

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA

**UN NUEVO MÉTODO GENERALIZADO
PARA INTEGRACIÓN DE RESPUESTAS
CON REDES NEURONALES
USANDO LÓGICA DIFUSA TIPO-2**

Tesis para obtener el grado de:

Doctor en Ciencias

Presenta:

OLIVIA MENDOZA DUARTE

Directora:

DRA. ELBA PATRICIA MELIN OLMEDA

Co-director:

DR. GUILLERMO LICEA SANDOVAL

Tijuana, B.C., Agosto de 2009

*Este trabajo está dedicado a
Miguel Alberto,
Eva Alejandra y Adriana Carolina*

AGRADECIMIENTOS

Este trabajo es resultado de una estrecha colaboración entre el programa de MYDCI de la UABC y la División de Estudios de Posgrado del ITT, particularmente con el Cuerpo Académico Sistemas Híbridos Inteligentes.

Quiero agradecer particularmente a la Dra. Patricia Melin, por dirigir mi tesis, por haber sido una excelente guía, induciéndome a mejorar la calidad de mi trabajo, por tener confianza en mis ideas, por las incontables oportunidades que me brindó, por su paciencia y trato humano.

Quiero expresar mi gratitud al Dr. Oscar Castillo, por sus invaluable consejos, por estar siempre presente cuando solicité su asesoría y apoyo, así como por haber dado a conocer mi trabajo en tantos foros internacionales.

Quiero agradecer al Dr. Guillermo Licea y al Dr. Antonio Rodríguez por su apoyo para mi ingreso al programa de doctorado, por sus consejos y asesoría durante todo el proceso. Les agradezco todas las conversaciones que mantuvimos antes y durante el periodo de estudios, en las que me expresaron su confianza y me motivaron a seguir adelante.

Quiero agradecer al Dr. Juan Ramón Castro por sus valiosas aportaciones en la implementación de la herramienta que complementó mi trabajo para el desarrollo de varias publicaciones.

Un gran agradecimiento a la UABC por haberme brindado esta gran oportunidad de superación profesional y personal. Gracias también al CONACYT por haberme apoyado durante dos años de estudios.

Dejo para el final el mas importante y mas grande agradecimiento, para mi familia, porque sin su apoyo, paciencia y amor, esta hazaña jamás hubiera sido posible. Gracias por creer en mí, por perdonar todas mis ausencias y mi falta de atención a ustedes y a nuestra casa.

RESUMEN

La tesis que se presenta nuevo método de integración de respuestas para sistemas neuronales multi-red que utiliza lógica difusa tipo-2 por intervalos, el cual demuestra un mejor desempeño que los métodos tradicionales, principalmente en aplicaciones de reconocimiento de patrones y predicción. Debido a que las redes neuronales son una de las técnicas mas flexibles utilizadas en reconocimiento de patrones y predicción, pueden ser aplicadas a diferentes tipos de patrones y tipos de datos, ya que excepto los datos de entrenamiento no es necesario suministrar información adicional.

Durante el desarrollo del nuevo método, se aplicaron sistemas difusos tipo-2 por intervalos, medidas difusas e integrales difusas. Se abordaron dos de las arquitecturas de sistemas multi-red, ensemble y modular. Muchos problemas de alta complejidad pueden ser resueltos desde el enfoque modular, el cual toma como base la modularidad del cerebro biológico, dedicando un grupo específico de neuronas para distintas funciones, mientras que desde el enfoque “ensemble”, también conocido como “comité”, el hecho de combinar redes redundantes con desempeños diferentes, permite aprovechar las mejores características de cada miembro, pudiendo aplicar para este propósito diversos métodos de integración. El reto entonces fué desarrollar un método eficiente de integración de resultados en ambos enfoques y que funcionara correctamente para distintas aplicaciones.

La implementación del método fue integrada y puede ser probada a la herramienta gráfica “Herramientas Multi-red”, la cual permite el modelado de sistemas multi-red por medio de una interfaz gráfica en Matlab.

Palabras Clave: Sistemas modulares, sistemas ensemble, medidas difusas, integral difusa, lógica difusa tipo-2 por intervalos.

ABSTRACT

The thesis that it is presented is a new response integration method for multi-net neural systems that uses interval type-2 fuzzy logic and demonstrate a better performance than the traditional methods, especially in pattern recognition and prediction applications. Because neural networks are one of the most flexible techniques used for pattern recognition and prediction, they can be applied to different pattern types and data type, since except the training data, is not necessary another additional information as input.

Throughout the new method development, interval type-2 fuzzy systems, fuzzy measures and fuzzy integrals were applied. Two of the multi-net systems architectures were approached, ensemble and modular. Many high complexity problems can be solved with modular approach, taking as beginning the biological brain modularity, dedicating a specific neuron group for distinct functions, since the ensemble approach, the fact of combining redundant nets with different performance, allows to be useful the best features of each member, being able to apply for this purpose distinct integration methods. The challenge was to develop an efficient response integration method for both approaches and that would work correctly for different applications.

The method was added and can be tested in the “Multi-net tool box”, developed for design and tests in multi-net systems.

Keywords: Modular systems, ensemble systems, fuzzy measures, fuzzy integral, interval type-2 fuzzy logic.

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA
COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 027

Tijuana, B. C., a 31 de julio de 2009

C. OLIVIA MENDOZA DUARTE
Pasante de: Doctor en Ciencias
Presente

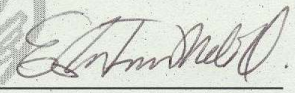
El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

Es propuesto, por los C. Dres. Elba Patricia Melin Olmeda y Guillermo Licea
Sandoval

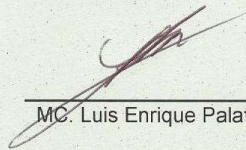
quienes serán los responsables de la calidad del trabajo que usted presente,
referido al tema UN NUEVO MÉTODO GENERALIZADO PARA INTEGRACIÓN DE
RESPUESTAS CON REDES NEURONALES USANDO LÓGICA DIFUSA TIPO 2.

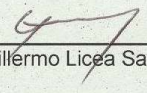
el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

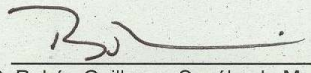
- I.- INTRODUCCIÓN
- II.- ANTECEDENTES
- III.- DESARROLLO DEL MÉTODO GENERALIZADO PARA INTEGRACIÓN DE
RESPUESTAS CON REDES NEURONALES USANDO LÓGICA DIFUSA TIPO 2
- IV.- APLICACIÓN 1: RECONOCIMIENTO DE IMÁGENES
- V.- APLICACIÓN 2: PREDICCIÓN DE SERIES DE TIEMPO
- VI.- CONCLUSIONES
- VII.- REFERENCIAS


Dra. Elba Patricia Melin Olmeda

Asesor


MC. Luis Enrique Palafox Maestre
Sub-Director Secretario


Dr. Guillermo Licea Sandoval
Asesor


MC. Rubén Guillermo Sepúlveda Marques
Director

CONTENIDO

ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	10
1 INTRODUCCIÓN.....	11
1.1 PUBLICACIONES ASOCIADAS AL TRABAJO DE TESIS	12
2 ANTECEDENTES	14
2.1 OPERADORES DE AGREGACIÓN	14
2.2 MEDIDAS DIFUSAS	16
2.2.1 <i>Medidas monótonas</i>	16
2.2.2 <i>Medidas de Sugeno</i>	16
2.3 INTEGRALES DIFUSAS	19
2.3.1 <i>La integral de Sugeno</i>	19
2.3.2 <i>La integral de Choquet</i>	20
2.4 SISTEMAS DIFUSOS TIPO-2 POR INTERVALOS	21
2.5 SISTEMAS NEURONALES MULTI-RED.....	26
2.5.1 <i>Estructuras</i>	27
2.5.2 <i>Métodos para combinar redes neuronales en ensembles</i>	29
2.5.3 <i>Métodos para combinar redes neuronales modulares</i>	30
2.6 RECONOCIMIENTO DE PATRONES.....	31
2.7 PREDICCIÓN DE SERIES TEMPORALES	32
3 DESARROLLO DEL MÉTODO GENERALIZADO PARA INTEGRACIÓN DE RESPUESTAS CON REDES NEURONALES USANDO LÓGICA DIFUSA TIPO-2	34
3.1 REDES NEURONALES Y DATOS DE ENTRENAMIENTO	35
3.2 FUNCIÓN DE DESEMPEÑO	36
3.3 SISTEMA DE INFERENCIA DIFUSO TIPO DOS POR INTERVALOS PARA ESTIMAR DENSIDADES DIFUSAS	37
3.3.1 <i>Variables difusas</i>	37
3.3.2 <i>Reglas difusas</i>	40
3.4 CÁLCULO DE LA INTEGRAL DIFUSA	41
3.5 PRESENTACIÓN DE LA SOLUCIÓN.....	42
4 APLICACIÓN 1: RECONOCIMIENTO DE IMÁGENES	43
4.1 ADQUISICIÓN DE DATOS	43
4.1.1 <i>Detección de bordes con un sistema difuso</i>	43
4.2 CÁLCULO DEL GRADIENTE Y MAGNITUD DEL GRADIENTE	44
4.2.1 <i>Sistema de inferencia difuso tipo-1 para detección de bordes</i>	45
4.2.2 <i>Sistema de inferencia difuso tipo-2 por intervalos para detección de bordes</i>	50
4.2.3 <i>Imágenes generadas con operadores de Sobel</i>	53
4.2.4 <i>Detección de bordes con el método de Sobel tradicional y el método de Sobel extendido con sistemas difusos tipo-1 y tipo-2</i>	54
4.3 SISTEMA NEURONAL MODULAR.....	56
4.3.1 <i>La estructura modular</i>	56
4.3.2 <i>Entrenamiento de módulos</i>	58
4.3.3 <i>Simulación de módulos</i>	58
4.3.4 <i>Función de desempeño de los módulos</i>	59
4.4 ESTIMACIÓN DE DENSIDADES DIFUSAS CON UN SISTEMA DIFUSO TIPO-2 POR INTERVALOS	60
4.5 FUSIÓN DE MÓDULOS CON LA INTEGRAL DE SUGENO	61
4.6 RESULTADOS DE RECONOCIMIENTO	63

5	APLICACIÓN 2: PREDICCIÓN DE SERIES DE TIEMPO.....	67
5.1	ADQUISICIÓN DE DATOS	67
5.1.1	<i>Retrazos en los datos</i>	<i>67</i>
5.2	SISTEMA NEURONAL ENSEMBLE	68
5.2.1	<i>La estructura ensemble.....</i>	<i>68</i>
5.2.2	<i>Entrenamiento de miembros</i>	<i>69</i>
5.2.3	<i>Simulación de miembros y su función de desempeño</i>	<i>69</i>
5.3	ESTIMACIÓN DE DENSIDADES DIFUSAS CON UN SISTEMA DIFUSO TIPO-2 POR INTERVALOS	70
5.4	COMBINACIÓN DE MIEMBROS CON INTEGRALES DIFUSAS	70
5.4.1	<i>Pruebas de validación cruzada.....</i>	<i>71</i>
6	CONCLUSIONES.....	73
7	REFERENCIAS	75
8	ANEXOS	80
	ANEXO A : PROGRAMAS	80
	ANEXO B : DEMOSTRACIÓN DE LA HERRAMIENTA MULTI-RED (MULTI-NET TOOLBOX).	91

ÍNDICE DE FIGURAS

FIGURA 2.1. SISTEMAS MODULARES Y ENSEMBLE EN NIVELES DE TAREA Y SUB-TAREA	28
FIGURA 2.2 COMBINACIÓN DE TIPO COOPERATIVO	30
FIGURA 2.3 COMBINACIÓN DE TIPO COMPETITIVO	31
FIGURA 2.4 COMBINACIÓN DE TIPO SECUENCIAL	31
FIGURA 2.5 COMBINACIÓN DE TIPO SUPERVISOR	31
FIGURA 3.1 ESQUEMA DEL MÉTODO GENERALIZADO DE INTEGRACIÓN DE RESPUESTAS	35
FIGURA 3.2 VARIABLES DE ENTRADA PARA EL FIS2 CON UN FACTOR=0.5 PATA LA FOU	40
FIGURA 3.3 ESQUEMA DE LA INTEGRACIÓN DE VECTORES DE SIMULACIÓN	41
FIGURA 4.1 VARIABLE DE ENTRADA DH PARA EL DETECTOR DE BORDES FIS1	46
FIGURA 4.2 VARIABLE DE ENTRADA DV PARA EL DETECTOR DE BORDES FIS1	47
FIGURA 4.3 VARIABLE M CON FUNCIONES DE MEMBRESÍA TIPO-1.	47
FIGURA 4.4 VARIABLE "EDGES" CON FUNCIONES DE MEMBRESÍA TIPO-1.	48
FIGURA 4.5 SUPERFICIE DE SOLUCIÓN PARA DV, DH Y EDGES EN EL DETECTOR DE BORDES FIS1.	49
FIGURA 4.6 SUPERFICIE DE SOLUCIÓN PARA M, DH Y EDGES EN EL DETECTOR DE BORDES FIS1.	49
FIGURA 4.7 SUPERFICIE DE SOLUCIÓN PARA M, DV Y EDGES CON FUNCIONES DE MEMBRESÍA TIPO-1	50
FIGURA 4.8 FUNCIONES DE MEMBRESÍA PARA LAS VARIABLES DH, DV, M Y EDGES PARE EL SISTEMA DIFUSO TIPO-2 POR INTERVALOS USADO COMO DETECTOR DE BORDES.	51
FIGURA 4.9 SUPERFICIE DE SOLUCIÓN PARA LAS VARIABLES DH, DV, M Y EDGES PARE EL SISTEMA DIFUSO TIPO-1 USADO COMO DETECTOR DE BORDES.	52
FIGURA 4.10 SUPERFICIE DE SOLUCIÓN PARA LAS VARIABLES DH, DV, M Y EDGES PARE EL SISTEMA DIFUSO TIPO-2 POR INTERVALOS USADO COMO DETECTOR DE BORDES.	52
FIGURA 4.11 IMAGEN ORIGINAL 1.PGM	53
FIGURA 4.12 IMAGEN GENERADA POR G_x	53
FIGURA 4.13 IMAGEN GENERADA POR G_y	53
FIGURA 4.14 DATOS DE ENTRENAMIENTO: SIETE MUESTRAS DE CADA PERSONA.	57
FIGURA 4.15 PATRÓN: UNA MATRIZ IDENTIDAD DE 40x40 PARA CADA MUESTRA.	57
FIGURA 4.16 ESTRUCTURA DE CADA RED NEURONAL MONOLÍTICA EN EL SISTEMA MODULAR.	58
FIGURA 4.17 ESQUEMA DE LAS MATRICES DE SIMULACIÓN PARA LOS TRES MÓDULOS.	59
FIGURA 4.18 VARIABLES DIFUSAS TIPO-2 POR INTERVALOS PARA LA ESTIMACIÓN DE DENSIDADES DIFUSAS.	60
FIGURA 4.19 COMPORTAMIENTO DE $F(A)$ PARA EL EJEMPLO PROPUESTO	62
FIGURA 4.20 VALORES DE LA INTEGRAL DE SUGENO PARA LA PERSONA 13	63
FIGURA 4.21 PORCENTAJES DE RECONOCIMIENTO PARA CADA PERMUTACIÓN Y ENTRENAMIENTO CON LOS SISTEMAS DIFUSOS TIPO-1 Y TIPO-2.....	65
FIGURA 4.22 LOS MEJORES PORCENTAJES DE RECONOCIMIENTO DE LOS SISTEMAS DIFUSOS TIPO-1 Y TIPO-2.....	65
FIGURA 5.1 ESTRUCTURA DE CADA MIEMBRO DEL ENSEMBLE.....	68
FIGURA 5.2 PORCENTAJES DE RECONOCIMIENTO PARA DIFERENTES FOU, USANDO VALIDACIÓN CRUZADA TIPO K-FOLD .71	
FIGURA 5.3 GRÁFICA DE LA PREDICCIÓN FUNCIÓN MACKEY-GLASS.....	72

ÍNDICE DE TABLAS

TABLA 2-I: OPERADORES DE AGREGACIÓN MÁS UTILIZADOS	15
TABLA 4-I: VALORES MÁXIMOS Y MÍNIMOS PARA POR GY Y GY Y G	54
TABLA 4-II: HISTOGRAMAS DE LAS IMÁGENES RESUTANTES DE LOS DETECTORES DE BORDES DE SOBEL, FIS1 Y FIS2.	55
TABLA 4-III: VALORES DE DE LAS DENSIDADES DIFUSAS ORDENADAS DESCENDENTEMENTE RESPECTO A X.	62
TABLA 4-IV: PORCENTAJES DE RECONOCIMIENTO USANDO EL FIS1 PARA ESTIMAR DENSIDADES DIFUSAS	64
TABLA 4-V: PORCENTAJES DE RECONOCIMIENTO USANDO EL FIS2 PARA ESTIMAR DENSIDADES DIFUSAS.....	64
TABLA 4-VI: PORCENTAJES DE RECONOCIMIENTO PARA LOS MÉTODOS: RBF, BDPCALDA, UEFMMC Y T1FL+MNN, IT2 FL+MNN.	66

1 Introducción

En este trabajo de tesis se presenta un nuevo método de integración de respuestas para sistemas multi-red. Debido a la flexibilidad de las redes neuronales, éste método puede usarse en una variedad de aplicaciones como reconocimiento de patrones o predicción de series de tiempo. Entonces podemos decir que el método es generalizado, porque puede ser aplicado para fusión de redes neuronales modulares o combinación de miembros en ensembles para diferentes tipos de aplicaciones.

Muchas aplicaciones son de alta complejidad debido al volumen de los datos; entonces algunas veces una red neuronal monolítica no es suficiente para procesar todos los datos, o si lo hace, el rendimiento es pobre. El diseño de sistemas multi-red puede ser una buena solución para este tipo de aplicaciones complejas. Algunos problemas pueden ser resueltos con el enfoque modular inspirado en la modularidad del cerebro biológico, en el cual un grupo específico de neuronas se enfoca a una función particular. En el enfoque ensemble, las redes neuronales son combinadas de forma redundante, así que cuando cada neuronal tiene rendimiento diferente a las demás, puede encontrarse el mejor resultado al combinarse [43][49].

En este punto los métodos de agregación juegan un rol fundamental, ya que deben ser capaces de estimar la relevancia de cada nodo del sistema multi-red con el propósito de obtener el mejor resultado. El método propuesto incluye la integral difusa como operador de agregación y un sistema de inferencia difusa para estimar densidades difusas. En la teoría de medidas difusas, las densidades difusas califican la relevancia de las fuentes de información en problemas de decisión multi-criterios [20] [53].

Este documento está organizado de la siguientes manera: en el capítulo 2 se explican brevemente los antecedentes de sistemas muti-red, medidas difusas, integrales difusas, operadores de agregación, lógica difusa tipo-2 por intervalos, reconocimiento de patrones y predicción. En el capítulo 3 se da una explicación del método generalizado, en el capítulo 4 se detalla cómo fue aplicado el método para el problema de reconocimiento de imágenes, y en el capítulo 5 se describe la aplicación del método para predicción de series de tiempo.

1.1 Publicaciones asociadas al trabajo de tesis

A continuación se enlistan las publicaciones donde se presentaron resultados relacionados con este trabajo de tesis.

Capítulos en libros

- Mendoza O, Melin P., Licea G., 2007, Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition en: Foundations of Fuzzy Logic and Soft Computing, Springer, Alemania, pp. 604-612.
- Mendoza O., Melin P., Licea G., 2008, Interval Type-2 Fuzzy Logic for Module Relevance Estimation en: Sugeno Integration of Modular Neural Networks, Soft Computing for Hybrid Intelligent Systems, Springer, Alemania, pp. 115-127.

Artículos en revistas

- Mendoza O., Melin P., Licea G., 2008, A Hybrid Approach for Image Recognition Combining Type-2 Fuzzy Logic, Modular Neural Networks and the Sugeno Integral, Information Sciences, Vol., 179, Elsevier, E.U.A, pp. 2078-2101.
- Mendoza O., Melin P., Licea G., 2007, Fuzzy Inference Systems Type-1 and Type-2 for Digital Images Edges Detection, Engineering Letters, International Association of Engineers, E.U.A., Volume 15 Issue 1 (Available on line: http://www.engineeringletters.com/issues_v15/issue_1/EL_15_1_7.pdf)

- Mendoza O., Melin P., Licea G., 2007, Type-2 Fuzzy Systems as a Method for Improving Training Data and Decision Making in Modular Neural Networks for Pattern Recognition, International Journal of Information Technology And Intelligent Computing, Academy of Humanities And Economics (Wshe), Polonia, Int. J. It&Ic Vol.1 No.4, p.p., available in <http://itic.wshe.lodz.pl>,

Memorias en congresos

- Mendoza O., Melin P., Licea G., 2007, Modular Neural Networks and Type-2 Fuzzy Logic for Face Recognition, NAFIPS 2007 International Conference, San Diego, E.U.A., pp. 622-627.
- Mendoza O., Melin P., Licea G., 2007, Type-2 Fuzzy Systems for Improving Training Data and Decision Making in Modular Neural Networks for Image Recognition, International Joint Conference on Neural Networks, Orlando, E.U.A, pp. 1766-1770.
- Mendoza O, Melin P., Licea G., 2007, A New Method for Edge Detection in Image Processing Using Interval Type-2 Fuzzy Logic, 2007 IEEE International Conference on Granular Computing (GRC 2007) , Silicon Valley. Ca, E.U.A., pp 151-156.
- Mendoza O., Melin P., Licea G., 2008, Estimating Modules Relevance on Sugeno Integration of Modular Neural Networks using Type-2 Fuzzy Logic, 2008 IEEE World Congress on Computational Intelligence, Hong Kong, China, pp. 1330-1336.

2 Antecedentes

A continuación se dará una breve semblanza de los temas tocados en esta tesis:

2.1 Operadores de agregación

El propósito de la agregación es hacer uso simultáneo de diferentes fuentes de información con el propósito de llegar a una sola conclusión o tomar una decisión. Los operadores de agregación son objetos matemáticos que tienen la función de reducir un conjunto de números a un único número representativo. Es importante notar que cualquier agregación o proceso de fusión hecho con una computadora implica agregación numérica [20] [53].

Los sistemas computacionales que necesitan hacer fusión de información usualmente requieren trabajo preliminar antes del proceso de agregación. De forma general, podemos distinguir entre los conceptos: integración de información, fusión de información y operadores de agregación.

- **Integración de información:** es el proceso completo de usar información de diferentes fuentes para un propósito particular. Por ejemplo, un sistema para toma de decisiones para el mejor destino de un viaje, basándose en información en línea.
- **Fusión de información:** son las funciones particulares y métodos para combinar información. Por ejemplo, métodos para integrar información de la web o funciones para combinar grados de satisfacción.
- **Operadores de agregación:** son las funciones particulares para combinar información cuando estas son formalizadas matemáticamente. Por ejemplo, el uso de la media aritmética para combinar grados de satisfacción. En la Tabla 2-I se enlistan algunos operadores de agregación.

TABLA 2-I: OPERADORES DE AGREGACIÓN MÁS UTILIZADOS

Operador	Definición
Media aritmética	$\sum_{i=1}^N a_i / N$
Media geométrica	$\left(\prod_{i=1}^N a_i\right)^{1/N}$
Media armónica	$\frac{N}{\sum_{i=1}^N \frac{1}{x_i}}$
Media ponderada	$\sum_{i=1}^N p_i a_i$
Ordered weighted averaging (OWA)	$\sum_{i=1}^N w_i a_{\sigma(i)}$
Integral de Choquet (CI)	$\sum_{i=1}^N [f(x_{s(i)}) - f(x_{s(i-1)})] \mu(A_{s(i)})$
Integral de Sugeno (SI)	$\max_{i=1, N} \min(f(x_{s(i)}), \mu(A_{s(i)}))$
Weighted OWA (WOWA)	$\sum_{i=1}^N w_i a_{\sigma(i)}$ con $w_i = w^*\left(\sum_{j \leq i} P_{\sigma(j)}\right) - w^*\left(\sum_{j < i} P_{\sigma(j)}\right)$ w^* función no decreciente de: $\left\{ \frac{i}{N}, \sum_{j \leq i} w_j \right\}_{i=1, \dots, N} \cup \{(0,0)\}$

2.2 Medidas difusas

2.2.1 Medidas monótonas

Las medidas monótonas se encuentran frecuentemente en la literatura como medidas difusas. Este nombre es confuso algunas veces, ya que no hay conjuntos difusos en la definición de medidas monótonas. Para prevenir esta confusión, el término “medidas difusas” deberían ser reservadas a medidas (aditivas o no-aditivas) que sean definidas en familias de conjuntos difusos [11] [52] [55].

Dado un conjunto universal X y una familia no vacía C de subconjuntos de X , una medida monótona μ en $\langle X, C \rangle$ es una función de la forma $\mu : C \rightarrow [0, \infty]$.

Se asume que el conjunto universal X es finito y que $C = P(X)$.

Esto es, se asume normalmente que las medidas monótonas que nos ocupan son conjuntos de funciones $\mu : P(X) \rightarrow [0, 1]$.

Un conjunto de medidas monótonas μ en un espacio X es un mapeo $\mu : P(X) \rightarrow [0, 1]$ tal que:

- 1) $\mu(\emptyset) = 0$
- 2) $\mu(X) = 1$
- 3) para toda $A, B \in P(X)$, si $A \subseteq B$, entonces $\mu(A) \leq \mu(B)$

2.2.2 Medidas de Sugeno

Las medidas de Sugeno o medidas λ son medidas monótonas especiales definidas de la siguiente forma:

Sea $x=\{x_1, \dots, x_n\}$ cualquier conjunto finito. Una medida difusa discreta en x es una función $\mu: 2^x \rightarrow [0,1]$ con las siguientes propiedades:

$$1) \mu(\phi) = 0 \text{ y } \mu(X) = 1;$$

$$2) \text{ dadas } A, B \in 2^x \text{ si } A \subset B \text{ entonces } \mu(A) \leq \mu(B) \text{ (propiedad de monotonicidad).}$$

El conjunto x es considerado para contener los nombres de fuentes de información (características, algoritmos, agentes, sensores, etc), y para un subconjunto $A \subseteq X$, $\mu(A)$ es considerado como el peso o grado relevancia de ese subconjunto de información **Error!**

Reference source not found..

Una medida difusa es una medida de Sugeno o λ -difusa, si esta satisface la condición de adición para alguna $\lambda > -1$, descrita con la expresión (1).

$$\mu(A \cup B) = \mu(A) + \mu(B) + \lambda \mu(A) \mu(B) \quad (1)$$

λ puede calcularse resolviendo la función de una sola variable representada por (2) o (3):

$$\mu(S) = \left[\prod_{x \in S} (1 + \lambda \mu(\{x\})) \right] / \lambda \quad (2)$$

$$\lambda + 1 = \prod_{i=1}^n (1 + \lambda \mu(\{x_i\})) \quad (3)$$

El valor del parámetro λ es determinado por las condiciones del siguiente teorema:

Teorema 1. Sea $\mu(\{x\}) < 1$ para cada $x \in x$ y $\mu(\{x\}) > 0$ para al menos dos elementos de x .

Entonces la ecuación (3) determina un único parámetro λ de la siguiente forma [20]:

- Si $\sum_{x \in X} \mu(\{x\}) < 1$, entonces λ es igual a una única raíz de la ecuación en el intervalo $(0, \infty)$.

- Si $\sum_{x \in X} \mu(\{x\}) = 1$, entonces $\lambda=0$; la cual es la única raíz de la ecuación.
- Si $\sum_{x \in X} \mu(\{x\}) > 1$, entonces λ es igual a una única raíz en la ecuación en el intervalo $(-1,0)$.

Cualquier medida- λ es entonces completamente determinada a partir de los valores de todos los conjuntos $\mu(\{x\})$, para toda $x \in X$ con la ecuación (3). De acuerdo con el teorema 1, se pueden distinguir tres situaciones:

- Si $\sum_{x \in X} \mu(\{x\}) < 1$, lo que significa que μ califica como una probabilidad mas baja, $\lambda > 0$.
- Si $\sum_{x \in X} \mu(\{x\}) = 1$, lo que significa que μ es una probabilidad clásica, $\lambda = 0$.
- Si $\sum_{x \in X} \mu(\{x\}) > 1$, lo que significa que μ califica como una probabilidad mas alta, $\lambda < 0$.

Las medidas difusas pueden calcularse de forma recursiva, usando (4), (5).

$$\mu(S_1) = \mu(x_i) \quad (4)$$

$$\mu(S_i) = \mu(S_i) + \mu(S_{i-1}) + \lambda \mu(S_i) \mu(S_{i-1}) \quad (5)$$

Donde $1 < i \leq n$ y los valores de $\mu(x_i)$ corresponden a las densidades difusas determinadas por un experto. Una restricción fundamental para utilizar este método recursivo es el ordenamiento de las densidades difusas en orden descendente respecto a las fuentes de información a combinar [54] .

2.3 Integrales difusas

Existen dos tipos de integral difusa: la integral de Choquet y la integral de Sugeno.

2.3.1 La integral de Sugeno

La integral difusa es un operador propuesto por Sugeno en 1974. Este operador es usado para resolver problemas de toma de decisiones multi-criterio, donde la información que es combinada están basadas en medidas difusas determinadas por un experto. La meta es la simulación del proceso humano de integración de diferentes fuentes de información.

La integral de Sugeno generaliza el mínimo ponderado y el máximo ponderado, expresando la relevancia de cada vector usando medidas difusas. La integral de Sugeno, debido a que la combinación de pesos y valores usando el operador mínimo, es definida por funciones en $[0,1]$ y medidas difusas normalizadas **Error! Reference source not found.**[20].

Definición 4: Sea μ una medida difusa en x ; entonces, la integral de Sugeno de una función $f : X \rightarrow [0,1]$ con respecto a μ es definida por la ecuación (6).

$$\int f d\mu = \max_{i=1,N} \min(f(x_{s(i)}), \mu(A_{s(i)})) \quad (6)$$

donde $f(x_{s(i)})$ indica que los índices deben ser permutados de manera que

$$0 \leq f(x_{s(1)}) \leq \dots \leq f(x_{s(N)}) \leq 1 \text{ y } A_{s(i)} = \{x_{s(i)}, \dots, x_{s(N)}\}.$$

Para funciones discretas, la ecuación (4) puede expresarse como la ecuación (7).

$$\max_i \min(f(x_{\sigma(i)}), \mu(A_{\sigma(i)})) \quad (7)$$

Algunas propiedades de la integral de Sugeno son las siguientes:

- $A_{\sigma(k)} = \{x_{\sigma(j)} \mid j \leq k\}$

- o, de manera equivalentes $A_{\sigma(k)} = \{x_{\sigma(1)}, \dots, x_{\sigma(k)}\}$ donde $k \geq 1$ y $A_{\sigma(0)} = \emptyset$
- σ es una permutación de tal que $f(x_{\sigma(i+1)}) \geq f(x_{\sigma(i)})$ para $i \geq 1$.

La integral de Sugeno puede ser aplicada para resolver varios problemas, que consideren un conjunto finito de n elementos $x = \{x_1, \dots, x_n\}$. Dependiendo de la aplicación, estos elementos pueden ser jugadores en un juego cooperativo, criterios en problemas de decisión multi-criterio, estados de la naturaleza en un problemas decisión bajo incertidumbre, atributos, expertos, o votantes en de opinión en un problema de encuestas, etc.

2.3.2 La integral de Choquet

Cuando se trabaja con probabilidades imprecisas la noción clásica del valor esperado de una función dada es generalizada. Este rango es capturado por valores esperados mas altos o mas bajos, los cuales se basan en funciones de probabilidad mas bajas y mas altas, respectivamente. Ya que las funciones de probabilidad mas bajas y mas altas son funciones monótonas, las cuales por lo general no son aditivas, la integral clásica de Lebesgue, la cual es aplicable solo a medidas aditivas, debe ser generalizada apropiadamente para capturar el valor esperado de la función con respecto a la medida monótona [52].

Generalmente en la literatura que una generalización apropiada de la integral de Lebesgue para medidas monótonas es una funcional llamada integral de Choquet [22] [31].

Cuando X es un conjunto finito y f es una función no negativa, es conveniente presentar la siguiente notación especial para simplificar el cálculo de la integral de Choquet [53].

- Sea $X = \{x_1, x_2, \dots, x_n\}$ y $f(x_k) = f_k$. Asuma que los elementos de X son permutados de manera que $f_1 \geq f_2 \geq \dots \geq f_n \geq 0$ y sea $f_{n+1} = 0$ por convención. Para cada $x_k \in X$, sea

$A_k = \{x_1, x_2, \dots, x_k\}$. Entonces, dada una medida monótona μ en X , la integral de Choquet de f en X con respecto a μ puede ser expresada con la siguiente fórmula

$$(C) \int_X f d\mu = \sum_{k=1}^n (f_k - f_{k+1}) \mu(A_k) \quad (8)$$

Donde $f_{n+1} = 0$ por convención.

2.4 Sistemas difusos tipo-2 por intervalos

La incertidumbre respecto de $\tilde{A}$ proyectada por la unión de todas las membresías primarias, llamadas huella de incertidumbre (FOU por sus siglas en inglés de footprint of uncertainty) de $\tilde{A}$, expresada como (9) [28].

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x = \{(x, y) : y \in J_x = [\underline{A}(x), \overline{A}(x)] \subseteq [0, 1]\} \quad (9)$$

Un IT2 FS (sistema difuso tipo-2 por intervalos o interval type-2 fuzzy set) $\tilde{A}$ es hasta este momento el tipo de conjunto difuso tipo-2 mas usado; y es descrito por la ecuación (10).

$$\tilde{A} = \int_{x \in X} \int_{u \in U} 1/(x, u) = \int_{x \in X} \left[\int_{u \in J_x} 1/u \right] / x \quad (10)$$

Donde x es la variable primaria, $j_x \subseteq [0, 1]$, es la membresía primaria de x , u es la variable secundaria, y $\int_{u \in J_x} 1/u$ es la función de membresía secundaria (mf) en x .

El conjunto difuso tipo-2 por intervalos está limitado por una función de membresía superior $\overline{A}(x) \equiv \overline{A}$ y una inferior $\underline{A}(x) \equiv \underline{A}$, ambas son conjuntos difusos tipo-1; por consecuencia, el grado de membresía de cada elemento de un IT2 FS es un intervalo $[\underline{A}(x), \overline{A}(x)]$ [30].

Note que im ITS FS puede también expresarse con la ecuación (11)

$$\tilde{A} = 1 / FOU(\tilde{A}) \quad (11)$$

Donde se entiende que esto significa situar un segundo grado de uno a todos los puntos del $FOU(\tilde{A})$.

Para universos de discurso discretos x y u , un conjunto difuso tipo-1 embebido tiene n elementos, uno de cada $J_{x_1}, J_{x_2}, \dots, J_{x_n}$, llamado $u_1, u_2, \dots, u_n$ expresado como (12).

$$A_\ell = \sum_{i=1}^N u_i / x_i \quad u_i \in J_{x_i} \subseteq U = [0,1] \quad (12)$$

La función de membresía superior (UMF por sus siglas en inglés: Upper membership function) y función de membresía inferior (LMF por sus siglas en ingles: Lower membership function) de $\tilde{A}$ son dos funciones de membresía tipo-1 que limitan la FOU. UMF está asociada con el límite superior de $FOU(\tilde{A})$ y es denotada por $\overline{\mu_{\tilde{A}}}(x), \forall x \in X$ como se ve en la ecuación (13), y la LMF está asociada con el límite inferior de $FOU(\tilde{A})$ y es denotada por $\underline{\mu_{\tilde{A}}}(x), \forall x \in X$ como se ve en la ecuación (14) [29].

$$\overline{\mu_{\tilde{A}}}(x) \equiv \overline{FOU(\tilde{A})} \quad \forall_x \in X \quad (13)$$

$$\underline{\mu_{\tilde{A}}}(x) \equiv \underline{FOU(\tilde{A})} \quad \forall_x \in X \quad (14)$$

Note que J_x es un intervalo expresado por la ecuación (15).

$$J_x = \left\{ (x, y) : y \in \left[\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x) \right] \right\} \quad (15)$$

$FOU(\tilde{A})$ también puede expresarse con la ecuación (16).

$$FOU(\tilde{A}) = \bigcup_{\forall x \in X} \left[\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x) \right] \quad (16)$$

Para universos de discurso discretos de x y u , un IT2 FS $\tilde{A}_e$ puede ser expresado por la ecuación (17).

$$\tilde{A}_e = \int_{x \in X} [1 / \theta] / x \quad \theta \in J_x \subseteq U = [0, 1] \quad (17)$$

El conjunto $\tilde{A}_e$ está embebido en $\tilde{A}$ tal que cada x solo tiene una variable secundaria (una función de membresía primaria cuyo grado secundario es igual a 1).

Ejemplos de $\tilde{A}_e$ son $1 / \overline{\mu}_{\tilde{A}}(x)$ y $1 / \underline{\mu}_{\tilde{A}}(x)$, $\forall x \in X$. Asociada con cada $\tilde{A}_e$ está una T1 FS embebida A_e , donde

$$A_e = \int_{x \in X} \theta / x \quad \theta \in J_x \subseteq U = [0, 1] \quad (18)$$

El conjunto A_e , que actúa como el dominio de $\tilde{A}_e$, es la unión de todas las funciones de membresía primarias del conjunto $\tilde{A}_e$ en (17). Ejemplos de A_e son $\overline{\mu}_{\tilde{A}}(x)$ y $\underline{\mu}_{\tilde{A}}(x)$, $\forall x \in X$.

A. Operaciones con sistemas difusos tipo-2 por intervalos.

Es relativamente directa la forma para determinar las operaciones conexión(join) y reunión(meet) de conjuntos difusos tipo-2 por intervalos. Los resultados se indican para universos de discurso de las funciones de membresía secundaria [28] [29].

a) Conexión (Join) de conjuntos difusos tipo-2 por intervalos [28].

Theorema 2: La conexión (join) $\sqcup_{i=1}^n F_i$, de n conjuntos tipo-1 por intervalos $F_1, \dots, F_n$, teniendo dominios $[l_1, r_1], \dots, [l_n, r_n]$, respectivamente, es un conjunto de intervalos con dominio $[(l_1 \vee l_2 \vee \dots \vee l_n), (r_1 \vee r_2 \vee \dots \vee r_n)]$, donde $\vee$ denota valor máximo.

La conexión (join) entre F y G , ya sea bajo las operaciones producto o t-norm, puede ser expresada con la ecuación (19).

$$F \sqcup G = \int_{u \in J_x^u} \int_{w \in J_x^w} 1/(u \vee w) \quad (19)$$

donde $J_x^u = [l_f, r_f]$ y $J_x^w = [l_g, r_g]$.

Para empezar, se demuestra que $r_f \vee r_g$ es el punto mas a la derecha en $F \sqcup G$. Para ver esto, note que para cualquier $u \in J_x^u$ y $w \in J_x^w$, desde $w \leq r_g$ entonces $u \vee w = \max(u, w) \geq u \vee l_g$ y desde $u \leq r_f$, $u \vee w \leq r_f \vee r_g$.

Cada término $F \sqcup G$ es igual a $u \vee w$ para algún $u \in J_x^u$ y $w \in J_x^w$, el término mas pequeño siendo $l_f \vee l_g$ y el término mas grande siendo $r_f \vee r_g$. Porque F y G teniendo dominios discretos, $F \sqcup G$ también tienen dominios continuos; por o tanto, $F \sqcup G$ es un conjunto de intervalos tipo-1 con dominio $[l_f \vee l_g, r_f \vee r_g]$.

b) Reunión (Meet) de conjuntos tipo-2 por intervalos [28].

Theorema 3: La reunión (meet), $\bigcap_{i=1}^n F_i$, de n conjuntos tipo-1 por intervalos $F_1, \dots, F_n$, teniendo dominios $[l_1, r_1], \dots, [l_n, r_n]$, respectivamente, donde $l_i \geq 0$ y $r_i \geq 0$ ($i=1, \dots, n$), es un conjunto de intervalos $[(l_1 * l_2 * \dots * l_n), (r_1 * r_2 * \dots * r_n)]$ donde $*$ denota ya sea mínimo o producto t-norm.

La reunión (meet) entre F y G , bajo producto t-norm, puede ser expresada como la ecuación

(20).

$$F \sqcap G = \int_{u \in J_x^u} \int_{w \in J_x^w} (1 \times 1) / (uw) \quad (20)$$

donde $J_x^u = [l_f, r_f]$ y $J_x^w = [l_g, r_g]$. Observe que cada término en $F \sqcap G$ es igual al producto uw para algún $u \in J_x^u$ y $w \in J_x^w$; ya que l_f, l_g, r_f, r_g son mas grandes o iguales a cero, el producto uw mas pequeño es l_f, l_g y el producto uw mas grande es r_f, r_g ; y ya que ambos F y G tienen

dominios continuos, $F \sqcap G$ también tienen dominios continuos; por lo tanto $F \sqcap G$ es un dominio tipo-1 por intervalos $[l_f l_g, r_f r_g]$.

2.5 Sistemas neuronales multi-red

El desarrollo de la investigación en el área de redes neuronales artificiales, se ha encaminado hacia la resolución de problemas cada vez más complejos, por lo que los sistemas con redes neuronales monolíticas son ya insuficientes dando paso a los sistemas neuronales multi-red [49].

La posibilidad de arquitecturas diferentes es muy grande, tomando en cuenta que pueden variar diferentes parámetros, tales como: el número de redes, los diferentes métodos de entrenamiento, las técnicas de muestreo de los datos, combinación entre componentes y el enfoque ya sea modular, ensemble y la combinación de ambos.

En la actualidad, existe software que funciona como herramienta para la creación, entrenamiento y pruebas con redes neuronales monolíticas como la herramienta gráfica de Matlab; para sistemas multi-red la única herramienta que existe al momento inició su desarrollo en 2004, es llamada “herramientas mlti-red” o “multi-net toolbox” [27] [33] [43].

Los sistemas neuronales multi-red pueden convertirse en la solución a tareas que no pueden resolverse con una sola red o que pueden ser resueltas con mayor eficiencia por medio de componentes neuronales modulares. De la misma forma, se puede alcanzar un mayor rendimiento cuando redes neuronales artificiales, son combinadas de manera redundante.

2.5.1 Estructuras

Es importante hacer una distinción entre combinaciones de redes neuronales artificiales de tipos modular y ensemble. El término ensemble es el más comúnmente usado para un conjunto de redes redundantes, aunque también se utilice el término “committee machine” [49].

En la combinación ensemble, las redes componentes son redundantes, ya que todas ellas dan una solución para la misma tarea, o componente de tarea, aunque esa solución se obtenga por diferentes métodos.

En contraste, bajo el enfoque modular, el problema o tarea es descompuesto en varias sub-tareas, y la solución completa requiere de la contribución de todos o algunos de los módulos. Ambas combinaciones ensemble y modular pueden existir tanto en el nivel de tarea como en el de sub-tarea como se muestra en la *Figura 2.1*.

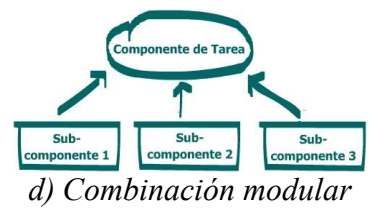
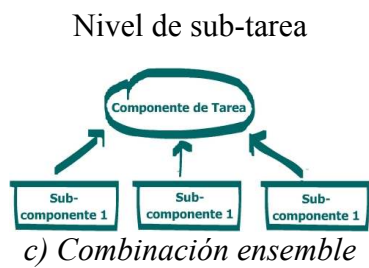
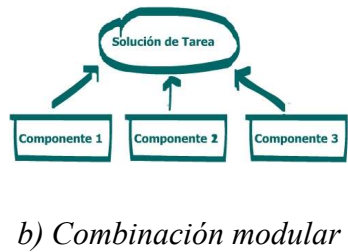
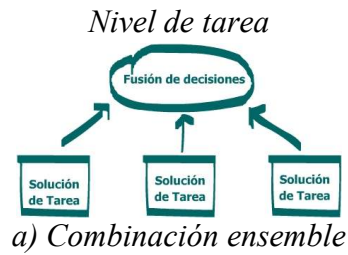


Figura 2.1. *Sistemas modulares y ensemble en niveles de tarea y sub-tarea*

Nivel de tarea: un ensemble puede consistir de un número de diferentes soluciones para completar una tarea o problema. De manera similar, una solución a una tarea puede ser construida por una combinación de módulos.

Nivel de sub-tarea: cuando una aplicación o tarea es descompuesta en componentes modulares, cada componente modular, puede por si mismo consistir de un ensemble de redes, cada una de

las cuales proveen una solución para el mismo componente modular. Alternativamente, cada módulo puede además ser subdividido aún más en módulos especializados.

En los dos niveles de la *Figura 2.1*, la diferencia entre las combinaciones modular y ensemble dependen de la presencia o ausencia de redundancia, nótese los componentes redundantes (varias versiones del subcomponente 1), en el ejemplo de sub-tarea del tipo ensemble mostrada en la *Figura 2.1 c)*, a comparación con la ausencia de redundancia (subcomponentes 1,2 y 3) en el ejemplo del nivel de sub-tarea modular mostrado en la *Figura 2.1 d)*.

2.5.2 Métodos para combinar redes neuronales en ensembles

Una vez que es creado un conjunto de redes, debe ser encontrado un método efectivo para combinar sus salidas. Existen varios métodos para la combinación [49]:

Promedio: una salida puede ser creada a partir de un conjunto de salidas de redes utilizando un simple promedio, o de otra manera, encontrando una relación entre la exactitud relativa de las redes combinadas.

Métodos de combinación no-lineales: los métodos de combinación no-lineales que se proponen incluyen métodos basados en convicción de Dempster-Shafer, combinaciones basados en información de rangos, voto y estadísticos.

Supra bayesianos: la filosofía fundamental del enfoque supra bayesiano, es que las opiniones de lo expertos son datos por sí mismos. Entonces la distribución de la probabilidad de los expertos pueden ser combinados con una distribución anticipada.

Generalización acumulada: bajo la generalización acumulada, una red lineal aprende cómo combinar las redes con pesos que varían sobre el espacio de características. Las salidas de un conjunto de generalizadores de nivel 0 son usados como entrada a un generalizador de nivel 1, el cual es entrenado para producir la salida adecuada.

2.5.3 Métodos para combinar redes neuronales modulares

Se identifican por ahora, cuatro diferentes formas de combinación de módulos; cooperativo, competitivo, secuencial y supervisor [49].

La principal diferencia entre las combinaciones de tipo cooperativo que se muestra en la *Figura 2.2* y de tipo competitivo mostrado en la *Figura 2.3*, es que en la combinación de tipo cooperativo, todos los componentes que van a combinarse harán alguna contribución en la toma de la decisión final, aunque a cada contribución se le de un peso distinto, en cambio en la combinación de tipo competitivo, solo se seleccionará y se tomará en cuenta la mejor contribución hacia la decisión final.

En la combinación de tipo secuencial mostrada en la *Figura 2.4*, el proceso es sucesivo, el cálculo de un módulo depende de la salida del modulo precedente. En la combinación de tipo supervisor mostrada en la *Figura 2.5*, un módulo es usado para supervisar el desempeño de otro módulo.



Figura 2.2 Combinación de tipo cooperativo

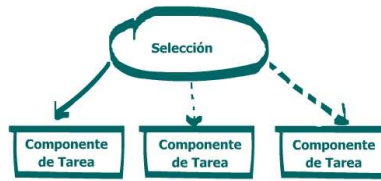


Figura 2.3 *Combinación de tipo competitivo*



Figura 2.4 *Combinación de tipo secuencial*



Figura 2.5 *Combinación de tipo supervisor*

2.6 Reconocimiento de patrones

Es el área de investigación que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades de o entre conjuntos de dichos objetos.

Los patrones a clasificar suelen ser grupos de medidas u observaciones, definiendo puntos en un espacio multidimensional apropiado.

Un sistema de reconocimiento de patrones completo consiste en un sensor que recoge las observaciones a clasificar, un sistema de extracción de características que transforma la información observada en valores numéricos o simbólicos, y un sistema de clasificación o descripción que, basado en las características extraídas, clasifica la medición [16][17].

El reconocimiento estadístico de patrones está basado en las características estadísticas de los patrones, asumiendo que han sido generados por un sistema probabilístico. El reconocimiento estructural de patrones está basado en las relaciones estructurales de las características.

Para la clasificación se puede usar un conjunto de aprendizaje, del cual ya se conoce la clasificación de la información y se usa para entrenar al sistema, siendo la estrategia resultante conocida como aprendizaje supervisado. El aprendizaje puede ser también no supervisado, el sistema no tiene un conjunto para aprender a clasificar la información, sino que se basa en cálculos estadísticos para clasificar los patrones.

Entre las aplicaciones del reconocimiento de patrones son el reconocimiento de voz, la clasificación de documentos, el reconocimiento de escritura, reconocimiento de caras humanas, o reconocimiento de huellas dactilares entre otras [14] [24] [25].

2.7 Predicción de series temporales

En estadística, procesamiento de señales, y econometría, una serie temporal o cronológica es una secuencia de datos, observaciones o valores, medidos en determinados momentos del tiempo, espaciados entre sí de manera uniforme y por tanto ordenados cronológicamente. El análisis de series temporales comprende métodos que ayudan a interpretar éste tipo de datos, extrayendo información representativa, tanto referente a los orígenes o relaciones subyacentes como a la posibilidad de extrapolar y predecir su comportamiento futuro.

De hecho uno de los usos más habituales de las series de datos temporales es su análisis para predicción y pronóstico. Por ejemplo de los datos climáticos, de las acciones de bolsa, o las

series pluviométricas. Resulta difícil imaginar una rama de las ciencias en la que no aparezcan datos que puedan ser considerado como series temporales.

3 Desarrollo del método generalizado para integración de respuestas con redes neuronales usando lógica difusa tipo-2 .

La variación del error en el rendimiento de redes neuronales es inherente a la aleatoriedad de los valores iniciales en los parámetros para los algoritmos de entrenamiento. El uso de sistemas neuronales multi-red permiten conocer los resultados de varios entrenamientos para el mismo problema o partes de un problema y tomar la decisión mas conveniente según la aplicación.

Sin embargo, la integración de los resultados de varias redes neuronales es un punto primordial para la toma de la decisión final, ya que no siempre todas las redes neuronales de un sistema aportan información de igual relevancia por lo que los resultados de redes no muy buenas pueden afectar en forma negativa los resultados de mejores redes en el momento de su integración. Por esta razón es necesario aplicar una serie de criterios para estimar la relevancia de cada red con el propósito de obtener el mejor resultado posible.

Se eligieron las integrales difusas de Sugeno y Choquet como operador de integración del método debido a que usan medidas difusas, lo cual permite estimar de una forma flexible la relevancia de cada red gracias a sus propiedades de adición. Para estimar las densidades difusas se utilizó un sistema difuso tipo-2, lo cual ayudó a modelar la incertidumbre de la función de desempeño de cada red neuronal.

El método desarrollado se aplica después de la simulación de las redes neuronales, por lo que en ese momento los valores de simulación $S(X_i)$ son la única información disponible para determinar el desempeño D_i de cada una de las redes neuronales RN_i .

En esta sección se explicará de una forma general cada bloque del método ilustrado en la *Figura 3.1*, tomando en cuenta que para implementarlo, los bloques pueden adaptarse a cada aplicación particular con el propósito de un mejor resultado.

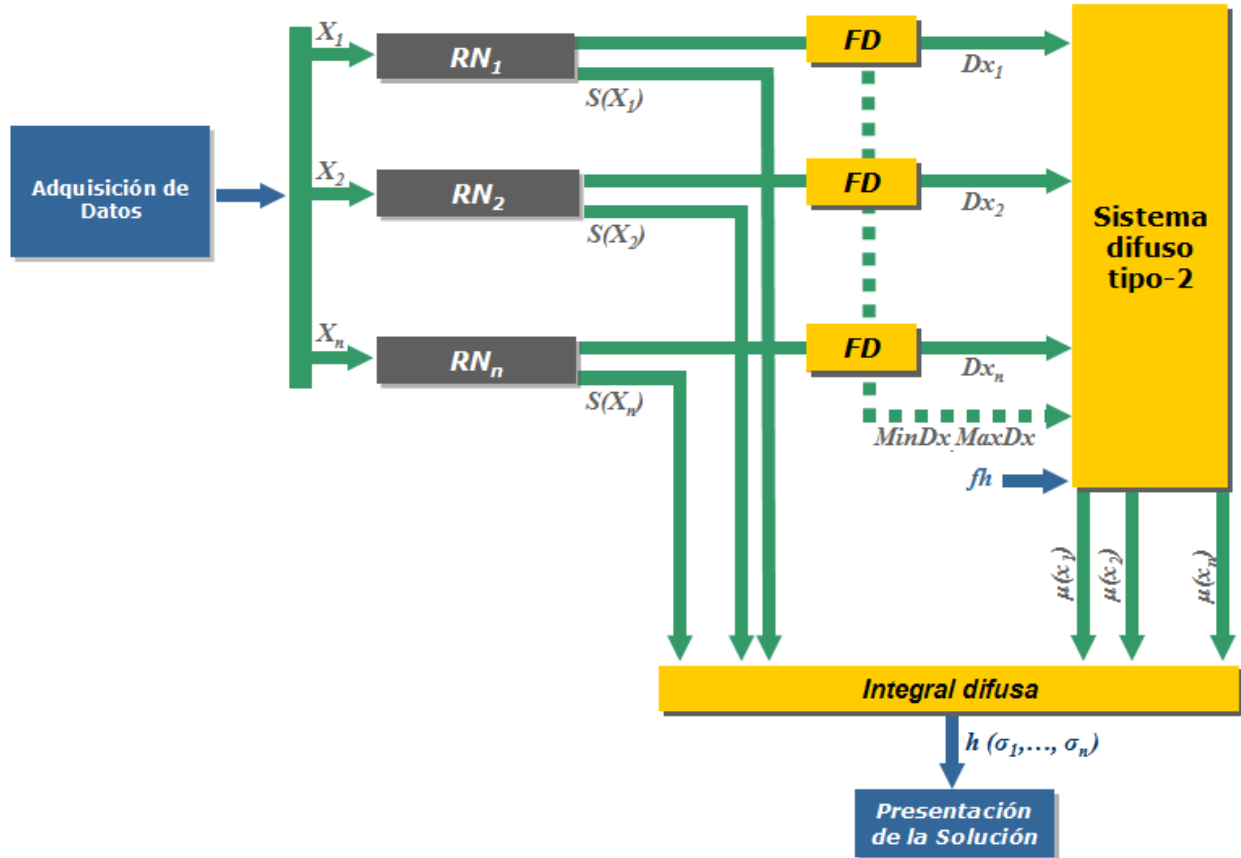


Figura 3.1 Esquema del método generalizado de integración de respuestas

3.1 Redes Neuronales y datos de entrenamiento

El bloque de adquisición de datos debe contener los procedimientos necesarios para obtener los datos de entrenamiento, de manera que las redes neuronales puedan entrenarse con algún algoritmo de entrenamiento supervisado.

3.2 Función de desempeño

El bloque llamado función de desempeño FD es una función que determina un valor en el intervalo $[0,1]$ para cada RN_i , y la forma de obtenerse varía según la naturaleza de la aplicación.

Si la red neuronal es un clasificador, la simulación $S(X_i)$ es por lo general un vector con valores en el intervalo $[0,1]$, en donde la posición del vector con el mayor valor es el elemento elegido.

Por lo que un buen clasificador debe tener por lo menos un valor cercano a 1. Entonces la función de desempeño se puede obtener calculando el valor máximo de $S(X_i)$ con la ecuación (21).

$$D_i = \max(S(X_i)) \quad (21)$$

Si la red neuronal es un predictor, la simulación $S(X_i)$ es la predicción misma por lo que su valor puede estar en cualquier intervalo y no puede tomarse como parámetro por si solo para estimar el desempeño de la red neuronal. Para estimar el desempeño de cada predictor se pueden comparar los resultados de simulación en el conjunto de datos de entrenamiento, calculando el promedio del error en todos los puntos del conjunto de datos de entrenamiento y tomando el complemento de este error como el valor de desempeño de cada predictor con la ecuación (22); de esta forma una red neuronal con un menor error en el modelado de los datos de entrenamiento tendrá mayor nivel de desempeño.

$$D_i = 1 - e_i \quad (22)$$

Donde e es el promedio de los errores absolutos o relativos para la simulación de todos datos de entrenamiento.

3.3 Sistema de inferencia difuso tipo dos por intervalos para estimar densidades difusas

De acuerdo a las definiciones de integrales difusas, la relevancia de las fuentes de información a combinar puede calificarse con las llamadas densidades difusas determinadas por un experto. Se podría pensar que ese experto pudiera el nivel de desempeño D_i descrito en el punto anterior, pero analizando los posibles valores que se obtendrían con dicha función, en muchos casos éstos no reflejan una gran diferencia en la relevancia de las redes neuronales, cuando los resultados finales sí varían en gran medida. Por esta razón es necesario enfatizar esta diferencia y un sistema difuso finalmente sería el responsable de estimar las densidades difusas a partir de los niveles de desempeño de las redes neuronales.

3.3.1 Variables difusas

Ya que el método generalizado está diseñado para integrar n redes neuronales, entonces el sistema de inferencia difuso también debe ser capaz de adaptarse a n cantidad de entradas y salidas.

3.3.1.1 Variables de entrada

El sistema difuso debe tener una variable entrada $\mathbf{Dx}_i$ que indique el desempeño de cada módulo o miembro en el sistema multi-red con n nodos.

Cada variable de entrada consta de tres funciones de membresía iguales: bajo, medio y alto, que deben ser proporcionales al rango de valores a analizar, por lo que sus parámetros también deben adaptarse dependiendo de cada aplicación en particular.

Se eligieron siempre funciones de membresía tipo gaussianas ya que sus parámetros se adaptan mejor a estos propósitos. Primeramente se deben calcular los valores para los centros de las funciones de membresía tipo gaussianas con las ecuaciones (23)(24)(25).

$$Min_{Dx} = \min(Dx_1, Dx_2, \dots, Dx_n) \quad (23)$$

$$Max_{Dx} = \max(Dx_1, Dx_2, \dots, Dx_n) \quad (24)$$

$$Med_{Dx} = (Min_{Dx} + Max_{Dx}) / 2 \quad (25)$$

Con los valores encontrados las funciones de membresía para las 4 variables de entrada están dadas por (26) (27) (28).

$$\mu(bajo) = e^{\frac{-(x - Min_{Dx})^2}{2(\sigma)^2}} \quad (26)$$

$$\mu(alto) = e^{\frac{-(x - Max_{Dx})^2}{2(\sigma)^2}} \quad (27)$$

$$\mu(medio) = e^{\frac{-(x - Med_{Dx})^2}{2(\sigma)^2}} \quad (28)$$

Para calcular un valor para la amplitud σ de la función gaussiana proporcional al rango de cada aplicación se puede calcular con (29).

$$\sigma = (Max_{Dx} - Min_{Dx}) / 8 \quad (29)$$

3.3.1.2 Variables de salida

Las variables de salida son las densidades difusas para los n nodos del sistema multi-red, por lo que de acuerdo a la teoría de medidas difusas, siempre deberán estar en el rango (0,1). Cada

variable consta de tres funciones de membresía: baja, media y alta, las cuales se definen por (30)(31)(32).

$$\mu_{baja} = e^{\frac{-(x-0)^2}{2\sigma^2}} \quad (30)$$

$$\mu_{media} = e^{\frac{-(x-0.5)^2}{2\sigma^2}} \quad (31)$$

$$\mu_{alta} = e^{\frac{-(x-1)^2}{2\sigma^2}} \quad (32)$$

donde $\sigma = 1/8$

3.3.1.3 Huellas de incertidumbre (FOU)

Para determinar el ancho de las FOU también se debe tomar en cuenta el rango de valores para las variables según la aplicación. Para este propósito se puede aplicar la ecuación (33) donde la huella se calcula multiplicando la mitad el ancho del rango de niveles de desempeño D_i por un factor variable fh que debe estar en $[0,1]$, de esta forma se pueden hacer pruebas con valores para la FOU con un mínimo de 0, lo que sería equivalente a un sistema difuso tipo-1 y un máximo de la mitad del rango de las variables, y de esta forma analizar su comportamiento respecto a cada aplicación particular.

$$FOU = (Max_{Dx} - Min_{Dx}) * fh / 2 \quad (33)$$

Donde fh debe estar en $[0,1]$.

La FOU se agrega a las funciones de membresía de forma simétrica.

Tomando en cuenta lo anterior, las variables de entrada y salida del sistema difuso con un factor=0.15 para la FOU, podrían graficarse como se ve en la *Figura 3.2*.

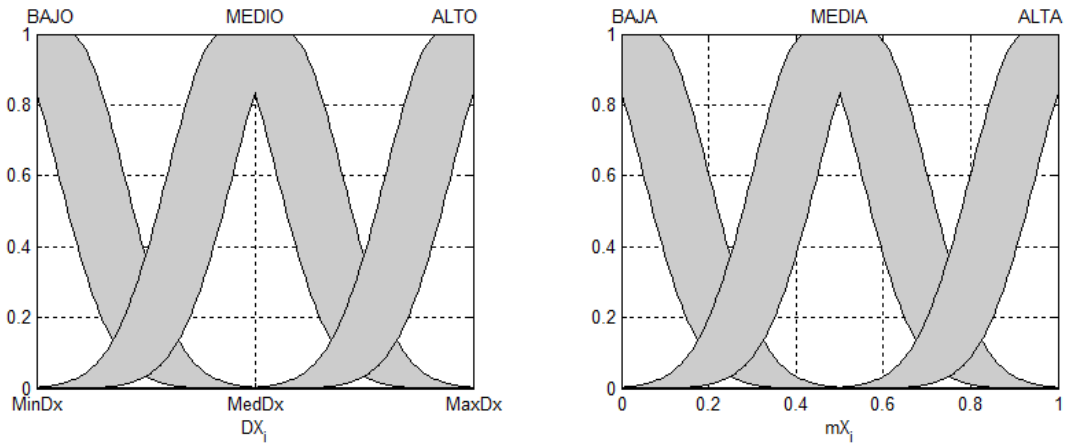


Figura 3.2 Variables de entrada para el FIS2 con un factor=0.5 para la FOU

3.3.2 Reglas difusas

Las reglas del sistema difuso son repetitivas para cada variable Dx_i , ya que el objetivo es enfatizar la diferencia de niveles de relevancia para cada uno de los n nodos del sistema neuronal, estimando n densidades difusas.

Por cada nodo se debe repetir el siguiente conjunto de 3 reglas:

1. Si Dx_i es ALTO entonces mX_i es ALTA
2. Si Dx_i es MEDIO entonces mX_i es MEDIA
3. Si Dx_i es BAJO entonces mX_i es BAJA

Las variables de entrada Dx_i están dadas en el rango $[MinDx, MaxDx]$ mientras que las variables mX_i están dadas en el rango $[0,1]$, lo que permite el mapeo entre las dos escalas de forma natural dentro del sistema difuso. Las reglas se simplifican aún más porque Dx_i siempre será menor o igual a $MaxDx$, entonces para que las salidas reflejen el nivel de relevancia de cada nodo en la escala $[0,1]$ solo son necesarias las siguientes afirmaciones:

- 1) Cuando el nivel de desempeño Dx_i del nodo i es muy cercano al nivel de desempeño máximo $MaxDx$ se estima una densidad difusa mX_i ALTA.

- 2) Cuando el nivel de desempeño Dx_i del nodo i es muy cercano al valor medio entre el valor de desempeño mínimo $MinDx$ y el máximo $MaxDx$ se estima una densidad difusa mX_i MEDIA.
- 3) Cuando el nivel de desempeño Dx_i del nodo i es muy cercano al nivel de desempeño mínimo $MinDx$ se estima una densidad difusa mX_i BAJA.

3.4 Cálculo de la integral difusa

Una vez estimadas las densidades difusas $\mu(X_i)$ se procede con el cálculo de la integral difusa de cada elemento de los n vectores de simulación $S(X_i)$, como se puede ver de una manera informal en la *Figura 3.3*. El vector resultante o simulación final Sf dará la integración de respuestas del sistema neuronal como una fusión de módulos o una combinación de miembros según los enfoques modular o ensemble respectivamente.

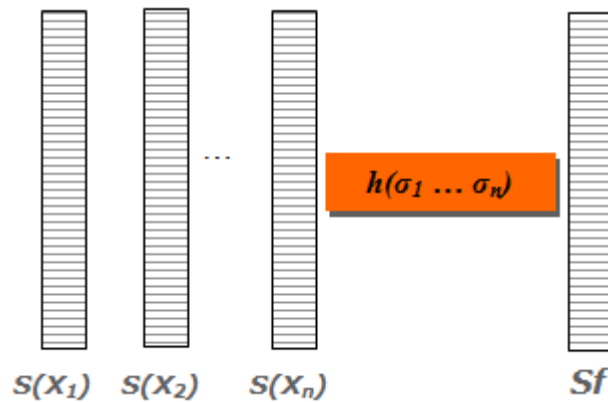


Figura 3.3 Esquema de la integración de vectores de simulación

Este bloque de cálculos se implementa exactamente igual para el enfoque modular o ensembe, lo mismo que para la integración de redes del tipo clasificadores o predictores.

3.5 Presentación de la solución

En este paso del método, lo que tenemos es un vector de simulación único, tal como el que se hubiera obtenido si se hubiera usado una red neuronal monolítica en lugar de un sistema multi-red. Lo que sigue es presentar la solución de acuerdo al tipo de aplicación:

Si el sistema neuronal es un clasificador, es decir una aplicación para reconocimiento de patrones, lo que esperamos como solución al problema es la identificación del elemento de entrada respecto a la posición de los elementos en el conjunto de entrenamiento. La solución del problema es entonces el elemento identificado o reconocido, el cual es el elemento j del vector de entrenamiento con el mismo índice del elemento con el mayor mas grande en el vector de simulación y se puede representar con la ecuación (34).

$$S=E(j), \text{ donde } Sf_j=\max(Sf) \quad (34)$$

Si el sistema neuronal es un predictor, es decir una aplicación para proyección de series de tiempo o funciones, lo que esperamos como solución al problema son los valores correspondientes a los puntos que se requieran predecir, los cuales se obtienen directamente del vector de simulación final Sf .

4 Aplicación 1: Reconocimiento de imágenes

En esta sección se describe una aplicación de reconocimiento de imágenes basada en el método generalizado propuesto en este trabajo de tesis [41]. Para explicar cada una de sus partes se hace referencia a cada bloque del método descrito en la sección 3 y la forma en que se pueden adaptar para esta aplicación particular. Para efectos de comparación de resultados se eligió la base de datos de imágenes ORL (Olivetty Research Laboratory) [2], la cual es usada comúnmente para medir la eficiencia de los sistemas de reconocimiento de imágenes.

La estructura del sistema neuronal es de tipo modular, donde cada módulo es una red neuronal que se entrena con una parte de la imagen [32] [34] [36].

4.1 Adquisición de datos

Este bloque consiste de dos procesos, primero se hace una detección de bordes de cada una de las imágenes de entrenamiento [45] [47] y luego una partición de las mismas para separarlas por módulos.

4.1.1 Detección de bordes con un sistema difuso

El conjunto de datos de entrada contienen solo la base de datos de imágenes, es decir, las imágenes del conjunto de entrenamiento sin modificaciones. Se describe el desarrollo y la comparación de tres métodos para detección de bordes: Sobel, un sistema de inferencia difuso tipo-1 (FIS1) [44] y un sistema de inferencia difuso tipo-2 (FIS2) [42]. A diferencia de los métodos tradicionales para detección de bordes e incluso en algunas extensiones de los mismos [3] [8] [10] [12] [44] [46] [50] [56], en los métodos difusos no solo se obtienen los bordes de la

imagen, sino que los valores de cada pixel se normalizan para lograr un entrenamiento mas rápido de las redes neuronales.

El operador Sobel aplicado sobre una imagen digital en escala de grises, calcula el gradiente de la intensidad de brillo de cada pixel dando la dirección del mayor incremento posible de negro a blanco, además calcula el monto de cambio de esa dirección [15] [58].

4.2 Cálculo del gradiente y magnitud del gradiente

El gradiente y la magnitud del gradiente para cada pixel es calculado aplicando el operador de Sobel como se muestra en las ecuaciones (35) y (36).

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (35)$$

$$Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (36)$$

Donde $Sobel_y$ y $Sobel_x$ son los operadores de Sobel a lo largo de los ejes x y y.

$$g_x = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{x,i,j} * I_{r+i-2,c+j-2} \quad (37)$$

$$g_y = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{y,i,j} * I_{r+i-2,c+j-2} \quad (38)$$

$$g = \sqrt{g_x^2 + g_y^2} \quad (39)$$

Donde g_x y g_y son los gradientes con respecto al eje x y al eje y respectivamente calculados con las ecuaciones (37) y (38) [21]. La magnitud del gradiente g es la imagen con los bordes y se calcula con (39) [18] [37].

4.2.1 Sistema de inferencia difuso tipo-1 para detección de bordes

El sistema de inferencia difuso tipo-1 (FIS1) para detección de bordes necesita como entradas las imágenes generadas por los operadores de Sobel.

4.2.1.1 Entradas para el FIS tipo-1

Para el sistema de inferencia difuso tipo-1, se usaron 3 entradas, 2 de ellas son los gradientes respecto al eje x g_x y el eje y g_y a las cuales llamaremos DH y DV respectivamente.

La tercera variable M es la imagen después de aplicar un filtro pasa-bajas HMF mostrado en la ecuación (40); este filtro permite detectar pixeles en la imagen pertenecientes a regiones donde el promedio de niveles de gris es muy bajo. Estas regiones son mayormente afectadas por ruido [44].

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (40)$$

El objetivo es diseñar un sistema, que detecte bordes en regiones de bajo contraste, pero sin favorecer la detección de bordes falsos por efectos de ruido.

Por lo tanto las variables de entrada para el sistema difuso son definidas por las expresiones

(41)-(43):

$$DH = Sobel_x \otimes I, \quad (41)$$

$$DV = Sobel_y \otimes I, \quad (42)$$

$$M = hMF \otimes I, \quad (43)$$

Donde $\otimes$ es el operador de convolución, e I la imagen a la que se le aplicará el filtro.

4.2.1.2 Variables difusas

Para todas las variables difusas, las funciones de membresía son de tipo gaussiana. De acuerdo a las pruebas realizadas, los valores de DH y DV , oscilan entre 0 y 500 como se puede ver en la *Figura 4.1* y la *Figura 4.2*; estos valores corresponden a la magnitud de cambio en cada dirección. La función de membresía bajo corresponde a magnitudes de cambio cercanas a 0, la función de membresía medio a magnitudes de cambio cercanas a 230, y la función de membresía alto a magnitudes de cambio cercanas a 365. Es posible encontrar algunas magnitudes de cambio mayores a 365, pero por ser tan pocas fue mejor ignorarlas con el propósito de considerar más bordes aun que la magnitud de cambio no sea de las mas altas.

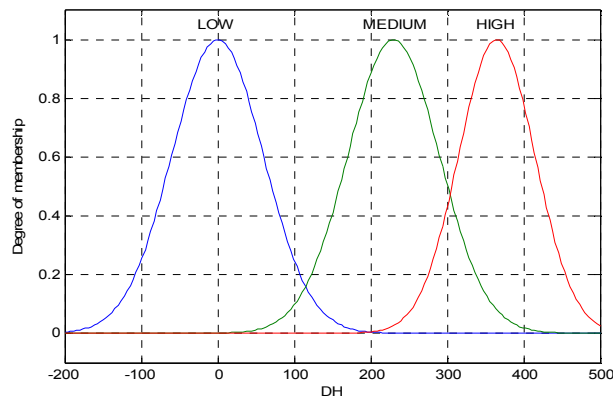


Figura 4.1 Variable de entrada DH para el detector de bordes FIS1

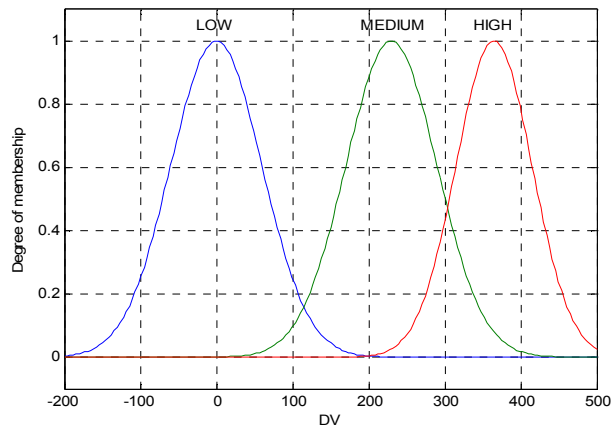


Figura 4.2 Variable de entrada DV para el detector de bordes FIS1

En el caso de la variable M, las pruebas arrojan valores en el rango $[0,255]$, así que las funciones de membresía fueron ajustadas a ese rango, como se puede apreciar en la *Figura 4.3*, en donde las funciones de membresía son de tipo gaussianas [1].

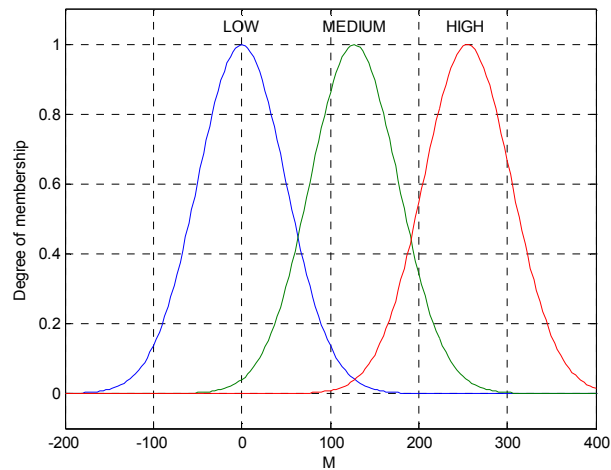


Figura 4.3 Variable M con funciones de membresía tipo-1.

En la *Figura 4.4*, se muestra la variables de salida “edges”, la cual fue ajustada al rango $[-4.5,1.5]$, ya que se requiere que la imagen sea normalizada para ser usada como conjunto de entrenamiento para redes neuronales.

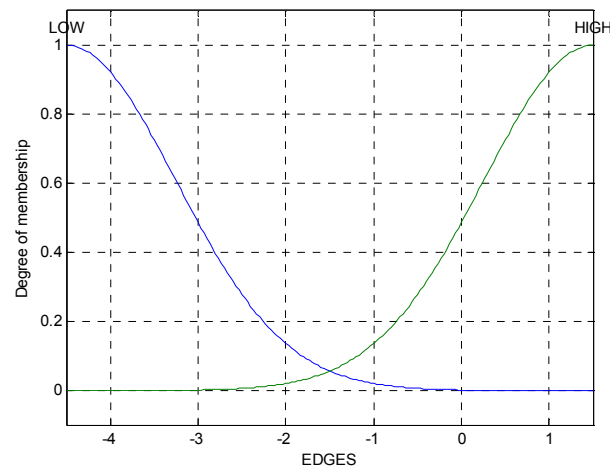


Figura 4.4 Variable “edges” con funciones de membresía tipo-1.

4.2.1.3 Reglas de inferencia difusas

Las cinco reglas que permiten la evaluación de las variables de entrada, son las siguientes:

- 1.- Si (DH es BAJO) y (DV es BAJO) entonces (BORDES es ALTO)
- 2.- Si (DH es MEDIO) y (DV es MEDIO) entonces (BORDES es BAJO)
- 3.- Si (DH es ALTO) y (DV es ALTO) entonces (BORDES es BAJO)
- 4.- Si (M es BAJO) y (DV es MEDIO) entonces (BORDES es BAJO)
- 5.- Si (M es BAJO) y (DH es MEDIO) entonces (BORDES es BAJO)

Según estas reglas la imagen de salida despliega los bordes en tonos de gris cercanos a 0 o negro (LOW), mientras que el fondo se despliega en tonos cercanos a 255 o blanco (HIGH).

Las superficies de solución del sistema difuso se muestra en la *Figura 4.5*, *Figura 4.6* y *Figura 4.7*. Las gráficas describen valores altos para bordes cuando las magnitudes de DH y DV están cercanos a 0, lo que significa que los tonos mas claros o el fondo de la imagen, y tonos bajos para los bordes cuando las magnitudes de DH y DV están entre 200 y 400, lo que significa tonos oscuros o los bordes de la imagen.

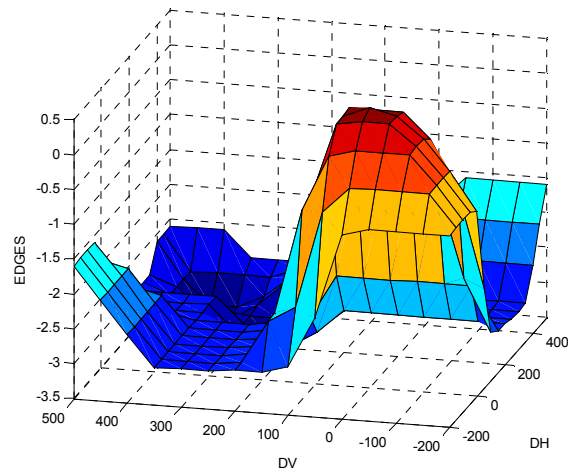


Figura 4.5 Superficie de solución para DV, DH y EDGES en el detector de bordes FIS1.

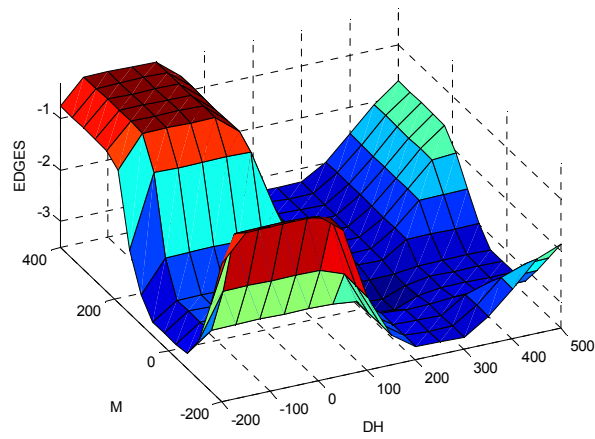


Figura 4.6 Superficie de solución para M, DH y EDGES en el detector de bordes FIS1.

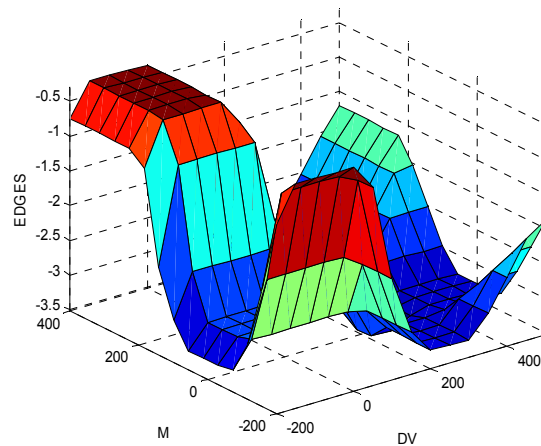


Figura 4.7 Superficie de solución para M , DV y $EDGES$ con funciones de membresía tipo-1

4.2.2 Sistema de inferencia difuso tipo-2 por intervalos para detección de bordes

El sistema difuso tipo-2 por intervalos (FIS2) para detección de bordes tiene la estructura, parámetros y reglas basados en el FIS1, lo que se hizo para comparar los resultados de ambos métodos [4] [6] [7].

El ancho de la huella de incertidumbre (FOU) fue elegida igual a 0.2 por arrojar los mejores resultados después de muchos experimentos. La *Figura 4.8* muestra las variables para el FIS2 en esta aplicación [35].

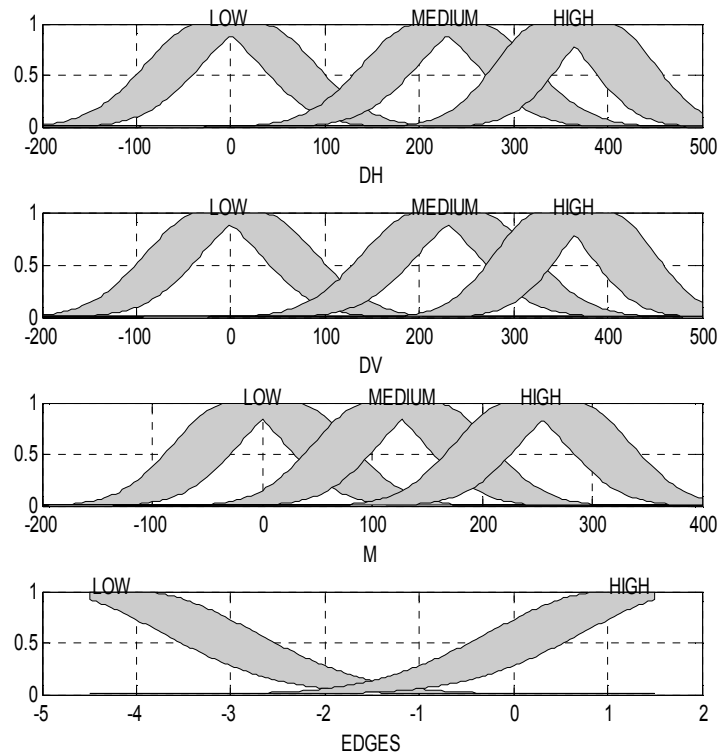


Figura 4.8 Funciones de membresía para las variables *DH*, *DV*, *M* y *EDGES* para el sistema difuso tipo-2 por intervalos usado como detector de bordes.

En la *Figura 4.9* y la *Figura 4.10* se muestran superficies de solución para el FIS2 en esta aplicación.

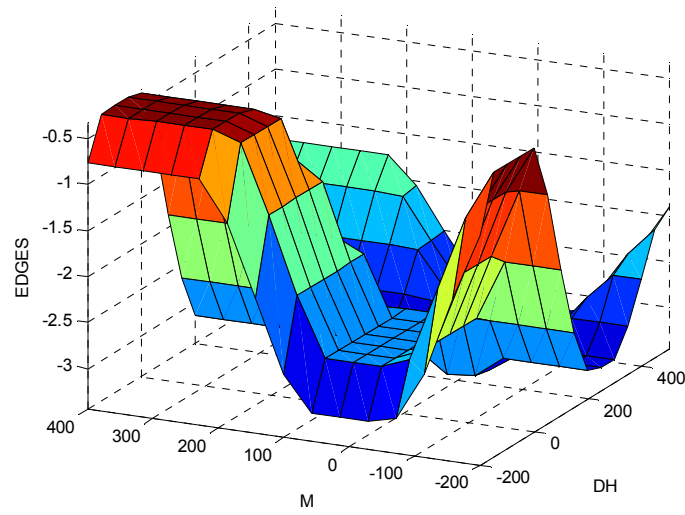


Figura 4.9 Superficie de solución para las variables DH , DV , M y $EDGES$ para el sistema difuso tipo-1 usado como detector de bordes.

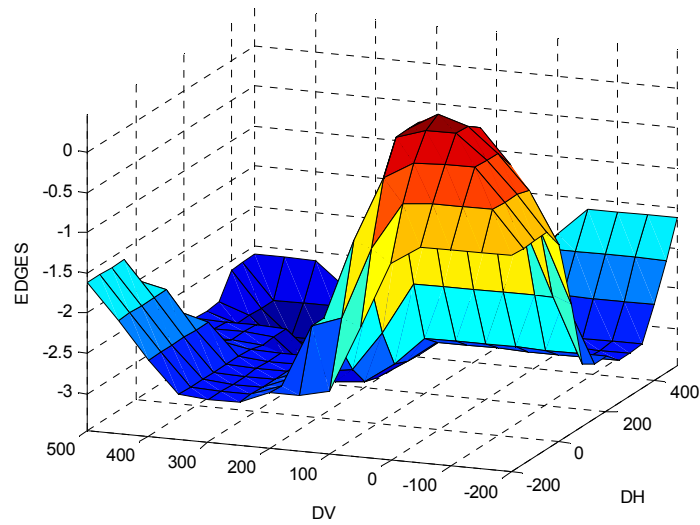


Figura 4.10 Superficie de solución para las variables DH , DV , M y $EDGES$ para el sistema difuso tipo-2 por intervalos usado como detector de bordes.

4.2.3 Imágenes generadas con operadores de Sobel

Se usará como ejemplo la primera muestra de la persona numero 1 de la base de datos ORL [2], mostrada en la *Figura 4.11* donde la escala de gris esta en el rango [0,255].

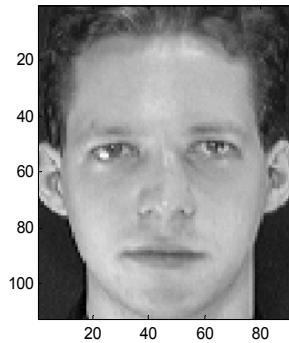


Figura 4.11 Imagen original 1.pgm

En la *Figura 4.12* y la *Figura 4.13* se muestran las imágenes resultantes del cálculo de los gradientes g_x y g_y respectivamente

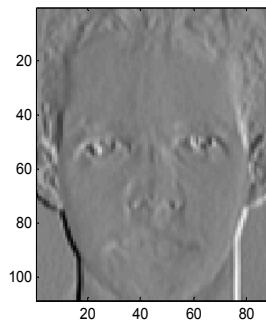


Figura 4.12 Imagen generada por g_x

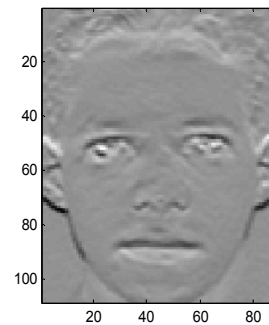


Figura 4.13 Imagen generada por g_y

Es importante analizar los rangos de valores generados por g_x y g_y , por lo que en la Tabla 4-I se muestra un ejemplo de los valores máximos y mínimos obtenidos para la imágenes que se usa como ejemplo, 1.pgm. Como podemos ver, el rango de valores es muy amplio, lo que sería un problema si estos datos se utilizaran como conjunto de entrenamiento para una red neuronal, si el

método supervisado usa como patrón (target) la matriz identidad, donde el valor 0 significa un elemento no reconocido y el valor 1 un elemento reconocido.

TABLA 4-I: VALORES MÁXIMOS Y MÍNIMOS PARA POR GY Y GY Y G .

	<i>l.pgm</i>	g_x	g_y	g
Tono mínimo	11	-725	-778	0
Tono máximo	234	738	494	792

La imagen obtenida con los bordes es ligeramente mas pequeña que la original, porque el resultado de la operación convolución es una matriz central donde existen todos los vecinos del cada pixel, lo que no sucede en los límites de la imagen. Por lo que en el ejemplo, cada imagen original de 112x92 pixeles se reduce a 108x88 pixeles.

4.2.4 Detección de bordes con el método de Sobel tradicional y el método de Sobel extendido con sistemas difusos tipo-1 y tipo-2.

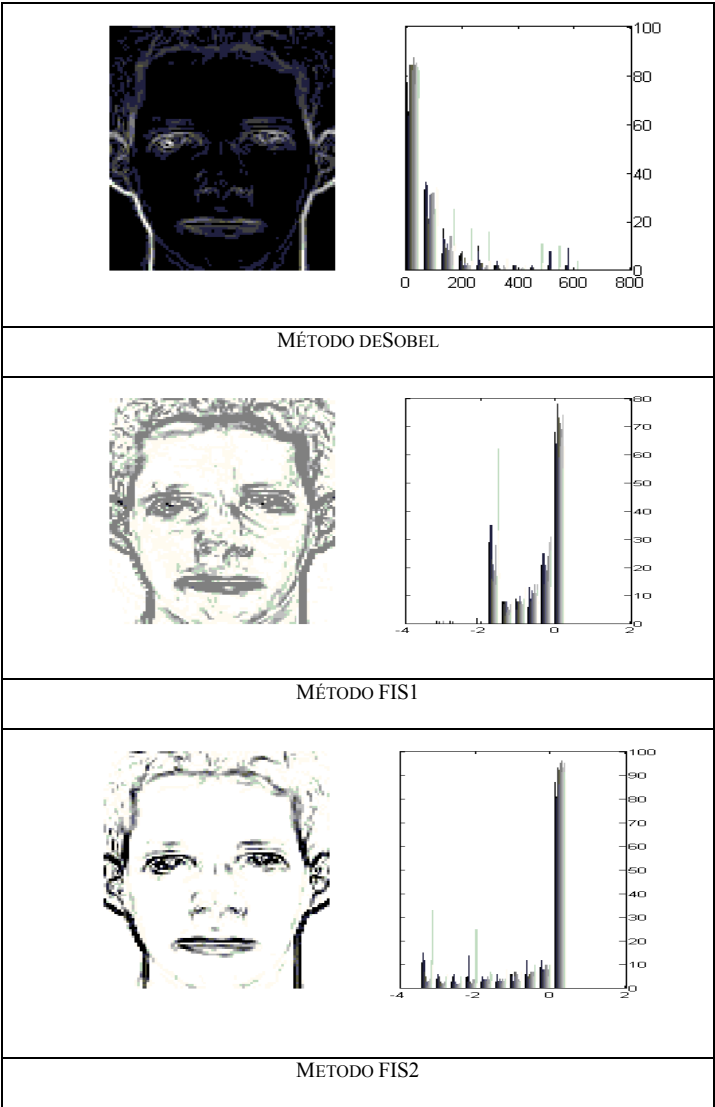
Las reglas de inferencia y funciones de membresía permiten la normalización de los valores de tonos de gris al mismo tiempo de la detección de bordes, por lo que se eligió el rango $[-4.5, 1.5]$, lo cual no es posible usando el método de magnitud de gradiente. Esto permite que el promedio de los valores de pixeles de cada imagen sea cercana a cero y su desviación Standard cercana a 1, lo cual es una gran ventaja si la imagen se usa como conjunto de entrenamiento para redes neuronales.

Otra ventaja del detector de bordes difuso, es que normaliza los valores de todas las imágenes a un rango homogéneo, independientemente de la luz, contraste o tono de fondo de cada una.

Con el propósito de obtener una comparación objetiva de las imágenes, fueron contruidos histogramas de frecuencia correspondientes a los resultados de detección de bordes con los métodos de Sobel, FIS1 y FIS2, los cuales se muestran en la Tabla 4-II.

Los histogramas muestran en el eje x la escala de grises obtenida con cada método y en el eje y la frecuencia en que aparece cada tono de la escala en la imagen [51] [57].

TABLA 4-II: HISTOGRAMAS DE LAS IMÁGENES RESUTANTES DE LOS DETECTORES DE BORDES DE SOBEL, FIS1 Y FIS2.



Los resultados tanto del FIS1 y el FIS2 son mejores porque la salida se encuentra normalizada a los parámetros ideales para ser datos de entrenamiento de redes neuronales. Comparando visualmente las imágenes, se nota claramente que el detector de bordes FIS2 es mejor que el FIS1.

4.3 Sistema Neuronal Modular

La aplicación para reconocimiento de imágenes presentada en este trabajo, consta de una red neuronal modular de tres módulos, donde cada módulo es un experto en clasificar y reconocer un tercio de cada una de las imágenes de entrenamiento y pruebas [5] [23].

Se hicieron diversas pruebas aplicando el método generalizado que se presenta en este trabajo, en algunas de ellas se optimizó el sistema difuso para obtener mejores resultados, como es el caso que a continuación se presenta, mismo que ya fue publicado en revista de prestigio internacional.

4.3.1 La estructura modular

La estructura modular está diseñada para una base de datos de imágenes como la base de datos de imágenes ORL [2], pero no está limitada a estos datos. Para medir el porcentaje de reconocimiento de forma objetiva, las redes neuronales fueron entrenadas con el conjunto de imágenes en orden aleatorio, este proceso es llamado permutación aleatoria [38][39][40].

La estructura modulara consiste de tres redes neuronales monolíticas de tipo feed-forward, y cada una fue entrenada con un método supervisado con las primeras siete muestras de los 40 individuos de la base de datos ORL [26].

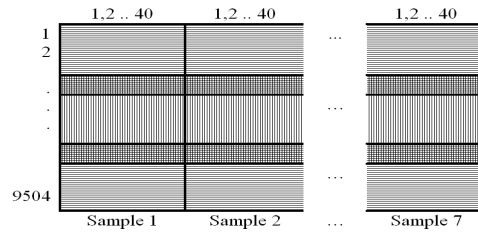


Figura 4.14 Datos de entrenamiento: Siete muestras de cada persona.

El vector de bordes de cada imagen se acumulado en una matriz, como se puede ver en la *Figura 4.14*. La matriz de imágenes completa es dividida en tres partes, cada módulo es entrenado con la parte correspondiente, con algunos renglones de traslape. El patrón para el método de entrenamiento supervisado es una matriz identidad de dimensiones 40 x 40 para cada muestra, haciendo una matriz de dimensiones (40 x 40 x 7), como se ve en la *Figura 4.15* [48].

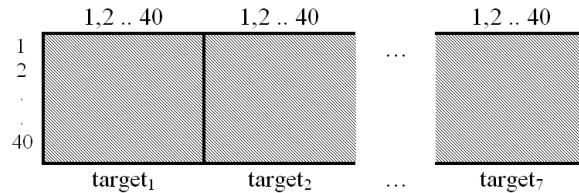


Figura 4.15 Patrón: Una matriz identidad de 40x40 para cada muestra.

Cada red neuronal monolítica tiene la misma estructura de la *Figura 4.16* y fue entrenada bajo las mismas condiciones usando la Toolbox de redes neuronales de Matlab:

1. Dos capas ocultas con 200 neuronas y funciones de transferencia tangente sigmoideal.
2. La capa del salida con 40 neuronas y funciones de transferencia lineales.
3. El algoritmo de entrenamiento de tipo retropopagación del error es gradiente descendente con momentum y tasa aprendizaje adaptiva (traingdx).

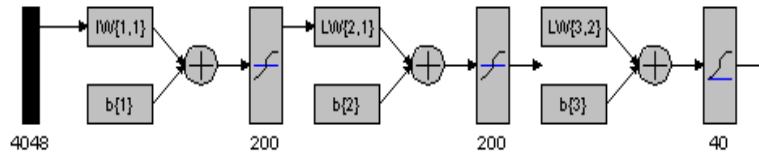


Figura 4.16 Estructura de cada red neuronal monolítica en el sistema modular.

4.3.2 Entrenamiento de módulos

El siguiente segmento de código implementado para entrenar cada módulo, donde newff es la función de la toolbox de matlab utilizada para crear la estructura de cada red neuronal [1].

```
Layer1=200; layer2=200; layer3=40;

Net=newff(minmax(p), [layer1,layer2,layer3], {'tansig', 'tansig', 'logsig'},

'traingdx');

Net.trainparam.goal=1e-5;

Net.trainparam.epochs=1000;
```

4.3.3 Simulación de módulos

Se utilizó Matlab para implementar la simulación de cada módulo con las 400 imágenes de la base de datos ORL, construyendo así una matriz con los resultados de simulación de cada módulo, como se ve en la *Figura 4.17*. Estas matrices fueron almacenadas en un archivo para ser analizadas posteriormente con el algoritmo de combinación de módulos.

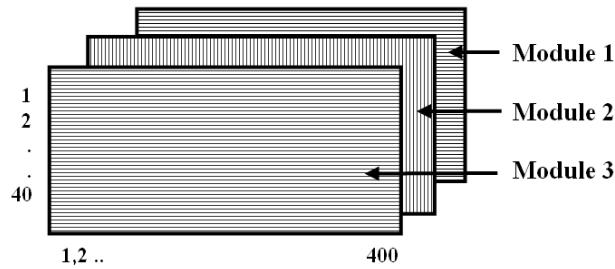


Figura 4.17 Esquema de las matrices de simulación para los tres módulos.

Cada vez que se simula una imagen en las redes neuronales, el resultado es un vector con los valores de activación para cada posición de las 40 imágenes, de esta forma, la posición con un valor mas cercano a uno corresponde a la imagen que mas se asemeja a la utilizada para el entrenamiento. Si las redes están bien entrenadas, al simular con las imágenes del conjunto de entrenamiento, el reconocimiento es del 100%, el reto son las imágenes fuera de este conjunto, ya que en algunos casos los valores de simulación resultan muy pequeños en todas las posiciones, lo que dificulta distinguir cual de ellas es la que mas se acerca al patrón, por esta razón es muy importante un buen criterio para la combinación de respuestas de las redes en la estructura multi-red.

4.3.4 Función de desempeño de los módulos

La manera de determinar la función de desempeño de cada modulo fue como se describe en la sección 3.2, obteniendo el valor máximo de activación de cada módulo al simular la muestra a clasificar.

4.4 Estimación de densidades difusas con un sistema difuso tipo-2 por intervalos.

Las variables de entrada m_i para el sistema difuso corresponden a los valores de simulación a combinar, y los valores max_i corresponden a la función de desempeño de cada módulo. Las variables de salida d_i corresponden a las densidades difusas que serán parámetro para la fusión de módulos con la integral de Sugeno.

Se hicieron pruebas con diferentes anchos de FOU, incluyendo FOU=0, lo que equivale a un sistema difuso tipo-1. En la *Figura 4.18* se muestran las variables difusas con FOU=0.2

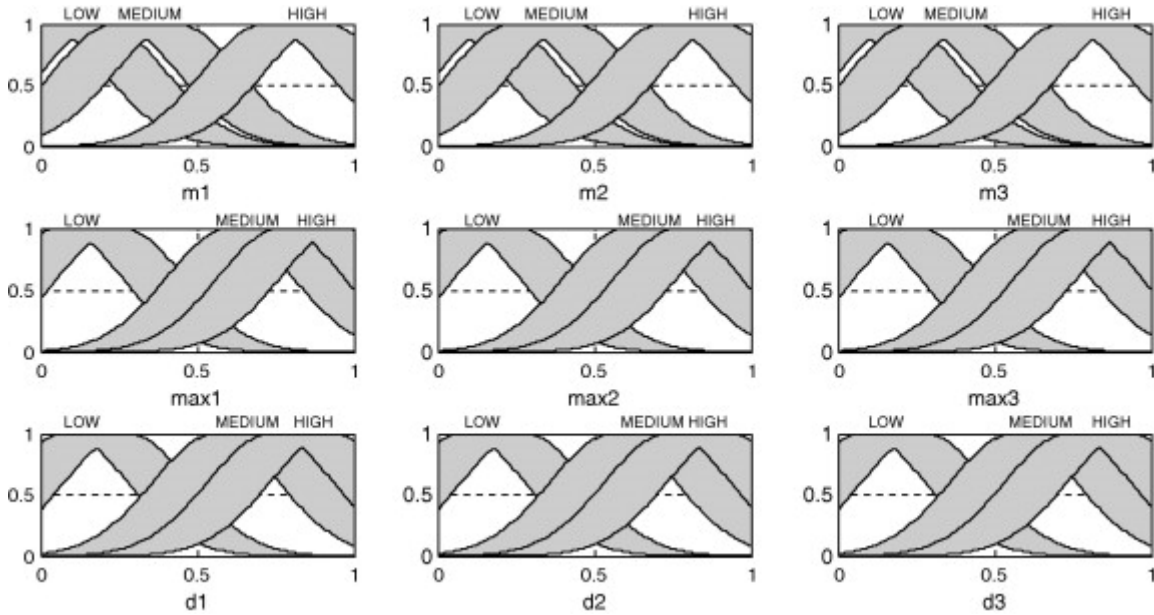


Figura 4.18 Variables difusas tipo-2 por intervalos para la estimación de densidades difusas

Las reglas de inferencia estiman densidades difusas cercanas a 1 cuando el valor a combinar es muy cercano a 1, cercanas a 0.5 cuando el valor a combinar es muy cercano a 0.5 y de 0 cuando el valor a combinar es muy cercano a 0. Además a diferencia de la propuesta del sistema difuso propuesto como generalizado, se toman en cuenta casos especiales cuando el valor máximo es cercano a 0.5 o a 0.

- 1.If (m1 is LOW) and (max1 is LOW) then (d1 is LOW)
- 2.If (m1 is MEDIUM) and (max1 is MEDIUM) then (d1 is MEDIUM)
- 3.If (m1 is HIGH) and (max1 is HIGH) then (d1 is HIGH)
- 4.If (m2 is LOW) and (max2 is LOW) then (d2 is LOW)
- 5.If (m2 is MEDIUM) and (max2 is MEDIUM) then (d2 is MEDIUM)
- 6.If (m2 is HIGH) and (max2 is HIGH) then (d2 is HIGH)
- 7.If (m3 is LOW) and (max3 is LOW) then (d3 is LOW)
- 8.If (m3 is MEDIUM) and (max3 is MEDIUM) then (d3 is MEDIUM)
- 9.If (m3 is HIGH) and (max3 is HIGH) then (d3 is HIGH)
- 10.If (m1 is LOW) and (max1 is MEDIUM) then (d1 is LOW)
- 11.If (m1 is MEDIUM) and (max1 is HIGH) then (d1 is MEDIUM)
- 12.If (m2 is LOW) and (max2 is MEDIUM) then (d2 is LOW)
- 13.If (m2 is MEDIUM) and (max2 is HIGH) then (d2 is MEDIUM)
- 14.If (m3 is LOW) and (max3 is MEDIUM) then (d3 is LOW)
- 15.If (m3 is MEDIUM) and (max3 is HIGH) then (d3 is MEDIUM)
- 16.If (m1 is LOW) and (max1 is HIGH) then (d1 is LOW)
- 17.If (m2 is LOW) and (max2 is HIGH) then (d2 is LOW)
- 18.If (m3 is LOW) and (max3 is HIGH) then (d3 is LOW)

4.5 Fusión de módulos con la integral de Sugeno

Después del proceso de simulación para una imagen, se tienen tres vectores a fusionar donde cada uno corresponde al resultado de simulación de un módulo. Para obtener un vector único a partir de los tres módulos, la fución se realiza elemento por elemento aplicando la integral de Sugeno con los parámetros obtenidos en la sección anterior.

Considere un ejemplo para la simulación de la muestra número 8 de la persona número 13 de la base de datos ORL.

	m1	m2	m3
1			
2			
3			
:			
13	0.1575	0.0094	0.0247
14			
:			
40			
	max1	max2	max3
	0.1575	0.0286	0.0574

El sistema difuso para estimación de densidades difusas arroja el siguiente resultado:

$$d_1 = 0.3069, \quad d_2 = 0.1788, \quad d_3 = 0.1890$$

Ahora es necesario resolver la el polinomio $f(\lambda) = (1 + 0.3069\lambda)(1 + 0.1788\lambda)(1 + 0.1890\lambda) - (1 + \lambda)$ de acuerdo a la ecuación (3).

$$f(\lambda) = (1 + 0.3069\lambda)(1 + 0.1788\lambda)(1 + 0.1890\lambda) - (1 + \lambda)$$

El comportamiento de la función se puede observar en la *Figura 4.19*.

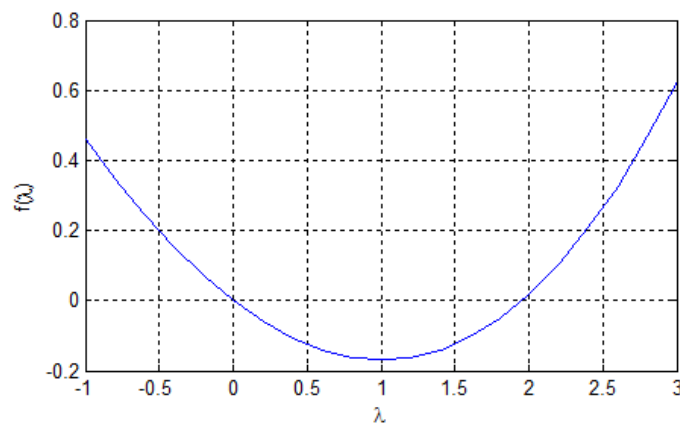


Figura 4.19 Comportamiento de $f(\lambda)$ para el ejemplo propuesto

La solución $\lambda = 1.9492$ para este ejemplo se obtuvo con el método de Newton-Raphson. Las medidas de Sugeno aplicando las formulas recursivas (4) y (5), se muestran en la Tabla 4-III.

TABLA 4-III: VALORES DE DE LAS DENSIDADES DIFUSAS ORDENADAS DESCENDENTEMENTE RESPECTO A X.

x_i	$\sigma(x_i) = \text{trimf}([0 \max(x) \max(x)], x_i)$	$\mu(x_i)$
0.1575	0.9989	0.3069
0.0247	0.0492	0.1890
0.0094	0.0249	0.1788

$$\mu(A_1) = 0.3069$$

$$\mu(A_2) = 0.1890 + 0.3069 + (1.9492)(0.1890)(0.3069) = 0.6090$$

$$\mu(A_3) = 0.1788 + 0.6090 + (1.9492)(0.1788)(0.6090) = 1$$

La integral de Sugeno puede ahora calcularse usando la ecuación (7):

$$h(0.9989, 0.0492, 0.0249) = \max(\min(0.9989, 0.3069), \min(0.0492, 0.6090), \min(0.0249, 1))$$

$$h(0.9989, 0.0492, 0.0249) = \max(0.3089, 0.0492, 0.0249) = 0.3089$$

Como se puede ver en la *Figura 4.20*, la persona número 13 obtuvo el valor mas alto para la integral de Sugeno y es reconocida correctamente.

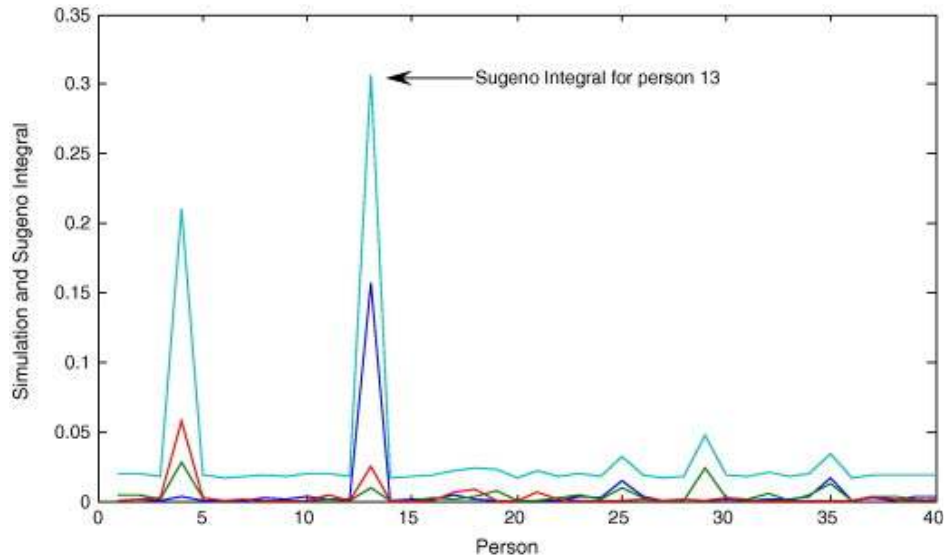


Figura 4.20 Valores de la integral de Sugeno para la persona 13

4.6 Resultados de reconocimiento

Con el propósito de medir los resultados de forma objetiva, se desarrolló un métodos de permutación aleatoria, en el cual se reordenan las muestras de cada persona antes de cada entrenamiento. Se hicieron 5 entrenamientos para cada permutación como se puede ver en la Tabla 4-IV usando el sistema difuso tipo-1 para estimación de densidades difusas, mientras que en la Tabla 4-V se usó el sistema difuso tipo-2 por intervalos para el mismo propósito.

TABLA 4-IV: PORCENTAJES DE RECONOCIMIENTO USANDO EL FIS1 PARA ESTIMAR DENSIDADES DIFUSAS

P	Image Recognition (%)				
	Train 1	Train 2	Train 3	Avg	Max
1	94.00	95.75	94.50	94.75	95.75
2	94.25	94.75	94.25	94.41	94.75
3	94.25	94.25	95.25	94.58	95.25
4	94.00	93.25	93.50	93.58	94.00
5	94.75	94.75	94.00	94.36	94.75
				94.36	95.75

TABLA 4-V: PORCENTAJES DE RECONOCIMIENTO USANDO EL FIS2 PARA ESTIMAR DENSIDADES DIFUSAS

P	Image Recognition (%)				
	Train 1	Train 2	Train 3	Avg	Max
1	97.25	96.25	95.00	96.17	97.25
2	94.75	95.25	95.75	95.25	95.75
3	95.50	97.50	96.00	96.33	97.50
4	95.25	95.00	95.50	95.25	95.50
5	96.50	97.00	96.00	96.50	97.00
				95.90	97.50

Para ver el rendimiento de cada prueba según el sistema difuso utilizado, en la *Figura 4.21* se graficaron los porcentajes de reconocimiento para cada permutación y entrenamiento con los dos sistemas difusos.

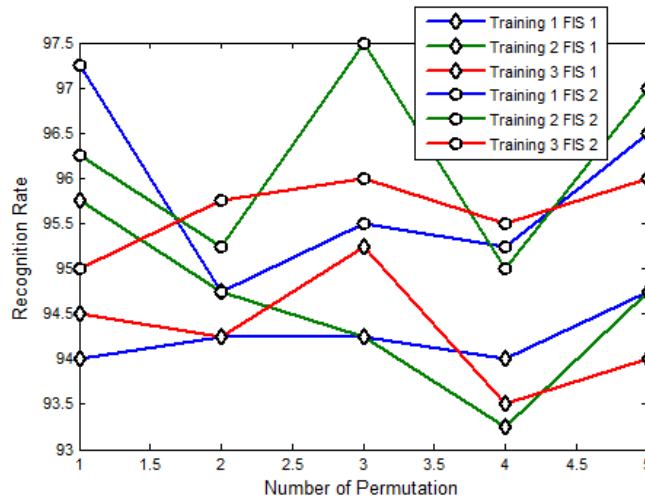


Figura 4.21 Porcentajes de reconocimiento para cada permutación y entrenamiento con los sistemas difusos tipo-1 y tipo-2

En la *Figura 4.22* se muestran los mejores porcentajes de reconocimiento de los dos sistemas. Como se puede ver en esta gráfica, los porcentajes de reconocimiento utilizando el sistema difuso tipo-2 por intervalos es siempre mejor.

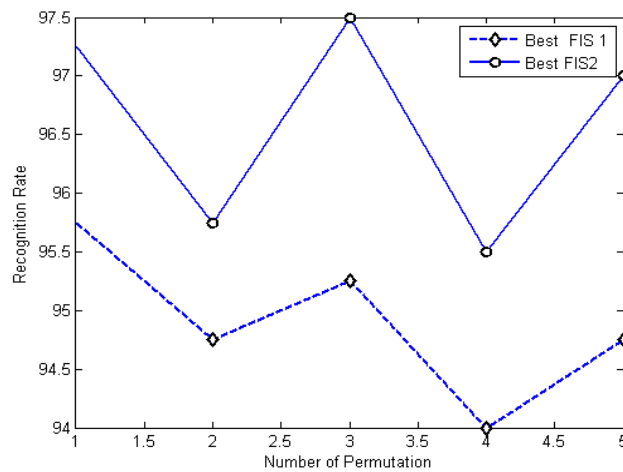


Figura 4.22 Los mejores porcentajes de reconocimiento de los sistemas difusos tipo-1 y tipo-2.

4.6.1.1 Comparación con otros métodos para reconocimiento de imágenes

En la Tabla 4-VI se presentan algunos porcentajes con otros métodos para reconocimiento de imágenes que publicaron sus pruebas usando la base de datos de rostros ORL: Face Recognition with Radial Basis Function(RBF) Neural Networks [19], The technique BBPCA plus LDA (Bidirectional Principal Component Analysis plus Linear Discriminant Analysis) [60], The method Uncertainty Estimation Using Fuzzy Measures for Multiclass Classification (FMMC) [13].

TABLA 4-VI: PORCENTAJES DE RECONOCIMIENTO PARA LOS MÉTODOS: RBF, BDPCALDA, UEFMMC Y T1FL+MNN, IT2 FL+MNN.

Método	Reconocimiento	Año
RBF	98.08	2002
BDPCA LDA	97.10	2006
UEFMMC	97.10	2007
T1FL+MNN	95.75	2007
IT2FL+MNN	97.50	2007

5 Aplicación 2: Predicción de series de tiempo

Se desarrolló una aplicación para predicción de series de tiempo usando el mismo método generalizado presentado en este trabajo, para demostrar que el método es eficiente en una aplicación diferente al reconocimiento de patrones, así que eventualmente con algunas consideraciones en los bloques también podría funcionar con otros tipos de aplicaciones.

5.1 Adquisición de datos

Para esta prueba se usó una serie de tiempo disponible en Matlab, cuyos valores se pueden leer directamente del archivo mgdata.dat. Esta serie es comúnmente utilizada por investigadores en redes neuronales y lógica difusa. Como se explica en la documentación de Matlab [1], la serie fue generada por la ecuación diferencial de Mackey–Glass (44).

$$x(t) = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t) \quad (44)$$

Esta serie es caótica, por lo que su periodo no está definido claramente. Para encontrar los valores de la serie en puntos enteros, se aplicó el método Runge-Kutta para encontrar la solución numérica de la ecuación. Se asume que $x(0)=1.2$, $t=17$, y $x(t)=0$ para $t<0$.

5.1.1 Retrazos en los datos

Para cada valor de salida de la red neuronal es necesario dar r valores anteriores como datos de entrenamiento, esperando que la red prediga el siguiente valor. En este experimento se usaron 4 retrasos r .

$$\begin{aligned} P(1,2..Q) &= T(1,1..(Q-1)) \\ P(2,3..Q) &= T(1,1..(Q-2)) \end{aligned}$$

$$P(3,4..Q) = T(1,1..(Q-3))$$

$$P(4,5..Q) = T(1,1..(Q-4))$$

Donde P son los datos de entrenamiento, T los datos objetivo para el entrenamiento supervisado y Q es la cantidad de puntos usados para la serie.

5.2 Sistema neuronal ensemble

El sistema neuronal ensemble se compone de 4 redes neuronales redundantes como miembros, entrenadas con diferentes algoritmos, con el propósito de obtener diferentes grados de aprendizaje y poder probar el buen funcionamiento del método generalizado.

5.2.1 La estructura ensemble

Se implementó una estructura multi-red tipo ensemble con 4 redes neuronales con la misma estructura como se ve en la *Figura 5.1*, pero entrenadas bajo distintas condiciones usando la Toolbox de redes neuronales de Matlab:

- La capa de entrada con 4 neuronas y una capa oculta de 8 neuronas, con funciones de transferencia tangente sigmoidal.
- La capa del salida con 1 neurona y función de transferencia lineal.

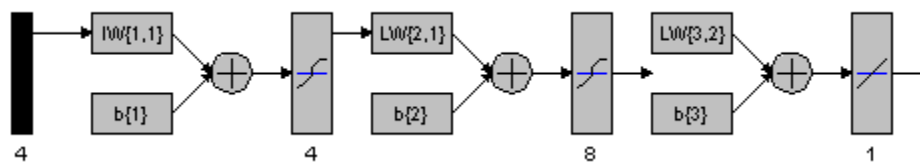


Figura 5.1 Estructura de cada miembro del ensemble

5.2.2 Entrenamiento de miembros

Las redes neuronales se entrenaron con diferentes algoritmos de retropropagación del error:

- Una red con el algoritmo de gradiente descendente con momentum y tasa aprendizaje adaptiva (traingdx).
- Una red con el algoritmo de gradiente descendente con tasa aprendizaje adaptiva (traingda).
- Dos redes con el algoritmo de optimización de Levenberg-Marquardt (trainlm).

Aunque todas las redes se entrenaron con un error meta de 1×10^{-5} y un máximo de 1000 épocas, los resultados varían debido a los algoritmos de entrenamiento y a la aleatoriedad de los pesos iniciales. Este es deseable para la demostración del sistema ensemble, ya que el método para combinar los resultados debe dar diferentes grados de importancia a cada miembro según su rendimiento.

5.2.3 Simulación de miembros y su función de desempeño

El proceso de simulación se realizó por separado con los puntos de la serie que se incluyeron en el conjunto de entrenamiento y con los puntos de la serie que se excluyeron del mismo.

Los resultados de simulación con los datos de entrenamiento se utilizaron para determinar la función de desempeño de los módulos, valor necesario como entrada al sistema difuso que estimará las densidades difusas para la combinación de miembros con la integral de Sugeno.

Una vez obtenida la simulación de cada miembro con los datos de entrenamiento, la función de desempeño se calcula como complemento del promedio del error absoluto en todos los puntos.

5.3 Estimación de densidades difusas con un sistema difuso tipo-2 por intervalos

Una vez obtenida la función de desempeño de cada módulo, se toman el valor mínimo y máximo de las mismas como extremos del rango para las variables difusas de entrada, de la misma forma que se propone en el método generalizado. De esta forma, aún que la diferencia en el desempeño de los miembros sea muy pequeña es posible normalizarlos obteniendo una mejor descripción del desempeño de los mimos.

Las reglas para el sistema difuso estiman densidades difusas cercanas a 0 cuando la función de desempeño es baja, cercanas a 0.5 cuando la función de desempeño es media, y cercanas a 1 cuando la función de desempeño es alta.

1. If (DX1 is BAJO) then (mX1 is BAJA) (1)
2. If (DX1 is MEDIO) then (mX1 is MEDIA) (1)
3. If (DX1 is ALTO) then (mX1 is ALTA) (1)
4. If (DX2 is BAJO) then (mX2 is BAJA) (1)
5. If (DX2 is MEDIO) then (mX2 is MEDIA) (1)
6. If (DX2 is ALTO) then (mX2 is ALTA) (1)
7. If (DX3 is BAJO) then (mx3 is BAJA) (1)
8. If (DX3 is MEDIO) then (mx3 is MEDIA) (1)
9. If (DX3 is ALTO) then (mx3 is ALTA) (1)
10. If (DX4 is BAJO) then (mx4 is BAJA) (1)
11. If (DX4 is MEDIO) then (mx4 is MEDIA) (1)
12. If (DX4 is ALTO) then (mx4 is ALTA) (1)

5.4 Combinación de miembros con integrales difusas

Si la combinación de los miembros se realiza calculando la integral de Sugeno para cada punto del conjunto de datos de prueba de la serie, los propios valores de la integral son la predicción de la misma.

5.4.1 Pruebas de validación cruzada

Se realizaron pruebas de validación cruzada de tipo 10-fold, obteniendo los resultados mostrados en la *Figura 5.2*, en promedio con diferentes anchos de FOU.

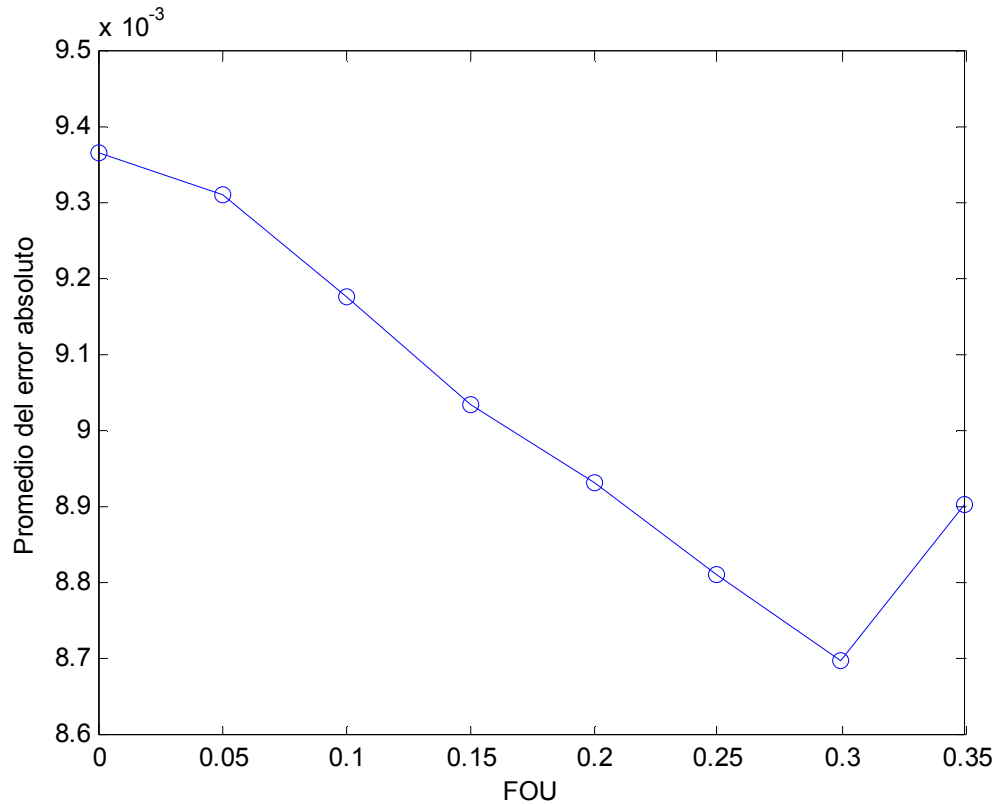


Figura 5.2 Porcentajes de reconocimiento para diferentes FOU, usando validación cruzada tipo *K*-fold

Como se puede observar en la gráfica, si existe una mejoría en los resultados cuando el ensemble se combina con la integral de Sugeno, usando densidades difusas obtenidas con el sistema difuso tipo-2 por intervalos.

En la *Figura 5.3* se observan las gráficas de la serie Mackey-Glass, obtenidas con el proceso de simulación para todos los puntos de la serie en cada miembro del ensemble $S(X_i)$, y la gráfica obtenida con la integración de los resultados del ensemble.

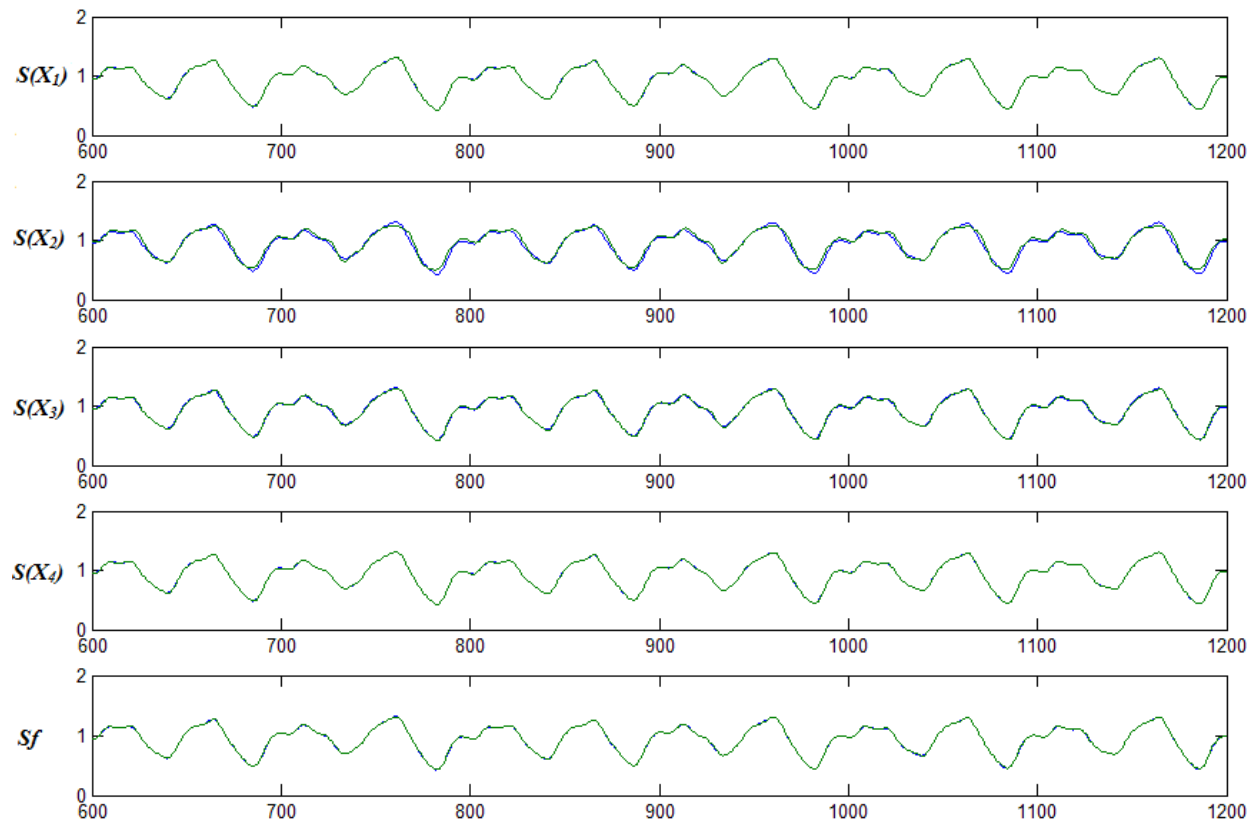


Figura 5.3 Gráfica de la predicción función Mackey-Glass

Como podemos observar aunque de manera superficial, la simulación final S_f toma lo mejor de los resultados de los 4 miembros para obtener una predicción más acertada.

6 Conclusiones

Se logró diseñar un método generalizado para integrar redes neuronales en sistemas multi-red, que se puede aplicar para resolver problemas distintos independientemente de la naturaleza y la cantidad de las fuentes de información. Se demostró que el método de integración funciona correctamente para dos tipos de aplicaciones muy distintas:

- Reconocimiento de imágenes con una arquitectura neuronal modular de tres módulos.
- Predicción con una arquitectura ensemble de tres miembros.

Tomando en cuenta las características de ambas aplicaciones, se puede esperar que este método funcione correctamente con otras muy diversas, solo con las adecuaciones pertinentes dentro de cada bloque del método, pero sin variaciones en su estructura esencial: Estimación de la relevancia de las fuentes de información con un sistema difuso tipo-2 por intervalos e integración de resultados con una integral difusa.

Como un agregado a este trabajo, ya que no se encuentra en el centro de atención del método de integración, sino del bloque de adquisición de datos, se logró la implementación de un detector de bordes basado en filtros de Sobel y un sistema difuso tipo-2 por intervalos, el cual mostró mejores resultados que el método tradicional.

En lo referente a las publicaciones, este método ha tenido buena aceptación en congresos y se lograron tres publicaciones en revistas, una de ellas indexada, además dos capítulos de libros, y

varias memorias en congresos. Lo cual es un buen parámetro para medir la calidad y madurez del trabajo.

En cuanto a la implementación de las aplicaciones, se realizó en Matlab para funcionar independientemente o para integrarse a la herramienta “Multi-net Toolbox”, desarrollada como mi trabajo de maestría en el ITT, la cual se ha venido enriquecido con las aportaciones de este trabajo de tesis de doctorado.

En cuanto al trabajo futuro, queda por experimentar con este mismo método, haciendo variaciones en los siguientes bloques:

- Adquisición de datos: Ubicación de objetos dentro de imágenes grandes o en videos y captura de imágenes en tiempo real.
- Detección de bordes en imágenes: Demostración de que los resultados obtenidos con el detector de bordes presentado en esta tesis es mejor que los métodos existentes, usando para ello métodos estadísticos de validación.
- Redes neuronales: Uso de redes neuronales distintas a las del tipo feed-forward.
- Sistema difuso: Optimización de parámetros con algoritmos evolutivos.
- Integral difusa: Implementación de la integral de Choquet o un nuevo operador de integración extendido para conjuntos difusos tipo-2 por intervalos.

Con esto en mente y debido al análisis realizado en este trabajo de tesis, puedo estimar que de cada uno de los puntos anteriores tienen potencial para convertirse en aplicaciones exitosas.

7 Referencias

- [1] © 1994-2006 The Math Works, Inc. Neural Network Toolbox 5.0.1, information on the web page: <http://www.mathworks.com/products/neuralnet/>
- [2] AT&T Laboratories Cambridge, The ORL database of faces, information on the web page: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [3] Bezdek J.C., Keller J., Pal N.R., 2005, Fuzzy Models and Algorithms for Pattern Recognition and Image Processing, Springer, U.S.A.
- [4] Castillo O., Melin P., 2008, Type-2 Fuzzy Logic: Theory and Applications, Springer-Verlag, Germany.
- [5] Castillo O., Melin P., Kacprzyk J., Pedrycz W., 2007, Hybrid Intelligent Systems: Design and Analysis, Springer-Verlag.
- [6] Castro J.R, Castillo O., Melin P., Rodriguez-Diaz A., 2008, Building fuzzy inference systems with a new interval type-2 fuzzy logic toolbox: In Transactions on Computational Science Book Chapter, Vol. 4750, Springer, p.p. 104-114.
- [7] Castro J.R., Castillo O., Martinez L.G., Tool Box para Lógica Difusa Tipo-2 por Intervalos, International Seminar on Computational Intelligence 2006 IEEE - CIS Mexico Chapter, Tijuana, Mexico, pp. 100-108, 2006.
- [8] Chuang M., Chang R., Huang Y., 2000, Automatic Facial Feature Extraction in Model-based Coding, Journal of Information Science and Engineering, Vol.16, pp. 447-458.
- [9] Dongrui W., Mendel J., 2007, Aggregation Using the Linguistic Weighted Average and Interval Type-2 Fuzzy Sets, IEEE Transactions on Fuzzy Systems, Vol. 15, No. 6, pp. 1145-1161.
- [10] Dowdalla J.B., Pavlidisa I., Bebisb G., 2003, Face Detection in the Near-IR Spectrum, Dept. of Computer Science, University of Nevada, Reno, NV.
- [11] Dubois D., Marichal J., Prade H., Roubens M., 2001, The use of the discrete Sugeno integral in decision-making: A survey, in Multiple Criteria Decision Aiding, Joint Research Center, The European Commission, Luxembourg, pp. 81-98.
- [12] El-Khamy S.E., Lotfy M., El-Yamany N., 2000, A Modified Fuzzy Sobel Edge Detector, Radio Science Conference, 17th NRSC 2000. Seventeenth National, Minufiya, Egypt, pp. C32/1-C32/9.
- [13] Graves K., Nagarajah R., 2007, Uncertainty Estimation Using Fuzzy Measures for Multiclass Classification, IEEE Transactions on Neural Networks, Vol. 18, No.1, pp. 128-140.

- [14] Guang D., Yeung D., Qian Y., 2007, Face Recognition using a Kernel Fractional-Step Discriminant Analysis Algorithm, *Pattern Recognition Journal*, Vol. 40, No. 1, pp. 229-243.
- [15] Heath M., Sarkar S., Sanocki T., and Bowyer K.W., 1997, A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 12, pp. 1338-1359.
- [16] Hidalgo D., Castillo O., Melin P., 2008, Interval Type-2 fuzzy inference systems as integration methods in modular neural networks for multimodal biometry and its optimization with genetic algorithms, *International Journal of Biometrics*, 1(1):114-128.
- [17] Hidalgo D., Castillo O., Melin P., 2008, Type-1 and Type-2 fuzzy inference systems as integration methods in modular neural networks for multimodal biometry and its optimization with genetic algorithms, *Journal of Automation, Mobile Robotics and Intelligent Systems*, 2(1):59-74.
- [18] Jacques C., Bauchspiess A., 2001, Fuzzy Inference System Applied to Edge Detection in Digital Images, *Proceedings of the V Brazilian Conference on Neural Networks*, Rio de Janeiro, Brazil, pp. 481-486.
- [19] Joo M., Wu S., Lu J., Lye H., 2002, Face Recognition with Radial Basis Function (RBF) Neural Networks, *IEEE Transactions on Neural Networks*, Vol. 13, No. 3, pp. 697-710.
- [20] Klir G., 2005, *Uncertainty and Information*, Wiley & Sons, Inc., U.S.A.
- [21] Liming H., Cheng H.D., Ming Z., 2007, A High Performance Edge Detector Based on Fuzzy Inference Rules, *Information Sciences: An International Journal*, Elsevier Science Inc. New York, NY, USA, Volume 177, Issue 21, pp. 4768-4784.
- [22] Detyniecki M., 2000, *Mathematical Aggregation Operators and their Application to Video Querying*, Pierre and Marie Curie University, Paris, (Available on line on: <http://www-poleia.lip6.fr/~marcin/publications.htm>).
- [23] Melin P., Castillo O., 2005, *Hybrid Intelligent Systems for Pattern Recognition*, Springer-Verlag, Heidelberg, Germany.
- [24] Melin P., Castillo O., 2007, An intelligent hybrid approach for industrial quality control combining neural networks, fuzzy logic and fractal theory, *Journal of Information Science*, Vol. 77, No.7, pp. 1543-1557.
- [25] Melin P., Gonzalez C., Bravo D., Gonzalez F., Martinez G., 2007, Modular Neural Networks and Fuzzy Sugeno Integral for Pattern Recognition: the case of Human Face and Fingerprint, In: *Hybrid Intelligent Systems Design and Analysis*, Springer-Verlag, Germany, pp. 311-326.

- [26] Melin P., Mancilla A., Lopez M., Mendoza O., 2007, A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting, *Journal of Applied Soft Computing*, Elsevier, Holland, Volume 7, Issue 4, pp. 1217-1226.
- [27] Melin P., Mendoza O., Soto D., Gutierrez M., Solano D., 2006, An Intelligent System for Pattern Recognition and Time Series Prediction Using Modular Neural Networks, *IEEE World Congress on Computational Intelligence*, Vancouver, Canada, p.p. 8197-8193.
- [28] Mendel J., 2001, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, E.U.A.
- [29] Mendel J., 2007, Advances in type-2 sets and systems, *Information Sciences*, Vol.177, Elsevier, E.U.A., pp.84-110.
- [30] Mendel J., 2007, Type-2 Fuzzistics for Nonsymmetric Interval Type-2 Fuzzy Sets: Forward Problems, *IEEE Transactions on Fuzzy Systems*, Vol. 15, No. 5, pp. 916-930.
- [31] Mendez-Vazquez A., Gader P., Keller J.M., Chamberlin K., 2008, Minimum Classification Error Training for Choquet Integrals with Applications to Landmine Detection, *IEEE Transactions on Fuzzy Systems*, Vol. 16, No. 1, pp. 225-238.
- [32] Mendoza O, Melin P., Licea G., 2007, Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition, In: *Foundations of Fuzzy Logic and Soft Computing*, Springer, Germany, pp. 604-612.
- [33] Mendoza O., Melin P., 2004, Sistema para el Modelado Diseño y Pruebas de Arquitecturas Neuronales Multi-Red, *International Seminar on Computational Intelligence*, Tijuana, Mexico.
- [34] Mendoza O., Melin P., 2007, The Fuzzy Sugeno Integral as a Decision Operator in the Recognition of Images with Modular Neural Networks, in Edited Book: *Hybrid Intelligent Systems: Design and Analysis*, Springer-Verlag, p.p. 299-310.
- [35] Mendoza O., Melin P., 2006, Sistemas de Inferencia Difusos Tipo-1 y Tipo-2 Aplicados a la Detección de Bordes en Imágenes Digitales, *International Seminar on Computational Intelligence 2006 IEEEICIS Chapter Mexico*, Tijuana, Mexico, pp. 117-123.
- [36] Mendoza O., Melin P., Castillo O., Licea G., 2007, Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition, In: *Foundations of Fuzzy Logic And Soft Computing*, Springer, Germany, pp. 604-612.
- [37] Mendoza O., Melin P., Licea G., 2007, Fuzzy Inference Systems Type-1 and Type-2 for Digital Images Edges Detection, *Engineering Letters*, International Association of Engineers, E.U.A., Volume 15 Issue 1 (Available on line: http://www.engineeringletters.com/issues_v15/issue_1/EL_15_1_7.pdf)

- [38] Mendoza O., Melin P., Licea G., 2007, Modular Neural Networks and Type-2 Fuzzy Logic for Face Recognition, NAFIPS 2007 International Conference, San Diego, E.U.A., pp. 622-627.
- [39] Mendoza O., Melin P., Licea G., 2007, Type-2 Fuzzy Systems as a Method for Improving Training Data and Decision Making in Modular Neural Networks for Pattern Recognition, International Journal of Information Technology And Intelligent Computing, Academy of Humanities And Economics (Wshe), Polonia, Int. J. It&Ic Vol.1 No.4, p.p., available in <http://itic.wshe.lodz.pl>,
- [40] Mendoza O., Melin P., Licea G., 2007, Type-2 Fuzzy Systems for Improving Training Data and Decision Making in Modular Neural Networks for Image Recognition, International Joint Conference on Neural Networks, Orlando, E.U.A., pp. 1766-1770.
- [41] Mendoza O., Melin P., Licea G., 2008, A Hybrid Approach for Image Recognition Combining Type-2 Fuzzy Logic, Modular Neural Networks and the Sugeno Integral, Information Sciences, Vol., 179, Elsevier, E.U.A, pp. 2078-2101.
- [42] Mendoza O., Melin P., Sistemas de Inferencia Difusos Tipo-1 y Tipo-2 Aplicados a la Detección de Bordes en Imágenes Digitales, International Seminar on Computational Intelligence 2006 IEEEICIS Chapter Mexico, Tijuana, Mexico, pp. 117-123, 2006.
- [43] Mendoza O., Torres G., Melin P., 2005, An Intelligent System for Modeling and Simulation with Modular Neural Networks, International Conference in Artificial Intelligence, Las Vegas, U.S.A., 406-411.
- [44] Miosso C.J., Bauchspiess A., 2001, Fuzzy Inference System Applied to Edge Detection in Digital Images, V Brazilian Conference on Neural Networks, Brasil.
- [45] Pham T.Q. and Van-Vliet L.J., 2005, Blocking artifacts removal by a hybrid filter method, Proceedings of the Eleventh Annual Conference of the Advanced School for Computing and Imaging, E.U.A, p.p. 372-377.
- [46] Revathy K., Lekshmi S., Prabhakaran R., 2005, Fractal-Based Fuzzy Technique for Detection of Active Regions From Solar, Journal of Solar Physics Vol. 228, pp. 43-53.
- [47] Rong W. Li-qun G., Shu Y. and Yu-hua C., 2006, An Edge Detection Method by Combining Fuzzy Logic and Neural Network, Advances in Machine Learning and Cybernetics , Springer , Berlin, pp. 930-937.
- [48] Salinas R., Larraguibel L., 2001, Red Neuronal de Arquitectura Paramétrica en Reconocimiento de Rostros, Divulgación electrónica de las Ciencias, Volumen 17, available on the web site: <http://cabieta.uchile.cl/revista/17/articulos/articulos.html>
- [49] Sharkey A., 1999, Ensemble and Modular Multi-Net Systems, In: Combining Artificial Neural Nets, Springer, Germany, pp. 1-25.

- [50] Starovoitov V.V., Samal D.I., Briliuk D.V., 2002, Three Approaches for Face Recognition, The 6-th International Conference on Pattern Recognition and Image Analysis, October 21-26, Velikiy Novgorod, Russia, pp. 707-711
- [51] Suzuki K., Horiba I., Sugie N., Nanki M., 2003, Contour extraction of left ventricular cavity from digital subtraction angiograms using a neural edge detector, *Sistems and Computers Japan*, Wiley Japan, p.p. 55-69
- [52] Torra V., Narukawa Y., 2007, A View of Averaging Aggregation Operators, *IEEE Transactions on Fuzzy Systems*, Vol. 15, No.6, pp. 1063-10067.
- [53] Torra V., Narukawa Y., 2007, Modeling Decisions, Information Fusion and Aggregation Operators, Springer-Verlag, Germany.
- [54] Verikas A., Lipnickas A., Malmqvist K., Bacauskiene M., Gelzinis A., 1999, Soft Combination of Neural Classifiers, *Pattern Recognition Letters*, pp. 429-444.
- [55] Yager R., 2008, A knowledge-based approach to adversarial decision making, *International Journal of Intelligent Systems*, Vol. 23, No. 1, Wiley, pp. 1-21.
- [56] Yang P., Du B., Shan S., Wen G., 2004, A Novel Pupil Localization Method Based On Gaboreye Model And Radial Symmetry Operator, *ICT-ISVISION Joint R&D Laboratory for Face Recognition*, Beijing, China
- [57] Yau-Hwang K., Chang-Shing L., Chao-Chin L., 1997, A New Fuzzy Edge Detection Method for Image Enhancement, *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Spain, Vol. 2, Issue 1-5, pp. 1069-1074.
- [58] Young I.T., Gerbrands J.J., Van-Vliet L.J., 1995, *Fundamentals of Image Processing*, Delft: PH Publications, The Netherlands.
- [59] Zhenyuan W., Klir G.J., 1991, *Fuzzy Measure Theory*, Plenum Press, New York.
- [60] Zuo W., Zhang D., Yang J., Wang K., 2006, BBPCA Plus LDA: A Novel Fast Feature Extraction Technique for Face Recognition, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 36, No.4, pp. 946-953.

8 Anexos

Anexo A : Programas

Programa para calcular la integral de Sugeno de n fuentes de información

```
function rintegral=finalsugeno_gral(total,g)
lambda=calc_lambda21(g);
rintegral=[];
[r,c]=size(total);
[rg,cg]=size(g);

if(rg<r)
    g=g';
end
for i=1:c
    htmp=[];
    htmp=[total(:,i) g];
    [orden,indh]=sortrows(htmp,-1);
    xorden=orden(:,1);
    gorden=orden(:,2);
    medidas=medidas_sugeno(gorden,lambda);
    minimos=[];
    for j=1:r
        minimo=min(xorden(j),medidas(j));
        minimos=[minimos minimo];
    end
    integral=max(minimos);
    rintegral=[rintegral integral];
end
```

Programa para calcular medidas difusas de Sugeno o Lambda difusas.

```
function [medidas]=medidas_sugeno(g,lambda)
[ren,col]=size(g);
if ren==1
    g=g';
    n=col;
else
    n=ren;
end

medidas(1)=g(1);
for i=2:n
    medidas(i)=g(i)+medidas(i-1)+lambda*g(i)*medidas(i-1);
end
```

Programa para calcular lambda para n densidades difusas g

```
function L=calc_lambda21(g)
n=length(g);
sfx=formaf(n,g);
grafica_funcion(sfx,-1,3,.2);
% Segun teorema 5.3 del libro Modeling Decisions [36]
sumag=sum(g);
    if sumag < 1
        x0=10;
    elseif sumag==1
        x0=0;
    else
        x0=-0.999;
    end
    [L,fvall,exitflag1,output1] = fzero(sfx,x0);
end

function sfx=formaf(n,g)
    %fx='(1+g(1)*L)*(1+g(2)*L)*(1+g(3)*L)-(1+L)' % ejemplo para tres
    %nodos
    sfx=[];
    for i=1:n
        if i>1
            sfx=[sfx,'*'];
        end
        sfx=[sfx,'(1+',num2str(g(i)),'*x)'];
    end
    sfx=[sfx,'-(1+x)'];
end

function grafica_funcion(sfx,li,ls,inc)
datx=li:inc:ls;
for i=1:length(datx)
    x=datx(i);
    datfx(i)=eval(sfx);
end

% figure('name',strcat('Funcion Lambda'),'Position',[50 400 600 300]);
figure
plot(datx,datfx);
set(gca,'XGrid','on');
set(gca,'YGrid','on');
end
```

Programa para combinar módulos con la integral de Sugeno, para la aplicación de reconocimiento de imágenes

```
function
cant_asiertos=combina_modulos_fis2_unoauno(fis,mod,modulos,m_total,k)
matriz_integral=[];
identificacion=[];
cant_muestras=m_total*k;

    for z=281:cant_muestras
        integrar=[];
        variables=[];
        for i=1:modulos
            vmin(i)=min(mod(:,z,i));
            vmax(i)=max(mod(:,z,i));
            integrar=[integrar mod(:,z,i)];
        end
        max_mod= repmat(vmax,40,1);
        entrada_fis=[integrar max_mod];
        densidades = evalifistype2(entrada_fis,fis,20);
        integral_sugeno=finalsugeno_gral(integrar',densidades');
        [C,indice] = max(integral_sugeno);
        identificacion=[identificacion indice];
        matriz_integral=[matriz_integral integral_sugeno'];
    end

    objetivo=repmat(eye(40),1,3);
    error=abs(objetivo-matriz_integral);
    u=1:40;
    objetivou=repmat(u,1,3);
    erroru=objetivou-identificacion;
    vector_asiertos=find(erroru==0);
    cant_asiertos=length(vector_asiertos);
```

Programa completo para entrenamiento, simulación y combinación con la integral de Sugeno estimando densidades difusas con un sistema difuso tipo-2 por intervalos. La estructura ensemble es de cuatro miembros para predicción de la función Mackey-Glass usando el método de validación cruzada k-fold.

```
clear all;
mg=load('mgdata.dat');
time=mg(1:1200,1)';
T=mg(1:1200,2)';
a=min(T);
b=max(T);
c=b;
Tfuzzy=trimf(T,[a b c]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RETRAZOS

Q = length(Tfuzzy);
miembros=4;

P = zeros(4,Q);
P(1,2:Q) = Tfuzzy(1,1:(Q-1));
P(2,3:Q) = Tfuzzy(1,1:(Q-2));
P(3,4:Q) = Tfuzzy(1,1:(Q-3));
P(4,5:Q) = Tfuzzy(1,1:(Q-4));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
K=10;
for i=1:K
    ind = crossvalind('Kfold',Q, K)
    indtest = (ind == i); indtrain = ~indtest;

    Ptrain=P(:,indtrain);
    Ptest=P(:,indtest);

    Ttrain=Tfuzzy(indtrain);
    Ttest=Tfuzzy(indtest);

    realtrain=T(indtrain);
    realtest=T(indtest);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    capa1=4;
    capa2=8;
    capa3=1;

    %modulo 1

net1=newff(minmax(Ptrain),[capa1,capa2,capa3],{'tansig','tansig','purelin'},'t
rainlm');
net1.trainParam.show=10;
net1.trainParam.goal=1e-3;
net1.trainParam.epochs=1000;
disp 'Inicia a entrenar modulo1'
tic
```

```

[net1,tr1]=train(net1,Ptrain,Ttrain);
tic
save('net_serie1','net1');

%modulo 2

net2=newff(minmax(Ptrain),[capa1,capa2,capa3],{'tansig','tansig','purelin'},'t
raingdx');
net2.trainParam.show=10;
net2.trainParam.goal=1e-3;
net2.trainParam.epochs=1000;
disp 'Inicia a entrenar modulo2'
tic
[net2,tr2]=train(net2,Ptrain,Ttrain);
toc
save('net_serie2','net2');

%modulo 3

net3=newff(minmax(Ptrain),[capa1,capa2,capa3],{'tansig','tansig','purelin'},'t
raingda');
net3.trainParam.show=10;
net3.trainParam.goal=1e-3;
net3.trainParam.epochs=1000;
disp 'Inicia a entrenar modulo4'
tic
[net3,tr3]=train(net3,Ptrain,Ttrain);
toc
save('net_serie3','net3');

%modulo 4

net4=newff(minmax(Ptrain),[capa1,capa2,capa3],{'tansig','tansig','purelin'},'t
rainlm');
net4.trainParam.show=10;
net4.trainParam.goal=1e-3;
net4.trainParam.epochs=1000;
disp 'Inicia a entrenar modulo4'
tic
[net4,tr4]=train(net4,Ptrain,Ttrain);
toc
save('net_serie4','net4');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ret=4; %retrazos

%Simulacion con datos de entrenamiento para funcion de utilidad
%Miembro1
[Y1fuzzye,Pfe,Afe,Ele,perfle]=sim(net1,Ptrain);
Y1e=Y1fuzzye.*(b-a)+a;
ele=mean(abs(realtrain-Y1e))

%Simulacion con datos de prueba para prediccion
[Y1fuzzy,Pf,Af,E1,perf1]=sim(net1,Ptest);
Y1=Y1fuzzy.*(b-a)+a;
e1=mean(abs(realtest-Y1))

```

```

%Miembro2
%Simulacion con datos de entrenamiento para funcion de utilidad

[Y2fuzzye, Pfe, Afe, Ele, perf2e]=sim(net2,Ptrain);
Y2e=Y2fuzzye.*(b-a)+a;
e2e=mean(abs(realtrain-Y2e))

%Simulacion con datos de prueba para prediccion
[Y2fuzzy, Pf, Af, E2, perf2]=sim(net2,Ptest);
Y2=Y2fuzzy.*(b-a)+a;
e2=mean(abs(realtest-Y2))

%Miembro 3
%Simulacion con datos de entrenamiento para funcion de utilidad

[Y3fuzzye, Pfe, Afe, E3e, perf3e]=sim(net3,Ptrain);
Y3e=Y3fuzzye.*(b-a)+a;
e3e=mean(abs(realtrain-Y3e))

%Simulacion con datos de prueba para prediccion
[Y3fuzzy, Pf, Af, E3, perf3]=sim(net3,Ptest);
Y3=Y3fuzzy.*(b-a)+a;
e3=mean(abs(realtest-Y3))

%Miembro 4
%Simulacion con datos de entrenamiento para funcion de utilidad

[Y4fuzzye, Pfe, Afe, E4e, perf4e]=sim(net4,Ptrain);
Y4e=Y4fuzzye.*(b-a)+a;
e4e=mean(abs(realtrain-Y4e))

%Simulacion con datos de prueba para prediccion
[Y4fuzzy, Pf, Af, E3, perf4]=sim(net4,Ptest);
Y4=Y4fuzzy.*(b-a)+a;
e4=mean(abs(realtest-Y4))

%Combinacion con integral de sugeno con datos de prediccion
fuzzytotal=[Y1fuzzy;Y2fuzzy;Y3fuzzy;Y4fuzzy];

fuzzytotale=[Y1fuzzye;Y2fuzzye;Y3fuzzye;Y4fuzzye];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% errores=[e1;e2;e3]; %Usando errores de prediccion
errores=[e1e;e2e;e3e;e4e]; %Usando errores de entrenamiento
Ex=1-errores;

maxEx=max(Ex);
minEx=min(Ex);

err(i,:)=errores';
fexpert(i,:)=Ex';

huella=[0 0.05 0.1 0.15 0.2 0.25 0.3 0.35];
% huella=[0 0.2]
for j=1:length(huella)
    fis=fis2_densidades_4mod_generalizado(minEx,maxEx,huella(j))
    g(i,:,j)=evalifistype2(Ex,fis)

```

```

fuzzySf=finalsugeno_gral(fuzzytotal,g(i,:,j));
fuzzySfe=finalsugeno_gral(fuzzytotale,g(i,:,j));

%%%%% Defusificacion de las integrales

Sf(i,:,j)=fuzzySf.*(b-a)+a;
Sfe(i,:,j)=fuzzySfe.*(b-a)+a;
%%%%%

%Combinacion de miembros
eSugeno2(i,j)=mean(abs(realtest-Sf(i,:,j)));
desvSugeno2(i,j)=std(abs(realtest-Sf(i,:,j)));

end
Sprom(i,:)=(Y1+Y2+Y3+Y4)/4; %promedio
eprom(i,:)=mean(abs(realtest-Sprom(i,:)));
desvprom(i,:)=std(abs(realtest-Sprom(i,:)));

end

% figure
% plot(huella,eSugeno2(1,:),huella,eSugeno2(2,:),huella,eSugeno2(3,:),...
% huella,eSugeno2(4,:),huella,eSugeno2(5,:),huella,eSugeno2(6,:),...
% huella,eSugeno2(7,:),huella,eSugeno2(8,:),huella,eSugeno2(8,:),...
% huella,eSugeno2(10,:));
% xlabel('FOU');ylabel('Error absoluto');

figure

promSugeno2=mean(eSugeno2);

plot(huella,promSugeno2,'-o');
xlabel('FOU');
ylabel('Promedio del error absoluto');

[minhuella,ifold]=min(eSugeno2)
[mine,jhuella]=min(minhuella)

indhuella=jhuella;
indfold=ifold(indhuella);

```

Programa par construir un sistema difuso tipo-2 por intervalos para la detección de bordes en imágenes.

```
function fis=imagen_bordes_fis2_blanco_nuevo_norm()

NumMf = [3 3 3 3];
MfType = str2mat('igaussmtype2','igaussmtype2','igaussmtype2','igaussmtype2');

MfParams{1}=[50 -30 30;50 97 157;50 225 285];
MfParams{2}=[60 0 60;60 200 260;50 330 400];
MfParams{3}=[60 0 60;60 200 260;50 330 400];
MfParams{4}=[1.25 -4 -5;1.25 1 2];

fis = newfistype2('bordes','mamdani');

fis = addvartype2(fis,'input','m',[-200 400]);
fis = addvartype2(fis,'input','gx',[-200 500]);
fis = addvartype2(fis,'input','gy',[-200 500]);
fis = addvartype2(fis,'output','eges',[-4.5 1.5]);
%
fis = addmftype2(fis,'input',1,'LOW',MfType(1,:),MfParams{1}(1,:));
fis = addmftype2(fis,'input',1,'MEDIUM',MfType(1,:),MfParams{1}(2,:));
fis = addmftype2(fis,'input',1,'HIGH',MfType(1,:),MfParams{1}(3,:));
%
fis = addmftype2(fis,'input',2,'LOW',MfType(2,:),MfParams{2}(1,:));
fis = addmftype2(fis,'input',2,'MEDIUM',MfType(2,:),MfParams{2}(2,:));
fis = addmftype2(fis,'input',2,'HIGH',MfType(2,:),MfParams{2}(3,:));
%
fis = addmftype2(fis,'input',3,'LOW',MfType(3,:),MfParams{3}(1,:));
fis = addmftype2(fis,'input',3,'MEDIUM',MfType(3,:),MfParams{3}(2,:));
fis = addmftype2(fis,'input',3,'HIGH',MfType(3,:),MfParams{3}(3,:));

fis = addmftype2(fis,'output',1,'LOW',MfType(4,:),MfParams{4}(1,:));
fis = addmftype2(fis,'output',1,'HIGH',MfType(4,:),MfParams{4}(2,:));
% Lista de reglas
ruleList= [0 1 1 0 2 1 1;
           0 2 2 0 1 1 1;
           0 3 3 0 1 1 1;
           1 0 2 0 1 1 1;
           1 2 0 0 1 1 1];
fis=addruletype2(fis,ruleList);
% Visualiza reglas
showrule(fis)
% Visualiza entradas / salidas
figure;
subplot(4,1,3);plotimftype2(fis,'input',1); grid on
xlabel('m');
subplot(4,1,1);plotimftype2(fis,'input',2); grid on
xlabel('g_x');
subplot(4,1,2);plotimftype2(fis,'input',3); grid on
xlabel('g_y');
subplot(4,1,4);plotimftype2(fis,'output',1); grid on
xlabel('edges');
```

end

Programa par construir un sistema difuso tipo-2 por intervalos para estimar densidades difusas de tres fuentes de información

```
function fis=fis2_densidades_3mod_generalizado(minEx,maxEx,factor)

s=(maxEx-minEx)/8;
hh=(maxEx-minEx)*factor/2;
hh=hh/2;
h=factor/4;

NumMf = [3 3 3 3 3 3];
MfType = str2mat('igaussmtype2','igaussmtype2','igaussmtype2',...
                'igaussmtype2','igaussmtype2','igaussmtype2');

cbajo=minEx;
cmedio=(maxEx+minEx)/2;
calto=maxEx;

bajol=cbajo-hh;
bajor=cbajo+hh;
mediol=cmedio-hh;
medior=cmedio+hh;
altol=calto-hh;
altor=calto+hh;

MfParams{1}=[s bajol bajor;s mediol medior;s altol altor];
MfParams{2}=[s bajol bajor;s mediol medior;s altol altor];
MfParams{3}=[s bajol bajor;s mediol medior;s altol altor];

cbajod=0;
cmediod=0.5;
caltod=1;
sm=(caltod-cbajod)/8;

bajold=cbajod-h;
bajord=cbajod+h;
mediold=cmediod-h;
mediord=cmediod+h;
altold=caltod-h;
altord=caltod+h;

MfParams{4}=[sm bajold bajord;sm mediold mediord;sm altold altord];
MfParams{5}=[sm bajold bajord;sm mediold mediord;sm altold altord];
MfParams{6}=[sm bajold bajord;sm mediold mediord;sm altold altord];

fis = newfistype2('densidades','mamdani');

fis = addvartype2(fis,'input','D1',[minEx maxEx]);
fis = addvartype2(fis,'input','D2',[minEx maxEx]);
fis = addvartype2(fis,'input','D3',[minEx maxEx]);
```



```

fis = addvartype2(fis, 'output', 'mx1', [0 1]);
fis = addvartype2(fis, 'output', 'mx2', [0 1]);
fis = addvartype2(fis, 'output', 'mx3', [0 1]);

%
fis = addmftype2(fis, 'input', 1, 'BAJO', MfType(1,:), MfParams{1}(1,:));
fis = addmftype2(fis, 'input', 1, 'MEDIO', MfType(1,:), MfParams{1}(2,:));
fis = addmftype2(fis, 'input', 1, 'ALTO', MfType(1,:), MfParams{1}(3,:));
%
fis = addmftype2(fis, 'input', 2, 'BAJO', MfType(2,:), MfParams{2}(1,:));
fis = addmftype2(fis, 'input', 2, 'MEDIO', MfType(2,:), MfParams{2}(2,:));
fis = addmftype2(fis, 'input', 2, 'ALTO', MfType(2,:), MfParams{2}(3,:));
%
fis = addmftype2(fis, 'input', 3, 'BAJO', MfType(3,:), MfParams{3}(1,:));
fis = addmftype2(fis, 'input', 3, 'MEDIO', MfType(3,:), MfParams{3}(2,:));
fis = addmftype2(fis, 'input', 3, 'ALTO', MfType(3,:), MfParams{3}(3,:));
%
fis = addmftype2(fis, 'output', 1, 'BAJA', MfType(4,:), MfParams{4}(1,:));
fis = addmftype2(fis, 'output', 1, 'MEDIA', MfType(4,:), MfParams{4}(2,:));
fis = addmftype2(fis, 'output', 1, 'ALTA', MfType(4,:), MfParams{4}(3,:));

fis = addmftype2(fis, 'output', 2, 'BAJA', MfType(5,:), MfParams{5}(1,:));
fis = addmftype2(fis, 'output', 2, 'MEDIA', MfType(5,:), MfParams{5}(2,:));
fis = addmftype2(fis, 'output', 2, 'ALTA', MfType(5,:), MfParams{5}(3,:));

fis = addmftype2(fis, 'output', 3, 'BAJA', MfType(6,:), MfParams{6}(1,:));
fis = addmftype2(fis, 'output', 3, 'MEDIA', MfType(6,:), MfParams{6}(2,:));
fis = addmftype2(fis, 'output', 3, 'ALTA', MfType(6,:), MfParams{6}(3,:));

% Lista de reglas
ruleList= [ 1 0 0    1 0 0    1 1;
            2 0 0    2 0 0    1 1;
            3 0 0    3 0 0    1 1;
            0 1 0    0 1 0    1 1;
            0 2 0    0 2 0    1 1;
            0 3 0    0 3 0    1 1;
            0 0 1    0 0 1    1 1;
            0 0 2    0 0 2    1 1;
            0 0 3    0 0 3    1 1
          ];

fis=addruletype2(fis, ruleList);

showrule(fis)

figure;
subplot(3,2,1);plotimftype2(fis, 'input', 1); grid on
xlabel('D_1');
subplot(3,2,3);plotimftype2(fis, 'input', 2); grid on
xlabel('D_2');
subplot(3,2,5);plotimftype2(fis, 'input', 3); grid on
xlabel('D_3');

subplot(3,2,2);plotimftype2(fis, 'output', 1); grid on
xlabel('mx_1');

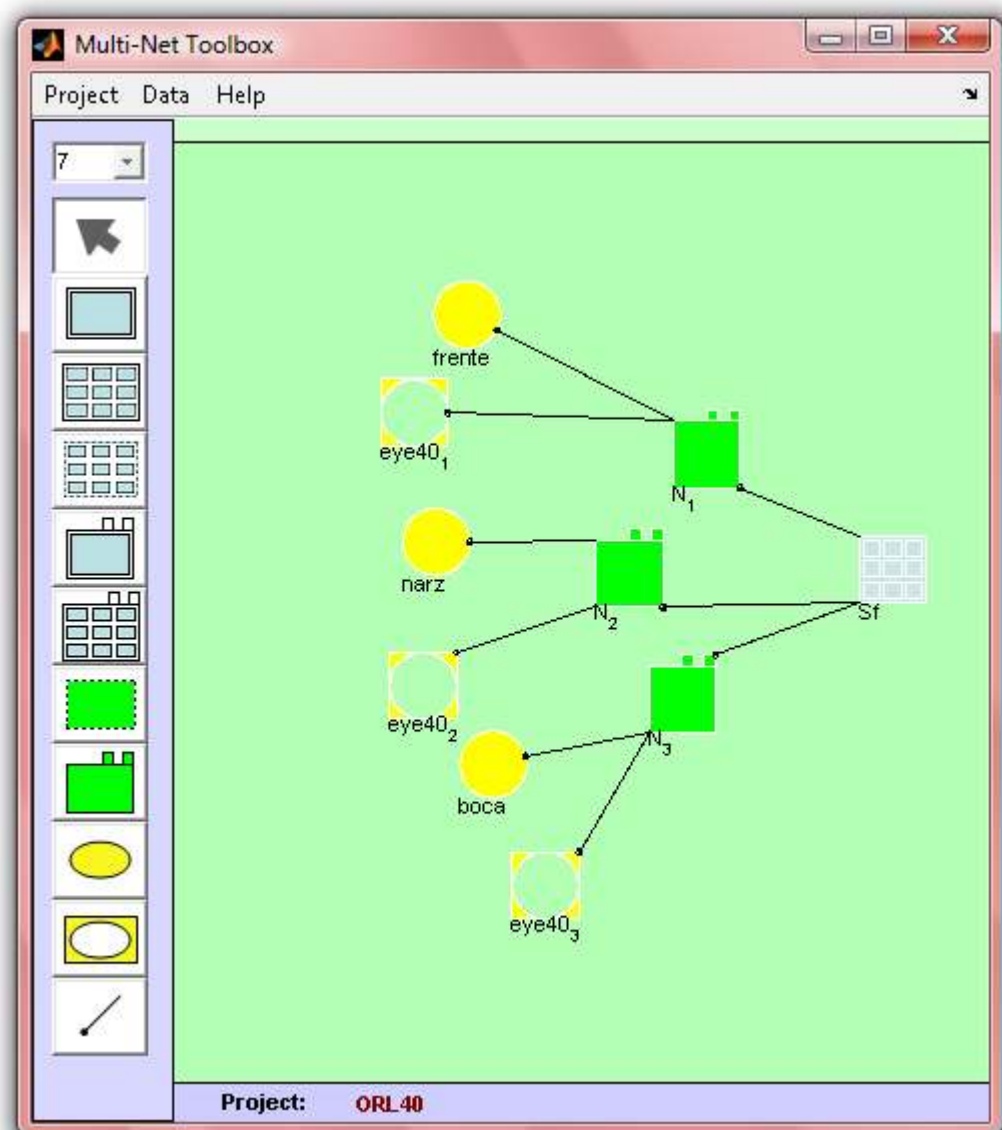
```

```
subplot(3,2,4);plotimftype2(fis,'output',2); grid on
xlabel('mx_2');
subplot(3,2,6);plotimftype2(fis,'output',3); grid on
xlabel('mx_3');

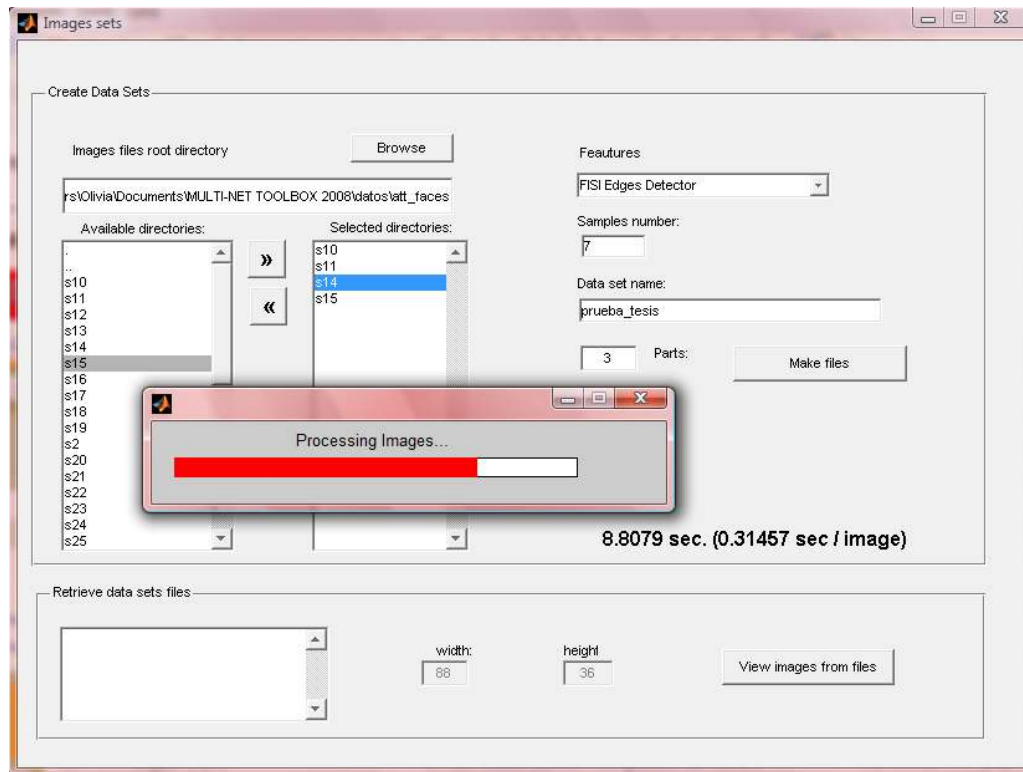
end
```

Anexo B : Demostración de la herramienta multi-red (multi-net toolbox).

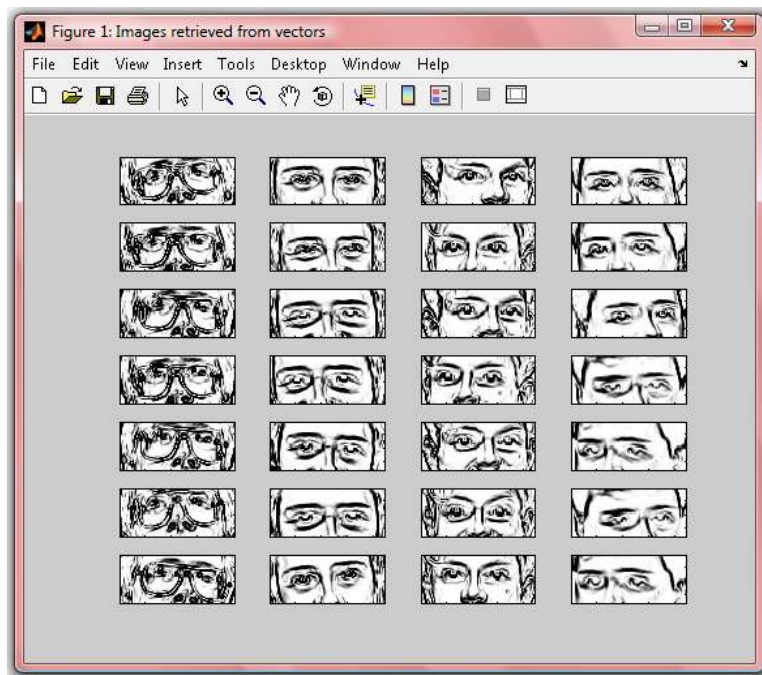
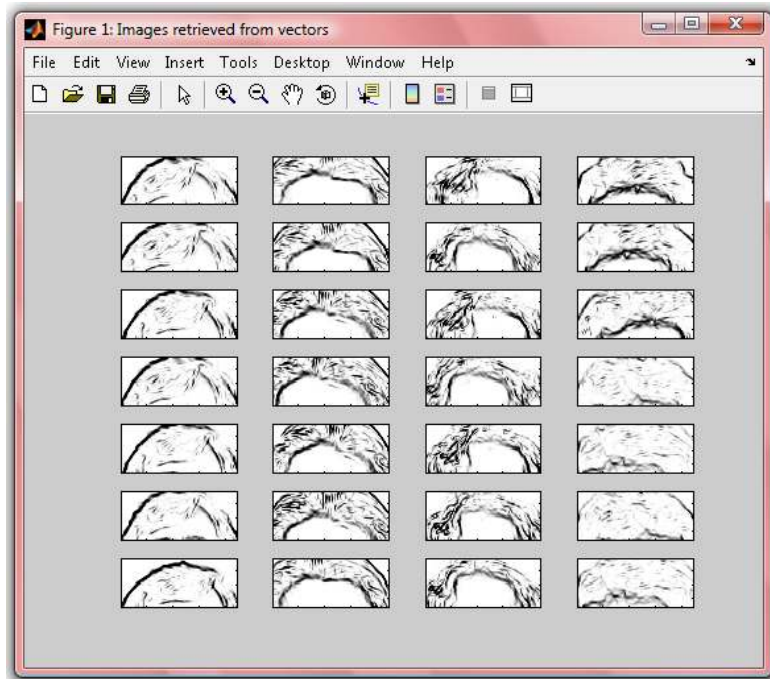
Sistema neuronal modular para reconocimiento de las personas de la base de datos ORL. Cada módulo N_i tiene como datos de entrenamiento uno de tres partes de 7 de las 10 muestras de las 40 personas en la base de datos.

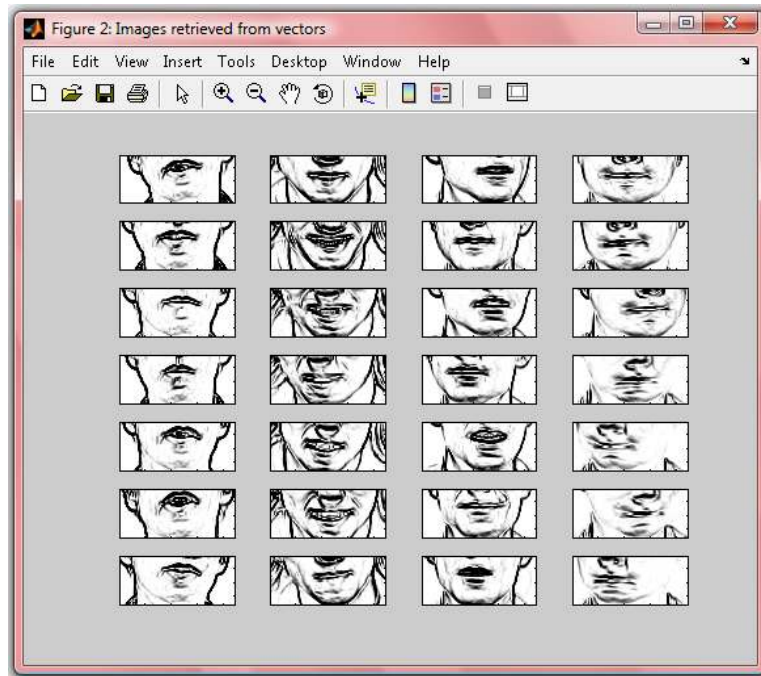


Pantalla para construir conjuntos de imágenes para entrenamiento.

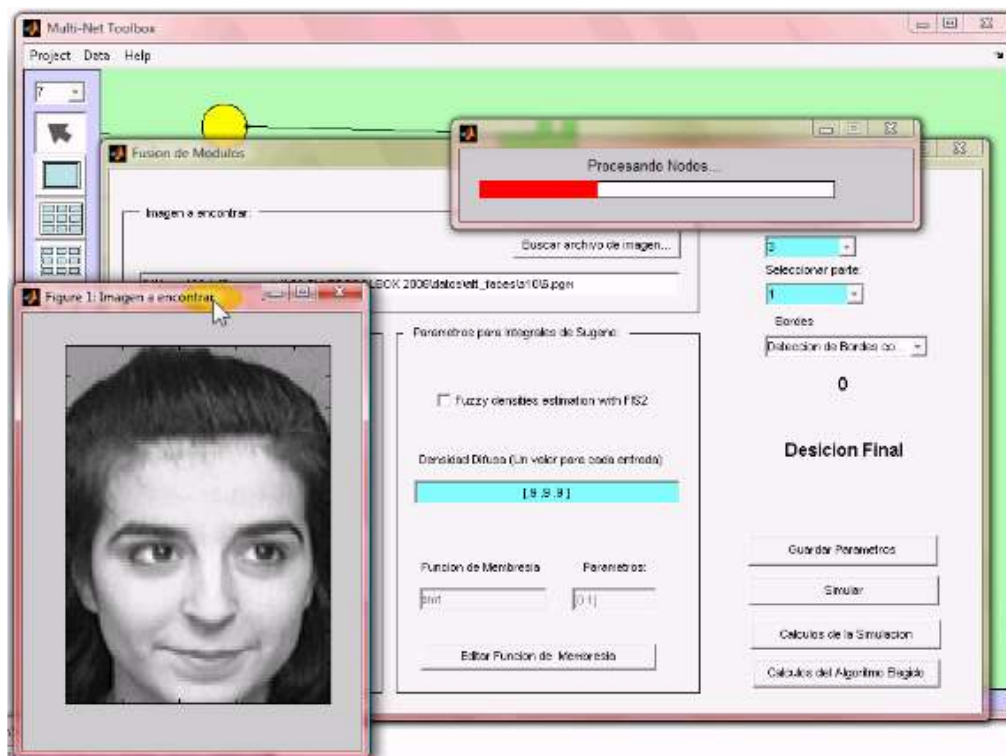


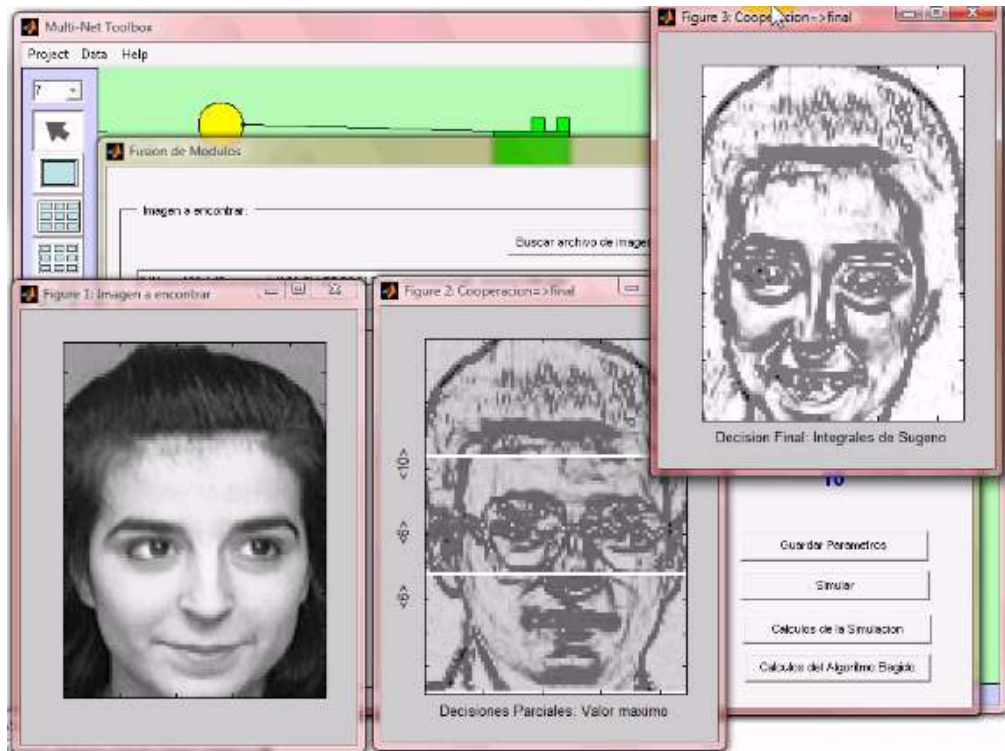
Muestras de imágenes divididas en tres partes:



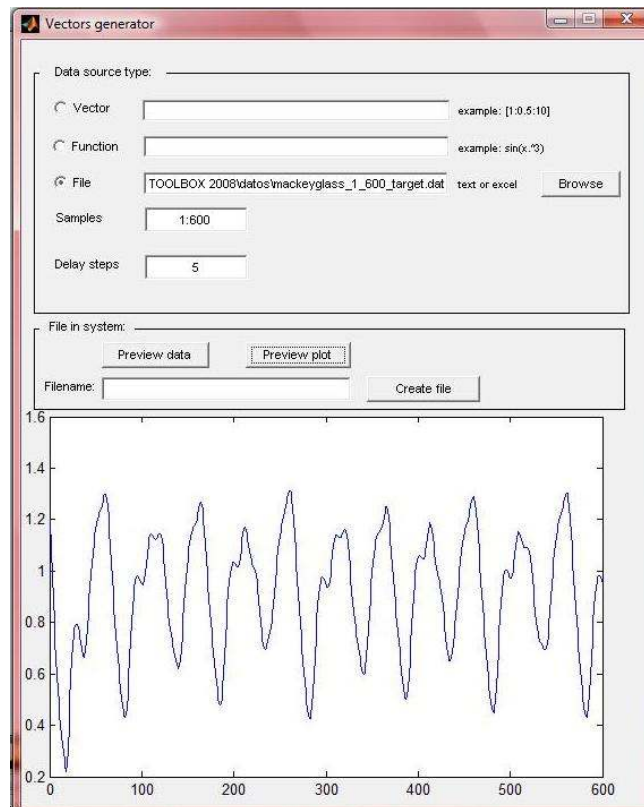
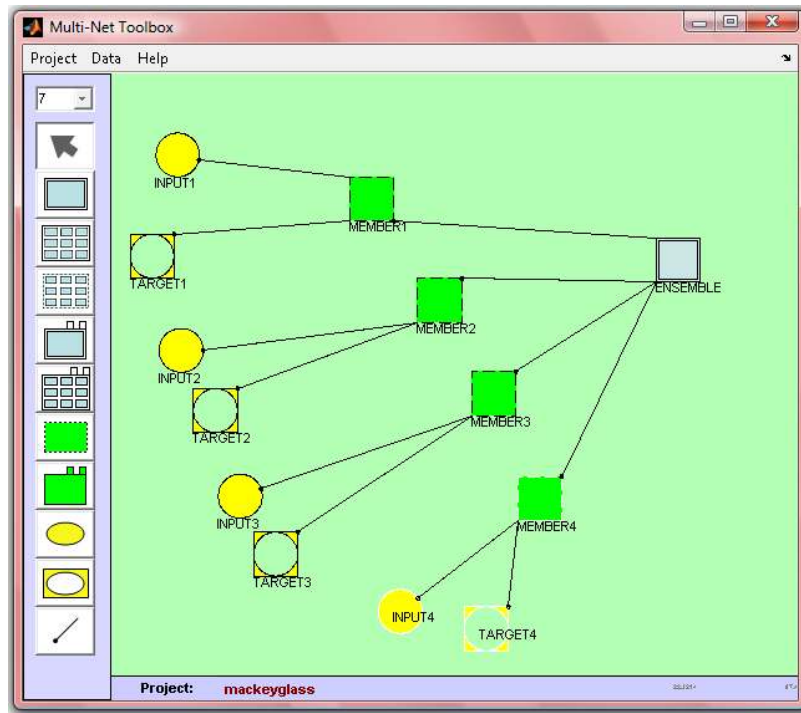


Reconocimiento de una imagen con el sistema neuronal modular.





Sistema neuronal ensemble para predicción de la serie Mackey-Glass. Cada miembro N_i tiene como datos de entrenamiento los datos de la serie en el rango $[1,600]$.



Simulación de la serie Mackey-Glass en el rango $[1,1200]$ con el sistema neuronal ensemble.

