

**Implementación de Redes Neuro-Difusas para ser Aplicadas en
Problemas de Clasificación y Modelización**

by
José D. Martín

ISBN: 1-58112-113-X

DISSERTATION.COM



USA • 2000

*Implementación de Redes Neuro-Difusas para ser
Aplicadas en Problemas de Clasificación y Modelización*

Copyright © 2000 José D. Martín
All rights reserved.

Dissertation.com
USA • 2000

ISBN: 1-58112-113-X

www.dissertation.com/library/112113xa.htm

IMPLEMENTACIÓN DE REDES NEURO-DIFUSAS PARA SER APLICADAS EN PROBLEMAS DE CLASIFICACIÓN Y MODELIZACIÓN



Ingeniería Electrónica

Proyecto Fin de Carrera:

José David Martín Guerrero

Director: Dr. Emilio Soria Olivas

ÍNDICE

Agradecimientos	4
Capítulo 1: Introducción a las redes neuro-difusas	5
1.1. Introducción a las redes neuronales artificiales	5
1.1.1. Analogía fisiológica	5
1.1.2. Nociones básicas	10
1.2. Sistemas basados en Lógica Difusa	10
1.3. Redes neuro-difusas	20
Capítulo 2: Descripción de las redes utilizadas	23
2.1. Redes basadas en sistemas difusos	23
2.1.1. ANFIS (<i>Adaptive Neuro-Fuzzy Inference Systems</i>)	23
2.1.2. MANFIS (<i>Multiple ANFIS</i>)	40
2.1.3. Espectro neuro-difuso	40
2.2. Redes basadas en MLP	42
2.2.1. MLP con función de coste difusa	43
2.2.2. MLP con neuronas difusas	46
2.2.3. MLP con clasificación difusa	50
Capítulo 3: Problemas utilizados	56
3.1. Problemas juguete: una primera prueba	56
3.1.1. Problema juguete en clasificación: puerta lógica XOR	56
3.1.2. Problema juguete en modelización: seno	57
3.2. Problema de los datos sintéticos de <i>B. D. Ripley</i>	58
3.3. Modelización de un seno de dos entradas	59
3.4. Problema de reconstrucción de señales binarias	59
3.5. Predicción de la serie caótica de <i>Mackey-Glass</i>	64
3.6. Clasificación de pacientes con riesgo de intoxicación por digoxina	64

Capítulo 4: Resultados	67
4.1. Problema juguete en clasificación: puerta lógica XOR	67
4.2. Problema juguete en modelización: seno	68
4.3. Modelización de un seno de dos entradas	69
4.4. Modelización de la serie de <i>Mackey-Glass</i>	69
4.5. Datos sintéticos de <i>Ripley</i>	70
4.6. Reconstrucción de señales binarias	77
4.6.1. Ecuación del canal $H_1(z)=0.5+z^{-1}$	77
4.6.2. Ecuación del canal $H_2(z)=0.3482+0.8704z^{-1}+0.3482z^{-2}$	91
4.7. Clasificación de pacientes con riesgo de intoxicación por digoxina	101
Capítulo 5: Conclusiones y proyección futura	105
5.1. ANFIS	105
5.2. MANFIS	105
5.3. MLP con función de coste difusa	106
5.4. MLP con neuronas difusas	106
5.5. MLP con clasificación difusa	107
5.6. Conclusiones finales y perspectiva futura	107
Bibliografía	109

AGRADECIMIENTOS

Primeramente, quiero agradecer a Emilio Soria las ideas y consejos para la realización del proyecto.

A mi compañero y amigo Javier Molina por introducirme en la Electrónica.

A Antonio J. Serrano y Gustavo Camps por la información que me han facilitado.

A mi familia y amigos.

CAPÍTULO 1. INTRODUCCIÓN A LAS REDES NEURO-DIFUSAS

RESUMEN DEL CAPÍTULO

*En este capítulo, se presenta una perspectiva general sobre el tipo de sistemas utilizados en este proyecto. Se trata de sistemas que combinan la capacidad de aprendizaje y generalización de las **RNs** con la robustez que supone trabajar con **FL** en lugar de usar Lógica Binaria, en una doble perspectiva: por un lado, las descripciones se aproximan más al conocimiento intuitivo humano, y, por otro, los problemas binarios se pueden tratar de un modo más sutil, lo que implica una mayor robustez.*

1.1. INTRODUCCIÓN A LAS REDES NEURONALES ARTIFICIALES

1.1.1. Analogía fisiológica

La creación de computadores que “piensen” es uno de los retos principales que se plantean ingenieros y programadores, que, día a día, se afanan por encontrar computadores más “inteligentes”. En particular, se desea que una máquina pueda aprender de la experiencia en lugar de repetir indefinidamente un conjunto explícito de instrucciones generadas por un programador humano.

Por ejemplo, supongamos que a una persona se le enseña una fotografía que muestra un dalmata visto de perfil, es decir, que estará viendo una serie de manchas negras sobre fondo blanco. Sin demasiado esfuerzo, esta persona identificará que la fotografía se corresponde con parte del cuerpo de un dalmata. Sin embargo, ¿cómo se podría escribir un programa de computador que llegue a esta conclusión, determinando qué manchas forman parte del contorno del perro, qué manchas pueden atribuirse a manchas de la piel y cuáles son únicamente manchas debidas a defectos en la fotografía?, y, ¿cómo es que nosotros podemos identificar rápidamente el perro al observar la fotografía? Este último hecho es curioso, si tenemos en cuenta que el tiempo de conmutación de un computador electrónico moderno es de más de siete órdenes de magnitud más rápido que el de las células de que constan nuestros sistemas neurobiológicos, y, sin embargo, la velocidad para el procesado de información auditiva

y visual es mucho mayor en el cerebro humano que en computadores modernos. Esto es debido básicamente a que la arquitectura del cerebro humano es distinta de la arquitectura de un computador convencional. Aunque el tiempo de respuesta de una célula neuronal individual es típicamente del orden de unas pocas decenas de milisegundos, el paralelismo masivo y la interconectabilidad que se observan en los sistemas biológicos complejos son la causa de la gran capacidad del cerebro para llevar a cabo complejos reconocimientos [Freeman-93].

En muchas aplicaciones de la vida real interesará que los computadores resuelvan problemas que necesiten una cierta experiencia que les permita aprender lo necesario para solventarlo.

Por esta razón se toman ciertas características de la fisiología del cerebro como base para unos modelos de procesamiento, conocidos como **redes neuronales artificiales**. Es importante tener siempre presente que estos elementos no modelizan el cerebro, sino que son, únicamente, sistemas que se optimizan al ir aprendiendo de la experiencia, y que toman como base la interconectabilidad entre neuronas ya que es donde reside la potencia del cerebro. La nomenclatura que se utiliza no debe llevar a confusión, aunque es conveniente tener algún conocimiento de neurofisiología para comprender mejor por qué se escogen determinadas arquitecturas para las redes, y qué modelos se intentan simular.

La Figura 1.1 representa los componentes principales de una célula nerviosa típica perteneciente al sistema nervioso central. La membrana de la neurona separa el plasma intracelular del fluido intersticial que se encuentra fuera de las neuronas, manteniendo una diferencia de potencial entre el fluido intracelular y el fluido extracelular.

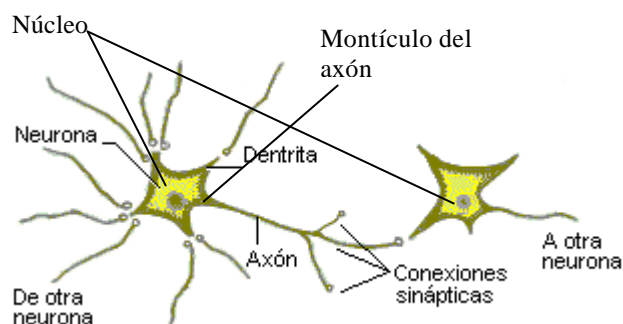


Figura 1.1. Estructuras fundamentales de una célula nerviosa.

Las entradas excitatorias que llegan a la célula reducen la diferencia de potencial que existe entre los dos lados de la membrana celular. La despolarización resultante en el montículo del axón altera la permeabilidad de la membrana lo que acaba resultando en una mayor despolarización. Este efecto da lugar al **potencial de acción**. Las fibras nerviosas son malos conductores por lo que la transmisión del potencial de acción a lo largo del axón es el resultado de una serie de despolarizaciones que tienen lugar en los **nodos de Ranvier** (puntos característicos situados en el **axón** de la neurona). Cuando uno de los nodos se despolariza, se desencadena la despolarización del siguiente nodo. El potencial de acción viaja a lo largo de la fibra en modo discontinuo, de un nodo a otro. Una vez que un potencial de acción ha pasado por un cierto punto, éste no puede volver a ser excitado durante un tiempo aproximado de 1 milisegundo, que es el tiempo que tarda en volver a su potencial de reposo. Esto limita la frecuencia de transmisión de los impulsos nerviosos a unos 1000 por segundo.

La unión entre dos neuronas se llama unión sináptica o **sinapsis**. La comunicación entre las neuronas tiene lugar como resultado de la liberación de unas sustancias llamadas **neurotransmisores** por parte de la célula presináptica, al ser absorbidas estas sustancias por la célula postsináptica.

Los neurotransmisores se difunden a través de la unión y se unen a la membrana postsináptica en ciertos lugares llamados receptores. La acción química que se produce en los receptores da lugar a cambios de permeabilidad de la membrana postsináptica para ciertas especies iónicas. Un flujo entrante de especies positivas hacia la célula tenderá a despolarizar el potencial de reposo; este efecto es excitatorio. Si entran iones negativos, el efecto será inhibitorio. Estos dos efectos son locales y actúan solamente a lo largo de una pequeña distancia hacia el interior de la célula, sumándose en el montículo del axón. Si la suma es mayor que un cierto valor umbral se genera un potencial de acción.

En la Figura 1.2 se muestran varias estructuras neuronales básicas que se encuentran en el sistema nervioso central. En la Figura 1.2 a y b se ilustran los principios de convergencia y divergencia de la topología neuronal. Cada neurona envía impulsos a muchas otras neuronas (divergencia) y recibe impulsos de otras muchas

neuronas (convergencia). Este hecho tan sencillo parece ser el fundamento de toda la actividad del sistema nervioso central.

En la Figura 1.2 b, c y d aparecen circuitos de realimentación. Como las conexiones sinápticas pueden ser excitatorias o inhibitorias, estos circuitos hacen que los sistemas de control puedan tener tanto realimentación positiva como negativa.

McCulloch y Pitts, en 1943 realizaron el primer intento significativo para explicar de qué manera se llegaba a la enorme potencia del cerebro humano desde las neuronas básicas, su funcionamiento y su interconexión; es decir, que fueron los pioneros en tratar al cerebro como un organismo computacional. No se trataba de una teoría precisa pero sí que fue de suma importancia en posteriores trabajos debido al sentido computacional que se le confirió al cerebro humano [Freeman-93].

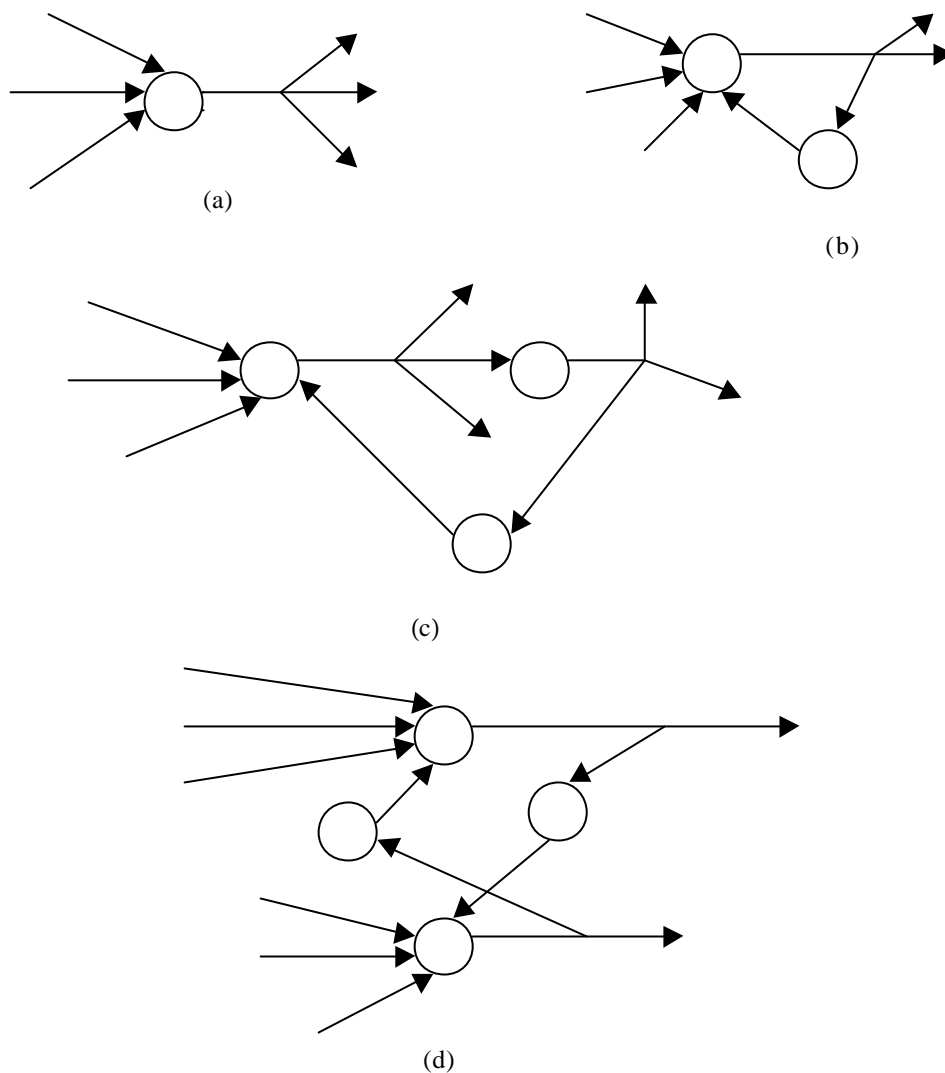


Figura 1.2. Ejemplos de estructuras neuronales del sistema nervioso central.

Anderson y Rosenfeld pusieron de manifiesto una idea fundamental que no se reflejaba en el artículo de McCulloch-Pitts: Aunque las neuronas son dispositivos sencillos, se puede obtener una gran potencia de cálculo cuando se interconectan adecuadamente dentro del sistema nervioso [Anderson-88].

Aprendizaje de Hebb

Los sistemas neuronales no nacen con el conocimiento y capacidades que llegarán a tener. Existe un proceso de aprendizaje que modifica la red para incluir la información.

En 1949, Hebb publicó *Organization of Behaviour*. En este libro reside la idea fundamental del aprendizaje de Hebb, y que es la siguiente [Hebb-49]:

Cuando un axón de la célula A está suficientemente próximo para excitar a una célula B o toma parte en su excitación de forma persistente, tiene lugar algún proceso de crecimiento o algún cambio metabólico en una de las células, o en las dos, de tal modo que la eficiencia de A, como una de las células que desencadena el disparo de B, se ve incrementada.

Aunque esta idea no explica todo lo que hace referencia a las conexiones neuronales, tiene una vital importancia por cuanto aparece en la mayoría de los modelos utilizados.

Como la conexión entre neuronas se hace a través de las sinapsis, se puede suponer que cualquier cambio durante el aprendizaje deberá producirse en ellas. Teorías recientes afirman que el responsable del aprendizaje es un incremento de la velocidad con que se libera el neurotransmisor en la célula presináptica [Freeman-93].

Por último, para terminar este repaso de neurofisiología, ha de quedar claro que a pesar de las muchas analogías entre conceptos básicos de la neurofisiología y los modelos de redes neuronales, estos últimos no son verdaderos modelos del cerebro, sino que han sido inspirados por la comprensión que se tiene de la neurofisiología.

1.1.2. Nociones básicas

Una RNA es un conjunto de unidades simples llamadas **nodos** o **neuronas** unidas entre sí por medio de **conexiones**. Cada elemento trabaja sólo con información local y asíncronamente. El conocimiento se almacena en las conexiones o **pesos sinápticos**. Estos pesos serán ajustados de acuerdo con los **patrones** de que dispongamos mediante una **regla de aprendizaje**. A esta parte del proceso en que la RNA aprende los patrones se le conoce como **entrenamiento**. El conocimiento adquirido en esta fase se refleja en unos determinados valores de los pesos sinápticos con los cuales la red debe ser capaz de generalizar, es decir, de dar la salida correcta ante una determinada entrada no vista anteriormente [Nigrin-93].

Resumiendo, mediante unos patrones de entrenamiento se consigue que la red aprenda, o lo que es lo mismo, que optimice sus conexiones para ser capaz de ofrecer la salida deseada ante una determinada entrada.

La implementación de las redes puede plantearse desde dos puntos de vista:

- *Software*: para aplicaciones sencillas, donde el tiempo de decisión no sea excesivamente elevado.
- *Hardware*: para aplicaciones que precisen alta velocidad de decisión [Kung-95].

1.2. SISTEMAS BASADOS EN LÓGICA DIFUSA

Un sistema basado en lógica difusa (*Fuzzy Logic System, FLS*) es el único tipo de sistema capaz de tratar simultáneamente con **variables numéricas** y con **variables lingüísticas** de modo formal. Las variables numéricas, que son las más habituales, pueden caracterizarse por un valor numérico, por ejemplo, la temperatura del ambiente es 35 °C. Las variables lingüísticas, sin embargo, se caracterizan por un adjetivo que las califica, por ejemplo, la temperatura del ambiente es alta.

Un **FLS** se basa en un mapeo no lineal de un vector de entrada en una salida escalar. La **Teoría de Conjuntos Difusos** y la **Lógica Difusa** establecen las especificaciones de este mapeo no lineal. Un FLS puede expresarse matemáticamente como una combinación lineal de **funciones base difusas**, y es un aproximador universal no lineal de funciones, propiedad que comparte con el perceptrón multicapa. El desarrollo en funciones base difusas es muy potente porque éstas pueden obtenerse a

partir de datos numéricos o bien conocimiento lingüístico; en ambos casos, se puede presentar en la forma de **reglas If-Then** [Mendel-95].

Existen dos tipos de conocimiento sobre un determinado problema:

- *Conocimiento objetivo*: cuantificado habitualmente mediante modelos matemáticos. Ejemplos: ecuaciones de movimiento de un robot, modelos convolucionales que describen un canal de comunicaciones, etc.
- *Conocimiento subjetivo*: contiene información lingüística que no es posible cuantificar mediante modelos matemáticos tradicionales. Un ejemplo de este tipo sería la siguiente regla que puede ser válida para seguir la pista de un objeto de grandes dimensiones moviéndose lentamente: *Si el objeto está localizado en un lugar en un instante temporal, en el siguiente instante no estará demasiado lejos de ese lugar.*

Aunque el conocimiento subjetivo suele ser ignorado a la hora de enfrentarnos a un problema, sí que se tiene en cuenta para evaluar la solución que se le da. Esto sugiere la posibilidad de utilizar ambos tipos de conocimiento para resolver problemas reales, que es lo que hace la Lógica Difusa o **Fuzzy Logic (FL)**.

En la literatura aparecen dos posibles formas de abordar el problema:

- *Basada en modelos*: la información objetiva se representa por modelos matemáticos, y la información subjetiva por afirmaciones lingüísticas convertidas en reglas, que son cuantificadas usando FL.
- *Libre de modelos*: las reglas se obtienen de los datos numéricos y se combinan con información lingüística (ofrecida por el experto) usando FL.

Este proyecto se centrará en la segunda, ya que puede aplicarse al mismo tipo de problemas que las RNAs.

Como se explicó en la introducción, un FLS es el mapeo no lineal de un vector de entrada en una salida escalar. La potencia de la FL es que hay muchas posibilidades distintas que conducen a mapeos diferentes [Mendel-95].

En cuanto a las razones para la utilización de la FL, hay que destacar que las clases imprecisamente definidas representan un importante papel en el pensamiento humano, particularmente en lo que concierne a reconocimiento de patrones, comunicación de información y capacidad de abstracción [Zadeh-65]. Además, ha de tenerse en cuenta el **Principio de Incompatibilidad**: “ *Conforme la complejidad de un sistema aumenta, nuestra capacidad para realizar afirmaciones precisas o incluso*

significantes sobre su comportamiento disminuye hasta un umbral más allá del cual la precisión y la relevancia se convierten prácticamente en características mutuamente exclusivas “ [Mendel-95].

Este hecho se puede entender mejor mediante un ejemplo: supongamos un sistema sencillo que conste de un objeto que se mueva hacia la izquierda o hacia la derecha dependiendo de que su temperatura sea mayor o menor que la temperatura ambiente; es fácil pues hacer afirmaciones precisas sobre su comportamiento, ya que se moverá hacia un lado u otro dependiendo de la temperatura. Si consideramos un sistema más complejo que consta de un objeto que puede moverse en cualquier dirección y a distintas velocidades dependiendo de su temperatura, presión y humedad en comparación con los del ambiente, las afirmaciones que hagamos sobre su comportamiento no pueden ser muy precisas, y si se intentan hacer no serán excesivamente relevantes, ya que una afirmación precisa sería por ejemplo dar en qué dirección y a qué velocidad se mueve el objeto para unos determinados valores de temperatura, presión y humedad, pero esto no es relevante ya que no da una perspectiva general sobre el comportamiento del sistema sino que se centra en una situación muy determinada.

En la Figura 1.3 se representa un FLS ampliamente usado en controladores y aplicaciones de Procesado de Señales basados en FL. Un FLS mapea entradas convencionales (no difusas) en salidas convencionales. Contiene 4 componentes: **reglas, difusor, bloque de inferencia y dedifusor.**

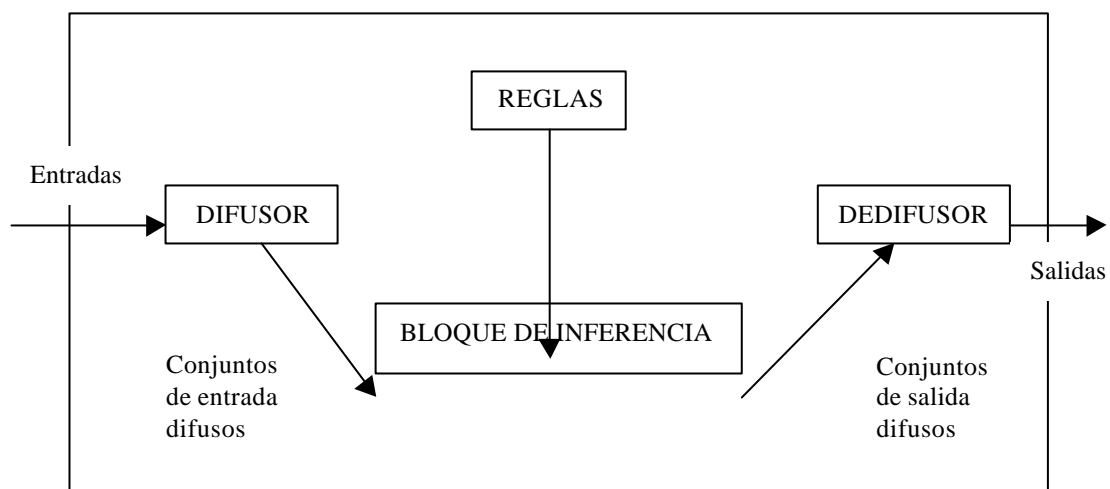


Figura 1.3. Sistema basado en Lógica Difusa (FLS).

El *difusor* mapea números convencionales en conjuntos difusos, lo cual es necesario para activar reglas que están en términos lingüísticos, y que tienen conjuntos difusos asociados con ellas.

El *bloque de inferencia* mapea conjuntos difusos en conjuntos difusos y se encarga de combinar las reglas.

El *dedifusor* mapea conjuntos de salida en números, debido a que en muchas aplicaciones, a la salida del FLS se desea obtener un número.

Las reglas, que pueden ser facilitadas por expertos o bien ser deducidas a partir de los datos numéricos, son una colección de sentencias IF-THEN, por ejemplo: “*Si a está caliente y b está bajo, entonces gira c algo hacia la derecha*”. Esta única regla revela que necesitamos tener conocimiento de:

- 1) Correspondencia entre variables lingüísticas y valores numéricos de una variable (por ejemplo, muy caliente se corresponde con 36°C).
- 2) Cuantificación de variables lingüísticas (por ejemplo, *a* puede tener asociado un número finito de términos lingüísticos asociados entre *extremadamente caliente* y *extremadamente frío*), lo cual se consigue mediante las **funciones de partición**, que se explicarán con posterioridad.
- 3) Conexiones lógicas para las variables lingüísticas.
- 4) Implicaciones.
- 5) Además, será necesario conocer cómo combinar más de una regla.

Aplicaciones

La FL tiene un amplio abanico de aplicaciones: problemas de control, clasificación, modelización, reconocimiento de patrones, diagnosis, etc. En la siguiente lista se exponen aplicaciones reales, así como las empresas o instituciones donde se están utilizando [Mendel-95]:

- 1) Aplicaciones de Control: control de aviones (*Rockwell Corporation*), control del metro de Sendai (*Hitachi*), control de crucero (*Nissan*), transmisión automática (*Nissan, Subaru*), modelo para autoaparcamiento de coches (*Tokyo Tech. Univ.*), y situación de una lanzadera espacial (*NASA*).
- 2) Catalogación y optimización: análisis del stock en el mercado (*Yamaichi Securities*), ascensor inteligente (*Hitachi, Fujitech, Mitsubishi*).

- 3) Análisis de señales: ajuste del color de TV (*Sony*), reconocimiento de caligrafía (*Sony Palm Top*), autoenfoco de videocámaras (*Sanyo/Fisher, Canon*), estabilizador de la imagen de video (*Matshushita/Panasonic*).

El campo de la FL está muy abierto y, por tanto, continuamente surgen nuevas aplicaciones. En particular destaca la conexión de las RNAs con la FL que permite su utilización en aquellos campos típicos de las RNAs; éste será el objetivo del presente proyecto.

Conjuntos difusos

Definición de conjunto

Sea A un conjunto en un dominio D . A puede definirse identificando todos aquellos elementos $x \in A$. Es decir, que A puede definirse como:

$$A = \{x/x \text{ cumple cierta condición}\}$$

También es posible definirlo mediante una función binaria que valga 1 si el elemento pertenece al conjunto y 0 si no pertenece.

Consideremos a modo de ejemplo, el conjunto de todos los números primos P en el dominio de los números naturales N . Entonces, podremos definir P como:

$$P = \{x \in N/x \text{ es primo}\}$$

o bien mediante la siguiente función binaria:

$$m(x) = \begin{cases} 1 & \text{si } x \text{ es primo} \\ 0 & \text{en otro caso} \end{cases}$$

Definición de conjunto difuso

Un **conjunto difuso** F en un dominio D se caracteriza por una función, llamada función de partición $m(x)$ que toma valores en el intervalo $[0,1]$. Es decir, que en el fondo un conjunto difuso es una generalización de un conjunto ordinario cuya función de partición solamente toma dos valores, cero o la unidad. La función de partición da una medida del grado de similitud de un elemento en D con el conjunto difuso.

Supongamos un ejemplo, la clasificación de un coche en nacional o extranjero. Un coche puede ser visto como nacional desde diferentes perspectivas. Una posibilidad es considerar que un coche es nacional si lleva el nombre de una empresa automovilística

española y extranjero en otro caso. No hay nada difuso en esta perspectiva. Sin embargo, si consideramos que un coche es nacional si está íntegramente fabricado en España la cosa cambia porque muchos de los componentes de la única marca española (SEAT) se fabrican en el extranjero. Además, muchas marcas extranjeras fabrican coches en España (Ford, Opel, Citroën, Renault,...). Lo que se puede hacer es considerar un coche más o menos nacional dependiendo del porcentaje de componentes fabricados en España. Podríamos pensar en considerar dos funciones de partición, una para coches nacionales y otra para coches extranjeros. En la Figura 1.4 puede observarse como un coche determinado pertenece simultáneamente a los dos conjuntos descritos por estas funciones de partición pero con diferente grado de similitud. Por ejemplo, si nuestro coche está realizado en un 75% en España, entonces:

$$\mu_N(75\%)=0.9 \text{ y } \mu_F(75\%)=0.25$$

que representan las funciones que describen lo coches nacionales y foráneos respectivamente. En este caso, podríamos decir que el coche es nacional. Lo principal es observar que un elemento puede estar en más de un conjunto, situación que no es posible en Lógica Ordinaria.

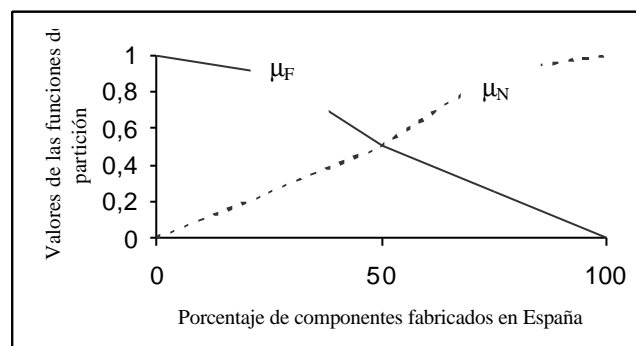


Figura 1.4. Funciones de partición para coches nacionales y foráneos basadas en el porcentaje de componentes fabricados en España.

Un conjunto difuso F puede representarse como un conjunto de pares ordenados de un elemento genérico x y el valor que toma la función de partición para este elemento, es decir, el grado de similitud del elemento con el conjunto:

$$F=\{(x, \mu_F(x))/x \in D\}$$

Cuando D es continuo, la notación habitual es:

$$F = \int_D \mathbf{m}_F(x)/x$$

donde el signo integral no denota integración, sino el conjunto de puntos $x \in D$ con la función de partición correspondiente. Análogamente cuando D es discreto, es habitual la notación:

$$F = \sum_D \mathbf{m}_F(x)/x$$

Variables lingüísticas

Zadeh, “*el padre de la Lógica Difusa*”, afirma textualmente que:

“Al perder precisión al crecer la complejidad, es natural estudiar el uso de las llamadas variables lingüísticas, es decir, variables cuyos valores no son números sino palabras o frases en un lenguaje natural o artificial,... La motivación del uso de palabras o frases en lugar de números es que las caracterizaciones lingüísticas son, en general, menos específicas que las numéricas” [Zadeh-65]

Denotamos por d una variable lingüística (por ejemplo, la temperatura). Los valores numéricos de una variable lingüística los expresamos por x , donde $x \in D$. Algunas veces se usan indistintamente x y d , especialmente cuando la variable lingüística es una letra. En principio, la caracterización de este tipo de variables la lleva a cabo el diseñador.

Una variable lingüística se descompone habitualmente en un conjunto de términos, $T(d)$, que cubren su dominio.

Ejemplo: Interpretemos la *temperatura* (d) como una variable lingüística. Puede hacerse la descomposición en el siguiente conjunto de términos:

$$T(\text{temperatura}) = \{\text{muy baja, baja, normal, alta, muy alta}\}$$

donde cada término en $T(\text{temperatura})$ se caracteriza por un conjunto difuso en el dominio $D = [-25^\circ\text{C}, 50^\circ\text{C}]$. Podemos interpretar *muy baja* como una temperatura inferior a -10°C , *baja* alrededor de los 0°C , *normal* como próxima a los 18°C , *alta* alrededor de los 30°C y *muy alta* si es superior a los 40°C . Estos términos pueden caracterizarse como conjuntos difusos cuyas funciones de partición se muestran en la Figura 1.5. Podemos observar que, por ejemplo, $x = -15^\circ\text{C}$ pertenece a los conjuntos difusos *baja temperatura* y *muy baja temperatura* pero con diferentes grados de similitud.

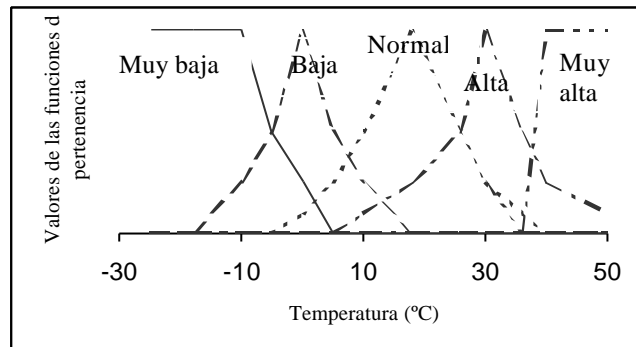


Figura 1.5. Funciones de partición para el ejemplo.

Funciones de partición

Las funciones de partición (*membership functions*, MF) están, en su mayor parte, asociadas con términos que aparecen en las reglas, ya sea en la parte antecedente o en la consecuente [Mendel-95]. La parte antecedente de una regla es la formada por las distintas premisas de ésta, mientras que la parte consecuente es la constituida por las consecuencias. Consideremos como ejemplo la siguiente regla:

Si la temperatura alcanza 50 °C encender el ventilador de refresco

La parte antecedente estaría formada por una única premisa, que la temperatura alcance los 50 °C. La parte consecuente sería la orden de encender el ventilador de refresco.

Las formas mayormente usadas para las MFs son la triangular, trapezoidal, lineal a trozos, gaussiana o con forma de campana (algo más cuadrada que la gaussiana y la más utilizada en redes neuro-difusas). Aunque lo habitual es elegir unas u otras basándonos en la propia experiencia, existen algunos procedimientos que permiten optimizar esta decisión [Mendel-95].

Cuanto mayor es el número de funciones mayor posibilidad de alcanzar una mayor resolución aunque, evidentemente, acarreará un mayor coste computacional, de tal manera que en cada caso se ha de llegar a la solución óptima por compromiso.

Aunque no es necesario que las MFs se solapen, gran parte de la potencia de la FL reside en esta posibilidad. En este sentido existe la capacidad de distribuir las decisiones sobre más de una clase de entrada, lo que permite implementar FLS más robustos.

A pesar de que las funciones no tienen por qué estar escaladas entre cero y la unidad, se suele hacer así. Siempre se puede normalizar un conjunto difuso dividiendo $\mu(x)$ por su valor máximo o bien se pueden seguir otras estrategias de normalización.

Nociones básicas de Lógica Difusa

Uno de los componentes principales de un FLS es el de las reglas. Nuestras reglas se expresarán como implicaciones lógicas en las formas de sentencias IF-THEN. Por ejemplo, si u es A entonces v es B . Una regla representa un tipo especial de relación entre A y B , cuya MF se denota por $\mu_{A \rightarrow B}(x,y)$. Para la elección adecuada de esta función se debe recurrir a la FL.

La extensión de la Lógica Ordinaria a la Lógica Difusa se realiza sustituyendo las funciones bivalentes por funciones difusas. Démonos cuenta que la MF da idea del grado en qué es verdad la relación de implicación entre x e y . Algunas funciones a tal efecto son:

$$\mu_{A \rightarrow B}(x,y) = 1 - \min[\mu_A(x), 1 - \mu_B(y)] \quad (1.1.1)$$

$$\mu_{A \rightarrow B}(x,y) = \max[1 - \mu_A(x), \mu_B(y)] \quad (1.1.2)$$

$$\mu_{A \rightarrow B}(x,y) = 1 - \mu_A(x)(1 - \mu_B(y)) \quad (1.1.3)$$

En Lógica Ordinaria, existen dos reglas de inferencia fundamentales: *Modus Ponens* y *Modus Tollens*.

Modus Ponens:

Premisa 1: “ x es A ”

Premisa 2: “si x es A , entonces y es B ”

Consecuencia: “ y es B ”

Modus Tollens:

Premisa 1: “ y es no B ”

Premisa 2: “si x es A entonces y es B ”

Consecuencia: “ x es no A ”

Pues bien, en FL, existe una versión generalizada de estas dos inferencias. *Modus Ponens Generalizado* dice que:

Premisa 1: “ u es A^* ”

Premisa 2: "si u es A, entonces v es B"

Premisa 3: "v es B*"

Del mismo modo se generaliza *Modus Tollens*. Aquí lo importante es que los conjuntos difusos A^* y B^* no tienen por qué ser los mismos que A y B . En Lógica Tradicional, la regla solamente era válida si la primera premisa era exactamente igual al antecedente de la regla, y el resultado era el consecuente de la regla. En FL, una regla vale mientras haya un grado de similitud no nulo entre la primera premisa y el antecedente de la regla, siendo el resultado un consecuente que tiene un grado de similitud no nulo con el consecuente de la regla.

Puede demostrarse, que las expresiones (1.1.*) no son adecuadas para expresar la MF asociada a la regla desde un punto de vista práctico, ya que puede romperse la relación de causalidad, por lo que estaríamos tratando con algo que no es implementable físicamente.

Mamdani, movido por razones de coste computacional, propone trabajar con la **implicación de mínimo** o *minimum implication* [Mamdani-74]:

$$\mu_{A \rightarrow B}(x,y) = \min[\mu_A(x), \mu_B(y)] \quad (1.2)$$

Larsen, también motivado por razones de coste computacional, propone la **implicación de producto** o *product implication* [Larsen-80]:

$$\mu_{A \rightarrow B}(x,y) = \mu_A(x) \cdot \mu_B(y) \quad (1.3)$$

En la actualidad, son estas dos inferencias las más utilizadas en aplicaciones de FL, sobre todo porque preservan la causalidad. Como se observa en la Tabla 1.1, ninguna de estas dos inferencias está de acuerdo con la Lógica Proposicional Clásica, considerando la siguiente proposición:

La salida es 0 si $\mu_2(y) > \mu_1(x)$; es 1 en cualquier otro caso.

Tabla 1.1: Demostración de que el mínimo y el producto están en desacuerdo con la Lógica Proposicional Tradicional.

m1(x)	m2(y)	min[m1(x), m2(y)]	m1(x)·m2(y)	Lógica Trad.
1	1	1	1	1
1	0	0	0	0
0	1	0	0	1
0	0	0	0	1

1.3. REDES NEURO-DIFUSAS

Las **redes neuro-difusas** son sistemas que incorporan aspectos de las **redes neuronales** en el sentido de que son sistemas con la capacidad de aprender y generalizar, y aspectos de la **Lógica Difusa** ya que trabajan con razonamientos lógicos basados en reglas de inferencia, incorporan la posibilidad de trabajar con variables lingüísticas (si el problema lo requiere) y, además, cambian el tratamiento binario del problema que hacen las **RNAs** por un tratamiento difuso.

Ya se ha visto que el cerebro humano permite interpretar información sensorial incompleta e imprecisa a través de sus órganos perceptivos. La **Teoría de Conjuntos Difusos** es un método sistemático para trabajar con esta información lingüística, pudiéndose interpretar la computación numérica usando etiquetas lingüísticas estipuladas por las **funciones de partición**. Además, una selección adecuada de **reglas If-Then** es el componente primordial de un **Sistema de Inferencia Difuso** o *Fuzzy Inference System (FIS)* que permite modelizar el conocimiento experto humano para una aplicación específica. Aunque un **FIS** tiene la representación del conocimiento estructurada en forma de **reglas If-Then**, carece de la adaptabilidad necesaria para cambiar cuando se modifiquen las condiciones externas. Por esta razón, se incorporan conceptos del aprendizaje de las **RNAs** a los **Sistemas de Inferencia Difusos**, cuyo resultado es un nuevo tipo de topologías conocidas como **redes neuro-difusas (RNDs)** [Jang-97].

Características

Las redes neuro-difusas tienen fundamentalmente las siguientes características:

- 1.- *Conocimiento humano*: se utiliza en forma de **reglas If-Then** para solucionar problemas reales.
- 2.- *Inspiración en modelos biológicos*: inspiradas por las redes neuronales biológicas, las **RNAs** son un pilar básico de las **RNDs**, permitiendo tratar con problemas de percepción, reconocimiento de patrones o clasificación.
- 3.- *Computación numérica*: la computación es básicamente numérica, aunque la perspectiva futura apunta hacia una computación simbólica más cercana a la **Inteligencia Artificial (IA)**.

- 4.- *Diversos dominios de aplicación*: procesado adaptativo de señales, control adaptativo, identificación de sistemas no lineales, regresión no lineal, reconocimiento de patrones, etc.
- 5.- *Aprendizaje libre de modelos*: las reglas se obtienen a partir de los datos numéricos.
- 6.- *Computación intensiva*: sin asumir un conocimiento *a priori* del problema, las **RNDs** dependen mucho de una computación intensiva que permita encontrar reglas o cualquier tipo de regularidad en conjuntos de datos.
- 7.- *Tolerancia a fallos*: tanto las **RNAs** como los **FISs** tienen tolerancia a fallos. La supresión de una neurona en una RNA o de una regla en un FIS, no tiene por qué destruir el sistema. En realidad, el sistema continúa trabajando debido a su arquitectura paralela y redundante aunque se va deteriorando gradualmente.
- 8.- *Distintos caminos para alcanzar el mínimo*: se puede llegar al mínimo error por diversos caminos. De hecho, esto no es importante en la medida en que el sistema vaya acercándose hacia el mínimo.
- 9.- *Muy indicadas en la resolución de problemas reales*: la mayoría de los problemas reales son de una magnitud considerable y se construyen sobre afirmaciones que no se conocen con total certeza. Esto excluye, por tanto, la utilización de métodos convencionales que requieren una descripción detallada del problema a resolver. Las **RNDs** permiten tratamientos que dan soluciones satisfactorias a los problemas del mundo real. De hecho, el campo de aplicación de estos sistemas crece constantemente, aplicándose a nuevas situaciones donde no se han obtenido resultados positivos con otros métodos.

Evolución histórica

Las **RNDs** son un campo de investigación que se encuentra en pleno apogeo, ya que su estudio y aplicación son todavía recientes. En la Tabla 1.2 se muestra una comparativa entre el origen y evolución de las **RNAs** y las **RNDs** [Jang-97]:

Tabla 1.2: Evolución histórica de las redes neuronales y los sistemas difusos.

Años	Redes neuronales	Sistemas difusos
1940s	1943: modelo neuronal de McCulloch-Pits	
1950s	1957: perceptrón	
1960s	1960s: Adalina Madalina	1965: Conjuntos difusos
1970s	1974: Nacimiento del algoritmo <i>Backpropagation</i> 1975: Cognitrón Neocognitrón	1974: Controlador difuso
1980s	1980: Mapeo autoorganizativo 1982: Red de Hopfield 1983: Máquina de Boltzmann 1986: <i>Boom</i> del algoritmo <i>Backpropagation</i>	1985: Modelización difusa (modelo TSK)
1990s		1990: Modelización neuro-difusa 1991: ANFIS 1994: CANFIS Actualidad: generalización difusa de modelos de RNAs

CAPÍTULO 2. DESCRIPCIÓN DE LAS REDES UTILIZADAS.

RESUMEN DEL CAPÍTULO

En este capítulo, se realiza una descripción de las redes neuro-difusas utilizadas en el proyecto. Se pueden dividir en dos grandes grupos:

- *Redes inherentemente difusas que se desarrollan sobre la base de un FIS: ANFIS, MANFIS.*
- *Redes basadas en RNAs tradicionales, en particular en el Perceptrón Multicapa o **Multilayer Perceptron (MLP)**. En este caso, se han ido haciendo difusas algunas partes del MLP como por ejemplo, las neuronas o la función de costes (concepto que explicaremos más adelante).*

2.1. REDES BASADAS EN SISTEMAS DIFUSOS

2.1.1. ANFIS (*Adaptive Neuro-Fuzzy Inference Systems*)

ANFIS es un tipo de red adaptativa funcionalmente equivalente a un FIS. El nombre de esta red se corresponde con las siglas de *Adaptive Network-based Fuzzy Inference System* aunque generalmente se considera que son las siglas de *Adaptive Neuro-Fuzzy Inference Systems*, que, semánticamente, es equivalente a la primera posibilidad.

Arquitectura

La red ANFIS tiene una estructura de cinco capas como la mostrada en la figura 2.1, donde los nodos circulares representan nodos fijos y los cuadrados nodos adaptativos. Para una mejor comprensión de la función que desempeña cada una de estas capas, suponemos un *Sistema de Inferencia Difuso* sencillo con dos entradas (x e y) y una salida z [Jang-97]. Suponemos además un conjunto de dos reglas *If-Then*:

- Regla 1: Si x es A_1 e y es B_1 , entonces $f_1 = p_1x + q_1y + r_1$.
 - Regla 2: Si x es A_2 e y es B_2 , entonces $f_2 = p_2x + q_2y + r_2$.