

**UNIVERSIDAD CENTROCCIDENTAL
"LISANDRO ALVARADO"**

**SISTEMA DIFUSO NEURONAL
APLICADO A LA CLASIFICACIÓN DE ENFERMEDADES
REPRESENTADAS POR PATRONES SOLAPADOS**

JOSE GERARDO CHACON RANGEL

Barquisimeto 2005

INTRODUCCIÓN

El desarrollo de la “lógica difusa” fue motivado en gran medida, por la necesidad de crear un marco conceptual para representación flexible del conocimiento, dentro de un ambiente impreciso e incierto [1]. El término de lógica difusa ha sido utilizado con dos significados diferentes. En un sentido restringido, se refiere a un sistema lógico que generaliza la lógica clásica bivalente, para razonamiento bajo incertidumbre. En un sentido amplio, la lógica difusa se refiere a todas las teorías y tecnologías que emplean conjuntos difusos, los cuales son clases con límites poco definidos. La teoría de conjuntos difusos constituye un método para especificar que también un objeto corresponde a una descripción vaga. La lógica difusa tiene una faceta relacional, que trata principalmente de la representación y manipulación de funciones y relaciones definidas en forma imprecisa. Las redes neuronales artificiales buscan emular el funcionamiento de las redes neuronales biológicas, en lo relativo al aprendizaje y procesamiento de información. Una red neuronal o red neural es un sistema de procesamiento digital complejo, con fundamento matemático no lineal, que posee la capacidad de almacenar conocimiento. La primera versión de la neurona artificial se le atribuye a W.McCulloch y W. Pitts en 1943 pero a partir de 1980 ocurrió un renacimiento en el interés por estudiar las redes Neuronales, posiblemente motivado por una mayor comprensión sobre el procesamiento de información en los seres vivos, los avances en la teoría de la computación y la disponibilidad de nuevas teorías de algoritmos. La lógica, las redes neuronales y sistemas híbridos que integren ambas disciplinas son considerados como campos de estudio dentro de la inteligencia artificial.

En los últimos años ha habido un interés creciente en investigar sobre la aplicación de la lógica difusa en el desarrollo de sistemas neuronales, tal como lo revela el considerable número de artículos sobre el tema en los números de revistas

especializadas como por ejemplo los artículos de: Liquin Ma, Camping Chen, Ziping Tian, Wanqing H [2] , Sushmita Mitra [7], Sankar Pal, [3] y Hayashi A. [34].

Un caso particularmente interesante para desarrollar un sistema difuso neuronal es el de diagnóstico de enfermedades en diferentes disciplinas medicas, en las cuales se cuentan con patrones solapados para el diagnóstico y se pueda inferir sobre la enfermedad que se padece, definiendo con precisión las áreas de solapamiento entre los datos, que impiden la formulación de una conclusión definitiva.

Para la realización de este proyecto se construyeron una serie de modelos con lógica difusa y redes neuronales a partir de un modelo inicial propuesto por Sankar K. Pal y Sushmita Mitra [21]. El modelo Difuso Perceptron Multicapa (FMLP), incorpora rasgos difusos en la capa de salida de una red neuronal tipo perceptron multicapa. En este modelo, el valor de pertenencia de un patrón a una clase se define como (μ) y se utiliza como etiqueta de entrenamiento durante la fase de aprendizaje. Durante el proceso de entrenamiento de la red, los pesos w_i son actualizados por retropropagación del error con respecto a estos valores de (μ) .

DEDICATORIA

A Dios Todopoderoso, por permitirme vivir y realizar su voluntad.

A mis Padres.

A mis amigos y compañeros de estudios, por su amistad incondicional, estímulo y aliento para finalizar esta meta.

José Gerardo

AGRADECIMIENTO

Al Dr. Carlos Ramírez, por su colaboración.

Al Laboratorio de Bioingeniería de la UNET, por su apoyo logístico.

En especial al Ing. MSc. Carlos Lameda, por su invaluable contribución.

A mis colegas que contribuyeron con su conocimiento y estímulo.

José Gerardo

INDICE

DEDICATORIA.	iv
AGRADECIMIENTO.	v
INDICE DE FIGURAS.	ix
INDICE DE TABLAS.	xi
RESUMEN.	xii
INTRODUCCIÓN.	1
CAPÍTULO	
I EL PROBLEMA	3
Planteamiento del problema.	3
Objetivos.	5
General.	5
Específicos.	5
Justificación e Importancia.	6
Alcance y Limitaciones.	7
II MARCO TEÓRICO	9
Antecedentes.	9
Bases teóricas.	13
La Neurona Biológica.	13
La Neurona Artificial.	15
El Perceptrón.	15
Arquitectura de redes Neuronales.	16

	Tipos de conexiones.	17
	Esquema de interconexión.	17
	Configuración de capas.	17
	Métodos de aprendizaje.	18
	Redes Perceptrón Multicapa.	20
	Algoritmo de entrenamiento Redes Perceptrón Multicapa.	22
	Lógica Difusa.	28
	Sistemas Neurodifusos.	33
	Generalidades de los Sistemas Neurodifusos.	34
	Limitaciones de los Sistemas Neurodifusos.	35
	Clasificación de los Sistemas Neurodifusos.	35
	Datos sintéticos con solapamiento predefinido.	47
	Datos reales con solapamiento aleatorio.	52
	Enfermedades representadas en esta investigación.	52
III	MARCO METODOLOGICO	56
	Tipo de investigación.	56
	Fases de estudio.	56
	I Diagnostico.	57
	II Factibilidad.	60

	III Diseño de la propuesta.	62
	IV Experimentación.	62
IV	PROPUESTA	63
	Datos sintéticos circulares.	63
	Datos sintéticos cuadrados.	66
	Etiquetamiento difuso de los patrones de Entrenamiento	69
	Algoritmo de entrenamiento de la red.	71
	Prueba y validación de la red.	74
V	RESULTADOS	80
	Resultados para los datos sintéticos.	75
	Resultados para los datos hepáticos.	80
	Generación de reglas datos hepáticos.	85
	Comparación de los resultados con trabajos anteriores con los datos hepáticos.	87
VI	CONCLUSIONES Y RECOMENDACIONES.	88
	Conclusiones.	88
	Recomendaciones.	91
	Referencias Bibliograficas.	93
	Anexos.	98

INDICE DE FIGURAS

La Neurona Biológica.	14
La Neurona Artificial.	15
Elementos de un Perceptron.	16
Taxonomía de la redes neuronales.	18
Algoritmos de entrenamiento.	19
Red Multicapa.	21
Función característica ordinaria.	30
Funciones de membresía.	32
Sistema ANFIS.	36
Sistema FMLP.	37
Función S.	38
Función π.	40
Solapamiento para un atributo.	41
Diagrama de bloque modelo FMLP.	43
Diagrama de bloque de inferencia.	44
Diagrama para selección de caminos.	47
Parámetros para datos circulares.	50
Parámetros para datos cuadrados.	52
Programa Solapamiento circular.	63
Programa genera patrones circulares.	64
Programa de almacenamiento.	65

Gráficos de solapamiento circular.	65
Programa Solapamiento cuadrado.	66
Gráficos de solapamiento cuadrado.	68
Programa gráfico para Difusificar.	71
Programa gráfico para entrenar la red.	72
Programa gráfico para resultados.	73
Programa gráfico para probar la Red.	74
Error Cuadrático Medio Vs Iteraciones Entrenamiento.	82
Simulación mejor red con datos de entrenamiento.	83
Simulación mejor red con datos de validación.	84
Error Cuadrático Medio Vs Iteraciones Validación.	85
Reglas para los datos hepáticos.	86

INDICE DE TABLAS

Características de Redes Neuronales y Lógica Difusa.	32
Estimación de gastos.	61
Resultados datos sintéticos circular 40 % solapamiento.	75
Resultados en datos sintéticos circular 40 % solapamiento 2 capas.	76
Resultados en datos sintéticos circular 40 % solapamiento 3 capas.	77
Mejores resultados obtenidos en datos sintéticos.	77
Reglas para mejores resultados datos sintéticos.	79
Muestra de los datos hepáticos.	80
Resultados de la red una capa 1000 iteraciones datos hepáticos.	81
Resultados de la red una capa 2000 iteraciones datos hepáticos.	81
Resultados de la red una capa 3000 iteraciones datos hepáticos	81
Resultados de la red 5 capas 4000 iteraciones 25 Neuronas por capa datos hepáticos.	83
Resultados de la red con mejor desempeño.	84
Justificación de las reglas datos hepáticos.	85
Comparación con otros trabajos.	87

CAPÍTULO I

EL PROBLEMA

Planteamiento del Problema

Los sistemas neuronales y la lógica difusa tienen los requisitos contrastantes para su utilización. Los sistemas difusos son apropiados si el conocimiento del especialista es suficiente sobre el proceso (conocimiento del experto) y está disponible, mientras en los sistemas neuronales es útil si los datos del proceso son suficientes (cuantos datos históricos se tienen y son suficientes) y están disponibles.

Ambos acercamientos no lineales de estos sistemas se basan en variables continuas limitadas, la diferencia es que en los sistemas neuronales el trato es de una manera numérica cuantitativa, considerando que los sistemas difusos se tratan en forma simbólica cualitativa. Los sistemas difusos, sin embargo presentan ambos rasgos simbólicos y numéricos. Por ejemplo, cuando se trata con colecciones de patrones encapsulados por etiquetas lingüísticas, ellos se prestan al proceso de funcionamiento simbólico basado en reglas, mientras refiriéndose a las definiciones de etiquetas lingüística sus funciones también son convenientes para el proceso numérico. Por consiguiente, se considera útil y conveniente la integración de las superioridades de los sistemas neuronales y difusos en una relación simbiótica en que los sistemas difusos mantienen un almacén poderoso para la representación del conocimiento del especialista, mientras que los sistemas neuronales mantienen capacidades de aprendizaje y la conveniencia excepcional en las aplicaciones de hardware computacional eficaz.

Un caso que es especialmente interesante para modelar un sistema difuso neuronal (Neurodifuso) es el de un clasificador de patrones, los cuales representan las

condiciones de sintomatología de un conjunto de pacientes (un patrón con múltiples atributos por paciente) a los cuales se le diagnosticó un tipo de enfermedad por un especialista. Pero que pasa cuando a pacientes con patrones de sintomatología iguales se les diagnostican enfermedades diferentes, como sería el comportamiento o la respuesta del sistema clasificador propuesto ante estos casos de solapamiento de patrones.

Esto motiva a plantear como trabajo de grado una investigación sobre posibles formas de modelar este tipo de sistemas, para patrones solapados que representen estudios preclínicos de pacientes cuyo diagnóstico de enfermedad ha sido diferente, combinando la lógica difusa y las redes neuronales para modelarlos.

Podemos concebir el problema como la creación de un sensor virtual capaz de predecir las salidas sobre la base de patrones con solapamiento de clases, aplicado al campo de la informática médica. El problema de investigación se enfoca hacia la construcción de un sistema híbrido basado en lógica difusa y redes neuronales para la clasificación de patrones con múltiples atributos y solapamiento de clases.

Objetivo del Estudio

Objetivo General

Desarrollar un sistema computacional difuso neuronal que permita clasificar enfermedades a través de patrones con comportamiento de salida solapada.

Objetivos Específicos

1. Diseñar y simular conjuntos de patrones de entrenamiento bidimensionales para una red multicapa con solapamiento progresivo.
2. Desarrollar un algoritmo para el etiquetamiento difuso de los patrones de entrenamiento de una red multicapa.
3. Implementar el algoritmo propuesto en un ambiente de programación visual (Utilizando la interfaz gráfica de Matlab).
4. Realizar pruebas al algoritmo propuesto y comparar los resultados con trabajos anteriores.
5. Implementar y validar el algoritmo propuesto para una base de datos real en el área de la informática medica.
6. Analizar resultados de los experimentos realizados.
7. Concluir sobre los resultados obtenidos.

Justificación e importancia

El presente trabajo de investigación estudia la posibilidad de utilizar un clasificador difuso acoplado a una red neuronal multicapa que indique la presencia de solapamiento para un conjunto de patrones de datos iguales con respuestas o salidas diferentes. La posibilidad de implementar este sistema en el campo de la informática medica en bioingeniería y que de alguna manera favorezca a médicos especialistas a responder a patrones con estos comportamientos es base suficiente para la justificación de esta investigación.

Este trabajo de investigación toma como datos iniciales un conjunto de patrones pertenecientes a un conjunto de datos sintéticos, estos son diseñados de forma que el solapamiento entre los patrones este definido en términos de porcentajes anticipadamente. El solapamiento se incrementa progresivamente en los patrones en forma controlada de manera que permita analizar y estudiar la influencia del etiquetado difuso en la red neuronal y medir el desempeño de la misma.

Este clasificador difuso se implementa para el diagnostico de enfermedades donde ocurre solapamiento con los patrones reales de sintomatología de una enfermedad específica en tiempo real. El trabajo produce un modelo eficiente de un sistema híbrido difuso neuronal para patrones con comportamiento solapado, con miras a establecer mejoras en cuanto a la rapidez en la determinación de diagnósticos, automatización en el entrenamiento del sistema, y facilidad de interpretación de las salidas del sistema por el usuario, utilizando técnicas de inteligencia artificial.

El modelo neurodifuso propuesto a través de la investigación permite generar modelos que pueden utilizarse para hacer nuevos análisis con miras a la creación de diseños mejorados de equipos para la clasificación de patrones con solapamiento de clases.

Este tipo de sistemas puede ser posteriormente aplicado en otras áreas de la bioingeniería que presenten casos de patrones con comportamientos similares tales como la biomecánica o ingeniería clínica.

Alcance y Limitaciones

Alcance.

Para el desarrollo del siguiente proyecto tomaremos en consideración el análisis de tópicos relacionados con sistemas integrados de lógica difusa y redes neuronales, estimación de parámetros y validación del modelo aplicado a patrones con solapamiento de salidas hasta un máximo de 2 clases.

Se analizan diversos modelos alternativos de sistemas para la clasificación de patrones con salidas solapadas, a partir de datos generados en forma controlada. La especificación de la estructura involucra la determinación de las neuronas en la capa intermedia de la red neural. La validación implica la prueba del sistema con criterio de desempeño de mínimo error.

Se espera que el sistema desarrollado sea capaz de clasificar y predecir las salidas en un conjunto de patrones con solapamiento de clases con un mínimo de error para el modelo:

- Una, dos y tres capas escondidas para los datos sintéticos.
- Un máximo de 5 capas ocultas para los datos reales.
- Para los datos sintéticos se toman 2 atributos por cada patrón (Patrón bidimensional).
- Para los datos reales se toman 9 atributos por cada patrón.

- Las funciones de activación de las neuronas pueden ser diferentes.
- Una neurona de salida por cada clase.
- La conexión entre las neuronas es total una a una.
- El procesamiento de la aplicación es uno por uno.

Limitaciones

Cuanto más complejo sea el problema a modelar, más grande debe ser la red, por lo tanto, mas ejemplos se necesitan para entrenarla y esto requiere de hardware adecuado y tiempo para la experimentación de modelos con muchos parámetros y datos de entrenamiento muy extensos. Las limitaciones para este proyecto se presentan a continuación:

- Maneja un máximo de 9 atributos por patrón de entrada a la red.
- Maneja un máximo de dos clases como salida para los datos sintéticos y cuatro clases para los datos reales.
- Un valor para cada uno de los factores de difusividad. (f_{no} , f_e , f_d).
- Se espera obtener un porcentaje de error en los experimentos menor al 10 %.
- El solapamiento de los patrones es sobre dos clases para los datos sintéticos.
- Para los datos sintéticos se utiliza hasta un máximo de tres capas ocultas, se experimenta con el número de neuronas por capa oculta hasta un máximo de 15.
- Para los datos reales se experimenta con varias capas ocultas hasta conseguir la de mejor desempeño.

CAPÍTULO II

MARCO TEÓRICO

Antecedentes

En los últimos diez años han surgido diversos trabajos de investigación relacionados con modelado de sistemas basados en redes neuronales, lógica difusa y su aplicación a sistemas físicos, en estos trabajos se han abordado diferentes problemas y enfoques. Como ejemplo podemos nombrar artículos publicados por universidades y centros de investigación como: Ovidiu Grigore, Inge Gavat en 1999. Yoshihiro Kobayashi and Kostas Terzidis en el 2000, Sankar K. Pal and Sushmita Mitra 1999, Somying Promcharoen and Yuttapong Rangsanseri en el 2000. En la Universidad Nacional Experimental Politécnica (UNEXPO) se realizaron proyectos entre ellos los de Gil, Alvaro en 1997, Volcanes, Rafael en el 2001. En la Universidad del Zulia (LUZ), Ocando, Atencio en 1992. En la Universidad Centro Occidental “Lisandro Alvarado” (UCLA), Sánchez, José en el 2000.

En [35], se presentan los resultados obtenidos en una investigación sobre la tarea de reconocimiento de vocales en lenguaje rumano aplicando redes neuronales difusas. Se presentan los resultados obtenidos por un modelo de red perceptron multicapa, un modelo de red perceptron multicapa difuso y un mapeo de Kohonen difuso. En este artículo se utiliza el algoritmo perceptron multicapas difuso (FMLP) propuesto por Sankar K. Pal [30].

El Trabajo [20] trata sobre un sistema de Diseño Asistido por Computadora (CAAD) que utiliza las tecnologías de Inteligencia Artificial (AI) y procesamiento de Imágenes. El objetivo es crear un sistema CAAD que detecta los edificios en las imágenes de satélite y genera modelos computarizados de la ciudad permitiendo que los usuarios del sistema manipulen los modelos utilizando tecnología de "machine learning" o de aprendizaje de máquina. La flexibilidad y aplicabilidad del sistema fue evaluada en casos de estudios. Se utilizaron técnicas de inteligencia artificial como redes neurales y lógica difusa como parte de la metodología del sistema.

En el trabajo [22] se presenta una descripción general del sistema neuronal perceptron multicapas difuso (FMLP). El autor plantea la necesidad de la integración de los sistemas neuronales y la lógica difusa, se describe el reconocimiento difuso basado en redes neuronales y hace un recuento sobre los diferentes modelos de redes neuronales.

En [23] se presenta un modelo de sistema experto basado en la versión del perceptron multicapas difuso (FMLP). En este modelo se asigna una etiqueta difusa como valor de la función de pertenencia del patrón a la clase y también produce una medida segura que expresa confianza en la decisión. El modelo tiene la capacidad del uso de los rasgos de entrada más importantes de la información y si lo requiere una entrada parcial de datos. En este modelo las magnitudes de las conexiones de los pesos en la red de entrenamiento son utilizadas en muchos estados de los procedimientos de inferencia propuestos.

En [26] se presenta un algoritmo basado en lógica difusa y redes neuronales para clasificación supervisada de imágenes satelitales multiespectrales. Para tratar la información difusa, los diferentes tonos de grises representados por cada conjunto de píxeles son convertidos en tres propiedades lingüísticas. Estas proveen rasgos difusos al vector de entrada de una red neuronal en términos de valores de membresía para cada categoría de grises. Los resultados de los experimentos son comparados con un clasificador de redes neuronales multicapas normal

En [16] se realizó una aplicación de redes neuronales en estimación, como instrumento de medición, de flujo multifásico. El autor uso redes neuronales multicapa con algoritmo de retropropagación de error para reconocer el porcentaje de agua, petróleo y gas producido en pozos de petróleo. Las simulaciones realizadas arrojaron una gran capacidad para identificar los patrones de flujo y estimar el flujo multifásico presente en una tubería; aún bajo la existencia de ruido aleatorio, saturación, corrimiento de calibración de los sensores y ocurrencia de valores fijos en el tiempo.

En [18] el principal objetivo de este trabajo de investigación es como obtener diagnósticos de lesiones en imágenes gastroscópicas más precisos, rápidos y confiables. Se empleo la técnica de la pirámide difusa para identificar una lesión gástrica sobre una imagen digitalizada proveniente de un endoscopio. Se usó reconocimiento de patrones por el método probabilista bayesiano y redes neuronales tipo perceptrón. Los resultados mostraron que el reconocimiento de patrones usando

redes neuronales proveen un diagnostico mas preciso en la evaluación de las lesiones gástricas.

En [19] se orienta al área de reconocimiento de patrones usando redes neuronales perceptrón multicapas. El autor reconoció con éxito patrones numéricos, simbólicos y gráficos.

En [36] se propone un modelo neuronal para lograr el reconocimiento en una escena captada a través de una cámara de video. Este sistema forma parte de un controlador para el movimiento de robots en un campo de fútbol. El modelo consiste básicamente en la identificación de objetos en la imagen a través de sus componentes cromáticas, y dar formato a esta información, en tiempo real, para que sea de utilidad a otros módulos del sistema de control. Se uso con éxito redes neuronales en el reconocimiento de los componentes de la escena basándose en un análisis cromático de los mismos.

En los artículos mencionados anteriormente se utilizan técnicas de inteligencia artificial como por ejemplo: Redes neurales multicapa (MLP), algoritmo de entrenamiento de red por retropropagación del error, algoritmo perceptron multicapas difuso (FMLP), en diferentes campos de acción, lo que refleja lo versátil de estas técnicas. En el presente proyecto de investigación se utiliza una combinación de algunas de estas técnicas aplicadas al campo de la informática médica.

Bases Teóricas

La Neurona Biológica

Las redes neurales o redes neuronales constituyen un enfoque metodológico inspirado en la emulación de algunas capacidades funcionales de las neuronas del cerebro humano como el aprendizaje, el procesamiento paralelo y que pueden ser utilizados para establecer relaciones complejas entre variables de proceso. Se estima que el cerebro humano posee billones de neuronas (alrededor de 10^{11}) y un número mayor de interconexiones y en un área de un milímetro cuadrado hay aproximadamente 50.000. El tamaño y forma de las neuronas es variable, pero todas poseen las mismas subdivisiones anatómicas. En la figura 2.1 se muestra una neurona biológica y sus partes. El soma o cuerpo de la célula contiene el núcleo y es el encargado de las actividades metabólicas de otras neuronas a través de sinopsis en su superficie.

Las dentritas son estructuras que parten del soma con ramificaciones, se especializan en la recepción de señales de otras células nerviosas por medio de conexiones sinápticas. El axón permite enviar impulsos a otras células nerviosas. En algunas neuronas los impulsos se inician en la unión del axón y el soma, y luego se transmiten a lo largo del axón a otras células nerviosas. Cuando el axón esta cerca de

sus células destino se divide en muchas ramificaciones que forman sinapsis tonel soma o axones de otras células.

La sinapsis es una conexión entre dos células nerviosas. Las sinapsis pueden ser excitadotas o inhibitoras según el neurotransmisor que se libere, cada neurona recibe de 10.000 a 100.000 sinapsis y su axón realiza una cantidad similar de sinapsis. La sinapsis es química por naturaleza pero tiene efectos eléctricos laterales que se pueden medir.

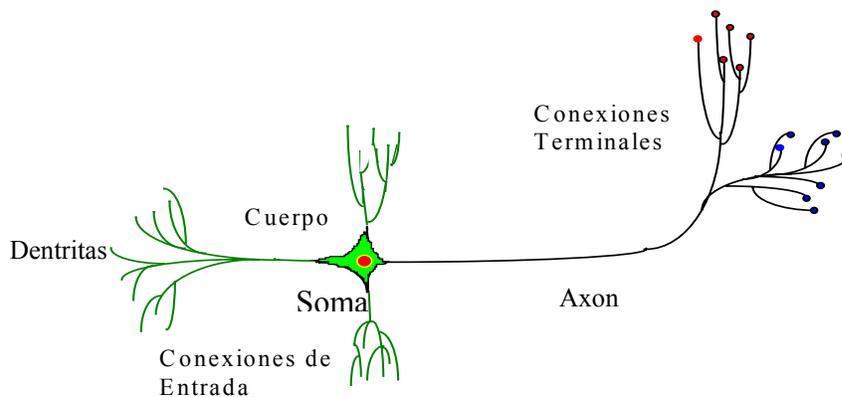


Figura 2.1
La neurona biológica y sus partes

La Neurona Artificial

En el trabajo W.S McCulloch y W. Pitts, publicado en 1943 se propone un modelo simple de la neurona biológica, en la figura 2.2 se puede observar el modelo. Las entradas son X_i , $i = 1, \dots, n$, las sinapsis se modelan por los pesos ω_i . La neurona se

dispara si la suma ponderada de las entradas excede el valor umbral μ , en este caso la salida de la neurona toma dos valores (-1, +1).

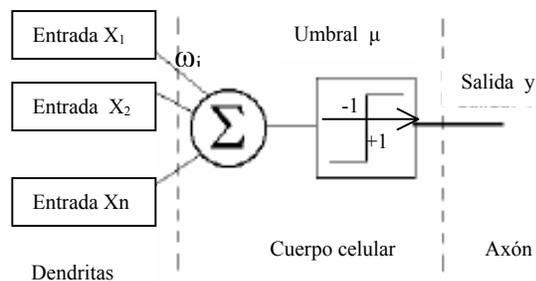


Figura 2.2
Modelo simplificado de la neurona biológica según McCulloch y Pitts

Una rata promedio de disparo de la neurona biológica es de 1000 pulsos por segundo.

El Perceptrón

El elemento básico de una red neuronal artificial es la neurona adaptativa, también conocida como perceptrón. Un Perceptrón esta constituido por los siguientes elementos:

1. Un punto de suma de entradas ponderadas.
2. Una función de activación no lineal.
3. Una regla de entrenamiento o aprendizaje.

En la figura 2.3 se muestran los elementos que conforman un perceptrón. Tal modelo de neurona se puede representar matemáticamente por las siguientes ecuaciones:

$$y_o = \sum_{i=1}^n w_i(t) x_i(t) + w_{n+1}(t) b \quad ; \quad (2.1)$$

$$z_o = \Gamma (y_o(t)) \quad (2.2)$$

Donde $X_1 \dots X_n$ son las entradas, $W_1 \dots W_n$ son los pesos de cada entrada, b es la señal umbral (bias) y Z_o es la función Γ de activación que representa la salida de la red neuronal.

Arquitectura de redes neuronales

El perceptrón presentado en el apartado anterior es de una aplicación muy reducida, por cuanto usualmente se diseñan redes de varias neuronas interconectadas usando diferentes topologías para representar un sistema de mayor complejidad.

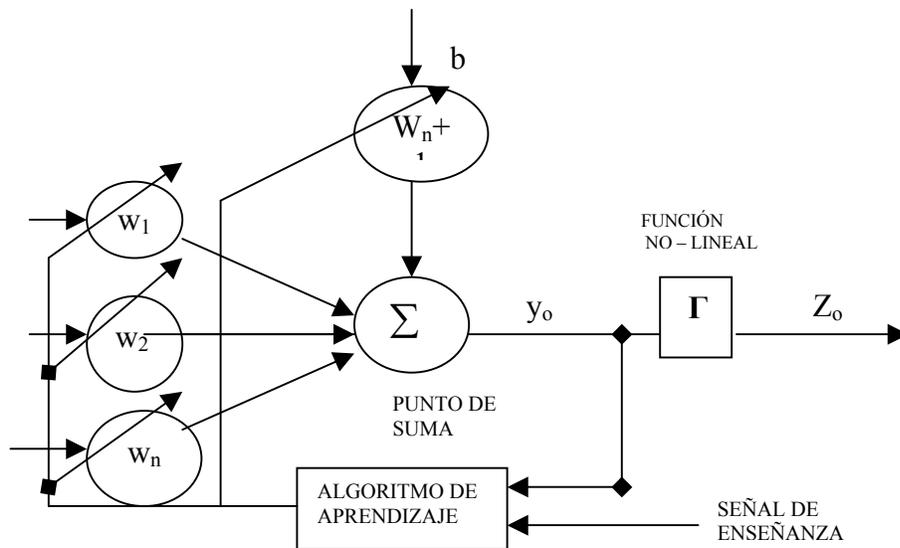


Figura 2.3
Elementos de un Perceptrón

Tipos de conexiones

Existen dos tipos de conexiones: excitatorias e inhibitorias. Las primeras, incrementan el valor de activación de una neurona y son normalmente representadas por señales o valores positivos, mientras que las segundas disminuyen el valor de activación en una neurona y son representadas por señales negativas.

Esquema de interconexión

Se pueden dividir en Intra – capas, inter – capas y recurrentes. El primer esquema establece conexiones entre neuronas que pertenecen a una misma capa. Una conexión inter – capa, conecta neuronas entre dos capas distintas. Finalmente, las conexiones recurrentes establecen un lazo entre la salida y la entrada de una misma neurona.

Configuración de capas

Las capas son configuradas dependiendo de la aplicación específica, existe una infinidad de configuraciones posibles, sin embargo, las más comunes son: Una capa con retroalimentación, dos capas con alimentación adelantada y tres capas con alimentación adelantada.

La taxonomía presentada por Lippman [27] muestra seis tipologías clasificadas para patrones fijos existentes, donde la clasificación depende de los valores de entrada (binarios ó continuos) y el tipo de entrenamiento usado (supervisado ó no

supervisado). Tal clasificación no diferencia entre reglas de entrenamiento adaptativas y fijas. La figura 2.4 muestra la clasificación según Lippman [27].



Figura 2.4
Taxonomía de redes neuronales según Lippman [27]

Métodos de Aprendizaje

Antes que una red neuronal pueda ser usada en su fase de aplicación esta debe ser entrenada. Es decir, debe pasar por un proceso de aprendizaje en donde se ajustan las ponderaciones de las interconexiones; los pesos dados a cada interconexión son intrínsecamente, la información aprendida por la red para proporcionar la salida deseada ante la presentación de un vector de entrada específico.

Los métodos de aprendizaje se pueden agrupar en dos categorías: Métodos de aprendizaje supervisados y métodos no supervisados. Los algoritmos de aprendizaje han sido recopilados por Simpson [28] y la clasificación se presenta en la figura 2.5.

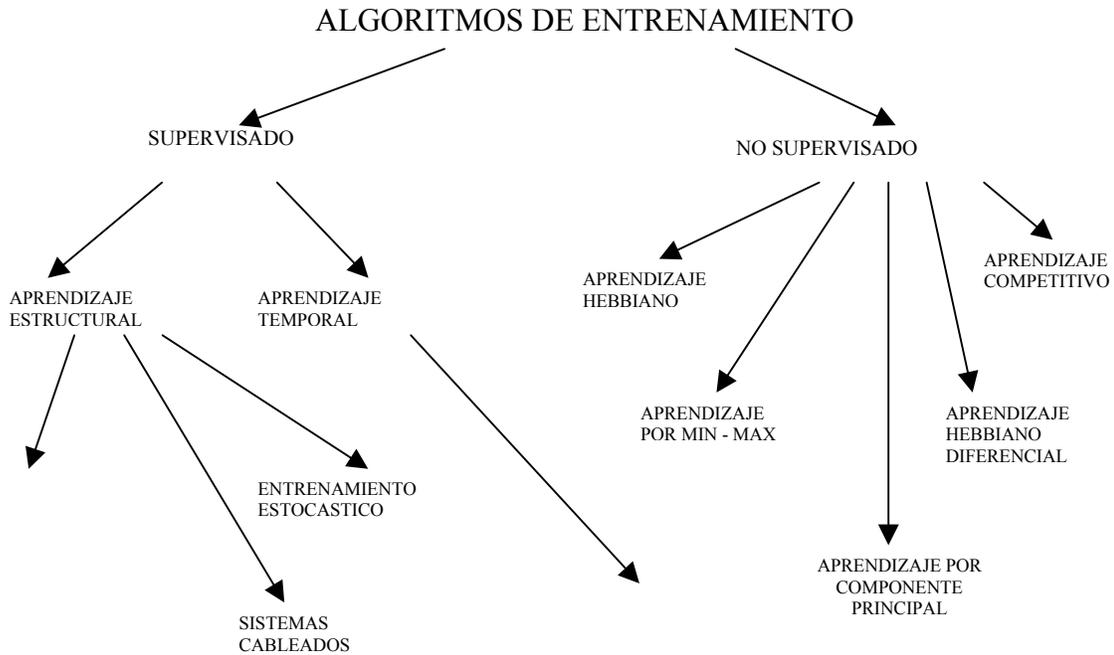


Figura 2.5

Clasificación de los algoritmos de entrenamiento según Simpson [28]

Métodos de aprendizaje supervisados

Se usa un “maestro” para guiar a la red en cada etapa de aprendizaje, indicándole el resultado correcto. Se puede subdividir en aprendizaje supervisado estructural y aprendizaje supervisado temporal. El aprendizaje supervisado estructural se utiliza para encontrar la mejor relación de entrada – salida correspondiente a cada patrón de

entrada. El aprendizaje supervisado temporal, esta relacionado con la captura de una secuencia de patrones necesarios para alcanzar un fin, como en problemas de predicción y control. Dentro de esta categoría de aprendizaje se encuentran los algoritmos de entrenamiento basados en corrección de error, entrenamiento por reforzamiento, entrenamiento estocástico y sistemas cableados.

Métodos de aprendizaje no supervisados

También llamado aprendizaje auto – organizado, no requiere de un maestro externo o crítico para guiar el proceso de aprendizaje. No existen ejemplos específicos de la función que debe aprender la red. Consiste en la auto – organización basada en información local. El entrenamiento hebbiano, el entrenamiento por componente principal y el entrenamiento competitivo son algunos de los métodos de aprendizaje no supervisado.

Redes Perceptron Multicapas (RPM)

Son redes de alimentación adelantada que pueden aceptar entradas continuas o binarias. Estas redes son entrenadas con métodos de aprendizaje supervisado que puede aceptar aprendizaje adaptativo. Se usan algoritmos de descenso por gradiente como método de entrenamiento estructural. Se usan algoritmos de corrección de error como métodos de entrenamiento temporal. Poseen una configuración en cascada de unidades de procesamiento interconectadas. Las neuronas conectadas a los vectores de entrada forman la capa de entrada, las neuronas conectadas a la salida forman la capa de salida, mientras que las capas de neuronas que se encuentran entre la capa de

entrada y la capa de salida se denominan capas escondidas. En la figura 2.6 se muestra una arquitectura de una red perceptrónica multicapa con una capa de entrada, una capa oculta y una capa de salida totalmente conectada, donde $x_i =$ entrada i -ésima; $W_{hi} =$ pesos de conexión entre la neurona i -ésima con la h -ésima y $V_{ch} =$ pesos de conexión entre la neurona h -ésima con la c -ésima neurona de salida. O_c representa la salida de la red. x_{n+1} , y_{h+1} representan los umbrales para cada capa.

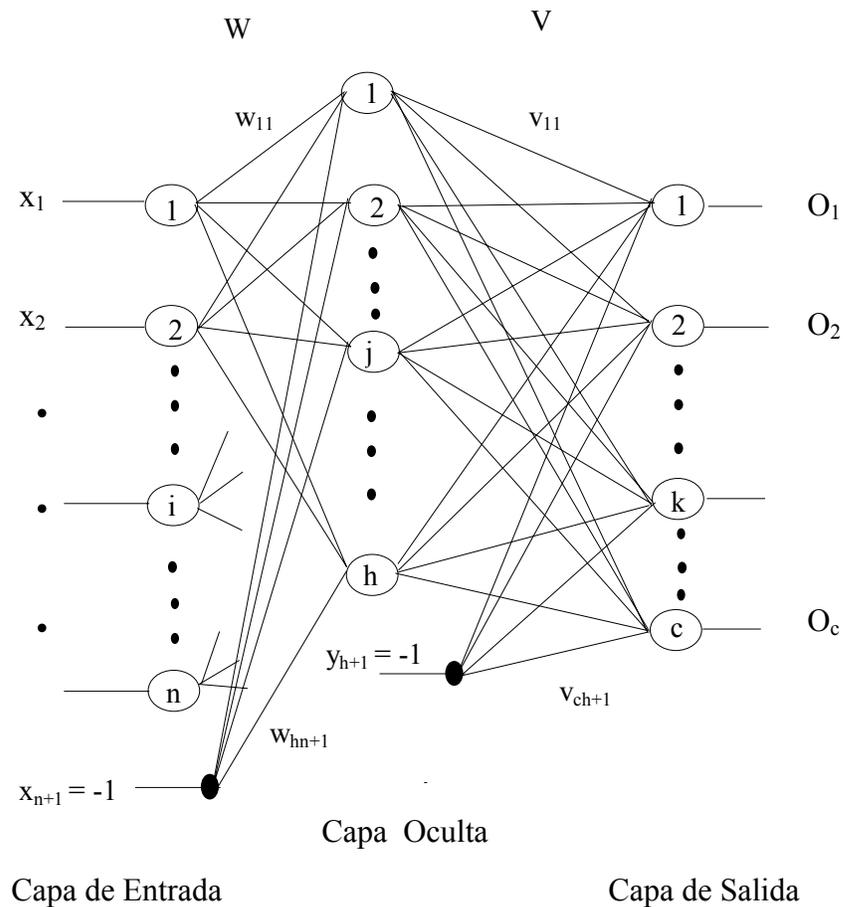


Figura 2.6

Red neuronal perceptrónica multicapa

Siempre que las señales de entrada sean bien seleccionadas, la capacidad de aproximación de una red neuronal perceptrónica multicapa depende del número de capas escondidas, el número de neuronas en cada capa y las funciones de activación lineales o no lineales que se utilicen.

Algoritmos de entrenamiento de una red Perceptron Multicapas

Para entrenar redes perceptrónicas multicapas frecuentemente se utiliza el algoritmo de retropropagación “backpropagation”, el cual es una generalización del algoritmo de mínimos cuadrados. Este algoritmo opera minimizando una función de costo igual a la diferencia cuadrática entre las salidas deseadas y las salidas de la red.

El procedimiento de propagar el error hacia atrás consiste en dos pases a través de las diferentes capas de la red; un barrido hacia delante y un barrido hacia atrás. En la pasada hacia delante un patrón de activación (vector de entrada) es presentado a los nodos sensores de la red, y su efecto se propaga a través de la red capa por capa.

Finalmente, se produce un conjunto de salida como respuesta a la entrada presentada a la red. Durante la pasada hacia delante, pase directo, todos los pesos de las conexiones de la red se mantienen constantes. Durante el barrido hacia atrás, también llamado paso inverso, se ajustan todos los pesos de las conexiones en concordancia con la regla de la corrección del error. Básicamente, la respuesta actual de la red es restada a la respuesta deseada (objetivo) para producir una señal de error.

Así el error es propagado hacia atrás de la red, desde la capa de salida hacia la capa de entrada ajustando los pesos para producir una salida más aproximada a la deseada Haykin [13].

Las principales características de una red perceptrónica multicapa son:

1. El modelo de cada neurona en la red incluye una función no lineal a la salida. El punto importante a destacar es que la función no lineal debe ser diferenciable en cualquier intervalo. Una función de activación muy usada es la sigmoideal. La presencia de la linealidad a la salida es muy importante porque, de no ser así, la relación de entrada – salida de una red perceptrónica multicapa se podría reducir a la red perceptrónica unicapa.
2. La red contiene una o más capas escondidas de neuronas que no forman parte de la entrada o salida de la red. Estas capas escondidas le dan a la red la capacidad de aprender tareas complejas al extraer progresivamente, características mas significativas de los patrones de entrada.
3. La red muestra un alto grado de conectividad, determinado por las conexiones de las redes. Un cambio en la conectividad de la red requiere un cambio en la cantidad de conexiones y sus pesos.

Descripción del algoritmo de retropropagación

Para la descripción de este algoritmo [33], es suficiente con mostrar el procedimiento a seguir para una capa escondida. De la figura 2.6 podemos definir las siguientes matrices de pesos W y V como :

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n+1} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2,n+1} \\ w_{31} & w_{32} & w_{33} & \dots & w_{3,n+1} \\ \cdot & & & & \\ \cdot & & & & \\ w_{h1} & w_{h2} & w_{h3} & \dots & w_{h,n+1} \end{bmatrix} \quad V = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{1n+1} \\ v_{21} & v_{22} & v_{23} & \dots & v_{2,n+1} \\ v_{31} & v_{32} & v_{33} & \dots & v_{3,n+1} \\ \cdot & & & & \\ \cdot & & & & \\ v_{h1} & v_{h2} & v_{h3} & \dots & v_{c,n+1} \end{bmatrix} \quad (2.3)$$

También se define a $X = (x_1, x_2, \dots)^T$ como el vector que representa los patrones de entrada y $D = (d_1, d_2, \dots, d_c)^T$ Es el vector que contiene la respuesta deseada, n es la dimensionalidad, h es el numero de neuronas en la capa escondida y c representa el numero de neuronas en la capa de salida. Podemos entonces definir a Y como el vector que contiene los valores de activación de las neuronas de la capa escondida, esto es $Y = (y_1, y_2, \dots, y_h)^T$ donde $y_j = f(Net_j)$ para $j = 1, 2, \dots, h$, lo cual corresponde a :

$$f(Net) = \frac{1}{1 + \exp^{-\lambda Net_j}} \quad (2.4)$$

Donde

$$Net = \sum_{i=1}^{n+1} w_{ij} x_i \quad (2.5)$$

Los valores de activación para la capa de salida son calculados en forma análoga, esto es $O = (o_1, o_2, \dots, o_c)^T$ donde $o_k = f(Net_j)$ para $k = 1, 2, \dots, h$, lo cual corresponde a :

$$f(Net) = \frac{1}{1 + \exp^{-\lambda Net_j}} \quad (2.6)$$

Donde

$$Net = \sum_{i=1}^{h+1} v_{kj} y_i \quad (2.7)$$

Dados los valores de activación de la capa de salida, la función de error para un determinado patrón de entrada p , para las neuronas de la capa de salida se define como:

$$E_p = \frac{1}{2} \sum_{k=1}^c (d_{pk} - o_{pk})^2 \quad (2.8)$$

Con $p = 1, 2, \dots, S$; donde S es el número de patrones de entrada disponibles para el proceso de entrenamiento de la red. El algoritmo de retropropagación consiste en ajustar los pesos de las conexiones entre las capas escondidas de tal manera que la función de error es minimizada. Esto es, se necesita un Δw_{ji} y un Δv_{ki} tal que las ecuaciones:

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad \text{para } j = 1, \dots, h \text{ e } i = 1, \dots, n+1 \quad (2.9)$$

$$v_{kj} = v_{kj} + \Delta v_{kj} \quad \text{para } k = 1, \dots, c \text{ y } j = 1, \dots, h + 1 \quad (2.10)$$

produzcan una reducción en el valor de Ep . Para lograr esto, los ajustes individuales de las matrices de pesos V y W son realizados siguiendo la dirección negativa del gradiente [33]. El algoritmo se resume en los siguientes pasos:

- Inicializar las matrices de pesos W y V a valores aleatorios pequeños.
- Mientras que $Ep < \varepsilon$ y el # de iteraciones se mantenga por debajo del predeterminado hacer.
 - Mientras existan patrones de entrenamiento hacer.
 - Introducir un nuevo patron X y calcular la respuesta Y y O de la red.
 - Calcular el error Ep .
 - Calcular los ajustes de los pesos W_{ji} y V_{kj} .
 - Ajustar pesos
 - Fin mientras
- Fin mientras.

El error mínimo que la red debe alcanzar con la finalidad de detener el algoritmo se define como ε .

Parámetros que afectan el algoritmo de entrenamiento

El algoritmo de retropropagación, al igual que otros algoritmos que buscan minimizar una función de error a través de métodos de aproximación, trae consigo

una serie de inconvenientes que afectan el desempeño y el éxito del algoritmo en ciertas situaciones [33]. Una serie de parámetros y reglas logradas por investigadores a través de años de práctica hacen del proceso de entrenamiento de una red neural multicapa un arte que muchos dominan y pocos entienden. Factores como la tasa de aprendizaje, el momento, número de iteraciones, configuración de la red, métodos de evaluación de desempeño y tratamiento de los datos pueden marcar una gran diferencia en el proceso de entrenamiento de dos redes exactamente iguales al inicio del aprendizaje.

La tasa de aprendizaje

El algoritmo de retropropagación proporciona una aproximación de la trayectoria a seguir en el espacio definido por los pesos mediante el método del descenso más pronunciado, la ecuación 2.9 se ve alterada por la tasa de aprendizaje α como se muestra en la ecuación 2.11.

$$w_{ij} = w_{ij} + \alpha \Delta w_{ij} \quad \text{para } j = 1, \dots, h \text{ e } i = 1, \dots, n + 1 \quad (2.11)$$

Mientras menor sea el valor de la tasa de aprendizaje α menor serán los cambios a los pesos en la red entre una iteración y otra. Adicionalmente, la trayectoria a seguir será mucho más estable. Esta mejora es sin embargo, obtenida al precio de una convergencia mucho más lenta. Si por el contrario, la tasa de aprendizaje toma un

valor alto, se puede acelerar el proceso de convergencia. Sin embargo, los cambios pronunciados en los pesos pueden volver inestable el proceso de aprendizaje.

El método del momento

Un método simple para incrementar la velocidad de convergencia sin incurrir en el peligro de inestabilidades consiste en agregar un término de momento en la ecuación general que actualiza los pesos de la red esto es:

$$w_{ji}(t+1) = w_{ji}(t) + \alpha \Delta w(t+1) + \eta \Delta w(t) \quad (2.12)$$

$$v_{kj}(t+1) = v_{kj}(t) + \alpha \Delta v(t+1) + \eta \Delta v(t) \quad (2.13)$$

Donde η es normalmente un número positivo llamado constante del momento. Este valor controla el bucle de retroalimentación que actúa sobre $\Delta w(t)$. La incorporación del momento en el algoritmo de retropropagación representa una modificación al proceso de actualización de pesos y tiene efectos beneficiosos en el comportamiento del algoritmo. El término de momento puede también prevenir que el proceso de aprendizaje termine en un mínimo local de la superficie de error [13]. Para asegurar la convergencia del algoritmo, la constante debe ser restringida al intervalo [0,1].

Lógica difusa

El desarrollo de la lógica difusa fue motivado en gran medida, por la necesidad de crear un marco conceptual para representación flexible del conocimiento, dentro de un ambiente impreciso e incierto. El concepto central de la lógica difusa es el

“conjunto difuso”, el cual es considerado como una extensión de los conjuntos ordinarios, y fue presentado en el trabajo desarrollado por L. Zadeh en 1965 . En la teoría sobre conjuntos difusos, la pertenencia de un elemento determinado a un conjunto dado obedece a una lógica multivaluada dentro del intervalo (0,1), que asigna al elemento un “grado de pertenencia” que puede ir desde la no pertenencia (0) hasta la pertenencia total (1). Así pues, dentro de esta idea, la “altura” de una persona es considerada una “variable difusa”, que puede estar clasificada por tres “ conjuntos difusos” etiquetados como “bajo”, “medio”, ”alto”, donde la transición de la pertenencia de un conjunto a otro, de una persona con una altura X , es gradual.

De esta manera, en lógica difusa todo es cuestión de grado donde el razonamiento exacto es pensado como un caso límite del razonamiento aproximado. El conocimiento dentro de esta lógica, es interpretado como una colección de proposiciones difusas, expresadas en lenguaje natural, sobre un conjunto de variables, y sometidas a un proceso de inferencia para determinar su veracidad.

Función característica

Un conjunto ordinario A esta definido sobre otro conjunto ordinario X . La función característica asociada a A es un mapa:

$$\mu_A : X \rightarrow \{0,1\} \tag{2.14}$$

que asigna a cada elemento $x \in X$ el valor de 1 si x es un elemento de A , o 0 si x no es un elemento de A . Esta función característica se ilustra en la figura 2.7

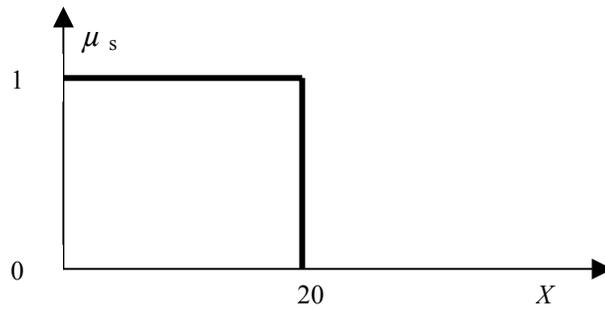


Figura 2.7
Función característica de un conjunto ordinario

Un conjunto difuso también es definido sobre un dominio ordinario llamado el universo de discurso, pero extiende el rango de la función característica sobre todo el intervalo unitario.

Función de membresía o Función de pertenencia.

Sea $X = \{x_1, x_2, \dots, x_n\}$ un universo de discurso. Luego, D es un conjunto difuso sobre X , con una función característica:

$$\mu_s : X \rightarrow [0,1] \quad (2.15)$$

Que asocia a cada $x_i \in X$ un valor sobre el intervalo unitario $I = [0,1]$ usualmente, en el marco de la teoría de conjuntos difusos, la función característica de un conjunto

difuso es llamada *función de membresía*, e indica el *grado* con el que un elemento $x_i \in X$ pertenece a dicho conjunto difuso, comúnmente llamado grado de membresía. Se debe notar que valores del grado de membresía cercanos a 1 indican una pertenencia fuerte al conjunto difuso, mientras que valores cercanos a 0 denotan una pertenencia débil a dicho conjunto. Es importante resaltar que el valor del grado de membresía depende del contexto bajo el cual se esta considerando la definición de conjuntos difusos. En la figura 2.8 se muestran la forma y las fórmulas de las funciones de membresía más utilizadas.

Los sistemas difusos y las redes neuronales tienen cualidades diferentes estas son:

- Las redes neuronales no pueden explicar la forma en que toman sus decisiones.
- Las redes neuronales constituyen una herramienta poderosa que incorpora características novedosas tales como aprendizaje, sin embargo su entrenamiento y selección de la configuración se ha convertido en una especie de arte que se logra sobre la base de la experiencia.
- Las redes neuronales poseen una amplia gama de aplicaciones y las características que exhiben las han convertido en una técnica popular.
- El entrenamiento de los sistemas difusos resulta difícil.
- Los sistemas difusos necesitan de un experto en el área donde se aplican.

- Los sistemas difusos son aplicables prácticamente en cualquier área.

Las características principales de las redes neuronales y la lógica difusa se resumen en la tabla 2.1

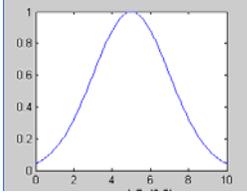
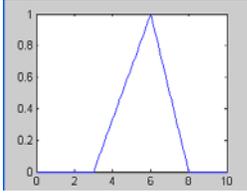
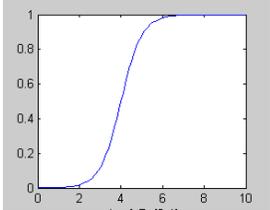
FORMA DE LA FUNCION	FORMULAS
	$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}$
	$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$
	$f(x, a, c) = \frac{1}{1 + e^{-a(x-c)}}$

Figura 2.8
Funciones de membresía más utilizadas.

CARACTERISTICA	REDES NEURONALES ARTIFICIALES	LOGICA DIFUSA
Paralelismo	Bueno	Bueno
Dominio	Continuo	Continuo
Linealidad	No lineal	No lineal
Base de conocimiento	Distribuida en pesos	Distribuida en base a reglas
Metodo de solución	Matemática mediante propagación de la información de entrada	Razonamiento sobre la base de reglas
Representación del conocimiento	Númerica	Lingüística
Conocimiento explotado	Datos empíricos	Conocimiento humano previo
Capacidad de aprendizaje	Alta	No tiene

Tabla 2.1
Características de las Redes Neuronales y la Lógica Difusa.

Sistemas Neurodifusos

El avance de los sistemas inteligentes y la búsqueda de soluciones cada vez más complejas para los problemas que pueden ser resueltos por estos métodos, han llevado a la unión de técnicas emergentes haciendo que las debilidades de unas se compensen con las bondades de otras. Este es el caso de los sistemas Neurodifusos los cuales surgen como una relación sinérgica de las técnicas de Redes Neuronales y Lógica Difusa. Los problemas a resolver con técnicas inteligentes generalmente poseen una componente de datos experimentales y otra de conocimientos previos, por lo que los Sistemas Neurodifusos son una gran alternativa para la búsqueda de soluciones.

Taxonomía de los sistemas Neurodifusos

La clasificación de los Sistemas Neurodifusos fue hecha por Nauck [29] de la siguiente manera:

Modelos Concurrentes

Cuando una red neuronal artificial y el sistema difuso trabajan juntos pero sin interactuar el uno con el otro, es decir, ninguno determina las características del otro.

Modelos Cooperativos

Cuando la Red Neuronal se usa para determinar los parámetros de funcionamiento del sistema difuso. Como ejemplo se puede mencionar la determinación de parámetros de control de un sistema difuso.

Modelos Híbridos

Cuando los Sistemas Difusos y las Redes Neuronales Artificiales trabajan juntas en una arquitectura homogénea que puede ser interpretada como una red neuronal con parámetros difusos o como un sistema difuso con parámetros o funcionamiento distribuidos. Como ejemplo, el Sistema de Inferencia Neurodifusa ANFIS [24].

Generalidades sobre los sistemas Neurodifusos

Los sistemas Neurodifusos permiten el uso de métodos cualitativos y cuantitativos en la construcción, en la etapa de aprendizaje, en la de funcionamiento y en la de

realimentación del conocimiento al usuario. Otras ventajas son la interpretación lingüística, introducción de conocimiento previo autoaprendizaje y generalización.

Limitaciones de los Sistema Neurodifusos

La dimensionalidad

Los Sistemas Neurodifusos presentan el problema del crecimiento exponencial del tamaño del modelo respecto al número de variables que poseen, lo que lo hace impráctico para sistemas de múltiples variables.

Calidad de los datos

Existen problemas con la cantidad, distribución y ruido en los datos, no siempre están disponibles.

Conocimiento previo

En estos sistemas se necesita del conocimiento previo y confiable de un experto en el sistema.

Clasificación de los Sistemas Neurodifusos según su estructura.

De acuerdo a la estructura general, los Sistemas Neurodifusos se pueden clasificar como:

- Modelo basado en reglas difusas usando una red neuronal con técnica de aprendizaje supervisado.

- Modelo basado en reglas difusas usando aprendizaje basado en reforzamiento.
- Modelo basado en reglas difusas usando redes neuronales para construir las particiones difusas del espacio de entrada.

Modelos basados en reglas difusas

Este tipo de modelo se puede estructurar de la siguiente manera:

- Sistema de Inferencia Neurodifuso Adaptativo (ANFIS).
- Razonamiento aproximado basado en redes neuronales Retropropagación.

Sistema de Inferencia Neurodifuso Adaptativo (ANFIS)

Es una red multicapa feedforward en la cual cada nodo ejecuta una función particular (función de nodo) sobre las señales de llegada, usando un conjunto de parámetros específicos a este nodo. Para reflejar diferentes capacidades adaptativas, se usan nodos circulares y cuadrados para representar las neuronas como se muestran en la figura 2.9. Aquí las entradas son x_i y las salidas γ_i .

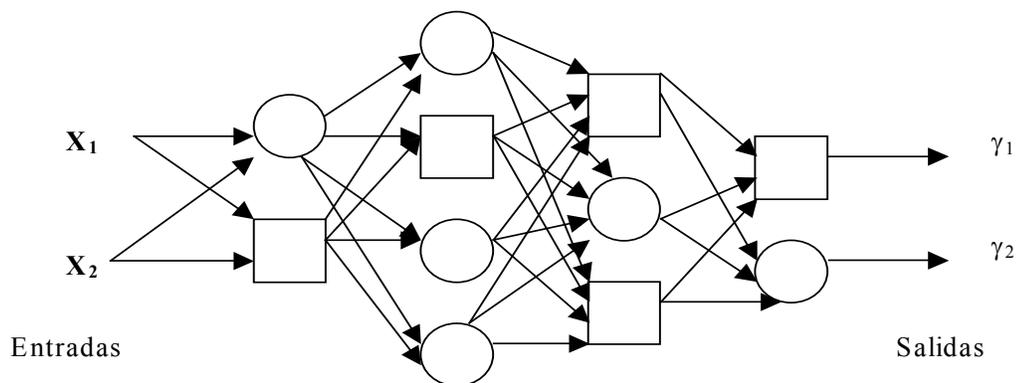


Figura 2.9
Sistema Neurodifuso Adaptativo (ANFIS)

Razonamiento aproximado basado en Redes Neuronales Retropropagación

Este modelo incorpora la fusividad en las capas de entrada y de salida de una red neuronal. Cualquier valor de los atributos de entrada son descritos en términos de alguna combinación de valores de pertenencia en el conjunto de propiedades lingüísticas bajo (L), medio (M) y alto (H). El valor de pertenencia de un patrón a una clase se define como (μ) , este se representa en la capa de salida. Durante el proceso de entrenamiento de la red, los pesos w_i son actualizados por retropropagación del error con respecto a valores de membresía (μ) . Este modelo se muestra en la figura 2.10 y se abrevia como modelo FMLP.

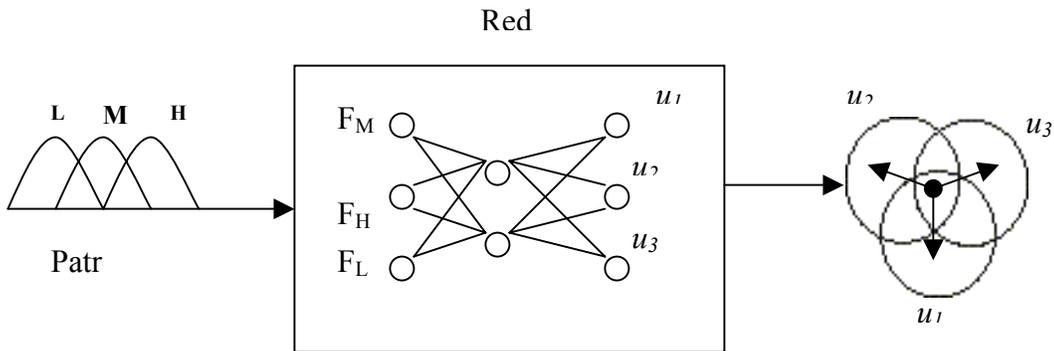


Figura 2.10

Modelo Razonamiento aproximado basado en Redes Neuronales Retropropagación

El modelo propuesto en esta investigación se basa en un sistema de razonamiento aproximado basado en redes neuronales retropropagación (FMLP). El algoritmo para este modelo es desarrollado en Matlab bajo un ambiente gráfico y se prueba su

desempeño para la clasificación de patrones los cuales tendrán un solapamiento previamente definido.

Funciones de membresía o pertenencia para el modelo FMLP

Para este modelo se definen las funciones de pertenencia S representada por la ecuación:

$$\begin{aligned}
 S(r ; \alpha, \beta, c) &= 0 && \text{para } r \leq \alpha \\
 &= 2\left(\frac{r - \alpha}{c - \alpha}\right)^2 && \text{para } \alpha \leq r \leq \beta \\
 &= 1 - 2\left(\frac{r - \alpha}{c - \alpha}\right)^2 && \text{para } \beta \leq r \leq c \\
 &= 1 && \text{para } r \geq c.
 \end{aligned} \tag{2.16}$$

En la figura 2.11 se representa el gráfico de $S(r ; \alpha, \beta, c)$:

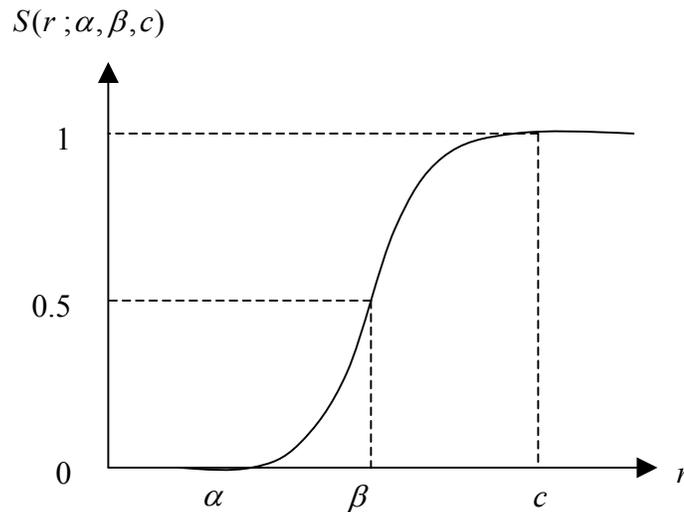


Figura 2.11
Gráfica para la función S

En el modelo FMLP, la función π (en su forma unidimensional) es usada para asignar valores de membresía a las características de entrada. La función π representa un conjunto de puntos agrupados “clustered” alrededor de c y la función π esta representada por la ecuación:

$$\pi(r ; c, \lambda) = \begin{cases} 2 \left(1 - \frac{\|r - c\|}{\lambda}\right)^2 & \text{Para } \lambda/2 \leq \|r - c\| \leq \lambda \\ 1 - 2 \left(\frac{\|r - c\|}{\lambda}\right)^2 & \text{Para } 0 \leq \|r - c\| \leq \lambda/2 \\ 0 & \text{Otra parte} \end{cases} \quad (2.17)$$

donde $\lambda > 0$ es el radio de la función π con c como el punto central y $|\cdot|$ denota la norma euclidiana. Esto se muestra en la figura 2.12 para $r \in \mathbb{R}^2$. Nótese que cuando el patrón r queda en el punto central c de una clase, entonces $\|r - c\| = 0$ y este valor de membresía es máximo, $\pi(c; c, \lambda) = 1$. El valor de membresía de un punto decrece como su distancia del punto central c , $\|r - c\|$ aumenta. Cuando $\|r - c\| = \lambda/2$ el valor de membresía de r es 0.5 y este es llamado un punto de cruce. En $S(r; \alpha, \beta, c)$, $\beta = (\alpha + c) / 2$ y en $\pi(r; c, \lambda)$, λ es el ancho de banda, esto es, la distancia entre el punto de corte de π , mientras c es el punto centro en el cual π es la unidad. En problemas de reconocimiento de patrones nosotros necesitamos a menudo representar una clase con límite difuso en términos de una función π .

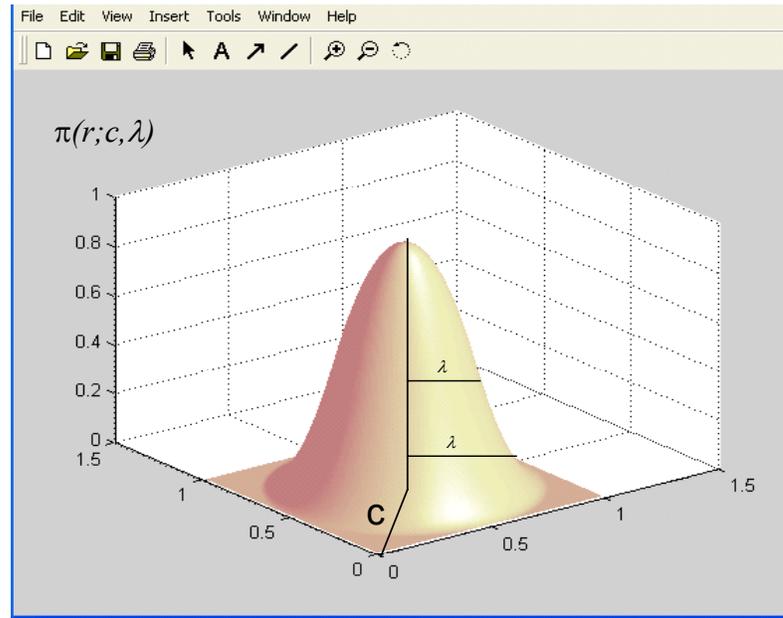


Figura 2.12
Gráfica para la función π

Representación de los patrones

En el modelo difuso FMLP , la función π (en la forma uni-dimensional) es usada para asignar valores de membresía para los atributos de entrada. Cada atributo de entrada F_j (en forma cuantitativa o lingüística) puede ser expresado en términos de valores de membresía de cada una de las tres propiedades lingüísticas *bajo*, *medio* y *alto*. Por lo tanto un patrón n dimensional $F_i = [F_{i1}, F_{i2}, \dots, F_{in}]$ puede ser representado como un vector de dimensión $3n$.

$$F_i = [\mu_{bajo}(F_1) \mu_{medio}(F_1) \mu_{alto}(F_1), \mu_{bajo}(F_2) \dots] \quad (2.18)$$

$$F_i = [x_1^0, \dots, x_{3n}^0] \quad (2.19)$$

donde x_1^0, \dots, x_{3n}^0 se refieren a las entradas neuronales de la capa de entrada del modelo FMLP. En la figura 2.13 se muestra la estructura de solapamiento de las tres funciones π , representando *bajo*, *medio* y *alto* para un atributo particular del patrón de entrada de F_j . Cada atributo de entrada genera tres valores difusos que representan la entrada de la red neuronal, es de hacer notar que esta descomposición aumenta la dimensionalidad del sistema y exige hardware para poder implementar los algoritmos que representan el modelo.

Cuando los atributos de entrada F_j son lingüísticos, estos valores de pertenencia para los conjuntos π bajo, medio y alto se representan por las ecuaciones 2.20, 2.21 y 2.22.

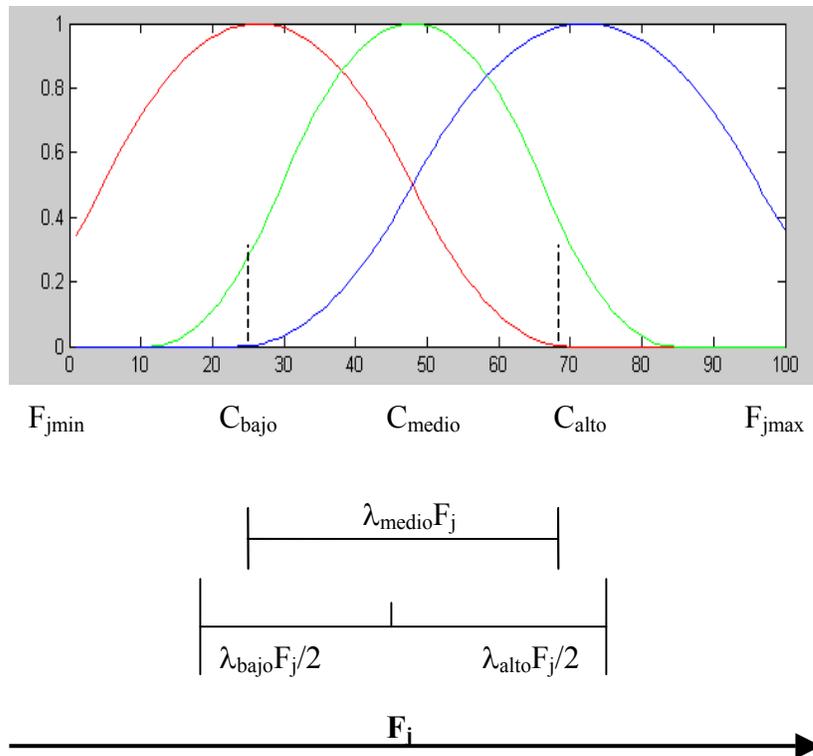


Figura 2.13
Estructura de solapamiento de la función π para un atributo del patrón F_j

$$Bajo \equiv \left\{ \frac{0.95}{L}, \frac{\pi(F_j(\frac{0.95}{L}); c_{medium}, \lambda_{medium})}{M}, \frac{\pi(F_j(\frac{0.95}{L}); c_{alto}, \lambda_{alto})}{H} \right\} \quad (2.20)$$

$$Medio \equiv \left\{ \frac{\pi(F_j(\frac{0.95}{M}); c_{bajo}, \lambda_{bajo})}{L}, \frac{0.95}{M}, \frac{\pi(F_j(\frac{0.95}{M}); c_{alto}, \lambda_{alto})}{H} \right\} \quad (2.21)$$

$$Alto \equiv \left\{ \frac{\pi(F_j(\frac{0.95}{H}); c_{bajo}, \lambda_{bajo})}{L}, \frac{\pi(F_j(\frac{0.95}{H}); c_{medium}, \lambda_{medium})}{M}, \frac{0.95}{H} \right\} \quad (2.22)$$

donde C_{bajo} , λ_{bajo} , C_{medium} , λ_{medium} , C_{alto} , λ_{alto} se refieren al centro y radio de las tres propiedades lingüísticas, y $F_j(0.95/L)$, $F_j(0.95/M)$, $F_j(0.95/H)$ se refiere a los correspondientes valores atributos F_j a que las tres propiedades lingüísticas logran valores de pertenencia de 0.95. (Nótese que el valor de membresía heurísticamente es considerado de 0.95 en lugar de la unidad cuando F_j se identifica como *bajo*, *medio* o *alto*). Dado el concepto difuso, la noción de contención estricta, el valor unitario de membresía, es evitado). Para F_j patrones de entrada numéricos el conjunto difuso π (Ecu. 2.17) se puede usar con apropiados valores de c y λ . Una entrada F_j con propiedades lingüísticas apropiadas divide eficazmente el rango dinámico de cada atributo de entrada en tres particiones solapadas como se muestra en la figura 2.13. Los centros y radios de las funciones π a lo largo de cada uno de los ejes de atributos son determinados automáticamente a partir de la distribución de los patrones de entrenamiento. El procesamiento de la capa de entrada es resumido en el diagrama de bloques de la figura 2.14.

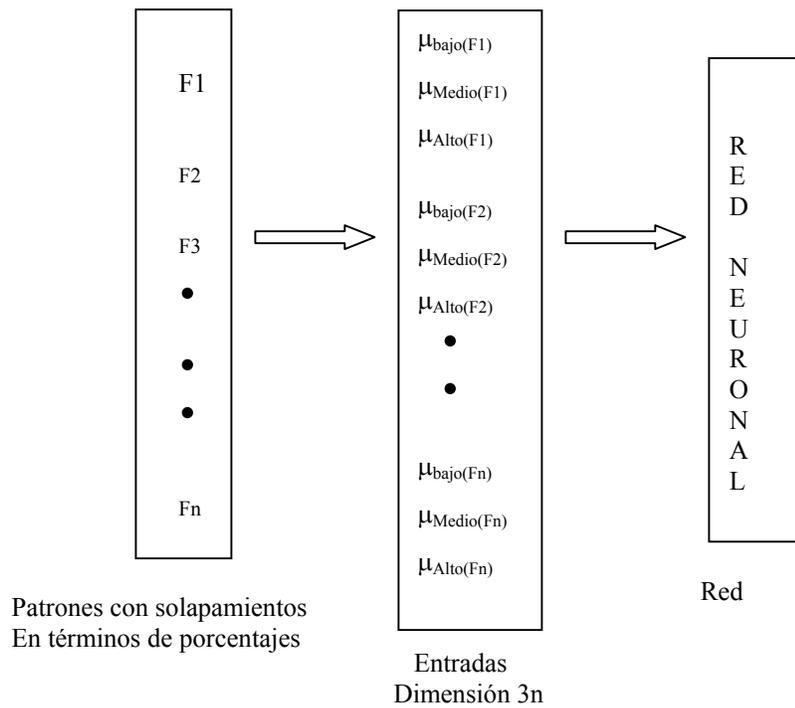


Figura 2.14
Diagrama de bloques del modelo FMLP

Inferencia y generación de las reglas para el sistema Neurodifuso (FMLP)

La generación de reglas y los aspectos asociados a la inferencia de conocimiento en modelos de reconocimiento de patrones neurodifusos, es la etapa final en el proceso del FMLP. El conocimiento es adquirido a partir de los datos del dominio, tomando las ventajas del aprendizaje de las técnicas neuronales. Este conocimiento, a menudo en términos difusos, es representado o codificado a través de los pesos entre las conexiones de la red neuronal ya entrenada. Los modelos son capaces de operar aún con datos no disponibles, y puede cuestionar al usuario acerca de datos adicionales cuando sea necesario. En el campo médico, por ejemplo, los datos pueden no estar

disponibles por varias razones: por ejemplo, algunos exámenes pueden ser riesgosos para el paciente o pueden existir contraindicaciones, se requiere hacer un diagnóstico de manera urgente y algunos resultados muy informativos pero de obtención prolongada pueden ser excluidos del conjunto de datos característicos, o en otros casos, el equipo técnico apropiado no se encuentra disponible. En tales casos, la red puede cuestionar al usuario por información adicional solo cuando es particularmente necesario para inferir una decisión. En resumen el modelo es capaz de:

- Hacer inferencia basado en información completa y/o parcial.
- Preguntarle al usuario por variables de entrada desconocidos que son claves de búsqueda de una decisión.
- Producir la justificación de información en la forma de reglas Si – Entonces.

En la figura 2.15 se muestra una vista general de las diversas etapas envueltas en el proceso de inferencia y generación de reglas .

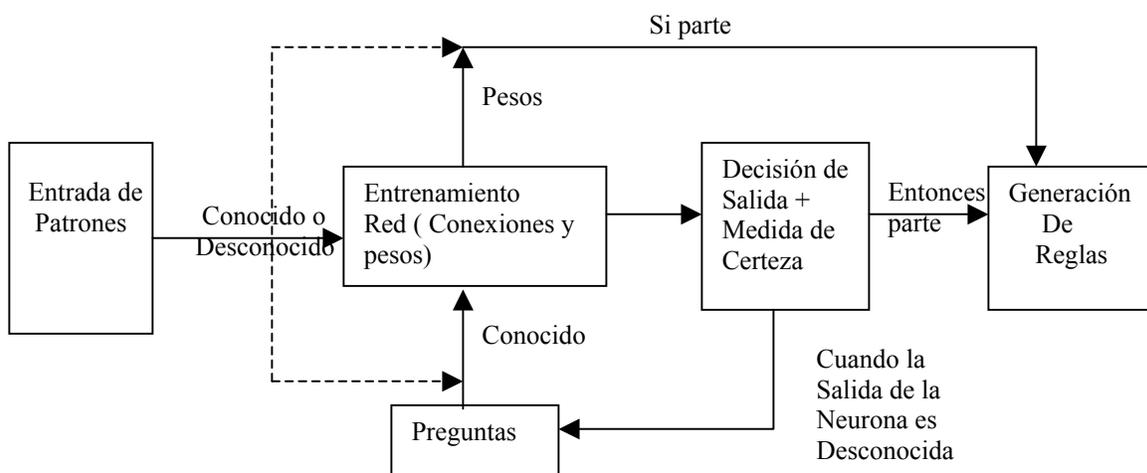


Figura 2.15
Diagrama de bloques de inferencia y de la fase de generación de reglas del sistema FMLP

Representación de los datos de Salida

Los datos de salida del Perceptrón Difuso Multicapa (FMLP) están representados por un conjunto de valores de membresía para categorías predefinidas. Adicionalmente a esta salida, el factor para estimar la credibilidad de los resultados es aplicado. El factor, bel_j^H indica adjetivamente como es el resultado en comparación a las otras escogencias, y esta definido como:

$$bel_j^H = y_j^H - \sum y_i^H \text{ para } i \neq j \quad (2.23)$$

Este valor es usado para escoger formas lingüísticas tales como muy parecido, parecido, más o menos, no parecido e incapaz de reconocer. Las formas escogidas son aplicadas como modificadoras a las cláusulas si-entonces en las reglas difusas generadas. A continuación se muestran las reglas de decisión:

- | | |
|-------------------------|----------------------------------|
| 1. Muy parecido | Para $0.8 \leq bel_j^H \leq 1$. |
| 2. Parecido | Para $0.6 \leq bel_j^H < 0.8$. |
| 3. Mas o menos | Para $0.4 \leq bel_j^H < 0.6$. |
| 4. No Parecido | Para $0.1 \leq bel_j^H < 0.4$. |
| 5. Incapaz de Reconocer | Para $bel_j^H < 0.1$. |
- (2.24)

Justificación de información en la forma de reglas Si - Entonces .

La justificación es el proceso de encontrar las trayectorias de máxima ponderación en la red y generar las reglas difusas basadas en Mitra [7]. A continuación se explica como encontrar las trayectorias de la neurona j en la capa de salida H . Primeramente, seleccionamos una neurona de salida j con $bel_j^H > 0$ Entonces, los pesos $w_{ji}^{H-1} > 0$ conectados a la neurona j son seleccionados en la capa inferior $H-1$. Estas neuronas i conectadas a la neurona j son descritas como m^{H-1} . Los pesos máximos wet_i^{H-1} en la capa $H-1$ es $wet_i^{H-1} > 0$. El algoritmo encuentra la trayectoria de mayores pesos desde la salida, en dirección de arriba hacia abajo. Para las capas ocultas h , el peso máximo wet_i^H es calculado por la suma de los pesos máximos predefinidos en las capas superiores y los pesos en las capas inferiores, y esta definido como:

$$y_j^h > 0.5$$
$$wet_{ih} = \max_{a_k^{h+1}} \{ wet_{ak^{h+1}} + wet_{ak^h} \}$$
(2.25)

donde a_k^{h+1} indica el índice del número de neuronas en la capa superior $h + 1$, wet_i^h es el peso predefinido en las capas inferiores a h . Entonces, las reglas difusas generadas entre una cantidad de m^0 neuronas hasta que:

$$\sum wet_{is}^0 > \sum wet_{in}^0$$
(2.26)

donde i_s indica las neuronas de entrada seleccionadas y in indica el resto de las neuronas del conjunto m^0 . Consideremos como ejemplo una red simple de tres capas como se muestra en la figura 2.16 .

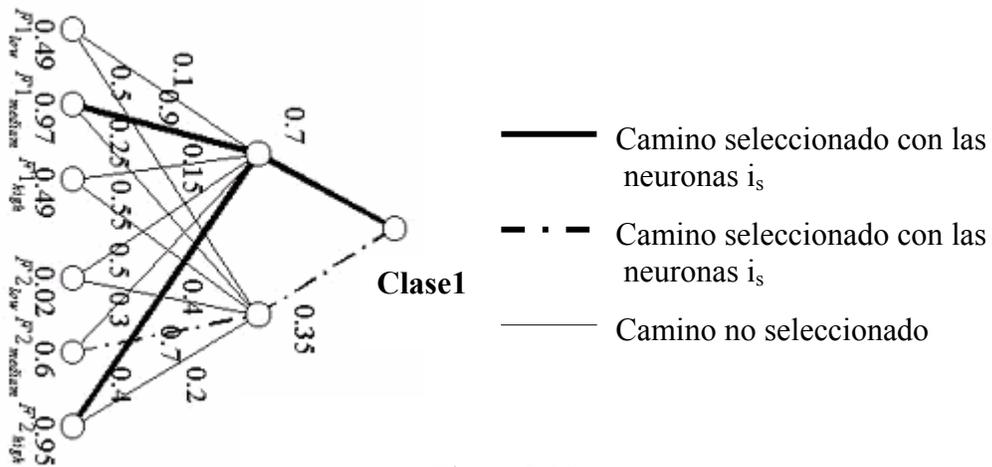


Figura 2.16
Diagrama de selección del camino para una red de tres capas

Sea el camino generado por la ecuación 2.25 un conjunto simple de conexiones, pesos de entrada w_{ji}^h , entradas de activación y_i^0 y las correspondientes etiquetas lingüísticas son descritas en la figura 2.16. En este caso, la regla generada a partir del sistema es:

“ SI F1 es muy médium y F2 es alto, ENTONCES es muy parecido a Clase1”.

Datos sintéticos con solapamiento entre clases predefinido

La capacidad de generalización de la red determina en buena medida las siguientes tres circunstancias: 1) la arquitectura de la red, 2) el número de ejemplos de entrenamiento y 3) la complejidad del problema [13]. Los tres puntos están muy

relacionados; en términos generales, cuanto mas complejo sea el problema a modelar, mas grande deberá ser la red (con más parámetros a ajustar) y, por lo tanto, más ejemplos se necesitaran para entrenarla (ejemplos que deben cubrir todo el espacio de entrada, contemplando todas las situaciones posibles).

A menudo el número de patrones ejemplo disponibles es limitado (reducido), y en proporción el número de parámetros efectivos de la red elegida (grados de libertad) suele ser muy grande. En [14] se demuestra ya lo cual ha sido corroborado por otros autores [13] que una red de n entradas y h neuronas ocultas, con un total de w pesos, requiere un número p de patrones de aprendizaje del orden de $p = w/e$ para proporcionar un error de generalización del orden de e . Así, si queremos que una red alcance un error de generalización del orden de por ejemplo, $e = 0.1$ (10 %), el número de patrones de aprendizaje necesarios p será del orden de $p = 10.w$, expresión que suele dar como indicativa del número aproximado de patrones que serían necesarios para entrenar adecuadamente una red neuronal de w pesos. Por ejemplo para una red [10 5 1] (10 neuronas de entrada, 5 neuronas en la capa oculta y una neurona de salida) que dispone de 61 parámetros, entre pesos y umbrales, el número de patrones necesarios para alcanzar un error del 10 % será de unos 610 lo que representa una cifra de patrones muy alta, no disponible en muchas aplicaciones prácticas. Se ilustra de nuevo la facilidad de incurrir en un sobreaprendizaje al entrenar una red neural, por lo que se toma una relación de 4 Patrones por cada w de la red con un grado de libertad de 1.

Tener en cuenta la llamada dimensionalidad [15], que consiste en que el número de datos necesarios para especificar un mapeo (mapping), en general crece exponencialmente con la dimensión del espacio de entrada, lo que agrava en los problemas de dimensión de entrada elevada el disponer de un número de patrones para el aprendizaje escaso. Disminuyendo el número de parámetros de la red (tamaño) se tiene una relación $P = w/e$ mas favorable. Para los datos sintéticos, una forma de reducirlos consiste en limitar el número de las entradas de la red, pues ello implica la disminución drástica del número de pesos. Por ejemplo una red con 200 entradas, 100 neuronas ocultas y 3 salidas, contendrá del orden de 20.000 pesos, con lo que se necesitan 200.000 patrones para entrenarla adecuadamente con un error de 0.1, si se reduce el número de entradas a 10 (revisando componentes principales a las variables de entrada, empleando ratios, etc.) el número de pesos se reduce a 143 con lo que se precisarían unos 1400 patrones de aprendizaje.

De acuerdo con la justificación anterior, 152 patrones representan una estructura ideal máxima [2 9 2], 2 atributos de entrada, 9 neuronas en capa oculta máximo y 2 neuronas de salida con un error máximo permitido del 25 % lo que nos da un numero de patrones $P = 38 / 0,25 = 152$, incluyendo los umbrales. El número de patrones aumentara a medida que el error permitido disminuye, por ser datos sintéticos para experimentar se decide subjetivamente.

Para la fase de experimentación se diseñaron dos conjuntos de datos bidimensionales, los datos con solapamiento entre clases en forma circular y los

datos con solapamiento entre clases en forma cuadrada. Esto se hizo con el objeto de estudiar si la forma del solapamiento tiene efecto de alguna manera en el proceso de entrenamiento de la red.

Datos sintéticos con solapamiento entre clases en forma circular

Se genera un conjunto de datos con 152 patrones bidimensionales, 91 patrones para entrenar, 15 patrones para prueba y 46 para validar en forma aleatoria. Los valores de los patrones se generaron en el intervalo $[0, 1]$ de esta forma se evita la normalización. En la figura 2.17 se muestran los parámetros a considerar para generar los datos en forma circular.

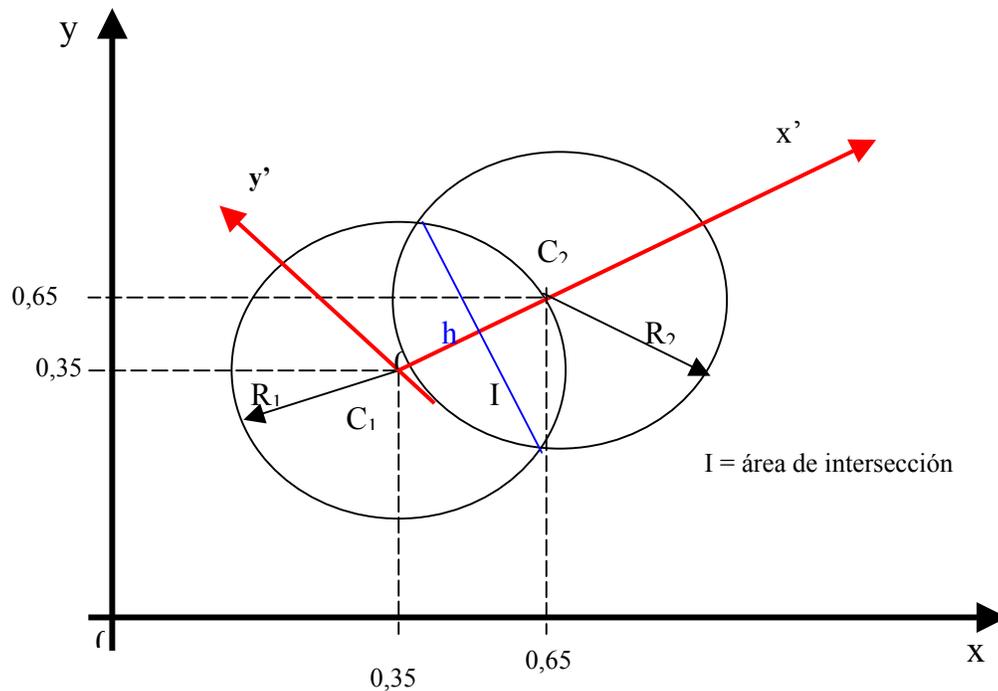


Figura 2.17
Parámetros en la data con solapamiento en zona circular

Las ecuaciones a utilizar para generar estos datos son :

$$C_1 = (0.35, 0.35)$$

$$R_1 = 0.30$$

$$C_2 = (0.65, 0.65)$$

$$R_2 = \text{Variable} \quad (2.27)$$

$$\overline{C_1 C_2} = \sqrt{(0.65 - 0.35)^2 + (0.65 - 0.35)^2} = 0.30\sqrt{2} \cong 0.4243$$

$$\text{Si } I = 0\% \Rightarrow R_2 = 0.4243 - 0.30 = 0.1243$$

$$\text{Si } I = 100\% \Rightarrow R_2 = 0.4243 + 0.30 = 0.7243$$

$$\Rightarrow 0.1243 \leq R_2 \leq 0.7243 \quad (2.28)$$

$$h = \sqrt{0.09 - \frac{(0.27 - R_2)^2}{0.72}} \quad (2.29)$$

$$\%I = \left[0.09 \text{sen}^{-1} \frac{h}{0.3} + R_2 \text{sen}^{-1} \frac{h}{R_2} - 0.30\sqrt{2}h \right] \frac{100}{0.09\pi} \quad (2.30)$$

Datos sintéticos con solapamiento entre clases en forma cuadrada

Se genera un conjunto de datos en forma análoga a la data circular con valores aleatorios en el intervalo [0 1] con el fin de evitar la normalización , este conjunto de datos esta formado por 91 patrones para entrenar la red, 15 patrones para probar la red y 46 patrones para validar la red para un total de 152 patrones bidimensionales. En la figura 2.18 se muestran los parámetros a tomar en cuenta para la generación del conjunto de datos con solapamiento entre clases en forma cuadrada.

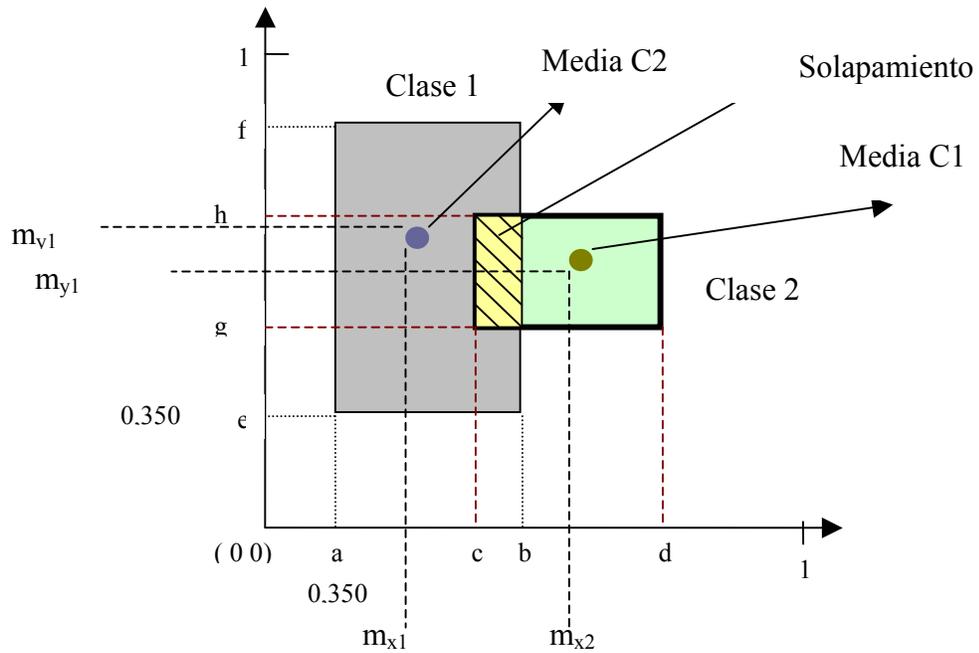


Figura 2.18
 Grafica de los patrones con solapamiento en zona cuadrada

Datos reales con solapamiento entre clases aleatorio

Los datos seleccionados para la investigación representan 4 enfermedades del hígado a través de 9 atributos numéricos, a continuación se describen las enfermedades que están representadas en los datos.

Enfermedades representadas en los datos reales a estudiar en esta investigación

La colecistitis

La colecistitis es la inflamación de la pared de la vesícula biliar y del revestimiento abdominal circundante. Por lo general, la colecistitis se debe a la presencia de cálculos en el conducto cístico, que conecta la vesícula biliar con el conducto

hepático. La causa de la colecistitis en esta investigación se debe a un tumor del páncreas o del hígado (Hematoma primary (PH)) o a cálculos en el conducto cístico (Cholelithiasis (C)). Otras causas de la colecistitis no consideradas en esta investigación son las siguientes:

. Infección bacteriana en el sistema de conductos biliares. El sistema de conductos biliares es un sistema de drenaje que transporta la bilis desde el hígado y la vesícula hasta la zona del intestino delgado llamada duodeno.

. Disminución de la circulación sanguínea a la vesícula biliar. Esto puede suceder a veces con la diabetes.

. Sedimentos en la vesícula biliar. Estos “sedimentos” son un material muy denso que la bilis no puede absorber y que por lo general, afectan a las mujeres embarazadas o a los individuos que han sufrido una rápida pérdida de peso. La colecistitis puede producirse de forma repentina o gradual a lo largo de muchos años.

Cirrosis del hígado (LC)

Una enfermedad inflamatoria degenerativa que da lugar a endurecer y a marcar con una cicatriz las células del hígado. El hígado llega a ser incapaz de funcionar correctamente debido al tejido fino marcado con una cicatriz que previene el paso normal de la sangre a través del hígado. La causa más común es el consumo excesivo de alcohol. La desnutrición y la inflamación crónica pueden también conducir al mal

funcionamiento del hígado. Éstas en las fases posteriores pueden desarrollar anemia, la contusión debido a sangrado debajo de la piel, y el edema.

El hígado desintoxica basuras, amoníaco, residuos de insecticidas/pesticidas, drogas, alcohol, venenos y metales pesados. Los síntomas son el estreñimiento o la diarrea, malestar estomacal, fatiga, debilidad, apetito pobre, pérdida del peso, agrandamiento del hígado, vomitos, las palmas rojas, e ictericia.

Enfermedad del hígado inducida por alcohol (ALD)

El flujo biliar puede estar bloqueado dentro del hígado, fuera del hígado o en ambos sitios.

CAPÍTULO III

MARCO METODOLÓGICO

Tipo de investigación

El tipo de investigación utilizada corresponde al de “proyecto factible”, ya que se trata de una proposición sustentada en un modelo viable, tendente a satisfacer una necesidad, como lo es el modelado con sistemas Neurodifusos de patrones que representan una enfermedad y que además, tienen solapamiento entre ellos. El proyecto se apoyo además en la investigación documental, ya que se han hecho revisiones de aspectos teórico-prácticos relacionados con redes neuronales y lógica difusa.

El trabajo desarrollado se encuentra dentro de la línea de investigación de Inteligencia Artificial de la maestría en Computación. Específicamente en el campo de las redes neuronales y la lógica difusa.

Fases de estudio

Para la culminación del proyecto, se siguieron una serie de fases, contempladas dentro del esquema establecido por el manual para la presentación de trabajos conforme al grado de maestría de la Universidad Centroccidental “Lisandro Alvarado”, estas son:

I Diagnóstica, II de factibilidad, III de diseño del proyecto y IV experimentación y puesta a punto de la propuesta. A continuación hacemos referencia a cada una de estas fases.

Fase I : Diagnóstica

Diseño de la investigación o procedimiento

Posteriormente a la investigación documental, se realizaron las siguientes etapas:

1. Diseño del modelo Neurodifuso, representativo de un sistema híbrido entre redes neuronales y lógica difusa para pruebas de patrones solapados.
2. Generación de conjuntos de datos sintéticos correspondientes a diversos casos de prueba, a partir de variaciones de los parámetros incluidos en el modelo matemático, utilizando la simulación del sistema, los cuales serán usados para la experimentación en el modelado con redes neuronales y lógica difusa.
3. Selección de los datos reales para experimentar.
4. Simulación del sistema para patrones solapados utilizando una herramienta de software (Matlab 6.5).
5. Se interpretan los resultados de las simulaciones realizadas del modelo para determinar la influencia de un etiquetado difuso en el estudio de patrones solapados.

Universo y muestra

El universo está formado por el conjunto de todos los posibles casos de experimentación sobre métodos que se pueden utilizar, y que se generan al combinar diferentes posibilidades de valores de indicadores para diversas variables independientes del problema, correspondiente a número de patrones de entrada, porcentaje de solapamiento de los patrones entre clases, número de atributos por cada patrón, formas de la función de membresía, tipos de algoritmos a probar, máximo de capas ocultas, grupos de neuronas por capa oculta, grupo de iteraciones, estructura de datos sintética. Dado que cada valor de indicadores para diversas variables independientes del problema se pueden fijar independientemente, el número de casos posible está dado por la multiplicación de los números de posibilidades de valores de dichas variables.

Se toma como premisa unos valores prefijados correspondientes al número de posibilidades de valores para las variables antes señaladas, de la siguiente forma.

Posibilidades de valores para:

Grupos Numero de patrones de entrenamiento (GNPE) = 1 (correspondiente a 152)

Distintas porcentajes de solapamiento (DPS) = 4 (correspondiente a 20 %, 30 %, 40 %, 50 %, solapamiento).

Grupos de números de atributos para los patrones de entrada (GAPE) = 1 (correspondiente a 2, atributos de entrada).

Forma de la función de membresía por patrón de entrada (FFM) =1 (correspondiente a la campana tridimensional.

Tipos de algoritmos a probar (TAP) = 1

Máximo de capas ocultas (MCO) = 3

Grupos de neuronas por capa oculta (GNCO) = 3

Grupo de Iteraciones (G I) = 4

Estructura data sintética (EDS) = 2 (Circular y cuadrada)

El universo (U) estará formado por:

$$(U) = (GNPE) \times (DSP) \times (GAPE) \times (FFM) \times (TAP) \times (MCO) \times (GNCO) \times (GI) = \\ 1 \times 4 \times 1 \times 1 \times 1 \times 3 \times 3 \times 4 \times 2 = 288$$

Podemos decir que bajo las premisas anteriores consideramos un universo de 288 casos de experimentación. Para efectos de la fase de prueba inicial del modelo tomaremos (GNPE) = 1; (DSP) = 4 ; (GAPE) =1 (FFM) =1; (TAP) = 1; (MCO) = 3; (GNCO) = 3; (G I) = 4; (EDS) = 1 . Por lo que para una primera aproximación tomaremos una muestra (MU) = de tamaño 144 para los data sintéticos (1 x 4 x 1 x 1 x 1 x 3 x 3 x 4 = 144).

Técnica e instrumentos de recolección de información

En la investigación documental, esta se lleva a cabo a través de consultas en la bibliografía de referencia, búsqueda en Internet y en bibliotecas de universidades

nacionales, entre ellas la UCLA, UNET, ULA, UCV, UFT y YACAMBU. Con respecto a la recolección de datos numéricos requeridos para la simulación inicial y evaluación, ésta se realiza en forma automática con el computador. Se desarrolla el software para que esto pueda realizarse adecuadamente.

Técnicas de análisis de datos

Los resultados obtenidos para los diversos modelos de sistemas Neurodifusos para el estudio de patrones con solapamiento de clases con un determinado algoritmo, se comparan sobre la base del error de aproximación y el número de parámetros de cada modelo.

La comparación de error de la aproximación nos permitirá estimar cual de las alternativas produce modelos de mayor exactitud.

Fase II: Estudio de factibilidad

Se han analizado la factibilidad técnica, operativa y económica del proyecto, estas se detallan a continuación.

Factibilidad técnica

La factibilidad técnica corresponde a un estudio de la funcionalidad, el rendimiento y las restricciones que pueden afectar la posibilidad de realización un de sistema aceptable. Podemos decir que el proyecto es factible técnicamente ya que existen en

el país los recursos humanos necesarios para llevarlo a cabo y producir un resultado eficiente y confiable, a un costo razonable.

Se dispone de la bibliografía adecuada, así como de la ayuda y colaboración de expertos en áreas afines a la temática del proyecto.

Factibilidad operativa

El resultado del trabajo de investigación se considera factible operativamente, ya que se establece un método que es aplicable en forma sistemática y razonablemente sencilla para lograr el modelado Neurodifuso deseado.

Factibilidad económica

La posibilidad financiera de este proyecto de investigación se ha establecido en función de si la inversión económica a realizar se encuentra dentro de un rango tolerable. Para ello se efectuó una estimación de gastos, la cual se resume en la siguiente tabla 3.1

Descripción	Costo (Bs)
Viaticos para consultas	350,000
Viaticos para consultas en bibliotecas	350,000
Adquisición de Bibliografía	600,000
Materiales	800,000
Aranceles y otros	800,000

Tabla 3.1 Estimación de gastos

La inversión económica requerida se considera razonable, y se tienen los recursos financieros para ello.

Fase III: Diseño de la propuesta.

1. Construcción de modelos con redes neuronales con lógica difusa para patrones con solapamientos de clases con diversos tipos de estructura, que puedan predecir satisfactoriamente la salida del sistema.
2. Evaluación de los modelos implementados con base a los errores de predicción.
3. Formulación del método para construir el modelo deseado, basándonos en la revisión de los pasos, algoritmos y rasgos que hayan producido los mejores resultados.

Fase IV: Experimentación y puesta a punto de la propuesta

1. Perfeccionamiento del método para el modelado de sistemas Neurodifusos que se experimenta más extensivamente.
2. Evaluación de los resultados obtenidos, en función de los errores de predicción, tiempos de entrenamiento y tiempos de simulación.
3. Desarrollo de conclusiones y recomendaciones oportunas.

CAPÍTULO IV

PROPUESTA

Generación de los datos sintéticos en forma circular

La generación de los datos en forma circular para entrenar, probar y validar la red se generó a través de un programa desarrollado en Matlab 6.5 con las características de solapamiento definidas a través del radio de las circunferencias. En la figura 4.1 se muestra el listado documentado donde se definen los porcentajes de solapamiento de este programa.

```
% Solapamiento con Distribución Uniforme del Número de Patrones de entrenamiento de dos clase
%-----
clear all;
%-----
% Definición de los limites en el eje x, y en eje y utilizando las ecuaciones (2.28),(2.29) y (2.30)
%-----
% radio2 = 0.1838 para un solapamiento del 10%
% radio2 = 0.2489 para un solapamiento del 20%
% radio2 = 0.3021 para un solapamiento del 30%
% radio2 = 0.3498 para un solapamiento del 40%
% radio2 = 0.3945 para un solapamiento del 50%
centro_x1 = 0.35
centro_y1 = 0.35
centro_x2 = 0.60
centro_y2 = 0.60
radio1 = 0.30;
radio2 = 0.1838;
tot_muestras = 91
%-----
% Definición del porcentaje de solapamiento
%-----
porc_overlap = input('Porcentaje de Solapamiento (%):')
mitad_muestras = tot_muestras / 2;
num_overlap = porc_overlap * mitad_muestras / 100;
mitad_porc_overlap = num_overlap / 2;
sum_overlap = 0;
cuenta = 0;
cuenta1 = 0;
cuenta2 = 0;
```

Figura 4.1
Parte del programa que define el solapamiento circular

En la figura 4.2 se muestra la parte del programa donde se generan los patrones con el solapamiento previamente definido.

```
%-----  
% Generación de los patrones para las dos clases  
%-----  
while sum_overlap < num_overlap  
    x = rand(1);  
    y = rand(1);  
    mod1 = sqrt((x - centro_x1)^2 + (y - centro_y1)^2);  
    mod2 = sqrt((x - centro_x2)^2 + (y - centro_y2)^2);  
    if (mod1 < radio1) & (mod2 < radio2)  
        sum_overlap = sum_overlap + 1;  
        cuenta1 = cuenta1 + 1;  
        clase1(cuenta1) = 1;  
        datos1(1, cuenta1) = x;  
        datos1(2, cuenta1) = y;  
        cuenta = cuenta + 1;  
    end  
end  
sol_1=cuenta1  
sol_2=cuenta2  
cuenta  
resto = sum_overlap;  
while resto < tot_muestras  
    x = rand(1);  
    y = rand(1);  
    mod1 = sqrt((x - centro_x1)^2 + (y - centro_y1)^2);  
    mod2 = sqrt((x - centro_x2)^2 + (y - centro_y2)^2);  
    if (mod1 < radio1) & (mod2 > radio2)  
        if (cuenta1 < mitad_muestras)  
            resto = resto + 1;  
            cuenta1 = cuenta1 + 1;  
            datos1(1, cuenta1) = x;  
            datos1(2, cuenta1) = y;  
            clase1(cuenta1) = 1;  
        end  
        else  
            if (mod2 < radio2)  
                if (cuenta2 < mitad_muestras)  
                    resto = resto + 1;  
                    cuenta2 = cuenta2 + 1;  
                    datos2(1, cuenta2) = x;  
                    datos2(2, cuenta2) = y;  
                    clase2(cuenta2) = 0;  
                end  
            end  
        end  
    end  
end  
end  
end
```

Figura 4.2
Programa que genera los patrones con solapamiento circular

En la figura 4.3 se muestra la parte del programa donde se grafica y almacenan en un formato tipo hoja de cálculo los patrones generados con el solapamiento previamente definido.

```

%-----
% Gráfica de los patrones en cada clase
%-----
plot(datos1(1,:),datos1(2,:),'r');
hold on;
plot(datos1(1,1:sol_1),datos1(2,1:sol_1),'or');
plot(datos2(1,:),datos2(2,:),'g');
plot(datos2(1,1:sol_2),datos2(2,1:sol_2),'og');
hold off;
datos = [datos1, datos2];
clases = [clase1, clase2];
%-----
% Almacena los patrones de cada clase
%-----
save entrecirc10 datos1 datos2 clase1 clase2 tot_muestras

```

Figura 4.3
Programa que gráfica y almacena los patrones con solapamiento circular

En la figura 4.4 se muestran las gráficas generadas al ejecutar el programa de los patrones de entrenamiento con áreas de solapamiento del 10 % y 50 % . Los patrones solapados aparecen con círculo alrededor.

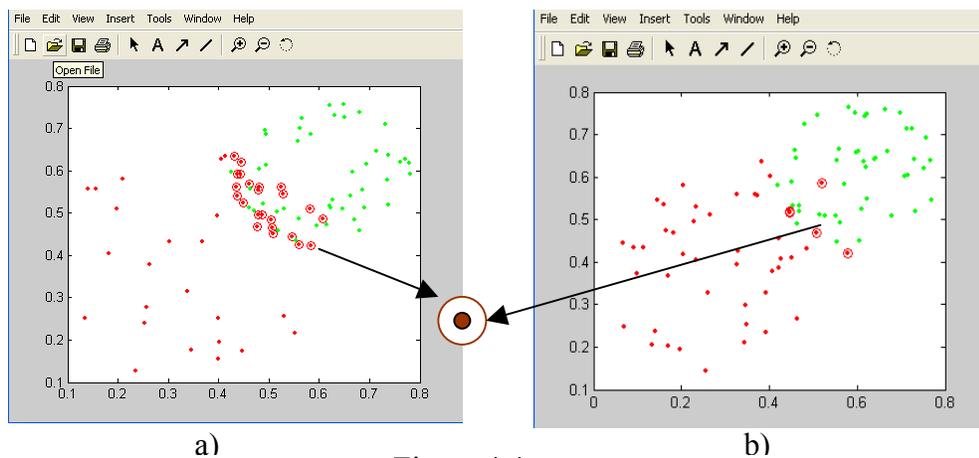


Figura 4.4
Gráfica de los patrones con solapamiento en zona circular
a) 50 % solapamiento b) 10 % solapamiento

Generación de los datos sintéticos en forma cuadrada

La generación de los datos en forma cuadrada para entrenar, probar y validar la red se genero través de un programa similar al desarrollado para generar los datos en forma circular con las características de solapamiento definidas a través de los porcentajes de solapamiento de las áreas de los cuadrados. En la figura 4.5 se muestra el listado documentado donde se definen los porcentajes de solapamiento.

```
% -----  
% Solapamiento con Distribución Uniforme del Número de Patrones de dos clase  
% -----  
clear all;  
% -----  
% Definición de los limites en el eje x, y en eje y  
% -----  
lim_sup_solx = 0.600;  
lim_sup_soly = 0.750;  
lim_inf_solx = 0.350;  
lim_inf_soly = 0.350;  
tot_muestras = 152;  
% -----  
% Definición del porcentaje de solapamiento  
% -----  
porc_overlap = input('Porcentaje de Solapamiento (%) :')  
num_overlap = porc_overlap * tot_muestras /100;  
mitad_porc_overlap = num_overlap / 2  
mitad_muestras = tot_muestras / 2;  
sum_overlap = 0;  
cuenta = 0;  
cuenta1 = 0;  
cuenta2 = 0;
```

Figura 4.5
Parte del programa que define el solapamiento cuadrado

En la figura 4.6 se muestra la parte del programa donde se generan los patrones con el solapamiento en forma cuadrada previamente definido. Este solapamiento se definió a través de los limites inferior y superior de los cuadrados.

```

%-----
%Generacion de los patrones para cada clase
%-----
while sum_overlap < num_overlap
x = rand(1);
y = rand(1);
if ( lim_inf_solx <= x ) & ( x <= lim_sup_solx ) & ( lim_inf_soly <= y ) & ( y <= lim_sup_soly )
sum_overlap = sum_overlap + 1;
if cuenta > ( mitad_porc_overlap - 1 )
cuenta1 = cuenta1 + 1;
clase1 (cuenta1) = 1;
datos1 (1, cuenta1) = x;
datos1 (2, cuenta1) = y;
else
cuenta2 = cuenta2 + 1;
clase1 (cuenta2) = 0;
datos2 (1, cuenta2) = x;
datos2 (2, cuenta2) = y;
end
cuenta = cuenta + 1;
end
end
sol_1=cuenta1
sol_2=cuenta2
cuenta
resto = sum_overlap;
while resto < tot_muestras
x = rand(1);
y = rand(1);
if ( lim_inf_solx <= x ) & ( x <= lim_sup_solx )
if ( y < lim_inf_soly ) | ( y > lim_sup_soly )
resto = resto + 1;
cuenta1 = cuenta1 + 1;
datos1 (1, cuenta1) = x;
datos1 (2, cuenta1) = y;
clase1 (cuenta1) = 1;
end
else
if ( x < lim_inf_solx )
resto = resto + 1;
cuenta1 = cuenta1 + 1;
datos1 (1, cuenta1) = x;
datos1 (2, cuenta1) = y;
clase1 (cuenta1) = 1;
else
if ( y < lim_sup_soly ) & ( y > lim_inf_soly )
resto = resto + 1;
cuenta2 = cuenta2 + 1;
datos2 (1, cuenta2) = x;
datos2 (2, cuenta2) = y;
clase2 (cuenta2) = 0;
end
end
end
end
end
end

```

Figura 4.6
Programa que genera los patrones con solapamiento cuadrado

En la figura 4.7 se muestra la parte del programa donde se grafica y almacenan en una data tipo hoja de cálculo los patrones generados con el solapamiento en forma cuadrada previamente definido.

```
% -----  
% Grafica de los patrones en cada clase  
% -----  
plot(datos1(1,:),datos1(2:),'r');  
hold on;  
plot(datos2(1,:),datos2(2:),'g');  
hold off;  
% -----  
% Almacena los patrones de cada clase  
% -----  
save entreccuad10 datos1 datos2 clase1 clase2 tot_muestras
```

Figura 4.7
Programa que grafica y almacena los patrones con solapamiento cuadrado

En la figura 4.8 se muestran las gráficas generadas al correr el programa de los patrones de entrenamiento con áreas de solapamiento del 10 % y 50 % . Los patrones solapados aparecen en rojo y azul con áreas de solapamiento del 10 % y 50 %.

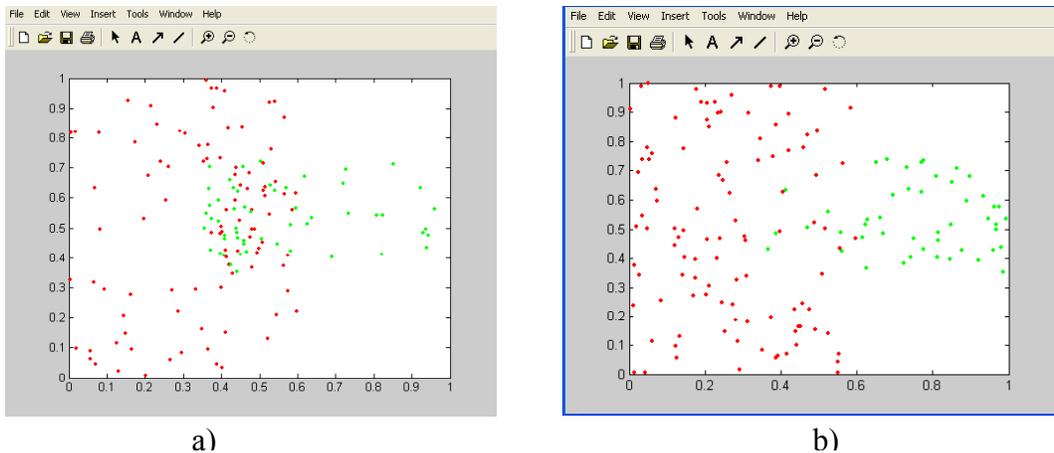


Figura 4.8
Parámetros con los datos de solapamiento en zona cuadrada
a) 50 % solapamiento b) 10 % solapamiento

Algoritmo propuesto para el etiquetamiento difuso de los patrones

El algoritmo general para el etiquetamiento difuso en forma generalizada [7] se resume como:

Inicio

- 1.- Generar el conjunto de patrones de entrada bidimensionales $F_i [X_j ; Y_j]$.
- 2.- Difuzificación de la entrada.
 - 2.1 Calcular la media de los atributos X_j , Y_j del total de patrones (m_j) .
 - 2.2 Buscar el valor mínimo (F_{\min}) y el valor máximo (F_{\max}) de X_j y Y_j .
 - 2.3 Contar cuantos valores de X_j y Y_j están por arriba y por debajo de la media.
 - 2.4 Calcular el promedio de los valores que están por debajo y por arriba de la media de X_j y Y_j (m_{jl} m_{jh}).
 - 2.5 .- Calcular

$$C_{\text{medium}(F_j)} = m_j$$

$$C_{\text{low}(F_j)} = m_{jl}$$

$$C_{\text{high}(F_j)} = m_{jh}$$

$$\lambda_{\text{low}(F_j)} = 2 (c_{\text{medium}(F_j)} - c_{\text{low}(F_j)})$$

$$\lambda_{\text{high}(F_j)} = 2 (c_{\text{high}(F_j)} - c_{\text{medium}(F_j)})$$

$$\lambda_{\text{medium}(F_j)} = \text{fno} * \frac{\lambda_{\text{low}(F_j)} * (F_{j \max} - c_{\text{medium}(F_j)}) + \lambda_{\text{high}(F_j)} * (c_{\text{medium}(F_j)} - F_{j \min})}{F_{j \max} - F_{j \min}}$$

- 2.6 Utilizar la función π de la ecuación 2.17 con los valores calculados en el paso 2.5 para calcular los valores de pertenencia bajo, medio y alto para

cada atributo de cada patrón, $F_i = [X_{ibajo} X_{imedio} X_{ialto}, Y_{ibajo} Y_{imedio} Y_{ialto}]$ y este vector es la entrada a la red.

3. Calcular las salidas deseadas.

$$z_{ik} = \sqrt{\sum_{j=1}^n \left[\frac{F_{ij} - m_k}{v_{kj}} \right]^2}$$

4. Calcular las etiquetas difusas.

$$\mu_1(F_i) = \frac{1}{1 + \left(\frac{z_{i1}}{f_d} \right)^{f_e}} \quad ; \quad \mu_2(F_i) = \frac{1}{1 + \left(\frac{z_{i2}}{f_d} \right)^{f_e}}$$

5. Guardar el vector difuso de entrada y las etiquetas de salida difusas.

Fin.

Los valores de f_{no} , f_d y f_e para esta investigación se toman fijos a 0.8, 5 y 1 respectivamente [7].

Para el etiquetamiento difuso de los patrones de entrada para los datos sintéticos como para los datos hepáticos se desarrollo bajo el ambiente gráfico de Matlab (GUI). El algoritmo implementado es el de razonamiento aproximado basado en redes neuronales de retropropagación (FMLP) [7]. En la figura 4.9 se muestra la pantalla de este programa donde se pide el nombre de los datos a difusificar, la extensión del solapamiento (f_{no}), el factor del denominador (f_d), el factor

exponencial (f_e) y el aumento de contraste, al oprimir el botón de difusificar, el programa genera una gráfica con las curvas bajo, medio y alto, para cada atributo del vector de entrada y genera una matriz n-dimensional que es la entrada a la red multicapa.

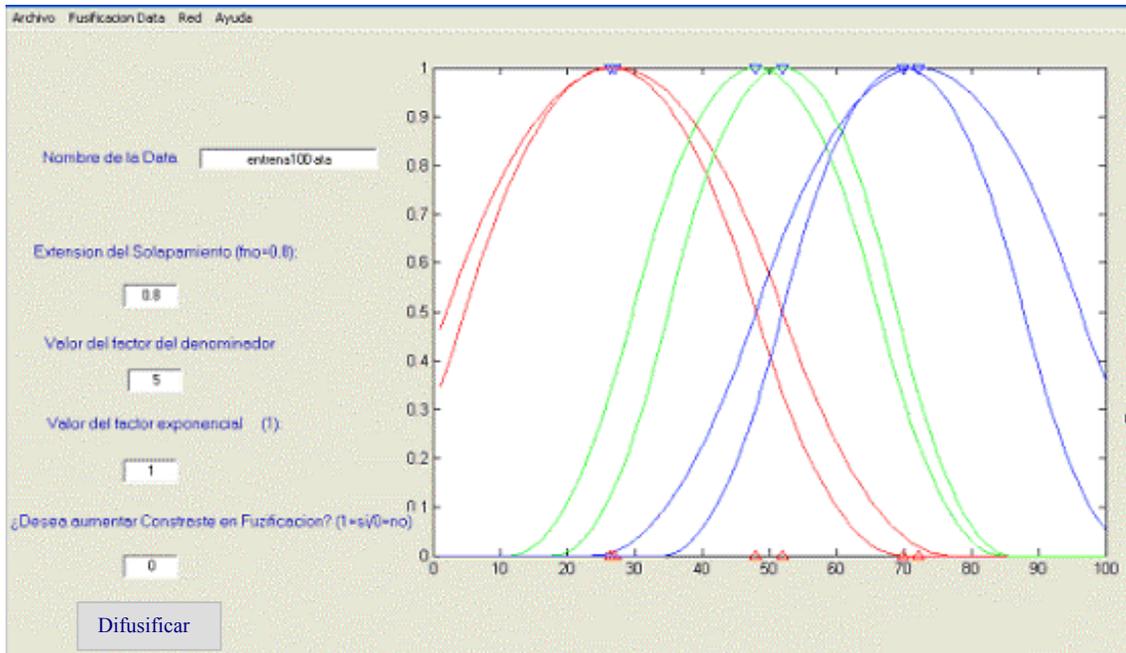


Figura 4.9
Programa para difusificar los patrones de entrada

Algoritmo para el entrenamiento de la red

Los patrones después de pasar por el proceso de difusificación entran a la red, los datos para el entrenamiento de la red se ingresan en la pantalla como se muestra en la figura 4.10. En esta pantalla se pide el número de capas ocultas de la red, el número de neuronas por capa oculta, la tasa de aprendizaje, el número de iteraciones y el

error máximo permitido, al oprimir el botón entrenar se despliega la gráfica dinámica que se muestra a la derecha de la figura 4.10.

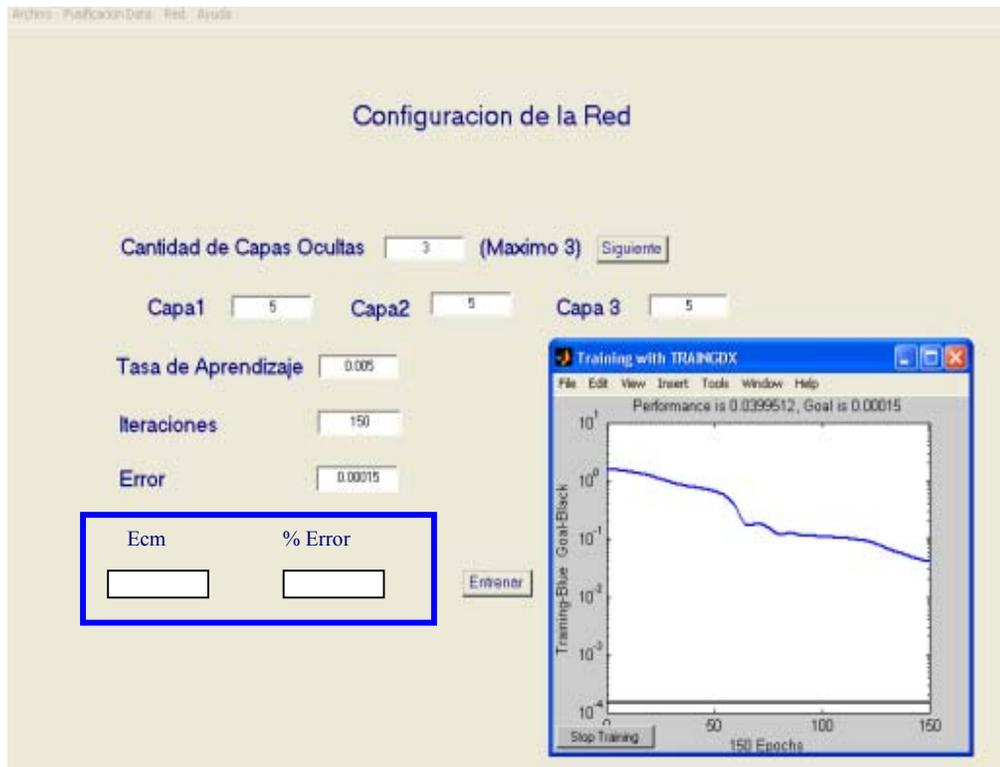


Figura 4.10.
Pantalla que pide los datos para entrenar la red

Al detenerse la gráfica dinámica que esta indicando el desarrollo del entrenamiento (Iteraciones vs error cuadrático medio) se muestra la gráfica del desempeño de la red con los datos de entrenamiento (patrones mal clasificados) como se muestra en la figura 4.11. El círculo verde con el centro en rojo representa un patrón que se clasificó bien, los puntos rojos solos o los círculos verdes solos que aparecen en la gráfica representan patrones mal clasificados por la red.

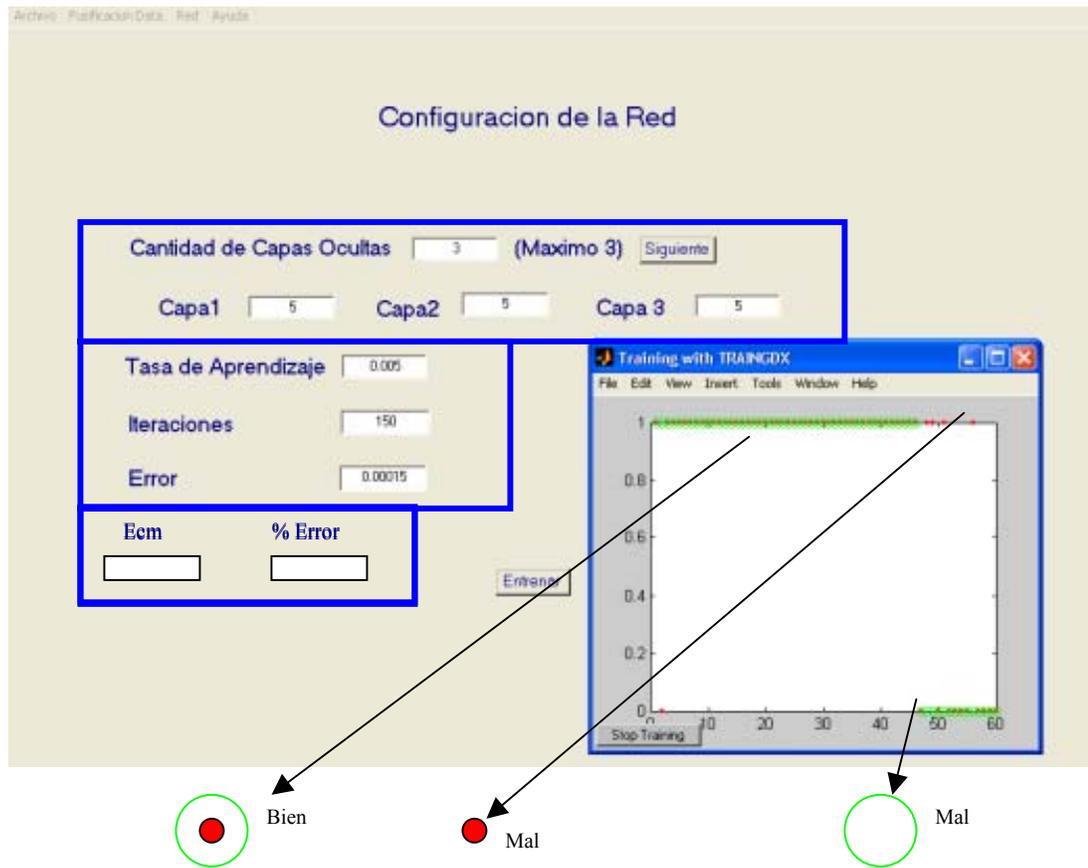


Figura 4.11
Pantalla que muestra los patrones clasificados

Finalmente, se muestra la gráfica que refleja la cantidad de errores cometidos por la red después del proceso de entrenamiento, si el resultado que se muestra en esta gráfica es una línea recta horizontal esto indica que no se cometieron errores al simular la red después del entrenamiento con los patrones de entrada, en caso de presentarse picos esto indica la cantidad de errores que comete la red después de la simulación, en la figura 4.12 se muestran los errores que se cometen en la validación de la red.

Prueba y validación de la Red

Para probar la red y validarla se sigue un procedimiento análogo al anteriormente descrito, la diferencia esta en que para el procedimiento de prueba se toman los datos de prueba y para el procedimiento de validación se toman los datos de validación.

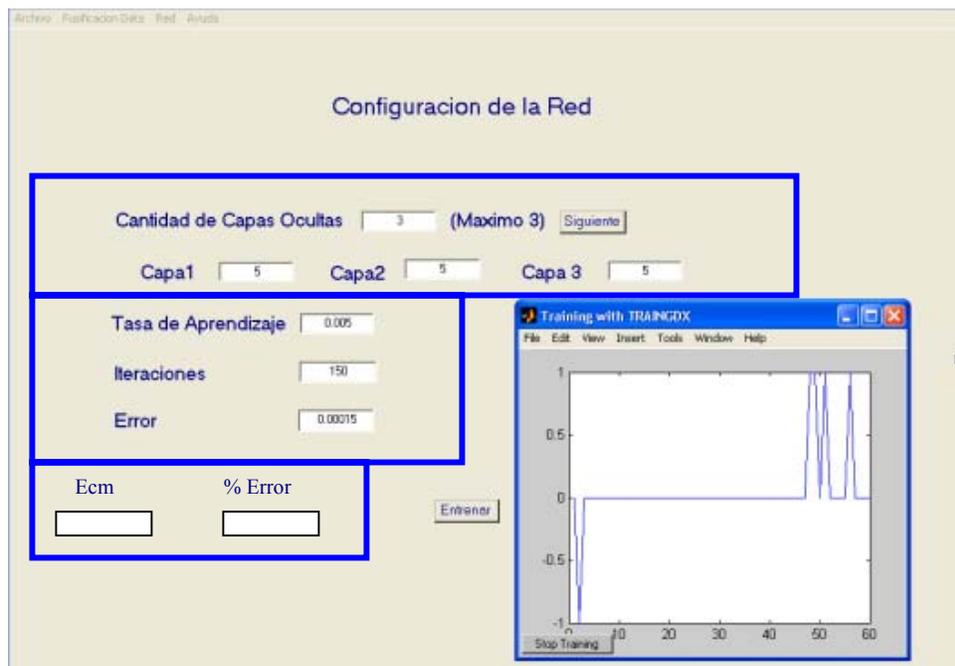


Figura 4.12
Pantalla que muestra los patrones clasificados por clase

CAPÍTULO V

RESULTADOS OBTENIDOS

Resultados obtenidos para los datos sintéticos

Para los datos sintéticos en forma circular y para los datos sintéticos en forma cuadrada se realizaron un total de 144 experimentos por tipo de datos, como ejemplo se muestran los experimentos realizados con los datos en forma circular para un solapamiento del 40 % y los resultados obtenidos para una red con 6 neuronas en la capa de entrada, una capa oculta con 5, 10 y 15 neuronas, y una capa de salida con 2 neuronas, variando el número de iteraciones entre 1000 y 10.000 se muestran en la tabla 5.1.

Red [6 5 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,0667	6,6667	0,1522	15,2174
3000	0,044	4,3956	0,2	20	0,2391	23,913
5000	0	0	0,2667	26,6667	0,1957	19,5652
10000	0	0	0,2	20	0,2391	23,913
Red [6 10 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,29	0,2	20	0,1087	10,8696
3000	0,033	3,2967	0,1333	13,3333	0,1522	15,2174
5000	0,022	2,1978	0,1333	13,3333	0,2174	21,7391
10000	0	0	0,2	20	0,1739	17,3913
Red [6 15 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1522	15,2174
3000	0,022	2,1978	0,2	20	0,1957	19,5652
5000	0,011	1,0989	0,1333	13,3333	0,2174	21,7391
10000	0,011	1,0989	0,1333	13,3333	0,1304	13,0435

Tabla 5.1
Resultados obtenidos para un solapamiento del 40 % y una capa escondida

Los experimentos realizados y los resultados obtenidos para una red con dos capas ocultas, 40 % de solapamiento, con 5, 10 y 15 neuronas por capa oculta y variando el numero de iteraciones entre 1000 y 10000 se muestran en la tabla 5.2.

Red [6 5 5 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,011	1,0989	0,2	20	0,1087	10,8696
3000	0,011	1,0989	0,2667	26,6667	0,1956	19,5652
5000	0	0	0,2	20	0,1739	17,3913
10000	0,022	2,19	0,1333	13,3333	0,1304	13,0435
Red [6 10 10 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,0549	5,4945	0,2	20	0,1957	19,5652
3000	0	0	0,2	20	0,1522	15,2174
5000	0,011	1,0987	0,26	26,6667	0,1304	13,0435
10000	0	0	0,2	20	0,1304	13,0435
Red [6 15 15 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435
3000	0	0	0,2	20	0,2391	23,913
5000	0	0	0,2667	26,6667	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,2826	28,2609

Tabla 5.2
Resultados obtenidos para un solapamiento del 40 % y dos capas escondidas

Los experimentos realizados y los resultados obtenidos para una red con tres capas ocultas, 40 % de solapamiento, con 5, 10 y 15 neuronas por capa oculta y variando el numero de iteraciones entre 1000 y 10000 se muestran en la tabla 5.3. En los anexos se presentan los experimentos y los resultados para 10 % 20 %, 30 % y 50 % de solapamiento de los datos en forma circular y los experimentos con los resultados

obtenidos para 10 % 20 % 30 % 40 % y 50 % de solapamiento de los datos en forma cuadrada organizados en forma análoga.

Red [6 5 5 5 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,011	1,0989	0,2	20	0,2826	28,2609
3000	0,02	2,1973	0,2667	26,6667	0,1521	15,2174
5000	0	0	0,2	20	0,1957	19,5652
10000	0	0	0,2	20	0,1739	17,3913
Red [6 10 10 10 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1739	17,3913
3000	0,011	1,0989	0,2667	26,6667	0,1957	19,5652
5000	0,022	2,1978	0,1333	13,3333	0,1739	17,3913
10000	0	0	0,1333	13,3333	0,1739	17,3913
Red [6 15 15 15 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		0,2607		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,2	20	0,1304	13,0435
3000	0	0	0,2	20	0,1739	17,3913
5000	0,022	2,1978	0,1333	13,3333	0,2609	26,087
10000	0	0	0,2	20	0,2174	21,7391

Tabla 5.3
Resultados obtenidos para un solapamiento del 40 % y tres capas escondidas

Resultados sobresalientes para los datos circulares

Las redes que arrojaron mejores resultados para cada uno de los porcentajes de solapamientos de los datos sintéticos en forma circular se muestran en la tabla 5.4.

Data Circular				
% Solapamiento	Red	ECM	% Error	Iteraciones
10	[6 15 15 15 2]	0,10870	10,86760	10000
20	[6 15 15 15 2]	0,10870	10,86760	5000
30	[6 15 15 2]	0,10870	10,86760	10000
40	[6 10 10 2]	0,13040	13,04350	10000
50	[6 15 15 15 2]	0,17390	17,39870	10000

Tabla 5.4
Mejores resultados obtenidos para cada uno de los porcentajes de solapamiento

Generación de reglas para los resultados sobresalientes de los datos circulares

La generación de las reglas se hace a través del proceso de justificación, se toma la red que presento el mejor rendimiento por cada porcentaje de solapamiento, se selecciona el mayor de los pesos desde la ultima capa hasta la primera y se genera la regla de acuerdo a los valores de los pesos. Como ejemplo se toma el caso para el solapamiento del 40 %. En la tabla 5.5 se muestran los valores de los pesos para un patrón de entrada cualquiera al cual se le conoce su salida, el camino desde la primera capa (Capa 0) hasta la ultima capa (Capa 3) siguiendo los valores mayores de los pesos en cada capa (Valores en rojo) justifican la regla.

Para Data 40% Solapamiento [6 10 10 2] 10000 ITERACIONES										
PATRON DE ENTRADA										
Neurona	X_{low}	X_{medium}	X_{high}	Y_{low}	Y_{medium}	Y_{high}	SALIDAS			
	0,06701	0,75515	0,87359	0,88907	0,02442	0,00000	Clase 1	Clase 2		
	1	2	3	4	5	6	0,99386	0,000		
CAP3 Sid	0,88532	0,76865								
CAPA 2	0,78721	0,56944	0,25931	0,54027	0,21365	0,84775	0,31226	0,00440	0,35632	0,09484
CAPA 1	0,98948	0,10718	0,56746	0,21449	0,67916	0,12182	0,03117	0,82832	0,89658	0,98805
CAP0 Ent	0,60644	0,10892	0,39812	0,95170	0,83242	0,98375				

Tabla 5.5
Tabla que muestra el camino que justifica la regla

Las reglas se basan en los valores de bel_j^H definidos en la ecuación 2.23 donde la regla dependen de:

1. Muy parecido Para $0.8 \leq bel_j^H \leq 1$.
2. Parecido Para $0.6 \leq bel_j^H < 0.8$.
3. Mas o menos Para $0.4 \leq bel_j^H < 0.6$.
4. No Parecido Para $0.1 \leq bel_j^H < 0.4$.
5. Incapaz de Reconocer Para $bel_j^H < 0.1$.

La regla para el patrón representado en la tabla 4.5 se muestra en la figura 5.1

REGLA:	SI X ES MUY PARECIDO A high y Y ES MUY PARECIDO A low ALTAMENTE PROBABLE CLASE 1
---------------	--

Las reglas para un patrón específico cualquiera en cada configuración mostrada en la tabla 5.4 se muestran en la tabla 5.6.

% Slp	Reglas	
10	REGLA:	SI X ES MUY PARECIDO A low y Y ES MUY PARECIDO A medium ALTAMENTE PROBABLE CLASE 1
20	REGLA:	SI X ES PARECIDO A high y Y ES MUY PARECIDO A MEDIUM ALTAMENTE PROBABLE CLASE 2
30	REGLA:	SI X ES MUY PARECIDO A high y Y ES MUY PARECIDO A low ALTAMENTE PROBABLE CLASE 1
40	REGLA:	SI X ES MUY PARECIDO A high y Y ES MUY PARECIDO A low ALTAMENTE PROBABLE CLASE 1
50	REGLA:	SI X ES MUY PARECIDO A mediun y Y ES MUY PARECIDO A mediun ALTAMENTE PROBABLE CLASE 2

Tabla 5.6
Reglas para un patrón en las mejores redes

El conjunto de reglas que represente el total de patrones de entrenamiento en la mejor red será el universo de las reglas y se generan por los pesos los cuales representan la base de conocimiento dentro de la red.

Experimentación con los datos reales

Los datos reales seleccionados para experimentar con el algoritmo FMLP se llaman datos Hepatos [31] los cuales representan desordenes hematológicos que caracterizan cuatro enfermedades hepáticas, estas son, daño severo al hígado (ALD), Hepatoma primario (PH), cirrosis de hígado (LS) y coleditiasis (C) Estos datos consisten de 536 patrones (Filas) con nueve atributos para cada patrón, estos atributos representan los resultados de diferentes exámenes médicos, las unidades no se toman en cuenta para los experimentos. Estos atributos son: Glutamic Oxalicetic

Transaminase (GOT), Glutamic Pyruvic transaminase (GPT), Lactate dehydrase (LDH), Gamma glutamyl transpeptidase (GGT), Bloon urea nitrogen (BUN), Mean corpuscular Volume (MCV), Mean corpuscular hemoglobin (MCH), Total bilirubin (Tbil) y creatinine (C). Una muestra de 6 patrones de estos datos hepáticos se representa en la tabla 5.7. Los datos completos se muestran en los anexos.

Clase	GOT	GPT	LDH	GGT	BUN	MCV	MCH	TBIL	CRTNN
1	290	89	624	120	21	114	37.5	29.8	1
1	53	25	283	182	12	99.2	32.6	2.4	0.9
1	154	96	476	967	8	103.4	31.3	9.4	1
1	78	58	352	535	9	99.7	34.3	2.1	1.4
1	35	18	253	342	22.8	106	34.5	1	1.5
1	35	13	276	34	22.4	90.3	30.5	0.8	0.8

Tabla 5.7
Muestra de los datos hepato

Difusificación de los datos hepatos

El proceso de difusificación de los datos hepatos es similar al utilizado para difusificar los datos sintéticos, la diferencia esta en que en esta, primero se tiene que normalizar los datos entre cero y uno antes de difusificar.

Resultados obtenidos para los datos hepatos (Real)

Se toman como entrada a la red la matriz P , resultante de aplicar el algoritmo de difusificación que genera tres entradas difusas por cada entrada nítida mediante el programa propuesto mostrado en la figura 4.9 , y como salida una matriz d resultante de difusificar las salidas: un patrón de salida con un valor 1, 2 3 o 4, es transformado en un vector con 4 salidas (Tres de las cuales son ceros y la cuarta corresponde al grado de pertenencia a la clase correspondiente). Se diseño un conjunto de

experimentos para determinar la configuración de la red con mejor desempeño, experimentando con el número de capas ocultas, el número de neuronas por capa y la cantidad de iteraciones para el entrenamiento. Se seleccionaron 376 patrones para entrenar y 160 para validar. Como ejemplo se muestran en la tabla 5.8 los resultados de la red con una capa oculta y un máximo de 25 neuronas en esta capa para 1000 iteraciones.

		Solapamiento Aleatorio Data Hepato 2000 iteraciones				
Número de Capas Ocultas		Capa 1				
No. Neuronas por capa		5	10	15	20	25
Error en Porc. (% prueba)		25,531900	19,946800	13,829800	12,500000	7,978700
	Para la Clase 1	37,037000	28,395100	23,456800	19,753100	8,642000
	Para la Clase 2	9,600000	6,400000	2,400000	2,400000	4,800000
	Para la Clase 3	44,827600	35,632200	24,137900	26,436800	14,942500
	Para la Clase 4	18,072300	15,662700	10,843400	6,024100	4,819300
Error en Porc. (% validacion)		36,875000	39,375000	29,375000	34,375000	33,125000
	Para la Clase 1	54,285700	60,000000	45,714300	57,142900	40,000000
	Para la Clase 2	22,641500	28,301900	16,981100	24,528300	30,188700
	Para la Clase 3	48,648600	48,648600	32,432400	43,243200	37,837800
	Para la Clase 4	28,571400	25,714300	28,571400	17,142900	25,714300
MSE (Prueba)		0.063158	0.052773	0.0428851	0.0416368	0.0357516
MSE (Validación)		1,133130	1,256300	1,012500	1,131300	1,037500

Tabla 5.8
Resultados para la red con una capa oculta y 1000 iteraciones

Los resultados de la red con una capa oculta y un máximo de 25 neuronas en esta capa para 2000 iteraciones se muestran en la tabla 5.9

		Solapamiento Aleatorio Data Hepato 1000 iteraciones				
Número de Capas Ocultas		Capa 1				
No. Neuronas por capa		5	10	15	20	25
Error en Porc. (% prueba)		29,7872	23,1383	16,4894	15,159600	15,1596
	Para la Clase 1	25,9259	32,0988	22,2222	25,925900	29,6296
	Para la Clase 2	7,2000	12,8000	7,2000	8,000000	4,8000
	Para la Clase 3	93,1000	35,6322	31,0345	20,689700	21,8391
	Para la Clase 4	1,2048	16,8675	9,6386	9,638600	9,6386
Error en Porc. (% validacion)		41,8750	35,0000	35,0000	29,375000	34,3750
	Para la Clase 1	60,0000	42,8571	48,5714	42,857100	60,0000
	Para la Clase 2	16,9811	30,1887	26,4151	20,754700	18,8679
	Para la Clase 3	94,5946	40,5405	45,9459	35,135100	45,9459
	Para la Clase 4	5,7143	28,5714	22,8571	22,857100	20,0000
MSE (Prueba)		0.0753779	0.0576188	0.049973	0.0474856	0.0478009
MSE (Validación)		1,2688	1,1187	1,0750	1,043800	0,9625

Tabla 5.9
Resultados para la red con una capa oculta y 2000 iteraciones

Los resultados de la red con una capa oculta y un máximo de 25 neuronas en esta capa para 3000 iteraciones se muestran en la tabla 5.10

		Solapamiento Aleatorio Data Hepato 3000 iteraciones				
Número de Capas Ocultas		Capa 1				
No. Neuronas por capa		5	10	15	20	25
Error en Porc. (% prueba)		23,670200	14,893600	11,436200	7,446800	5,851100
Para la Clase 1		40,740700	17,284000	19,753100	13,580200	9,876500
Para la Clase 2		11,200000	3,200000	3,200000	1,600000	4,000000
Para la Clase 3		29,885100	29,885100	17,241400	11,494300	5,747100
Para la Clase 4		19,277100	14,457800	9,638600	6,024100	4,819300
Error en Porc. (% validacion)		39,375000	33,125000	34,375000	28,750000	33,750000
Para la Clase 1		65,714300	57,142900	48,571400	42,857100	37,142900
Para la Clase 2		32,075500	16,981100	18,867900	16,981100	32,075500
Para la Clase 3		35,135100	40,540500	56,756800	37,837800	43,243200
Para la Clase 4		28,571400	25,714300	20,000000	22,857100	22,857100
MSE (Prueba)		0.0534413	0.0455515	0.0404017	0.0335413	0.0282266
MSE (Validación)		1,087500	1,131300	1,100000	0,906300	0,943700

Tabla 5.10
Resultados para la red con una capa oculta y 3000 iteraciones

Se realizaron experimentos hasta 4000 Iteraciones, con un máximo de 5 capas escondidas con 25 neuronas máximo por capa para un total de 100 experimentos, los resultados de estos experimentos se muestran en los anexos.

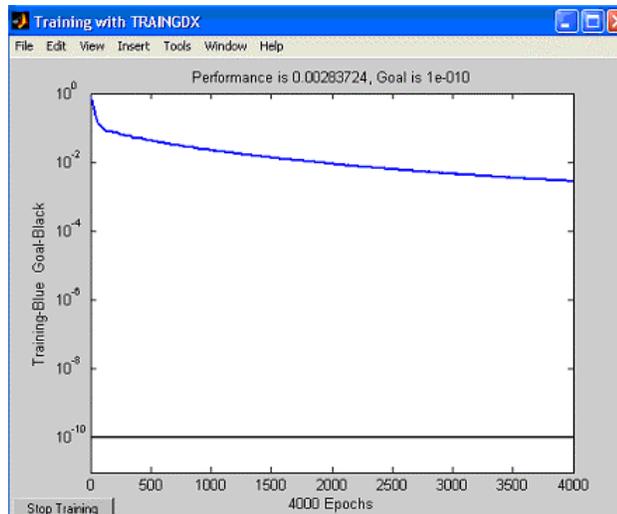


Figura 5.2
Error cuadrático medio vs Iteraciones

La red con mejor desempeño para los datos hepato con solapamiento aleatorio se obtuvo en la simulación para 5 capas ocultas, con 25 neuronas y función de decisión *logsig* en cada neurona con 4000 iteraciones, se obtuvo el 98,9 % de precisión en entrenamiento y 70 % de precisión en validación. En la figura 5.2 se muestra la gráfica del error cuadrático medio vs Iteraciones para un total de 4000 iteraciones. La simulación de esta red para los datos de entrenamiento se muestra en la figura 5.3

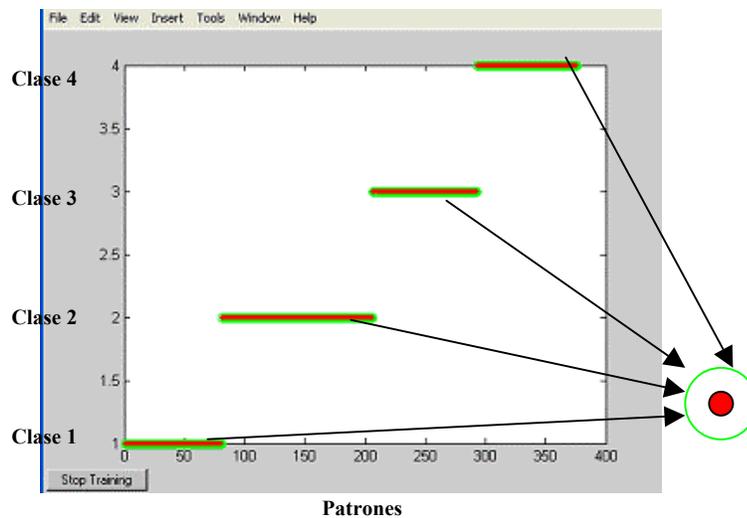


Figura 5.3
Simulación de la mejor red para la data de entrenamiento

Validación de la red para los datos hepato

La validación de la red con mejor desempeño para los datos hepato se realizó con un total de 160 patrones obteniéndose el 70 % de precisión en la tabla 5.11 se muestra los resultados obtenidos. En la figura 5.4 se muestra la gráfica de validación, el punto rojo rodeado de círculo verde significa que el patrón fue bien clasificado, si aparece el círculo rojo solo o el círculo verde solo significa que el patrón fue mal clasificado.

Validacion:	
Error cuadratico medio	1,05660
porc_error =	30,01890
errclas1 =	42,17390
errclas2 =	28,57140
errclas3 =	44,00000
errclas4 =	8,69570

Tabla 5.11
Resultados para la red con mejor desempeño

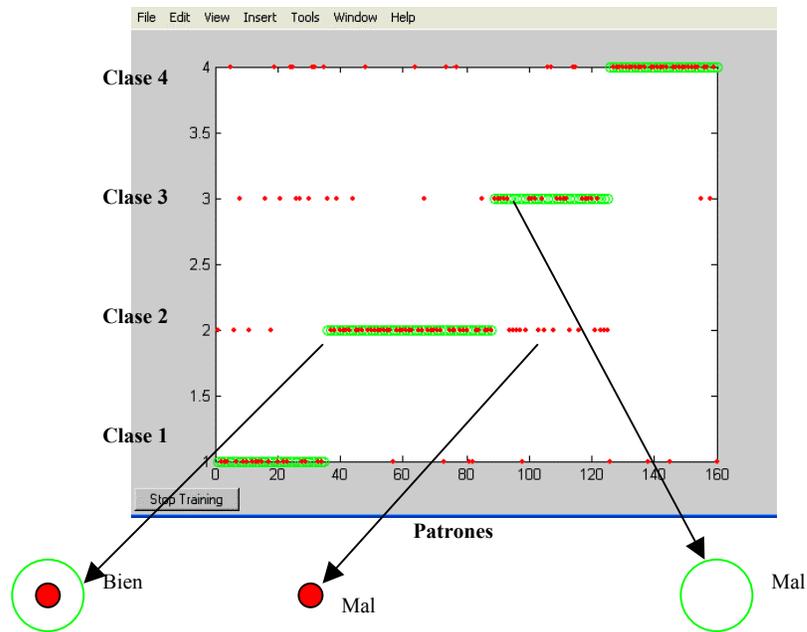


Figura 5.4
Simulación de la mejor red para los datos de validación

En la figura 5.5 se muestra la gráfica para los errores de clasificación cometidos por la red para los datos de validación, los picos que aparecen en la gráfica significa un error cometido por la red, lo ideal de esta gráfica sería una línea recta horizontal.

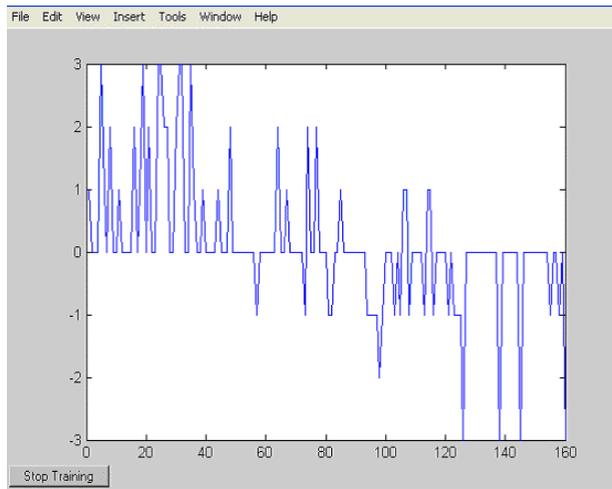


Figura 5.5
Gráfica de los errores cometidos por la red para los datos de validación

Generación de las reglas para los resultados sobresalientes de los datos hepato

La generación de las reglas se hace a través del proceso de justificación de la misma forma como se obtuvieron las reglas para los datos sintéticos, en la tabla 5.12 se muestra un ejemplo para un patrón de entrada a la red cualquiera, de los datos de validación (Ptn.Entd), el camino a seguir (Valores en rojo) desde la capa de entrada (capa 0) hasta la capa de salida (capa 5) representa la justificación.

Ptn. Entd	GGT low	GGT median	GGT high	GPT low	GPT median	GPT high	LDH low	LDH median	LDH high	GGT low	GGT median	GGT high	BUN low	BUN median	BUN high
	0,92182	0,74056	0,31650	0,93470	0,72473	0,28318	0,95176	0,67682	0,26537	0,00000	0,25680	0,87013	0,96401	0,19617	0,06235
Neurons	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CAPA 5	0,91400	0,00020	0,00120	0,00160											
CAPA 4	0,99936	0,00544	0,17301	0,53645	0,57423	0,93667	0,72341	0,99767	0,83844	0,70688	0,61837	0,87246	0,15425	0,83501	0,86609
CAPA 3	0,44300	0,55167	0,87001	0,37528	0,46224	0,88317	0,60718	0,16783	0,72724	0,01078	0,00668	0,89717	0,36251	0,32323	0,49287
CAPA 2	0,81376	0,61105	0,15330	0,37575	0,00127	0,39207	0,00264	0,25496	0,55288	0,30221	0,30539	0,33461	0,99747	0,48908	0,00722
CAPA 1	0,12650	0,38813	0,22634	0,31485	0,48103	0,66866	0,02243	0,33071	0,19177	0,11258	0,14741	0,11821	0,30866	0,37763	0,00570
CAPA 0	0,01157	0,88666	0,01532	0,95562	0,53452	0,09148	0,33110	0,97798	0,62032	0,32097	0,77185	0,53192	0,85812	0,07128	0,87904
MCV low	MCV median	MCV high	MCH low	MCH median	MCH high	TBH low	TBH median	TBH high	RTNN low	RTNN median	RTNN high	ALD	PH	LC	C
0,09230	0,75612	0,89192	0,02764	0,53576	0,37609	0,99871	0,49280	0,28851	0,06454	0,63611	0,89961	0,34590	0,00000	0,00000	0,00000
16	17	18	19	20	21	22	23	24	25	ALD (Alcoholic Live)					
0,98812	0,01762	0,01564	0,01229	0,26291	0,08744	0,12545	0,30058	0,09551	0,03915						
0,33769	0,43534	0,95747	0,00822	0,98872	0,57055	0,99802	0,07884	0,01918	0,33728						
0,13310	0,34334	0,16737	0,30705	0,36486	0,64678	0,37857	0,39928	0,05538	0,39168						
0,87667	0,35602	0,07446	0,07439	0,04295	0,61860	0,63225	0,34051	0,99657	0,81805						
0,82172	0,33463	0,06070	0,00529	0,34973	0,31827	0,99485	0,22148	0,25346							

Tabla 5.12
Justificación de la regla para un patrón cualquiera de los datos de validación

El patrón en este caso pertenece a la clase 1 (ALD Alcoholic Live Damage). La regla para este patrón específico se muestra en la figura 5.6

Regla :

SI GOT ES MUY PARECIDO A low Y GPT ES MUY PARECIDO A low y LDH ES MUY PARECIDO low
Y GGT ES MUY PARECIDO A high Y BUN ES MUY PARECIDO A low y MCV ES MUY PARECIDO A high
Y MCH ES MUY PARECIDO A high y Tbil ES MUY PARECIDO A low Y CRTNN ES MUY PARECIDO A high
ES ALTAMENTE PROBABLE ALD (Alcoholic Live Damage)

Figura 5.6
Regla para el patrón mostrado en la tabla 4.12

Un segundo ejemplo para otro patrón se muestra en la tabla 5.13 en este caso el patrón pertenece a la clase (4) y la regla se muestra en la figura 5.7.

Ptn. Entd	GGT low	GGT median	GGT high	GPT low	GPT median	GPT high	LDH low	LDH median	LDH high	GGT low	GGT median	GGT high	BUN low	BUN median	BUN high	MCV low
Neurona	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CAPA 5	0,00042	0,01210	0,00427	0,29395												
CAPA 4	0,99594	0,00078	0,60834	0,67944	0,28362	0,99471	0,35814	0,97763	0,41866	0,96339	0,17247	0,98138	0,97167	0,55531	0,23844	0,77666
CAPA 3	0,68878	0,50821	0,33354	0,40524	0,43414	0,90845	0,38814	0,32490	0,89414	0,00865	0,76230	0,13584	0,87935	0,50425	0,80757	0,92433
CAPA 2	0,38601	0,97229	0,01862	0,31485	0,00289	0,84466	0,00135	0,51057	0,11978	0,04022	0,98541	0,09101	0,96331	0,65546	0,25820	0,27579
CAPA 1	0,03989	0,99898	0,56255	0,88302	0,98221	0,45154	0,33084	0,96034	0,43184	0,71385	0,02008	0,00475	0,97196	0,66785	0,00415	0,99590
CAPA 0	0,01857	0,99999	0,02538	0,90011	0,00881	0,02003	0,78028	0,98153	0,62810	0,80173	0,02641	0,82958	0,09918	0,20467	0,82470	0,97063
MCH low	MCH median	MCH high	MCH low	MCH median	MCH high	Tbil low	Tbil median	Tbil high	CRTNN low	CRTNN median	CRTNN high	ALD	PH	LC	C	
0,99827	0,35915	0,00685	0,72504	0,94700	0,27086	0,00000	0,10208	0,80122	0,06454	0,63611	0,83361	0,00000	0,00000	0,00000	0,29010	
16	17	18	19	20	21	22	23	24	25							
0,77666	0,08415	0,02236	0,00063	0,11689	0,08632	0,96407	0,08450	0,92935	0,56383							
0,92433	0,30524	0,34872	0,13025	0,97338	0,14378	0,90073	0,08076	0,00085	0,88718							
0,27579	0,39812	0,42211	0,92877	0,31620	0,78851	0,98607	0,99929	0,30909	0,99888							
0,99590	0,07360	0,01242	0,22397	0,07897	0,06375	0,65770	0,88642	0,99457	0,97160							
0,97063	0,71387	0,17749	0,00053	0,05480	0,97059	0,42417	0,99348	0,03640	0,00064							

Tabla 5.13
Justificación de la regla para un segundo patrón cualquiera de los datos de validación

La regla para el patrón mostrado en la tabla 5.13 se muestra en la figura 5.7

SI GOT ES MUY PARECIDO A high Y GPT ES MUY PARECIDO A high y LDH ES MUY PARECIDO A medium
Y GGT ES MUY PARECIDO A high Y BUN ES MUY PARECIDO A low y MCV ES MUY PARECIDO A low C (Cholelithiasis)
Y MCH ES MUY PARECIDO A medium y Tbil ES MUY PARECIDO A high Y CRTNN ES MUY PARECIDO A high
ES POCO PROBABLE C (Cholelithiasis)

Figura 5.7
Regla para el patrón mostrado en la tabla 4.13

Comparación de los resultados de esta investigación y otros trabajos

Los experimentos sobre los datos sintéticos donde el solapamiento entre clase fue progresivo y porcentualmente creciente se considera pionero, todos los trabajos que se consultaron se han realizado sobre datos con solapamiento aleatorio, por lo tanto los resultados no se pudieron compara con trabajos anteriores. Para el caso de los datos hepato, en el trabajo [32] se presenta los resultados obtenidos utilizando diferentes técnicas para clasificación de patrones. La comparación de las diferentes técnicas en función de la precisión en los datos de entrenamiento y la precisión en los datos de validación se muestran en la tabla 5.14. Los resultados obtenidos por ésta investigación se encuentran resaltados en azul fuerte en la última fila de la tabla, trabajos similares pero utilizando técnicas diferentes para calcular el error cuadrático medio se encuentra resaltados en azul suave.

Metodo	Precision Data entrenamiento	Precision Data validación	Referencia
IB2-IB4	81.2 - 85.5	43.6 - 44.6	WEKA
Naive Bayes	-----	46.6	WEKA,
1R (rules)	58.4	50.3	WEKA,
T2 (rules from decision tree)	67.5	53.3	WEKA,
FOIL (inductive logic)	99	60.1	WEKA,
FSM, initial 49 crisp logical rules	83.5	63.2	FSM
LDA (statistical)	68.4	65.0	WEKA,
DLVQ (38 nodes)	100	66.0	WEKA,
C4.5 decision rules	64.5	66.3	WEKA,
Mejor modelo fuzzy MLP ESM (0.8 0.6 0.4 0.2)	75.5	66.3	S. Mitra
MLP with RPROP		68.0	WEKA,
Cascade Correlation		71.0	WEKA,
Fuzzy neural network ANFIS	100	75.5	Hayashi
C4.5 decision tree	94.4	75.5	WEKA,
FSM, Gaussian functions	93	75.6	WEKA,
FSM, 60 triangular functions	93	75.8	WEKA,
IB1c (instance-based)		76.7	WEKA,
kNN, k=1, Camberra, raw	76.1	80.4	WD/SBL
K* method		78.5	WEKA,
1-NN, 4 features removed, Manhattan	76.9	80.4	WEKA, KG
1-NN, Camberra, raw, removed 2, 6, 8, 9	77,2	83.4	WEKA, KG
FMLP con ESM y Momentun Dinamico	98,9	70	Propuesta actual

Tabla 5.14
Resultados obtenidos por diferentes métodos para la clasificación de los datos hepato

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

En cuanto a la experimentación con los datos sintéticos se puede concluir lo siguiente:

. Los resultados obtenidos al realizar los experimentos con los datos sintéticos en forma circular y los resultados obtenidos con la data sintética en forma cuadrada son muy similares, lo que indica que la forma geométrica en que se encuentran agrupados los datos y la forma geométrica del solapamiento entre los datos no son relevantes para los resultados del comportamiento del sistema neurodifuso FMLP.

. El porcentaje de aciertos de las mejores redes obtenidas con los datos sintéticos disminuyen notablemente a partir del 50 % de solapamiento entre clases. Para solapamientos entre 10 % y 40 % se alcanza un buen desempeño de la red con relativa facilidad.

. El solapamiento entre clases no afecta el entrenamiento de la red, esta se comporta en forma homogénea.

En cuanto a la experimentación con los datos reales (hepato) se puede concluir lo siguiente:

. Se logro obtener un alto desempeño en la mejor red obtenida a través del entrenamiento (98,9 % entrenamiento, 70 % validación) en comparación con los resultados de autores anteriores, esto ocurre por utilizar el concepto de momento en el algoritmo de entrenamiento de la red .

. El proceso de difusificación aumenta el espacio dimensional de clasificación de los patrones, esto hace que el método utilizado para difusificar el patrón de entrada sea interesante desde el punto de vista matemático como caso de estudio.

. El sistema difuso neuronal (FMLP) propuesto entrenado apropiadamente tiende a dar respuestas razonables ante vectores de entrada que no fueron utilizados en el entrenamiento, aproximando sus propiedades a los vectores de entrada similares a los empleados en el entrenamiento, para producir una salida correcta. Esta generalización hace posible entrenar una red con un conjunto representativo de pares de entrada / salida y obtener un resultado correcto sin necesidad de entrenar la red con todos los pares de valores posibles de entrada salida. El sistema propuesto cumple a cabalidad con esta propiedad.

. El prototipo propuesto de sistema difuso neuronal para el reconocimiento de patrones con solapamiento aleatorio entre clases aplicado a enfermedades hepáticas ha sido plenamente probado a través de los experimentos realizados.

. Éste trabajo propone un algoritmo generalizado para un sistema difuso neuronal aplicado a la clasificación de enfermedades representadas por patrones solapados mediante el cual un médico puede obtener en tiempo real, un pronóstico del diagnóstico de una enfermedad hepática.

Recomendaciones

El despistaje de enfermedades objetos de estudio en ésta investigación mediante técnicas de reconocimientos de patrones a través de sistemas neurodifusos posee diversos campos de acción aparte del tipo hepático: El despistaje de cáncer en el estómago, la radiología en estudios oncológicos mediante el suministro de radiactivos o la biopsia propiamente dicha, en el diagnóstico de enfermedades renales que forman parte de otras áreas posibles, donde se pueden aplicar las técnicas acá expuestas.

Es parte importante de la ingeniería contribuir con la calidad de vida del ser humano y es por ello que se recomienda la implementación de técnicas de sistemas neurodifusos en la solución de problemas como los citados anteriormente, en función de proveer una mayor y mejor calidad de vida .

Una base de datos mayor, puede mejorar considerablemente los resultados obtenidos, en función de entrenar el sistema con un número mayor de muestras.

Otras técnicas como la de soporte de maquinas vectoriales, podrían ser implementadas a fin de evaluar su eficiencia en el reconocimiento de patrones.

La implementación de los algoritmos para la generación de las reglas difusas puede hacer el sistema más comprensible al usuario; estos algoritmos son el NeuroRules y

NeuroLinear. El primero se aplica para discretizar los conjuntos de datos y generar reglas de clasificación simbólicas. El segundo por su parte, genera funciones por tramos (Piece-wise) lineales discriminantes para estos conjuntos de datos.

. La utilización de las rutinas en Matlab para la simulación de redes neurales simplifica notablemente la implementación de los sistemas neurodifusos.

Referencias Bibliograficas

1. (Aguilar J. Rivas F. 2001). Computación Inteligente. Universidad de los Andes, Mérida, Venezuela junio 2001.
2. (Liquin Ma, Camping Chen, Ziping Tian, Wanqing He). Monitoring the particle in CFB Using Fuzzy Neural Network . IEEE Engineering in medicine and biology” . " Neural Networks, Vol. 7, No. 2, 1999.
3. (Sankar Pal). "Multilayer perceptron, fuzzy sets, and classification," IEEE Transactions on Neural Networks, Vol. 3, No. 5, September 1992”.
4. (Shigeo Abe 2001). Pattern classification. Editorial Springer – Verlag London Berlin, Great Britain Edición 2001.
5. (Ludmila I. Kuncheva.). Fuzzy Classifier Design. Editorial Physica-Verlay A springer-Verlay Heidelberg New Yor Company Edición 2000.
6. (Christopher M. Bishop). Neural Networks for Pattern Recognition. Editorial Oxford University Press. Edición 2000. 1ª Publicación 1995.
- 7.(Sankar K. Pal, Sushmita Mitra). Neuro-Fuzzy Pattern Methods in soft Computing, Indian Statistical Institute Calcuta.. Editorial Wiley and Sons,INC. Edición 1999.
8. (John Yen; Reza Langari). Fuzzy Logic Intelligence, control Información. Prentice Hall, New Jersey . Edición 1999.

9. (Nikhil R. Pal.). Pattern Recognition in Soft Computing Paradigm. Editor Takeshi Yamakawa . Published: Vol 1. 2001.
10. (Yon Ham Pao). Adaptive Pattern Recognition and Neural Network. Editorial Addison-Wesley publishing Company INC 1989..
11. (Ernest Czogala, Jaccklesky). Fuzzy and Neuro-Fuzzy Intelligent Systems. Editorial: Physica-verlag Heidelberl. Edición 2000.
- 12 (Bonifacio Martin del Brío; Alfredo Sanz Molina). Redes Neuronales y Sistemas Borrosos. Universidad de Zaragoza. Editorial *ra-ma* Edición 2001
- 13 (Haykin, S). Neural Networks, a Comprehensive Foundation. Edición 2000.
- 14 (Baum, E, B, Haussler, D). What size net gives valid generalization. Neural Computation, 1, 151-160, 1989.
- 15 (Bishop, C.M). Neural Networks and their applications. Rev.Sci.Instrum., 65, 6, pp.1803-1832,1994.
- 16 (Gil, Alvaro). “Redes Neuronales para la estimación de flujo multifásico”. Universidad Nacional Experimental Politécnica (UNEXPO) (1997).

17 (Sarmiento, Jesús). “Redes Neuronales para la identificación de un sistema de reacción continuo ”. Universidad Nacional Experimental Politécnica (UNEXPO) (2000).

18 (Volcanes, Rafael). “Diseño y construcción de un sistema de reconocimiento de patrones en imágenes digitales”. Universidad Nacional Experimental Politécnica (UNEXPO) (2001).

19 (Ocando, Atencio). “Sistema para el reconocimiento de patrones, basado en redes neuronales, con aprendizaje supervisado, utilizando el algoritmo de retropropagación”. Universidad del Zulia (LUZ) 1992.

20 (Yoshihiro Kobayashi and Kostas Terzidis). “Extracting the Geometry of Buildings from Satellite Images Using Fuzzy Multiple Layer Perceptrons” The Department of Architecture and Urban Design .School of Art and Architecture University of California, Los Angeles 2000.

21 (Sankar K. Pal, Sushmita Mitra). Multilayer Perceptron, Fuzzy Sets, and Classification IEEE Transactions on neural networks, Vol 3, N° 5 Septiembre 1992 .

22 (Sankar K. Pal and Sushmita Mitra). “Neuro-Fuzzy Pattern Recognition Methods in Soft Computing. Wiley-Interscience John WILE & Sons .Publication New York 1999”.

23 (Sankar K. Pal and Sushmita Mitra). "Multilayer Perceptrons, Fuzzy Sets, and Classification" IEEE Transactions on neural networks, Vol 6 Enero 1995.

24 (Jyh-Shing Roger Jang). "ANFIS: Adaptive-Network-Based Fuzzy Inference System " IEEE Transactions on Systems, Man, and Cybernetics, Systems", volumen 23, páginas 665--684, (1993).

25 <http://www.isical.ac.in/~sushmita/patterns/kzar.dat>

26 (Somying Promcharoen and Yuttapong Ranganseri). "Supervised Classification of Multispectral Satellite Images using Fuzzy logic and Neural Network Department of Telecommunications Engineering, Faculty of Engineering King Mongkut's Institute of Technology Ladkrabang, Bangkok . Thailand . 2000.

27 (Lippman, R.P). An introduction to computing with neural nets IEEE ASSP Magazine, April, (4-22) 1987.

28 (Simpson , P.K). Foundations of Neural Networks IEEE press New York 1992.

29 (Nauck, Detlef). "Perspective and Directions", in Technical University of Branschweig, Germany 1995.

30 (Sankar K. Pal, Sushmita Mitra). Multilayer Perceptron, Fuzzy Sets, and Classification , IEEE Transactions on neural networks, Vol 3, N° 5 Septiembre 1994 .

31 (Y. Hayashi). Neural expert system using fuzzy teaching input and its application to medical diagnosis , Information sciences applications, vol 1 pp 47-58 1994.

32 (W. Duch, R. Adarnczak y K Grabczewski). A new methodology of extraction, optimization and application of crisp and fuzzy logical rules . IEE trans. Neural Network, vol 12, PP 277 – 306 Marzo 2001.

33 (C. Ramirez). Fuzzy Neural Networks for Classification Problems with Uncertain Data Input . Tesis Doctoral Universidad de Surry. Guildford, Surrey, Inglaterra, Reino Unido 1996.

34 (Hayashi, A. Imura, K. Yoshida). “Fuzzy neural expert system and its application to medical diagnosis”, in: 8th International Congress on Cybernetics and Systems, New York City 1990, pp. 54-61.

35 (Ovidiu Grigore, Inge Gavtat) “Neuro-fuzzy Models for Speech Pattern Recognition in Romanian Language”. Polytechnic University of Bucharest, Dept. Electronic and Telecommunications Str. Rosia Montana nr.1, bl. 17, sc. 1, ap. 5, sect. 6, 77584 Bucharest, Romania 1999.

36 (Sánchez, José). “Sistema de reconocimiento de formas y patrones en una imagen digital a color por medio de redes Neuronales”. Universidad Centro Occidental “Lisandro Alvarado” (UCLA) (2000).

ANEXOS

Resultados de los experimentos para datos circulares.

Red [6 5 2] 10 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	211978	0,1333	13,3333	0,1504	15,0435	1000	0	0	0,2	20	0,1304	13,0435
3000	0	0	0,1333	13,3333	0,1304	13,0435	3000	0	0	0,2667	26,6667	0,1739	17,3913
5000	0	0	0,2	20	0,1304	13,0435	5000	0	0	0,2	20	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,1739	17,3913	10000	0	0	0,0667	6,6667	0,1304	13,0435
Red [6 10 2] 10 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,011	1,0989	0,1333	13,3333	0,1522	15,2173	1000	0,022	2,1978	0,2667	26,6667	0,2609	26,087
3000	0	0	0,1333	13,3333	0,1304	13,0435	3000	0	0	0,2	20	0,1956	19,5652
5000	0	0	0,2667	26,6667	0,1522	15,2174	5000	0	0	0,2	20	0,2391	12,0879
10000	0	0	0,2667	26,6667	0,1522	15,2174	10000	0	0	0,2667	26,6667	0,2826	14,2857
Red [6 15 2] 10 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435	1000	0,011	1,0989	0,2	20	0,1739	17,3891
3000	0	0	0,2	20	0,1304	13,0435	3000	0,033	3,2467	0,3333	33,3333	0,2391	12,0879
5000	0	0	0,1333	13,3333	0,1522	15,2174	5000	0	0	0,2	20	0,1522	15,2174
10000	0	0	0,2	20	0,1957	19,5652	10000	0	0	0,2	20	0,1304	13,0435

Red [6 5 5 5 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0	0	0,1333	13,3333	0,1304	13,0435
3000	0	0	0,2	20	0,1957	19,5652
5000	0	0	0,3333	33,3333	0,1957	19,5652
10000	0	0	0,2	20	0,1522	15,2174
Red [6 10 10 10 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,667	6,6667	0,1087	0,8696
3000	0	0	0,2667	26,6667	0,1087	0,8696
5000	0	0	0,1333	13,3333	0,087	8,6957
10000	0	0	0,1333	13,3333	0,1739	17,3913
Red [6 15 15 15 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0	0	0,2	20	0,1304	13,0435
3000	0	0	0,2667	26,6667	0,1522	15,2174
5000	0	0	0,2667	26,6667	0,1957	19,5652
10000	0	0	0,2	20	0,1956	19,5652

Resultados de los experimentos para datos circulares.

Red [6 5 2] 20 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,044	4,3956	0,2	20	0,1522	15,2174	1000	0,033	3,2967	0,2	20	0,1957	19,3652
3000	0,033	3,2967	0,2	20	0,1522	15,2174	3000	0	0	0,2	20	0,1522	15,2174
5000	0,022	2,1978	0,1333	13,3333	0,1739	17,3913	5000	0	0	0,2	20	0,2391	23,913
10000	0	0	0,1333	13,3333	0,1522	15,2174	10000	0,011	1,0989	0,2	20	0,1957	19,5652
Red [6 10 2] 20 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1522	15,2174	1000	0,0539	5,4945	0,1333	13,3333	0,1957	19,5632
3000	0,033	3,2967	0,2667	26,6667	0,1739	17,3913	3000	0	0	0,2	20	0,1957	19,5652
5000	0	0	0,2667	26,6667	0,1522	15,2174	5000	0	0	0,2	20	0,1739	17,3913
10000	0	0	0,2	20	0,1522	15,2174	10000	0	0	0,2	20	0,1522	15,2174
Red [6 15 2] 20 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
3000	0,04	4,3956	0,1333	13,333	0,152	15,217	3000	0,033	3,2967	0,0667	6,6667	0,1522	15,2174
5000	0	0	0,2667	2,6667	0,1304	13,0435	5000	0,022	2,1978	0,2	20	0,1957	19,5652
10000	0,04	1,0989	0,2	20	0,1957	19,565	10000	0	0	0,0667	6,6667	0,1522	15,2174

Red [6 5 5 5 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,044	4,3956	0,0667	6,6667	0,1739	17,3913
3000	0,033	3,2976	0,2	20	0,1522	15,2174
5000	0,011	1,0989	0,1333	13,3333	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,1739	17,3913
Red [6 10 10 10 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1739	17,3913
3000	0,022	2,1978	0,2667	26,6667	0,1522	15,2174
5000	0	0	0,2	20	0,1087	10,8676
10000	0	0	0,2	20	0,1522	15,217
Red [6 15 15 15 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,044	4,3956	0,0667	6,6667	0,1739	17,3913
3000	0,022	2,1978	0,2	20	0,2174	21,7391
5000	0	0	0,2	20	0,1739	17,3913
10000	0	0	0,1333	13,3333	0,2604	26,087

Resultados de los experimentos para datos circulares.

Red [6 5 2] 30 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,2667	26,6667	0,087	8,6957	1000	0,022	2,1978	0,4	40	0,2391	23,913
3000	0,011	1,0989	0,2667	26,6667	0,1957	19,5652	3000	0,011	1,0987	0,4	40	0,1956	19,5652
5000	0,011	1,0989	0,2667	26,6667	0,3043	30,4348	5000	0	0	0,5333	53,333	0,1739	17,3913
10000	0,011	1,0989	0,4	40	0,1522	15,2174	10000	0	0	0,4667	46,6667	0,1957	19,5653
Red [6 10 2] 30 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,05492	5,4955	0,4	40	0,1739	17,3913	1000	0,0333	3,2967	0,3333	33,3333	0,13	13,0435
3000	0,033	3,2667	0,3333	33,3333	0,2174	21,7391	3000	0	0	0,5333	53,3333	0,2609	26,087
5000	0,011	1,0989	0,4	40	0,2174	21,7391	5000	0	0	0,4	40	0,2391	23,913
10000	0	0	0,3333	33,3333	0,1739	17,3913	10000	0	0	0,4	40	0,1739	17,3913
Red [6 15 2] 30 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,4	40	0,1739	17,3913	1000	0,022	2,1978	0,2667	26,6667	0,1087	10,8696
3000	0	0	0,4	40	0,1739	17,3913	3000	0	0	0,4667	46,6666	0,2391	23,913
5000	0,022	2,1978	0,3333	53,3333	0,1304	13,0435	5000	0	0	0,4	40	0,1739	17,3913
10000	0	0	0,4667	46,6667	0,1739	17,3913	10000	0	0	0,4	40	0,1087	10,8676

Red [6 5 5 5 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,3333	33,3333	0,1304	13,0435
3000	0,011	1,0984	0,5333	53,3333	0,1304	13,0435
5000	0	0	0,3333	33,3333	0,0152	15,274
10000	0	0	0,4667	46,6667	0,1739	17,3913
Red [6 10 10 10 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,3333	33,3333	0,1304	13,0435
3000	0,022	2,1978	0,4667	46,6667	0,2174	21,7391
5000	0	0	0,4667	46,6667	0,2174	26,7391
10000	0	0	0,3333	33,3333	0,2607	26,087
Red [6 15 15 15 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		0,2607		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,011	1,0989	0,2667	26,6667	0,2826	28,2609
3000	0	0	0,3333	33,3333	0,1733	17,3913
5000	0	0	0,4	40	0,1739	17,3913
10000	0	0	0,5333	53,3333	0,2391	23,913

Resultados de los experimentos para datos circulares.

Red [6 5 2] 50 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,0667	6,6667	0,1522	15,2174	1000	0,011	1,0989	0,2	20	0,1087	10,8696
3000	0,044	4,3956	0,2	20	0,2391	23,913	3000	0,011	1,0989	0,2667	26,6667	0,1956	19,5652
5000	0	0	0,2667	26,6667	0,1957	19,5652	5000	0	0	0,2	20	0,1739	17,3913
10000	0	0	0,2	20	0,2391	23,913	10000	0,022	2,19	0,1333	13,3333	0,1304	13,0435
Red [6 10 2] 50 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,29	0,2	20	0,1957	19,5652	1000	0,0549	5,4945	0,2	20	0,1957	19,5652
3000	0,033	3,2967	0,1333	13,3333	0,1739	17,3913	3000	0	0	0,2	20	0,1522	15,2174
5000	0,022	2,1978	0,1333	13,3333	0,1522	15,2174	5000	0	0	0,1333	13,3333	0,1304	13,0435
10000	0	0	0,2	20	0,1739	17,3913	10000	0	0	0,1333	13,3333	0,1087	10,8696
Red [6 15 2] 50 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1522	15,2174	1000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435
3000	0,022	2,1978	0,2	20	0,1957	19,5652	3000	0,011	1,0989	0,2	20	0,2391	23,913
5000	0,011	1,0989	0,1333	13,3333	0,2174	21,7391	5000	0	0	0,2667	26,6667	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,1304	13,0435	10000	0	0	0,1333	13,3333	0,2826	28,2609

Red [6 5 5 5 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,2826	28,2609
3000	0,02	2,1973	0,2667	26,6667	0,1521	15,2174
5000	0	0	0,2	20	0,1957	19,5652
10000	0	0	0,2	20	0,1739	17,3913
Red [6 10 10 10 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1739	17,3913
3000	0,011	1,0989	0,2667	26,6667	0,1957	19,5652
5000	0,011	1,0989	0,1333	13,3333	0,1739	17,3913
10000	0	0	0,1333	13,3333	0,1739	17,3913
Red [6 15 15 15 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		0,2607		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,2	20	0,2609	26,087
3000	0,033	3,2967	0,2	20	0,1739	17,3913
5000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435
10000	0	0	0,2	20	0,1304	13,0435

Resultados de los experimentos para datos cuadrados.

Red [6 5 2] 50 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,0667	6,6667	0,1522	15,2174	1000	0,011	1,0989	0,2	20	0,1087	10,8696
3000	0,044	4,3956	0,2	20	0,2391	23,913	3000	0,011	1,0989	0,2667	26,6667	0,1956	19,5652
5000	0	0	0,2667	26,6667	0,1957	19,5652	5000	0	0	0,2	20	0,1739	17,3913
10000	0	0	0,2	20	0,2391	23,913	10000	0,022	2,19	0,1333	13,3333	0,1304	13,0435
Red [6 10 2] 50 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,29	0,2	20	0,1957	19,5652	1000	0,0549	5,4945	0,2	20	0,1957	19,5652
3000	0,033	3,2967	0,1333	13,3333	0,1739	17,3913	3000	0	0	0,2	20	0,1522	15,2174
5000	0,022	2,1978	0,1333	13,3333	0,1522	15,2174	5000	0	0	0,1333	13,3333	0,1304	13,0435
10000	0	0	0,2	20	0,1739	17,3913	10000	0	0	0,1333	13,3333	0,1087	10,8696
Red [6 15 2] 50 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1522	15,2174	1000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435
3000	0,022	2,1978	0,2	20	0,1957	19,5652	3000	0,011	1,0989	0,2	20	0,2391	23,913
5000	0,011	1,0989	0,1333	13,3333	0,2174	21,7391	5000	0	0	0,2667	26,6667	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,1304	13,0435	10000	0	0	0,1333	13,3333	0,2826	28,2609

Red [6 5 5 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,2826	28,2609
3000	0,02	2,1973	0,2667	26,6667	0,1521	15,2174
5000	0	0	0,2	20	0,1957	19,5652
10000	0	0	0,2	20	0,1739	17,3913
Red [6 10 10 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1739	17,3913
3000	0,011	1,0989	0,2667	26,6667	0,1957	19,5652
5000	0,011	1,0989	0,1333	13,3333	0,1739	17,3913
10000	0	0	0,1333	13,3333	0,1739	17,3913
Red [6 15 15 2] 50 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		0,2607		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,2	20	0,2609	26,087
3000	0,033	3,2967	0,2	20	0,1739	17,3913
5000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435
10000	0	0	0,2	20	0,1304	13,0435

Resultados de los experimentos para datos cuadrados.

Red [6 5 2] 20 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,044	4,3956	0,2	20	0,1522	15,2174	1000	0,033	3,2967	0,2	20	0,1957	19,3652
3000	0,033	3,2967	0,2	20	0,1522	15,2174	3000	0	0	0,2	20	0,1522	15,2174
5000	0,022	2,1978	0,1333	13,3333	0,1739	17,3913	5000	0	0	0,2	20	0,2391	23,913
10000	0	0	0,1333	13,3333	0,1522	15,2174	10000	0,011	1,0989	0,2	20	0,1957	19,5652
Red [6 10 2] 20 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1522	15,2174	1000	0,0539	5,4945	0,1333	13,3333	0,1957	19,5632
3000	0,033	3,2967	0,2667	26,6667	0,1739	17,3913	3000	0	0	0,2	20	0,1957	19,5652
5000	0	0	0,2667	26,6667	0,1522	15,2174	5000	0	0	0,2	20	0,1739	17,3913
10000	0	0	0,2	20	0,1522	15,2174	10000	0	0	0,2	20	0,1087	10,8696
Red [6 15 2] 20 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,04	4,3956	0,1333	13,333	0,152	15,217	1000	0,033	3,2967	0,0667	6,6667	0,1522	15,2174
3000	0	0	0,2667	2,6667	0,1304	13,0435	3000	0,022	2,1978	0,2	20	0,1957	19,5652
5000	0,04	1,0989	0,2	20	0,1957	19,565	5000	0	0	0,0667	6,6667	0,1522	15,2174
10000	0	0	0,2	20	0,152	15,21	10000	0	0	0,2	20	0,1087	10,8696

Red [6 5 5 5 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,044	4,3956	0,0667	6,6667	0,1739	17,3913
3000	0,033	3,2976	0,2	20	0,1522	15,2174
5000	0,011	1,0989	0,1333	13,3333	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,1739	17,3913
Red [6 10 10 10 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,087	8,6957
3000	0,022	2,1978	0,2667	26,6667	0,1522	15,2174
5000	0	0	0,2	20	0,1957	19,5652
10000	0	0	0,2	20	0,1522	15,217
Red [6 15 15 15 2] 20 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,044	4,3956	0,0667	6,6667	0,1739	17,3913
3000	0,022	2,1978	0,2	20	0,2174	21,7391
5000	0	0	0,2	20	0,087	8,6957
10000	0	0	0,1333	13,3333	0,2604	26,087

Resultados de los experimentos para datos cuadrados.

Red [6 5 2] 30 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,2667	26,6667	0,1739	17,3913	1000	0,022	2,1978	0,4	40	0,2391	23,913
3000	0,011	1,0989	0,2667	26,6667	0,1957	19,5652	3000	0,011	1,0987	0,4	40	0,1956	19,5652
5000	0,011	1,0989	0,2667	26,6667	0,3043	30,4348	5000	0	0	0,5333	53,333	0,1739	17,3913
10000	0,011	1,0989	0,4	40	0,1522	15,2174	10000	0	0	0,4667	46,6667	0,1957	19,5653
Red [6 10 2] 30 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,05492	5,4955	0,4	40	0,1739	17,3913	1000	0,0333	3,2967	0,3333	33,3333	0,13	13,0435
3000	0,033	3,2667	0,3333	33,3333	0,2174	21,7391	3000	0	0	0,5333	53,3333	0,2609	26,087
5000	0,011	1,0989	0,4	40	0,2174	21,7391	5000	0	0	0,4	40	0,2391	23,913
10000	0	0	0,3333	33,3333	0,1739	17,3913	10000	0	0	0,4	40	0,1739	17,3913
Red [6 15 2] 30 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,4	40	0,1739	17,3913	1000	0,022	2,1978	0,2667	26,6667	0,1087	10,8696
3000	0	0	0,4	40	0,1739	17,3913	3000	0	0	0,4667	46,6666	0,2391	23,913
5000	0,022	2,1978	0,3333	53,3333	0,1304	13,0435	5000	0	0	0,4	40	0,1739	17,3913
10000	0	0	0,4667	46,6667	0,1739	17,3913	10000	0	0	0,4	40	0,1087	10,8696

Red [6 5 5 5 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,3333	33,3333	0,1304	13,0435
3000	0,011	1,0984	0,5333	53,3333	0,1304	13,0435
5000	0	0	0,3333	33,3333	0,0152	15,274
10000	0	0	0,4667	46,6667	0,1739	17,3913
Red [6 10 10 10 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,3333	33,3333	0,1304	13,0435
3000	0,022	2,1978	0,4667	46,6667	0,2174	21,7391
5000	0	0	0,4667	46,6667	0,2174	26,7391
10000	0	0	0,3333	33,3333	0,2607	26,087
Red [6 15 15 15 2] 30 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		0,2607		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,011	1,0989	0,2667	26,6667	0,2826	28,2609
3000	0	0	0,3333	33,3333	0,1733	17,3913
5000	0	0	0,4	40	0,1739	17,3913
10000	0	0	0,5333	53,3333	0,2391	23,913

Resultados de los experimentos para datos cuadrados.

Red [6 5 2] 40 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,0667	6,6667	0,1522	15,2174	1000	0,011	1,0989	0,2	20	0,1087	10,8696
3000	0,044	4,3956	0,2	20	0,2391	23,913	3000	0,011	1,0989	0,2667	26,6667	0,1956	19,5652
5000	0	0	0,2667	26,6667	0,1957	19,5652	5000	0	0	0,2	20	0,1739	17,3913
10000	0	0	0,2	20	0,2391	23,913	10000	0,022	2,19	0,1333	13,3333	0,1304	13,0435
Red [6 10 2] 40 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,29	0,2	20	0,1087	10,8696	1000	0,0549	5,4945	0,2	20	0,1957	19,5652
3000	0,033	3,2967	0,1333	13,3333	0,1522	15,2174	3000	0	0	0,2	20	0,1522	15,2174
5000	0,022	2,1978	0,1333	13,3333	0,2174	21,7391	5000	0,011	1,0987	0,26	26,6667	0,1304	13,0435
10000	0	0	0,2	20	0,1739	17,3913	10000	0	0	0,1333	13,3333	0,1957	19,5652
Red [6 15 2] 40 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1522	15,2174	1000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435
3000	0,022	2,1978	0,2	20	0,1957	19,5652	3000	0	0	0,2	20	0,2391	23,913
5000	0,011	1,0989	0,1333	13,3333	0,2174	21,7391	5000	0	0	0,2667	26,6667	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,1304	13,0435	10000	0	0	0,1333	13,3333	0,2826	28,2609

Red [6 5 5 5 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,011	1,0989	0,2	20	0,2826	28,2609
3000	0,02	2,1973	0,2667	26,6667	0,1521	15,2174
5000	0	0	0,2	20	0,1957	19,5652
10000	0	0	0,2	20	0,1739	17,3913
Red [6 10 10 10 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,2	20	0,1739	17,3913
3000	0,011	1,0989	0,2667	26,6667	0,1957	19,5652
5000	0,022	2,1978	0,1333	13,3333	0,1739	17,3913
10000	0	0	0,1333	13,3333	0,1304	13,0435
Red [6 15 15 15 2] 40 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		0,2607		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,033	3,2967	0,2	20	0,1304	13,0435
3000	0	0	0,2	20	0,1739	17,3913
5000	0,022	2,1978	0,1333	13,3333	0,2609	26,087
10000	0	0	0,2	20	0,2174	21,7391

Resultados de los experimentos para datos cuadrados.

Red [6 5 2] 10 % solapamiento (91 E , 15 P , 46 V)							Red [6 5 5 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0	0	0,1333	13,3333	0,1504	15,0435	1000	0	0	0,2	20	0,1304	13,0435
3000	0,011	1,0989	0,1333	13,3333	0,1304	13,0435	3000	0	0	0,2667	26,6667	0,1739	17,3913
5000	0	0	0,2	20	0,1304	13,0435	5000	0	0	0,2	20	0,1522	15,2174
10000	0	0	0,1333	13,3333	0,1739	17,3913	10000	0	0	0,0667	6,6667	0,0652	6,5617
Red [6 10 2] 10 % solapamiento (91 E , 15 P , 46 V)							Red [6 10 10 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0	0	0,1333	13,3333	0,1522	15,2174	1000	0,022	2,1978	0,2667	26,6667	0,1957	19,5652
3000	0,0220	21,978	0,1333	13,3333	0,1304	13,0435	3000	0	0	0,2	20	0,2174	21,74
5000	0	0	0,2667	26,6667	0,1522	15,2174	5000	0	0	0,2	20	0,2391	23,91
10000	0	0	0,2667	26,6667	0,1522	15,2174	10000	0	0	0,2667	26,6667	0,2826	28,26
Red [6 15 2] 10 % solapamiento (91 E , 15 P , 46 V)							Red [6 15 15 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación		Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error		MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,1333	13,3333	0,1304	13,0435	1000	0,011	1,0989	0,2	20	0,1739	17,3912
3000	0	0	0,2	20	0,1304	13,0435	3000	0,033	3,2467	0,3333	33,3333	0,2391	23,91
5000	0	0	0,1333	13,3333	0,1522	15,2174	5000	0	0	0,2	20	0,1739	17,3912
10000	0	0	0,2	20	0,1957	19,5652	10000	0	0	0,2	20	0,1304	13,0435

Red [6 5 5 5 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0	0	0,1333	13,3333	0,1087	10,8696
3000	0	0	0,2	20	0,1957	19,5652
5000	0	0	0,3333	33,3333	0,1957	19,5652
10000	0	0	0,2	20	0,1522	15,2174
Red [6 10 10 10 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0,022	2,1978	0,667	6,6667	0,1087	10,8696
3000	0	0	0,2667	26,6667	0,1087	10,8696
5000	0	0	0,1333	13,3333	0,087	8,6957
10000	0	0	0,1333	13,3333	0,1739	17,3913
Red [6 15 15 15 2] 10 % solapamiento (91 E , 15 P , 46 V)						
Iteraciones	Entrenamiento		Prueba		Validación	
	MSE	% Error	MSE	% Error	MSE	% Error
1000	0	0	0,2	20	0,1304	13,0435
3000	0	0	0,2667	26,6667	0,1522	15,2174
5000	0	0	0,2667	26,6667	0,1957	19,5652
10000	0	0	0,2	20	0,1087	10,8667

Resultados experimentos datos hepáticos.

1000 Iteraciones

Número de Capas Ocultas		Capa 1					Capa 2		
No. Neuronas por capa		5	10	15	20	25	5	10	15
Error en Porc. (% prueba)		29,7872	23,1383	16,4894	15,159600	15,1596	30,8511	13,5638	15,1596
	Para la Clase 1	25,9259	32,0988	22,2222	25,925900	29,6296	39,5062	32,0988	33,3333
	Para la Clase 2	7,2000	12,8000	7,2000	8,000000	4,8000	21,6000	7,2000	5,6000
	Para la Clase 3	93,1000	35,6322	31,0345	20,689700	21,8391	60,9195	16,0920	22,9885
	Para la Clase 4	1,2048	16,8675	9,6386	9,638600	9,6386	4,8193	2,4096	3,6145
Error en Porc. (% validación)		41,8750	35,0000	35,0000	29,375000	34,3750	48,7500	35,0000	40,6250
	Para la Clase 1	60,0000	42,8571	48,5714	42,857100	60,0000	68,5714	60,0000	68,5714
	Para la Clase 2	16,9811	30,1887	26,4151	20,754700	18,8679	33,9623	28,4151	20,7547
	Para la Clase 3	94,5946	40,5405	45,9459	35,135100	45,9459	81,0811	48,6486	56,7568
	Para la Clase 4	5,7143	28,5714	22,8571	22,857100	20,0000	17,1429	8,5714	25,7143
MSE (Prueba)		0.0753779	0.0576188	0.049973	0.0474856	0.0478009	0.0706061	0.0395263	0.0430767
MSE (Validación)		1,2688	1,1187	1,0750	1,043800	0,9625	1,4500	1,0375	1,3188

Capa 3					Capa 4				
5	10	15	20	25	5	10	15	20	25
33,7766	14,6277	9,3085	7,7128	2,65960000	26,5957	12,5000	7,9787	8,2447	1,5957
22,2222	19,7531	18,5185	11,1111	4,93830000	33,3333	28,3951	20,9877	17,2840	2,4691
18,4000	8,0000	3,2000	4,8000	0,00000000	16,0000	8,0000	0,8000	5,6000	1,6000
93,1034	29,8851	6,8966	16,0920	5,74710000	58,6207	11,4943	10,3448	10,3448	1,1494
6,0241	3,6145	12,0482	0,0000	1,20480000	2,4096	4,8193	3,6145	1,2048	1,2048
43,7500	30,0000	34,3750	31,2500	29,37500000	37,5000	38,7500	38,7500	35,6250	38,1250
54,2857	48,5714	57,1429	48,5714	45,71430000	57,1429	60,0000	60,0000	57,1429	40,0000
22,6415	28,3019	26,4151	22,6415	24,52830000	24,5283	37,7358	30,1887	30,1887	33,9623
97,2973	32,4324	37,8378	54,0541	37,83780000	67,5676	45,9459	43,2432	43,2432	59,4595
8,5714	11,4286	20,0000	2,8571	11,42860000	5,7143	11,4286	25,7143	14,2857	20,0000
0.0749671	0.0416562	0.0325267	0.02922	0.0194616	0.0654613	0.0375746	0.0311353	0.0272441	0.0148183
1,4188	1,1187	0,9938	0,8187	0,93750000	1,0625	0,9750	1,1063	1,2000	0,9875

2000 Iteraciones

Número de Capas Ocultas		Capa 1					Capa 2		
No. Neuronas por capa		5	10	15	20	25	5	10	15
Error en Porc. (% prueba)		25,531900	19,946800	13,829800	12,500000	7,978700	14,893600	11,436200	4,787200
	Para la Clase 1	37,037000	28,395100	23,456800	19,753100	8,642000	9,876500	23,456800	11,111100
	Para la Clase 2	9,600000	6,400000	2,400000	2,400000	4,800000	5,600000	6,400000	8,000000
	Para la Clase 3	44,827600	35,632200	24,137900	26,436800	14,942500	35,632200	14,942500	5,747100
	Para la Clase 4	18,072300	15,662700	10,843400	6,024100	4,819300	12,048200	3,614500	3,614500
Error en Porc. (% validación)		36,875000	39,375000	29,375000	34,375000	33,125000	38,125000	37,500000	31,250000
	Para la Clase 1	54,285700	60,000000	45,714300	57,142900	40,000000	57,142900	65,714300	51,428600
	Para la Clase 2	22,641500	28,301900	16,981100	24,528300	30,188700	28,301900	20,754700	26,415100
	Para la Clase 3	48,648600	48,648600	32,432400	43,243200	37,837800	43,243200	54,054100	37,837800
	Para la Clase 4	28,571400	25,714300	28,571400	17,142900	25,714300	28,571400	17,142900	11,428600
MSE (Prueba)		0.063158	0.052773	0.0428851	0.0416368	0.0357516	0.042361	0.0398722	0.0250047
MSE (Validación)		1,133130	1,256300	1,012500	1,131300	1,037500	1,325000	1,337500	1,018800

Capa 3					Capa 4				
5	10	15	20	25	5	10	15	20	25
19,148900	5,053200	4,521300	1,595700	1,329800	22,872300	3,732400	1,595700	1,329800	0,266000
23,456800	11,111100	12,345700	3,703700	3,703700	41,975300	2,469100	1,234600	4,938300	0,000000
17,600000	4,000000	0,800000	0,800000	0,800000	15,200000	1,600000	0,000000	0,000000	0,000000
33,333300	3,448300	6,896600	0,000000	1,149400	36,781600	10,344800	4,597700	1,149400	1,149400
2,409600	2,409600	0,000000	2,409600	0,000000	1,204800	1,204800	1,204800	0,000000	0,000000
36,875000	36,875000	43,750000	36,875000	35,000000	40,000000	38,125000	39,375000	35,000000	32,500000
54,285700	57,142900	71,428600	48,571400	57,142900	57,142900	51,428600	60,000000	65,714300	51,428600
35,849100	26,415100	43,396200	22,641500	33,962300	28,301900	28,301900	41,509400	26,415100	28,301900
48,648600	51,351400	45,945900	59,459500	45,945900	64,864900	56,756800	43,243200	40,540500	35,135100
8,571400	17,142900	14,285700	22,857100	2,857100	14,285700	20,000000	11,428600	11,428600	17,142900
0.0482193	0.0213782	0.022195	0.0132823	0.0143975	0.0557242	0.0154601	0.0102226	0.0101373	0.0121596
1,250000	1,181200	1,387500	1,106300	0,862500	1,175000	1,243700	1,043800	0,981200	1,125000

3000 Iteraciones.

Número de Capas Ocultas	Capa 1					Capa 2		
	5	10	15	20	25	5	10	15
No. Neuronas por capa	23,670200	14,893600	11,436200	7,446800	5,851100	17,287200	6,914900	5,319100
Error en Porc. (% prueba)	40,740700	17,284000	19,753100	13,580200	9,876500	27,160500	23,456800	8,642000
Para la Clase 1	11,200000	3,200000	3,200000	1,600000	4,000000	10,400000	0,800000	0,800000
Para la Clase 2	29,885100	29,885100	17,241400	11,494300	5,747100	22,988500	3,448300	6,896600
Para la Clase 3	19,277100	14,457800	9,638600	6,024100	4,819300	12,048200	3,614500	7,228900
Para la Clase 4	39,375000	33,125000	34,375000	28,750000	33,750000	35,000000	39,375000	26,875000
Error en Porc. (% validación)	65,714300	57,142900	48,571400	42,857100	37,142900	60,000000	65,714300	42,857100
Para la Clase 1	32,075500	16,981100	18,867900	16,981100	32,075500	22,641500	24,528300	24,528300
Para la Clase 2	35,135100	40,540500	56,756800	37,837800	43,243200	37,837800	48,648600	24,324300
Para la Clase 3	28,571400	25,714300	20,000000	22,857100	22,857100	25,714300	25,714300	17,142900
Para la Clase 4	0.0534413	0.0455515	0.0404017	0.0335413	0.0282266	0.0471601	0.0248469	0.0252976
MSE (Prueba)	1,087500	1,131300	1,100000	0,906300	0,943700	1,268800	1,206300	0,937500
MSE (Validación)								

Capa 3					Capa 4				
5	10	15	20	25	5	10	15	20	25
30,585100	3,457400	0,797900	0,000000	0,266000	17,553200	4,255300	0,797900	1,063800	0,266000
25,925290	7,407400	1,234600	0,000000	0,000000	22,222200	2,469100	1,234600	3,703700	0,000000
13,600000	2,400000	1,600000	0,000000	0,800000	10,400000	4,000000	0,000000	0,000000	0,000000
87,356300	3,448300	0,000000	0,000000	0,000000	37,931000	5,747100	1,149400	0,000000	0,000000
1,204800	1,204800	0,000000	0,000000	0,000000	2,409300	4,819300	1,204800	1,204800	1,204800
53,125000	33,750000	36,250000	30,000000	38,125000	45,625000	40,000000	40,000000	37,500000	36,250000
65,714300	51,428600	45,714300	42,857100	54,285700	57,142900	68,571400	54,285700	57,142900	51,428600
39,622600	33,962300	37,735800	33,962300	39,622600	33,962300	24,528300	28,301900	32,075500	37,735800
94,594600	40,540500	40,540500	29,729700	43,243200	72,973000	48,648600	62,162200	43,243200	40,540500
17,142900	8,571400	20,000000	11,428600	14,285700	22,857100	25,714300	20,000000	20,000000	14,285700
0.0666471	0.0138736	0.0109671	0.00725269	0.00637632	0.0460495	0.0198061	0.00589434	0.0104609	0.0034823
2,043800	1,062500	1,081300	0,856200	0,881200	1,112500	1,118700	1,331300	1,100000	0,987500

4000 Iteraciones

Número de Capas Ocultas	Capa 1					Capa 2		
	5	10	15	20	25	5	10	15
No. Neuronas por capa	27,12770000	15,15960000	7,18090000	5,31910000	3,45740000	26,32980000	2,65960000	2,65960000
Error en Porc. (% prueba)	30,86420000	28,39510000	20,98770000	9,87650000	7,40740000	17,28400000	1,23460000	1,23460000
Para la Clase 1	12,00000000	4,00000000	0,80000000	2,40000000	0,00000000	2,40000000	0,00000000	2,40000000
Para la Clase 2	51,72410000	20,68970000	6,89660000	6,89660000	6,89660000	93,10340000	5,74710000	5,74710000
Para la Clase 3	20,48190000	13,25300000	3,61450000	3,61450000	1,20480000	1,20480000	1,20480000	1,20480000
Para la Clase 4	36,87500000	40,62500000	38,12500000	40,00000000	28,75000000	43,12500000	40,00000000	40,00000000
Error en Porc. (% validación)	48,57140000	68,57140000	68,57140000	57,14290000	42,85710000	62,85710000	62,85710000	42,85710000
Para la Clase 1	20,75470000	18,86790000	28,30190000	37,73580000	26,41510000	15,09430000	37,73580000	37,73580000
Para la Clase 2	54,05410000	59,45950000	40,54050000	45,94590000	29,72970000	89,18920000	59,45950000	59,45950000
Para la Clase 3	31,42860000	25,71430000	20,00000000	20,00000000	17,14290000	17,14290000	20,00000000	20,00000000
Para la Clase 4	0.0613336	0.0472686	0.0359094	0.0296367	0.0272408	0.0654665	0.0142677	0.0142677
MSE (Prueba)	1,33750000	1,30000000	1,21880000	1,16880000	0,96880000	1,22500000	1,24370000	1,24370000
MSE (Validación)								

Capa 3					Capa 4				
5	10	15	20	25	5	10	15	20	25
18,88300000	6,64890000	0,26600000	0,00000000	0,00000000	32,71280000	1,86170000	0,26600000	0,53190000	0,26600000
23,45680000	23,45680000	0,00000000	0,00000000	0,00000000	50,61730000	3,70370000	1,23460000	0,00000000	0,00000000
13,60000000	1,60000000	0,00000000	0,00000000	0,00000000	9,60000000	3,20000000	0,00000000	0,80000000	0,80000000
39,08050000	1,14940000	0,00000000	0,00000000	0,00000000	73,56320000	0,00000000	0,00000000	0,00000000	0,00000000
1,20480000	3,61450000	1,20480000	0,00000000	0,00000000	7,22890000	0,00000000	0,00000000	1,20480000	1,20480000
45,62500000	41,25000000	40,00000000	39,37500000	33,75000000	45,00000000	39,37500000	36,25000000	39,37500000	39,37500000
51,42860000	62,85710000	42,85710000	60,00000000	42,85710000	62,85710000	40,00000000	54,28570000	57,14290000	51,42860000
50,94340000	35,84910000	39,62260000	35,84910000	39,62260000	24,52830000	47,16980000	26,41510000	43,39620000	43,39620000
59,45950000	56,75680000	64,86490000	45,94590000	29,72970000	83,78380000	51,35140000	51,35140000	32,43240000	32,43240000
17,14290000	11,42860000	11,42860000	17,14290000	20,00000000	17,14290000	14,28570000	17,14290000	22,85710000	22,85710000
0.045097	0.0214394	0.00452766	0.00460704	0.00389674	0.0716932	0.00937006	0.00490268	0.00435301	0.00435301
1,26880000	1,18750000	1,04380000	1,23130000	0,93750000	1,33130000	1,05620000	1,09380000	1,23750000	1,23750000