entrada a la red), sin tener en cuenta si se deseaba obtener o no esos estados de activación.

Este tipo de aprendizaje fue empleado por Hopfield en la conocida red que lleva su nombre (Red Hopfield), introducida en 1982 y muy extendida en la actualidad, debido principalmente a la relativa facilidad en su implementación en circuitos integrados VLSI.

2.4.4. Ventajas de las Redes Neuronales

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Estas ventajas incluyen [39]:

- Aprendizaje adaptativo. Capacidad de aprender a realizar tareas basada en un entrenamiento o una experiencia inicial
- Autoorganización. Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje
- Tolerancia a fallos. La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño
- Operación en tiempo real. Los computadores neuronales pueden ser realizados en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad
- Fácil inserción dentro de la tecnología existente. Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello

facilitará la integración modular en los sistemas existentes.

Capítulo 3

REDES NEURONALES CELULARES

En este capítulo se definen las redes neuronales celulares, se describen sus fundamentos analíticos y se incluyen los resultados existentes con relación a ellas.

El capítulo está organizado como sigue: en la sección 3.1 se presenta la definición y rasgos principales de las redes neuronales celulares (RNC); en la sección 3.2 se analiza la estructura (arquitectura) de las RNC, se describen sus elementos y se da su ecuación dinámica; en la sección 3.3 se tratan las condiciones que permiten el diseño de las RNC y se analiza su rango dinámico; en la sección 3.4 se presentan la definición de la función de Lyapunov para RNC y los teoremas que aseguran la estabilidad de las RNC.

3.1. Definición y Características de Redes Neuronales Celulares

En esta sección se presenta un circuito con una nueva arquitectura, llamado red neuronal celular, el cual posee algunos de los principales aspectos de las redes neuronales, el cual tiene importantes aplicaciones potenciales en áreas tales como procesamiento de imágenes y reconocimiento de patrones.

Definición 3.1

Una **Red Neuronal Celular** (RNC) es un arreglo regular (rectangular, hexagonal, etc.) de sistemas dinámicos idénticos llamados celdas las cuales satisfacen dos propiedades:

- 1) la mayoría de las interacciones son locales dentro de un radio finito r
- 2) todos los valores de estado son señales continuas.

Algunos de los términos básicos usados en la teoría de RNC son:

- **Celda base**, o simplemente celda: es la unidad de procesamiento general de una RNC. En teoría y modelaje de circuitos se usa, en la mayoría de los casos, una celda de primer orden que consiste en un capacitor lineal, un resistor lineal, una fuente de corriente constante y una fuente de corriente de voltaje controlado adicional (i.e. las interacciones desde las celdas vecinas). La medida de voltaje en el capacitor lineal es el valor del estado de la celda base. Cada celda tiene su propia entrada y su propia salida, que es calculada a través de las salidas características de los valores de estado
- **Salida característica**: la salida esta definida por una función general no lineal (en la mayoría de los casos del tipo sigmoide); conexión del estado de una celda simple
- **Capas de RNC**: En su forma más general una RNC es un arreglo tridimensional,

pero puede ser interpretado como un arreglo de procesamiento bidimensional de múltiples capas

3.2. Arquitectura de Redes Neuronales Celulares

La unidad de circuito básico de redes neuronales celulares es llamada una **celda**. Esta contiene elementos de circuitos lineales y no lineales, los cuales generalmente son capacitores lineales, resistores lineales, fuentes controladas lineales y no lineales y fuentes independientes [7]. La estructura de redes neuronales celulares es similar a la del autómata celular; a saber, cualquier celda en una red neuronal celular esta conectada sólo a sus celdas vecinas. Las celdas adyacentes pueden interactuar directamente unas con otras. Las celdas que no están conectadas directamente pueden afectarse unas a otras indirectamente a causa de los efectos de propagación de la dinámica continua en el tiempo de las redes neuronales celulares.

Se considera una **red neuronal celular** $M \times N$, como una red que tiene $M \times N$ celdas arregladas en M hileras y N columnas. Un ejemplo de una red neuronal celular de dos dimensiones se muestra en la figura 8. Teóricamente, puede definirse una red neuronal celular de cualquier dimensión, pero conviene concentrarse en el caso de dos dimensiones debido a que se aplicará en problemas de procesamiento de imágenes. Los resultados pueden ser fácilmente generalizados a casos de más dimensiones. Para representar a la celda sobre el i -ésimo renglón y la j -ésima columna se usa $C(i, j)$. Ahora se define una r -vecindad de $C(i, j)$.

Definición 3.2

La **r -vecindad** de una celda $C(i, j)$, en una red neuronal celular está definida por:

$$N_r(i, j) = \{C(k, l) : \max\{|k - i|, |l - j|\} \leq r, 1 \leq k \leq m; 1 \leq l \leq n\} \quad (3.1)$$

donde r es un número entero positivo.

En la figura 9 se muestran 2 vecindades de las celdas centrales (las cuáles están sombreadas), con $r = 1$ y $r = 2$, respectivamente. Usualmente a la 1-vecindad ($r = 1$) se le llama una vecindad 3x3; a la 2-vecindad ($r = 2$), una vecindad 5x5; a la 3-vecindad ($r = 3$), una vecindad 7x7, etc. Es fácil mostrar que el sistema de vecindades exhibe una propiedad de simetría en el sentido siguiente si $C(i, j) \in N_r(k, l)$, entonces $C(k, l) \in N_r(i, j)$ para toda $C(i, j)$ y $C(k, l)$ en una red neuronal celular.

Un ejemplo típico de una celda $C(i, j)$ de una red neuronal celular aparece en la figura 10 donde los subíndices u , x e y denotan la entrada, estado y salida, respectivamente. El nodo de voltaje v_{xij} de $C(i, j)$ es llamado el estado de la celda y su condición inicial se supone que tiene una magnitud menor o igual que 1. El nodo de voltaje v_{uij} es llamado la entrada de $C(i, j)$ y se supone que es constante con magnitud menor o igual a 1. El nodo de voltaje v_{yij} es llamado la salida.

Observe en la figura 10 que cada celda $C(i, j)$ contiene una fuente de voltaje independiente E_{ij} , una fuente de corriente independiente I , un capacitor lénel C , dos resistores lineales R_x y R_y y a lo más $2m$ fuentes de corriente de voltaje controlado las cuales están acopladas a sus celdas vecinas por medio de la entrada de voltaje controlado v_{ukl} , y la retroalimentación del voltaje de salida v_{ykl} de cada celda vecina $C(k, l)$, donde m es el número de celdas vecinas. En particular, se tiene que $I_{xy}(i, j; k, l)$ y $I_{xu}(i, j; k, l)$ son fuentes de corriente de voltaje controlado. Con las siguientes características: $I_{xy}(i, j; k, l) = A(i, j; k, l)v_{ykl}$ e $I_{xu}(i, j; k, l) = B(i, j; k, l)v_{ukl}$, $\forall C(k, l) \in N_r(i, j)$, la

función de corriente de voltaje controlado $I_{yx} = (I/R_y)f(v_{xij})$ es el único elemento no lineal en cada celda, es una fuente de corriente de voltaje controlado lineal por pedazos con función característica $f(v)$. Como se muestra en la figura 11.

Todas las fuentes controladas lineales y lineales por tramos usadas en la red neuronal celular pueden ser realizadas fácilmente usando amplificadores operacionales (op amps) [49],[50].

Sin pérdida de generalidad, la estructura del circuito celda de la figura 10 será usado a través de este escrito.

Aplicando KCL y KVL, para $1 \leq i \leq M$; $1 \leq j \leq N$, las ecuaciones del circuito de una celda son fácilmente derivadas como sigue:

Ecuación de Estado :

$$C \frac{dv_{xij}(t)}{dt} = \frac{-1}{R_x} v_{xij}(t) + \sum_{C(k,l) \in Nr(i,j)} A(i,j;k,l) v_{ykl}(t) + \sum_{C(k,l) \in Nr(i,j)} B(i,j;k,l) v_{ukl} + I \quad (3.2 \text{ a})$$

Ecuación de Salida :

$$v_{yij}(t) = \frac{1}{2} (| v_{xij}(t) + 1 | - | v_{xij}(t) - 1 |), \quad (3.2 \text{ b})$$

Ecuación de Entrada :

$$v_{uij} = E_{ij}, \quad (3.2 \text{ c})$$

Condiciones de Restricción :

$$| v_{xij}(0) | \leq 1, \quad (3.2 \text{ d})$$

$$|v_{ij}| \leq 1, \quad (3.2 \text{ e})$$

Hipótesis sobre Parámetros :

$$A(i, j; k, l) = A(k, l; i, j), \quad (3.2 \text{ f})$$

$$C > 0, R_x > 0. \quad (3.2 \text{ g})$$

Notas:

3.2.1 Todas las celdas interiores de una red neuronal celular tienen la misma estructura y elementos valorados (valores de elementos); la **celda interior** es aquella que tiene $(2r + 1)^2 - 1$ celdas vecinas, donde r se define en (3.1).

Todas las otras celdas son llamadas celdas frontera. Una red neuronal celular está completamente caracterizada por el conjunto de ecuaciones diferenciales (3.2) asociadas con las celdas en el circuito.

3.2.2 Cada celda de una red neuronal celular tiene a lo más 3 nodos. Algunas veces elegimos $E_{ij} = 0$ si $B(i, j; k, l) = 0$ para todas las celdas en una red neuronal celular. En este caso, hay sólo dos nodos en una celda circuito. Ya que todas las celdas tienen los mismos datos (# de nodos) y todos los elementos de circuitos son de voltaje controlado, nuestra red neuronal celular está idealmente situada para análisis nodal. Es más, ya que las interacciones son locales, la matriz asociada a la ecuación de nodos es extremadamente escasa para circuitos grandes.

3.2.3 La dinámica de una red neuronal celular tiene salida retroalimentada y mecanismo de control de entrada. Los efectos de la salida retroalimentada dependen del parámetro interactivo $A(i, j; k, l)$ y los efectos del control de entrada de $B(i, j; k, l)$. Consecuentemente, es instructivo, algunas veces referirse a $A(i, j; k, l)$ como operador de control. Los

supuestos en (3.2f) son razonables debido a la propiedad de simetría del sistema de vecindades.

3.2.4 Los valores de los elementos del circuito pueden elegirse convenientemente en la práctica. R_x y R_y determinan la potencia disipada en el circuito y se eligen usualmente de manera que estén entre $1\text{ K}\Omega$ y $1\text{ M}\Omega$. CR_x es la constante de tiempo de la dinámica del circuito y usualmente se elige entre 10^{-8} - 10^{-5} s.

3.2.1. Rango Dinámico de Redes Neuronales Celulares

Antes de diseñar una red neuronal celular física, es necesario conocer su rango dinámico para garantizar que satisface los supuestos en la ecuación dinámica estipulada en la sección precedente. El siguiente teorema da los fundamentos para el diseño de la red.

Teorema 3.3.1

Todos los estados v_{xij} de las celdas en una red neuronal celular están acotados para todo tiempo $t > 0$ y la cota v_{max} puede calcularse por la siguiente fórmula para cualquier red neuronal celular:

$$v_{max} = 1 + R_x |I| + R_x \max_{1 < i < m, 1 < j < n} \left[\sum_{C(k,l) \in Nr(i,j)} (|A(i,j;k,l)| + |B(i,j;k,l)|) \right] \quad (3.3)$$

Para cualquier red neuronal celular, los parámetros R_x , C , I , $A(i,j;k,l)$ y $B(i,j;k,l)$ son constantes finitas, por lo tanto la cota sobre el estado de las celdas v_{max} , es finita y puede calcularse usando la fórmula (3.3).

Observación:

En el diseño del circuito real, es conveniente elegir la escala de los parámetros de manera tal que se satisfagan las condiciones siguientes:

$$R_x |I| \cong 1, R_x |A(i,j;k,l)| \cong 1 \text{ y } R_x |B(i,j;k,l)| \cong 1, \forall_{i,j,k,l} \quad (3.4)$$

Entonces se estima fácilmente la cota superior del rango dinámico de nuestra red neuronal celular 3x3, podemos tener $v_{max} \cong 20$ V, lo cual está dentro del suministro de potencia típico del rango de voltaje de circuitos IC.

3.2.2. Estabilidad de Redes Neuronales Celulares

¿Cómo podemos garantizar la convergencia de redes neuronales celulares? ¿Cuáles son las condiciones o restricciones para que tal convergencia sea posible? En esta sección se discute la propiedad de convergencia y los problemas relacionados para redes neuronales celulares.

Una de las técnicas más efectivas para analizar las propiedades de convergencia de circuitos dinámicos no lineales es el método de Lyapunov. Entonces, primero se define una función de Lyapunov para redes neuronales celulares.

Definición 3.3

Se define la **Función de Lyapunov**, $E(t)$ de una red neuronal celular como la función escalar

$$E(t) = -\frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} A(i,j;k,l) v_{yij}(t) v_{ykl}(t) + \frac{1}{2R_x} \sum_{(i,j)} v_{yij}(t)^2 - \sum_{(i,j)} \sum_{(k,l)} B(i,j;k,l) v_{yij}(t) v_{ukl} - \sum_{(i,j)} I v_{yij}(t) \quad (3.5)$$

Observaciones:

3.4.1 Obsérvese que la función de Lyapunov dada por (3.5) es solo una función de los voltajes de entrada v_u , y de salida v_y del circuito. Aunque esta no contiene la información completa contenida en las variables de estado v_{xij} , sin embargo, podemos derivar las propiedades de estado estable de las variables de estado de las propiedades de $E(t)$.

3.4.2 La función de Lyapunov definida arriba, puede interpretarse como “la energía generalizada” de una red neuronal celular, aunque el significado exacto físico no es muy claro. Como los siguientes teoremas lo mostrarán, $E(t)$ siempre converge a un mínimo local, donde la red neuronal celular produce la salida deseada.

El siguiente teorema muestra que $E(t)$ es acotada.

Teorema 3.4.1

La función definida en (3.5) está acotada por:

$$\text{Max}_t | E(t) | \leq E_{max} \quad (3.6)$$

$$E_{max} = \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} | A(i, j; k, l) | + \sum_{(i,j)} \sum_{(k,l)} | B(i, j; k, l) | + MN \left(\frac{1}{2R_x} + | I | \right) \quad (3.7)$$

para una red neuronal celular MxN .

Teorema 3.4.2

La función escalar $E(t)$ definida en (3.5) es una función monótona decreciente, esto es:

$$\frac{d}{dt} E(t) \leq 0 \quad (3.8)$$

Observaciones:

3.4.3 Para análisis futuros (corolario del teorema 3.4.3), es conveniente escribir $\frac{d}{dt} E(t)$ como sigue:

$$\frac{d}{dt} E(t) = - \sum_{|v_{xij}| < 1} C \left[\frac{d}{dt} v_{xij}(t) \right]^2 = \sum_{|v_{yij}| < 1} -C \left[\frac{d}{dt} v_{yij}(t) \right]^2 = - \sum_{(i,j)} C \left[\frac{d}{dt} v_{yij}(t) \right]^2 \leq 0 \quad (3.9)$$

3.4.4 En la prueba del teorema anterior, se asume que $v_{yij} = f(v_{xij})$ es una función lineal por pedazos y se ha definido su derivada en los puntos de ruptura. En cualquier implementación hardware (e.g. VLSI), $v_{yij} = f(v_{xij})$ es una función suave en el sentido

que es continuamente diferenciable. Afortunadamente, se puede probar que el teorema 3.4.2 sigue siendo válido para cualquier función sigmoideal que satisfaga la condición. $\frac{d}{dv_{xij}}v_{yij}(t) \geq 0$. Para probar esto simplemente reemplace la segunda expresión en la función de Lyapunov en (3.5) por:

$$\frac{1}{R_x} \sum_{(i,j)} \int_0^{v_{yij}(t)} f^{-1}(v)dv \quad (3.10)$$

y diferenciando directamente $E(t)$ para obtener

$$\frac{d}{dt}E(t) = - \sum_{(i,j)} C \frac{d}{dv_{xij}}v_{yij}(t) \left[\frac{d}{dt}v_{xij}(t) \right]^2 \leq 0 \quad (3.11)$$

De los teoremas 3.4.1 y 3.4.2, puede probarse fácilmente el siguiente resultado.

Teorema 3.4.3

Para cualquier entrada v_u y cualquier estado inicial v_x de una red neuronal celular, se cumple:

$$\text{Lim}_{t \rightarrow \infty} E(t) = \text{constante} \quad (3.12)$$

$$\text{Lim}_{t \rightarrow \infty} \frac{d}{dt} E(t) = 0 \quad (3.13)$$

Corolario:

Después que el estado transitorio de una red neuronal celular haya decaído a cero, se obtiene una salida constante. En otras palabras, se tiene que:

$$\text{Lim}_{t \rightarrow \infty} v_{yij}(t) = \text{constante}, \quad 1 \leq i \leq M; 1 \leq j \leq N \quad (3.14)$$

$$\text{Lim}_{t \rightarrow \infty} \frac{d}{dt} v_{yij}(t) = 0, \quad 1 \leq i \leq M; 1 \leq j \leq N. \quad (3.15)$$

En seguida, se investiga el comportamiento de estado estable de redes neuronales celulares. Se sigue de la prueba del teorema 3.4.2 y de la función característica de salida lineal por pedazos (3,2b) que bajo la condición $\frac{d}{dt}E(t) = 0$, hay tres casos posibles para el estado de una celda cuando t tiende a infinito:

caso 1:

$$\frac{dv_{xij}(t)}{dt} = 0 \quad y \quad |v_{xij}(t)| < 1 \quad (3.16)$$

caso 2:

$$\frac{dv_{xij}(t)}{dt} = 0 \quad y \quad |v_{xij}(t)| \geq 1 \quad (3.17)$$

caso 3:

$$\frac{dv_{xij}(t)}{dt} \neq 0 \quad y \quad |v_{xij}(t)| \geq 1 \quad (3.18)$$

Es claro que $v_{yij}(t) = v_{xij}(t)$ si consideramos la figura 3.4 cuando $|v_{xij}(t)| < 1$, y entonces $\frac{dv_{yij}(t)}{dt} = \frac{dv_{xij}(t)}{dt}$. Del teorema 3.4.3 y su corolario se sigue el caso 1. Pero para $|v_{xij}(t)| > 1$ puesto que $v_{yij}(t) \neq v_{xij}(t)$, donde $v_{yij}(t) = \pm 1$ es una constante, no tenemos la forma precisa de onda de $v_{xij}(t)$. En el caso donde $v_{xij}(t)$ es una constante, tenemos el caso 2. De otra manera, aplica el caso 3 y $v_{xij}(t)$ puede ser una función del tiempo periódica o no, pero acotada, en vista del teorema 3.4.1.

¿Es posible que coexistan los 3 casos entre las diferentes variables de estado cuando una red neuronal celular está en su estado estable, o puede existir sólo uno o dos de los 3 casos? Se asegura que sólo el caso 2 puede existir para toda v_{xij} en el estado estable

bajo un supuesto débil (suave) sobre el rango de los parámetros del circuito $A(i, j; k, l)$ y R_x .

Para probarlo se reescribe la ecuación de la celda (3.2) como sigue:

$$C \frac{dv_{xij}(t)}{dt} = -f(v_{xij}(t)) + g(t) \quad (3.19)$$

donde

$$f(v_{xij}(t)) = (-0.5) A(i, j; i, j) (|v_{xij}(t) + 1| - |v_{xij}(t) - 1|) + \frac{1}{R_x} v_{xij}(t) \quad (3.20)$$

y

$$g(t) = \sum_{C(k,l) \in N_r(i,j), C(k,l) \neq C(i,j)} (A(i, j; k, l)v_{ykl}(t) + B(i, j; k, l)v_{ukl}(t)) + I. \quad (3.21)$$

Luego se hacen algunas restricciones sobre la función f en (3.20). Suponiendo por conveniencia que $A(i, j; k, l) > 1/R_x$ y sin perder generalidad, sea $A(i, j; i, j) = 2$, $R_x = 1$ y $C = 1$, en el siguiente análisis. Entonces $f(v_{xij})$ es la función característica mostrada en la figura 12.

Considerando el circuito equivalente a una celda en una red neuronal celular que se muestra en la figura 13.

Existen solamente 3 elementos en el circuito; un capacitor lineal con capacitancia C positiva, un resistor de voltaje controlado lineal por tramos (pedazos) con su conducción característica $i_R = f(v_r)$ (f es la misma función de la figura 12), y una fuente de corriente independiente variable en el tiempo cuya salida está dada por $g(t)$. Los dos circuitos en las figuras 10 y 13 son equivalentes porque ambos están descritos por (3.19) la cual reescribimos por simplicidad como sigue:

$$\frac{dv(t)}{dt} = -f(v(t)) + g(t) = F(v, t) \quad (3.22)$$

para $g(t) = 0$, los puntos de equilibrio y la ruta dinámica [49], [50] del circuito equivalente se muestran en la figura 14a. Existen 3 puntos de equilibrio en este circuito, uno de ellos, $v = 0$ es inestable denotado por un círculo; los otros dos, $v = -2$ y $v = 2$, son estables y se denotan por puntos sólidos. El punto de equilibrio inestable nunca se observa en circuitos electrónicos físicos, a causa de su inevitable ruido térmico. Por consiguiente, después de su estado transitorio y dependiendo del estado inicial, el circuito siempre se aproximará a uno de sus puntos de equilibrio estable y permanecerá ahí después de eso. Por ejemplo, si el estado inicial del circuito es $v = 0.5$, entonces el estado estable será observado en el punto de equilibrio $v = 2$; pero si el estado inicial del circuito es $v = -0.5$, entonces el estado estable se observara en el punto de equilibrio estable $v = -2$.

Si $g(t) = \text{una constante}$ diferente de cero, se tienen seis casos del comportamiento dinámico del circuito equivalente mostrados en las figuras 14(b) -14(g). Para los casos en la figura 14(b) y 14(c) hay también 3 puntos de equilibrio; uno de ellos es inestable, mientras los otros dos son estables. Para los casos en las figuras 14(d) y 14(e), hay dos puntos de equilibrio; uno es inestable y el otro es estable. Para la ruta dinámica en las figuras 14(f) y 14(g), existe solamente un punto de equilibrio para el circuito, y este es estable. Observe que todos los puntos de equilibrio estables corresponden a las 7 rutas dinámicas con el circuito equivalente a una celda en una red neuronal celular comparten la propiedad común $|v| > 1$.

Volviendo ahora al circuito básico de una celda de una red neuronal celular, ya que $g(t)$ es una función solo de las salidas $v_{ykl}(t)$ y las entradas v_{ukl} de la vecindad de una celda, se sigue del resultado del teorema (3.4.3) que todas las salidas de estado estable de la red neuronal celular son constantes. Entonces, después del estado transitorio

inicial el supuesto $g(t) = constante$ es válido para el estudio del comportamiento de estado estable de una red neuronal celular.

En las figuras 14(a)-14(g) se muestran las rutas dinámicas y los puntos de equilibrio del circuito equivalente para diferentes valores de $g(t)$.

Las observaciones anteriores se resumen en el siguiente teorema.

Teorema 3.4.4

Si los parámetros del circuito satisfacen la condición

$$A(i, j; i, j) > \frac{1}{R_x}. \quad (3.23)$$

Entonces cada celda de la red neuronal celular debe situarse en un punto de equilibrio estable después que su estado transitorio haya decaído a cero. Más aun, la magnitud de todos los puntos de equilibrio estables es mayor que 1. En otras palabras, se tienen las siguientes propiedades:

$$\text{Lim}_{t \rightarrow \infty} |v_{xij}(t)| > 1, \quad 1 \leq i \leq M; \quad 1 \leq j \leq N \quad (3.24)$$

$$\text{Lim}_{t \rightarrow \infty} |v_{yij}(t)| = \pm 1, \quad 1 \leq i \leq M; \quad 1 \leq j \leq N \quad (3.25)$$

Observaciones:

3.4.5 El teorema anterior es significativo para redes neuronales celulares porque implica que el circuito no oscila ni se vuelve caótico [11], [12].

3.4.6 El teorema 3.4.4 asegura que las salidas de la red neuronal celular tiene valores binarios. Esta propiedad es crucial para resolver problema de clasificación en aplicaciones de procesamiento de imágenes [13].

3.4.7 Es posible mostrar por la misma técnica que sin la restricción dada en (3.23) ambos casos 1 y 2 pueden coexistir pero no el caso 3. Esto implica que la observación 3.4.5 es verdadera aun sin la condición (3.23).

3.4.8 Ya que $A(i, j; i, j)$ corresponde a la retroalimentación de la salida de la celda $C(i, j)$ a su entrada, la condición (3.23) estipula una cantidad mínima de retroalimentación de la salida de la celda $C(i, j)$ a su entrada, la condición (3.23) estipula una cantidad mínima de retroalimentación positiva para garantizar que la salida en estado estable de cada celda es $+1$ o -1 . Note que esta condición siempre es violada en la red neuronal de Hopfield ya que los coeficientes de acoplamiento diagonal se supone que sean cero. Para garantizar salidas binarias en la red de Hopfield, es necesario elegir un slope infinito [61] en la región lineal de la función no lineal f de la figura 11. En contraste, el correspondiente slope (pendiente) en una red neuronal celular se elige siempre igual a uno.

Definición 3.4

Para cualquier planilla clonada T , la cual define la regla dinámica del circuito de la celda, el **operador convolución** $*$ se define por:

$$T * v_{ij} = \sum_{C(k,l) \in N_r(i,j)} T(k-i, l-j) v_{kl} \quad (3.26)$$

Donde $T(m, n)$ denota la entrada en el m -ésimo renglón y n -ésima columna de la planilla clonada, $m = -1, 0, 1$ y $n = -1, 0, 1$, respectivamente.

Note que en la definición anterior se supone que $A(i, j; k, l)$ es independiente de i y de j para esta red neuronal celular. Esta propiedad se dice que es invariante en el espacio, lo que implica que $A(i, j; k, l)$ puede expresarse como $C(k-i, l-j)$. A menos que se especifique lo contrario, se considera que todas las redes neuronales celulares son invariantes en el espacio. Esta propiedad permite especificar la regla dinámica de redes neuronales celulares usando planillas (clonadas).

Definición 3.5

El **estado de equilibrio estable** (v_{xij}^*) de una celda típica de una red neuronal celular se define como la variable de estado v_{xij} de la celda $C(i, j)$, la cual satisface

$$\frac{d}{dt}v_{xij}(t) \Big|_{v_{xij}=v_{xij}^*} = 0 \quad \text{y} \quad |v_{xij}^*(t)| > 1 \quad (3.27)$$

Bajo el supuesto de que $v_{ykl} = \pm 1$ para todas las celdas vecinas $C(k, l) \in N_r(i, j)$.

Comentario:

La definición 3.5 vale para cualquier combinación asumida de $v_{ykl} = \pm 1$, y por consiguiente, puede no representar un componente actual de un estado de equilibrio estable del circuito total.

Definición 3.6

Un **punto de equilibrio estable** del sistema de una red neuronal celular se define como el vector de estado con todos sus componentes consistentes de estados de equilibrio estables de celda.

De las definiciones anteriores se sigue que una red neuronal celular está siempre en uno de sus puntos de equilibrio estable del sistema después que su estado transitorio ha decaído a cero. Desde el punto de vista de la teoría de sistemas dinámicos, el estado transitorio de una red neuronal celular es simplemente la trayectoria que empieza en algún estado inicial y termina en un punto de equilibrio del sistema. Ya que los puntos de equilibrio estables del sistema de una red neuronal celular, como se define en la definición 3.6, es un punto límite de un conjunto de trayectorias de las correspondientes ecuaciones diferenciales (3.2). Tales puntos límites tienen una cuenca de atracción, a saber, la unión de todas las trayectorias que convergen a ese punto. Por consiguiente, el espacio estado de una red neuronal celular puede ser dividida en un conjunto de regiones centradas en los puntos de equilibrio estables del sistema.

Generalmente, una red neuronal celular procesa señales mapeándolas desde un espacio de señales en otro. En este caso, la red neuronal celular puede ser usada para mapear un estado inicial de un sistema en cualquiera de los distintos puntos de equilibrio del sistema. Si consideramos el espacio de estado inicial como $[-1.0, 1.0]^{M \times N}$, y el espacio de salidas como $\{-1, 1\}^{M \times N}$, luego el mapeo dinámico F , puede definirse como

$$F : [-1.0, 1.0]^{M \times N} \longrightarrow \{-1, 1\}^{M \times N}. \quad (3.28)$$

Esto significa que el mapeo puede ser usado para dividir un espacio de señales continuas en varias regiones de atracción de los puntos de equilibrio estables del sistema vía un proceso dinámico. Esta propiedad puede ser explotada en el diseño de memoria asociativa, códigos de corrección de error y sistemas tolerantes a fallas.

En general, el conjunto límite de un sistema no lineal complejo, es muy difícil, si no imposible de determinar, ya sea analíticamente o numéricamente, aunque para circuitos lineales por pedazos es posible encontrar todas las soluciones (dc) usando algoritmos de fuerza bruta [55] o algunos mas eficientes, esto puede no ocurrir si el sistema es muy grande o se consume mucho tiempo. Para la red neuronal celular, en vista de la propiedad interactiva entre vecinas cercanas, se puede resolver para todos los puntos de equilibrio del sistema determinando, primero, los estados de equilibrio estables de cada celda y entonces usar la regla interactiva entre vecinas para encontrar los correspondientes puntos de equilibrio del sistema.

Capítulo 4

DISEÑO PARA REDES NEURONALES BASADO EN EL ALGORITMO DEL PERCEPTRÓN

El capítulo se organiza de la siguiente manera: en la sección 4.1, se presentan los antecedentes del problema y la ecuación dinámica de la red neuronal a diseñar así como también la terminología usada; en la sección 4.2, se introducen los preliminares concernientes al algoritmo de entrenamiento del perceptrón y su teorema de convergencia.

El resto del capítulo se organiza como sigue: en la sección 4.3 se desarrolla un nuevo algoritmo de diseño para memorias asociativas basado en el algoritmo de entrenamiento del perceptrón; en la sección 4.4, se establecen los resultados concernientes a las propiedades de la red neuronal con restricciones sobre los elementos de la diagonal de la matriz de conexión y concernientes a la existencia de tal red; en la sección 4.5 se presentan los algoritmos diseñados para redes neuronales con restricciones de conectividad (simetría y/o escasez).

En este capítulo se desarrolla un nuevo enfoque de diseño para memorias asociativas basado en el algoritmo de entrenamiento del perceptrón [16] y [15]. El problema de diseño de redes neuronales retroalimentadas para memorias asociativas se formula como un conjunto de desigualdades tales que el uso del entrenamiento del perceptrón es evidente. El entrenamiento del perceptrón en el diseño de algoritmos garantiza su convergencia.

4.1. Antecedentes

Las redes neuronales retroalimentadas conectadas completamente de capa simple son un caso especial de sistemas dinámicos no lineales dotados con algunos puntos de equilibrio asintóticamente estables (memorias estables) así como también de equilibrios inestables. El estudio de tales sistemas ha sido de interés en años recientes. Estos análisis tienen que ver con el análisis cualitativo de tales sistemas (redes neuronales) y el diseño de metodologías para dichas redes. El estudio de redes neuronales retroalimentadas escasamente conectadas también ha sido de interés.

En este capítulo se considera la realización de **memorias asociativas** por medio de una clase de red neuronal (el Perceptrón). El objetivo de memorias asociativas es almacenar un conjunto de patrones deseados como memorias estables, tales que, un patrón almacenado pueda ser recuperado cuando el patrón de entrada (o patrón inicial) contiene suficiente información acerca del patrón almacenado. En la práctica los patrones de memoria deseados son representados usualmente por vectores bipolares (o vectores binarios).

En la literatura hay disponibles varios métodos de diseño muy conocidos incluyendo el método del producto externo [26], la regla de aprendizaje de proyección [63], [64]; y el método de eigen-estructura [51, 57, 59]. El método de producto externo requiere que los patrones deseados sean mutuamente ortogonales para que los datos (patrones deseados) sean almacenados en la red. La regla de aprendizaje de proyección no requiere que los patrones prototipo sean mutuamente ortogonales; pero este método no puede garantizar que el equilibrio correspondiente a una memoria deseada sea asintóticamente estable. El método de valores propios parece ser el más efectivo. Este puede garantizar que cualquier conjunto de patrones (vectores) bipolares almacenados como memorias estables corresponden a equilibrios asintóticamente estables de la red neuronal sin que necesiten

ser mutuamente ortogonales. El método de eigen-estructura o valores propios ha sido generalizado para el diseño de redes con restricciones predeterminadas sobre la estructura de interconexiones [56]. En estos métodos de diseño, se formula un conjunto de ecuaciones lineales que se resuelve para el diseño de la red. En metodologías de diseño avanzadas [21] y [22], se formula un conjunto de desigualdades lineales y se resuelve por el método de mínimo cuadrado (error mínimo cuadrado óptimo MSE) usando el método de Ho-Kashyap [37]. En el método desarrollado en [53] se formula y resuelve un conjunto de desigualdades lineales usando técnicas de optimización. En el método presentado en [65], el conjunto de desigualdades se resuelve usando programación lineal.

En este capítulo se diseña una red neuronal con retroalimentación para trabajos de memoria asociativa basándose en el algoritmo de entrenamiento de perceptrones. Este diseño se desarrolla formulando y resolviendo un conjunto de desigualdades lineales. Las desigualdades se resuelven entrenando un conjunto de perceptrones para obtener la matriz de conexión, y el entrenamiento de perceptrones garantiza la convergencia de la red neuronal (del algoritmo) diseñada sin restricciones en la matriz de conexión. Se establecen condiciones sobre las cuales existen soluciones y se proporcionan los algoritmos para redes con restricciones de simetría y/o escasez sobre la matriz de conexión y con restricciones sobre los elementos en la diagonal de la matriz de conexión.

Red Neuronal con Retroalimentación

La clase de redes neuronales considerada en este trabajo esta descrita por las siguientes ecuaciones:

$$\begin{aligned} \dot{x} &= -Ax + T \text{sat}(x) + I \\ y &= \text{sat}(x) \end{aligned} \tag{4.1}$$

donde $x \in \mathfrak{R}^n$ es el vector de estado, \dot{x} denota la derivada de x con respecto al tiempo

po t , $y \in D^n = \{x \in \mathfrak{R}^n : -1 \leq x_i \leq 1, i = 1, 2, 3, \dots, n\}$, es el vector de salida, $A = \text{diag}[a_1, a_2, \dots, a_n]$ con $a_i > 0$ para $i = 1, 2, \dots, n$, $T = [T_{ij}] \in \mathfrak{R}^{n \times n}$ es la matriz coeficiente (matriz de conexión), $I = [I_1, I_2, \dots, I_n]^t \in \mathfrak{R}^n$ es el vector bias y $\text{sat}(x) = [\text{sat}(x_1), \text{sat}(x_2), \dots, \text{sat}(x_n)]^t$ representa la función de activación, donde $\text{sat}(x)$ se define como:

$$\text{sat}(x_i) = \begin{cases} 1, & x_i > 1 \\ x_i, & -1 \leq x_i \leq 1 \\ -1, & x_i < -1 \end{cases} \quad (4.2)$$

se supone que el estado inicial de (4.1) satisface $|x_i(0)| \leq 1$ para $i = 1, 2, \dots, n$. El sistema (4.1) es una variante del modelo analógico de Hopfield [27] con función de activación $\text{sat}(x)$. Hay que aclarar que la red descrita por el sistema 4.1 no es una red neuronal celular.

El problema de diseño de la red neuronal (4.1) para memorias asociativas será formulado de manera tal que el uso de perceptrones resulte evidente.

4.2. Preliminares

En esta sección se introducen los preliminares necesarios para lograr el diseño de la red neuronal deseada, incluyendo el algoritmo de entrenamiento del perceptrón y su teorema de convergencia. Un número de diferentes tipos de perceptrones se describen en [30] y [18]. El que será utilizado está descrito por:

$$z = \text{sgn}(Wu) \quad (4.3)$$

donde $u = [u_1, u_2, \dots, u_n]^t$, $W = [w_1, w_2, \dots, w_n, \theta]$, y

$$\text{sgn}(\ell) = \begin{cases} 1, & \ell \geq 0 \\ -1, & \ell < 0 \end{cases} \quad (4.4)$$

Este perceptrón simple puede realizar clasificación de patrones (entre dos clases denotadas por X_1 y X_2). El vector de pesos W puede obtenerse por el siguiente algoritmo.

Algoritmo de entrenamiento del perceptrón:

Dados m patrones de entrenamiento α^k , $k = 1, 2, \dots, m$, los cuales se sabe pertenecen a las clases X_1 (correspondiente a $z = 1$) o X_2 (correspondiente a $z = -1$), el vector de pesos W puede obtenerse por el siguiente algoritmo:

- 1) Inicializar el vector de pesos $W(\ell)$, para $\ell = 0$.
- 2) Para $\ell = 0, 1, 2, \dots$
 - a) si $W(\ell)u(\ell) \geq 0$ y $u(\ell) \in X_2$, entonces se actualiza W por $W(\ell + 1) = W(\ell) - \rho u(\ell)$;
 - b) si $W(\ell)u(\ell) < 0$ y $u(\ell) \in X_1$, entonces se actualiza W por $W(\ell + 1) = W(\ell) + \rho u(\ell)$;
 - c) de otra forma, $W(\ell + 1) = W(\ell)$, donde $u(\ell) = \alpha^k$, para $1 \leq k \leq m$,

y $\rho > 0$ es la razón de aprendizaje del perceptrón.

3) Parar el entrenamiento cuando no sean necesarias más actualizaciones del vector de pesos W , es decir, el entrenamiento termina cuando todos los patrones de entrenamiento puedan ser correctamente clasificados por W .

El siguiente resultado es muy conocido [30].

Teorema 4.1 (de convergencia del algoritmo del perceptrón):

El algoritmo de entrenamiento del perceptrón converge si y sólo si X_1 y X_2 son linealmente separables.

Comentario: Nótese que se puede continuar el entrenamiento del perceptrón hasta que el vector de pesos W obtenido sea tal que

$$W\alpha^k > 0 \text{ si } \alpha^k \in X_1 \text{ y}$$

$$W\alpha^k < 0 \text{ si } \alpha^k \in X_2 \text{ para } k = 1, 2, \dots, m.$$

Este algoritmo será usado para el diseño de redes neuronales con retroalimentación para realizar memorias asociativas [61].

4.3. Un Nuevo Algoritmo de Diseño

Definición 4.3.1

Un **punto de equilibrio** x_e del sistema (4.1) es **asintóticamente estable** si:

1) x_e es **estable en el sentido de Lyapunov**, es decir, si $\forall \varepsilon > 0$ existe $\delta = \delta(\varepsilon)$ tal que $\|x(t) - x_e\| < \varepsilon \quad \forall t \geq 0$, cuando $\|x(0) - x_e\| < \delta$ ($\|\cdot\|$ denota cualquier norma vectorial).

2) x_e es un **atractor local**; es decir, si $x(t) \rightarrow x_e$ cuando $t \rightarrow \infty$ y $x(0) \in D(x_e)$, donde $D(x_e)$ es una vecindad abierta en \mathfrak{R}^n que contiene a x_e .

El conjunto más grande $D(x_e)$ para el cual la propiedad precedente es verdadera es llamado **dominio de atracción o región de atracción** de x_e .

Definición 4.3.2

Un vector α será llamado un **vector de memoria estable** o simplemente una memoria del sistema (4.1) si $\alpha = sat(\beta)$ y β es un punto de equilibrio asintóticamente estable del sistema (4.1).

Se utiliza B^n para denotar el conjunto de vectores bipolares n -dimensionales, esto es:

$$B^n = \{ x \in \mathfrak{R}^n : x_i = 1 \text{ o } x_i = -1, i = 1, 2, 3, \dots, n \}. \quad (4.5)$$

A B^n también se le conoce como hipercubo [52].

Definición 4.3.3

Para $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]^t \in B^n$ se define **la clase de α** , $C(\alpha)$ como:

$$C(\alpha) = \{ x \in \mathfrak{R}^n : x_i \alpha_i > 1, i = 1, 2, 3, \dots, n \}. \quad (4.6)$$

Un resultado para memorias bipolares establecido en [53] y [55] enuncia el siguiente lema:

Lema 4.3.1

si $\alpha \in B^n$ y si

$$\beta = A^{-1}(T\alpha + I) \in C(\alpha) \quad (4.7)$$

entonces (α, β) es un par que consta de un vector de memoria estable y un punto de equilibrio asintóticamente estable de (4.1).

Para $\alpha \in B^n$ y $x \in C(\alpha)$ la primera ecuación de (4.1) puede ser escrita como:

$$\dot{x} = -Ax + T\text{sat}(\alpha) + I \quad (4.8)$$

El sistema (4.8) tiene un único punto de equilibrio en $x_e = A^{-1}(T\alpha + I)$ y

$x_e = \beta \in C(\alpha)$ por hipótesis. Claramente este equilibrio es asintóticamente estable, ya que en el sistema (4.8) todos los eigen valores λ_i de $-A$ son negativos porque los de A son positivos $\lambda_i(A) = a_i > 0$.

El siguiente problema de síntesis concierne al diseño de la red neuronal determinada por el conjunto de ecuaciones (4.1) para usar memorias asociativas.

Problema de Diseño (Síntesis):

Dados m vectores $\alpha^1, \alpha^2, \dots, \alpha^m$ en B^n , elegir $\{A, T, I\}$ de manera que:

- 1) $\alpha^1, \alpha^2, \dots, \alpha^m$ sean vectores de memoria estable del sistema (4.1)
- 2) El sistema no tenga soluciones caóticas ni oscilatorias
- 3) El número total de vectores de memoria espurios, es decir, vectores de memoria no deseados, sea lo más pequeño posible, y el dominio de atracción de cada memoria deseada sea lo más grande posible.

El punto 1) del problema de síntesis puede garantizarse eligiendo $\{A, T, I\}$ tal que cualquier α^i satisfaga la condición (4.7) del lema 4.3.1.

El punto 2) puede conseguirse diseñando una red neuronal con matriz de conexión simétrica T , lo cual será tratado en la sección 4.5. El punto 3) puede asegurarse parcialmente restringiendo los elementos de la diagonal de la matriz de conexión que se discutirá en la sección 4.4.

Se mostrará que el problema de síntesis puede resolverse aplicando el algoritmo de entrenamiento del perceptrón resumido en la sección 4.2.

Para resolver el problema de síntesis, es necesario determinar A, T e I de (4.3) con $i = 1, 2, \dots, n$, $k = 1, 2, \dots, m$, con $\alpha = \alpha^k$, i.e., uno necesita determinar A, T e I de forma tal que

$$A^{-1}(T(\alpha^k + I)) \in C(\alpha). \quad (4.9)$$

La condición (4.9) puede escribirse equivalentemente como:

$$T_i \alpha^k + I_i > a_i \quad \text{si } \alpha_i^k = 1 \quad \text{y} \quad T_i \alpha^k + I_i < -a_i \quad \text{si } \alpha_i^k = -1 \quad (4.10)$$

donde T_i representa el i -ésimo renglón de T , I_i denota el i -ésimo elemento de I y α_i^k es la i -ésima entrada de α^k . De la expresión anterior (4.10), o equivalentemente, de (4.9) se puede obtener ahora el algoritmo de síntesis (algoritmo de diseño) basado en el algoritmo de entrenamiento del perceptrón.

Algoritmo de diseño 4.3.1:

Usando el algoritmo de entrenamiento del perceptrón, obtener n vectores de pesos (para n perceptrones) $W^i = [w_1^i, w_2^i, \dots, w_{n+1}^i]$, para $i = 1, 2, \dots, n$, tal que:

$$W^i \bar{\alpha}^k \geq 0 \text{ si } \alpha_i^k = 1 \text{ y } W^i \bar{\alpha}^k < 0 \text{ si } \alpha_i^k = -1 \quad (4.11)$$

donde $\bar{\alpha}^k$ está definida por:

$$\bar{\alpha}^k = \begin{bmatrix} \alpha^k \\ 1 \end{bmatrix} \text{ para } k = 1, 2, \dots, m. \quad (4.12)$$

Sea $A = \text{diag}[a_1, a_2, \dots, a_n]$ con $a_i > 0$ para $i, j = 1, 2, \dots, n$.

Elija $T_{ij} = w_j^i$ si $i \neq j$, en caso contrario, $T_{ii} = w_i^i + a_i \mu_i$ con $\mu_i > 1$ y $I_i = w_{n+1}^i$.

El siguiente resultado esta dirigido a establecer la validez del algoritmo de síntesis enunciado arriba.

Teorema 4.3.1

1) Las matrices A , T y el vector I obtenidos por el algoritmo de síntesis 4.3.1 satisfacen la condición (4.9), es decir, (4.1) por lo que satisfacen claramente el punto 1) del problema de diseño.

2) El entrenamiento del perceptrón en el algoritmo 4.3.1 siempre converge.

Comentarios:

4.3.1 Note que siempre se requiere elegir $\mu_i > 1$ en el algoritmo 3.1 del lema 4.3.1 y la prueba del teorema 4.3.1.

4.3.2 La condición (4.9) también es equivalente a:

$$A^{-1}(T\alpha^k + I) = E_k \alpha^k \text{ para } k = 1, 2, \dots, m. \quad (4.13)$$

Donde $E_k = \text{diag}[e_1^k, e_2^k, \dots, e_n^k]$ con $e_i^k > 1$ (por determinarse) para $i = 1, 2, \dots, n$ y para $k = 1, 2, \dots, m$. Del algoritmo 4.3.1 resulta $e_i^k > \mu_i$, esto implica que para μ_i grandes resultan e_i^k más grandes para (4.13). Si se elige A como la matriz identidad y

$$T\alpha^k + I = \mu\alpha^k \text{ para } k = 1, 2, \dots, m. \quad (4.14)$$

Con cualquier $e_k^i = \mu > 1$, entonces se satisface (4.9). La ecuación (4.14) ha sido empleada en [56].

4.3.3 Es claro del algoritmo 4.3.1 que se tiene la libertad de elegir la matriz $A = \text{diag}[a_1, a_2, \dots, a_n]$ con $a_i > 0$ para el sistema (4.1). Sin perder generalidad se puede elegir $a_i = 1$, i.e., se elige A como la matriz identidad en el algoritmo 4.3.1.

4.3.4 La razón de aprendizaje ρ en el algoritmo de entrenamiento del perceptrón puede ser cualquier número real positivo [30], [18]. Si $\rho = 1$ o cualquier otro entero positivo y si se toman los pesos iniciales W^i como el vector cero o cualquier vector con componentes enteras o ceros, entonces, la matriz T y el vector I obtenidos en el algoritmo 4.3.1 tendrán sólo componentes enteras. La razón de aprendizaje ρ especifica la medida (tamaño) del paso en cualquier actualización del vector de pesos durante el entrenamiento del perceptrón. Un ρ grande da pasos grandes lo cual implica una búsqueda amplia en el espacio de soluciones de W^i . En la mayoría de los casos, se elige ρ entre 0 y 1 ($0 < \rho < 1$) para que el entrenamiento del perceptrón converja rápidamente.

4.3.5 Del algoritmo 4.3.1 y del teorema 4.3.1, se puede ver que no hay restricciones sobre como algunos vectores bipolares pueden ser almacenados como memorias estables del sistema (4.1) diseñado para el presente enfoque. Esto implica que la capacidad de almacenaje (máximo número permisible m para patrones deseados) puede ser muy grande. Esto se verá en la sección 4.4, sin embargo, hay ciertas restricciones sobre patrones de memorias deseadas si uno quiere diseñar una red neuronal (4.1) con algunas restricciones preespeci-

ficadas sobre los elementos diagonales de la matriz de conexión T (cf. Comentarios 4.4.5 y 4.4.7 de la próxima sección).

4.3.6 Si se desea que el algoritmo de síntesis resulte en un sistema de la fórmula (4.1) con $I = 0$, se puede modificar (4.11) en el algoritmo 4.3.1 como sigue:

$$W^i \alpha^k \geq 0 \text{ si } \alpha_i^k = 1 \text{ y } W^i \alpha^k < 0 \text{ si } \alpha_i^k = -1$$

donde $W^i = [w_1^i, w_2^i, \dots, w_n^i]$, eligiendo A y T como en el algoritmo 4.3.1 y tomando $I = 0$, el teorema 4.3.1 también es válido.

4.3.1. Redes Neuronales con Restricciones en los Elementos de la Diagonal de la Matriz de Conexión

Los aspectos principales en la evaluación del diseño de memorias asociativas son :

- Capacidad de almacenamiento
- Memorias espurias
- Región (dominio) de atracción de memorias deseadas

Como se indicó en el comentario 4.3.5, **la capacidad de almacenamiento** (máximo número permisible de patrones deseados) en el presente caso puede ser muy grande y cuando el número de patrones de memoria deseados se vuelve demasiado grande, la matriz de conexión T obtenida con algunos de los métodos existentes (ver [19], [26], [50], [51], [53], [59], [63], [64]) es diagonalmente dominante o muy próxima a diagonalmente dominante. Una matriz de conexión T diagonalmente dominante hará que casi cualquier vector (patrón) bipolar (cualquier esquina del hipercubo), se vuelva un vector de memoria estable lo cual resulta en muchas memorias espurias (memorias estables no deseadas) lo que a su vez, implica obtener una red inútil. En la práctica, uno tiene que considerar un balance entre la capacidad de almacenaje y el número de memorias espurias.

Generalmente hablando, elementos grandes positivos en la diagonal de la matriz de conexión T resultará en una cantidad grande de memorias espurias. Las regiones de atracción de memorias deseadas serán más pequeña conforme el número total de memorias espurias sea grande en una red. Una de las metas del enfoque de síntesis es diseñar redes neuronales con restricciones, sobre los elementos en la diagonal de la matriz de conexión (v.g., cotas superiores y/o inferiores), de tal manera que el número de memorias espurias

pueda reducirse y el tamaño de las regiones de atracción de memorias deseadas puede incrementarse.

El siguiente corolario proporciona algunos indicadores sobre qué cotas inferiores pueden ser usadas para los elementos en la diagonal en la matriz de conexión T , cuando se desea que la red neuronal (4.1) tenga sólo vectores de memoria estables bipolares. Esto resulta en una generalización del teorema 5 de [7] y teorema 1 de [62] (para modelos diferentes) del presente caso.

Corolario 4.4.1

Suponga que en el sistema (4.1) $T_{ii} \geq a_i$ para $i = 1, 2, \dots, n$, donde $a_i > 0$ es el elemento i -ésimo de la diagonal de la matriz A . Entonces, todos los vectores de memoria estable del sistema (4.1) están en B^n , es decir, el sistema tendrá memorias no estables en $D^n \setminus B^n$, donde $D^n = \{x \in \mathfrak{R}^n : -1 \leq x_i \leq 1, i = 1, 2, \dots, n\}$ representa el hipercubo n -dimensional.

Definición 4.4.1

Se define la distancia de Hamming ($H(\alpha, \gamma)$) entre dos vectores bipolares α y γ como el número de bits en los que difieren α y γ , es decir, $H(\alpha, \gamma) = \sum_{i=1}^n |\alpha_i - \gamma_i|$.

El siguiente resultado resume algunas propiedades importantes para redes neuronales de la forma (4.1) con matriz de conexión T teniendo cotas superiores de los elementos diagonales.

Teorema 4.4.1

Suponga que α es un vector de memoria estable del sistema (4.1) con

$$A = \text{diag}[a_1, a_2, \dots, a_n], \text{ donde } a_i > 0 \text{ para } i=1, 2, \dots, n.$$

1) Ninguno de los vectores $\gamma \in B^n$ tal que $H(\alpha, \gamma) = 1$ puede ser vector de memoria del sistema (4.1) si $T_{ii} \leq a_i$ para $i=1, 2, \dots, n$.

2) Suponga que $T_{ii} \leq a_i$ para algún i , $\gamma \in B^n$ tal que $H(\alpha, \gamma) = 1$ y α y γ difieren en el i -ésimo bit. Entonces, la i -ésima componente del estado del sistema (4.1) se moverá en

la dirección hacia α si el sistema inicia en $x(0) = \gamma$.

Comentarios:

4.4.1 Los resultados del teorema 4.4.1 hacen muy probable que cualquier esquina del hipercubo la cual pertenezca a la vecindad inmediata (con distancia de Hamming 1) de un vector de memoria estable esté en el dominio de atracción de tal vector de memoria estable. El teorema 4.4.1 también implica que ninguno de dos patrones deseados puede tener distancia de Hamming 1 si se requiere que los elementos de la diagonal de T satisfagan $T_{ii} \leq a_i$ para $i=1,2,\dots,n$, donde $a_i > 0$ es el i -ésimo elemento diagonal de la matriz A .

4.4.2 Claramente es deseable en la práctica diseñar redes neuronales con $T_{ii} = a_i$ con $i=1,2,\dots,n$ para tomar ventaja de los resultados del corolario 4.1 y del teorema 4.4.1, donde a_i es el i -ésimo elemento diagonal de la matriz A . Las restricciones de los elementos de la diagonal de T dadas por $T_{ii} = a_i$ para $i = 1, 2, \dots, n$ serán referidas como las restricciones óptimas en este capítulo. Entonces, bajo restricciones óptimas:

- Una red neuronal tendrá sólo vectores de memoria estable bipolares
- Cualquier esquina del hipercubo el cual pertenezca a la vecindad inmediata de un vector de memoria estable no puede volverse una memoria estable que esté en el dominio de atracción de ese vector de memoria estable

Resultados experimentales muestran que las redes neuronales determinadas por (4.1) que satisfacen las restricciones óptimas, usualmente, tienen menos memorias espurias y dominios de atracción más grandes para memorias deseadas que aquellas redes que no satisfacen las restricciones óptimas.

4.4.3 Cuando $I = 0$ en el sistema (4.1), se puede mostrar que $\alpha \in B^n$ es memoria estable si y sólo si $-\alpha$ es memoria estable. Esto implica que la parte 1) del teorema 4.4.1

será verdadera $\forall \gamma \in B^n$ tal que $H(\alpha, \gamma) = 1$ o $H(\alpha, \gamma) = n - 1$ si $I = 0$ en el sistema (4.1). Un resultado similar se ha obtenido en [62] para modelos diferentes.

4.4.4 Las técnicas adaptables de Ho-Kashyap, pueden tratar con diseños óptimos en términos del tamaño de las regiones de atracción [21], [22]. Como puede verse de los comentarios previos, el enfoque presente también puede optimizar el tamaño de regiones de atracción empleando restricciones sobre los elementos de la diagonal de la matriz de conexión.

4.4.5 Del algoritmo de síntesis 4.3.1 se sabe que $T_{ii} = w_i^i + a_i \mu_i$ donde μ_i se elige de manera que $\mu_i > 1$. Cuando el w_i^i obtenido por el algoritmo 3.1 satisface que $w_i^i < 0$, se puede elegir $a_i > 0$, $T_{ii} = a_i$, y $\mu_i = 1 - w_i^i/a_i > 1$, o elegir $a_i > 0$ y $T_{ii} < a_i$, de manera que $\mu_i = [(T_{ii} - w_i^i)/a_i] > 1$. Puede probarse fácilmente que una red neuronal diseñada con el i -ésimo elemento diagonal de la matriz de conexión T satisfaciendo $T_{ii} \leq a_i$, donde $a_i > 0$ es el i -ésimo elemento diagonal de la matriz A , puede obtenerse si y sólo si $w_i^i < 0$ en el algoritmo de síntesis 4.3.1.

4.4.6 Si el algoritmo termina con $w_i^i \geq 0$ para algún i , entonces se puede repetir el entrenamiento del perceptrón con un peso inicial $W^i = [w_1^i, \dots, w_{i-1}^i, -\xi_i, w_{i+1}^i, \dots, w_{n+1}^i]$ con $\xi_i > 0$. Los experimentos muestran que usualmente terminan con $w_i^i < 0$ si tal solución existe.

La mayoría de los métodos de síntesis existentes ([19, 26, 50, 51, 53, 59, 61, 63, 64]), [67] no pueden ser aplicados si alguno o todos los elementos de la matriz de conexión están restringidos (v.g. $T_{ii} \leq a_i$ para algún $a_i > 0$) excepto los métodos desarrollados en [62] y [65] donde se usan técnicas de optimización y de programación lineal. El siguiente resultado proporciona una condición necesaria y suficiente para la existencia del diseño de una red neuronal con elementos de la diagonal de la matriz de conexión que estén acotados superiormente.

Teorema 4.4.2

Un diseño de red neuronal con el i -ésimo elemento diagonal de T satisfaciendo $T_{ii} \leq a_i$ donde $a_i > 0$ es el i -ésimo elemento de la diagonal de la matriz A , puede ser realizado si y sólo si los patrones deseados $\alpha^1, \alpha^2, \dots, \alpha^m$ con la i -ésima entrada eliminada son linealmente separables donde las dos clases están determinadas de acuerdo al i -ésimo elemento del patrón deseado.

Para usar el resultado del teorema 4.4.2 se tiene que entrenar el perceptrón mediante el algoritmo 4.3.1 para ver si el entrenamiento converge ya que, en general, no hay criterios simples para probar separabilidad lineal de 2 clases de patrones; esto produce, sin embargo, que un criterio simple se puede usar para el caso presente en el cual todos los patrones de entrenamiento están representados en el espacio bipolar. El próximo resultado proporciona una condición suficiente (criterio simple) para la existencia del diseño de la red neuronal con matriz de conexión T teniendo cotas superiores sobre los elementos diagonales.

Teorema 4.4.3

Sean los vectores

$$Y = [\alpha^1 : \alpha^2 : \dots : \alpha^m], \quad (4.15)$$

$$Y^i = [Y \text{ con el } i - \text{ésimo renglón borrado}] \quad (4.16)$$

Si se cumple que el

$$\text{rango}(Y) = \text{rango}(Y^i) \quad \text{para algún } i, 1 \leq i \leq n. \quad (4.17)$$

Entonces el algoritmo de síntesis 4.3.1 puede conducir al diseño de una red neuronal con el i -ésimo elemento diagonal de T satisfaciendo $T_{ii} \leq a_i$ donde $a_i > 0$ es el i -ésimo

elemento de la diagonal de la matriz A .

Comentario 4.4.7 Es claro del teorema 4.4.3 que si $\text{rango}(Y) = n$, entonces la condición (4.17) en el teorema nunca puede ser satisfecha. Es decir, si se desea diseñar una red neuronal (4.1) con algunas restricciones preespecificadas sobre los elementos diagonales de la matriz de conexión T para que puedan conseguirse menos memorias espurias y dominios de atracción más grandes para memorias deseadas. Es deseable que los patrones prototipo satisfagan la condición: $\text{rango}(Y) < n$, donde Y se define como en (4.17)

4.3.2. Redes Neuronales con Restricciones de Simetría y Escasez Sobre la Estructura de Interconexión

En esta sección se discutirán dos restricciones de conectividad, a saber, simetría y escasez.

La restricción de escasez sobre la estructura de interconexión de (4.1), en general, puede ser expresada como elementos cero preespecificados en la matriz de conexión T en locaciones dadas [55], [60].

Definición 4.5.1

Una matriz $S = [S_{ij}] \in \mathfrak{R}^{n \times n}$ se dice que es una matriz indizada si satisface

$$S_{ij} = 1 \text{ o } 0.$$

Definición 4.5.2

La restricción de la matriz $F = [f_{ij}] \in \mathfrak{R}^{n \times n}$ a una matriz indizada S , denotada por $F \setminus S$, esta definida por $F \setminus S = [h_{ij}]$, donde

$$h_{ij} = \left\{ \begin{array}{l} f_{ij}, \text{ si } S_{ij} = 1 \\ 0, \text{ de otra forma} \end{array} \right\} \quad (4.18)$$

Definición 4.5.3

El sistema (4.1) se dice que es una red neuronal con matriz de coeficiente escasa si $T = T \setminus S$ para alguna matriz indizada $S \in \mathfrak{R}^{n \times n}$. Note que la matriz indizada S determina la estructura de interconexión de una red neuronal retroalimentada. Es también de notarse que para una elección especial de la matriz S , la red neuronal (4.1) tendrá la misma estructura que la de una red neuronal celular.

El siguiente problema con escasez ha sido considerado en [53]- [56].

Problema de diseño con escasez: Dada una matriz indizada $n \times n$ $S = [S_{ij}]$ con $S_{ii} = 1$ para $i = 1, 2, \dots, n$, y m vectores $\alpha^1, \alpha^2, \dots, \alpha^m$ en B^n , elegir $\{A, T, I\}$ con $T = T \setminus S$ de tal manera que $\alpha^1, \alpha^2, \dots, \alpha^m$ sean vectores de memoria estables del sistema (4.1).

El algoritmo de síntesis 4.3.1 desarrollado en la sección 4.3 puede ser modificado para resolver el problema de diseño con escasez.

Algoritmo de diseño con escasez 4.5.1

Usando el algoritmo de entrenamiento del perceptrón obtener n vectores de pesos

$W^i = [w_1^i, w_2^i, \dots, w_{n+1}^i]$ con $i = 1, 2, \dots, n$, donde w_j^i se pone en cero si $S_{ij} = 0$, de manera que

$$W^i \bar{\alpha}^k \geq 0 \quad \text{si } \alpha_i^k = 1 \text{ y}$$

$$W^i \bar{\alpha}^k < 0 \quad \text{si } \alpha_i^k = -1$$

Para $k = 1, 2, \dots, m$, donde $\bar{\alpha}^k$ se define como

$$\bar{\alpha}^k = \begin{bmatrix} \alpha^k \\ 1 \end{bmatrix} \quad \text{para } k = 1, 2, \dots, m.$$

A, T e I se eligen de la misma manera que en el algoritmo 4.3.1.

El Algoritmo de diseño con escasez es una modificación del algoritmo 4.3.1, en el cual se necesita poner $w_j^i = 0$ si $S_{ij} = 0$.

La convergencia del entrenamiento de los perceptrones en el algoritmo de diseño con escasez puede probarse siguiendo líneas similares a las de la prueba del teorema 4.3.1 considerando $S_{ii} = 1$ para $i = 1, 2, \dots, n$.

Si se desea realizar un diseño de redes neuronales con escasez con restricciones sobre los elementos diagonales de la matriz de conexión T , los siguientes resultados proporcionan una condición necesaria y suficiente y un criterio simple para la existencia de tal diseño. Sus pruebas siguen líneas similares a las pruebas de los teoremas 4.4.2 y 4.4.3.

Corolario 4.5.1:

Suponga que en la matriz indizada S , $S_{ii} = 1$ para $i = 1, 2, \dots, n$. Un diseño de red neuronal con escasez con el i -ésimo elemento diagonal de T satisfaciendo $T_{ii} \leq a_i$ donde $a_i > 0$ es el i -ésimo elemento de la diagonal de la matriz A , puede realizarse si y sólo si los patrones $\alpha^k \setminus S_i^t$ con la i -ésima entrada eliminada son linealmente separables para $k = 1, 2, \dots, m$; donde S_i es el i -ésimo renglón de la matriz indizada S y las dos clases están determinadas de acuerdo al i -ésimo elemento de los patrones deseados.

Corolario 4.5.2:

Suponga que en la matriz indizada S , $S_{ii} = 1$ para $i = 1, 2, \dots, n$. Sean S_i el i -ésimo renglón de S ,

$$Q = [\alpha^1 \setminus S_i^t : \alpha^2 \setminus S_i^t : \dots : \alpha^m \setminus S_i^t] \quad (4.19)$$

y

$$Q^i = [Q \text{ con el } i - \text{ésimo renglón borrado}] \quad (4.20)$$

Si $\text{rango}(Q) = \text{rango}(Q^i)$ para algún i , $1 \leq i \leq n$, entonces el algoritmo de diseño con escasez puede conducir a un diseño con el i -ésimo elemento diagonal de T satisfaciendo

$T_{ii} \leq a_i$ donde $a_i > 0$ es el i -ésimo elemento de la diagonal de la matriz A .

Los parámetros de perturbación debidos a errores de implementación se representan por:

$$\Delta A = \text{diag}[\Delta a_1, \Delta a_2, \dots, \Delta a_n], \quad \Delta T \in \mathfrak{R}^{n \times n} \text{ y } \Delta I \in \mathfrak{R}^n. \quad (4.21)$$

El sistema (4.1) con parámetros de perturbación puede escribirse como

$$\dot{\chi} = - (A + \Delta A)x + (T + \Delta T)\text{sat}(x) + (I + \Delta I) \quad (4.22)$$

Donde A , T , I y $\text{sat}(x)$ se definen como en (4.1).

Para modelos con perturbación descritos por (4.22) el siguiente resultado de análisis de robustez ha sido establecido en [53] y [56]. La notación

$$\partial(x) = \min_{1 \leq i \leq m} \{ |x_i| \} \text{ para } x \in \mathfrak{R}^n \quad (4.23)$$

será usada en lo que sigue.

Lema 4.5.1

Suponga que $\alpha^1, \alpha^2, \dots, \alpha^m$ en B^n son vectores de memoria estable del sistema (4.1) y suponga que $\beta^1, \beta^2, \dots, \beta^m$ son puntos de equilibrio asintóticamente estables del sistema (4.1) correspondientes a $\alpha^1, \alpha^2, \dots, \alpha^m$, respectivamente, o sea:

$$\beta^k = A^{-1}(T\alpha^k + I) \quad \text{Para } k = 1, 2, \dots, m. \quad (4.24)$$

si

$$\mu = \min_{1 \leq k \leq m} \{ \partial(\beta^k) \} > 1. \quad (4.25)$$

Entonces $\alpha^1, \alpha^2, \dots, \alpha^m$ son también vectores de memoria estables del sistema (4.22) con tal que

$$\left\|A^{-1} \Delta A\right\|_{\infty} + \left\|A^{-1} \Delta T\right\|_{\infty} + \left\|A^{-1} \Delta I\right\|_{\infty} < \mu - 1 \quad (4.26)$$

donde $\|\cdot\|_{\infty}$ denota la norma de matrices inducida por la norma de vectores l_{∞} .

El lema 4.1 implica que los puntos de equilibrio β^k asintóticamente estables correspondientes a α^k después de la perturbación de parámetros satisface que:

$\beta^k = (A + \Delta A)^{-1} [(T + \Delta T)\alpha^k + (I + \Delta I)] \in C(\alpha^k)$ para $k=1,2,\dots,m$, dado que (4.22) se satisface. Recuérdese que la norma de matrices inducida por la norma de vectores l_{∞} para una matriz $F = [f_{ij}] \in \mathfrak{R}^{n \times n}$ está definida por

$$\|F\|_{\infty} = \max_{1 \leq i \leq m} \left\{ \sum_{j=1}^n |f_{ij}| \right\}. \quad (4.27)$$

Es claro del lema 4.1 que los algoritmos 4.3.1 y 4.5.1 garantizan que $\alpha^1, \alpha^2, \dots, \alpha^m$ sean vectores de memoria estables de (4.22) con tal que (4.26) sea satisfecha, donde μ está dada por (4.25) y los β^k están dados por (4.24). Note que los algoritmos 4.3.1 y 4.5.1 garantizan que:

$$\mu = \min_{1 \leq k \leq m, 1 \leq i \leq n} \{e_i^k\} \quad (4.28)$$

De acuerdo al comentario 4.3.2, donde $e_i^k \geq \mu_i$. Note, además que el procedimiento descrito en el comentario 4.4.6 para obtener w_i^i negativos puede ser repetidamente utilizado con ξ_i grandes los cuales permiten elegir $T_{ii} \leq a_i$ y para elegir μ_i grandes en los algoritmos 4.3.1 y 4.5.1.

Lo anterior, permite lograr una cota superior para la imprecisión de los parámetros encontrados en la implementación de un diseño dado para almacenar un conjunto deseado

de patrones bipolares en el sistema (4.1). Esta cota puede ser controlada por el diseñador en el procedimiento diseñado. Específicamente, los algoritmos de síntesis evocados arriba incorporan dos aspectos que son muy importantes en la implementación VLSI de redes neuronales artificiales:

- Permite al diseñador elegir una estructura de interconexión para la red neuronal
- Esta toma en cuenta las imprecisiones que trae la realización de redes neuronales por hardware

Para la T y la I determinadas por el algoritmo de síntesis 4.3.1 con $\mu_i > 1$, sea $\Delta T = (T^t - T)/2$. De esta manera, se tiene que $T_s \triangleq T + \Delta T = (T + T^t)/2$ es una matriz simétrica.

Del lema 4.5.1 se nota que si $\|A^{-1} \Delta T\|_\infty = \|A^{-1}(T^t - T)\|_\infty / 2 < \mu - 1$, donde μ se obtiene de (4.25) y donde se supone $\Delta A = 0$ y $\Delta I = 0$, la red neuronal (4.1) también almacenará todos los patrones deseados como memorias, con una matriz de conexión simétrica $T + \Delta T = T_s$.

La observación anterior aumenta la posibilidad de diseñar una red neuronal (4.1) con estructura de interconexión preespecificada y con una matriz de conexión simétrica. (Note que en estos casos se requiere que $S = S^t$). Tal capacidad es de gran interés ya que la red neuronal (4.1) será globalmente estable cuando T sea simétrica [7], [55]. Estabilidad global significa que para cualquier estado inicial, la red siempre converge a algún punto de equilibrio asintóticamente estable y las soluciones periódicas o caóticas no existen. Del algoritmo 4.5.1 presentado arriba usualmente resulta una matriz T no simétrica. El procedimiento desarrollado en [17] para diseño simétrico puede ser usado para determinar una matriz de conexión T_s la cual satisface que $T_s \setminus S$ para $S = S^t$. Por último, el procedimiento de diseño simétrico se resume abajo.

Algoritmo de diseño simétrico 4.5.2

Dados m vectores $\alpha^1, \alpha^2, \dots, \alpha^m$ en B^n los cuales son almacenados como vectores de memoria estables para el sistema (4.1) y S una matriz indizada $S = S^t$ con $S_{ii} = 1$ para $i = 1, 2, \dots, n$, se procede como sigue para obtener un diseño simétrico.

1. De acuerdo al algoritmo 4.5.1, determinar $A, T = T \setminus S$ e I para el sistema (4.1) con $\mu_i > 1$.
2. Calcular $\beta^k = A^{-1}(T\alpha^k + I)$ para $k = 1, 2, \dots, m$ y sea $\mu = \min_{1 \leq k \leq m} \{\partial(\beta^k)\}$.
3. Si $T = T^t$ o $\mu \leq 1 + \eta$, donde η es un número pequeño positivo (por ejemplo, $\eta = 0.01$) parar, de otra manera ir al paso 4.
4. Evaluar $\Delta T = (T^t - T)/2$. Si $\|A^{-1}\Delta T\|_\infty < \mu - 1$, elija $\lambda = 1$. Si no es así, elegir $\lambda = \frac{\mu - 1}{\|A^{-1}\Delta T\|_\infty} - \varepsilon$ donde ε es un número positivo pequeño (podría ser, $\varepsilon = 0.001$).

Tomar $\tilde{T} = T + \lambda\Delta T$.

5. Calcular $\tilde{\beta}^k = A^{-1}(\tilde{T}\alpha^k + I)$ para $k = 1, 2, \dots, m$ y determinar $\nu = \min_{1 \leq k \leq m} \{\partial(\tilde{\beta}^k)\} > 1$.
6. Reemplace μ por ν y T por \tilde{T} , ir al paso 3. Si termina con $T = T^t$, se ha encontrado una solución para el problema de diseño simétrico. Si termina con $\mu < 1 + \eta$ y $T \neq T^t$, el procedimiento de diseño simétrico es insuficiente para encontrar una T simétrica para el problema dado.

Capítulo 5

DISEÑO Y APLICACIÓN DE UNA RED NEURONAL CELULAR A UN PROBLEMA DE INGENIERÍA

En este capítulo se diseña una red neuronal celular de orden 2×1 para usarla en la aplicación de memorias asociativas para clasificar un conjunto de señales obtenidas de un nodo de una red eléctrica. En la sección 5.1 se diseña la red neuronal celular (RNC), tomando en cuenta la definición de RNC y las restricciones que deben satisfacerse para la misma [16],[9], [7]. Luego se utiliza la metodología empleada por Derong Liu y Zanjun Lu que aparece en el capítulo 4, para poder usarla en reconocimiento de patrones. En la sección 5.2 se usa dicha red para clasificar patrones (el estado de la señal) después de indicarse el procedimiento empleado para ello y de establecerse el criterio de clasificación.

5.1. Diseño de una Red Neuronal Celular Basada en el Uso de Memorias Asociativas Para Aplicarla en Reconocimiento de Patrones

Básicamente, el diseño de la red se obtiene empleando la metodología usada por Derong Liu y Zanjun Lu, y la teoría (definición y propiedades) de redes neuronales celulares presentados en [16],[9], [7]. Partiendo de esto, se diseña una red neuronal celular que realice memorias asociativas, para obtener la clase de red neuronal celular que se requiere. Es necesario considerar dos tipos de redes neuronales: las correspondientes a redes neuronales retroalimentadas empleadas por Derong Liu que designaremos como tipo 1, o sea, aquellas cuya dinámica están descritas por las ecuaciones (4.1) y las redes neuronales celulares que denotamos como tipo 2, con el conjunto de ecuaciones dadas por (3.2).

Antes que nada, tiene que determinarse si es posible aplicar el algoritmo de diseño empleado por Derong Liu a redes neuronales celulares, ya que el diseño se realizó para ciertas redes neuronales con retroalimentación que no son, necesariamente, redes neuronales celulares. Para lograrlo, hay que ver bajo que circunstancias una red tipo 1 puede convertirse en una red tipo 2, o viceversa. Esto se puede hacer de dos formas: una de ellas es imponer ciertas condiciones sobre los componentes involucrados; la otra, es por comparación.

Se procede a comparar los dos tipos de redes neuronales, por un lado, tenemos las redes neuronales celulares determinadas por las ecuaciones (3.2) [16] y, por el otro, a las redes neuronales con retroalimentación determinadas por las ecuaciones (4.1).

Para obtener la clase de red neuronal celular que se requiere, se consideran las ecuaciones que describen las dinámicas de las redes neuronales que están determinadas por las

ecuaciones (4.1) y (3.2). Tomando las ecuaciones (3.2) [16] para compararlas con las ecuaciones (4.1), partiendo de esto, se diseña una red neuronal celular que realice memorias asociativas. Las ecuaciones (4.1) están dadas por:

$$\begin{aligned}\dot{x} &= -Ax + T * sat(x) + I_\theta \\ y &= sat(x)\end{aligned}$$

y las ecuaciones (3.2) con $1 \leq i \leq M$; $1 \leq j \leq N$, son:

Ecuación de Estado :

$$C \frac{dv_{xij}(t)}{dt} = \frac{-1}{R_x} v_{xij}(t) + \sum_{C(k,l) \in Nr(i,j)} A(i,j;k,l) v_{ykl}(t) + \sum_{C(k,l) \in Nr(i,j)} B(i,j;k,l) v_{u_{kl}} + I_\alpha \quad (3.2 a)$$

Ecuación de Salida :

$$v_{yij}(t) = \frac{1}{2} (|v_{xij}(t) + 1| - |v_{xij}(t) - 1|), \quad (3.2 b)$$

Si en las ecuaciones (3.2a) y (3.2b) se hacen los siguientes cambios de variables $x = v_x$, $y = v_y$, $u = v_u$, y tomando $C = 1$, $R_x = 1$ y A una matriz simétrica, entonces las ecuaciones (3.2a) y (3.2b) pueden expresarse de la siguiente forma:

$$\dot{x} = -I_2 x + A * sat(x) + Bu + I_\alpha \quad (5.a)$$

$$y = sat(x).$$

Como la red que se va a aplicar es una red neuronal celular cuya dinámica está determinada por las ecuaciones (4.1), se requiere incluir matrices de orden 2x2 y vectores 2x1,

por el tipo de datos con los que se trabaja. Por lo que los valores permitidos para i, j son 1 y 2 en lo que corresponde al diseño de la red. Por lo tanto, se elige $A = I_2$ la matriz identidad de orden 2x2. De esta forma el sistema (4.1) queda expresado como:

$$\dot{x} = -I_2 x + T * sat(x) + I_\theta \quad (5.b)$$

$$y = sat(x).$$

Para aplicar la metodología de Liu a redes neuronales celulares se requiere hacer ciertos ajustes. Para ello, primero se comparan las ecuaciones (5.a) y (5.b) y después, igualando términos se obtiene lo siguiente:

1) La matriz coeficiente de los estados en ambas ecuaciones es el negativo de la matriz identidad I_2 .

2) La matriz A de (5.a) debe corresponder a la matriz de conexión T de (5.b), de donde se tiene que $T = A$ es una matriz simétrica, por lo que $T_{12} = T_{21}$.

3) El término $Bu + I_\alpha$ de (5.a) corresponde al vector bias I_θ de (5.b), luego $I_\theta = Bu + I_\alpha$.

Teniendo en cuenta lo anterior, la ecuación (5.a) puede expresarse como:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &+ \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} I_{\alpha 1} \\ I_{\alpha 2} \end{bmatrix} \end{aligned} \quad (5.1)$$

donde el vector bias I_θ está dado por:

$$\begin{bmatrix} I_{\theta 1} \\ I_{\theta 2} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} I_{\alpha 1} \\ I_{\alpha 2} \end{bmatrix}.$$

Se retomamos el problema de diseño y el algoritmo de diseño empleada por D. Liu y Z. Lu en la metodología para el diseño de la red neuronal celular. Se considera el caso de orden 2x1 debido a que los datos a clasificar están en R^2 .

Problema de Diseño (Síntesis)

Dados 2 vectores α^1, α^2 en B^2 , encontrar $\{A, T, I_\theta\}$ de manera que:

- 1) α^1, α^2 sean vectores de memoria estable del sistema (3.1), para este problema, $\alpha^1 = [1, 1]^t$ y $\alpha^2 = [-1, -1]^t \in B^2$
- 2) El sistema no tenga soluciones caóticas ni oscilatorias
- 3) El número total de vectores de memoria espurios (vectores de memoria no deseados) sea lo más pequeño posible, y el dominio de atracción de cada memoria deseada sea lo más grande posible

El punto 1) del problema de síntesis puede garantizarse eligiendo $\{A, T, I_\theta\}$ tal que cualquier α^i satisfaga la condición (3.7) del lema 4.3.1, o sea, $A^{-1}(T \alpha^k + I_\theta) \in C(\alpha^k)$.

El punto 2) puede conseguirse diseñando una red neuronal con matriz de conexión simétrica.

El punto 3) puede asegurarse parcialmente construyendo los elementos de la diagonal de la matriz de conexión.

Algoritmo de Diseño 4.3.1

Usando el algoritmo de entrenamiento del perceptrón, obtener 2 vectores de pesos (para 2 perceptrones) $W^i = [w_1^i, w_2^i, w_{2+1}^i]$, para $i = 1, 2$, tal que:

$$\begin{aligned} W^i \bar{\alpha}^k &\geq 0 & \text{si } \alpha_i^k = 1 & \text{ y} \\ W^i \bar{\alpha}^k &< 0 & \text{si } \alpha_i^k = -1, \end{aligned}$$

donde $\bar{\alpha}^k$ está definida por:

$$\bar{\alpha}^k = \begin{bmatrix} \alpha^k \\ 1 \end{bmatrix} \text{ para } k = 1, 2. \quad (4.12)$$

Elegir $A = \text{diag}[a_1, a_2]$ con $a_i > 0$,

Elija $T_{ij} = w_j^i$ si $i \neq j$, para $i, j = 1, 2$.

$T_{ii} = w_i^i + a_i \mu_i$ con $\mu_i > 1$ y $I_i = w_{n+1}^i$.

Para la aplicación de este algoritmo, en este problema, es necesario que los patrones de entrenamiento del perceptrón sean vectores $\alpha^k \in B^2$ con la razón de aprendizaje del perceptrón $\rho > 0$ y los α^k pertenecen a las clases X_1 (correspondiente a $z = 1$) o X_2 (correspondiente a $z = -1$). Aquí, la i -ésima componente de α^k determina si $\alpha^k \in X_1$ o $\alpha^k \in X_2$.

Tomando $\alpha^1 = [1, 1]^t$ y $\alpha^2 = [-1, -1]^t$ como patrones de entrenamiento del perceptrón se aplica el algoritmo de diseño 4.3.1 para obtener las matrices A , T y el vector I_θ .

El teorema 4.3.1 asegura que las matrices A y T y el vector I_θ obtenidos por este algoritmo satisfacen el punto 1) del problema de diseño, esto es, los α^k son vectores de memoria estable.

De acuerdo con el comentario 4.4.2, en la práctica es deseable diseñar redes neuronales con $T_{ii} = a_i > 0$ para aprovechar las ventajas de los resultados del corolario 4.4.1 y del teorema 4.4.1, donde a_i es el i -ésimo elemento de $A = \text{diag}[a_1, a_2]$. Según el comentario 3.4.5, del algoritmo 4.3.1 se tiene que los elemento diagonales de T están dados por $T_{ii} = w_i^i + a_i \mu_i$ con $\mu_i > 1$.

Cuando los $w_i^i < 0$, obtenidos por el algoritmo 3.3.1, se pueden elegir $a_i > 0$, $T_{ii} = a_i$, $\mu_i = 1 - \frac{w_i^i}{a_i} > 1$, o bien, $a_i > 0$ y $T_{ii} < a_i$ tal que $\mu_i = \frac{T_{ii} - w_i^i}{a_i} > 1$. Lo anterior implica que $T_{ii} \leq a_i$ si y sólo si $w_i^i < 0$ para alguna i en el algoritmo 3.3.1.

El teorema 4.4.2 asegura que el diseño de redes neuronales con $T_{ii} \leq a_i$ se consigue si y sólo si los patrones deseados α^1, α^2 con la i -ésima componente eliminada son linealmente separables, donde las dos clases son determinadas de acuerdo al i -ésimo componente de los patrones deseados. Lo cual está garantizado por la elección de los vectores α^k .

Por último, si se desea que la matriz T obtenida por el algoritmo 4.3.1 sea simétrica y no lo es, entonces se utilizan los algoritmos 4.5.1 y/o 4.5.2 para que lo sea, siempre que sea posible.

Para resolver el problema de síntesis, se necesita determinar A, T e I_θ de (4.7) con $\alpha = \alpha^k, k = 1, 2$, i.e., de manera que se cumpla

$$A^{-1}(T \alpha^k + I_\theta) \in C(\alpha^k) \quad (4.9)$$

para que los puntos de equilibrio asociados a los patrones de entrenamiento pertenezcan a la clase de dicho patrón, es decir, $\beta^k = A^{-1}(T \alpha^k + I_\theta) \in C(\alpha^k)$. La condición (4.9) puede escribirse equivalentemente como:

$$\begin{aligned} T_i \alpha^k + I_{\theta i} &> a_i \quad \text{si } \alpha_i^k = 1, \\ T_i \alpha^k + I_{\theta i} &< -a_i \quad \text{si } \alpha_i^k = -1 \end{aligned}$$

y para $i = 1, 2$; T_i representa el i -ésimo renglón de T , $I_{\theta i}$ denota el i -ésimo elemento de I_θ y α_i^k es la i -ésima entrada de α^k . De (4.10) o equivalentemente, de (4.9) se puede obtener ahora el algoritmo de síntesis (algoritmo de diseño) basado en el algoritmo de entrenamiento del perceptrón.

Diseño de la Red Neuronal Celular

De acuerdo al algoritmo 4.3.1, primero se elije $A = \text{diag}[a_1, a_2]$ con $a_i > 0$, en nuestro caso, el sistema es el (5.1) y por lo tanto, la matriz $A = I_2$ ya está determinada. Quedan por determinar (encontrar) la matriz T y el vector I_θ . Para obtenerlos, hay que entrenar dos perceptrones aplicando el algoritmo de entrenamiento del perceptrón para obtener, en este caso, dos vectores de pesos W^i , usando como patrones de entrenamiento dos vectores de memoria estable α^k del sistema (5.1) Tales que:

$$\begin{aligned} W^i \bar{\alpha}^k &\geq 0 & \text{si } \alpha_i^k &= 1 \\ \text{y } W^i \bar{\alpha}^k &< 0 & \text{si } \alpha_i^k &= -1, \end{aligned}$$

donde $\bar{\alpha}^k$ está defininida por (4.12):

$$\bar{\alpha}^k = \begin{bmatrix} \alpha^k \\ 1 \end{bmatrix} \text{ para } k = 1, 2. \quad (4.12)$$

Una vez encontrados los W^i para $i, j = 1, 2$, se elijen $T_{ij} = w_j^i$ si $i \neq j$ y $T_{ii} = w_i^i + a_i \mu_i$ con $\mu_i > 1$ y $I_i = w_{n+1}^i$. Con esto quedan determinados T e I_θ .

Como el objetivo es clasificar, por medio de RNC, datos bidimensionales pertenecientes a dos clases distintas: estado de falla y estado de oscilación, se requieren solamente dos patrones de entrenamiento bidimensionales binarios en

$$B^2 = \{\alpha^1, \alpha^2, \alpha^3, \alpha^4\} = \{[1, 1]^t, [1, -1]^t, [-1, 1]^t, [-1, -1]^t\},$$

que no sean esquinas adyacentes del cuadrado cuyos vértices son los puntos asociados a estos vectores. Las gráficas de las clases $C(\alpha^k)$ se muestran en las figuras 15 y 16.

Por la naturaleza del problema, se toman los dos vectores $\alpha^1 = [1, 1]^t$ y $\alpha^2 = [-1, -1]^t \in B^2$ como patrones de entrenamiento, y con ellos se entrenan dos perceptrones de acuerdo

con la regla mencionada arriba. Para obtener dos vectores de pesos que resultaron iguales dado que los patrones de entrenamiento son los mismos para los dos perceptrones, los vectores que se obtienen son $W^1 = W^2 = [\rho, \rho, -\rho]$ los cuales determinan la matriz de pesos T y el vector bias I_θ , donde $\rho > 0$ es la razón de aprendizaje del perceptrón. En caso de ser necesario, se aplica el algoritmo 4.5.1 o el 4.5.2. de manera que la matriz T sea simétrica y sus elementos en la diagonal sean unos. Particularmente, para las redes 5.2, 5.3.a y 5.3.b no se requieren, necesariamente, que los elementos en la diagonal de T sean unos.

Considerando $W^1 = W^2 = [\rho, \rho, -\rho]$ para obtener la matriz T y el vector I_θ , donde $\rho > 0$. Se tienen:

las matrices $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $T = \begin{bmatrix} \rho + \mu_1 & \rho \\ \rho & \rho + \mu_2 \end{bmatrix}$ y el vector $I_\theta = \begin{bmatrix} -\rho \\ -\rho \end{bmatrix}$, fijando los valores $\mu_1 = \mu_2 = 1.5$ y $\rho = .5$ se tiene la red neuronal celular (5.2) que se deseaba, a saber:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 & \frac{1}{2} \\ \frac{1}{2} & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \text{sat}(x_1) \\ \text{sat}(x_2) \end{bmatrix}. \quad (5.2)$$

Se sabe que si β^1 y β^2 son puntos de equilibrio asintóticamente estables del sistema (5.2) correspondientes a α^1 y α^2 respectivamente, entonces $\beta^k = A^{-1}(T \alpha^k + I_\theta)$ para $k = 1, 2$; como $A = I_2$, entonces $A^{-1} = I_2$ por lo que se tiene $\beta^k = T \alpha^k + I_\theta$, teniéndose que $\beta^1 \in C(\alpha^1)$ y $\beta^2 \in C(\alpha^2)$ de acuerdo con el lema 3.3.1.

Por lo que, los puntos de equilibrio del sistema (5.2) asociados a los vectores:

$$\alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

son respectivamente

$$\beta^1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad y \quad \beta^2 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}.$$

El plano de fase correspondiente al sistema (5.2) se muestra en la figura 17. Todos los planos de fase correspondientes a las distintas redes, mencionadas en lo que resta de esta sección, se obtuvieron utilizando SIMULINK.

Otra RNC se obtiene con $\mu_1 = \mu_2 = 2$ y la razón de aprendizaje $\rho = 1$ con lo que se tiene

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \text{sat}(x_1) \\ \text{sat}(x_2) \end{bmatrix} \quad (5.3.a)$$

los puntos de equilibrio del sistema (5.3.a) asociados a los vectores:

$$\alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

son respectivamente

$$\beta^1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \quad y \quad \beta^2 = \begin{bmatrix} -5 \\ -5 \end{bmatrix}.$$

El plano de fase correspondiente al sistema (5.3.a) se muestra en la figura 18.

Es posible obtener otras ecuaciones que determinen la red neuronal celular teniendo en cuenta ciertas restricciones o condiciones para los elementos involucrados. Algunas de estas opciones se presentan en seguida. Sin embargo, en todas ellas no se aplican las restricciones del algoritmo de diseño.

La red también se logra tomando en cuenta las siguientes restricciones dadas por Chua [7]. Haciendo $B(i, j; k, l) = 1$, con $v_{uij} = E_{ij}$ constante, $C = 1$ y $R_x = 1$; y aplicando el algoritmo 4.3.1, o en su defecto, el 4.5.1 o el 4.5.2. Si se eligen $\mu_1 = \mu_2 = 1$ y la razón de aprendizaje $\rho = 1$ para el entrenamiento de los perceptrones en el algoritmo 4.3.1 se obtienen la matriz simétrica T y el vector I_θ dados por:

$$T = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad I_\theta = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Se usaron los vectores $\alpha^1 = [1, 1]$ y $\alpha^2 = [-1, -1] \in B^2$ como patrones de entrenamiento, y con ellos, se entrenaron dos perceptrones de acuerdo con la regla mencionada arriba para obtener, una vez entrenados los perceptrones, la matriz simétrica de pesos T y el vector de bias I_θ de acuerdo al algoritmo de diseño 4.3.1.

y tomando en cuenta que para

$$u = v_u = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad I_\alpha = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

se tiene de

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} I_{\alpha 1} \\ I_{\alpha 2} \end{bmatrix} = \begin{bmatrix} I_{\theta 1} \\ I_{\theta 2} \end{bmatrix}$$

que

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix},$$

de donde resulta la siguiente red neuronal celular:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} I_{\alpha 1} \\ I_{\alpha 2} \end{bmatrix},$$

o sea:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} \quad (5.3.b)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \text{sat}(x_1) \\ \text{sat}(x_2) \end{bmatrix}$$

que satisface todos los requerimientos exigidos.

Por lo tanto, los puntos de equilibrio del sistema (5.3.b) asociados a los vectores:

$$\alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

son respectivamente

$$\beta^1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad y \quad \beta^2 = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

El plano de fase correspondiente al sistema (5.3.b) se muestra en la figura 19.

En lo que resta de esta sección se suprime la ecuación de las salidas de cada sistema.

Otra opción para obtener una red neuronal celular, sin tomar en cuenta el método de diseño usado en los dos casos anteriores, se pueden conseguir siguiendo algunas de las recomendaciones de Chua [7], por ejemplo, haciendo $A(i, j; k, l) = 1$, $B(i, j; k, l) = 1$ con $i, j = 1, 2$.

$$u = v_u = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, I_\alpha = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

se tiene de

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} I_{\alpha 1} \\ I_{\alpha 2} \end{bmatrix} = \begin{bmatrix} I_{\theta 1} \\ I_{\theta 2} \end{bmatrix}$$

que

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}.$$

de donde resulta la siguiente red neuronal celular:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} \quad (5.4)$$

que satisface todos los requerimientos exigidos.

Los puntos de equilibrio del sistema (5.4) asociados a los vectores:

$$\alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

son respectivamente

$$\beta^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \beta^2 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}$$

El plano de fase correspondiente al sistema (5.4) aparece en la figura 20.

Si usamos la RNC dada por (5.4) con $I_\theta = 0$, resulta:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \quad (5.5)$$

Los puntos de equilibrio del sistema (5.5) correspondientes a los vectores:

$$\alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

son respectivamente

$$\beta^1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad y \quad \beta^2 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

cuyo diagrama de plano de fase se muestra en la figura 21.

Observaciones:

5.1 Si en el sistema (5.5), se toman $u \neq 0$ e $I_\alpha = 0$, entonces (5.5) se obtiene también con $u = [-1/2, -1/2]^t$ manteniendo fijo a I_θ .

5.2 Si se quiere que T en (5.4) satisfaga condiciones de escasez, además de simetría, se elige la matriz indizada $S = I_2$ que satisface $S = S^t$. La restricción de T a S estaría dada por $T \setminus S = T_s = I_2$ por lo que la RNC queda definida por:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} I_{1\theta} \\ I_{2\theta} \end{bmatrix} \quad (5.6)$$

donde

$$\begin{bmatrix} I_{1\theta} \\ I_{2\theta} \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

como $T = I_2$, entonces los puntos de equilibrio asintóticamente estables de (5.6) están dados por $\beta^k = \alpha^k + I_\theta$ de donde obtenemos que para

$$\alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

los puntos de equilibrio son

$$\beta^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad y \quad \beta^2 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

por lo que el sistema (5.6) con $I_\theta \neq [0, 0]^t$ no es útil para nuestra aplicación ya que $\beta^1 \notin C(\alpha^1)$ y $\beta^1 \neq \alpha^1$. Este sistema no es adecuado para la resolución del problema. Por lo tanto, no se toma en cuenta su gráfica. Si a la red (5.6) se le agrega la restricción $I_\theta = 0$ da lugar a la siguiente red:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (5.7)$$

Los puntos de equilibrio de (5.7) son de la forma $\beta^k = \alpha^k$, o sea

$$\beta^1 = \alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \beta^2 = \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

los cuales claramente no satisfacen que $\beta^k \in C(\alpha^k)$. Es evidente que la red (5.7) es la más simple de todas las obtenidas. El plano de fase correspondiente al sistema (5.7) se muestra en la figura 22.

5.3 Es importante recalcar que solamente se toman en cuenta las cuatro regiones del plano determinadas por las clases $C(\alpha^i)$ y de estas, las que realmente importan son las definidas por $C(\alpha^1)$ y $C(\alpha^2)$. Por lo que todos los pares de puntos de equilibrio mencionados en cada uno de sistemas se ubica en estas regiones del plano.

5.4 La razón por la que se descarta la RNC (5.6) es porque uno de sus puntos de equilibrio queda fuera de las regiones antes mencionadas.

Para esta aplicación, es indistinto si se usa cualesquiera de las redes neuronales celulares definidas por (5.2), (5.3.a), (5.3.b), (5.4), (5.5) o (5.7), ya que en todas ellas se obtienen regiones de atracción lo suficientemente grandes correspondientes a sus respectivos puntos de equilibrio β^1 y β^2 . Además, en todas se han usado los mismos patrones de entrenamiento (memorias deseadas) $\alpha^1 = [1, 1]^t$ y $\alpha^2 = -\alpha^1$ y en todas las RNC se tiene que: $\beta^1 \in C(\alpha^1)$ o $\beta^1 = \alpha^1$ y $\beta^2 \in C(\alpha^2)$ o $\beta^2 = \alpha^2$ que son las condiciones necesarias requeridas en el criterio de clasificación establecido en la siguiente sección. El uso de una de ellas no depende tanto de la simplicidad de la red, sino de la ubicación de los datos correspondientes a los estados de la señal. Lo cual exige que los datos correspondientes al estado de falla pertenezcan a una de las clases y los datos correspondientes al estado de oscilación pertenezcan a la otra.

En el anexo se presentan todas las figuras ordenadas por capítulos.

5.2. Aplicación de Redes Neuronales Celulares a Reconocimiento de Patrones.

Esta aplicación consiste en utilizar la red neuronal celular 2x1 diseñada en la sección 5.1 para reconocimiento de patrones (clasificar datos) usando memorias asociativas. Los datos a clasificar pertenecen a dos conjuntos de medidas de señales en un nodo de una red eléctrica, los cuales corresponden a dos estados de las señales: estado de oscilación y estado de falla.

Dado que los datos a clasificar corresponden a medidas de señales en un nodo de una red eléctrica, entonces, podemos ubicar el problema como un problema de ingeniería. De ahí el título de la tesis: **Diseño y Aplicación de una Red Neuronal Celular a un Problema de Ingeniería.**

Ahora bien, se tiene que las medidas de los datos a clasificar están dados en números complejos de la forma $z = a + bi$, esto permite representarlos como pares ordenados (a, b) en \mathbb{R}^2 , o bien, como vectores en el plano $[a, b]$. Aquí se consideran como puntos en \mathbb{R}^2 . La pregunta pertinente es ¿Qué se requiere para que una red neuronal celular diseñada de acuerdo con la metodología empleada por D. Liu pueda clasificar este tipo de datos?

Se sabe que los datos z representan medidas de señales que están en estado de oscilación o en estado de falla; denotaremos con X_1 al conjunto de medidas en estado de oscilación y con X_2 al conjunto de medidas en estado de falla. De esta manera, se tiene que si z es un dato a clasificar, entonces $z \in X_1$ o $z \in X_2$ pero no en ambos a la vez.

Para que la red neuronal celular diseñada clasifique apropiadamente a los datos originales se requiere lo siguiente:

1) Cada dato a clasificar debe ser considerado como un estado inicial $x(0)$ del sistema dinámico de la red neuronal en cuestión, es decir, $\forall z, z = x(0)$ donde $z \in X_1 \cup X_2$.

2) Si α^1 y α^2 son los vectores de memoria deseados de la red diseñada, entonces z debe pertenecer sólo a una de las clases definidas por α^1 y α^2 , o sea, $z \in C(\alpha^1)$ o $z \in C(\alpha^2)$.

3) Si se cumplen los requisitos anteriores, se asegura que z está en la región de atracción de uno de los puntos de equilibrio asintóticamente estables β^1 o β^2 , asociados a los vectores de memoria deseados de la red diseñada, lo que implica que $z \rightarrow \beta^1$, es decir que z converge a β^1 , o bien, $z \rightarrow \beta^2$.

En este caso, los datos originales no satisfacen la segunda condición. Por lo tanto, tal como se encuentran, no pueden ser clasificados por la red neuronal celular diseñada en la sección anterior, la figura 23 muestra la gráfica de dichos datos.

Con la finalidad de lograr que los datos puedan ser clasificados en forma apropiada por la red diseñada, estos deben ser transformados, es decir, hay que encontrar una transformación $T : C \rightarrow C$ del plano complejo tal que:

1) Cada dato a clasificar debe ser considerado como un estado inicial del sistema dinámico de la red neuronal en cuestión, es decir, $\forall z, T(z) = x(0)$ donde $z \in X_1 \cup X_2$.

2) Si α^1 y α^2 son los vectores de memoria deseados de la red diseñada, entonces $T(z)$ debe pertenecer solo a una de las clases definidas por α^1 y α^2 , esto es, $T(z) \in C(\alpha^1)$ o $T(z) \in C(\alpha^2)$, o en su defecto $T(z) \in C(\alpha^2)$ y/o $T(z) = \alpha^2$. Además si $T(z) \in C(\alpha^1)$ o $T(z) = \alpha^1$ implica que $z \in X_1$, o en caso contrario, si $T(z) \in C(\alpha^2)$ o $T(z) = \alpha^2$ implica que $z \in X_2$.

3) Si se cumplen los requisitos anteriores, se asegura que $T(z)$ está en la región de atracción de alguno de los puntos de equilibrio asintóticamente estables β^1 o β^2 , asociados a los vectores de memoria deseados de la red diseñada, lo que implica que $T(z)$ converge a β^1 o $T(z)$ converge a β^2 .

Por último, falta determinar el criterio de clasificación; lo que se desea es que cuando

$T(z) \in C(\alpha^1)$ se cumpla $z \in X_1$ y si $T(z) \in C(\alpha^2)$ entonces $z \in X_2$. Se puede determinar si $T(z) \in C(\alpha^k)$ con $k=1, 2$ de dos formas:

Primer criterio: Determinar el punto de equilibrio al que converge $T(z)$. Si $T(z)$ converge a β^1 , entonces $z \in X_1$ y si $T(z)$ converge a β^2 entonces $z \in X_2$.

Segundo criterio: Determinar la salida de la red, o sea, evaluar $y = sat(T(z))$. De esta manera, se tiene que si $y = sat(T(z)) = \alpha^1$ implica $z \in X_1$ y si

$$y = sat(T(z)) = \alpha^2 \text{ implica } z \in X_2.$$

Como los vectores de entrenamiento de los perceptrones

$$\alpha^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad y \quad \alpha^2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

son las memorias estables de nuestra red neuronal celular que determinan las clases $C(\alpha^1)$ y $C(\alpha^2)$, se requiere encontrar una transformación $T : C \rightarrow W$ tal que $T(z) \in C(\alpha^1)$ o $T(z) \in C(\alpha^2)$, pero no en ambas a la vez, o bien, $T(z) = \alpha^1$ o $T(z) = \alpha^2$.

Para lograr que las imágenes de los datos correspondientes al estado de falla pertenezcan a $C(\alpha^2)$ o coincidan con α^2 y las imágenes de los datos correspondientes al estado de oscilación esten en $C(\alpha^1)$ o coincidan con α^1 , se realizó una transformación T del plano C en el plano W usando MATLAB, llamamos a tal transformación `trans_plan`. Esta transformación racional aplica tres puntos distintos Z_1, Z_2 y Z_3 del plano C sobre tres puntos distintos prefijados W_1, W_2 y W_3 del plano W , respectivamente. La transformación T está dada por cualesquiera de las tres ecuaciones siguientes [68]:

$$(Z - Z_3)(W - W_1)(Z_2 - Z_1)(W_2 - W_3) = (Z - Z_1)(W - W_3)(Z_2 - Z_3)(W_2 - W_1), \quad (5.9)$$

$$A * Z * W + B * Z + C * W + D = 0, \quad (5.10)$$

$$W = (-B * Z - D) / (A * Z + C). \quad (5.11)$$

El resultado de aplicar esta transformación a los datos del problema se muestra en la figura 24 donde los datos originales aparecen en rojo, mientras que los datos transformados aparecen en azul. Esta transformación cumple los requisitos impuestos arriba para realizar la clasificación.

Con esto puede resolverse el problema. Como los datos a clasificar pertenecen a dos clases distintas se requiere una RNC que posea dos puntos de equilibrio asintóticamente estables β^1 y β^2 con regiones de atracción suficientemente grandes, y que además, los datos a clasificar (o sus imágenes bajo T) pertenezcan a las clases $C(\alpha^k)$ o a las regiones de atracción de los puntos de equilibrio β^k .

Solución del Problema:

Primero, se determina una de las **redes neuronales celulares diseñadas** para utilizarla en la clasificación de los datos, esta puede ser la RNC (5.2), (5.3.a) o (5.3.b); se elige la RNC (5.2) que tiene los puntos de equilibrio asintóticamente estables $\beta^1 = [2, 2]^t \in C(\alpha^1)$ y $\beta^2 = [-3, -3]^t \in C(\alpha^2)$ con regiones de atracción suficientemente grandes, como es deseable, ya que estos puntos de atracción tienen como dominio de atracción a un semiplano cada uno de ellos como lo muestra la figura 17.

En segundo lugar, se establece el segundo criterio de clasificación, a saber:

Si $T(z) \implies \text{sat}(T(z)) = y = 1$, entonces $z \in X_1$, es decir, el dato corresponde al estado de oscilación. Por el contrario,

Si $T(z) \implies \text{sat}(T(z)) = y = -1$, entonces $z \in X_2$, o sea, el dato corresponde al estado de falla.

Por último, se procede a relizar la clasificación de los datos ya transformados, para lo cual se procede de la siguiente manera:

1.- Se selecciona arbitrariamente cualquier dato a clasificar $T(z)$, este dato debe ser considerado como un estado inicial $x(0)$ del sistema dinámico de la red neuronal celular (5.2).

2.- Determinar la salida de la red, o sea, evaluar $y = sat(T(z))$. De esta manera, se sabe que si $y = sat(T(z)) = 1$, entonces se tiene que $T(z) \in C(\alpha^1)$ lo que indica que $z \in X_1$. Por otra parte, si $y = sat(T(z)) = -1$, se tiene que $T(z) \in C(\alpha^2)$ y $z \in X_2$, lo cual establece que z pertenece al estado de falla.

En el presente caso se satisface esta condición al aplicar la transformación del plano (5.10) sobre los datos originales z . Esta transformación es tal que permite el uso de las RNC (5.2), (5.3.a) y (5.3.b) diseñadas en la sección 5.1 y también, las redes donde no se aplica el algoritmo de diseño como son las RNC (5.4), (5.5) y (5.7). Finalmente, si se desea, se verifica si la red clasifica adecuadamente a los datos a clasificar, lo cual no es necesario ya que el diseño se hace para que esto ocurra con seguridad.

Para verificar que la red reconoce el estado de la señal, primero, se toman datos arbitrarios que pertenezcan a una de las clases y se observa que efectivamente la red establece correctamente el estado de la señal, se procede igual para datos que pertenezcan a la otra clase; lo anterior puede realizarse con datos de ambas clases. Como ejemplo se toman los siguientes valores para los vectores Z_1 y Z_2 que pertenecen al estado de falla y al estado de oscilación, respectivamente, se obtienen sus imágenes bajo T , $W_1 = T(Z_1)$ y $W_2 = T(Z_2)$ introduciéndolos como valores para los estados iniciales $x(0)$ en la red los cuales generan las salidas Y_r con $r = 1,2$.

Por ejemplo, para

$$Z_1 = [z_{11}, z_{12}, z_{13}] = [0.00697 + 0.11685 * i, 0.007057 + 0.11695 * i, 0.00735 + 0.11725 * i] \quad (5.12)$$

se tiene que su transformación $W_1 = T(Z_1)$ es

$$W_1 = [-2,644 - 1,7806 * i, -2,6411 - 1,7781 * i, -2,6208 - 1,7926 * i] \quad (5.13)$$

$$\text{con salida } Y_1 = [y_{11}, y_{12}, y_{13}] = [\text{sat}(T(z_{11})), \text{sat}(z_{12}), \text{sat}(z_{13}))]$$

$$Y_1 = [-1 - i, -1 - i, -1 - i] = [\alpha^2, \alpha^2, \alpha^2] \quad (5.14)$$

por lo que $y_{1k} = \text{sat}(T(z_{1k})) = \alpha^2 \implies z_{1k} \in X_1$, con $k = 1, 2, 3$, i.e., z_{1k} pertenecen al estado de falla.

Ahora tomando

$$Z_2 = [1.5717 + 2.2627 * i, 1.9860 + 1.1928 * i, 2.0026 + 1.5317 * i] \quad (5.15)$$

cuya transformación es $W_2 = T(Z_2)$ está dado por

$$W_2 = [9.9094 + 17.9584 * i, 14.1358 + 15.4698 * i, 13.3815 + 16.4876 * i] \quad (5.16)$$

$$\text{tiene como salida a } Y_2 = [y_{21}, y_{22}, y_{23}] = [\text{sat}(T(z_{21})), \text{sat}(z_{22}), \text{sat}(z_{23}))]$$

$$Y_2 = [1 + i, 1 + i, 1 + i] = [\alpha^1, \alpha^1, \alpha^1] \quad (5.17)$$

lo cual indica que $y_{2k} = \text{sat}(T(z_{2k})) = \alpha^1 \implies z_{2k} \in X_2$ con $k = 1, 2, 3$, i.e., z_{2k} pertenecen al estado de oscilación, con lo anterior se tiene que, efectivamente, la RNC elegida permite clasificar perfectamente a los datos, tal como se esperaba.

Bibliografía

- [1] Simon Haykin, “Neural Networks Comprehensive Foundations.” Macmillan College Publishing Company, New York. Cap.1, pp. 1-40, 1994.
- [2] R. P. Lippmann, “An Introduction to Computing With Neural Nets.” IEEE ASSAP Magazine, pp. 4-22, 1987.
- [3] B. Widrow, M. A. Lehr, “30 Years of Neural Networks: Perceptron, Madaline, and Backpropagation.” IEEE, Vol.78, No.1, pp. 1415-1442, September, 1990.
- [4] D. Hebb, “The Organization of Behavior.” John Willey & Sons, 1949.
- [5] Judith E. Dayhoff “Neural Network Architectures An Introduction.” Van Nostrand Reinhold, New York, 1990.
- [6] Fangyue Chen, Guanrong Chen, Guolong He, Xiubin Xu and Qinbin He, “Universal Perceptron and DNA-Like Learning Algorithm for Binary Neural Networks: LSBF and PBF Implementations.” IEEE Transactions on Neural Networks Vol 20, No 10, pp.1645-1658, October, 2009.
- [7] L.O. Chua & L. Yang. “Cellular Neural Networks: Applications.” IEEE Transactions on Circuits and Systems, Vol.35, No.10, pp. 1273-1289, October, 1998.

- [8] L. O. Chua and P. M. Lin. "Computer Aided Analysis of Electronics Circuits: Algorithms and Computational Techniques." Englewood Clifs, N.J. Prentice Hall, 1975.
- [9] L. O. Chua & L. Yang. "Cellular Neural Networks: Theory." IEEE Transactions on Circuits and Systems, Vol.35, No. 10, pp. 1257-1272, October, 1998.
- [10] Leonid I. Perlovsky, "Neural Mechanisms of the Mind, Aristotle, Zadeh, and fMRI." IEEE Transactions on Neural Networks Vol 21, No 5, pp.718-733, May, 2010.
- [11] K. Preston Jr. and M. J. B. Duff. "Special Issue on Chaotic Systems." Proc. IEEE, August, 1987.
- [12] L. O. Chua and R. N. Madan "The sights and sounds of chaos" IEEE Circuits Devices Mag., pp. 3-13, Jan. 1998.
- [13] A. Rosenfeld and A.C. Kak "Digital Picture Processing." New York Academic, 1982.
- [14] Cong Wang, David J.Hill, "Deterministic Learning and Rapid Dynamical Pattern Recognition." IEEE Transactions on Neural Networks Vol 18, No 3, pp.617-630, May, 2007.
- [15] D. Liu & Z. Lu "Associative Memory Design via Perceptron Learning." IEEE Transactions on Circuits and Systems, Vol.35, No. 10, pp. 1273-1289, October, 1998.
- [16] Derong Liu and Zanjun Lu "A New Synthesis Approach for Feedback Neural Networks Based on the Perceptron Training Algorithm." IEEE Transactions on Neural Networks, Vol.8, No. 6, pp. 1468-1882, November, 1997.
- [17] F. Rossenblatt "The Perceptron: A probabilistic model for information storage and organization in the brain." Psychological Review, 65, pp. 386-408, 1958.

- [18] F. Rosenblatt "Principles of Neurodynamics" New York, N.Y. Spartan, 1962.
- [19] S. R. Das "On the synthesis of Nonlinear Continuous Neural Networks." IEEE Transactions on Systems, Man & Cybernetics, vol. 21, pp. 413-418, Marzo. 1991.
- [20] Charlotte Yuk-Fan Ho, Bingo Wing-Kuen Ling and, Hak-Keung Lam and Muhammad H. U. Nasir, "Global Convergence and Limit Cycle Behavior of Weights of Perceptron." IEEE Transactions on Neural Networks Vol 19, No 6, pp.938-947, June, 2008.
- [21] M. H. Hassoun "Associative Neuronal Memories: Theory and implementation." Oxford, U. K. Oxford Univ. Press, 1993.
- [22] M. H. Hassoun and A. M. Youssef, "Associative Neuronal Memories Capacity and Dynamics." in proc. Int. J. Conf. Neural Networks, San Diego, CA, vol.1, June, 1990, pp.763-769.
- [23] T. Kohonen "Associative Memory: A system theoretical approach." Springer-Verlag, 1997.
- [24] T. Kohonen "An Introduction to Neural Computing" Neural Networks, Vol, pp. 3-16, 1988.
- [25] Florent Cousseau, Tomoko Ozeki and Shun-ichi Amari, "Dynamics of Learning in Multilayer Perceptrons Near Singularities." IEEE Transactions on Neural Networks Vol 19, No 8, pp.1313-1328, August, 2008.
- [26] J. J. Hopfield "Neural Networks and physical systems with emergent collective computational abilities." National Academy of sciences, 81, pp. 2554-2558, 1982.

- [27] J. J . Hopfield “Neurons with graded response have colective computational properties like those of two-state neurons.” In proc. National Academy of sciences, 81, pp.3088-3092, 1984.
- [28] S. Grossberg “Studies of Mind and Brian Neural Principles of Learning Perception, Development, Cognition and Motor Control.” Ed. Reidel Press, Amsterdam, 1982.
- [29] W. S. McCulloch & W. A. Pitts “A Logical Calculus of the Ideas Inmanent in Nervous Activity” Bouulletin of mathemartics and Biophysics, 5, pp. 115-133, 1943.
- [30] M. L. Minsky & S.A. Papert “Perceptrons: An Introduction to computational Geometry.” Cambridge, MA: MIT Press, 1969 y 1988 (expanded version).
- [31] José R. Hilera , Víctor J. Martínez “Redes Neuronales Artificiales Fundamentos modelos y aplicaciones.” Addison-Wesley Iberoamericana, 1995.
- [32] J. Anderson & E. Rosenfeld “Neurocomputing.” Ed. MIT Press, pp. 126-134, 1988.
- [33] J. Anderson, J. Silverstein, S. Ritz y R. Jones “Distinticve features, categorical perception and probability learning: some aplicaciones on a neural model.” Psychological Review, 84, pp. 413-451, 1977.
- [34] R. Hecht-Nielsen “Neurocomputing: Picking the Human Brain.” IEEE Spectrum, 25, pp. 36-41, Marzo, 1988.
- [35] G. Hinton, D. Ackley “Boltzmann machines: Constraint satisfaction networks than learn.” Carnegie-Mellon Depaetament of Computer Science Technical Report, 1984.
- [36] Y. Le Cunn “Modeles connexionnistes de l’apprentissage.” Ph. D. Thessis. Université Pierre et Marie Curie, París, VI, 1987.

- [37] Y. Ho and R. L. Kashyap, "An Algorithmic for linear inequalities and its applications," *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 683-688, 1965.
- [38] D. Parker "Learning Logic." Invention Report, S81-64, File 1. Office of technology Licensing, Stanford University, 1982.
- [39] A.J. Maren, C.T. Harston y R.M. Pap "Handbook of Neural Computing Applications." Ed. Academic Press, 1990.
- [40] P. Werbos "Beyond Regression: New tools for prediction and analysis in the behavioral sciences." Ph. D. Thesis. Harvard University, 1974.
- [41] R. Perfeti. "A neural networks to design neural networks." *IEEE Transactions on Circuits and Systems*, vol. 38, pp. 1099-1103, Sep. 1991.
- [42] A. Barto y P. Anandan "Pattern recognizing stochastic learning automata." *IEEE Transactions on Systems, Man & Cybernetics*, 15, pp. 360-375, 1985.
- [43] A. Barto, R. Sutton y C. Anderson "Neuron-like adaptive elements than can solve difficult learning control problems." *IEEE Transactions on Systems, Man & Cybernetics*, 13, pp. 834-846, 1983.
- [44] K. Narendra y M. Thathacher "Learning automata. A survey." *IEEE Transactions on Systems, Man & Cybernetics*, 4, pp. 323-334, 1974.
- [45] R. Rohwer y B. Forrest "Training time-dependence in neural networks. " In *Proceedings of the IEEE First International Conference on Neural Networks*, Vol. II, pp. 701-708, 1987.
- [46] F. Pineda "Generalization of Back-propagation to recurrent neural networks" *Physical Review Letters*, 59, pp. 2229-2232, 1987.

- [47] L. Almeida “A learning rule for asynchronous perceptrons with feedback in a combinatorial environment.” Proceedings of the IEEE First International Conference on Neural Networks, Vol. II, pp. 609-618, 1987.
- [48] H. Szu “Fast simulated annealing.” AIP Conference Proceedings 151: Neural Networks for Computing (J. Denker ed.), pp. 420-425, 1986.
- [49] S. Hui and S.H. Zak. “Dynamical analysis of the Brian-state in a box (BSB) neural models.” IEEE Trans. Neural Networks, Vol. 3, pp. 86-94, Jan. 1992.
- [50] J. H. Li, A. N. Michel, and W. Porod “Qualitative analysis and synthesis of a class of Neural Networks.” IEEE Transactions on Circuits and Systems, vol. CAS-35, pp. 976-986, Aug. 1988.
- [51] J. H. Li, A. N. Michel, and W. Porod “Analysis and synthesis of a class of Neural Networks: Linear system operating on a closed hypercube.” IEEE Transactions on Circuits and Systems, vol. 36, pp. 1405-1422, Nov. 1989.
- [52] D. Liu and A. N. Michel “Asymptotic stability of systems operating on a closed hypercube” Systems Control Lett., vol. 19, pp. 281-285, Oct. 1992.
- [53] D. Liu and A. N. Michel “Dynamical Systems with Saturation Nonlinearities: Analysis and Design .” New York, N.Y. Springer-Verlag, 1994.
- [54] D. Liu and A. N. Michel “Cellular Neural Networks for Associative Memories.” IEEE Transactions on Circuits and Systems. II, vol. 40, pp. 119-121, Feb. 1993.
- [55] D. Liu and A. N. Michel “Sparsely Interconnected Cellular Neural Networks for Associative Memories with Applications to Cellular Neural Networks.” IEEE Transactions on Circuits and Systems. II, vol. 41, pp. 295-307, Apr. 1994.

- [56] D. Liu and A. N. Michel “Robustness analysis and design of a class of Neural Networks with Sparse Interconnecting structure.” *Neurocomputing*, vol. 12, pp. 59-76, June, 1996.
- [57] A. N. Michel , J. A. Farrell “Associative Memories via artificial Neural Networks.” *IEEE Control Systems Mag.*, vol. 10, pp. 6-17, Apr. 1990.
- [58] A. N. Michel and J. A. Farrell and H.F. Sun “Analysis and synthesis techniques for Hopfield type synchronous discrete time Neural Networks with applications to associative memory.” *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 1356-1366, Nov. 1990.
- [59] A. N. Michel, J. Si and G. Yen “Analysis and synthesis of a class of discrete time Neural Networks described on hipercubes.” *IEEE Transactions Neural Networks* , vol. 2, pp. 36-46, Jan. 1991.
- [60] A. N. Michel , K. Wong “Qualitative limitations incurred in implementations of recurrent neural networks” *IEEE Control Systems Mag.*, vol. 15, no. 3, pp. 52-65, June 1995.
- [61] R. Perfetti “A neural network to design neural networks.” *IEEE Transactions on Circuits and Systems*, vol. 38, pp. 1099-1103, Sep. 1991.
- [62] R. Perfetti “A synthesis procedure for brain state in a box neural networks.” *IEEE Transactions Neural Networks* , vol. 6, pp. 1071-1080. Sep. 1995.
- [63] L. Personnaz, I. Guyon and G. Dreyfus “Information storage and retrieval in spin glass like neural networks .” *J. Phys. Lett.*, vol. 46, pp.1,359-365. Apr. 1985.

- [64] L. Personnaz, I. Guyon and G. Dreyfus “Collective computational properties of neural networks: New learning mechanisms.” *Phys. Rev. A*, vol. 34, pp. 4217-4228, Nov. 1986.
- [65] G. Seiler, A. J. Schuler, and J. A. Nossek “Design of robust cellular neural networks.” *IEEE Transactions on Circuits and Systems I*, vol. 40, pp. 358-364, May. 1993.
- [66] Enrique Romero and Josep María Sopena, “Performing Feature Selection With Multi-layer Perceptrons.” *IEEE Transactions on Neural Networks* Vol 19, No 3, pp.431-441, March, 2008.
- [67] Z. B. Xu, J.Q. Hu and C.P. Kwong “Some Efficient Estrategies for Improving the eigen estructure method in syntesis of feedback neural networks.” *IEEE Transactions Neural Networks* , vol. 7, pp. 233-245. Jan. 1996.
- [68] R.V. Churchill, and James W. Brown “Variable Compleja y Aplicaciones.” Mc.Graw Hill, 5^a edición, España, pp. 242-245, 1995.
- [69] Clifford Lau “Neural Network Theoretical Foundations and Analysis.” IEEE PRESS, New York, 1992.

NAME="GraphicsSaveCONTENT=" 32»

LISTA DE FIGURAS

Figuras	Página
Figura 1.- Neurona y Sus Componentes	130
Figura 2.- Gráfica De La Función Escalón	130
Figura 3.- Gráfica De La Función Lineal	130
Figura 4.- Gráfica De La Función Mixta	130
Figura 5.- Gráfica De La Función Sigmoidal	131
Figura 6.- Gráfica De La Función Gaussiana	131
Figura 7.- Distintas Capas y Tipos de Neuronas En Una Red Neuronal	131
Figura 8.- Red Neuronal Celular (RNC) Bidimensional	132
Figura 9.- Vecindades De Las Celdas Centrales C(2,2) y C(3,3)	132
Figura 10.- Ejemplo Del Circuito de Una Celda	133
Figura 11.- Gráfica de la Función Característica	
Figura 12.- Gráfica de la Función Característica Del Resistor En Una Celda en un Circuito Equivalente	133
Figura 13.- Circuito Equivalente Del Estado Estable De Una Celda En Una RNC	133
Figura 14.- Rutas Dinámicas y Puntos de Equilibrio Del Circuito Equivalente	134
Figura 15.- Regiones Correspondientes a Las Clases C[α]	134
Figura 16.- Regiones Dadas Por Las Clases C[α]	134
Figura 17.- Diagrama Del Plano De Fase Del Sistema 5.2	135
Figura 18.- Diagrama Del Plano De Fase Del Sistema 5.3.a	135
Figura 19.- Diagrama Del Plano De Fase Del Sistema 5.3.b	135
Figura 20.- Diagrama Del Plano De Fase Del Sistema 5.4	135
Figura 21.- Diagrama Del Plano De Fase Del Sistema 5.5	135
Figura 22.- Diagrama Del Plano De Fase Del Sistema 5.6	135
Figura 23.- Diagrama Del Plano De Fase Del Sistema 5.7	135
Figura 24.- Gráfica De Datos Originales	136
Figura 25.- Gráfica De Datos Originales y Transformados	136

LISTA DE SIMBOLOS

Símbolos	Páginas
1.- X, Y	Vectores o matrices 3
2.- $C[\alpha]$	Clase Del Vector α 3
3.- ϵ	Es elemento de (pertenece a) 3
4.- $X = [a_1, a_2, \dots, a_n]$	Vector n-dimensional 14
5.- \sum	Sumatoria 15
6.- $f(t)$	Función de t 16
7.- (a, b)	Intervalo abierto de a hasta b 18
8.- $*$	Multiplicación (.) 21
9.- df/dt	Derivada de f con respecto a t 26
10.- \leq	Es menor o igual que 28
11.- $\max\{a, b\}$	El valor máximo entre a y b 40
12.- $\forall x$	Para toda x 40
13.- $ a $	El valor absoluto de a 41
14.- $>$	Es mayor que 41
15.- $K \Omega$	Kilo Ohms 43
16.- s	Segundos 43
17.- V	Volts 44
18.- $\int f(t) dt$	Integral de f(t) 47
19.- $\lim_{t \rightarrow \infty} f(t)$	El límite de f(t) cuando t tiende a infinito 47
20.- $A \times B$	El producto Cartesiano (cruz) de conjuntos 54
21.- $\mathcal{R} \times \mathcal{R} \times \dots \times \mathcal{R} = \mathcal{R}^n$ (n veces \mathcal{R}) 59
22.- A'	La transpuesta de la matriz A 59
23.- $\ F\ $	La norma del vector (o función) F 62
24.- \geq	Mayor o igual que 65

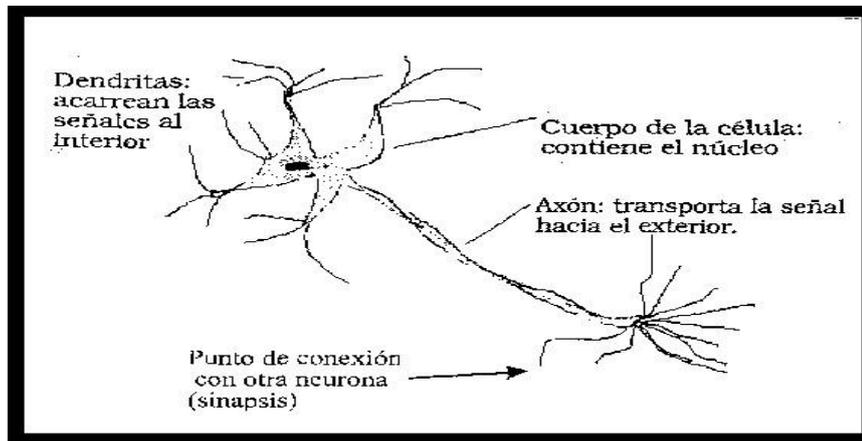
ANEXO

Esta sección se divide en dos partes, en la primera se muestran todas las gráficas y figuras referidas en los distintos capítulos de la tesis; en la segunda parte se hacen algunas aclaraciones acerca del sistema de ecuaciones diferenciales que se usa para el diseño de la red neuronal celular.

PRIMERA PARTE

FIGURAS DEL CAPÍTULO 2

Figura 1.- Neurona y sus Componentes.



GRAFICAS DE FUNCIONES DE TRANSFERENCIA

Figura 2.- Gráfica de la Función Escalón.

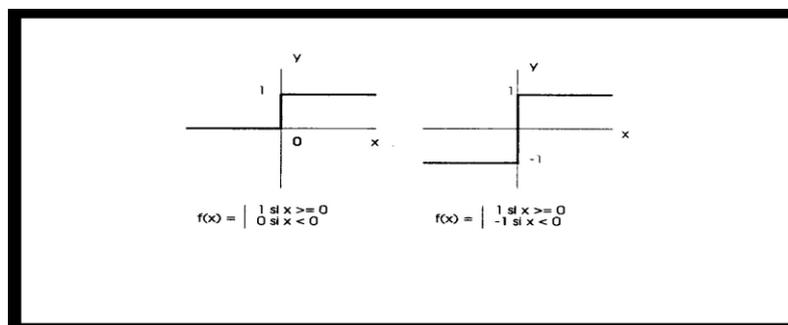


Figura 3.- Gráfica de la Función Lineal $f(t) = t$.

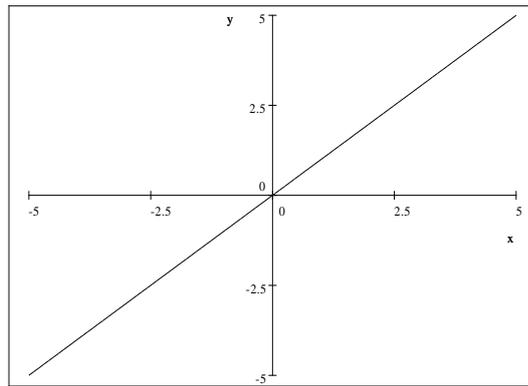


Figura 4.- Gráfica de la Función Mixta.

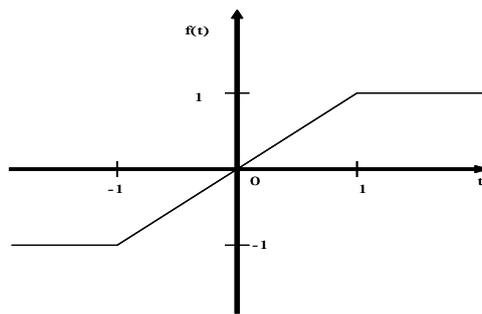


Figura 5.- Gráfica de la Función Sigmoideal $f(x) = 1/(1 + \exp(-x))$.

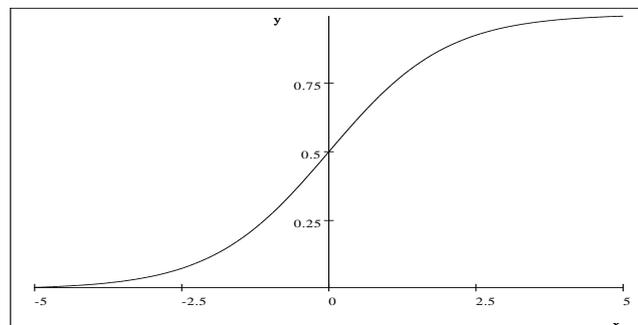


Figura 6.- Gráfica de la Función Normal (Gaussiana).

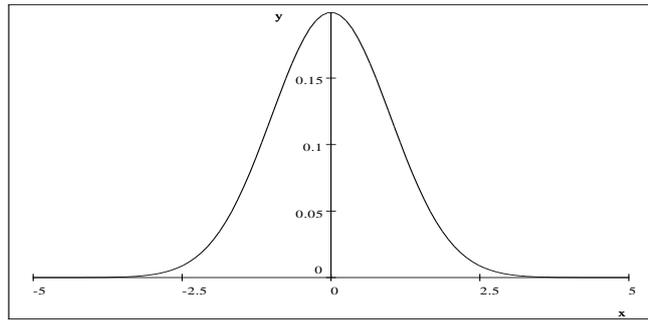
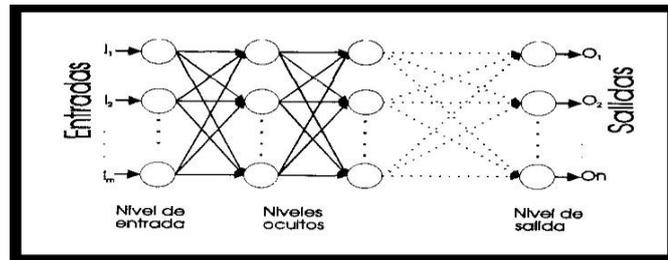
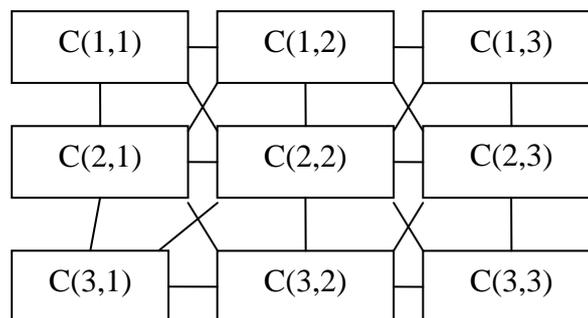


Figura 7.- Distintas Capas de una Red Neuronal y Tipos de Neuronas: de entradas, ocultas y de salida.



FIGURAS DEL CAPÍTULO 3

Figura 8.- Red Neuronal Celular (RNC) Bidimensional.



El tamaño del circuito es 3x3. Los cuadrados son unidades de circuitos llamados celdas. Las conexiones entre las celdas indican que hay interacción entre ellas.

Figura 9.- Las Vecindades de las Celdas C(2, 2) para $r=1$ y C(3, 3) con $r=2$, respectivamente.

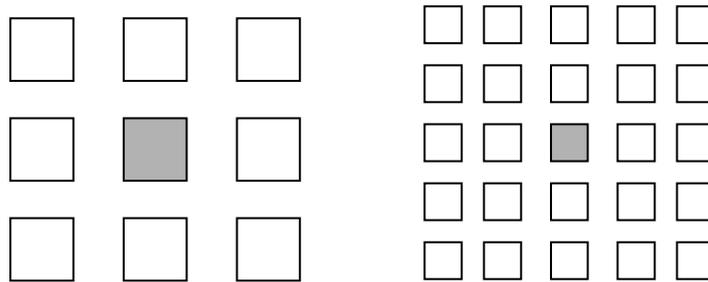
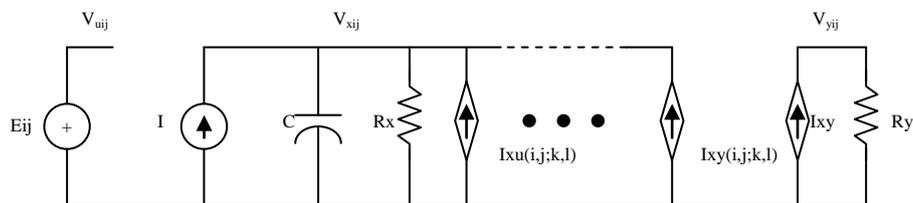


Figura 10.- Ejemplo del Circuito de una Celda.



C es un capacitor lineal; R_x , R_y y R_u son resistores lineales; I es una fuente de voltaje independiente; $I_{xu}(i, j ; k , l)$ y $I_{xy}(i , j ; k, l)$ son fuentes de corriente de voltaje controlado lineales con funciones características:

$I_{xy}(i, j ; k, l) A(i,j;k,l)v_{ykl}$ y $I_{xu}(i,j;k,l)B(i,j;k,l)v_{ukl}$ para toda $C(i, j)$ en $N(i, j)$; $I_{xy}(1/2R_y)f(v_{xij})$ es una fuente de voltaje controlado lineal por pedazos con función característica $f(\cdot)$ como se muestra en la figura 3.4; E_{ij} es una fuente independiente de voltaje.

Figura 11.- Gráfica de la Función Característica $f(v) = (1/2) (|v+1| - (|v-1|))$.

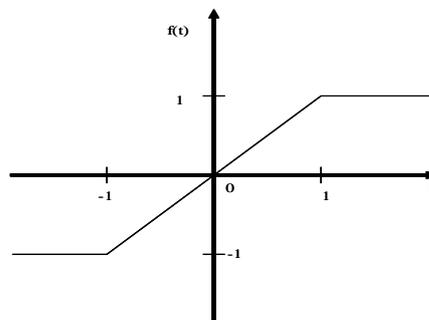


Figura 12.- Gráfica de la Función Característica del Resistor no Lineal de una Celda en un Circuito Equivalente.

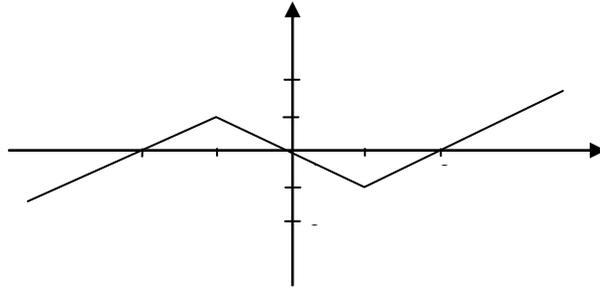


Figura 13.- El Circuito Equivalente de Estado Estable de una Celda en una Red Neuronal Celular.

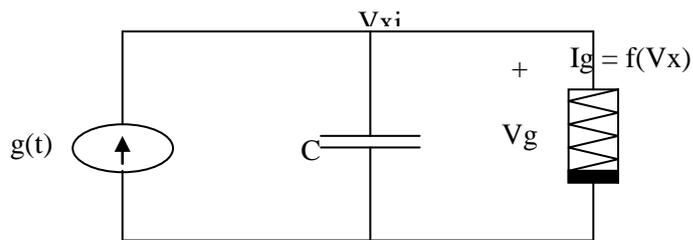
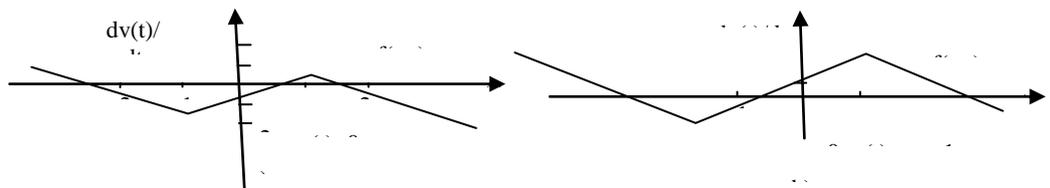
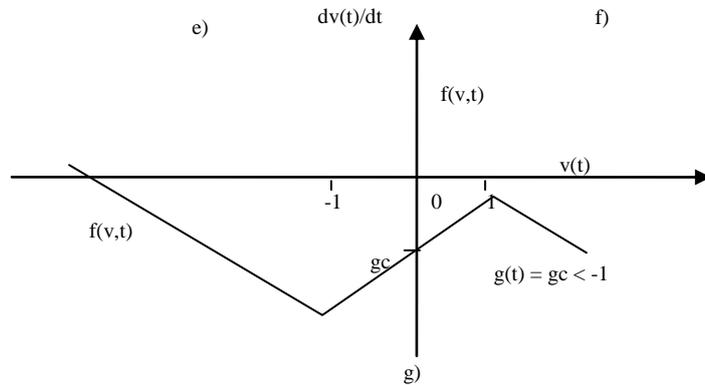
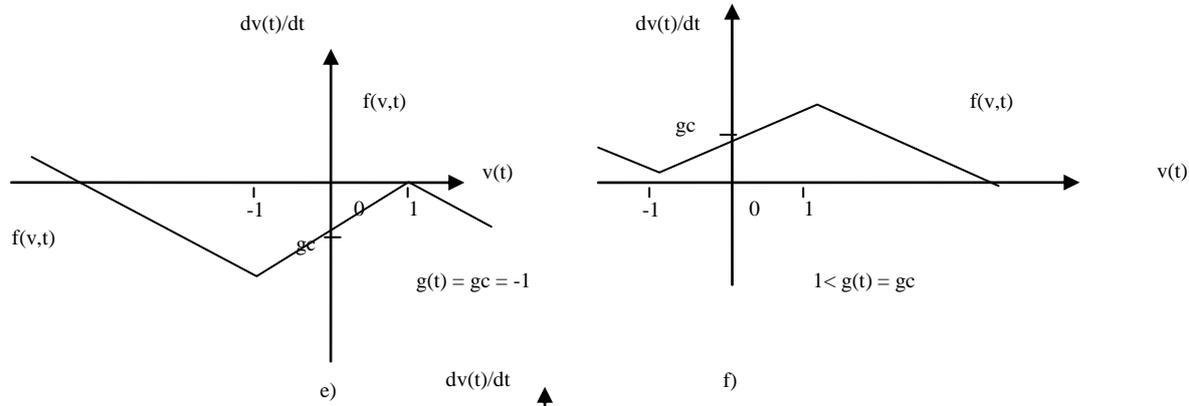
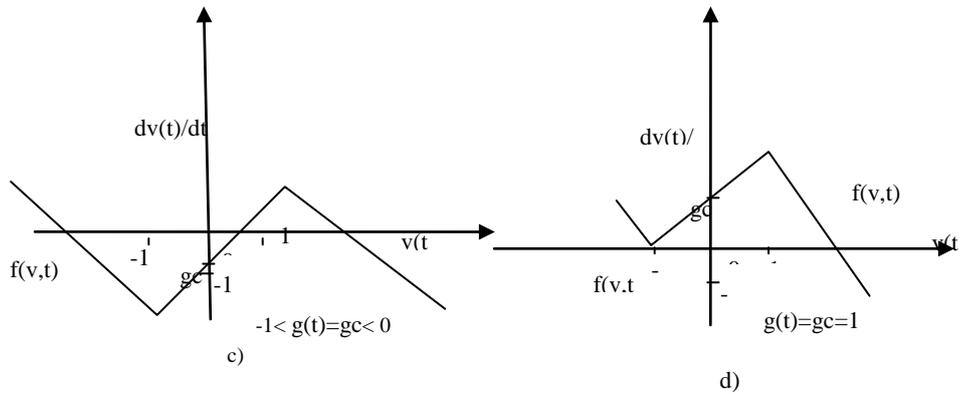


Figura 14.- Rutas Dinámicas y Puntos de Equilibrio del Circuito equivalente. En las gráficas 14(a) hasta 14(g) se muestran las rutas dinámicas y los puntos de equilibrio del circuito equivalente para diferentes valores de $g(t)$.



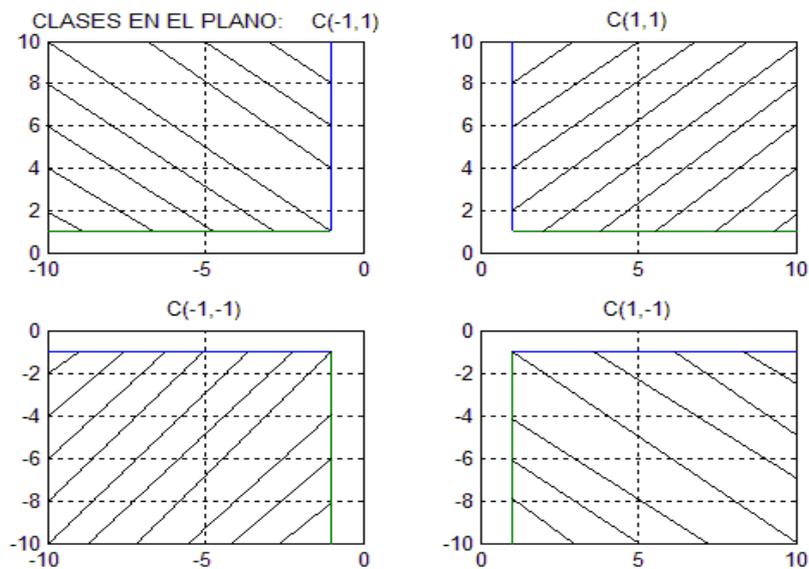


FIGURAS DEL CAPÍTULO 4

Figura 15.- En esta gráfica se muestran las cuatro regiones determinadas por las clases $C(\alpha)$ con $\alpha = (1,1)$, $(-1,1)$, $(-1,-1)$ y $(1,-1)$ respectivamente. Las regiones cuadrículadas indican las dos clases importantes.



Figura 16.- Regiones dadas por las clases $C(\alpha)$ con $\alpha = (1,1)$, $(-1,1)$, $(-1,-1)$ y $(1,-1)$ respectivamente. Por cuestiones de claridad, se muestran separadas.



FIGURAS DEL CAPITULO 5

Figura 17.- Diagrama del Plano de Fase del Sistema 5.2.

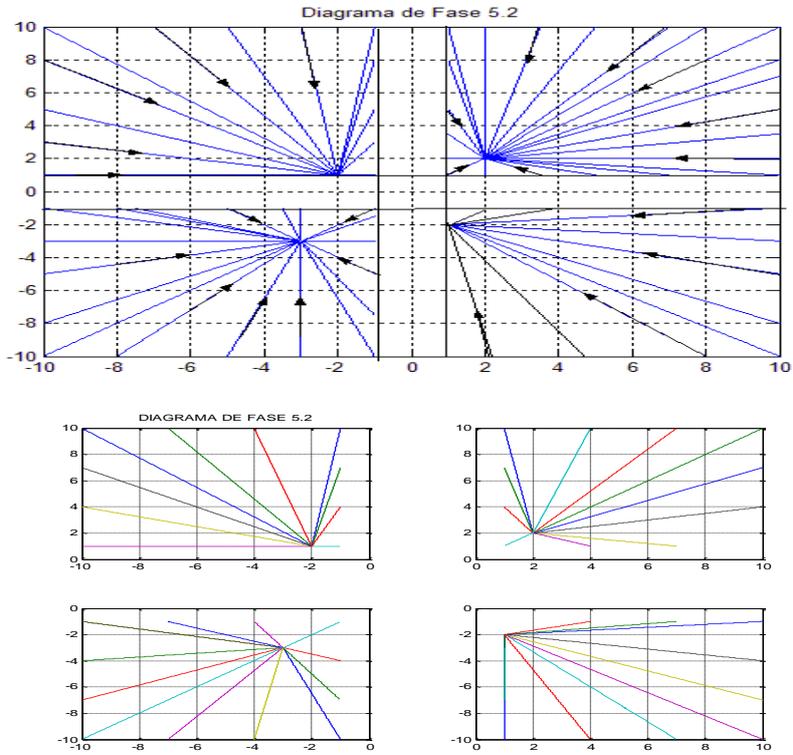


Figura 18.- Diagrama del Plano de Fase del Sistema 5.3.a.

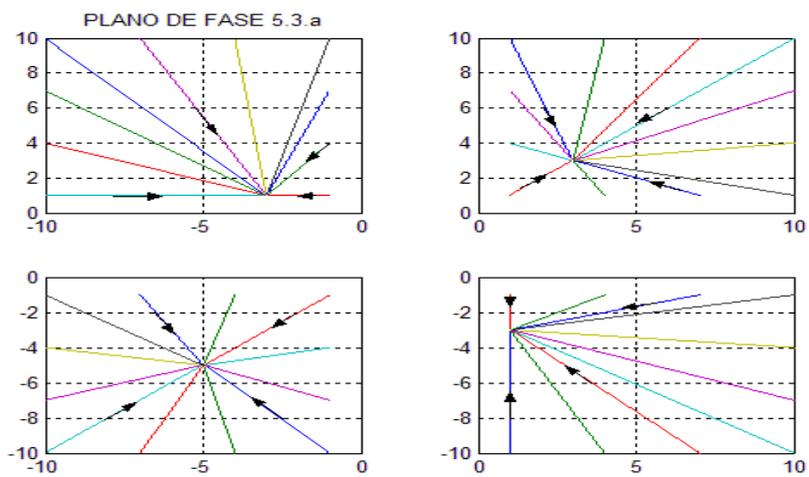


Figura 19.- Diagrama del Plano de Fase del Sistema 5.3.b.

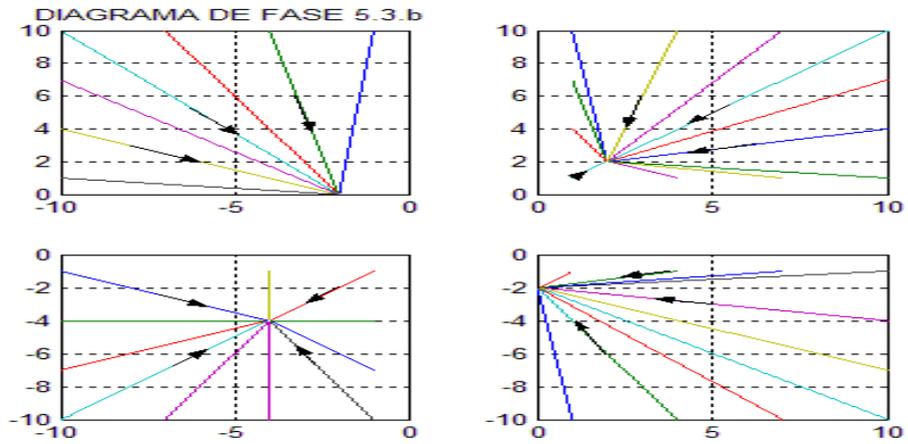


Figura 20.- Diagrama del Plano de Fase del Sistema 5.4.

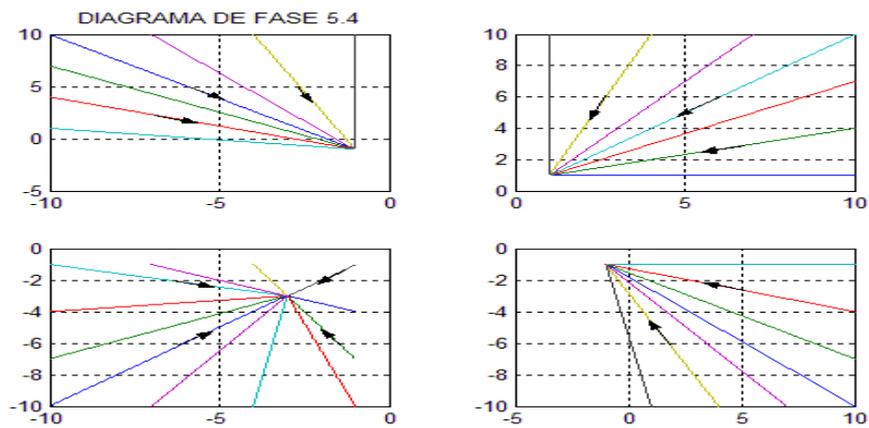


Figura 21.- Diagrama del Plano de Fase del Sistema 5.5.

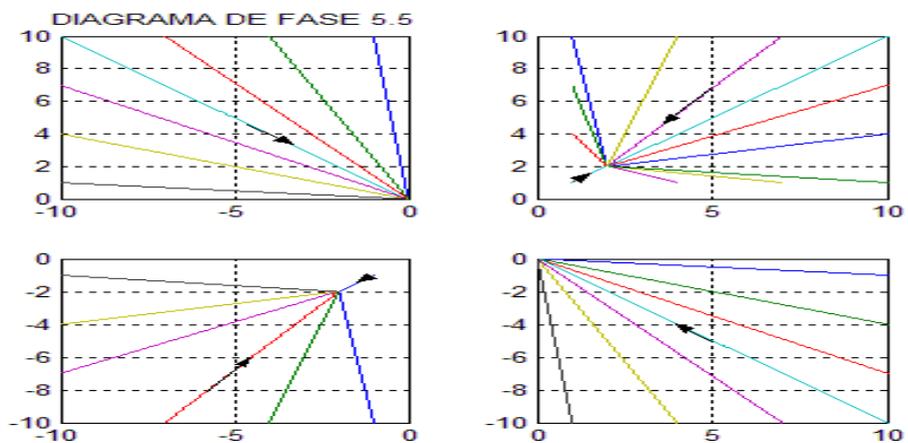
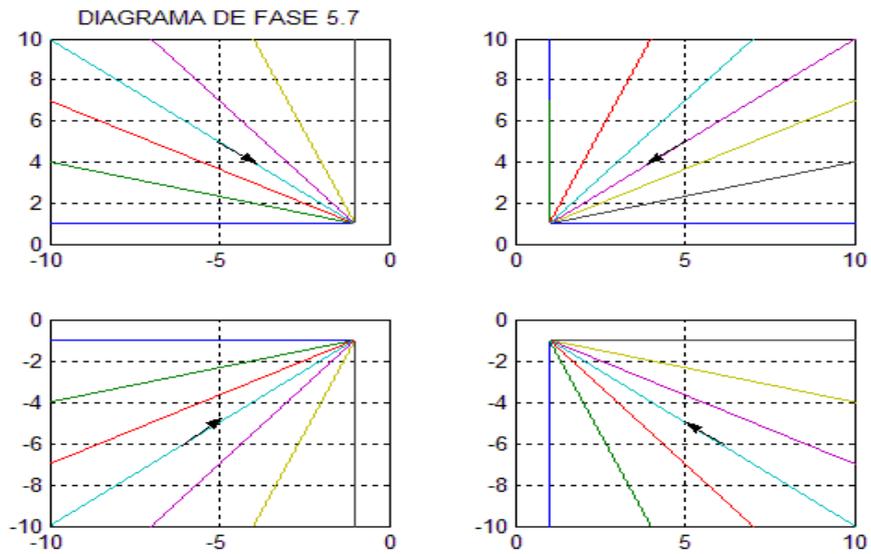
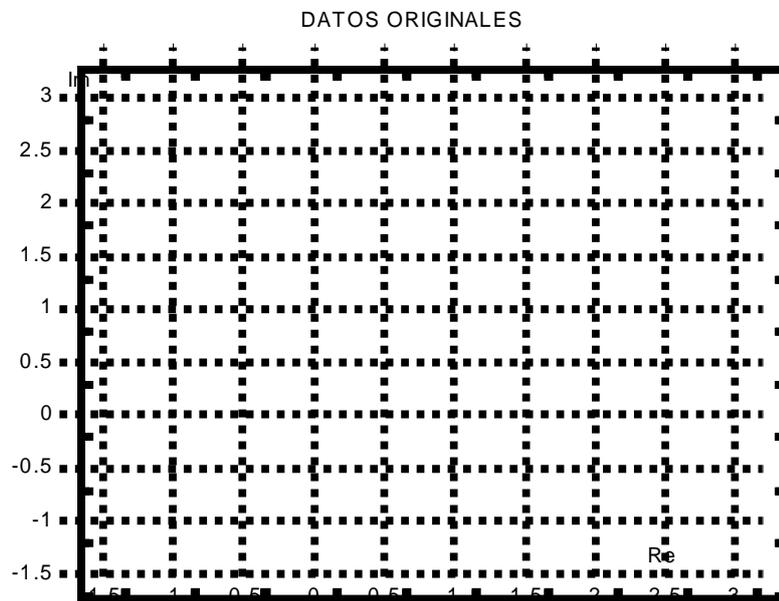


Figura 22.- Diagrama del Plano de Fase del Sistema 5.7.



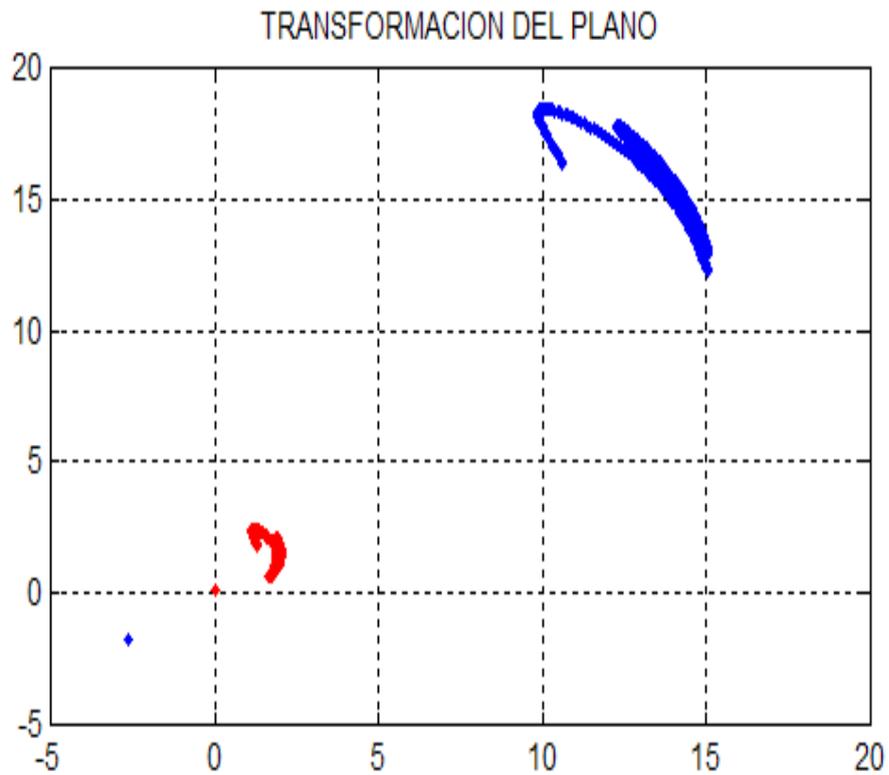
En las siguientes gráficas se muestran los conjuntos de datos a clasificar.

Figura 23.- Representación Gráfica de los Datos Originales.



El conjunto de datos correspondiente al estado de falla aparecen acumulados en lo que parece un punto, observe que no pertenecen a ninguna de las clases $C(i,j)$. Mientras que los datos asociados al estado de oscilación, se representan en lo que parece una curva. Parte de estos datos pertenecen a la clase $C(1,1)$ y la parte restante no pertenece a ninguna otra clase.

Figura 24. Representación Gráfica de los Datos Originales y Transformados.



En esta gráfica, los datos originales aparecen en rojo, mientras que los datos transformados se muestran en azul. Al igual que en la gráfica anterior, la clase de datos en estado de falla (F) se acumulan en lo que parece un "punto", en tanto que los datos asociados al estado de oscilación se acumulan en lo que parece una "curva". Note que los datos (F), ya transformados, pertenecen a la clase C (-1,-1) y los de estado de oscilación a la de C (1,1) como se deseaba.

SEGUNDA PARTE

En primer lugar, se tiene que el sistema de ecuaciones diferenciales, asociado a la RNC, está dado por:

$$\begin{aligned} X &= -Ax + T \operatorname{sat}(x) + I, \\ y &= \operatorname{sat}(x). \end{aligned}$$

Donde A es la matriz identidad, T es una matriz simétrica y la función $\operatorname{sat}(x)$ está dada por $\operatorname{sat}(x) = (1/2)(|x+1| - |x-1|)$.

De lo anterior, se tiene que el sistema queda expresado de la siguiente manera:

$$\begin{aligned} x_1' &= -x_1 + c_1 \operatorname{sat}(x_1) + c_2 \operatorname{sat}(x_2) + I_1, \\ x_2' &= -x_2 + c_2 \operatorname{sat}(x_1) + c_3 \operatorname{sat}(x_2) + I_2. \end{aligned}$$

Las dos ecuaciones tienen la misma forma. Con la finalidad de obtener una solución y por cuestiones de simplicidad se toman los elementos de la matriz simétrica $c_1 = c_3 = 1 = c_2 = 0$ y $I_1 = 0 = I_2$ para reducir el sistema a una ecuación diferencial. Usando la notación tradicional resulta la ecuación:

$$y' = -y + (|y+1| - |y-1|). \text{ Donde } y = f(t) \text{ es una función de la variable } t.$$

La solución exacta obtenida usando SWP v.3 esta dada por: $y^2 - [*] + t = C$.

Donde $[*] = \ln(y - |y+1| + |y-1|) / (y^2 - 1 - |y+1|y + |y+1| + |y-1|y + |y-1|)$.

Lo cual indica que la solución no es fácil de obtener directamente.

En segundo lugar, como la función $\operatorname{sat}(x)$ está definida por:

$$\operatorname{Sat}(x_i) = \{1, \text{ si } x_i > 1; x_i, \text{ si } -1 \leq x_i \leq 1; -1, \text{ para } x_i < -1 \text{ con } i=1, 2\}$$

De ahí se observa que para cada evaluación de x_1 se tienen tres posibles valores de x_2 , por lo que se tiene que el plano \mathbb{R}^2 queda dividido en 9 regiones ajenas entre sí, las cuales quedan definidas por:

$$\begin{aligned} R_1 &= \{(x_1, x_2): x_1 > 1 \text{ y } x_2 > 1\}, \\ R_2 &= \{(x_1, x_2): x_1 > 1 \text{ y } -1 \leq x_2 \leq 1\}, \\ R_3 &= \{(x_1, x_2): x_1 > 1 \text{ y } x_2 < -1\}, \\ R_4 &= \{(x_1, x_2): -1 \leq x_1 \leq 1 \text{ y } x_2 > 1\}, \\ R_5 &= \{(x_1, x_2): -1 \leq x_1 \leq 1 \text{ y } -1 \leq x_2 \leq 1\}, \\ R_6 &= \{(x_1, x_2): -1 \leq x_1 \leq 1 \text{ y } x_2 < -1\}, \\ R_7 &= \{(x_1, x_2): x_1 < -1 \text{ y } x_2 > 1\}, \\ R_8 &= \{(x_1, x_2): x_1 < -1 \text{ y } -1 \leq x_2 \leq 1\} \text{ y} \\ R_9 &= \{(x_1, x_2): x_1 < -1 \text{ y } x_2 < -1\}. \end{aligned}$$

Por lo tanto, el sistema de ecuaciones diferenciales:

$$x'1 = -x1 + c1 * \text{sat}(x1) + c2 * \text{sat}(x2) + I1$$

$$x'2 = -x1 + c2 * \text{sat}(x1) + c3 * \text{sat}(x2) + I2.$$

Queda determinado por la región donde sea ubicado el par $(x1, x2)$, lo que da lugar a nueve sistemas de ecuaciones diferenciales.

Para la aplicación en la tesis solo son de interés cuatro de estas regiones a saber: R1, R3, R7 y R9. Se tendrán cuatro sistemas de ecuaciones para cada RNC, uno en c/u de las regiones mencionadas.

Así se tienen los sistemas indicados para cada región:

Para la RNC (5.2):

$$R1: x'1 = -x1 + 2 \text{ y } x'2 = -x2 + 2;$$

$$R3: x'1 = -x1 + 1 \text{ y } x'2 = -x2 - 2;$$

$$R7: x'1 = -x1 - 2 \text{ y } x'2 = -x2 + 1;$$

$$R9: x'1 = -x1 - 3 \text{ y } x'2 = -x2 - 3.$$

Igualmente para la RNC (5.3):

$$R1: x'1 = -x1 + 2 \text{ y } x'2 = -x2 + 2;$$

$$R3: x'1 = -x1 \text{ y } x'2 = -x2 - 2;$$

$$R7: x'1 = -x1 - 2 \text{ y } x'2 = -x2;$$

$$R9: x'1 = -x1 - 4 \text{ y } x'2 = -x2 - 4.$$

De la misma forma en RNC (5.5):

$$R1: x'1 = -x1 + 2 \text{ y } x'2 = -x2 + 2;$$

$$R3: x'1 = -x1 \text{ y } x'2 = -x2;$$

$$R7: x'1 = -x1 \text{ y } x'2 = -x2;$$

$$R9: x'1 = -x1 - 2 \text{ y } x'2 = -x2 - 2.$$

Finalmente en la RNC (5.7) se tienen:

$$R1: x'1 = -x1 + 1 \text{ y } x'2 = -x2 + 1;$$

$$R3: x'1 = -x1 + 1 \text{ y } x'2 = -x2 - 1;$$

$$R7: x'1 = -x1 - 1 \text{ y } x'2 = -x2 + 1;$$

$$R9: x'1 = -x1 - 1 \text{ y } x'2 = -x2 - 1.$$

El total de ecuaciones diferenciales distintas que aparecen en los diferentes sistemas y sus soluciones son:

$$x' = -x + 2; \quad y' = -y + 2, \text{ la solución exacta es: } y(t) = 2 + C1 e^{-t};$$

$$x' = -x + 1; \quad y' = -y + 1, \text{ y } (t) = 1 + C2 e^{-t};$$

$$x' = -x; \quad y' = -y, \quad y(t) = C_3 e^{-t};$$

$$x' = -x - 1; \quad y' = -y - 1, \quad y(t) = -1 + C_4 e^{-t};$$

$$x' = -x - 2; \quad y' = -y - 2, \quad y(t) = -2 + C_5 e^{-t};$$

$$x' = -x - 3; \quad y' = -y - 3, \quad y(t) = -3 + C_6 e^{-t};$$

$$x' = -x - 4; \quad y' = -y - 4, \quad y(t) = -4 + C_7 e^{-t}.$$