

**Extracción de reglas
en redes neuronales difusas
Un modelo basado en la
discretización de los conjuntos
difusos de entrada
(propuesta del modelo BCD)**

**Extracción de reglas
en redes neuronales difusas
Un modelo basado en la
discretización de los conjuntos
difusos de entrada
(propuesta del modelo BCD)**

ANTONIO PONCE ROJO



UNIVERSIDAD DE GUADALAJARA
Centro Universitario de la Costa
División de Ingenierías
Departamento de Informática

Primera edición, 2005

© D.R. Antonio Ponce Rojo

© D.R. Universidad de Guadalajara
Centro Universitario de la Costa
División de Ingenierías
Departamento de Informática
Puerto Vallarta, Jalisco, México

ISBN 968-7846-92-5

Impreso y hecho en México
Printed and made in Mexico

Índice

Resumen	9
Agradecimientos	11
1. Introducción	13
2. Lógica Difusa	21
3. Redes neuronales.....	55
4. Extracción de reglas en redes neuronales nítidas y redes neuronales difusas	83
5. La extracción de reglas en redes neuronales difusas con base en la Binarización de los Conjuntos Difusos. La propuesta del Modelo BCD	117
6. Evaluación del modelo BCD. Metodología empleada	129
7. Evaluación del modelo BCD. pruebas y resultados obtenidos	143
8. Conclusiones	193
Anexo A. La clasificación de la planta iris.....	203
Anexo B. Conjunto de reglas iniciales extraídas por el Modelo de extracción de reglas difusas basado en la Binarización de los Conjuntos Difusos (BCD)	209
Bibliografía	211

Resumen

Una ventaja que ofrecen los sistemas cuya representación del conocimiento se da en un nivel simbólico, frente a los sistemas con representaciones subsimbólicas, es su mayor comprensibilidad para el ser humano. Hoy en día, las redes neuronales aún son consideradas como la “caja negra” de la inteligencia artificial, ya que resulta muy difícil comprender para el ser humano, la forma en la que éstas llegan a determinar la salida indicada ante un conjunto de patrones presentados. Este problema de falta de comprensibilidad en las redes neuronales, ha propiciado que éstas no sean usadas en campos en los que podría parecer natural su intervención en el reconocimiento de patrones complejos, cuando, se requiere no solamente el reconocimiento, sino una justificación del proceso realizado. Si un sistema inteligente resuelve efectivamente un problema, se puede hablar de cierto grado de éxito, sin embargo, la comprensibilidad ofrece ventajas adicionales como las siguientes: a) permite al usuario conocer exactamente cómo es que el sistema ha llegado a una conclusión determinada, b) permite al humano tener un conocimiento preciso acerca de cuál es el conjunto de conocimientos a partir de los cuales el sistema está realizando los procesos de inferencia en la resolución de problemas.

En los últimos años se han intentado diferentes estrategias para dotar de comprensibilidad a los sistemas subsimbólicos en general, incluyendo entre éstos a las redes neuronales, como una de las puntas de lanza del procesamiento subsimbólico. De entre los esfuerzos realizados sobresalen la generación de árboles de conocimientos y la extracción de reglas simbólicas; que han ofrecido resultados exitosos, pero solamente en ámbitos muy restringidos de aplicación.

En general, los intentos realizados a la fecha para “abrir esta caja negra” presentan aún ciertas desventajas: en algunos casos el procedimiento es muy complicado; los recursos de cómputo requeridos para poder realizar el procesamiento son muy demandantes; o incluso, los resultados son poco confiables. Estos logros son más limitados aún cuando se con-

sidera la búsqueda de comprensibilidad en sistemas híbridos, es decir, cuando se busca obtener el conocimiento de sistemas que articulan la tecnología de las redes neuronales, con alguna otra tecnología, como la lógica difusa.

Esta investigación estableció como objetivo principal, aportar hacia la resolución de este problema realizando un modelo propio para extracción de reglas en redes neuronales difusas, probándolo y evaluando sus resultados, comparándolos con los productos que ofrecen otros modelos de extracción de reglas, reconocidos por la literatura especializada internacional como los más exitosos a la fecha. El modelo desarrollado se basa en una estrategia de simplificación de los conjuntos difusos que se ofrecen como entrada en una red neuronal difusa, discretizando sus valores con base en los grados de pertenencia de dicho valor, a diferentes conjuntos difusos. Esta estrategia permite reducir las demandas en cuanto a procesamiento se refiere, sin un menoscabo significativo en la calidad de los resultados obtenidos, tal y como la investigación lo demostró.

Este modelo de extracción de reglas basado en la discretización de los conjuntos de entrada a la red fue llamado por el autor, Modelo de Binarización de los Conjuntos Difusos (BCD). Esta investigación, no solamente se enfocó al diseño del Modelo, sino también a su prueba, para lo que fue sometido a la resolución de un problema de Clasificación que se usa como punto de referencia para evaluación de modelos de extracción de reglas (Andrews, Diederchich y Tickle, 1995): el problema de clasificación de la planta *iris* (Fisher, 1935). Comparándose los resultados y desempeño con los logrados por dos de los modelos que han sido, en estos últimos años, de los más reconocidos a escala internacional en el campo de la extracción de reglas en redes neuronales difusas: los modelos FUNEGEN¹ (Halgamuge y Glesner, 1994) originalmente propuesto en 1994 y reformulado a lo largo del tiempo, incluyéndose la versión más reciente en el 2004 y NEFCLASS² (Knauck y Kruse, 1994) que también fue desarrollado en 1994 y cuenta con una revisión en el 2003 y 2004.

El Modelo BCD demostró ser capaz de obtener reglas efectivas, comprensibles y generales a partir de un conjunto de datos establecidos, pero no sólo eso; demostró, además, que es posible alcanzar iguales o mejores niveles de desempeño que los modelos contemporáneos con los que fue comparado, a un menor costo en cuanto a recursos y complejidad de procesamiento se refiere.

1 *Fuzzy Neuro Generator System*, en inglés (Sistema Difuso Neuro Generador).

2 *Neuro Fuzzy Classifier System*, en inglés (Sistema Clasificador Neurodifuso).

Agradecimientos

Para la realización de esta obra, es invaluable la ayuda recibida de las siguientes personas a las cuales el autor expresa su agradecimiento infinito, el Lic. José Trinidad Padilla López, Rector General de la Universidad de Guadalajara; el M en C. Jeffry S. Fernández Rodríguez, Director General de Difusión Cultural y Ex Rector del Centro Universitario de la Costa; el Ing. Edmundo Sánchez Medina, Director de la División de Ingenierías del Centro Universitario de la Costa; la Dra. Sonia Reynaga Obregón, Jefa de la Unidad para el Desarrollo de la Investigación y el Posgrado de la Universidad de Guadalajara; el Dr. Javier Orozco Alvarado, Rector del Centro Universitario de la Costa de la Universidad de Guadalajara; el Dr. Edgar Vallejo Clemente; la M en C Cintia Elizabeth Castro Sánchez; la Sra. Zenaida Rojo Vda. de Ponce; el Ing. Pedro Ponce Rojo; el Ing Juan Carlos Ponce Rojo y el Arq. Arturo Ponce Rojo.

1

Introducción

La extracción de reglas es un tema que tiene relativamente poco tiempo de ser estudiado en el mundo entero. Sin embargo, es creciente el número de avances logrados en los últimos años (Andrews, Diederich y Tickle, 2001). Los estudios en este campo corren a la par de los trabajos relacionados sobre el aprendizaje de máquina, la minería de datos, la minería del conocimiento y la inducción de máquina. Asimismo, los logros en el campo de la extracción de reglas son paralelos también a los alcances realizados por trabajos en otras ramas de la inteligencia artificial y las ciencias cognitivas.

En general, el uso de reglas atiende a una visión especial acerca del conocimiento y la forma en que éste puede ser representado. Usar reglas implica formas particulares de concebir a los problemas y por supuesto, implica también el reconocimiento de ciertas formas de resolución de los mismos. Estos hechos han ocasionado una polémica en torno al uso de esta forma de representación del conocimiento. La discusión se vuelve más enérgica, sobre todo, debido a que en la actualidad, existen y son ya ampliamente difundidas, muchas otras maneras de representar el conocimiento en una forma más eficiente y sin el menoscabo de la información que se trata de “cuadricular para poderla meter en una regla”.

Sin embargo, a pesar de la discusión en torno a su uso o no, las reglas son reconocidas aún como el medio más efectivo —e incluso, *natural*— para transmitir conocimientos entre los humanos, tomar decisiones y justificar las mismas (Craven, 1996). Asimismo, las reglas son reconocidas como un medio muy efectivo para la producción de conocimientos nuevos. En inteligencia artificial, las reglas son uno de los medios más eficientes para lograr que sean comprensibles ciertos dispositivos tecnológicos cuya representación del conocimiento se da a nivel subsimbólico y que en los últimos años han tenido un auge significativo: las redes neuronales y los sistemas híbridos neuronales. Es por ello, entre otras cosas, por lo que se

sigue concediendo importancia al estudio de las reglas, aún a pesar del surgimiento de nuevas formas de representar el conocimiento. La extracción de reglas, por otra parte, es un tema que se ha estudiado desde hace ya diez años. Sin embargo, existe cierto consenso en que aún queda mucho por investigar y conocer al respecto (Andrews, Cable, Diederich, Geva, Golea, Hayward, Ho-Stuart y Tickle, 1995; y Andrews, Diederich y Tickle, 2001).

En este documento, se propone un modelo para poder extraer y representar en forma de reglas, el conocimiento contenido en redes neuronales difusas³. Estos dispositivos se producen a partir de la integración, combinación y complemento de dos de los enfoques más importantes de los últimos años en Inteligencia Artificial: las redes neuronales y la lógica difusa. Además, se integran también en este reporte de investigación los resultados obtenidos tras las pruebas realizadas al modelo y el análisis de los mismos.

El fin último de la investigación que en este documento se reporta fue realizar una pequeña aportación que lleve hacia una mayor *comprensibilidad*⁴ en redes neuronales y redes neuronales híbridas.

La hipótesis principal que sirvió de guía para el trabajo fue la siguiente:

Es posible desarrollar un modelo para la extracción de reglas en Redes Neuronales Difusas que cuente con las siguientes características:

- Debe ser un modelo general, escalable y mostrar un desempeño general competitivo.
- Las bases de reglas extraídas deben ser generales, modificables y fieles al conocimiento a partir del cual fueron creadas.
- Cada regla extraída debe ser válida, comprensible y general.

En donde:

- *La generalidad de un modelo*, se debe entender como el grado de independencia que el modelo permite con respecto a la topología de la red en la que se utiliza. Asimismo, incluye el grado de independencia sobre los regímenes de entrenamiento de la red y funciones de activación utilizadas.

³ *Fuzzy Neural Network*, en inglés.

⁴ La comprensibilidad de un sistema se puede definir como el grado de facilidad en un sistema, para que el conocimiento representado en éste, sea accesible al ser humano (Taha y Gosh, 1996).

- *La escalabilidad de un modelo*, es el grado de independencia que el modelo permite con respecto al tamaño o escala del problema a resolver.
- *El desempeño general competitivo* de un modelo se refiere a la capacidad del modelo para competir con la forma y los resultados obtenidos por modelos similares, ante problemas similares.
- Por *generalidad de los conjuntos de reglas extraídos* debe entenderse el grado de cobertura que las reglas del conjunto logran en el dominio, comparado con la cobertura lograda por la red neuronal entrenada.
- *La modificabilidad de los conjuntos de reglas extraídos* es la capacidad del modelo para permitir la modificación de las reglas extraídas a partir de cambios subsecuentes en el dominio del conocimiento o en la estructura de la red que los soporta.
- *La fidelidad de los conjuntos de reglas extraídos* indica el grado con que el conjunto de reglas refleja realmente el conocimiento representado en la red. Es decir, se puede entender como el grado de equivalencia que guardan con el conjunto de conocimientos representados.
- *La validez de cada regla extraída* es el grado de eficacia con que cada una de las reglas generada refleja un hecho o relación real en el dominio del conocimiento.
- *La comprensibilidad de cada una de las reglas extraídas* es el grado de facilidad con el que el ser humano puede interpretarlas en forma inmediata.
- *Generalidad de cada una de las reglas extraídas* es el grado de cobertura que una regla al interior del dominio del conocimiento en el cual se aplica.

Derivados de la hipótesis presentada en este apartado, los objetivos generales que guiaron este trabajo fueron planteados de la siguiente manera:

- Primero. Demostrar que es posible diseñar un modelo de extracción de reglas que cumpla con las características establecidas en la hipótesis presentada en este mismo documento
- Segundo. Demostrar que el modelo diseñado es capaz de extraer bases de reglas que cumplen con las características especificadas en la hipótesis presentada en este mismo documento.
- Tercero. Demostrar que las bases de reglas extraídas por el modelo de extracción diseñado están compuestas por reglas que cumplen con

los requerimientos especificados en la hipótesis presentada en este documento.

A partir de los objetivos generales planteados, se establecieron los siguientes objetivos específicos:

- Establecer un conjunto de referentes empíricos relacionados con los modelos de extracción de reglas, con los conjuntos o bases de reglas y con las reglas, que pueda ser usado como base para el diseño de un modelo eficiente para extracción de reglas en redes neuronales difusas.
- Diseñar un modelo de extracción de reglas en redes neuronales difusas, que cumpla con las características que la hipótesis del presente trabajo de investigación ha marcado como deseables tanto para el modelo de extracción, como para las reglas extraídas (individualmente y en conjunto).
- Establecer los mecanismos necesarios para evaluar las características de un modelo de extracción de reglas en redes neuronales difusas
- Evaluar el modelo diseñado de acuerdo al mecanismo de evaluación establecido
- Analizar los resultados de la evaluación del modelo diseñado y reportar resultados, conclusiones obtenidas del trabajo general, contribuciones y limitaciones identificadas, y posibilidades de trabajos posteriores en la misma línea.

Para lograr un modelo eficiente de extracción de reglas en redes neuronales difusas y cumplir con los objetivos, planteados se estableció un modelo inicial con base en las características que, tras la revisión de la información disponible, se consideraron como ideales. Este modelo se fue adecuando y refinando con base en la evaluación de su eficacia, a partir del análisis de información interna y externa; tomando en cuenta, además, las fallas de modelos alternativos existentes y las posibilidades reales de sortear los problemas presentados.

En el diseño de modelos no se puede hablar de “recetas” o “metodologías específicas” de creación, más bien se puede hablar de estrategias para sistematizar y formalizar esta actividad eminentemente creativa. En este trabajo, la estrategia general seguida para el diseño del modelo que aquí se propone, fue la identificación de deficiencias y obstáculos en los modelos existentes y la búsqueda de soluciones nuevas con base en la

transferencia y/o adecuación de los conocimientos previos en ámbitos diferentes.

Para verificar los logros y los alcances del BCD se realizaron pruebas que permitieron comparar los resultados que el modelo arrojó, frente a los resultados obtenidos con dos modelos diseñados para los mismos propósitos: el *Sistema Neurodifuso de Clasificación* (NEFCLASS⁵) de Nauck y Kruse (Nauck y Kruse, 1994, 1995, 1998 y 1999; Nauck y Kruse, 1995; Nauck, 1995, 2002; Nauck, Klawonn y Kruse, 1992 y 1995; y Nauck, Nauck y Kruse, 1996) y el *Sistema Difuso Neurogenerador de reglas* (FUNEGEN⁶) de Halgamuge y Glesner (Halgamuge y Glesner, 1994a, 1994b; Chan y Halgamuge, 2003; y Saad y Halgamuge, 2004).

Los modelos NEFCLASS, FUNEGEN y BCD fueron sometidos a la resolución de un problema de “laboratorio” clásico en la literatura referente a inteligencia artificial, reconocimiento de patrones, aprendizaje de máquina, e incluso, estadísticas y probabilidad: el “problema de clasificación de la planta *iris*” (Fisher, 1935), o “el problema de la planta iris”, como comúnmente se le conoce. Al trabajar todos los modelos en la resolución del mismo problema y comparar los resultados obtenidos por cada uno contra los otros, se estuvo en condiciones de poder ubicar y valorar las características del modelo aquí propuesto (BCD). Este fue el razonamiento subyacente para el empleo de esta estrategia de trabajo.

Una vez explicados los objetivos y la hipótesis que sirvieron de guía al trabajo del que en este documento se da cuenta; se consideró pertinente presentar al lector una visión general acerca de lo que encontrará en cada uno de los apartados del presente documento:

Este libro se estructura de manera que permita llevar al lector de lo simple a lo complejo, de lo básico a lo especializado y desde las estructuras, computacional y matemática, en las que tiene lugar la extracción de reglas; hasta a los modelos que existen en la actualidad para realizar la extracción de reglas. Se consideró pertinente presentar el modelo BCD y los resultados de su evaluación, sólo hasta después de haber presentado una breve, pero concisa, revisión del soporte teórico que hay detrás del trabajo de extracción de reglas. La extracción de reglas en redes neuronales difusas implica el uso de los conocimientos actuales en el campo de las redes neuronales, lógica difusa, redes neuronales difusas y modelos extracción de reglas. Es por ello que, en este trabajo, se dedica un capítulo a cada uno de estos temas.

5 *Neuro Fuzzy Classifier System* en inglés.

6 *Fuzzy Neuro Generator System* en inglés.

En el Capítulo 1 se presenta una introducción breve a la extracción de reglas y su importancia como campo de investigación actual en ciencias computacionales y ciencias cognoscitivas. Continúa enunciando las hipótesis y los objetivos generales y específicos que guiaron el proceso. Ofrece un esbozo general del trabajo realizado y sus lineamientos generales.

Debido que los insumos principales para este trabajo de investigación y desarrollo del modelo fueron las redes neuronales difusas, producto de la articulación entre lógica difusa y redes neuronales, es que en los Capítulos 2 y 3 se pretende establecer cuáles son los enfoques teóricos a partir de los cuales se ha podido dar esta articulación.

El Capítulo 2 presenta al lector una introducción a la lógica difusa, sus conceptos básicos, aplicaciones, críticas y desarrollos a futuro. Asimismo, el capítulo ofrece una revisión a detalle medio, de los mecanismos básicos que los sistemas difusos emplean en la resolución de problemas como la inferencia y el control difuso.

El Capítulo 3 ofrece una introducción a las redes neuronales, sus componentes y su funcionamiento básico bajo uno de los paradigmas principales: el *aprendizaje supervisado*. Se revisa a mediana profundidad el aprendizaje basado en la *retropropagación del error* como uno de los casos más simples de este paradigma, ya que es precisamente este modelo de red neuronal el que fue seleccionado para poner a prueba el modelo que en este trabajo se propone. La parte final de este capítulo presenta las bases para la comprensión de la articulación entre redes neuronales y lógica difusa, así como las nuevas líneas de trabajo en redes neuronales, entre las cuales se encuentra el trabajo en sistemas híbridos.

El Capítulo 4, denominado “Extracción de reglas en redes neuronales nítidas y redes neuronales difusas”, pretende establecer el marco conceptual de referencia que permite considerar las dimensiones que tienen lugar a propósito de la extracción de reglas. Para ello se abordan las formas de uso de las reglas en los sistemas neuronales y los procesos implicados en la extracción de las mismas de sistemas neuronales y neuronales difusos. Se revisan los métodos contemporáneos más sobresalientes, a juicio personal, para extracción de reglas de sistemas neuronales. Cabe mencionar que incluso, se revisan modelos de extracción no considerados en los estudios ya clásicos sobre este tema. El capítulo cierra con una apreciación personal acerca del futuro de la extracción de reglas como campo de investigación.

El Capítulo 5, presenta el modelo de extracción de reglas difusas basado en la Binarización de los Conjuntos Difusos (BCD). Inicia con las

generalidades del modelo y discute la diferencia entre binarización de entradas de red, método usado por algunos modelos contemporáneos, y la binarización de conjuntos difusos en las entradas del sistema, que es la estrategia en la cual se basa el modelo BCD. Este capítulo incluye también una discusión acerca los alcances de la extracción de reglas y las posibles generalizaciones de los hallazgos realizados cuando se usa solamente un conjunto de datos de entrenamiento para realizar la extracción. Finalmente, se exponen las características específicas del modelo BCD, su esquema de trabajo y sus cinco fases de trabajo.

El Capítulo 6, presenta la evaluación que fue realizada para conocer los alcances y limitaciones del modelo. En investigación en ciencias computacionales y, específicamente, sobre extracción de reglas, se reconocen dos tipos de evaluación: una, interna, que está relacionada con el desempeño de un modelo a través del tiempo, y la otra, externa, que utiliza como estrategia la comparación, ya sea entre pares, o con estándares establecidos. La metodología diseñada para este trabajo contempla trabajo de evaluación de los dos tipos y bajo las dos modalidades de comparación, es por ello que además de reportar el trabajo con el modelo BCD se reporta el trabajo realizado con otros dos modelos de extracción de reglas. El capítulo inicia con la explicación general de la metodología empleada, así como de los criterios seleccionados para la evaluación. Continúa reportando los resultados obtenidos, por cada uno de los tres modelos, tras su puesta a prueba ante un *problema de laboratorio*. Los resultados se presentan organizados de acuerdo a los criterios de evaluación establecidos, ofreciendo al final la evaluación realizada.

El capítulo 7 muestra a manera de conclusión, los aprendizajes obtenidos en la propuesta del modelo, así como las contribuciones, limitaciones del trabajo y del modelo mismo, y finalmente se esbozan los lineamientos propuestos para estudios futuros. En este capítulo se hace una evaluación personal, usando como punto de referencia los objetivos inicialmente establecidos en la propuesta y los resultados arrojados en cada una de las pruebas realizadas a los tres modelos.

2

Lógica Difusa

2.1. Introducción

El manejo de información vaga, incierta o inexacta es una tarea que parece no representar gran dificultad para el cerebro humano. Sin embargo, las dimensiones del problema se aprecian en su justa dimensión si se considera que los sistemas computacionales aún no han logrado resolverlo efectivamente.

La búsqueda por procesar este tipo de información tiene ya casi cien años. Desde inicios del siglo XX el polaco Jan Lukasiewicz creó la *lógica multivaluada*, lógica en la que los enunciados podían tener valores de verdad fraccionarios comprendidos entre el cero y el uno. Asimismo, el filósofo Max Black publicó un artículo en 1937 en la revista *Philosophy of Science*, en el que aplicaba la lógica multivaluada de Lukasiewicz a listas de objetos para establecer las primeras curvas de pertenencia múltiple a varias clases. El trabajo de Black es trascendental ya que ofrecía una alternativa al postulado clásico del tercero excluido de la lógica aristotélica.

Una generalización de la teoría convencional de conjuntos, que proponía formas novedosas para procesar información vaga, inexacta o incierta fue la *Lógica Difusa*,¹ propuesta en 1965 por Lofti A. Zadeh. El trabajo en lógica difusa ha sido ampliamente desarrollado en las dos últimas décadas por investigadores como B. Kosko (Kosko, 1992a), (Kosko, 1992b), (Kosko, 1993), (Kosko, 1997); M. M. Gupta (Gupta y Qi, 1992) y W. Pedrycz (Pedrycz, 1993), quienes han extendido los trabajos de Zadeh hacia el control automático de procesos y la identificación de patrones, con lo que han dado origen a una nueva disciplina: *La ingeniería difusa*.²

1 *Fuzzy Logic*, en inglés.

2 *Fuzzy Engineering*, en inglés.

Dado que las redes neuronales difusas, insumo principal del trabajo realizado, tienen parte de su fundamentación en el marco de la lógica difusa, el presente capítulo tiene como propósito principal ofrecer al lector una breve introducción sobre los conceptos más comunes de esta disciplina.

2.1.1. Variables lingüísticas

En la vida diaria, el ser humano es capaz de emplear, comprender y transmitir información expresada en términos que no necesariamente se tienen que definir con exactitud. Por ejemplo, es posible utilizar en forma vaga, inexacta o incierta - *difusa*, bajo la óptica de Zadeh -, términos como *bajo*, *medio* o *alto*, para referirse a la estatura de las personas en un grupo. El uso de estos términos se hace sin que necesariamente se tenga que especificar claramente la interpretación personal que se hace de ellos; tampoco es necesario aclarar los límites que se están empleando para cada una de las clases a las que se hace referencia; incluso, en el proceso de comunicación humana no es necesario que la forma en que los términos se usan sea compartida con exactitud entre emisor y receptor.

En el marco de la lógica difusa los términos usados difusamente, es decir, inexactos, vagos o inciertos, son llamados *variables lingüísticas*³ (Kosko, 1992). La lógica difusa facilita el trabajo con este tipo de variables ya que posibilita el establecimiento de conjuntos con restricciones o límites elásticos que se mantienen vigentes para un universo de discurso dado en un momento determinado (Pedrycz, 1993).

Una variable lingüística emplea valores difusos, ya sean números difusos o términos difusos y se caracteriza por los elementos descritos en la ecuación (2,1).

$$(x, T(x), U, G, M) \tag{2,1}$$

3 El término de variable lingüística (en inglés *Linguistic variable*) no pretende tener el mismo significado que tiene en lingüística o algunas otras disciplinas cuyo objeto de estudio es el lenguaje. Probablemente una traducción al español más clara de este término sería el de *variables verbales*. Sin embargo, el término que se emplea en este documento se ha conservado dada su amplia difusión, ya que es la forma en la que se ha traducido en la literatura especializada en español.

En donde x es el nombre de la variable, $T(x)$ es el conjunto de términos de x , es decir, el conjunto de nombres de los valores de x , para los que cada valor es un número difuso definido en U , G es la regla usada para generar los valores de x , y M es la regla para asociar cada valor con su significado.

Por ejemplo, si se quisiera usar el término *estatura* como variable lingüística, el conjunto de términos $T(\textit{estatura})$ podría ser definido como el que se muestra en la ecuación (2,2).

$$T(\textit{estatura}) = \{\textit{baja}, \textit{media}, \textit{alta}\} \quad (2,2)$$

En la ecuación (2,2) cada término en $T(\textit{estatura})$ es caracterizado por un conjunto difuso en un universo de discurso $U = [0,100]$.

Para este ejemplo cada término podría interpretarse de la siguiente manera:

- *Baja*, como “estatura menor a 1.50 mts.”
- *Media*, como “estatura alrededor de 1.70 mts.”
- *Alta*, como “estatura mayor a 1.90 mts.”

Al caracterizar estos términos como conjuntos difusos se obtienen las funciones de membresía que se muestran en la ilustración 1.

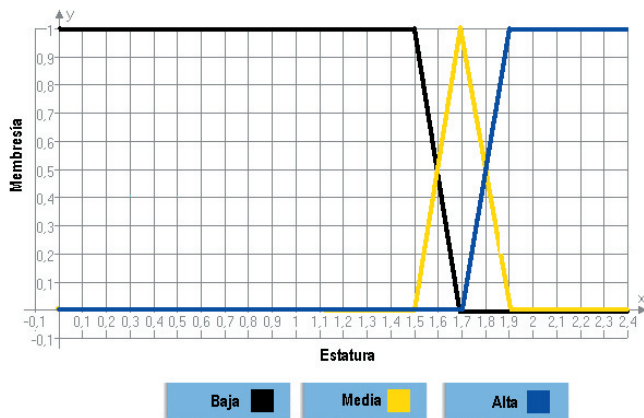


ILUSTRACIÓN 1. Funciones de membresía para la variable lingüística *estatura*.

Con base en el ejemplo anterior, se podría pensar que las variables lingüísticas deben necesariamente guardar un orden, dado que la definición de los términos sugiere que *alta* > *media* > *baja*. Pero esto no siempre es así. Se pueden dar casos en los que los términos son definidos con base en límites distintos, con lo que se perdería esta ilusión de orden. Por ejemplo, si se define a la variable lingüística *estatura* como se muestra en la ecuación (2,3).

$$T(\textit{estatura}) = \{\textit{baja}, \textit{media}, \textit{alta}, \textit{elevada}, \textit{mínima}, \textit{medianamente elevada}, \textit{regular}\} \quad (2,3)$$

Y si cada término de esta nueva definición en $T(\textit{estatura})$ fuese especificado ahora de la siguiente manera:

- *Baja*, como “estatura menor a 1.50 mts.”
- *Media*, como “estatura alrededor de 1.70 mts.”
- *Alta*, como “estatura mayor a 1.90 mts.”
- *Elevada*, como “estatura mayor a 1.85 mts.”
- *Mínima*, como “estatura menor a 1.73 mts.”
- *Medianamente elevada*, como “estatura alrededor de 1.65 mts, y 1.87 mts.”
- *Regular*, como “estatura alrededor de 1.70 mts y 1.80 mts.”

Entonces, la caracterización de estos términos como conjuntos difusos sería como se muestra en la ilustración 2.

2.1.2. Funciones de membresía

Witold Pedrycz (Pedrycz, 1993) tras un análisis de las técnicas usadas en la literatura especializada, ha encontrado dos enfoques principales de las técnicas existentes para el establecimiento de funciones de membresía:

- a) Un enfoque horizontal, cuya idea subyacente es la reunión de toda la información disponible sobre los grados de membresía de los elementos, en un universo de discurso a partir del cual se van a definir los conjuntos difusos. Bajo este enfoque contempla la participación de expertos en este universo de discurso a cada uno de los cuales se haría la pregunta: “¿Puede ser visto (x_0) como un elemento compatible con el concepto implicado en el conjunto difuso A ?”, en donde

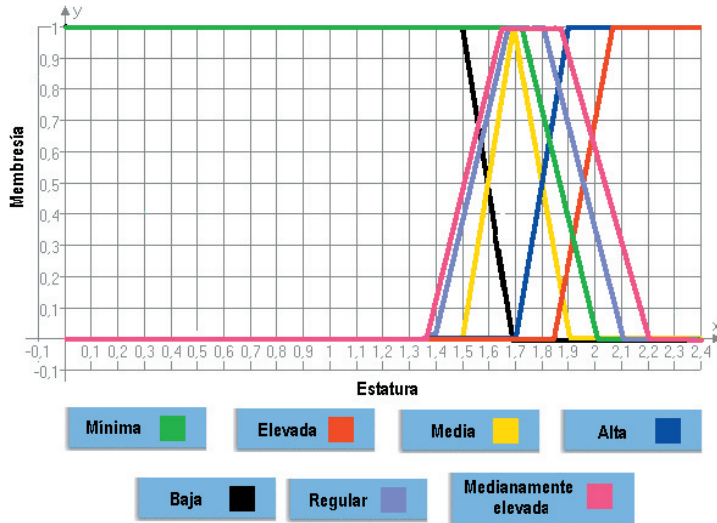


ILUSTRACIÓN 2. Nuevas funciones de membresía para la variable lingüística *estatura*.

(x_0) es un elemento fijo en este universo de discurso. El total de respuestas afirmativas se usaría para calcular la función de membresía para este elemento con base en la ecuación (2,4).

$$\mu_A(x_0) = \frac{n(x_0)}{N} \quad (2,4)$$

En donde N es el total de respuestas relacionadas con (x_0) . La ventaja de este método radica en su facilidad de aplicación.

Derivado de éste, se tienen métodos más complejos, como el expuesto en la ecuación (2,5), que integran la desviación estándar de las estimaciones obtenidas para establecer límites de aceptación de los grados de membresía obtenidos.

$$\sigma^2(A)(x_0) = \left[A(x_0) \frac{1 - A(x_0)}{N} \right]^{1/2} \quad (2,5)$$

En la ecuación (2,5) si la desviación obtenida no excede al umbral λ , entonces $A(x_0)$ se acepta.

- b) Por otro lado, distingue también un *enfoque vertical*, que se basa en un nivel fijo de membresía, denotado por α que se selecciona arbitrariamente, y se compara con el grado de pertenencia que los expertos asignan a cada elemento para saber el *grado en que cada elemento es no es menor a α* .

El conjunto de métodos existentes para estimar las funciones de membresía a conjuntos difusos son en algún grado derivaciones de alguno de estos dos enfoques (Pedrycz, 1993).

2.1.3. Teoría del razonamiento con base en aproximaciones

En 1979 Lofti Zadeh propuso la Teoría del razonamiento con base en aproximaciones.⁴ El aspecto central de esta teoría es la representación de las proposiciones como afirmaciones con conjuntos difusos. Estos conjuntos difusos son asignados como si fuesen valores en una variable.

2.1.3.1. Reglas de inferencia difusa

A continuación se enuncian las reglas de inferencia difusa más comunes de esta teoría.

- a) Si se supone que existen dos variables x y y tales que $x \in X$ y $y \in Y$, y hay una relación causal entre x y y que es totalmente conocida, es decir,

$$y = f(x) \tag{2,6}$$

Y se conoce que y es una función de x , entonces se puede realizar una inferencia fácilmente si se considera que:

<i>Premisa</i>	$y = f(x)$	(2,7)
<i>Hecho</i>	$x = x'$	
<i>Consecuencia</i>	$y = f(x')$	

⁴ *Aproximate Based Reasoning*, en inglés.

La regla de inferencia presentada en (2,7) se puede interpretar de la siguiente manera: si se tiene que $y = f(x)$, $\forall x \in X$ y se observa que $x = x'$, entonces y toma el valor de $f(x')$.

- b) Sin embargo, lo que sucede más a menudo es que no se conoce por completo la relación causal y de cualquier manera se tiene que realizar una inferencia a partir de la información disponible. Para estos casos en que para la liga f entre x y y , sólo se conocen algunos de los valores de x en $f(x)$, es decir, se conocen las siguientes reglas:

$$\mathfrak{R}_1 : \quad \text{Si } x = x_1 \text{ entonces } y = y_1$$

$$\mathfrak{R}_2 : \quad \text{Si } x = x_2 \text{ entonces } y = y_2$$

...

$$\mathfrak{R}_n : \quad \text{Si } x = x_n \text{ entonces } y = y_n$$

Si se tiene una $x' \in X$ y se quiere encontrar una $y' \in Y$ la cual corresponde a x' bajo la base de reglas, entonces:

$$\mathfrak{R}_1 : \quad \text{Si } x = x_1 \text{ entonces } y = y_1$$

$$\mathfrak{R}_2 : \quad \text{Si } x = x_2 \text{ entonces } y = y_2$$

...

$$\mathfrak{R}_n : \quad \text{Si } x = x_n \text{ entonces } y = y_n$$

$$\frac{\text{hecho : } \quad x = x}{\text{consecuencia : } \quad y = y'}$$

(2,9)

Este problema a menudo es conocido como *interpolación*.⁵

- d) Ahora bien, si se tiene que dos variables lingüísticas x y y tales que x es “alto” y y es “pequeño”. El problema básico del razonamiento con base en aproximaciones es encontrar la función de membresía de la consecuencia C a partir de la base de reglas $\{\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_n\}$ y el hecho A , entonces:

⁵ *Interpolation* en inglés.

$$\begin{array}{l}
\mathfrak{R}_1 : \quad \text{Si } x \text{ es } A_1 \text{ entonces } y \text{ es } C_1 \\
\mathfrak{R}_2 : \quad \text{Si } x \text{ es } A_2 \text{ entonces } y \text{ es } C_2 \\
\dots \\
\mathfrak{R}_n : \quad \text{Si } x \text{ es } A_n \text{ entonces } y \text{ es } C_n \\
\hline
\text{hecho : } \quad x \text{ es } A \\
\hline
\text{consecuencia :} \quad y \text{ es } C
\end{array} , \tag{2,10}$$

2.1.3.2. Reglas de traslación

Zadeh también propuso un conjunto de reglas de traslación que permiten la representación de algunas afirmaciones.

A continuación se presentan algunas de esas reglas de traslación:

- a) Regla de asignación por herencia de clase:⁶

$$\begin{array}{l}
x \text{ es } A \\
A \subset B \\
\hline
x \text{ es } B
\end{array} \tag{2,11}$$

Por ejemplo: Si Juan es *muy alto*, y *muy alto* es un subconjunto de *alto*, entonces Juan es *alto*.

- b) Regla de conjunción⁷

$$\begin{array}{l}
x \text{ es } A \\
x \text{ es } B \\
\hline
x \text{ es } A \cap B
\end{array} \tag{2,12}$$

Por ejemplo: Si Juan es *muy alto*, y Juan es *muy estudioso*, entonces Juan es *muy alto y es muy estudioso*.

⁶ *Entailment Rule* en inglés.

⁷ *Conjunction Rule* en inglés.

c) Regla de conjunción con negación⁸

$$\frac{\begin{array}{l} x \text{ no es } A \\ x \text{ no es } B \end{array}}{x \text{ no es } A \cap B} \quad (2,13)$$

Por ejemplo: Si Juan es *no es muy alto*, y Juan *no es muy bajo*, entonces Juan *no es muy alto y no es muy bajo*.

d) Regla de disyunción⁹

$$\frac{\begin{array}{l} x \text{ es } A \\ \vee x \text{ es } B \end{array}}{x \text{ es } A \cup B} \quad (2,14)$$

Por ejemplo: Si Juan es *muy alto*, o Juan es *muy estudioso*, entonces Juan es *muy alto* o es *muy estudioso*

e) Regla de disyunción con negación¹⁰

$$\frac{\begin{array}{l} x \text{ no es } A \\ \vee x \text{ no es } B \end{array}}{x \text{ no es } A \cup B} \quad (2,15)$$

Por ejemplo: Si Juan *no es muy alto*, o Juan *no es muy estudioso*, entonces Juan *no es muy alto* o *no es muy estudioso*.

8 *Conjunction Rule with negation* en inglés.

9 *Disjunction Rule* en inglés.

10 *Disjunction Rule with negation* en inglés.

f) Regla de proyección¹¹

$$\frac{(x, y) \text{ tienen una relación } R}{x \text{ es } \prod_x(R)} \quad (2,16)$$

$$\frac{(x, y) \text{ tienen una relación } R}{y \text{ es } \prod_y(R)} \quad (2,17)$$

Por ejemplo:

Si (x, y) está cerca de $(5, 4)$
entonces : x está cerca de 5

Si (x, y) están cerca de $(5, 4)$
entonces : y está cerca de 4

g) Regla de negación¹²

$$\frac{\text{no } (x \text{ es } A)}{x \text{ es } \neg A} \quad (2,18)$$

Por ejemplo: Si *no es cierto* que Juan es *muy alto*, entonces Juan *no es muy alto*

h) Reglas de inferencia por composición¹³

De las reglas de inferencia por composición, una puede ser vista como un *Modus Ponens* Difuso:

$$\begin{array}{l} \text{Premisa : } \quad \text{Si } x \text{ es } A \text{ entonces } y \text{ es } B \\ \text{Hecho : } \quad \quad x \text{ es } A' \\ \hline \text{Consecuencia : } \quad \quad y \text{ es } B' \end{array} \quad (2,19)$$

11 *Projection Rule*, en inglés.

12 *Negation Rule*, en inglés.

13 *Compositional Rule of Inference*, en inglés.

y la otra como un *Modus Tolens* Difuso:

Premisa : Si x es A entonces y es B

Hecho : y es B'

(2,20)

Consecuencia : x es A'

2.1.4. El proceso de Inferencia difusa

Reglas tan sencillas como las presentadas en los apartados anteriores han sido combinadas por enfoques completos de razonamiento difuso para aplicaciones en sistemas difusos.

Algunos de los enfoques más importantes en razonamiento difuso son:

- El enfoque de Mamdani (Bandemer y Gottwald, 1995).
- El enfoque de Tsukamoto (Fullér, 1995).
- El enfoque de Sugeno (Jager, 1995).
- El enfoque de Larsen (Fullér, 1995).

En este documento no se pretende abordar estos enfoques de razonamiento difuso, pero si se ha considerado pertinente describir y ejemplificar en términos generales, como es que se daría un razonamiento difuso bajo condiciones específicas.

2.1.4.1. Etapas de la inferencia difusa

En general, el proceso de inferencia difusa se compone de tres o cuatro etapas, dependiendo de las necesidades específicas del problema que se quiere resolver:

- *Etapas de difuminación de los valores nítidos*. En esta etapa se deben definir las variables lingüísticas, sus valores difusos y las funciones de membresía que se deben emplear para determinar el grado de pertenencia a cada conjunto difuso de un valor nítido determinado.
- *La Etapa de Realización de inferencias parciales*. En esta etapa se calcula el valor de verdad de cada una de las premisas de una regla y este valor se aplica a la parte correspondiente en la conclusión. El

objetivo es obtener por cada regla disparada, el subconjunto difuso al que esta hace referencia en la conclusión. En esta etapa se hace uso de las reglas de inferencia difusa ya presentadas.

- En la *Etapa de Composición* todos los subconjuntos difusos obtenidos en la fase anterior se combinan para conformar un solo conjunto difuso de salida. El conjunto resultante es el correspondiente a la conclusión obtenida en el proceso de inferencia difusa y también es calculado con base en el uso de alguna de las reglas de inferencia difusa ya presentadas.
- Finalmente, *La Etapa de Nitidización*. Si las características del problema que se trata de resolver lo ha marcado como necesario, entonces se debe obtener un valor nítido resultante de este proceso de inferencia difusa.

Un ejemplo de inferencia difusa para explicar la manera específica en que se podrían realizar cada una e las etapas mencionadas podría ser el siguiente: si se asume la existencia de las variables lingüísticas x, y, z , cuyos valores están comprendidos entre $[0,10]$ y se tuviese un sistema difuso que posibilite calcular el valor difuso de z a partir de los valores de x, y , por medio de las siguientes reglas:

Regla 1: Si x es *Bajo* y y es *Bajo*, entonces z es *Alto* (2,21)

Regla 2: Si x es *Bajo* y y es *Alto*, entonces z es *Bajo* (2,22)

Regla 3: Si x es *Alto* y y es *Bajo*, entonces z es *Bajo* (2,23)

Regla 4: Si x es *Alto* y y es *Alto*, entonces z es *Alto* (2,24)

Bajo las siguientes funciones de membresía:

Para la variable lingüística x las funciones para calcular la pertenencia al conjunto difuso *Bajo* y *Alto* se presentan en las ecuaciones (2,25) y (2,26):

$$Bajo_{(x)} = 1 - (x / 10) \quad (2,25)$$

$$Alto_{(x)} = (x / 10) \quad (2,26)$$

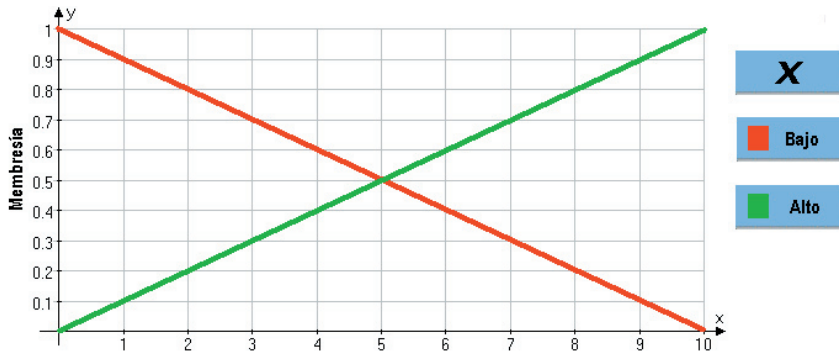


ILUSTRACIÓN 3. Funciones de membresía para los conjuntos difusos de la variable lingüística x .

Para la variable lingüística y y las funciones para calcular la pertenencia al conjunto difuso *Bajo* y *Alto* se presentan en las ecuaciones (2,27) y (2,28):

$$Bajo_{(z)} = 1 - (z/10) \quad (2,27)$$

$$Alto_{(z)} = (z/10) \quad (2,28)$$

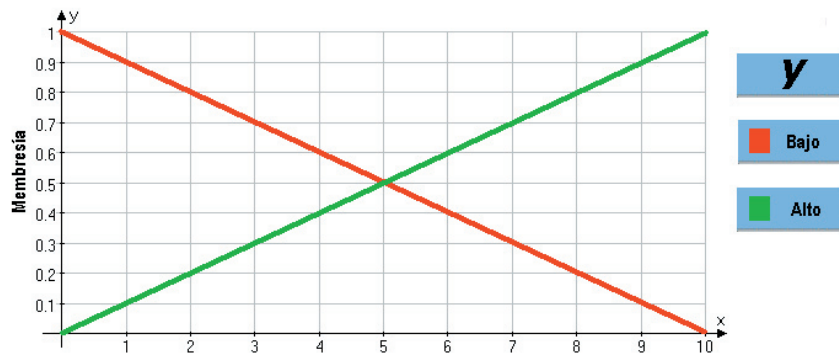


ILUSTRACIÓN 4. Funciones de membresía para los conjuntos difusos de la variable lingüística y .

Para la variable lingüística z las funciones para calcular la pertenencia al conjunto difuso *Bajo* y *Alto* se presentan en las ecuaciones (2,29) y (2,30):

$$Bajo_{(z)} = 1 - (z/10) \quad (2,29)$$

$$Alto_{(z)} = (z/10) \quad (2,30)$$

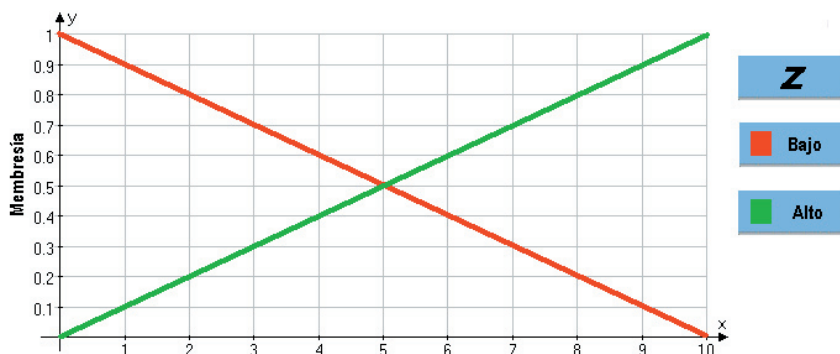


ILUSTRACIÓN 5. Funciones de membresía para los conjuntos difusos de la variable lingüística z .

Si se tienen los valores siguientes como entradas en el sistema: $x = 0.0$, $y = 3.2$, el proceso de inferencia para calcular el valor de z se daría de la siguiente manera:

- Etapa 1, Difuminación de los valores nítidos.

En esta etapa se tendrían que calcular los valores de pertenencia de x y de y a los conjuntos difusos correspondientes, mediante el empleo de las funciones de membresía establecidas en las ecuaciones (2,25) y (2,26) para x , y (2,27) y (2,28) para y . Así, se tendrían los siguientes valores:

$$Bajo(x) = 1.0 \quad Alto(x) = 0.0 \quad Bajo(y) = 0.68 \quad Alto(y) = 0.32$$

El paso siguiente sería ahora calcular el valor de verdad para cada regla en este caso en particular. A menudo, a este valor de verdad se le llama valor α (alfa). Este valor se calcula tomando como referencia alguna de las reglas de inferencia como las que se abordan en el apartado 2.1.3

de este mismo documento. Para este caso específico, dado que las reglas del sistema difuso que se está ejemplificando hacen uso de la conjunción en la parte de las premisas de la regla, entonces se hace uso de la regla de inferencia de *conjunción* mencionada en el inciso b) del apartado 2.1.3. Así, el valor α para cada una de las reglas sería:

$$\alpha_1 = 0.68, \quad \alpha_2 = 0.32, \quad \alpha_3 = 0.0 \quad \text{y} \quad \alpha_4 = 0.0$$

Si el valor α de una regla es mayor que cero entonces se dice que la regla “dispara”.

- Etapa 2. *Realización de inferencias parciales.*

En esta etapa también se hace uso de las reglas de inferencia presentadas en el apartado 2.1.3 para calcular la parte correspondiente del conjunto difuso de salida que aporta cada una de las reglas disparadas en el proceso de inferencia. Usando la regla de inferencia de conjunción para calcular el conjunto difuso de salida, función de membresía saliente se establece a partir de la altura correspondiente a la proporción estimada en el valor α de la regla utilizada. Para el caso específico de la regla 1, dado que el valor $\alpha_1 = 0.68$ y dado que el rango de valores va de $[0,10]$, entonces el valor proporcional a partir del cual se establece la función de membresía es 6.8.

Con lo que para esta regla, el valor parcial de z se calcula a partir de la función presentada en la ecuación (2,31), que como podrá observarse, está basada en la función de membresía empleada para calcular $Alto(z)$ ya que es precisamente esta la conclusión hacia la que apunta la regla 1.

$$Re\ gl\ 1_{(z)} = \begin{cases} z / 10 & \text{si } z \leq 6.8 \\ 0.68 & \text{si } z > 6.8 \end{cases} \quad (2,31)$$

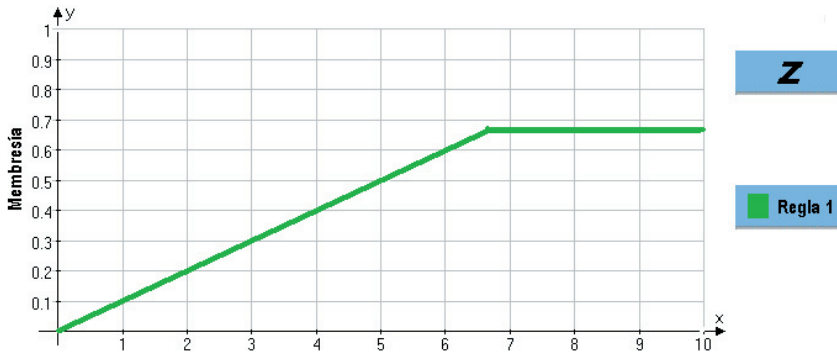


ILUSTRACIÓN 6. Conjunto difuso resultante de la inferencia con la regla 1.

El valor parcial de z correspondiente a la Inferencia con la Regla 2 se muestra en la ecuación (2,32). En este caso podrá observarse que está basada en la función de membresía empleada para calcular $Bajo(z)$ ya que es precisamente esta la conclusión hacia la que apunta la regla 2.

$$Regla1_{(z)} = \begin{cases} 0.32 & \text{si } z \leq 6.8 \\ 1 - (z/10) & \text{si } z > 6.8 \end{cases} \quad (2,32)$$



ILUSTRACIÓN 7. Conjunto difuso resultante de la inferencia con la regla 2.

Dado que las reglas 3 y 4 no disparan, su aportación al conjunto difuso resultante no se calcula.

- Etapa 3. *Composición.*

Durante la etapa de composición también se hace uso de las reglas de inferencia mencionadas en el apartado 2.1.3 de este mismo capítulo. El objetivo es obtener el conjunto difuso resultante de la composición de los conjuntos que representan la aportación que cada regla hace a la conclusión final. En este caso, el conjunto difuso resultante se presenta en la ecuación (2,33).

$$Conclusión (Z) = \begin{cases} 0.32 & \text{si } z \leq 3.2 \\ z / 10 & \text{si } 3.2 < z \leq 6.8 \\ 0.68 & \text{si } 6.8 < z < 10 \end{cases} \quad (2,33)$$

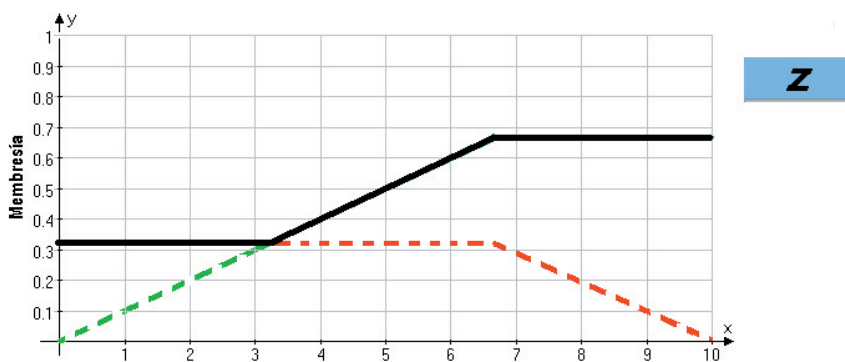


ILUSTRACIÓN 8. Conjunto difuso resultante de la etapa de *composición*.

Si es suficiente con el resultado difuso obtenido, entonces el proceso de inferencia tiene fin en esta fase. En cambio, si el resultado deseado debe ser expresado en un valor nítido, entonces se hace necesario un paso más: la nitidización del conjunto difuso resultante.

- Etapa 4. *Nitidización.*

Como último paso en el proceso de inferencia difusa, se hace necesario en algunos casos el establecimiento de un valor nítido a partir del conjunto difuso resultante. Para ello, se puede emplear cualquiera de los métodos de nitidización expuestos en el apartado 2.1.4 de este documento. Si para este caso específico se empleara el método *Primero de los máximos*, entonces el valor nítido de z sería igual a 6.8 (ver ilustración 9).

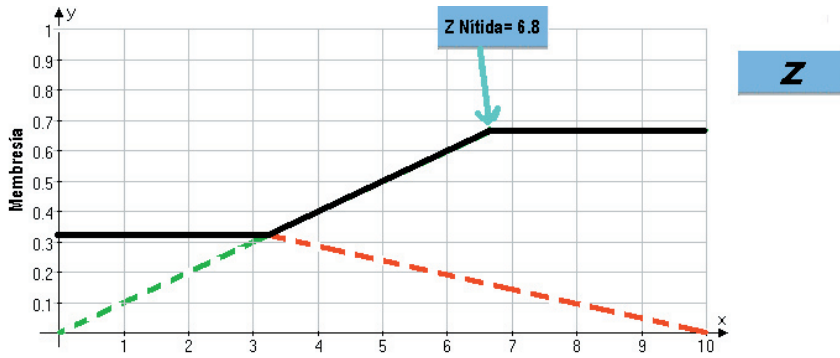


ILUSTRACIÓN 9. Valor nítido de z , calculado con base en el *primero de los máximos*.

Si se usa el método de nitidización llamado *Media de los máximos*, entonces el valor nítido de z es igual a 8.4 (ver ilustración 10).

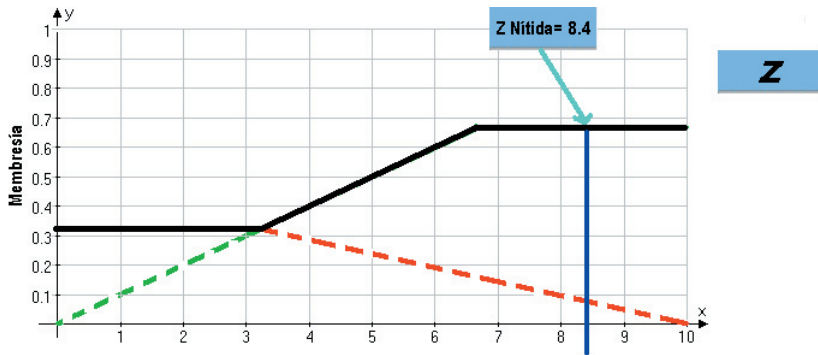


ILUSTRACIÓN 10. Valor nítido de z , calculado con base en la *media de los máximos*.

Si el valor de z se calcula con base en el método *Centro de Gravedad*, entonces su valor es igual a 5.6 (ver ilustración 11).

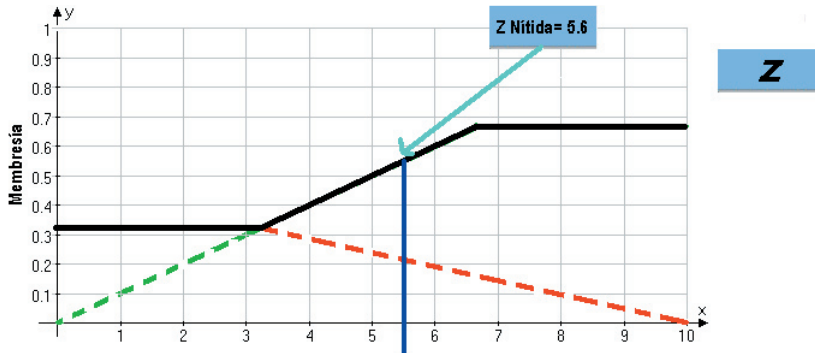


ILUSTRACIÓN 11. Valor nítido de z , calculado con base en el *centro de gravedad*.

2.1.5. Nitidización¹⁴

La salida de un sistema difuso es el producto de una o varias inferencias realizadas, sin embargo, esta salida se ofrece en términos difusos que, a menudo, no son los más adecuados para la realización de una acción o la toma de una decisión. Es por ello que se requiere, en muchos sistemas, realizar un proceso para volver *nítidas* las salidas difusas.

La nitidización de una salida difusa se puede definir como un proceso consistente en la selección de un elemento representativo de una salida difusa C inferida por un sistema difuso:

$$Z_0 = \text{Nitidización}(C) \quad (2,34)$$

En donde Z_0 es la salida nítida, y $Z_0 = \text{Nitidización}(C)$ es el operador aplicado sobre el conjunto difuso C para obtener la salida nítida.

Los operadores de nitidización dan nombre al método que con base en el uso de éstos pretende la conversión de un valor difuso en un valor nítido. Entre los métodos de nitidización más comunes se puede mencionar a:

¹⁴ *Defuzzification* en inglés.

1. *Centro de área* o *Centro de Gravedad*.¹⁵ El valor nítido de un conjunto difuso C es definido a partir de su *centroide difuso* calculado para un universo de discurso finito W :

2.

$$Z_0 = \frac{\sum z_j C(z_j)}{\sum C(z_j)} \quad (2,35)$$

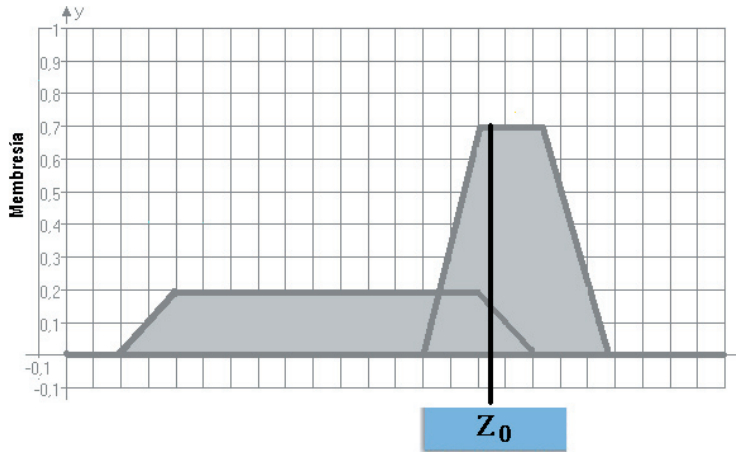


ILUSTRACIÓN 12. Valor nítido de una salida difusa calculado con base en el centro de gravedad.

3. *Centro del área mayor*. El valor nítido de un conjunto difuso C es definido a partir del centroide, pero solo calculado a partir del área del conjunto difuso mayor en un universo de discurso finito W :

¹⁵ *Center of Area*, o *Center of Gravity*, en inglés.

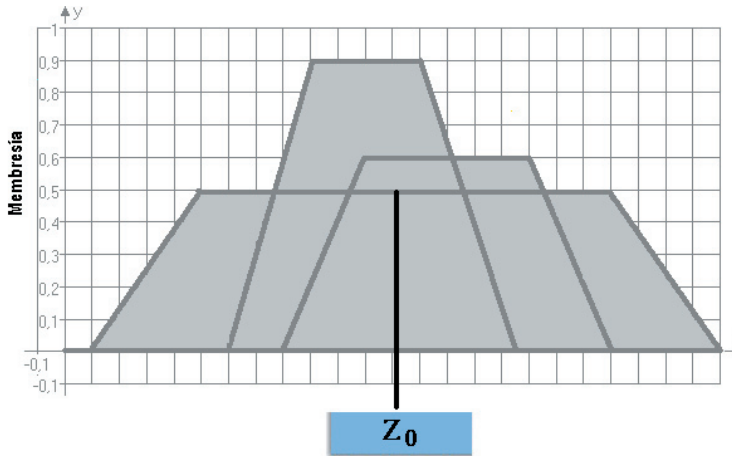


ILUSTRACIÓN 13. Valor nítido de una salida difusa calculado con base en el centro del área mayor.

- Primero de los máximos. El valor nítido de un conjunto difuso C es el menor de los elementos con más alto grado de membresía.

$$z_0 = \min\{z \mid C(z) = \max_w C(w)\} \quad (2,36)$$

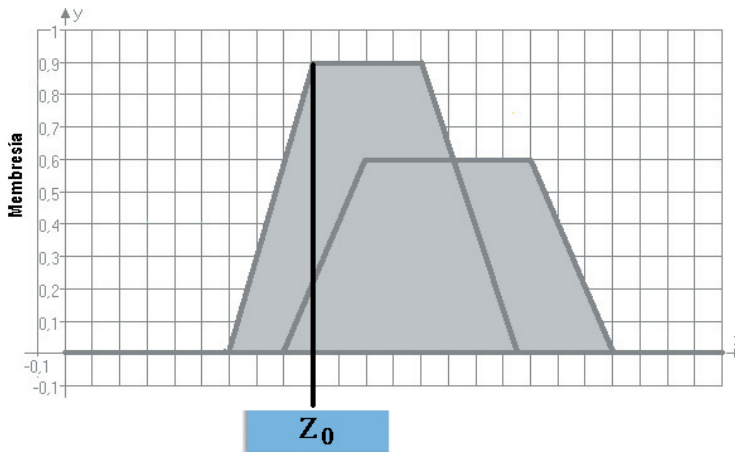


ILUSTRACIÓN 14. Valor nítido de una salida difusa calculado con base en el primero de los máximos.

5. Mitad de los máximos. El valor nítido de un conjunto difuso discreto C está definido como la media de todos los valores del universo de discurso que tienen los grados de membresía más altos.

$$z_0 = \frac{1}{N} \sum_{j=1}^N z_j \quad (2,37)$$

En donde: $\{z_1, \dots, z_N\}$ es el conjunto de elementos discretos en el universo W que tiene el máximo valor de C .

Si C no es un conjunto discreto, entonces el valor nítido se calcula con base en:

$$z_0 = \frac{\int_G z \cdot dz}{\int_G dz} \quad (2,38)$$

En donde G denota el conjunto de maximización de elementos de C

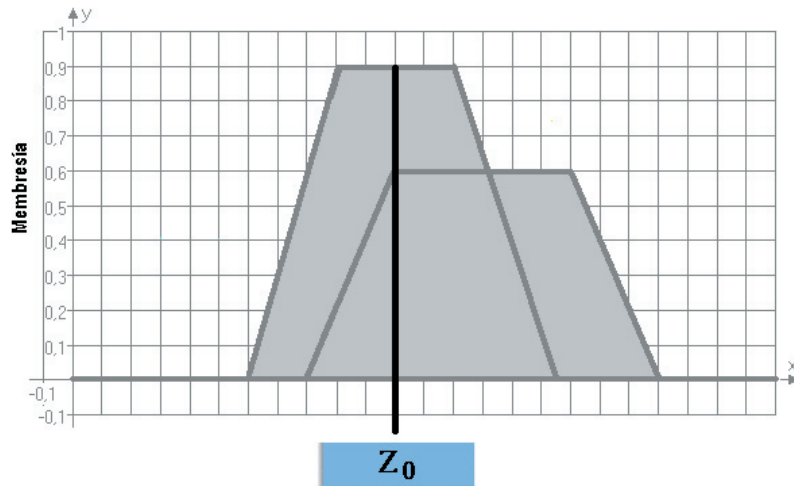


ILUSTRACIÓN 15. Valor nítido de una salida difusa calculado con base en la mitad de los máximos.

6. Máximo a criterio. Esta técnica consiste en la elección de un valor arbitrario, del conjunto de elementos máximos de C .

$$z_0 \in \{z \mid C(z) = \max_w C(w)\} \quad (2,39)$$

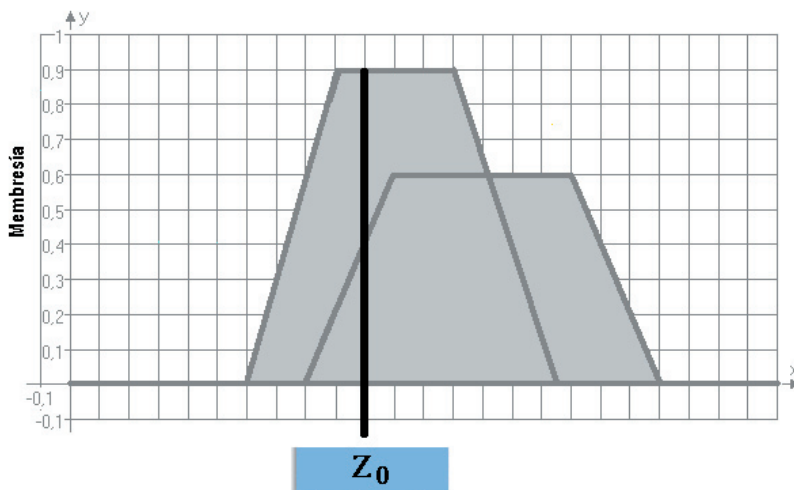


ILUSTRACIÓN 16. Uno de los posibles valores nítidos calculado con base en un máximo a criterio.

7. Nitidización con nivel mínimo¹⁶

En esta técnica, los elementos del universo de discurso W que tienen grados de membresía menores que cierto nivel α son completamente ignorados y el valor nitidizado z_0 se calcula a partir de la técnica basada en el centro del área, sobre aquellos elementos que tienen un grado de membresía no menor a α :

$$z_0 = \frac{\int_{[C]^\alpha} zC(z)dz}{\int_{[C]^\alpha} C(z)dz} \quad (2,40)$$

¹⁶ Height defuzzification en inglés.

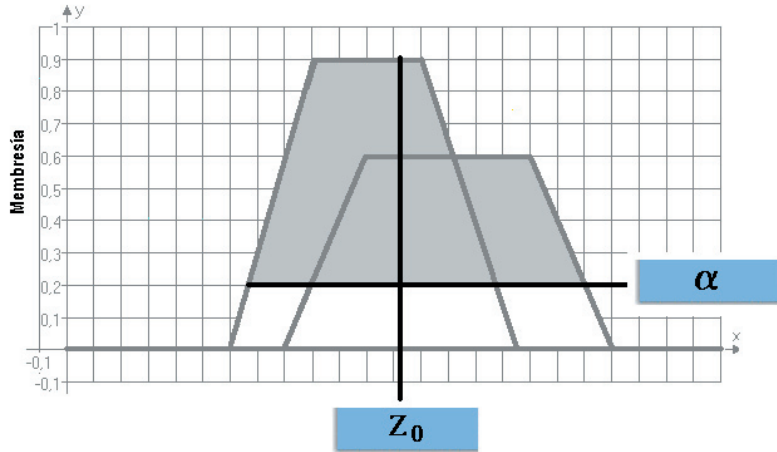


ILUSTRACIÓN 17. Valor nítido de una salida difusa calculado con base en un nivel mínimo.

Donde $[C]^\alpha$ denota al nivel mínimo α establecido para el conjunto C .

2.2. Elementos y organización de un sistema difuso

Un sistema difuso es un conjunto de elementos organizados con el propósito de procesar información difusa con fines de inferencia y control; establece una relación de correspondencia entre una entrada x y una salida $F(x)$ (Kosko, 1992).

Un sistema difuso puede, en teoría, modelar cualquier sistema (ver (Kosko, 1992) y (Hecht-Nielsen, 1988)), e incluso llegar más lejos; con la ayuda de la lógica difusa se pueden modelar o controlar sistemas donde no se tiene un modelo matemático acerca de cómo las salidas dependen de las entradas. Un sistema difuso usa entonces, reglas difusas para producir un puente desde el espacio de entradas al espacio de salidas, en lugar de emplear modelos matemáticos precisos o *modelos de planta*, como también se les llama (Kosko, 1993).

Los sistemas difusos, usualmente se componen de cuatro partes principales: una interfaz de difuminación, una base de reglas difusas, una máquina de inferencias difusas y una interfaz de nitidización (ver ilustración 18).

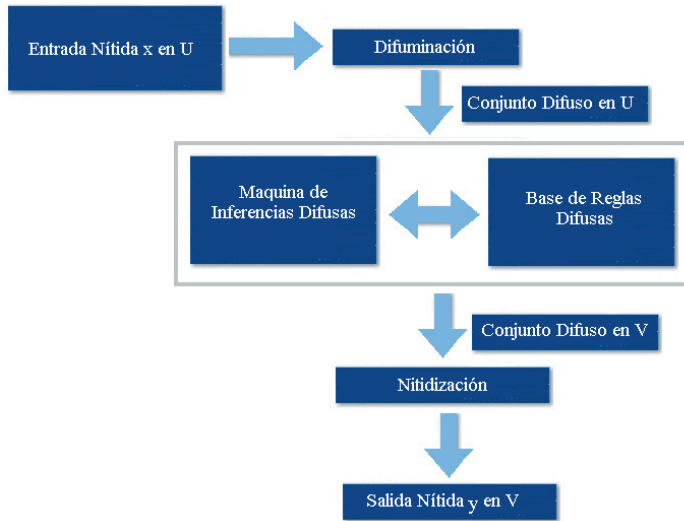


ILUSTRACIÓN 18. Esquema de un sistema difuso típico.

2.2.1. Interfaz de difuminación

La interfaz de difuminación tiene como función el proceso inicial de las entradas del sistema. Es en este componente donde se convierten las entradas nítidas alimentadas en el sistema, en entradas difusas. Para ello se puede emplear, de acuerdo a las características específicas del problema que se pretende resolver, alguno de los dos enfoques de difuminación ya descritos, o una combinación de ambos.

2.2.2. Base difusa de conocimientos

La base de conocimientos de un sistema difuso típico debe integrar por lo menos dos elementos: una *base de datos* y una *base de reglas difusas*. Los conceptos asociados con una base de datos se usan para caracterizar reglas de control difuso o para manipulación de datos difusos en el sistema. Estos conceptos se definen subjetivamente y dependen del juicio y la experiencia del programador. Es por ello que se puede afirmar que la elección de las funciones de membresía y de las variables lingüísticas tiene un papel muy importante en el desempeño de un sistema difuso.

La construcción de una base de conocimientos difusos sólida implica los siguientes pasos:

1. La creación de tantos conjuntos difusos como sean necesarios, de manera tal que la unión de sus soportes¹⁷ haga una cobertura del universo de discurso en relación con un nivel ε previamente establecido. A esta propiedad de un sistema difuso se le llama *entereza* ε .¹⁸ En general se elige un nivel de *entereza* ε para los puntos de cruce de los conjuntos difusos que implique que las reglas difusas se asocian con el sistema con un nivel de certeza sólido. Este nivel, por lo general se fija como mínimo en 0,5 (ver ilustración 19).

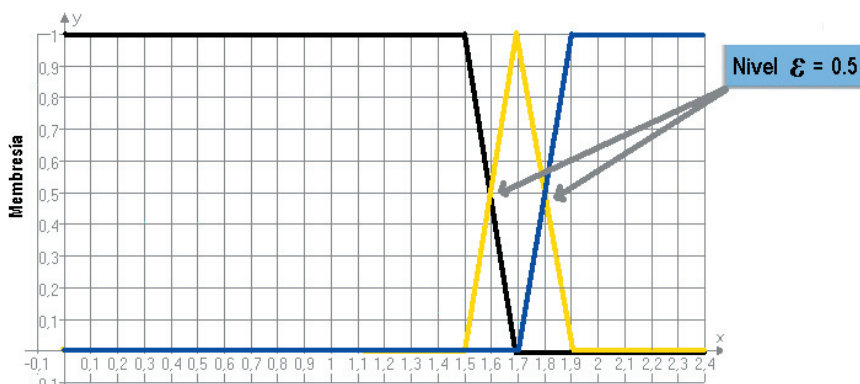


ILUSTRACIÓN 19. Fijación del nivel de entereza de un sistema difuso.

2. Cuantificación de los universos de discurso, o también conocido como discretización. El objetivo es cuantificar un universo segmentándolo en un cierto número de bloques (llamados *niveles de cuantificación*). Cada bloque se etiqueta como elemento genérico y conforma un universo discreto. Entonces se define un conjunto difuso asignándole un valor que indica el grado de membresía a cada elemento de este nuevo universo discreto.

17 En lógica difusa se define *soporte de un conjunto difuso* de la siguiente manera: sea A un subconjunto difuso de X , el soporte de A , denotado como $sup(A)$, es el subconjunto *nítido*, es decir un subconjunto con límites no difusos, de elementos de un conjunto X , que tienen grados de membresía mayores de cero, $sup(A) = \{x \in X \mid A(x) > 0\}$.

18 ε *completeness* en inglés.

En el caso de universos de discurso continuos, el número de niveles de cuantificación debe ser lo suficientemente grande para proveer una aproximación que sea adecuada, pero lo menor posible de tal manera que permita ser manejada (estableciendo esto con base en los recursos disponibles para el sistema). La elección de los niveles de cuantificación es determinante para el grado de finura o resolución que se quiere manejar en el sistema.

Por ejemplo, si un universo continuo es cuantificado con un bloque cada cinco unidades en lugar de cada diez, entonces el sistema tendrá una resolución mayor.

3. Partición difusa de los espacios de entrada y salida. Una variable lingüística en el antecedente de una regla difusa conforma un espacio de entradas difusas en relación con cierto universo de discurso, mientras que la consecuencia de la regla conforma un espacio difuso de salidas. En general, una variable lingüística se asocia con un conjunto de términos definido en el mismo universo de discurso. Una partición difusa determina entonces cuántos términos deben existir en el conjunto de términos.
4. Entereza.¹⁹ Como se mencionó ya en párrafos anteriores, la entereza de un sistema se mide por el grado de cobertura que el sistema tiene sobre los espacios posibles de estados del problema.
5. Elección de las funciones de membresía de los conjuntos primarios. Existen dos métodos dependiendo del universo de discurso del que se trate. Si el universo de discurso es discreto entonces se requiere de una definición funcional de las funciones de membresía, en cambio, si el universo de discurso es continuo, entonces se requiere de una definición de funciones numérica.
 - Definición funcional de funciones de membresía: Una definición funcional de funciones de membresía, comúnmente usa funciones de tipo campana, trapezoidales o triangulares. Estas funciones son usadas en los sistemas difusos debido a que ayudan a la manipulación de la aritmética difusa (Gupta y Qi, 1992). La definición funcional de las funciones de membresía ayuda a adaptar las funciones establecidas a cambios en la normalización del universo de discurso. La asignación de los valores de membresía descansa en procesos de decisión subjetivos.

19 *Completeness*, en inglés.

- Definición numérica de las funciones de membresía: en universos continuos, las funciones de membresía de conjuntos difusos se representan como vectores de números cuya dimensión depende del grado de discretización empleado. Al igual que en el caso de universos discretos, las asignaciones de los valores de membresía dependen directamente de la subjetividad del programador del sistema o del conjunto de asesores que hayan participado en la elaboración del sistema.
6. Base de reglas. Un sistema difuso se caracteriza por un conjunto de afirmaciones realizadas con base en el conocimiento proporcionado por expertos. El conocimiento de los expertos usualmente se obtiene representado con reglas del tipo “Si- Entonces”, que pueden ser fácilmente implementar en condicionales difusas. Esta colección de reglas difusas es conocida como base de reglas o base de reglas difusas en un sistema difuso.
 7. Fuente y generación de las reglas difusas. Existen cuatro modos principales de generación de reglas difusas para sistemas difusos:
 - Recuperación de las experiencias de los expertos. Las reglas difusas son afirmaciones condicionales difusas que conectan estados variables a una decisión, conclusión o acción de control, que es consecuencia directa de los mismos.
 - Acciones del experto. En muchos casos, sobre todo relacionados con el conocimiento de tipo procedural, los expertos no tienen en mente la sistematización de sus acciones: simplemente las realizan, es por ello que otra forma de obtener el conocimiento de los expertos es el monitoreo y registro de sus acciones. En este caso los términos lingüísticos que se asignan a las acciones en un proceso, son definidos y afinados con base en la observación de las acciones.
 - Modelado difuso de un proceso. La descripción verbal de muchas de las características de un proceso puede ser usada como un modelo difuso del mismo proceso, a partir del cual se pueden general reglas de decisión o de control. Este proceso por sí sólo resulta complicado y necesita la ayuda de alguno de los dos enfoques anteriores.
 - Aprendizaje. Muchos de los sistemas difusos han sido diseñados para emular algunos de los aspectos que tienen lugar en la toma

de decisiones del ser humano, sobre todo, en lo relacionado con el aprendizaje que da soporte a una toma de decisiones adecuada. Es decir, se relaciona con la habilidad de modificar una regla generada con base en sus efectos y la experiencia obtenida a través de varios procesos.

2.2.3. Motor de inferencias difusas

El motor de inferencias difusas aplica alguno de los enfoques descritos en el apartado 2.1.4. para realizar inferencias difusas. Sus insumos principales son las reglas de la base de conocimientos y las entradas actuales del problema en cuestión que el sistema está tratando de resolver.

2.2.4. Interfaz de nitidización

La interfaz de nitidización en un sistema difuso tiene como objetivo aplicar alguna de las técnicas de nitidización como las revisadas en el apartado 2.1.4 de este mismo capítulo. Esta interfaz ofrece, finalmente, al usuario o al sistema receptor, un valor nítido que dé la pauta sobre la acción o las acciones pertinentes a realizar.

Existen casos en los que se debe dotar a la interfaz de nitidización con la habilidad para aplicar más de una técnica de nitidización y con la capacidad, además, de tomar decisiones en cuanto a la técnica más pertinente para cada caso. Esto es lo que permite a un sistema difuso un desempeño eficiente aún cuando la naturaleza de la salida, a partir de la cual el sistema difuso debe calcular el valor nítido, presente problemas.²⁰

2.3. Funcionamiento de un sistema difuso

El funcionamiento típico de un sistema difuso está relacionado estrechamente con el tipo de problema real que se intenta resolver con él y para el cual fue creado. Sin embargo, se pueden identificar aspectos generales de su funcionamiento:

20 Este aspecto es importante sobre todo, cuando se calcula la salida nítida en el sistema con base en un solo conjunto difuso de la salida (por ejemplo en el método de *Centro del área mayor*) y los criterios para la selección se cumplen para más de un conjunto.

1. Dado que la mayoría de los sistemas difusos ha sido diseñada para tareas de control, el papel que el usuario suele jugar en otro tipo de sistemas, es realizado en este caso, por otro sistema quien provee las entradas que se procesarán.
2. El sistema recibe las entradas y realiza un proceso para convertir el valor nítido de cada una de ellas a valores difusos. Estos valores difusos son usados por el motor de inferencias difusas para, a partir de una base de reglas o conocimientos difusas, producir una salida difusa que en el caso de una aplicación para control, indica el tipo de acción que se debe realizar a partir de estas entradas. Para este proceso de inferencia el sistema puede emplear reglas de inferencia como las presentadas en el apartado 2.1.3 las cuales se aplican con base en las reglas contenidas en la base de conocimientos difusos que el mismo sistema integra.
3. El usuario, o más comúnmente en este caso, el sistema efector, recibe la recomendación acerca de la acción a realizar, expresada en valores nítidos, después de que la salida difusa de la fase anterior ha sido nitidizada por la interfaz dispuesta para tal efecto.

2.4. Sistemas difusos e inteligencia artificial

El procesamiento de la información de forma difusa difiere grandemente de las formas tradicionales de proceso de información incierta en la Inteligencia Artificial. Al igual que en otras líneas de investigación como las redes neuronales, la lógica difusa no se considera como una línea de trabajo de la inteligencia artificial por muchos autores, aunque si se reconoce que es una disciplina que está estrechamente asociada. Inclusive en un nivel práctico, la lógica difusa se considera como una herramienta eficiente para el manejo de información incierta aún en las líneas clásicas de la Inteligencia Artificial (Gallant, 1994).

Las características que permiten establecer las diferencias entre la lógica difusa y las líneas clásicas en inteligencia artificial son:

- La forma de manejar la incertidumbre. Mientras que en las líneas clásicas de la inteligencia artificial se hace uso del manejo estadístico de la incertidumbre, la lógica difusa ofrece su propio marco teórico para manejarla, en el que el proceso estadístico juega el papel solamente de herramienta entre muchas otras.

- La forma de realizar el razonamiento bajo condiciones de incertidumbre también difiere de las formas clásicas, aunque en algunos casos es posible identificar aún la influencia de algunas de las técnicas usadas en MYCIN sobre las reglas de inferencia propuestas por Zadeh (apartado 2.1.3. en este mismo capítulo). Este hecho puede ser que no se deba a una inspiración directa, pero al menos si sugiere un origen común.

2.5. Críticas actuales a los sistemas difusos

Los sistemas difusos han llegado a ser considerados como una verdadera opción para el proceso de información incierta, inexacta o incompleta. Sobre todo en el área de control, donde parece haber tenido más repercusiones la implementación de la lógica difusa. Sin embargo, este reconocimiento no es generalizado, existen muchas críticas a esta forma especial de proceso de la información, principalmente de dos vertientes distintas: a) Se crítica a los sistemas difusos desde sus mismos orígenes, es decir, desde las críticas que los lógicos y los matemáticos formales clásicos imponen a la lógica difusa, y b) Se critica a los sistemas difusos desde los ámbitos de trabajo en donde las aplicaciones con este tipo de tecnología se ponen en práctica.

En los apartados siguientes se ofrece un breve sumario de los puntos de crítica más importantes desde estas dos vertientes.

2.5.1. Críticas heredadas de la lógica difusa

Es significativo el hecho de que la lógica difusa, y por herencia los sistemas difusos, no haya tenido sus detractores en el campo de las aplicaciones prácticas que de ésta se realizan; las críticas más sobresalientes provienen de las matemáticas formales. Asimismo, es igualmente significativo que estas críticas alcancen a los sistemas difusos, no obstante su gran éxito en la práctica.

Quizá la crítica más difundida la ha realizado Susan Haack, una matemática formal inglesa quien actualmente realiza investigaciones principalmente en la Escuela de Leyes de la Universidad de Miami en los Estados Unidos. Haack, argumenta que solamente hay dos áreas en las que se pudiese demostrar que la lógica difusa se puede realmente necesi-

tar y mantiene la postura de que en ambos casos se puede demostrar que no se necesita (Haack, 1979):

- La primera área es la naturaleza de la verdad y la falsedad, si se puede mostrar, ella mantiene que en este caso se tiene que hablar de valores difusos y no discretos, entonces la necesidad de la lógica difusa podría demostrarse con base en estos argumentos.

Considerando esta *primera necesidad identificada*, como la Dra. Haack le llama, argumenta que verdadero y falso son términos discretos. Expresiones como “el cielo es azul”, por ejemplo, pueden ser cualquiera de estas dos opciones: *verdaderas* o *falsas*. Cualquier medida difusa de la oración viene de la definición imprecisa de los términos y no de la naturaleza de la verdad.

- La otra área tiene una estrecha relación con los sistemas difusos y su utilidad. Dice Haack que si se pudiese demostrar que la lógica difusa puede auxiliar a la lógica clásica en algunos cálculos que ésta no pueda realizar. Solo entonces, la lógica difusa, podría ser considerada como un subconjunto de la lógica clásica general, dado que se habría demostrado que es *completamente necesaria*.

A este respecto, Haack argumenta, que la lógica difusa no se hace necesaria, ya que lejos de ayudar en la realización de algunos cálculos *los complejiza aún más*. Con lo que problemas originalmente simples para la lógica tradicional, se vuelven más complejos.

Estos dos argumentos son la base para demostrar, en palabras de la Dra. Susan Haack, que la lógica difusa no es necesaria y por herencia, los sistemas difusos tampoco; ya que a decir de ella cualquier resultado que se obtenga por medio de la lógica difusa se puede obtener por medios formales (Haack, 1979).

Fox (Fox, 1981), en cambio, ha colaborado con sus trabajos a responder a las objeciones de la Dra. Haack, indicando que existen tres áreas en las que la lógica difusa puede ser benéfica:

- Primero, como un *aparato requerido* para describir relaciones en el mundo real que son inherentemente difusas,
- Segundo, como un *aparato prescriptivo* debido a que algunos datos son difusos y por lo tanto se requiere de cálculos difusos, y
- Finalmente, la lógica difusa es benéfica también como un *aparato descriptivo*, debido a que algunos sistemas de inferencia son inherentemente difusos.

El argumento más sólido de Fox proviene de la noción de que lógica difusa y lógica clásica no deben ser vistas como oponentes en competencia, sino como marcos complementarios. Fox mantiene que las objeciones de Haack se basan en las deficiencias de la claridad semántica y que, en último caso, las oraciones difusas podrían traducirse en frases para las cuales los lógicos clásicos no tuvieran objeción.

Finalmente, Fox argumenta que a pesar de las objeciones por parte de los lógicos clásicos, la lógica difusa ha encontrado un camino de desarrollo en el mundo de las aplicaciones prácticas y ha probado ser exitosa en ello. Fox sostiene pragmáticamente que esta es una razón suficiente para continuar con la búsqueda del desarrollo de este campo.

2.5.2. Críticas restringidas solamente a los sistemas difusos

Pero no todas las objeciones a los sistemas difusos son derivadas directamente de las críticas a la lógica difusa. Existe otro conjunto de objeciones que atañen a problemas que no se han podido resolver en la implementación de este tipo de sistemas. A continuación se describen tres de los problemas más sobresalientes:

- *Su imposibilidad para obtener o aprender conocimientos por medios automáticos.* En los sistemas difusos el proceso de estructuración de la base de conocimientos difusos tiene que ser definido y modificado, si es el caso, por el creador del sistema (Von Altrock, 1995).
- *Estabilidad no asegurada.* No existe una garantía teórica de que los sistemas difusos permanezcan estables, es decir, que no se vuelvan caóticos, ante el manejo de enormes bases de conocimientos difusos o mucho más complejas de lo que actualmente son.
- *Alta dependencia de la subjetividad humana.* No existe una manera de asegurar que las funciones de membresía han sido calculadas adecuadamente. Hasta el momento los procedimientos para este cálculo dependen en gran medida del ser humano que lo realiza y sus criterios muy personales.
- *Verificación y validación.* Estos dos procesos siguen siendo predominantemente artesanales y muy costosos. Muchas de las aplicaciones actuales no pueden ser tomadas muy en serio, pero no se debe a deficiencias en el diseño del sistema en sí, sino a deficiencias en el proceso de refinamiento de los conocimientos.

2.6. Actualidad y nuevas líneas de trabajo en sistemas difusos

Una nueva ola de productos comerciales difusos ha dado impulso a la lógica difusa aún cuando el interés por ésta no ha dejado de incrementarse desde su creación. En 1980 la firma de contratistas F.L. Smidth & Co, fue la primera en usar un sistema difuso para controlar un horno para cocimiento de cemento. El sistema difuso diseñado para este fin, usa reglas para establecer relaciones entre la temperatura y el oxígeno contenidos en la salida del gas, el torque del horno, la temperatura del caolín y los rangos de flujo del aire y de alimentación.

En 1988 Hitachi logró controlar un tren subterráneo metropolitano con un sistema difuso,²¹ en la ciudad japonesa de Sendai, desde ese entonces, Japón ha producido cientos de sistemas “inteligentes” que usan lógica difusa; productos que van desde cámaras de video, máquinas lavadoras, hornos de microondas, televisores, transmisiones de autos, lectores de discos para computadoras, máquinas automatizadas para cortar pasto, tostadores de pan, etc.

El crecimiento del uso de la lógica difusa como enfoque para la creación de sistemas de control ha sido muy significativo. Baste decir que de 8 aplicaciones industriales con base en sistemas difusos registradas en 1986, pasaron a ser 1500 para 1993 (Fullér, 1995, citando a Munkata y Jani, 1994). En la actualidad una búsqueda simple en Internet puede conducir a más de 2000 centros o laboratorios de investigación asociados a Universidades o empresas privadas en todo el mundo, cada uno de los cuales pueden ofrecer desarrollos propios entre los cuales se integra algo del proceso difuso de la información.

Actualmente es posible también implementar sistemas difusos con microprocesadores. Los circuitos integrados para proceso difuso que los sistemas actuales usan varían en número y características, de acuerdo al problema que se quiere resolver. Por ejemplo, tostadores difusos de pan y máquinas lavadoras con control de temperatura difuso pueden tener sólo un circuito integrado, mientras que un automóvil moderno o una máquina perforadora para minas a cielo cerrado pueden tener entre cincuenta y cien.

21 El control difuso ha probado incluso ser más eficiente en el ahorro de energía y permite a los humanos mayor confort en el manejo del tren. Actualmente el sistema maneja el tren en las *horas pico*, mientras que en las horas sin tráfico se sigue dejando su manejo a seres humanos, pero solamente con la finalidad de que éstos no olviden cómo se maneja.

3

Redes neuronales¹

3.1. introducción

El cerebro humano no razona como lo hace actualmente la mayoría de las computadoras, es por ello que, filósofos de la mente como Pagels (Pagels, 1991), Ruyer (Ruyer, 1984), Dreyfus (Dreyfus, 1993), Graubard (Graubard, 1993), Haugeland (Haugeland, 1988), o aún más directamente, James McClelland y David Rummelhart (McClelland y Rummelhart, 1981), han señalado la necesidad de cambiar las formas, tradicionalmente binarias, de trabajo de los sistemas computacionales, para obtener algo más cercano al desempeño del cerebro humano. Sin embargo, a pesar de que todos estos autores hacen referencia a la necesidad de cambiar la forma en la que la información se procesa actualmente, las posturas sobre la dirección específica que debe tomar este cambio son diversas. Destaca, sobre todo, como punto en común, el énfasis que ponen en que el objetivo debe ser la implementación de abstracciones a partir del funcionamiento y arquitectura cerebrales,² más que la replicación del cerebro en una máquina.

1 En el campo de las neurociencias, cuando se habla de redes neuronales se hace referencia a redes biológicas de neuronas, y por ello, en este dominio se hace necesario emplear el término de *redes neuronales artificiales* para referirse a las redes no biológicas. En cambio, en el campo de la inteligencia artificial o las ciencias computacionales, el término *redes neuronales*, hace referencia a las redes neuronales artificiales y para hacer referencia a las biológicas es necesario especificar que se habla de *redes neuronales biológicas*. Por la naturaleza de este documento, el término *redes neuronales* se utiliza para las redes artificiales siguiendo la terminología usada en inteligencia artificial.

2 Al respecto H. R. Pagels (Pagels, 1991) explica como a través de los años el hombre tratado de imitar a la naturaleza para lograr, con máquinas, tareas complejas que sólo ciertos organismos pueden realizar. Asimismo explica cómo la solución no viene tras la imitación, sino después de un proceso de abstracción a partir de los modelos observados. Por ejemplo se puede tomar a las máquinas que vuelan; no lo hacen imitando a las aves, sino con base en modelos abstraídos a partir del vuelo de éstas.

A estas abstracciones se les conoce, en conjunto, como *Computación Neuronal* o *Neurocomputación* (Hecht-Nielsen, 1989). Se les ha propuesto, no como alternativa a los enfoques actuales, sino más bien como una manera obtener aplicaciones prácticas y funcionales, que cuenten con las propiedades y ventajas que el procesamiento paralelo y distribuido ofrece (McClelland, Rummelhart y Hinton, 1994). Los sistemas computacionales realizados con base en éstos modelos son conocidos como *Redes Neuronales Artificiales* (RNA)³ o, simplemente *redes neuronales* (RN).

Parece no haber una definición universal acerca de lo que son las redes neuronales. Sin embargo, con base en su arquitectura y función, se podría decir que las redes neuronales artificiales son dispositivos computacionales compuestos a partir de la interconexión de un cierto número de unidades simples, cuya función principal es la transmisión de información numérica. Los canales de transmisión de la información, las reglas para que esta transmisión se lleve a cabo, e incluso la misma arquitectura específica de la red, dependen de la función para la cual la red ha sido diseñada y entrenada; función relacionada, principalmente, con la identificación de patrones.

El estudio de las redes neuronales inició en la primera década de los cuarenta, cuando McCulloch y Pitts (ver (McCulloch y Pitts, 1943)) sentaron las bases para tratar al cerebro como un organismo computacional basándose en cinco supuestos fundamentales de la neurofisiología de los cuarenta:

- La transmisión neuronal es un proceso regido por la *Ley del Todo o nada*. Es decir, las neuronas son binarias en lo que respecta a su actividad: o transmiten, o no lo hacen. No hay puntos intermedios de transmisión.
- Para la excitación de una neurona se hace necesario un número fijo de sinapsis en un periodo de adición latente. Las neuronas cuentan con un umbral de excitación fijo, que solamente es rebasado cuando todas las sinapsis que en un tiempo determinado entran a la neurona, producen un estímulo total suficiente para sobrepasar este umbral.
- No hay retardos en la actividad del sistema nervioso ajenos al mismo retardo sináptico.
- Cuando la sinapsis es inhibitoria, la actividad de la neurona implicada se suspende en un periodo de tiempo.

3 A menudo se usa también el término *Redes neuronales* (o *Neural networks*, en inglés) para hacer referencia a la Neurocomputación en general.

- Las interconexiones entre neuronas no cambian con el tiempo.

Con base en estos supuestos, McCulloch y Pitts idearon un modelo computacional basado en redes de elementos sencillos que podían emular este supuesto funcionamiento del sistema nervioso. Las redes configuradas eran capaces de realizar operaciones lógicas complejas con base en la combinación de módulos sinápticos que realizaban sólo tres operaciones básicas: la conjunción, la negación y la disyunción.

Sin embargo, el trabajo de McCulloch y Pitts adquirió mayor relevancia cuando a estas redes artificiales de neuronas se les añadió la capacidad de aprender con base en las ideas propuestas por Donald Hebb en 1949 (Ver (Hebb, 1949)):

...cuando un axón de una neurona A está lo suficientemente próximo para excitar a una neurona B, o cuando ésta toma parte frecuentemente en la excitación para producir su disparo, se da un proceso de crecimiento o cambio de metabolismo en una o ambas neuronas, de modo que la eficiencia de A para producir el disparo de B se ve incrementada” (Hebb, 1949, pág. 50).

Este hecho le sugirió a Donald Hebb la posibilidad de haber encontrado el sustrato neurofisiológico del aprendizaje:

“... se debe asumir que este cambio estructural a nivel sináptico sucede en situaciones de aprendizaje [...] ya que varias neuronas que mantienen actividad repetidamente terminan *asociadas...*” (Hebb, 1949, pág. 52).

Hoy en día se ha descubierto ya, que los supuestos en los que McCulloch y Pitts fundamentaron su trabajo, así como la idea de Hebb acerca del aprendizaje, no son del todo ciertos. Funcionaron como base de un nuevo modelo computacional de procesamiento de información que es hoy conocido como redes neuronales, y que poco a poco ha ido desarrollando sus propios modelos sin que éstos necesariamente observen características que hoy conocemos en los seres vivos. El Desarrollo histórico de las redes neuronales artificiales ha sido entonces, independiente de los avances en materia de neurociencias, aunque una parte de las redes neuronales no se aparta de la búsqueda de la plausibilidad de los fenómenos computacionales logrados, en tejidos vivos.

Actualmente se conoce mucho más acerca de la organización anatómica del Sistema Nervioso de los mamíferos; los tipos de células nerviosas que intervienen en los circuitos neuronales vivos; los mecanismos biológicos, eléctricos y químicos que desencadenan los disparos neuronales y el papel de los neurotransmisores en la actividad nerviosa y la propagación de las señales (ver Kandel, Schwartz y Jessell, 2000); sin embargo, muchos de los desarrollos actuales en redes neuronales, no se apegan ya a estos descubrimientos, y aquellos que sí pretenden hacerlo en alguna forma, han pasado a formar parte de lo que se conoce como *vida artificial* (Pollack, Bedau, Husbands, Ikegami y Watson, 2004).

3.2. Elementos y organización de una red neuronal

Existen muchos tipos diferentes de redes neuronales, y todos ellos implican la integración de diversos elementos y formas de organización. Sin embargo, todos los tipos contemplan unidades de proceso como la que se describe en el apartado siguiente.

3.2.1. Unidades de proceso

Las unidades de proceso en una red neuronal se llaman *neuronas artificiales*, *nodos*, o simplemente *neuronas*.

Cada unidad de proceso en una red neuronal se caracteriza por:

- a) Un nivel de actividad. El nivel de actividad de la unidad de proceso es binario y representa el estado de polarización de la neurona biológica,
- b) Un valor de salida. El valor de salida de las unidades de proceso en una red neuronal representa el disparo de la neurona, que se propagará a otras neuronas a través de las conexiones que ésta mantenga.
- c) Un conjunto de conexiones de entrada. Este conjunto de conexiones representa al conjunto de sinapsis entrantes en una neurona biológica y en las unidades de proceso de las redes neuronales está compuesto por el total de conexiones que el nodo recibe de otros nodos. Cada conexión tiene asociado un peso o *fuerza de sinapsis* que determina el efecto de la entrada, estos pesos pueden ser positivos (para una sinapsis excitadora) o negativos (para una sinapsis inhibitora).

- d) Un umbral de disparo, que representa el grado de resistencia al disparo, que las neuronas biológicas presentan y en el caso de las unidades de proceso se expresa como un valor.
- e) Un conjunto de conexiones de salida, que representa a las sinapsis salientes de una neurona biológica, y en redes neuronales consiste en las conexiones que de una neurona salen hacia otras. Cada conexión de salida tiene asociado también un peso o *fuerza de sinapsis* que determina el efecto de la salida. Al igual que en las conexiones de entrada, estos pesos pueden ser positivos (para una sinapsis excitadora) o negativos (para una sinapsis inhibitoria).
- f) Una función de salida o también conocida como función de activación. La interacción de las características anteriormente descritas permite establecer el funcionamiento de la unidad de proceso con base en una función de salida a partir de la cual se calcula el disparo o no que la unidad realizará. Esta función de salida puede ser de varios tipos según el modelo de la red que se esté implementando. En la ilustración 20 se muestran las tres funciones no lineales más usadas, en el entendido de que sólo son las más usadas, pero no las únicas (Lipmann, 1987).

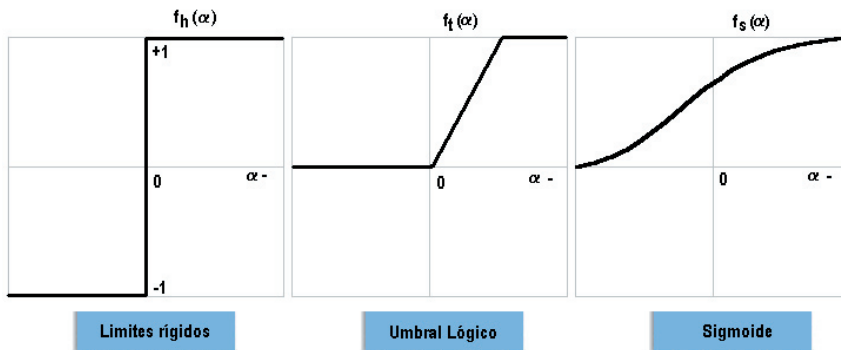


ILUSTRACIÓN 20. Tres funciones no lineales de las más usadas en redes neuronales.

Los elementos que caracterizan a las unidades de procesamiento, listados en la página anterior, se representan matemáticamente por medio de números reales (ver el ejemplo de la ilustración 21).

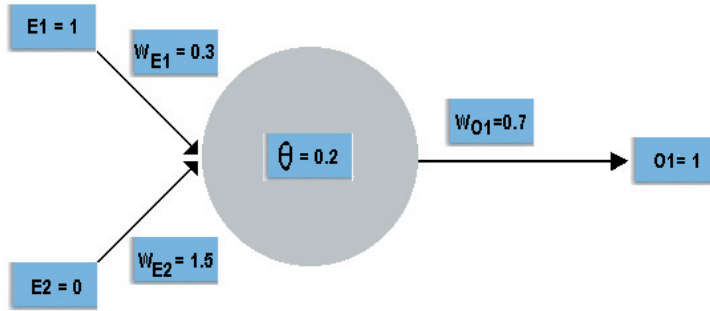


ILUSTRACIÓN 21. Unidad de proceso con valores de ejemplo.

En la ilustración 21, se muestra:

- Una neurona cuyo conjunto de conexiones de entrada esta compuesto por: $E1$ y $E2$.
- La fuerza de sinapsis o peso de conexión para $E1$ es $W_{E1} = 0.3$ y para $E2$ es $W_{E2} = 1.5$
- En ambos casos ($E1$ y $E2$) se trata de sinapsis excitadoras.
- El conjunto de conexiones de salida está compuesto por una sola salida que en este caso se ha nombrado $O1$.
- La fuerza de sinapsis o peso de conexión para $O1$ es $W_{O1} = 0.7$
- El umbral de disparo está dado por $\theta = 0.2$
- Dado que la entrada $E2$ es la única entrada activa a la neurona, el efecto total de la entrada es igual a $W_{E2} = 1.5$, que al ser mayor que el umbral implica que la neurona dispare. Es por ello que el nivel de actividad de la neurona de ejemplo es $= 1$.
- Al ser el nivel de actividad de la neurona $= 1$, se da un disparo cuya fuerza es $W_{O1} = 0.7$.
- La función de salida para la neurona de este ejemplo es la que se muestra en la ecuación (3,1) y en la ilustración 22:

$$O1 = \begin{cases} 1 & \text{si } \sum_{i=1}^n E_i \cdot WE_i \geq \theta \\ 0 & \text{de cualquier otra manera} \end{cases} \quad (3,1)$$

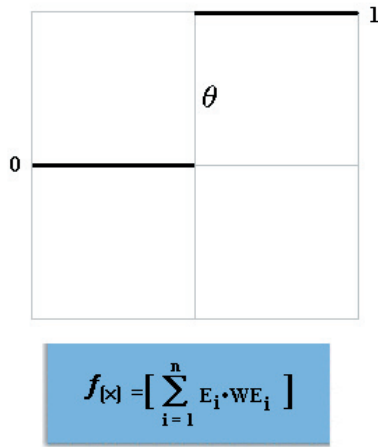


ILUSTRACIÓN 22. Función de salida para la unidad de proceso del ejemplo presentado anteriormente.

3.2.2. Organización por capas de unidades de proceso

Las redes neuronales típicamente organizan sus unidades de proceso en capas. Estas capas están compuestas de un número de nodos interconectados que usualmente emplean funciones de activación iguales.

Las entradas se presentan a la red neuronal a través de la primera capa o capa de entrada,⁴ la cual comunica con una o más capas ocultas de la red en donde el proceso de la información de entrada se realiza con base en un sistema de pesos de conexión. Las capas ocultas o intermedias, están conectadas a una capa de salida, que es el medio a través del cual la red arroja resultados (ver ilustración 23).

4 Muchos autores cuentan como capas de procesamiento solamente los nodos en los que se realiza algún tipo de cálculo. Desde este punto de vista, las neuronas de entrada no son consideradas como una capa.

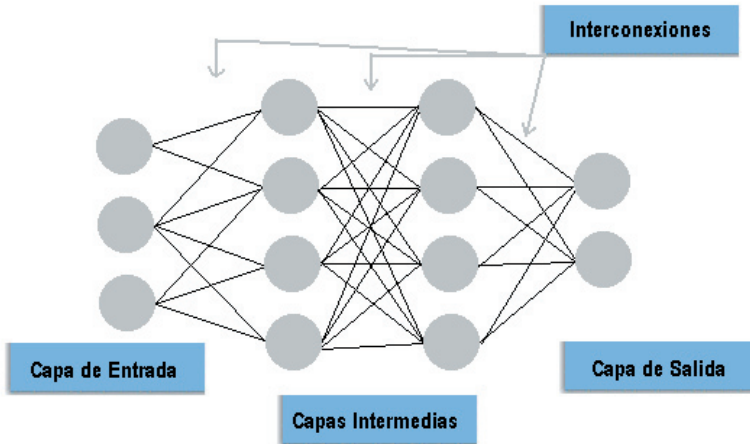


ILUSTRACIÓN 23. Capas e interconexiones en una red neuronal típica.

Las redes neuronales utilizan una regla de aprendizaje, que modifica los pesos de las conexiones entre las células de acuerdo a los patrones de entrada que le son presentados y sus salidas correspondientes. Así, estos dispositivos aprenden de la manera en que se supone que lo hacen las redes neuronales biológicas: reforzando aquellas conexiones que llevan al éxito en el desempeño de una tarea.

3.2.3. Tipos de Arquitectura

La manera en la que las neuronas de una red neuronal se estructuran se relaciona estrechamente con el modelo de aprendizaje a partir del cual la red debe funcionar. En general, se pueden identificar varios tipos de arquitectura de red, de los cuales se presentan a continuación los más comunes.

3.2.3.1. Redes neuronales de una sola capa y un solo sentido de propagación de señal⁵

La forma más simple de arquitectura de redes neuronales, en cuanto a las capas que las integran, es la red neuronal de una sola capa (o dos

⁵ *Single-layer Feedforward Network*, en inglés

capas, si se considera como capa a las unidades de entrada). Este tipo de redes neuronales se conforman a partir de una serie de nodos de entrada que están conectados directamente a los nodos de la capa de salida (ver ilustración 24).

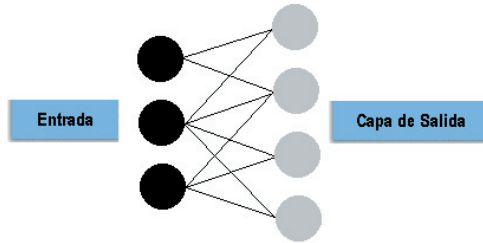


ILUSTRACIÓN 24. Red neuronal de una sola capa.

Una de las características principales de este tipo de redes es su imposibilidad para regresar la señal a las unidades de la capa de entrada. Es decir, este tipo de redes propaga la señal en un solo sentido que va desde la entrada hasta éste es de la entrada a la salida (lo que en el ámbito de las redes neuronales se conoce como propagación hacia delante o sentido de transmisión delantero).

Las memorias asociativas lineales son un ejemplo típico de esta clase de redes neuronales, en las cuales se realiza una asociación entre el patrón de entrada y la información guardada en la estructura de la red. Como en toda red neuronal, la información se integra en la estructura de la red a partir de las configuraciones de los pesos de interconexión en cada nodo de ésta.

3.2.3.2. Redes neuronales multicapa de un solo sentido de propagación de señal⁶

Este tipo de redes neuronales se distingue por la presencia de una o más capas intermedias entre la entrada y la capa de salida. A estas capas intermedias a menudo se les conoce también como capas escondidas u ocultas.⁷ La función de las unidades de las capas escondidas es dotar a la red de capacidades de cálculo más poderosas, necesarias sobre todo

6 *Multilayer Feedforward Network*, en inglés

7 *Hidden Layers*, en inglés

cuando se trata de procesar vectores de entrada sumamente complejos o extensos.

En el caso de las redes neuronales multicapa, cada una de las salidas de las unidades de procesamiento en cada capa sirve como entrada para los nodos de la siguiente capa.

Se puede hablar de redes neuronales multicapa completamente interconectadas, cuando las neuronas de cada una de las capas están conectadas con todas y cada una de las neuronas de la siguiente capa (ver ilustración 25). En cambio, cuando algunas de las neuronas de la siguiente capa no son conectadas a un nodo determinado, entonces se habla de redes neuronales parcialmente conectadas (ver ilustración 26). Este tipo de redes se usan comúnmente en casos en los que se busca algún tipo de proceso especial en cada uno de los nodos que integran la red, o cuando se ha atribuido un significado a los nodos de las capas intermedias. Un ejemplo claro de este caso es el de los sistemas expertos neuronales, en los que los nodos de las capas intermedias están relacionados con padecimientos específicos y los nodos de la capa de salida con tratamientos específicos para estos trastornos (ver: Gallant, 1994).

Tanto en el caso de las redes parcialmente interconectadas, como en el de las redes totalmente interconectadas, se habla de que la propagación de la señal se da en un solo sentido. Y al igual que en las redes de una sola capa, el sentido de propagación es *hacia delante*.

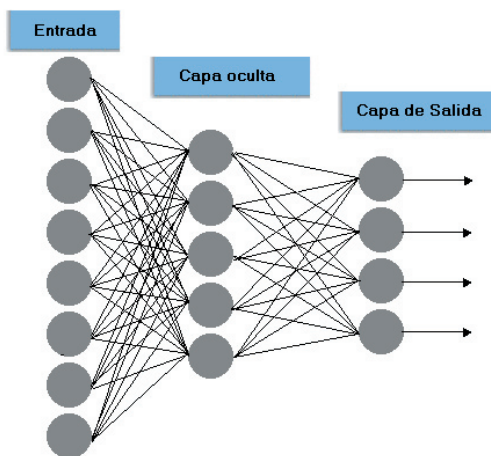


ILUSTRACIÓN 25. Red neuronal totalmente conectada.

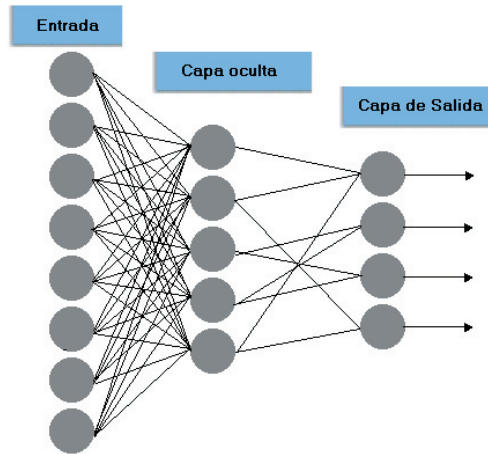


ILUSTRACIÓN 26. Red neuronal parcialmente conectada.

3.2.3.3. Redes neuronales recurrentes⁸

Una red neuronal recurrente se distingue por tener por lo menos una conexión de retroalimentación. Es decir, integra un nodo en alguna de las capas intermedias o de salida que tiene conexión, no solamente con alguna o algunas de las neuronas de la capa siguiente, sino que también integra conexiones con neuronas de capas anteriores (ver ilustración 27). Este tipo de redes se emplea sobre todo, cuando se requiere lograr algún tipo de retardo en el proceso de las entradas de la red, o cuando se quiere mantener algún tipo de información latente que propicie un comportamiento de la red no- lineal.

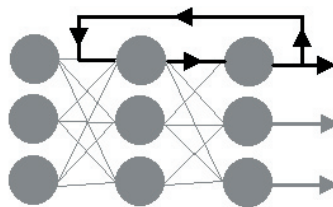


ILUSTRACIÓN 27. Red neuronal recurrente.

⁸ *Recurrent Network*, en inglés

En la ilustración 27 la red neuronal presentada integra una conexión entre un nodo de la capa de salida y un nodo de la capa intermedia

3.2.3.4. Redes neuronales estructuradas en Látices

Un látice consiste en un arreglo, unidimensional, bidimensional o multidimensional de neuronas con un conjunto correspondiente de nodos de entrada que proporcionan las señales de entrada al arreglo. Las dimensiones del látice hacen referencia al número de dimensiones en el espacio en el que el grafo descansa. La ilustración 28 muestra un ejemplo de red neuronal tipo látice unidimensional, y la ilustración 29 muestra uno bidimensional. Nótese que en ambos casos cada nodo de entrada está conectado con cada neurona en el látice. Las estructuras en látices pueden ser vistas como redes neuronales de un solo sentido con las neuronas de salida organizadas en filas y columnas (Haykin, 1994)

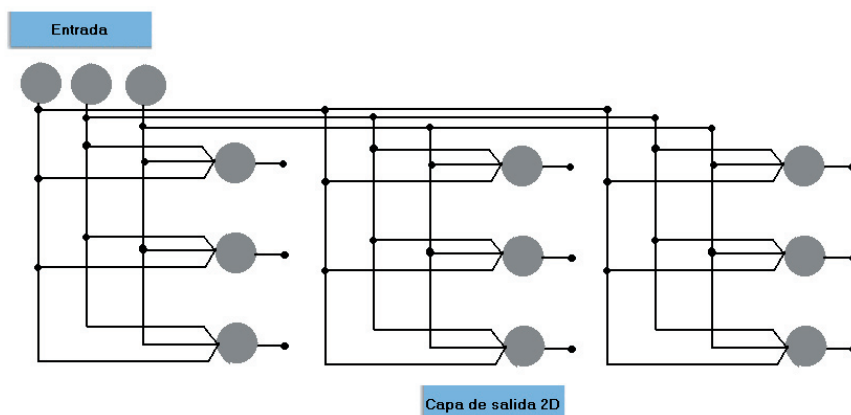


ILUSTRACIÓN 28. Red neuronal tipo látice de una dimensión.

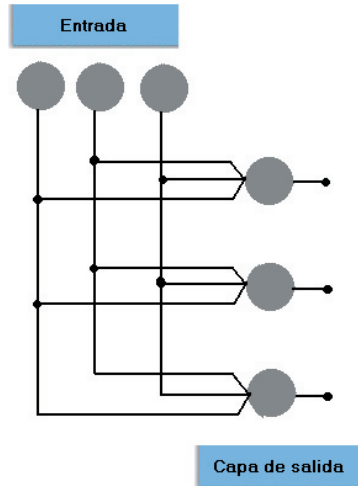


ILUSTRACIÓN 29. Red neuronal tipo láctice de dos dimensiones.

3.3. funcionamiento de una red neuronal

El funcionamiento de una red neuronal se puede dividir en dos grandes procesos:

- Transmisión de los impulsos a través de la estructura interna de la red, y
- Aprendizaje.

Estos dos grandes procesos están estrechamente asociados, respectivamente, con las dos funciones básicas atribuidas a una red neuronal: la identificación o clasificación de patrones de entrada (que se realiza mediante la propagación de los impulsos) y la aproximación de funciones, que es la base del aprendizaje de red.

3.3.1. Transmisión de los impulsos a través de la estructura de la red

Los componentes de una red neuronal interactúan para transmitir o propagar, datos a través de toda la estructura de la red, de la manera siguiente:

- a) Cada una de las unidades de la capa de entrada se asigna a alguna característica o atributo del problema a tratar. Así, se tendrán tantas unidades en la capa de entrada, como características sobresalientes en el problema a tratar. Las neuronas de esta capa de entrada se alimentan con los valores correspondientes de cada característica del problema en un caso específico.
- b) Las entradas se procesan en cada nodo de la red conforme a lo especificado en el apartado 3.2.1. de este mismo documento y se transmite el valor resultante a cada una de las unidades con las cuales, el nodo actual, tiene conexión. Así, los valores resultantes de cada nodo se alimentan al siguiente nodo como entrada para un nuevo procesamiento. La transmisión de las entradas termina cuando los nodos de la capa de salida son alimentados y éstos han ofrecido su resultado al usuario.
- c) La salida de la red representa la solución al problema. En la capa de salida de una red neuronal se tendrán entonces, tantas unidades como soluciones pueda tener el problema y el nodo activado es el que indica cuál es la solución calculada para el problema procesado.

3.3.2. Aprendizaje en una red neuronal

Por aprendizaje en una red neuronal se debe entender al proceso en virtud del cual los pesos de conexión de cada nodo de la red sufren ajustes de manera tal que la capa de salida pueda ofrecer la solución adecuada al problema presentado en la capa de entrada.

Las redes neuronales usan formas muy específicas para realizar este ajuste de pesos de conexión, que van de acuerdo al modelo de red bajo el cual están funcionando. Es posible clasificar los métodos de aprendizaje en dos grandes grupos: el aprendizaje no supervisado y el aprendizaje supervisado.

3.3.2.1. Aprendizaje no supervisado

El aprendizaje no supervisado es otra modalidad usada para el ajuste de pesos en las redes neuronales. En este tipo de aprendizaje a la red no se le proporcionan modelos de respuestas correctas a obtener a partir de ciertas entradas. Simplemente se introducen los ejemplos del conjunto

de casos de entrenamiento en el sistema y se espera que la red neuronal agrupe los ejemplos con base en la cercanía que entre cada uno de ellos existe. El cálculo de cercanía para el agrupamiento posterior se puede realizar de diversas maneras que van desde procedimientos sencillos como los que se basan en la *distancia de Hamming*,⁹ hasta otros más complejos (Ver Skapura, 1995) y (Maravall Gómez-Allende, 1994)

Para que el sistema realice el procedimiento de agrupamiento de ejemplos se necesita la constante modificación de los pesos de conexión entre las neuronas que integran la red que permita posteriormente, asignar correctamente cada uno de los ejemplos al grupo adecuado. En este caso también se espera que la red neuronal ya entrenada sea capaz de trabajar eficientemente con casos similares pero no introducidos previamente.

Entre otros, los modelos que se mencionan a continuación son representativos de la línea de investigación basada en el aprendizaje no supervisado.

- Redes de Hopfield (Hopfield, 1982).
- Mapas auto- asociativos de Kohonen (Kohonen, 1982).
- Los Mapas auto- asociativos bidimensionales de Kohonen (Kaski, 1997).
- Clasificadores de Carpenter/ Grossberg (Lipmann, 1987).
- Redes con base en la Teoría de la Resonancia Adaptativa (ART) (ver: Hertz, Krogh y Palmer, 1991).

3.3.2.2. Aprendizaje supervisado

En el aprendizaje supervisado la adecuación de los pesos de conexión entre cada nodo se realiza evaluando el desempeño general de la red para lograr una salida deseada ante una configuración de patrones de entrada determinados.

En el proceso de aprendizaje supervisado, el entrenamiento de una red consiste en la evaluación de la salida actual que la red neuronal presenta ante una entrada, y su comparación con la salida que la red *debería*

⁹ La distancia de Hamming (D_H) puede ser definida como el total de elementos distintos en valor y posición, entre dos vectores. Por ejemplo el vector $1 = \{1,0,1,1,0\}$ se encuentra a una $D_H = 1$, del vector $2 = \{1,1,1,1,0\}$.

presentar ante esta entrada. La diferencia entre la salida obtenida y la salida deseada se usa para calcular el ajuste que debe darse al interior de la red en cada uno de los pesos de conexión, a fin de que se logre obtener la salida.

A este tipo de aprendizaje se le conoce como aprendizaje supervisado ya que el uso de salidas deseadas ante entradas determinadas permite supervisar el desempeño de la red en cada momento y con ello, la realización de los ajustes de pesos necesarios en la dirección correcta.

Una de las características de las redes neuronales asociadas con este tipo de aprendizaje es la capacidad para generalizar los aprendizajes logrados. Se considera que un proceso de aprendizaje supervisado en una red neuronal ha tenido éxito cuando el ajuste de pesos, realizado a partir de un conjunto de ejemplos, le permite tener un desempeño efectivo ante problemas similares a los ejemplificados, pero nunca antes introducidos en el sistema.

3.3.2.2.1. Retropropagación del error

Quizá el algoritmo de aprendizaje más famoso en redes neuronales sea el que se basa en la retropropagación del error.¹⁰ Este algoritmo fue descrito en el libro de Rumelhart y McClelland (Rumelhart, Hinton y Williams, 1986a) y (Rumelhart, Hinton y Williams, 1986b) aunque parece haber un reconocimiento generalizado de que los creadores del algoritmo de aprendizaje fue propuesto por Werbos (Werbos, 1974) y por Parker (Parker, 1985).

La arquitectura de una red con aprendizaje basado en la retropropagación del error, tiene un diseño jerárquico consistente en un conjunto de capas de unidades de proceso completamente interconectadas. El proceso de la información bajo la retropropagación tiene por objetivo realizar la aproximación de una función $f : A \subset R^n \rightarrow R^m$ a partir de un subconjunto A de un espacio euclidiano n -dimensional a un subconjunto delimitado $f[A]$ de un espacio euclidiano m -dimensional, por medio de un conjunto de ejemplos de entrenamiento $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ de la correspondencia, en donde $y_k = f(x_k)$. Se asume que estos ejemplos de la correspondencia f son generados seleccionando de A , x_k vectores al azar de acuerdo con una función fija de probabilidad $p(x)$.

¹⁰ *Backpropagation*, en inglés.

Actualmente existen muchas variables del algoritmo. A continuación se describe uno de los más usuales y efectivos que implica el uso de una sola capa de unidades ocultas (ver: Freeman y Skapura, 1991), en el entendido de que su extensión a más unidades es sencilla.

1. Se aplica un vector de entrada a la red y se calculan los correspondientes valores de salida.
2. Se comparan las salidas obtenidas con las salidas correctas y se determina una medida del error.
3. Se determina en qué dirección (positiva o negativa) debe cambiar cada peso con objeto de reducir el error.
4. Se determina la cantidad en que es preciso cambiar cada peso
5. Se aplican las conexiones a los pesos
6. Se repiten los pasos del 1 al 5 con todos los patrones de entrenamiento hasta que el error para todos ellos se haya reducido hasta una cantidad aceptable

ALGORITMO 1. Aprendizaje por retropropagación del error (procedimiento general).

Como podrá observarse en el Algoritmo 1 el trabajo con la retropropagación del error implica la realización de dos fases o momentos en la red neuronal:

- Una fase de propagación de la señal, que es precisamente la fase de operación o consulta de la red. En esta fase, gracias a la propagación de la señal desde las unidades de entrada, hasta la capa de salida, la red puede establecer una correspondencia entre vectores de entrada y una salida correspondiente.
- La segunda fase es la de propagación del error hacia atrás o retropropagación. Esta fase que tiene un papel muy importante en el proceso de entrenamiento de la red, ya que gracias a ésta precisamente, es que se pueden ajustar los pesos en la arquitectura interna, para que en la próxima propagación de la señal, la red sea mucho más eficiente. En términos neurocomputacionales, a este ajuste de pesos de le llama también *aproximación de la función*.

Los pasos que tienen lugar en estas dos fases se presentan en el Algoritmo 2 y el Algoritmo 3.

1. Se localiza la primera unidad de procesamiento de la capa que se encuentre inmediatamente por encima de la capa actual.
2. Se pone a cero el total actual de la entrada.
3. Se calcula el producto del primer peso de conexión de entrada, por la salida de la unidad transmisora.
4. Se añade el producto al total acumulado
5. Se repiten los pasos 3 y 4 para todas las conexiones de entrada
6. Se calcula el valor de salida para esta unidad aplicando la función de salida
7. $f(x) = 1(1 + e^{-x})$ en donde x es la entrada total
8. Se repiten los pasos del 2 al 6 para todas las unidades de esta capa
9. Se repiten los pasos del 1 al 7 para todas las capas de la red.

ALGORITMO 2. Propagación de la señal hacia delante.

Una vez que se ha calculado un valor de salida para todas las unidades de la red, se comparan los valores calculados para las unidades de la capa de salida, con la salida que se ha establecido como deseada, elemento por elemento como se muestra en el Algoritmo 3.

Se calcula un valor de error en cada unidad de salida. Estos términos de error son realimentados en todas las demás unidades de la estructura de la red mediante la siguiente secuencia de pasos:

1. Se localiza la primera unidad de proceso que se encuentre inmediatamente por debajo de la capa de salida.
2. Se pone a cero el error total actual.
3. Se calcula el producto del peso de conexión de la primera salida por el error proporcionado por la unidad de la capa superior.
4. Se añade ese producto al error acumulado
5. Se repiten los pasos 3 y 4 para todas las unidades de salida
6. Se multiplica el error acumulado por $o(1 - o)$ en donde o es el valor de salida de la unidad de la capa oculta que se ha producido durante la operación de la propagación de la señal hacia delante
7. Se repiten los pasos del 2 a 6 para todas las unidades de la capa
8. Se repiten los pasos 1 al 7 para todas las capas
9. Se localiza la primera unidad de proceso que esté en a capa superior a la capa de entrada
10. Se calcula el valor de cambio del peso para la primera conexión de

entrada de esta unidad añadiendo una fracción del peso acumulado en esta unidad al valor de entrada de la unidad.

11. Se modifica el término de cambio de peso añadiendo un término de momento, el cual es igual a una fracción del valor del cambio de peso procedente de la iteración anterior.
12. Se guarda el nuevo valor del cambio de peso como valor anterior del cambio de peso para esta conexión.
13. Se modifica el peso de la conexión añadiendo el valor del nuevo cambio de peso de conexión al peso anterior de conexión
14. Se repiten los pasos del 10 al 13 para todas las conexiones de entrada de esta unidad
15. Se repiten los pasos del 10 al 14 para todas las unidades de esta capa
16. Se repiten los pasos del 10 al 15 para todas las capas de la red

ALGORITMO 3. Propagación del error hacia atrás.

Dado que el algoritmo de retropropagación del error es altamente costoso en términos computacionales, se han realizado numerosos esfuerzos para acelerar su convergencia. Sin embargo, estas modificaciones aún no han sido lo suficientemente buenas como para desplazar del lugar exitoso que ha ocupado el algoritmo original desde su propuesta (Freeman y Skapura, 1991).

Otros ejemplos de redes neuronales que funcionan bajo el esquema del aprendizaje supervisado son, entre muchos, los siguientes:

- Perceptrón (Rosenblatt, 1958).
- Contrapropagación (Freeman y Skapura, 1991).
- Máquinas de Boltzmann (Hertz, Krogh y Palmer, 1991).

3.4. redes neuronales e inteligencia artificial

El procesamiento de la información que se logra con las redes neuronales difiere grandemente de las formas tradicionales de proceso en Inteligencia Artificial, es por ello que hay autores consideran a la computación neuronal como una disciplina *aparte* de la Inteligencia Artificial (Turban, 1992). Sin embargo, debido a la coincidencia entre las redes neuronales y la inteligencia artificial en lo referente a algunos de los fines últimos y

muchos de los elementos de trabajo utilizados, se considera a la Neurocomputación como una *disciplina estrechamente asociada* (Haykib, 1994; y Churchland, 1992), o incluso, como una línea de trabajo más en la inteligencia artificial bajo una concepción extendida de la disciplina (Freeman, 1985; Minsky, 1991; y Freeman y Skapura, 1991).¹¹

Las características que permiten establecer diferencias entre las redes neuronales y otras líneas en inteligencia artificial son las siguientes:

- a) La representación del conocimiento. En las redes neuronales el conocimiento es representado por medio de valores numéricos distribuidos en la estructura interna de la red, sin implicar una estructura sintáctica compleja. La configuración completa de estados en un sistema puede representarse por medio de vectores y el trabajo de la red puede verse como una simple relación entre espacios de vectores (uno de entrada y uno de salida). Este tipo de representación ha llevado la reformulación de los problemas como la identificación de patrones o el aprendizaje a niveles matemáticamente tan sencillos como encontrar el vector más adecuado (Minsky, 1991).
- b) El proceso paralelo. El alto grado de interconexión entre las unidades de procesamiento implica un grado elevado de resistencia a errores. Esto se ve reflejado en la robustez del sistema. Igualmente, la posibilidad de contar con muchas unidades de proceso facilita la realización de tareas complejas que mediante un proceso secuencial necesitarían una mayor inversión de recursos.
- c) El proceso distribuido. Cada unidad de proceso realiza tareas muy sencillas, que no van más allá de la suma de entradas y pesos de conexión, y la redefinición del resultado con base en una función de transferencia. La complejidad del proceso radica más en la arquitectura y funcionamiento globales del sistema, que en la estructura y funcionamiento de sus unidades por separado.

11 La concepción original de Inteligencia artificial consideraba solamente al proceso simbólico de la información, por lo que, desde la definición misma, las redes neuronales y otros enfoques de proceso no simbólico como la computación genética, quedan excluidos. Sin embargo, en los autores más recientes se puede identificar una concepción *ampliada* o *extendida* que incluye ya a otras líneas de trabajo no tradicionales.

3.5. Redes neuronales y lógica difusa

En los últimos años se ha buscado mejorar el desempeño de los sistemas inteligentes integrando técnicas y metodologías correspondientes a diversos enfoques teóricos. A esta estrategia de integración se le conoce como *hibridación*, término que le fue conferido por analogía con el proceso de combinación de clases en genética.

Sin embargo, el término debe ser interpretado con precaución, ya que en ciencias computacionales puede usarse para hacer referencia una gran variedad de sistemas que van desde la simple unión de varias aplicaciones en un solo paquete, hasta la verdadera articulación de varios enfoques teóricos en una sola aplicación.

3.5.1. Esquema de clasificación de las redes neuronales difusas

Fullér (Fullér, 1995) ha establecido que los sistemas neuronales se pueden enriquecer con la implementación del manejo difuso de la información en diversas formas, de acuerdo a las necesidades específicas del problema que se quiere resolver con la combinación.

Las maneras posibles de integración de la lógica difusa en los sistemas neuronales han llevado al establecimiento de los siguientes tipos de sistemas neuronales difusos:

TABLA 1
Tipos de integración de la lógica difusa en sistemas neuronales.

Sistema	Entradas	Pesos	Salidas
Tipo 1	<i>Difusas</i>	<i>Nítidos</i>	<i>Nítida</i>
Tipo 2	<i>Difusas</i>	<i>Nítidos</i>	<i>Difusa</i>
Tipo 3	<i>Difusas</i>	<i>Difusos</i>	<i>Difusa</i>
Tipo 4	<i>Nítidas</i>	<i>Difusos</i>	<i>Difusa</i>
Tipo 5	<i>Nítidas</i>	<i>Nítidos</i>	<i>Difusa</i>
Tipo 6	<i>Nítidas</i>	<i>Difusos</i>	<i>Nítida</i>
Tipo 7	<i>Difusas</i>	<i>Difusos</i>	<i>Nítida</i>

En los apartados siguientes se presenta una breve descripción de cada uno de estos siete tipos de redes neuronales difusas, sin embargo, debe entenderse en cada uno de ellos, que los esquemas presentados pretenden solamente, servir como referencia, ya que se incluyen solamente los elementos básicos de la clasificación. Asimismo, en el esquema, con fines

de ilustración solamente, se presentan redes neuronales completamente interconectadas entre capa y capa, sin embargo, las arquitecturas son tan variadas como los modelos a los que atienden.

3.5.1.1. Redes neuronales difusas del tipo 1

Este tipo de redes neuronales difusas son dispositivos muy cercanos a las redes neuronales tradicionales, incluso algunos autores las ubican como la difuminación de redes neuronales tradicionales. Se integran por una red neuronal tradicional extendida para manejar con sus pesos nítidos, una entrada difusa y poder ofrecer una salida nítida.

Este tipo de sistemas es usado, a menudo, en problemas de clasificación de entradas difusas y su asignación a clases nítidas.

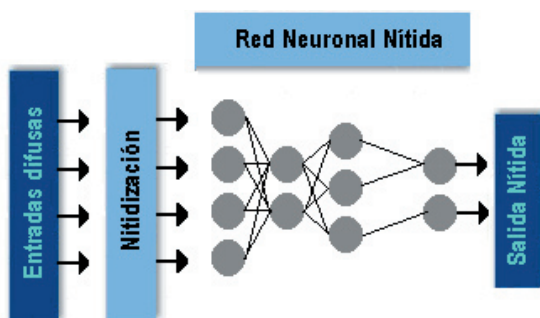


ILUSTRACIÓN 30. Red neuronal difusa del tipo 1.

Redes neuronales difusas de este tipo han sido usadas por el Dr. Hayashi (Hayashi, 1994) en un *sistema experto neuronal difuso* para diagnóstico médico. Este sistema se usa en el Departamento de Medicina Preventiva y Salud Pública, de la Universidad de Keio en Japón, en situaciones de diagnóstico real.

3.5.1.2. Redes neuronales difusas del tipo 2

Las redes neuronales difusas del tipo dos son aquellas redes que integran entradas difusas en una red cuyos pesos de interconexión son nítidos, pero su salida es también difusa. Este tipo de red neuronal difusa puede

ser vista como una red neuronal nítida que emplea dos dispositivos adicionales: uno para nitidizar las entradas difusas, y otro para difuminar las salidas nítidas de la red.

Este tipo de redes neuronales difusas es uno de los primeros implementados. Su utilización ha sido reportada por Gallant y Hayashi a inicios de la década de los noventa (Gallant y Hayashi, 1990), quienes realizaron sistemas especializados en la identificación de patrones difusos para el diagnóstico médico también.

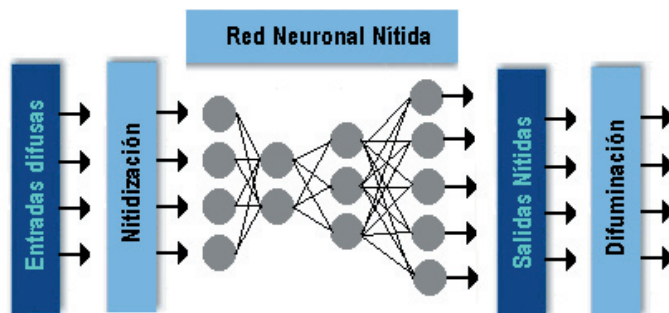


ILUSTRACIÓN 31. Red neuronal difusa del tipo 2.

Asimismo, Hayashi y otros (Hayashi, Imura y Yoshida, 1990b) la emplearon con éxito en el diagnóstico de síndromes inmunológicos especiales.

3.5.1.3. Redes neuronales difusas del tipo 3

La ilustración 32 muestra el esquema general de una red neuronal difusa del tipo 3. En ésta puede apreciarse que este tipo de sistemas emplea una red neuronal difusa para procesar entradas difusas y ofrecer como resultado salidas difusas.

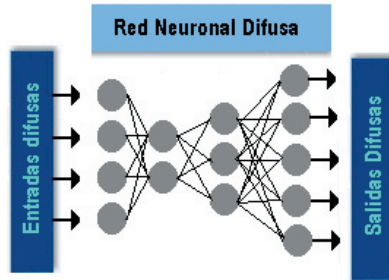


ILUSTRACIÓN 32. Red neuronal difusa del tipo 3.

Este tipo de redes neuronales ha sido uno de los más usados a la fecha. Sus aplicaciones en control pueden revisarse en (Pedrycz, 1993) y (Kosko, 1997), o en el Perceptrón completamente difuso de Goh y otros (Goh, Wang y Lui, 1991).

A este tipo de red neuronal también se le ha llegado a conocer como *red neuronal completamente difusa* (Gupta y Qi, 1992). También este tipo de red se ha usado como componente neuronal básico de sistemas híbridos del alto orden como los sistemas expertos neuronales difusos (Frydenberg y Gallant, 1987)

3.5.1.4. Redes neuronales difusas del tipo 4

En la ilustración 33 se muestra una red neuronal difusa del tipo 4. Este tipo de sistemas integran una red neuronal difusa con un dispositivo de difuminación de entradas nítidas.

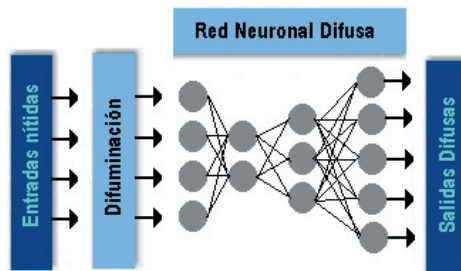


ILUSTRACIÓN 33. Red neuronal difusa del tipo 4.

Un ejemplo del uso de este tipo de redes neuronales difusas puede revisarse en el trabajo de Cristoph Herrmann en el diagnóstico difuso de electrocardiogramas en tiempo real (Herrmann, 1995).

3.5.1.5. Redes neuronales difusas del tipo 5

Las redes neuronales del tipo 5 son aquellas que se componen de una red neuronal típica que modificada con un dispositivo de difuminación, puede ofrecer salidas difusas.

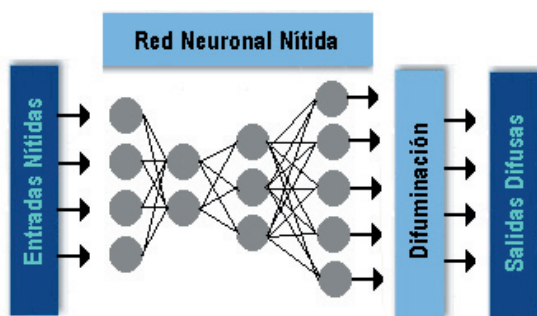


ILUSTRACIÓN 34. Red neuronal difusa del tipo 5.

Ejemplos del uso de este tipo de red neuronal difusa se pueden encontrar descritos en (Frydenberg y Gallant, 1987).

3.5.1.6. Redes neuronales difusas del tipo 6

En la ilustración 35 se muestra una red neuronal del tipo 6. Este tipo de sistema integra una red neuronal difusa completa que se adapta para poder manejar entradas y salidas nítidas. Este tipo de proceso podría parecer ocioso, ya que en principio este tipo de red parece complicar los problemas que se pueden resolver por medio de redes neuronales nítidas, aunque en la práctica no es así. El Dr. Nikola Kasabov y otros (Kasabov, 1998a) y (Kasabov, 1998b) han demostrado que se puede lograr una convergencia de la red neuronal, en menor tiempo que en una red neuronal, si para algunos casos se emplea la lógica difusa para la aproximación de la función de correspondencia.

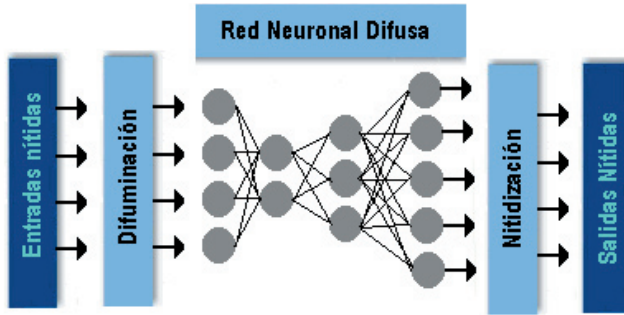


ILUSTRACIÓN 35. Red neuronal difusa del tipo 6.

3.5.1.7. Redes neuronales difusas del tipo 7

Finalmente, la red neuronal difusa del tipo 7, como la que se muestra en la ilustración 36, es un sistema compuesto por una red neuronal completamente difusa modificada para poder ofrecer una salida nítida. Usualmente este tipo de sistemas ha tenido éxito en aplicaciones de control en las que se requiere el proceso de información difusa, pero una salida nítida que pueda guiar una acción específica en un sistema efector.

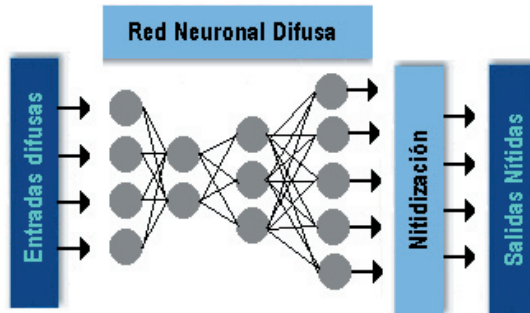


ILUSTRACIÓN 36. Red neuronal difusa del tipo 7.

Aplicaciones de sistemas que emplean este tipo de redes neuronales difusas pueden encontrarse reportadas en (Hayashi, Imura y Yoshida, 1990a) para aplicaciones en control médico, y en (Pedrycz, 1993) para aplicaciones en control industrial.

3.6. las redes neuronales en la actualidad

Actualmente las redes neuronales se han constituido en una de las tecnologías más estudiadas en los últimos años. Día con día se implementan nuevos modelos de aprendizaje o nuevas topologías de organización que se muestran más efectivas en la resolución de problemas ya clásicos e incluso surgen nuevas aplicaciones.

Sin embargo, los problemas persisten, algunos de los cuales son tan viejos como las mismas redes neuronales. En la última década la investigación se ha enfocado a tratar de dar solución a los siguientes problemas asociados con el uso de redes neuronales:

- a) El aprendizaje que una red neuronal realiza, sea supervisado o no supervisado, depende de los casos de ejemplo que se le proporcionen. No solamente de la cantidad de los mismos, sino también de la calidad. Por lo tanto, si no se alimenta al sistema con un conjunto de ejemplos representativos de los estados del problema, se corre el riesgo de excluir a las excepciones que posibilitan que el sistema tenga un desempeño efectivo en problemas reales (Yu, 1995).
- b) Mucho del proceso inductivo que se puede realizar en otros sistemas es imposible de lograr en las redes neuronales, debido a su imposibilidad de asociar el conjunto de estados de entrada con un contexto más amplio (Yu, 1995).
- c) Dado que solamente se puede usar un conjunto finito de características para describir cualquier objeto, la elección de las características que se considerarán como entradas al sistema es determinante en el proceso de identificación o clasificación que realiza. El gran número de posibilidades implica innumerables formas de clasificación de un mismo objeto, que no pueden ser contempladas en la red neuronal. El trabajo con redes neuronales implica asumir un mundo cerrado en el que las características usadas son las únicas y no existe posibilidad de apertura (Yu, 1995).
- d) Aunque las redes neuronales han tenido un éxito creciente como dispositivos para proceso de información en paralelo, la brecha entre éstas y los sistemas biológicos que las inspiraron en sus principios, es cada vez mayor. Las expectativas iniciales de que estos sistemas ayudaran a explicar los fenómenos biológicos en torno al funcionamiento del cerebro humano cada vez están más lejos de lograrse (Dawson y Shamanski, 1994).

- e) Las redes neuronales pueden ser consideradas como *sistemas de caja negra*, es decir, sistemas en los que sólo son conocidas con exactitud las entradas y las salidas. Las redes neuronales pueden tener cientos o miles de unidades ocultas que manejan parámetros con números reales, que representan relaciones entre la entrada y la salida que no son lineales ni monotónicas. Este hecho implica la imposibilidad de justificar o explicar cada uno de sus resultados, dado que las decisiones tomadas se encuentran distribuidas en toda la arquitectura de la red (Hinton, 1986; Craven y Shavlik, 1995; Yu, 1995; y Alexander y Mozer, 1996).

3.7. nuevas líneas de trabajo en redes neuronales

Actualmente la investigación en redes neuronales se ha centrado en los siguientes aspectos principalmente:

- a) El desarrollo de nuevos modelos de aprendizaje, entrenamiento y de interconexión entre redes neuronales de varios tipos; el diseño de nuevas arquitecturas de red y el diseño de circuitos electrónicos neuronales.
- b) El desarrollo de aplicaciones en nuevos campos del conocimiento, entre los que se pueden encontrar: detección de fallas por medios acústicos, espectroscopia neuronal, diagnóstico médico por imagenología, soporte a toma de decisiones, visión artificial especializada, etc.
- c) El desarrollo de sistemas híbridos que combinen el trabajo en redes neuronales con otro tipo de sistemas como los sistemas difusos, los sistemas expertos, las bases de datos, los tutores inteligentes, la computación evolutiva, la realidad virtual, etc.

4

Extracción de reglas en redes neuronales nítidas y redes neuronales difusas

4.1 Introducción

La habilidad para aprender a partir de ejemplos ha sido considerada desde siempre como una actividad inteligente superior. La inducción es la base sobre la que descansan el resto de las actividades superiores en el ser humano. Gracias a este proceso cognoscitivo es se puede llegar a dar el paso desde lo conocido hacia nuevos conocimientos, hecho que adquiere gran importancia si consideramos que, sobre todo, durante los primeros diez años de vida de un sujeto, la actividad cognoscitiva predominante es el descubrimiento (Piaget, 1985).

Para la inteligencia artificial y más específicamente en el área del *aprendizaje de máquina*, el aprendizaje por inducción ha sido un tema especialmente estudiado en los últimos años, dada la importancia que se ha reconocido a las formas automatizadas de descubrir conocimientos nuevos. Sin embargo, a pesar de que muchos logros se han alcanzado en este campo, existen problemas que no se han podido resolver, sobre todo en los sistemas de inducción automática subsimbólicos como las redes neuronales, en los que la extracción de los conocimientos aprendido es aún el talón de Aquiles.

4.1.1. Formas de uso de las reglas en los sistemas neuronales

En las redes neuronales, tanto tradicionales como difusas, se hace necesario el uso de reglas por las siguientes razones:

- Las reglas facilitan el proceso de adquisición de conocimientos, refinamiento de los mismos y la configuración de otros sistemas con base en el trabajo de un sistema neuronal,

- Las reglas también posibilitan un medio para que las redes neuronales puedan justificar las inferencias realizadas, y finalmente,
- Las reglas permiten a las redes neuronales una forma de representación de los conocimientos que facilitaría el establecimiento de bases de conocimientos completas a partir de lo que el sistema ha aprendido. Es decir, el uso de reglas permite conocer por completo lo que el sistema ha aprendido.

Emplear reglas en las redes neuronales tiene que ver entonces, i) con la simplificación del proceso de adquisición e introducción de los conocimientos para el entrenamiento de una red neuronal, y auxilio en la configuración de un sistema, independientemente de su arquitectura o enfoque, ii) con la mejora de la interacción hombre- máquina, ya que ofrece razones que ayudan al ser humano en el proceso de toma de decisiones a partir de los resultados ofrecidos por el sistema, y iii) en la estructuración y descubrimiento de conocimientos nuevos, ya que la extracción de reglas posibilita también a la identificación de relaciones hasta antes desconocidas o menospreciadas.

4.1.2.1. Uso de reglas para la adquisición de conocimientos y configuración de sistemas

En la adquisición de conocimientos, las reglas tienen una especial importancia, sobre todo, cuando la fuente de los conocimientos es el mismo ser humano. Es gracias al proceso de ingeniería de conocimiento con este fin, que pueden quedar claros muchos de los conocimientos de manera informal, los expertos usan en la resolución de problemas especializados. Además también, el uso de reglas permite conocer qué aspectos o factores es importante considerar y por el otro lado, qué elementos resultan irrelevantes para los fines que se persiguen.

Es también pertinente destacar el hecho de que, es solamente hasta que se han convertido los conocimientos de los expertos a una representación formal, que se puede establecer si existen inconsistencias o contradicciones inconscientes, pero reales, en éste; y por lo tanto, se posibilita el diseño de métodos y estrategias de articulación o de resolución de este tipo de conflictos, diseño que se hace necesario sobre todo, por ejemplo, cuando se intenta compaginar el conocimiento de varios expertos. Finalmente, en el caso de muchas de las arquitecturas empleadas en nuevos

sistemas, la misma estructuración de las reglas es la que establece en algunos casos la arquitectura de la red que las integra.

4.1.2.2. Uso de reglas para la justificación de las inferencias realizadas

Una característica definitoria de los sistemas expertos es su capacidad para justificar las conclusiones a las que han llegado. Aunque muchos sistemas neuronales son capaces de llegar a una toma de decisiones efectiva, se hace necesario que la puedan justificar.¹ Si esta justificación de una inferencia se realiza a partir del uso de reglas, se facilita a los expertos humanos la revisión y refinado de los conocimientos manejados por el sistema neuronal. Este hecho permite asimismo, mantener bases de conocimientos confiables y válidas.

4.1.2.3. El uso de reglas para el establecimiento de bases de conocimientos

También se pueden usar las reglas, para que en conjunto, se logre estructurar bases de datos completas para dominios específicos, a fin de obtener conocimientos nuevos y también con el objetivo de permitir la depuración de los conocimientos que los sistemas van a usar posteriormente. Estas bases de conocimientos son en realidad un conjunto de reglas que se aproxima a la colección de reglas del tipo *Si- Entonces*² que están implícitamente codificadas en la red neuronal del sistema experto.

La producción de conjuntos de reglas es un proceso mucho más complejo que la justificación de las inferencias realizadas en las redes neuronales. Cuando se intenta generar conjuntos de reglas para explicar el conocimiento en una red neuronal se tiene que hacer frente a la posibilidad de una explosión combinatoria del número de reglas que se pueden extraer.

Es por estas razones que en el caso de sistemas expertos neuronales que tienen como base redes neuronales extensas, se hace necesario el

1 Considerése, por ejemplo, a un sistema que asigna límites de crédito a clientes en un banco, o un sistema que diagnostica enfermedades cerebro- vasculares a partir de ecosonogramas o electrocardiogramas. En ambos casos la explicación de cómo se ha resuelto el problema se convierte en parte de la tarea encomendada al sistema.

2 *If-Then*, en inglés.

diseño de estrategias que recorten o limiten esta explosión, mediante el establecimiento de algunas restricciones que impidan tener que generar y manejar todas las reglas posibles.

4.1.2. La extracción de reglas en redes neuronales

Como se ha mencionado párrafos antes, un acercamiento para comprender las hipótesis implícitas en un sistema de aprendizaje por inducción, es la búsqueda de una adecuada representación de las mismas. En una red neuronal entrenada se busca la traslación de estas hipótesis a un lenguaje más comprensivo y más cercano a la forma en que por excelencia se ha representado el conocimiento: *las reglas*. Varios enfoques que buscan lograr este tipo de representación a partir de la configuración de una red neuronal entrenada se conocen bajo el nombre genérico de *extracción de reglas*, aunque el mismo nombre genérico es motivo ya de polémica para los diversos autores: Gallant (Gallant, 1994), Omlin y Giles (Omlin y Giles, 1996), Taha y Gosh (Taha y Gosh, 1996a) y (Taha y Gosh, 1996b), por ejemplo, utilizan el nombre de *extracción de reglas* basado en el supuesto de que las reglas estén de alguna manera *ya contenidas* en la misma configuración de la red neuronal. Bart Kosko (Kosko, 1992) y Tom Ojalá (Ojalá, 1995) sugieren que más bien se trata de una *generación de reglas* a partir de la configuración ya establecida en una red neuronal. Su postura se basa en el hecho, también cierto, de que es la configuración final de una red neuronal la que da la pauta para la generación y que el producto obtenido es uno de los muchos posibles. Ellos argumentan que hablar de extracción de reglas, en cambio, implicaría la existencia de conjuntos únicos. En el presente documento se usa el término *extracción de reglas* para hacer referencia a este campo, dada la difusión que esta denominación ha tenido aún más allá de la postura teórica que le subyace. Asimismo, en un afán de reconocer la razón que en cierto grado ambas posturas tienen, se ha respetado el término que cada autor emplea cuando se hace referencia a su trabajo en este documento.

Actualmente la *extracción de reglas* es considerada como un campo dentro del estudio de las redes neuronales, los sistemas híbridos y la inducción o aprendizaje de maquina. Su reconocimiento es reflejo de la gran cantidad de esfuerzos, que en todo el mundo, se han abocado a la interpretación de las redes neuronales por medio de reglas de inferencia.

Para los propósitos del presente trabajo, la extracción de reglas puede entenderse como la descripción de las hipótesis contenidas implícitamente en la configuración de una red neuronal, buscando aproximarse lo más cercanamente posible al comportamiento predictivo de la red neuronal (Craven y Shavlik, 1994).

El trabajo realizado hasta ahora por los grupos de investigación en este campo ha llevado al establecimiento de las características generales que cualquier algoritmo de extracción de reglas debería buscar integrar:

- *Comprensibilidad.* Un algoritmo de extracción de reglas debe ser capaz de establecer representaciones simbólicas que sean accesibles a los seres humanos en lo que respecta a comprensión y manejo. Originalmente la facilidad de manejo no era una característica considerada como un aspecto importante de la comprensibilidad, sin embargo, se ha visto que las reglas se vuelven incomprensibles cuando son estructuradas con un número grande de premisas. Igualmente vuelve poco comprensible a una regla un número extenso de elementos en las conclusiones o incluso, el exceso de conectivos lógicos tanto en las premisas como en la conclusión (Craven y Shavlik, 1995).
- *Fidelidad.* Las representaciones simbólicas producidas deben modelar en forma fidedigna al comportamiento de la red neuronal de la cual fueron extraídas. Incluso, se ha llegado a mencionar que el objetivo final debería ser la búsqueda de la equivalencia de sistemas (McMillan, Mozer y Smolensky, 1993).
- *Escalabilidad.* Se debe buscar que el método sea escalable a redes con mayores espacios de entrada y números grandes de unidades en la red así como gran número de pesos de conexión. Aunque no se descarta el hecho de que algunos algoritmos atiendan a algún tipo específico de arquitectura de sistema diseñado para alguna aplicación específica, en los que se podría dar el caso de que la búsqueda de escalabilidad no tenga sentido (Craven y Shavlik, 1995).
- *Generalidad.* Se ha llegado a establecer el ideal de tener métodos con un grado alto de generalidad que lleve a tener también independencia del método de entrenamiento del sistema, esto a fin de lograr tener el menor número de restricciones posible. Específicamente el ideal establecido tiende hacia el diseño de métodos aplicables en redes con distintas topologías y/o funciones de activación (Craven y Shavlik, 1995).
- *Modificabilidad de las reglas,* que es la capacidad del algoritmo para permitir la modificación de las reglas extraídas a partir de cambios

subsecuentes en el dominio del conocimiento o en la estructura de la red que los soporta (Taha y Gosh, 1996a y 1996b).

Estos son los criterios que, dados como características ideales, se pueden usar para evaluar cualquier algoritmo de extracción de reglas, con el fin de tener un conocimiento exacto de sus potencialidades y limitaciones frente a problemas específicos. Además, el uso de estos parámetros de evaluación posibilita tener una visión exacta acerca de los aportes que el algoritmo puede hacer al conocimiento general acerca de la extracción de reglas.

Los sistemas de inducción automática que actualmente se usan para el manejo de conocimientos expertos y cuya base de representación de los mismos se basa en el uso de reglas, han llevado a una redefinición de las reglas que estos sistemas deben manejar (Gallant, 1994). No basta ahora con la conformación de proposiciones en las que se establezcan claramente antecedentes, compuestos por una o varias condiciones, unidas por conectivos lógicos, y en el otro extremo, un consecuente, cuya certeza dependería del cumplimiento de sus antecedentes correspondientes. En los nuevos sistemas diseñados para el manejo de conocimientos, se hace necesario que las reglas cumplan satisfactoriamente con tres criterios básicos que deben integrarse como parte de su misma naturaleza: *validez, máxima generalidad y número viable de condiciones*.

- *Validez*. Una regla, producto natural de un proceso de inducción, debe ser estructurada de tal manera que se posibilite su vigencia a pesar de la variabilidad del resto del contexto que no está contemplado en la formulación de la misma. Es decir, la validez de una regla implica que los enunciados que la componen estén, en cierto grado, aislados del resto de un contexto y por lo tanto, que lo que pueda ocurrir en éste no afecte a la regla.

Si usamos como ejemplo un sistema experto neuronal que tenga un conjunto E de unidades en la capa de entrada, y un conjunto S de unidades en la capa de salida. Una regla que relacione a sólo una parte de las entradas E' (donde $E' \subset E$) con una o algunas neuronas S' (donde $S' \subseteq S$) en la capa de salida, debe mantenerse vigente, independientemente del valor que puedan tomar el resto de las entradas no contempladas en la regla.

- *Máxima generalidad*. En el proceso de inducción que antecede a la generación de una regla, se tiene implícito un proceso de generaliza-

ción de la información, a costa de la pérdida de las posibles particularidades de cada caso. Con la generalización se gana la posibilidad de tener una cobertura más amplia en el espacio de casos que un problema pueda tener, sin embargo, esta cobertura es menos aguda, dado que se tienen que eliminar muchos aspectos puntuales a fin de llegar a obtener esta generalidad.

Se habla de que se ha logrado ya una *máxima generalidad* en una regla cuando las condiciones que la integran se han vuelto absolutamente necesarias, y quitar alguna de estas condiciones tendría como resultado la invalidación de la regla.

En el caso de las redes neuronales expertas, donde se busca establecer la relación entre parte de las entradas E' (donde $E' \subset E$) con neuronas en la capa de salida S' (donde $S' \subseteq S$), debe procurarse que la relación incluya al número mínimo de entradas que permita conservar aún la salida deseada.

Un problema estrechamente ligado a la búsqueda de la *máxima generalidad* es el peligro de llegar a una *sobregeneralización*, que consiste en la eliminación de tantas condiciones en la regla que finalmente ésta se vuelva inútil o trivial aunque permanezca siendo válida. Es el análisis de las reglas obtenidas, en cada caso particular, el que permite detectar cuando se ha llegado a sobregeneralizar una regla. Si este caso se presenta, se hace necesaria una redefinición de las condiciones o antecedentes que se han considerado para caracterizar el problema y a partir de las cuales se habrán de conformar las reglas.

- *Número viable de condiciones.* Las reglas deben integrar un número de condiciones limitado, de manera tal que sea viable su manejo, revisión y comprensión. Sin embargo, no existen un consenso o alguna forma establecida de poder saber cuándo se habla de un número adecuado de condiciones. Hasta la fecha, el establecimiento del número máximo de condiciones permitidas en las reglas de un sistema, es un trabajo artesanal y que depende de los criterios personales de quien controla o diseña el sistema (Gallant, 1994) y (Taha y Gosh, 1996).

4.1.3. Extracción de reglas en redes neuronales difusas

Paralelo a la extracción de reglas en redes neuronales artificiales, se ha trabajado en el campo de la extracción de reglas en redes neuronales difusas. Sin embargo, por razones técnicas que atañen a la misma estructura de los modelos de extracción diseñados, el proceso fue considerado en sus inicios como un *refinamiento* solamente, ya que necesariamente se tenía que incluir un conjunto de reglas iniciales, el cual era depurado y modificado. Hasta antes de la propuesta conocida como FUNEGEN, que posteriormente se explica en este mismo apartado, no era posible partir sin este conjunto de reglas preestablecidas.

Para poder realizar una extracción de reglas en redes neuronales difusas es necesario un proceso que implica tres fases de trabajo:

- a) El primero incluye un conjunto de mecanismos o procedimientos para insertar el conocimiento experto existente en forma difusa, al interior de una red neuronal artificial - esta fase a menudo se conoce como *fase de inicialización* (Fullér, 1995)-. En esta fase también tiene lugar la generación de las representaciones de las correspondientes funciones de membresía.
- b) La segunda fase en este proceso es el entrenamiento de la red neuronal. En este caso el entrenamiento se centra específicamente en la afinación de las funciones de membresía, que se realiza de acuerdo a los patrones proporcionados en el conjunto de datos de entrenamiento.
- c) El tercer elemento es el análisis y la extracción del conocimiento refinado que está incluido en forma de un conjunto de funciones de membresía.

El proceso de extracción de reglas difusas no solamente depende del conjunto de casos de entrenamiento que se usan para la configuración del sistema, sino que también se ve afectado por el establecimiento de las funciones de membresía.³

3 Para más detalle sobre el establecimiento de funciones de membresía remítase a la parte final del segundo capítulo de este mismo documento.

4.2. métodos contemporáneos para la extracción de reglas

En esta sección se revisan algunos de los métodos para extracción de reglas contemporáneos más sobresalientes a fin de ofrecer un marco de referencia a la presentación, que en capítulos posteriores se hace, del método propuesto para extracción de reglas difusas.

A la fecha se ha trabajado en métodos para la extracción de reglas neuronales en dos líneas principalmente (Gallant, 1994; Taha y Gosh, 1996a y 1996b): i) la primera, basada en las neuronas directamente conectadas a la salida en cuestión, también denominada por Andrews (Andrews, Diederich y Tickle, 1995) en su estudio sobre los métodos de extracción de reglas más eficientes, como *métodos locales*, y ii) la segunda línea, basada solamente en la relación entre entradas y salidas a la red (Saito y Nakano, 1990), sin contemplar las posibles conexiones en las capas intermedias. Esta segunda línea es conocida también como la línea de los *métodos globales* o *métodos basados en la descomposición* (Andrews, Diederich y Tickle, 1995).

Sin embargo, ambos enfoques se basan en principios similares que permiten un abordaje genérico de los mismos. De hecho cada modelo de extracción nuevo se basa en la modificación o mejora de alguno de los modelos existentes, implicando variaciones en alguno de estos dos enfoques o de la combinación de ambos (Gallant, 1994).

4.2.1. extracción de reglas desde un enfoque global

Los algoritmos basados en este enfoque se caracterizan por ser independientes de la estructura interna de la red ya que sólo interesan las entradas y las salidas. Este tipo de algoritmos es conocido como *algoritmos de caja negra* (Taha y Gosh, 1996a) y (Taha y Gosh, 1996b), o como *métodos globales* o *pedagógicos* (Andrews, Diederich y Tickle, 1995).

Este tipo de métodos tiene una aplicabilidad amplia y su gran difusión actual se debe a que reúnen, entre algunos otros, dos de los requisitos más importantes considerados como ideales para la extracción de reglas: generalidad y escalabilidad.⁴

4 Ver: párrafos anteriores el apartado referente a las reglas y las generalidades de la extracción de reglas.

Cabe destacar también que, dado el diseño de algunos de los algoritmos que se pueden ubicar en esta línea, también pueden ser usados para la extracción de reglas bajo un enfoque local, es decir, con un enfoque de extracción por descomposición.

4.2.1.1. El Modelo de remoción

Un método en esta línea de los *algoritmos de caja negra* propuesto por Gallant (Gallant, 1994) para generar reglas máximamente generales y válidas, es el algoritmo de *remoción* (ver algoritmo 4).

1. VARIABLES_USADAS = {Todas las variables que pueden aparecer como condiciones en una regla, sean ya conocidas, o no disponibles (se descartan las desconocidas temporalmente). No hay variables examinadas inicialmente por lo que EXAMINADAS = {vacío}.
2. Si todas $j \in$ VARIABLES_USADAS han sido EXAMINADAS, entonces se forma una regla usando las variables contenidas en VARIABLES_USADAS y se llega a FIN DEL PROGRAMA
3. Recoger todas las $j \in$ VARIABLES_USADAS que no han sido EXAMINADAS.
4. Marcar j como EXAMINADAS, y si la regla formada por VARIABLES_USADAS - $\{j\}$ es válida (Esto usando algún método de validación de reglas) entonces se remueve j de VARIABLES_USADAS.
5. Ir al paso 2.

ALGORITMO 4. Algoritmo de remoción.

En el algoritmo de remoción simplemente se examina cada variable en cualquier orden y se checa si es posible removerla teniendo aún una inferencia válida. Nótese que en el paso 4 se determina la validez de las inferencias usando algún método de validación⁵ para prevenir el análisis de todos los posibles valores de las variables desconocidas. Esto significa que si el método de remoción produce reglas que implican a variables de entrada en redes de varias capas, entonces puede producir también una regla que sea válida y aplicable, pero esta regla no sería máximamente general.

⁵ El autor sugiere usar el método de validación de reglas que se implementa en el Modelo de Extracción MACIE, que se revisa páginas más adelante en este mismo documento, sin embargo, la aplicación no se restringe al uso de este método.

En este modelo se pueden usar células que estén directamente conectadas o células de la capa de entrada y la de salida, aunque su autor lo considera básicamente como un enfoque Global.

4.2.1.2. El Modelo BIORE⁶

Propuesto por Taha y Gosh (Taha y Gosh, 1996a), sus características permiten ubicarlo dentro de la clasificación de *algoritmos de caja negra* o como *método global*, ha demostrado amplias posibilidades de aplicación a pesar de su sencillez.

El nombre de este algoritmo se debe a que se extraen reglas binarias a partir de la configuración de cualquier red neuronal entrenada con entradas binarias, basándose la correspondencia entre entradas y salidas que la red efectúa.

Este algoritmo plantea la necesidad de contar con entradas binarias en la red, por lo que se requiere, en el caso de redes no binarias, que las entradas sean convertidas a binario, de la siguiente forma:

$$y_i = \begin{cases} 1 & \text{si } x_i \geq \mu_i \\ 0 & \text{de cualquier otra manera} \end{cases} \quad (4,1)$$

En donde y_i es la entrada x_i convertida a binario, x_i es el valor original de entrada, y μ_i es la media aritmética del conjunto de entradas X_i .

Después de que las entradas han sido convertidas a binario, se aplica el procedimiento presentado en el algoritmo 5.

Para una red neuronal ya entrenada:

1. Obtenga la salida de la red $O(Y) = \{o_j(Y) \mid o_j \in [0,1]\}$ correspondiente a cada patrón de entrada binaria Y . Si el número de nodos de entrada es n , entonces se requieren n^2 patrones de entrada como máximo, aunque se pueden reducir, eliminando aquellos patrones que no pueden ocurrir en la vida real.
2. Generar una tabla de verdad concatenando cada entrada Y del paso 1 y su correspondiente salida $O(Y)$ en el sistema. Las salidas $o_j(Y)$

⁶ *Binarized Input-Output Rule Extraction* en inglés (Extracción de reglas de entrada/salida convertidas a binario).

se igualan a 1 si el nodo correspondiente es activado (es decir, si se excede el umbral establecido), o se iguala a 0 si esto no sucede.

3. Generar la función booleana a partir de la tabla de verdad del paso 2. El formato para la representación de las reglas binarias es el siguiente: *Si [No] Variable de entrada [y [No] Variable de entrada]* → Consecuencia_i*. En donde [·] es un término opcional y ·]* significa que el término [·] puede ser repetido 0 o n veces.
4. Las reglas obtenidas se reescriben de la siguiente manera:
 - Si la regla producida contiene un valor de entrada binario Y_1 verdadero, éste se reescribe como $X_1 > \mu_1$.
 - Si la regla producida contiene un valor de entrada binario Y_1 falso, éste se reescribe como $X_1 \leq \mu_1$.

ALGORITMO 5. Algoritmo BIORE.

A continuación se presentan algunas de las características sobresalientes de este algoritmo (Taha y Gosh, 1996b):

1. No se requiere de ningún tipo de información acerca de la estructura interna del sistema, solamente se requieren los patrones de entrada y las correspondientes salidas. Y tampoco se requiere de algún número específico de capas intermedias en la red.
2. No se requiere algún tipo específico de enfoque de entrenamiento del sistema.
3. Ya que las reglas extraídas son óptimas, no se hace necesario ningún proceso de reescritura o redefinición de las reglas.
4. Las reglas se representan en una forma comprensible y entendible.

Finalmente, existe otra característica que podría afectar a la comprensión de las reglas y en ciertos casos podría ser vista como una desventaja: el número de premisas incluidas en cada regla. Sin embargo, más que depender directamente del algoritmo, esta característica es derivada de la naturaleza específica del conocimiento manejado. Para este algoritmo el número de premisas en cada regla es igual al número de nodos de entrada en el sistema.

4.2.2. Extracción de reglas desde un enfoque local

Las reglas que se extraen por los métodos de extracción locales, se estructuran a partir de las conexiones inmediatas entre neuronas de dos capas contiguas en el sistema. Desde este enfoque, el proceso de extracción de reglas es un proceso de acumulación de dependencias entre el nodo actual y sus antecesores.

La metodología que subyace a este tipo de métodos de extracción de reglas descansa en dos pasos principales (Taha y Gosh, 1996b):

- a) Primero, para cada nodo de la capa intermedia o de salida seleccionado, se buscan diferentes combinaciones de ligas, cuyos pesos de conexión sobre pasen el umbral del nodo y por lo cual, provoquen su disparo, y
- b) Segundo, se debe formar una regla compuesta precisamente por un conjunto de premisas representadas por cada una de estas combinaciones y una conclusión integrada por el disparo del nodo en cuestión.

Si se trata de un nodo de alguna capa intermedia, sus resultados pasarían a integrar, junto con los de algunos otros nodos de la misma capa, los antecedentes para el establecimiento de una regla para algún nodo de la siguiente capa. El procedimiento sería el mismo hasta llegar a establecer los antecedentes para un nodo de la capa de salida.

A pesar de que este enfoque es ampliamente usado, los métodos generados a partir de éste presentan aún problemas como los siguientes:

- a) Su complejidad es del orden de $O(2^l)$ para cada nodo con una amplitud de entradas = l , asumiendo que las entradas sean binarias.
- b) Se llegan a extraer grandes grupos de reglas, de hasta $\beta_p * (1 + \beta_n)$ donde β_p y β_n son los subconjuntos de ligas positivas y ligas negativas respectivamente.
- c) Algunas de las reglas generadas son repetitivas.
- d) Los algoritmos implican el conocimiento profundo de la red y su estructura, lo que a veces es imposible.
- e) Funcionan en redes de tamaño reducido, con una sola o pocas capas intermedias y con pocos valores discretos de entrada. En caso contrario las reglas obtenidas se vuelven muy complejas.

Sin embargo, también se ha reportado que en algunos dominios este enfoque es altamente eficiente y sobre todo, muy fácil de comprender y aplicar.

A continuación se explican algunos de los modelos más eficientes que funcionan bajo este enfoque.

4.2.2.1. El Modelo MACIE

Este modelo propuesto por Gallant (Gallant, 1994), (Gallant, 1990) y (Gallant, 1994) para el sistema de diagnóstico MACIE⁷ (de ahí se deriva el nombre dado al modelo), es uno de los ejemplos más difundidos dentro de la línea de extracción por métodos locales.

El sistema MACIE fue propuesto por S. Gallant a mediados de la década de los ochenta. El objetivo central al que fue dirigido este sistema experto neuronal fue el diagnóstico de ciertas enfermedades en seres humanos, así como la sugerencia de tratamientos para las mismas.

Independientemente del éxito que el sistema mismo representó para sentar las bases de los sistemas expertos neuronales, el algoritmo de extracción de reglas que en éste se implementó, marcó también la pauta para el estudio de la extracción de reglas de redes neuronales entrenadas.

Para explicar en forma general cómo se da la extracción de reglas en el modelo MACIE, se usará como ejemplo la red neuronal presentada en la ilustración 37. En este caso, una regla extraída implicaría el establecimiento de una relación entre cualquiera o todas las unidades u_1 a la u_6 ⁸ como premisas de la regla y la unidad u_7 de salida como conclusión para esta misma regla.

7 *Matriz Controlled Inference Engine*, en inglés (Motor de inferencias controlado por matrices).

8 La ilustración podría corresponder a una red de dos capas solamente, o a una parte de una red con tres o más capas en las que las unidades u_1 a la u_6 podrían ser elementos de una capa intermedia.

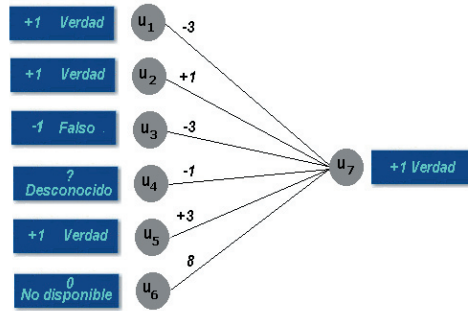


ILUSTRACIÓN 37. Red neuronal con algunos valores de entrada no determinados y valores desconocidos.⁹

El procedimiento general es el siguiente, primero se tiene que definir una *variable de contribución*, que en un sentido estricto es aquella variable que no mueve la suma de los pesos en una dirección errónea. Si se está tratando de explicar una inferencia C^i para una célula u_i entonces se debe considerar que la célula u_i solo contribuye si $C^i w_{i,j} u_j \geq 0$. En el caso de la ilustración 37, las variables contribuyentes son: $\{u_2, u_3, u_5, u_6\}$. También se define un *valor de contribución*, definido por $|w_{i,j}|$. Así, u_3 contribuye con 3. Ahora bien, la idea básica del algoritmo es añadir cláusulas a una regla que contribuyan a una inferencia de acuerdo al tamaño de contribución, hasta que haya suficientes cláusulas para conformar una regla válida.

El proceso completo del modelo se presenta en el algoritmo 6.

Dada una Célula u_i inferida con un valor $c_i = \pm 1$.

1. $ACTUAL = W_{i,0}$ (el umbral para u_i),

$VARIABLES_NO_USADAS = \{j | w_{i,j} \neq 0 \text{ y } j \neq 0\}$ conjunto de células (no se incluye el umbral), conectadas a u_i que no son usadas en la regla

$DESCONOCIDAS = \sum_{j \in VARIABLES_NO_USADAS} |w_{i,j}| =$ máximo valor que $ACTUAL$ puede obtener.

⁹ Nótese que para la unidad u_4 el valor, aunque es desconocido por el momento, podría ser determinado en el futuro, pero el valor para u_6 no está disponible para la corrida entera. Se tiene aún suficiente información, sin embargo, para concluir que u_7 es cierta. Lo que faltaría entonces, es la generación de una regla que justificara la inferencia realizada.

2. Si $ACTUAL > DESCONOCIDAS$ Entonces se detiene la ejecución, las cláusulas generadas son ya actualmente una justificación válida que tiene (usualmente) generalidad máxima.
3. Encuentre una entrada $i \in VARIABLES_NO_USADAS$ tal que: $C^i w_{i,j} u_j \geq 0$ y $|w_{i,j}|$ y $|w_{i,j}|$ maximizado (las ligas se rompen arbitrariamente)
4. De salida a una condición de regla usando la célula u_j y el valor de ésta (que podría ser también “no disponible”).
5. $ACTUAL = ACTUAL + w_{i,j} u_j$
 $DESCONOCIDAS = DESCONOCIDAS - |w_{i,j}|$
 $VARIABLES_NO_USADAS = VARIABLES_NO_USADAS - \{j\}$
6. Ir al paso 2.

ALGORITMO 6. Justificación de inferencias en el sistema MACIE.

Al examinar el proceso presentado en el algoritmo 6 se deben tener en cuenta las siguientes consideraciones:

- a) Nótese que si el umbral es lo suficientemente grande, entonces es posible tener una célula que sea siempre verdadera (o falsa), sin importar las entradas. Así, en la ilustración 4, la célula asumiría una activación constante de +1 (verdad) si el umbral fuese de +200 en lugar de 2 (siendo los demás pesos iguales). En este caso la justificación para la inferencia sería “ u_7 es siempre verdadera”.
- b) La prueba en el paso 3 pretende encontrar variables de contribución con un tamaño máximo de contribución. Estas variables pueden ser variables *conocidas* o *no disponibles*, pero no *temporalmente desconocidas*. Así u_1 añade una cantidad negativa a la suma de los pesos que es finalmente positiva, de manera que no aparecerá en una regla máxima general.
- c) También se debe notar que es posible que las entradas no disponibles participen en las reglas, como ocurre en el ejemplo expuesto aquí.
- d) Así, la regla producida sería:

Si: $u_6 = DESCONOCIDA$ y $u_3 = FALSO$ y $u_5 = VERDADERO$, Entonces: se concluye que $u_7 = VERDADERO$

El algoritmo MACIE produce justificaciones válidas que son usualmente de máxima generalidad y corre rápidamente, además, las justificaciones no necesariamente son únicas (Gallant, 1994). Este hecho puede ser considerado como una ventaja o una desventaja, dependiendo de la aplicación específica que se le quiera dar al sistema.

4.2.2.2. El Modelo RN de Variables de aproximación

Saito y Nakano (Saito y Nakano, 1990) desarrollaron un modelo diferente para extraer reglas de las redes neuronales. RN toma ventaja del conjunto de ejemplos de entrenamiento que fueron inicialmente usados para entrenar a la red, generando un conjunto pequeño de reglas que se van aproximando gradualmente a las reglas implícitamente codificadas en la red, con esfuerzos que crecen cercanos al cuadrado del número de ejemplos de entrenamiento.

RN trabaja tanto con modelos continuos como con modelos discretos y representa las reglas mediante condiciones booleanas modificadas, tales como:

$$u_1 \in [.2,.8] \text{ Y } u_3 \in [.3,.7] \text{ O NO } (u_5 \in [.2,.3] \text{ Y } u_3 \in [.6,.8])$$

En el algoritmo 7 se presenta la versión más conocida del modelo.

Dada una red neuronal N y un conjunto de ejemplos de entrenamiento $\{E^k\}$:

1. Seleccione un ejemplo positivo de entrenamiento E^k aún no cubierta por cualquier término.
Para cada variable de entrada j :
2. Encuentre un rango cercano a E_j^k que N clasifique como positivo (con otros componentes de E^k sin cambio)
3. Intersecte (Y's juntos) todos aquellos rangos que producen un nuevo término T , correspondiente a un hiperrectángulo,
4. Para cada ejemplo de entrenamiento negativo E^l no clasificado por T :
5. Encuentre rangos para E^l como en el paso 2a. y sustraiga los rangos de T , modificando T de tal manera que clasifique pocos puntos como verdaderos. No deje que el hiperrectángulo sustraído sobrepase a E^k en alguna coordenada.

6. Si algún ejemplo positivo no está cubierto por algún término, regrese al paso 1.
7. Una todos los términos por medio de O's para formar la regla final.

ALGORITMO 7. Algoritmo RN de Saito y Nakano.

4.2.2.3. El Modelo de los subconjuntos¹⁰

Fue propuesto por Towell y Shavlik (Towell, 1992) y consiste en una derivación del modelo RN de Saito y Nakano (Saito y Nakano, 1990) y también de la propuesta de Fu (Fu, 1994). Se le llama el modelo de los subconjuntos precisamente porque se basa en la búsqueda de subconjuntos de unidades de entrada que exceden el umbral de la unidad a la que se conectan en la siguiente capa. El modelo se puede mejorar usando técnicas ya clásicas de búsqueda, como la de primero en amplitud o primero a profundidad. En el caso de usarlo combinado con la búsqueda del primero en amplitud, se comenzaría seleccionando una unidad de la capa de salida, que sería la unidad asociada con una conclusión. En segundo lugar se determinarían si existen unidades en la capa inmediata anterior, que estén ligadas a esta y que por sí mismas proporcionen una activación lo suficientemente grande como para sobrepasar el umbral de esta unidad de la capa de salida. En caso de que no existan unidades que cumplan esta condición, entonces se buscarían conjuntos de unidades cuya combinación de pesos den satisfacción a la condición anterior.

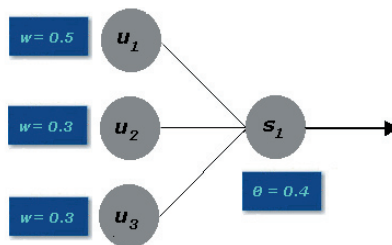


ILUSTRACIÓN 38. Red en la que una sola entrada es suficiente para producir un disparo.

¹⁰ *Subset*, en inglés.

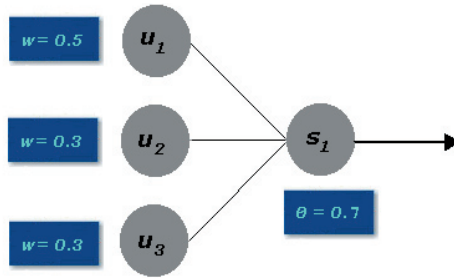


ILUSTRACIÓN 39. Red en la que el disparo depende de la combinación de varios pesos de entrada.¹¹

En la red de la ilustración 38, la unidad u_1 es suficiente para producir una activación en la unidad de salida, dado que se ha sobrepasado el umbral, esta condición no se cumple para las demás unidades. En el caso de la red de la ilustración 39, las unidades por sí solas no producen una activación en la unidad de salida, por ello se hace necesario buscar los posibles conjuntos de dos unidades que posibilitarían que la unidad s_7 se active, en este caso podría ser $\{u_1, u_2\}$ o $\{u_1, u_3\}$.

En este modelo de extracción la búsqueda prosigue incrementándose el tamaño de los subconjuntos de unidades buscados hasta que se ha explorado todo el campo. Con cada subconjunto se genera una regla de la forma Si- Entonces, en donde las condiciones se conforman a partir de las unidades seleccionadas, conectadas por conjunciones (Y's), y la conclusión se compone de la unidad de salida para la cual se han explorado y seleccionado los subconjuntos. Finalmente se anularían las reglas con menor generalidad dejando solamente a las de mayor generalidad.

En redes con más de una capa oculta, el procedimiento se realiza por cada capa y se hace necesaria la existencia de un subconjunto temporal en donde se almacenen las unidades seleccionadas participantes.

El procedimiento completo se describe en el algoritmo 8.

11 En este caso los valores individuales de las unidades de entrada no son suficientes por sí mismos para sobrepasar el umbral de la unidad de la capa de salida, por lo que se hace necesaria la combinación de dos o más unidades para la activación. Aquí la extracción de reglas debe contemplar estas posibles combinaciones.

Con cada unidad intermedia y de salida seleccionada:

1. Se extraen hasta β_p subconjuntos de unidades conectadas con pesos positivos cuya suma sea más grande que el umbral de la unidad seleccionada.
2. Con cada subconjunto P de los β_p subconjuntos encontrados en el paso 1:
3. Extraer hasta β_n conjuntos mínimos de ligas con pesos negativos cuya suma sea mayor que la suma de P menos el umbral de la unidad.
4. Siendo Z un nuevo predicado no usado antes.
5. Para cada subconjunto N de los β_n subconjuntos encontrados en el paso B.1 se conforma la regla: "Si N entonces Z"
6. Formar la regla "Si P y no Z entonces <Nombre de la unidad seleccionada>

ALGORITMO 8. Algoritmo de extracción por subconjuntos.

Uno de los problemas principales que se han identificado con este método es que el encontrar todos los subconjuntos en el espacio de búsqueda tiene un costo que crece conforme crece el número de unidades y ligas en la red. Por lo que se ha dicho que su uso eficiente se reduce exclusivamente a redes simples usadas en problemas pequeños.

4.2.2.4. El Modelo para extracción de reglas M de N.¹²

El modelo para extraer reglas M de N, también llamadas *MOFN*, trata de dar solución a los problemas que ha presentado el método de subconjuntos. Se buscan específicamente reglas del tipo:

Si: (al menos M de las siguientes N premisas es verdad), Entonces: (el concepto relacionado con el nodo en cuestión es verdadero).

Fue propuesto por Towell y Shavlik (Towell, 1992) debido a que el trabajo continuo con el modelo de los subconjuntos producía reglas del tipo M de N y esto precisamente fue lo que dio origen a la idea. Su uso se extendió rápidamente debido a que es posible obtener reglas disyuntivas ($M=I$) y también reglas conjuntivas (cuando $M=N$).

¹² *M of N (MOFN)*, en inglés.

La idea que subyace al modelo M de N es que los antecedentes (ligas) no tienen una importancia única, sino que más bien son los grupos de antecedentes de la misma clase de equivalencia los que comparten la misma importancia. De esta manera, cualquier unidad antecedente de una clase se puede intercambiar con las otras de la misma clase. Son las clases de equivalencia las que permiten al modelo M de N considerar grupos de ligas sin preocuparse por una liga particular al interior del grupo.

A continuación se presenta el esquema general de esta propuesta, en el algoritmo 9 (Craven y Shavlik, 1993).

1. Los pesos de las unidades de cada capa intermedia y unidad de salida de un sistema entrenado se agrupan. Inicialmente cada peso se trata como un grupo. Sucesivamente los dos grupos más cercanos se unen hasta que no existen grupos a una distancia previamente establecida. Las distancias se miden a partir de la diferencia entre las medias de cada grupo. Los pesos que no alcanzan a ser lo suficientemente significativos (también de acuerdo a parámetros previamente establecidos) se eliminan.
2. La magnitud de cada peso se iguala al promedio del grupo en el que fue incluido.
3. Se eliminan los pesos de las unidades que no son necesarios a fin de activar correctamente la unidad a la que están conectadas. Se aplican dos procedimientos de eliminación: uno es algorítmico y el otro heurístico. La eliminación algorítmica identifica los grupos que no pueden tener un efecto en los nodos de la siguiente capa, o aquellos que no exceden el valor umbral establecido. La eliminación heurística consiste en la selección y retiro de aquellos grupos que no han tenido algún efecto durante los casos de entrenamiento.
4. Los umbrales de las unidades se modifican para adaptarse a los cambios que se han realizado en los pasos anteriores.
5. Cada unidad de la capa oculta y de la capa de salida se traduce en una regla con antecedentes ponderados de manera tal que la consecuencia es verdad si la suma de los antecedentes ponderados excede al umbral.
6. Los pesos y los umbrales se eliminan y las reglas se presentan en formato M de N.

ALGORITMO 9. Algoritmo *M de N*.

Este enfoque hace uso tanto de ejemplos de entrenamiento, como de consultas al mismo sistema, para poder extraer conjuntos de reglas que se acercan mucho a la descripción del sistema al que están representando. Sin embargo, una de las principales desventajas identificadas en la actualidad, es su dependencia directa de la cantidad y naturaleza de los ejemplos de entrenamiento seleccionados para el proceso de extracción de reglas.

4.2.2.5. El Modelo de refinamiento de reglas en redes neuronales difusas de Masuoka

Otro de los primeros trabajos en extracción/ refinamiento de reglas difusas es el realizado por Masuoka y otros (Masuoka, Watanabe y Kamura, 1990). Esta propuesta se basa en la realización de un proceso compuesto por tres fases distintas:

- a) En la fase de inicio, Masuoka implementó una red neuronal artificial de tres capas que se componía de solamente una célula de entrada, dos en la capa intermedia y una en la capa de salida. El objetivo de esta red sencilla en la capa de entrada era el proceso de las entradas y la representación de las funciones de membresía para cada antecedente de las reglas.
- b) Esta segunda fase llamada RULENET tenía por objetivo realizar operaciones difusas como la conjunción difusa y la disyunción difusa entre otras, sobre los resultados del proceso realizado en la fase de entrada.
- c) Los consecuentes de las reglas se constituían a partir de las funciones de membresía que eran calculados en la misma forma que en la fase inicial.

El aspecto más sobresaliente de este modelo de afinación de reglas difusas fue el hecho de que en la fase RULENET se realizaba una poda de aquellas conexiones cuyo valor no alcanzara a rebasar el umbral, esto con el fin de reducir considerablemente la complejidad del espacio de búsqueda.

Aplicaciones posteriores como las propuestas por Berenji (Berenji, 1991), Horikawa y otros (Horikawa, Fusuhashi y Uchikawa, 1992) y Halgamuge y Glesner y otros (Halgamuge y Glesner, 1994), (Halgamuge, Poechmueller y Glesner, 1994) reportaron las ventajas del uso de redes neuronales especializadas para la afinación de las reglas mediante el cálculo de fun-

ciones de membresía y la identificación de nodos relevantes para las reglas. A continuación se presenta el modelo que ha sido reconocido como el más sobresaliente de estos tres: el modelo FUNEGEN de Halgamuge y Glesner.

4.2.2.6. El Modelo de extracción de reglas en redes neuronales difusas de Halgamuge y Glesner

El modelo FUNEGEN fue propuesto por S.K. Halgamuge y M Glesner del Instituto de Sistemas Microelectrónicos de la Darmstadt University of Technology en Alemania. Según el mismo Halgamuge, su propuesta es derivada de la propuesta presentada por S. Horikawa y otros (Horikawa, Fusuhashi y Uchikawa, 1992) en cuanto al método de evaluación de las premisas y a la creación de las funciones de membresía en los antecedentes de cada regla. Sin embargo, FUNEGEN no requiere de una base de reglas inicial, debido a su peculiar arquitectura y el modelo difuso. Esta es realmente una de las características principales del modelo que lo hacen sobresalir del resto de los intentos anteriores, debido a que es el punto en que la extracción de reglas se desliga del concepto de “solamente refinamiento” con el que hasta antes se les identificaba.

Otra de las ventajas de esta propuesta es su facilidad para manejar números grandes de entradas o variables. Si se previene la aparición repetida de una entrada en la premisa, el número de todas las posibles reglas simples conjuntivas y disyuntivas dependen del número de entradas (N_I) y el número de adjetivos (N_A) seleccionados:

$$N_i * N_A + 2 * \left(\sum_{i=2}^{N_I} N_A^i * \binom{N_I}{i} \right) \quad (4,2)$$

Si el número máximo de variables en la premisa se limita a 2 (haciendo uso de una de las estrategias antes mencionadas para reducir la complejidad de los problemas a tratar), entonces el número posible de reglas se vuelve mucho más manejable, ya que se reduce a:

$$N_i * N_A (N_A (N_I - 1) + 1) \quad (4,3)$$

FUNEGEN (ver ilustración 40) emplea un enfoque diferente a las propuestas anteriores para encontrar la base inicial de reglas. En los otros

modelos la base inicial se establecía a partir de la intervención de expertos en el dominio, mientras que FUNEGEN fue diseñado para efectuar esta tarea en forma automática a partir de la selección de nodos con influencia significativa tanto para reglas conjuntivas (Si) como para reglas disyuntivas (Si No).¹³

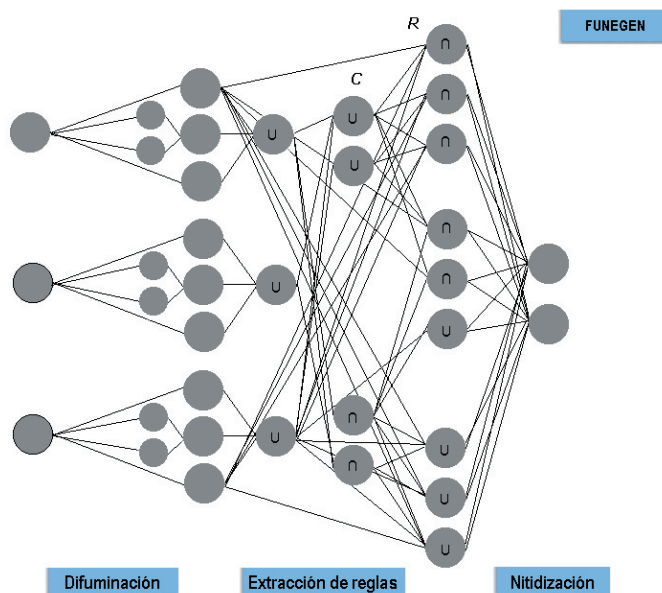


ILUSTRACIÓN 40. Red del tipo FUNEGEN para extracción de reglas.

El método empleado para la preselección de las reglas reduce drásticamente la base de reglas inicial (que en un principio se compone por todos los casos del conjunto de entrenamiento). Después de un número de pasos con funciones de membresía “congeladas”, que le permiten a FUNEGEN prevenir los mínimos locales en la generación de sus reglas, la red es entrenada con funciones de membresía de “movimiento libre”. Además, algunos nodos se activan solamente con una función sigmoide y otros con una función lineal.

En el ejemplo de la ilustración 40 los nodos marcados con $\cup(\cap)$ tienen cálculos de entrada Soft Min y Soft Max (siendo K una variable que puede sufrir un incremento hasta el infinito):

¹³ *IF* e *IF NOT*, respectivamente, en inglés.

$$Min_K(I_1 \dots I_n) = \frac{\sum_{i=1}^n I_i \cdot e^{-K \cdot I_i}}{\sum_{i=1}^n e^{-K \cdot I_i}} \quad (4,4)$$

$$Max_K(I_1 \dots I_n) = \frac{\sum_{i=1}^n I_i \cdot e^{K \cdot I_i}}{\sum_{i=1}^n e^{K \cdot I_i}} \quad (4,5)$$

El resto de las neuronas sólo suman las entradas que reciben (Nauck y Kruse, 1995).

Si se considera un ejemplo donde $N_A = 3$, es decir, un caso con tres adjetivos, el procedimiento a seguir sería el mostrado en el algoritmo 10.

Considerando que los valores difusos de un conjunto nítido I_i son L_i , M_i y H_i .

1. El primer paso sería tomar la decisión acerca de si la i ésima tiene alguna influencia sobre una regla conjuntiva (Min) (ver ilustración 41).
2. Se conecta la capa de difuminación al nodo C_i (sub capa "C" en la capa de extracción de reglas de la ilustración 41) que selecciona a partir de los valores de membresía más fuertes para todas las entradas, menos la entrada i :

$$C_i = Max_{[j \neq i]}^{\forall i} [Max_j(L_j, M_j, H_j)] \quad (4,6)$$

3. Las conexiones correspondientes a los nodos R_{L_i} , R_{M_i} y R_{H_i} (capa R en la ilustración 41) se realizan como sigue:

$$R_{L_i} = Min(C_i, L_i) \quad (4,7)$$

$$R_{M_i} = Min(C_i, M_i) \quad (4,8)$$

$$R_{H_i} = Min(C_i, H_i) \quad (4,9)$$

4. Los pesos W_{L_i} , W_{M_i} y W_{H_i} que conectan a los nodos correspondientes con la salida, se inicializan al azar.

5. Después del proceso de entrenamiento los pesos se analizan para decidir cual de las tres entradas difusas es relevante para la regla *Mínima si*¹⁴ o *Mínima si no*.¹⁵
6. El último paso sería el listado de todos los nodos relevantes, la fuerza de la influencia se considera a partir de una medida definida por el usuario.

ALGORITMO 10. Pasos principales en la extracción de reglas que realiza FUNEGEN.

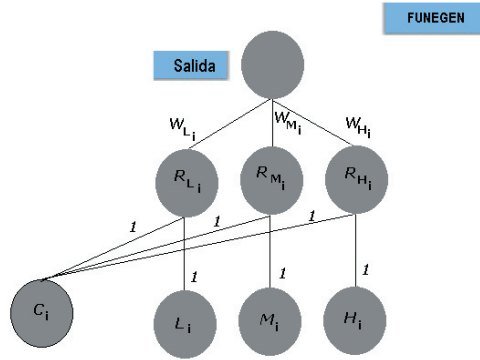


ILUSTRACIÓN 41. Extracción de nodos relevantes en FUNEGEN.

El Proceso para la extracción de los nodos relevantes para la regla Max es análogo al presentado en el algoritmo 10 método anterior, considerando:

$$C_i = \text{Min}_{[j \neq i]}^{\forall i} [\text{Max}_j (L_j, M_j, H_j)] \quad (4,10)$$

y estableciendo los nodos R_{L_i} , R_{M_i} y R_{H_i} de la siguiente manera:

$$R_{L_i} = \text{Max}(C_i, L_i) \quad (4,11)$$

$$R_{M_i} = \text{Max}(C_i, M_i) \quad (4,12)$$

$$R_{H_i} = \text{Max}(C_i, H_i) \quad (4,13)$$

14 *Min If*, en inglés.

15 *Min if not*, en inglés.

Todos los nodos relevantes de regla Max y Min pueden ser considerados como candidatos para nodos simples de regla. Los resultados obtenidos para el proceso de extracción de nodos de regla relevantes Min del ejemplo utilizado aquí, se muestran en la ilustración 42.

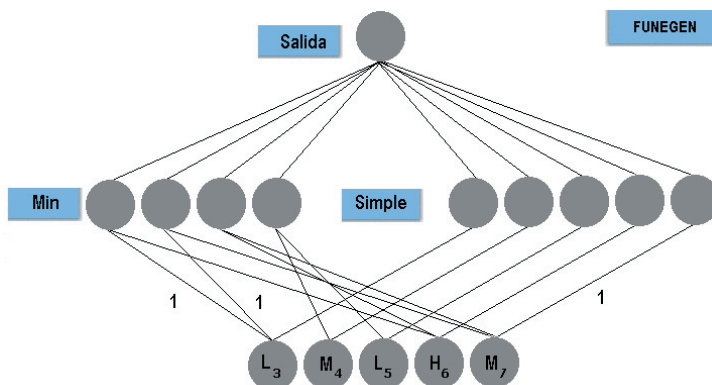


ILUSTRACIÓN 42. Base de reglas inicial para el proceso de extracción de reglas en FUNEGEN.

Después de que el proceso de creación de la base inicial de reglas ha sido completado, se puede comenzar la extracción a partir del mismo conjunto de entrenamiento. FUNEGEN posibilita al usuario el establecimiento de límites para la cantidad de reglas, dado que en algunos casos el número puede ser muy grande, esto se logra estableciendo un nodo a manera de umbral que permita remover todas las reglas que no alcancen a sobrepasar su límite. FUNEGEN ofrece la posibilidad de hacer modificaciones sobre la marcha misma del proceso de extracción, esto permite que el establecimiento de estos parámetros mínimos para la generación de reglas se pueda realizar “en línea” a fin de reducir complejidad y tiempos de ejecución.

Sin embargo, el encontrar la cantidad mínima de reglas es un proceso que debe ser realizado a partir de iteraciones y no deja de ser un procedimiento “manual” que descansa en el criterio del usuario del sistema. El sistema posibilita la extracción de reglas a partir de un conjunto inicial de casos sin necesidad de preestablecer reglas de trabajo, pero las reglas extraídas no necesariamente son reglas máximamente generales.

4.2.2.8. El Modelo de extracción de reglas de redes neuronales difusas de Nauck y Kruse.

Detlef Nauck y Rudolf Kruse del Departamento de Ciencias Computacionales de la Universidad Técnica de Braunschweig, en Braunschweig, Alemania, propusieron el modelo NEFCLASS¹⁶ (Nauck y Kruse, 1995; Nauck, Klawonn y Kruse, 1995; Nauck, Nauck y Kruse, 1996; Nauck, 2003 y 2004) que se basa en métodos de agrupamiento difuso.¹⁷

El modelo NEFCLASS fue diseñado para determinar la clase o categoría correcta C , que es un subconjunto difuso de \mathfrak{R}^n , de patrones de entrada $X = (x_1, \dots, x_n) \in \mathfrak{R}^n$ en una red neuronal, asumiendo que la intersección de dos clases distintas es un conjunto vacío. Los valores de los patrones de características se representan como conjuntos difusos y la clasificación se describe por medio de un conjunto de reglas lingüísticas. Para cada característica de entrada x existen q_i conjuntos difusos $\mu_1^{(i)}, \dots, \mu_{q_i}^{(i)}$ y la base de reglas contiene k reglas difusas desde R_1, \dots, R_k (ver ilustración 43).

La base de reglas es una aproximación para una función desconocida $\phi: \mathfrak{R}^n \rightarrow \{0,1\}^m$ que representa la tarea de clasificación, donde $\phi(x) = (c_1, \dots, c_m)$ de manera que $c_i = 1$ y $c_j = 0 (j \in \{1, \dots, m\}, j \neq i)$, por ejemplo, cuando x pertenece a la clase C_i . Inicialmente la base de reglas no se aproxima a ϕ , pero la función $\phi': \mathfrak{R}^n \rightarrow [0,1]^m$ sí lo hace. Se obtiene $\phi(x)$ por medio de la ecuación $\phi(x) = \Psi(\phi'(x))$ donde Ψ representa la interpretación de la clasificación resultante obtenida por NEFCLASS.

Las características principales de NEFCLASS son:

- a) El conjunto de reglas lingüísticas y los conjuntos difusos que realizan esta aproximación se obtienen a partir del conjunto de casos de entrenamiento.
- b) Se tienen pesos compartidos en algunas de las conexiones, de manera que los valores lingüísticos tengan una sola representación como conjunto difuso para todas las unidades de regla (R_1, \dots, R_k en la ilustración 43).

Esta última característica impide que dos conjuntos difusos que sean idénticos al inicio del entrenamiento del sistema, se lleguen a desarrollar

16 *Neuro Fuzzy CLASSification*, en inglés (Clasificación neuro-difusa).

17 *Fuzzy Clustering*, en inglés.

en forma distinta. Igualmente, propicia que las conexiones que tienen los pesos compartidos siempre provengan de la misma unidad.

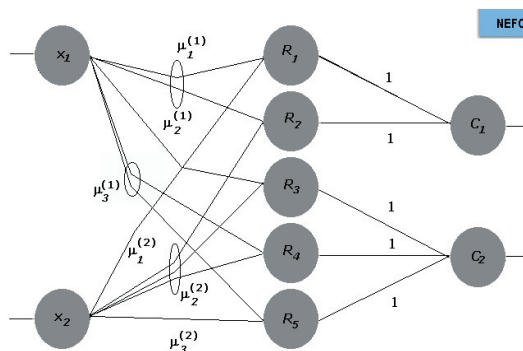


ILUSTRACIÓN 43. Un sistema tipo NEFCLASS con dos entradas, cinco reglas y salida con dos clases.

El algoritmo 11 presenta completo el procedimiento para el aprendizaje/extracción de reglas de NEFCLASS propuesto en (Nauck y Kruse, 1995).

Considérese un Neuroclasificador difuso con n unidades de entrada, x_1, \dots, x_n , con R_1, \dots, R_k unidades de regla, donde $k \leq k_{max}$ y m unidades de salida que van desde c_1, \dots, c_m . Y donde también se ha establecido una tarea de aprendizaje $\tilde{L} = (n, t_1), \dots, (p_s, t_s)$ con s patrones consistentes, cada uno en $n \in \mathcal{R}^n$ entradas, y una salida que también consiste en un patrón $t \in \{0,1\}^m$.

El algoritmo que se emplea para crear k unidades de regla, consiste en los siguientes pasos:

1. Seleccione el siguiente patrón (p, t) de \tilde{L}
2. Para cada unidad $x_i \in U_1$ encuentre la función de membresía $\mu_{j_i}^{(i)}$ tal que: $\mu_{j_i}^{(i)}(p_i) = \max_{j \in \{1, \dots, q_i\}} \{\mu_j^{(i)}(x_i)\}$
3. Si existen menos nodos de regla que k y no existe un nodo de regla R con $W(x_i, R) = \mu_{j_i}^{(1)}, \dots, W(x_n, R) = \mu_{j_n}^{(n)}$, entonces se crea el nodo y se conecta al nodo de salida c_l si $t_l = 1$.
4. Si quedan aún patrones sin procesar en \tilde{L} y $k \leq k_{max}$ entonces se debe ir al paso (i). Si no es así, entonces se detiene.
5. Para adaptar los conjuntos difusos se corre cíclicamente un algoritmo

de aprendizaje supervisado a lo largo de \tilde{L} repitiendo los siguientes pasos hasta que se alcanza un criterio preestablecido:

a) Se selecciona el siguiente patrón (p, t) de \tilde{L} y se propaga a lo largo del sistema NEFCLASS, a continuación se determina el vector c .

b) Para cada unidad de salida c_i se determina el valor delta $\delta_{c_i} = t_i - o_{c_i}$

6. Para cada unidad de regla R con $o_R > 0$:

a) Determine el valor delta: $\delta_R = o_R(1 - o_R) \sum_{c \in U_3} W(R, c) \delta c$

b) Encuentre una x' tal que: $W(x', R)(o_{x'}) = \min_{x \in U_1} \{W(x, R)(o_x)\}$

c) Para el conjunto difuso determine los valores delta para los parámetros a, b, c usando el rango de aprendizaje $\delta > 0$:

$$\delta_b = \sigma \cdot \delta_R \cdot (c - a) \cdot \text{sgn}(o_{x'} - b),$$

$$\delta_a = -\sigma \cdot \delta_R \cdot (c - a) + \delta_b,$$

$$\delta_c = \sigma \cdot \delta_R \cdot (c - a) + \delta_b,$$

y aplique los cambios a $W(x', R)$ si no comete ninguna violación del conjunto de restricciones preestablecido ϕ . (Debe considerarse que el peso $W(x', R)$ puede estar compartido por otras conexiones y en este caso deben ser cambiados más de una vez).

7. Si un ciclo es completado y se alcanza el criterio preestablecido de terminación, entonces el programa para. Si no es así, se regresa al paso (i) de esta segunda fase del algoritmo.

ALGORITMO 11. Algoritmo del modelo NEFCLASS.

Una de las principales ventajas del modelo es la posibilidad de extraer reglas durante el mismo proceso de entrenamiento a partir de un conjunto de patrones de entrada iniciales. Sin embargo, existen desventajas considerables: a) se presupone la existencia de un conjunto de reglas iniciales a partir de las cuales se da la configuración inicial de la red, por lo que se trata de un modelo de refinamiento de reglas y no de extracción; y b) no es capaz de producir reglas máximamente generales, de hecho, puede darse el caso de que considere reglas que podrían ser

clasificadas dentro de un solo grupo, como si fuesen miembros de dos clases distintas.

4.2.3. Estrategias para aligerar el proceso de extracción de reglas

En los métodos existentes para extracción de reglas, a menudo una limitante importante es el peligro de enfrentar una explosión combinatoria o de tratar con un problema de complejidad no deseada. Es por ello que se usan algunas estrategias que permiten delimitar el espacio de trabajo, a fin de sortear estos problemas.

4.2.3.1. Limitación del número de condiciones

Una estrategia que ayuda a prevenir los riesgos de la explosión combinatoria y bajar la complejidad, es la reducción del número aceptable de condiciones en cada regla y el establecimiento de un límite. A esta estrategia se le conoce también como *establecimiento de contextos* (Gallant, 1994).

Establecer el número máximo de variables que pueden aparecer como condiciones en una regla no es una limitación severa como podría aparecer a primera vista. Se puede sacar ventajas de esta delimitación, si se les considera como una exploración detallada de subconjuntos en el espacio de búsqueda. Sin embargo, también puede dar el caso de que se lleguen a ignorar condiciones que podrían ser de suma importancia en el establecimiento de la conclusión. Establecer topes en el número de condiciones de las reglas es una estrategia que se fundamenta en el supuesto de que la generalidad máxima, por lo general, se logra a partir de las reglas mínimas. Aunque siempre existe la posibilidad de que esta medida lleve a pasar por alto combinaciones de elementos de entrada que, por exceder el tope no son exploradas y cuya configuración no se pueda reducir a ninguna de las reglas mínimas encontradas dentro de los límites establecidos.

4.2.3.2. Reducción del conjunto de entrenamiento

En el caso de los problemas que implican el manejo de variables continuas, una estrategia que posibilita poder trabajar sin tener que ensayar todas las posibles combinaciones, es la reducción del conjunto de casos de prueba.

Esta estrategia contempla el uso de solo algunos casos “modelo”, que son los casos más significativos dentro de ciertos rangos establecidos a partir del uso de medidas de tendencia central.

Otro criterio de selección es también la factibilidad con la que ciertos casos se pueden dar en la vida real.

4.2.3.3. Uso de conjuntos de entrenamiento como si fuesen universos completos

Otra estrategia más para reducir la complejidad del proceso requerido, es basar la extracción de reglas solamente en el conjunto de entrenamiento del sistema, es decir, considerando al conjunto de casos de entrenamiento como si fuese un universo completo y no una muestra. Por supuesto que la efectividad de esta estrategia descansa y depende completamente del proceso de selección e integración del conjunto de casos de entrenamiento.

4.3. Futuro de la extracción de reglas

En la actualidad son muchos ya los esfuerzos que se están realizando para poder dotar a las redes neuronales de la posibilidad de ser comprendidas mediante la traducción de sus configuraciones internas subsimbólicas a reglas del tipo Si- Entonces. Algunos de los investigadores de más prestigio en este campo apuntan que el futuro de la extracción de reglas es promisorio (Andrews, Cable, Diederich, Geva, Golea, Hayward, Ho-Stuart y Tickle, 1996 y Andrews, Diederich y Tickle, 2001), incluso han llegado a establecer que el futuro de las redes neuronales está estrechamente ligado al de la extracción de reglas, de manera tal que los nuevos modelos de redes neuronales aplicados a ciertos problemas del mundo real tienen que contemplar forzosamente la posibilidad de facilitar el proceso de extracción de reglas desde su propia fase de diseño.

Las demandas de modelos de extracción de reglas que se prevén para un futuro próximo atienden a dos grandes grupos de necesidades y aplicaciones:

- a) Se requieren de modelos de extracción de reglas para redes neuronales ya existentes. Es preciso conocer todo el conocimiento que se ha

logrado manejar ya con éxito en redes neuronales desde hace treinta años, y

- b) Se requieren nuevos modelos de extracción de reglas (e incluso nuevos modelos de redes neuronales) que atiendan a necesidades nuevas. Es común la existencia de cúmulos de datos relacionados con dominios específicos del conocimiento, para los cuales no se conocen correlaciones internas y no se han establecido conocimientos nuevos a partir de ellos. Este campo de desarrollo es virgen aún, por lo que las formas específicas de participación de los modelos de extracción de reglas en éste aún están delineándose en la actualidad. Un ámbito en el que se ha incrementado recientemente el uso de redes neuronales con capacidades de extracción de reglas es la *minería de datos* para los sitios de Internet, demanda que exige que se calculen reglas en forma automática acerca del comportamiento de visitantes a sitios WWW, con la finalidad de presentarles en forma personalizada, la información que tiene mayor probabilidad de ser útil para cada uno de ellos.

5

La extracción de reglas en redes neuronales difusas con base en la Binarización de los Conjuntos Difusos. La propuesta del Modelo BCD

Los modelos actuales para extraer o generar reglas a partir de las configuraciones logradas en una red neuronal nítida o una red neuronal difusa, se basan en alguno de los dos posibles enfoques, o en la combinación de ambos:

- a) Un enfoque *duro*, que prueba y extrae cada una de las posibles relaciones entre entradas y salidas de la red. La prueba se realiza nodo por nodo, con el riesgo sumamente cercano a enfrentarse a una complejidad de proceso que vuelve intratables los problemas de escala mediana y grande.
- b) Un enfoque estadístico combinado con agrupaciones y simplificaciones, que reduce las posibilidades de especificidad y granularidad de la información, pero ofrece, a cambio, mejores posibilidades de manejo y viabilidad de implementación

Estos modelos se han implementado usando una estrategia de recorte de campo de posibilidades. Estrategia que se fundamenta en la representatividad de los conjuntos de datos de entrenamiento y que implica el uso del conjunto de datos de entrenamiento como universo único de extracción.

Sin embargo, en la actualidad no se puede hablar de que uno u otro enfoque o determinada forma especial de combinación de ambos tenga un mejor desempeño en problemas del mundo real de manera tal que se pueda excluir a los demás. El presente apartado tiene por objetivo proponer y presentar al lector un algoritmo original inspirado en los modelos de extracción de reglas que han sido diseñados a la fecha. El modelo creado pretendió incorporar algunas de las ventajas de sus predecesores y superar algunas de las deficiencias más significativas. Además, junto con la presentación del modelo se proponen modificaciones substancia-

les a la estrategia de recorte de campo que lleva a una cobertura más amplia de las reglas extraídas.

5.1. El modelo BCD: una solución distinta para el mismo problema

El modelo que en documento se propone y presenta es el *Método de Extracción de Reglas Difusas basado en la Binarización de los Conjuntos Difusos* (BCD).

El BCD es un intento por lograr la simplicidad en el proceso de extracción de reglas difusas, tratando de superar, o al menos igualar a un menor costo, las ventajas y beneficios que ofrecen los enfoques contemporáneos en extracción de reglas.

BCD implica una forma de solucionar el problema de la extracción de reglas en una forma distinta:

- A) BCD se basa en la simplificación de los conjuntos difusos que se pueden establecer en un grupo de datos de entrada, a partir de su binarización. Esta estrategia permite lograr una reducción significativa en la complejidad de proceso de un conjunto de datos, y
- B) BCD implica el uso de todas las posibilidades de configuraciones de entrada para extraer reglas a partir de la forma en que la red entrenada responde a ellas. Con esta forma de trabajo la extracción de reglas no se realiza en un campo recortado, sino que contempla todas las posibles formas de configuración que un problema pueda presentar.

El modelo BCD, por sus características podría ser ubicado dentro de la clasificación de *algoritmos de caja negra* o *métodos globales*, debido a que su uso es aplicable a partir de entradas y salidas en el sistema. Es aplicable a una gran variedad de redes neuronales que integren entre sus características el proceso difuso de la información e incluso a algunos sistemas difusos no-neuronales de manejo del conocimiento.

5.1.1. Binarización de entradas vs. Binarización de conjuntos difusos

Antes de explicar el algoritmo completo del modelo BCD se hace pertinente explicar un concepto que es nodal en el mismo: la *binarización de los conjuntos difusos*.

La binarización como estrategia de simplificación de problemas de clasificación es relativamente nueva, ya que no es sino hasta la década de los noventa que se empieza a trabajar en modelos propuestos para extracción de reglas. Los trabajos de Taha y Gosh (Taha y Gosh, 1996a y 1996b); Herrmann (Herrmann, 1995) y Kasabov (Kasabov, 1998) son ejemplos tácitos del uso de la binarización, sin embargo, no se aplicaba sobre conjuntos difusos, sino sobre rangos preestablecidos de valores (Kasabov, 1998 y Kasabov, Kozma y Duch, 1998); sobre ciertos valores “pico” (Herrmann, 1995), o inclusive para representar la activación mayor a la media aritmética esperada en las unidades de entrada de una red neuronal.

El modelo BCD fue ideado con base en las propuestas anteriores, aunque se tuvieron que hacer modificaciones substanciales que permitieran el manejo difuso que se requiere en la extracción de reglas difusas.

En BCD, la binarización no se da directamente sobre los valores de las entradas, sino más bien sobre los conjuntos difusos establecidos para cada variable. Para binarizar los conjuntos difusos en una red neuronal se hace necesario, en primer lugar, la creación de tantos nodos de conexión en la capa de entrada, como conjuntos difusos se hayan estructurado tras el análisis de cada variable y su posterior difuminación. Finalmente, el valor que cada uno de estos nodos creados tendría se establece acorde con la función de membresía calculada para el conjunto de datos con el cual se va a trabajar.

Las entradas difusas se binarizan de la siguiente manera:

$$Entrada_nueva_i = \begin{cases} 1_si_ \mu_A(x_i) > \theta \\ 0_de_otra_forma \end{cases} \quad (5,1)$$

En donde $Entrada_nueva_i$ es la entrada correspondiente a x_i que se introducirá en un nuevo sistema. La entrada x_i se convierte a binario con base en su grado de membresía al conjunto difuso A , denominado $\mu_A(x_i)$, comparado con un umbral θ preestablecido.

Dado que pueden existir clases que no son linealmente separables, una estrategia propuesta por James M. Keller y Douglas J. Hunt (Keller y Hunt, 1992) es el uso de los valores de membresía que sobrepasan a $\frac{1}{2}$ solamente. En BCD se plantea el uso de esta estrategia y se ha establecido como límite $\theta = \frac{1}{2}$. Si el umbral θ es sobrepasado, entonces la activación del nodo de entrada correspondiente al conjunto difuso A (derivado del nodo de entrada original X) es igualado a 1.

5.1.2. Uso de un subconjunto de datos vs. uso de todas las posibilidades

Existen diferencias entre usar el conjunto de todas las combinaciones posibles de configuración como entradas para el problema y el uso de solo los datos de entrenamiento. Esta es una de las premisas en las que se basó esta investigación.

La diferencia principal estriba en la generalidad de la base de reglas obtenida tras la extracción. Por ejemplo, considérese una regla como la presentada en (5,2) extraída solamente con base en el conjunto de datos de entrenamiento:

Si Variable_de_entrada_1 es (Chica y Mediana) [no Grande]... (5,2)

Podría darse el caso de que, usando todas las posibles configuraciones de entrada, se pudiese llegar a la conclusión de que no existe tal restricción, por lo que la regla resultante debería integrar entonces, en vez de la premisa como en (5,2), una premisa como la presentada en (5,3).

Si Variable_de_entrada_1 es [Cualquiera]... (5,3)

Sin embargo, a pesar de que se reconoce que existen diferencias a este respecto, usar la población entera es prácticamente imposible en la mayoría de los problemas de la vida real, sobre todo en los casos en los que se habla de variables continuas y no discretas. Este problema se ha tratado de abordar usando la estrategia de partición del conjunto de datos de entrenamiento al azar y utilizando una de las partes como conjunto de entrenamiento y la otra como conjunto de prueba. Bajo una estrategia de este tipo, la extracción de reglas sería, sin embargo, *general solamente al conjunto de datos de entrenamiento*, ya que se ha visto cómo una red responde efectivamente ante problemas no usados, pero que son parte de este mismo conjunto de entrenamiento. Es decir, esta estrategia se basa en la prueba de una red neuronal entrenada con base en una muestra representativa, mostrándole otra muestra igualmente representativa.

En este documento se propone una estrategia distinta. En el trabajo con el modelo BCD se entrena a la red con base en un conjunto de datos asumidos como representativos de una población y se extraen las reglas de la red a partir de todas las posibles configuraciones de entrada que se pueden presentar a la red.

Por supuesto que este enfoque tiene detrás tres supuestos principales:

- a) Extraer reglas de una red neuronal representativa a partir del monitoreo de su comportamiento ante todas las posibles entradas, se posibilita el monitoreo y la indagación sobre el comportamiento de la red neuronal ante casos especiales.
- b) El conocimiento que se obtiene mediante esta estrategia de trabajo es más completo, ya que permite conocer exactamente cómo se comporta la red neuronal ante cada uno de los posibles casos y asimismo la extracción de reglas realizada asegura contar con reglas que cubren todos los casos posibles.

Sin embargo, aún queda otro problema. ¿Cómo evitar una posible explosión combinatoria?. BCD permite el uso de todas las posibilidades al binarizar los conjuntos difusos. Con ello, el número de estas posibilidades de entrada que se tienen que considerar se reduce considerablemente.¹

5.1.3. El algoritmo del modelo BCD

El algoritmo usa inicialmente la correspondencia entre entrada/salida que el sistema entrenado realiza, y la usa para configurar de una red neuronal nueva a partir de las mismas. Ya con la red entrenada, se introduce un conjunto más amplio de datos que contiene todas las posibles configuraciones derivadas de los resultados iniciales, para finalmente componer reglas a partir de estas nuevas configuraciones de entrada y salida correspondientes.

El BCD ofrece amplias posibilidades de aplicación a pesar de su sencillez, debido a que el algoritmo no implica algún proceso especial de entrenamiento, ni depende de alguna arquitectura específica de red. Además, se caracteriza por ser un método simple que permite el uso de procesos bien conocidos como podría ser la retropropagación del error y la minimización lógica.

BCD requiere solamente, los patrones de entrada y las salidas correspondientes a estas entradas, por lo que su uso puede ser extendido a

1 No se presenta aquí a la *Binarización de los conjuntos difusos* como la *panacea* a este problema, no se descarta el hecho de que existen problemas que por la cantidad de variables y conjuntos difusos, hicieran prácticamente imposible el trabajo con todas las posibilidades. Sin embargo, las reglas extraídas en este tipo de casos serían tan complejas de interpretación para el humano, que la extracción misma de reglas perdería sentido. Por lo que este tipo de problemas no se tratan en este trabajo.

otro tipo de sistemas, incluso más allá de los sistemas conexionistas o neuronales.

5.1.4. Esquema general del proceso

El modelo de extracción de reglas basado en la Binarización de los Conjuntos Difusos, que aquí se presenta integra cinco fases generales de trabajo:

1. Fase de preparación de datos de entrada.
2. Fase de creación y entrenamiento de la red con base en estos nuevos patrones de entrada.
3. Creación del conjunto total de posibles patrones de entrada.
4. Evaluación de este conjunto total de posibilidades en la red neuronal creada, establecimiento de las nuevas relaciones entre estos patrones de entrada y sus salidas correspondientes y extracción del conjunto inicial de reglas.
5. Refinamiento del conjunto de reglas inicial para llegar a reglas con una máxima generalidad.

Los siguientes apartados de este documento presentan el procedimiento completo para la realización de cada una de las fases anteriores.

5.1.5. Procedimiento para la fase 1: Preparación de los datos de entrada

El procedimiento para esta primera fase del proceso de extracción de reglas BCD implica varias etapas:

- a) Recolección de los conjuntos de datos de entrenamiento que se usarán para el ajuste inicial de pesos en la red neuronal creada para el proceso de extracción.
- b) Definición de las categorías difusas que comprende el dominio del conocimiento bajo el cual el proceso de extracción de reglas operará, y el cálculo de funciones de membresía acordes a esta definición.
- c) Armado de un conjunto de valores calculados con base en las funciones de membresía recién definidas. Este conjunto de valores se

puede considerar ahora como el conjunto de datos de entrada que el modelo emplea para el proceso de extracción de reglas.

El Algoritmo 12 presenta el procedimiento completo para la fase 1 del Modelo BCD:

1. Se organiza el conjunto de datos de entrenamiento del problema original en tuplas dispuestas en pares $\{entrada, salida\}$. Siendo una entrada el vector en el que cada uno de sus elementos corresponde a cada una de las características de entrada que se quieren clasificar, la salida se compone por la categoría o clase que se ha identificado como correspondiente para este vector de entrada.
2. Para cada variable del programa original (cada elemento del vector de entradas) se establecen n conjuntos difusos correspondientes a las categorías difusas que los valores de entrada pueden adoptar. Aunque la elección de estos conjuntos se realiza con base en las características del problema en cuestión, se depende en gran medida del arbitrio del usuario que aplique el modelo.
3. Se calculan las funciones de membresía que a la postre indicarán el grado de membresía que cada elemento tiene para cada conjunto.
4. Se calculan los grados de membresía que el valor de cada variable tiene para un patrón determinado en cada uno de los conjuntos difusos establecidos para esta variable
5. Se arma para cada patrón una nueva tupla $\{entrada_nueva, salida\}$ en el que $entrada_nueva$ se compone ahora de vectores formados a partir de los valores de las funciones de membresía que cada variable tiene en los conjuntos difusos definidos previamente

ALGORITMO 12. Fase 1 del modelo de extracción BCD.

5.1.6. Procedimiento para la fase 2: Creación y entrenamiento de la red

La siguiente fase del modelo implica la creación de una red neuronal con base en las características del conjunto de datos de entrada que fue creado en la fase 1. Asimismo, en esta fase se definen las salidas deseadas en la red neuronal para que el entrenamiento se realice, si es que el paradigma de aprendizaje seleccionado fue el supervisado. En este documento se asume el trabajo con una red neuronal que aprende con

base en la retropropagación del error, es decir, que aprende en forma supervisada, por lo cual el proceso presentado en el algoritmo 13 contempla ese paso.

En el algoritmo 13 se presenta el procedimiento completo para la fase 2 del modelo BCD.

6. Crear una red neuronal que cuente con los siguientes requisitos:

a) El número de entradas debe ser:

$$\text{Entradas_de_la_red} = \sum_{i=1}^n \text{entrada}_i \cdot C_{\text{entrada}_i} \quad (5,4)$$

En donde *entrada* es el conjunto de entradas para los patrones del problema original, cuyo número va desde 0 hasta *n*, y C_{entrada_i} es el número de conjuntos difusos establecidos para *entrada_i*.

b) El número de salidas debe ser igual al número de clases finales distintas establecidas en el problema original.

c) Para proporcionar los patrones de entrada a la red se tienen que binarizar los datos contenidos en cada uno de los vectores que están contenidos en $\{\text{entrada_nueva}, \text{salida}\}$ de la siguiente manera:

$$\text{entrada_nueva}_i = \begin{cases} 1 & \text{si } \mu_A(x_i) > \theta \\ 0 & \text{de cualquier otra forma} \end{cases} \quad (5,5)$$

En donde *Entrada_nueva_i* es la entrada correspondiente a *x_i* que se introducirá en la red. La entrada *x_i* se convierte a binario con base en su grado de membresía al conjunto difuso *A*, denominado $\mu_A(x_i)$, comparado con un umbral $\theta = 1/2$.

d) Como salida esperada debemos esperar que el nodo asociado con una categoría de salida específica tenga una activación mayor que el resto de los nodos de salida.

e) Entrenar la red con base en algún régimen de entrenamiento (y mediante la arquitectura interna que el mismo implique), para lograr que ésta sea capaz de asociar los patrones nuevos de entrada con su nodo de clasificación correspondiente en la salida.

ALGORITMO 13. Fase 2 del modelo de extracción BCD.

Una red neuronal creada bajo las restricciones que se establecen en el Algoritmo 13 tendría la arquitectura que se muestra en la ilustración 44.

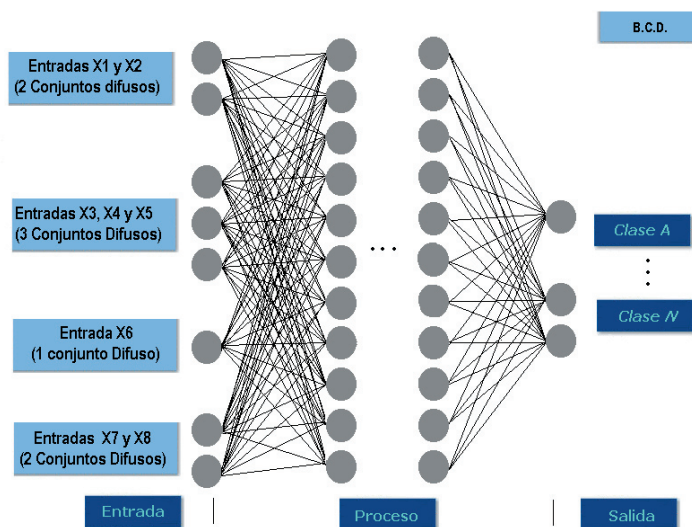


ILUSTRACIÓN 44. Red neuronal creada en la fase 2 del modelo BCD.

5.1.7. Procedimiento para la fase 3: Creación del conjunto de patrones

Esta fase tiene como objeto la creación del conjunto total de patrones de entrada, para lo cual se deben considerar las restricciones establecidas en el algoritmo 14. El conjunto total de patrones de entrada se establece a partir del conjunto total de datos de entrada creado en la fase 1.

7. Después de haber creado y entrenado la red neuronal para que sea capaz de clasificar los patrones nuevos construidos a partir de la binarización de los conjuntos difusos establecidos para cada variable de entrada del problema original, se crea un nuevo conjunto de patrones de entrada que contenga todas las posibles configuraciones de entrada que podríamos alimentar a la red:
 - a) Cada patrón posible deberá contener solamente un conjunto difuso activado (valor 1) para cada variable de entrada, mientras que

el resto de los conjuntos difusos de la misma variable deberán tener un valor de activación de 0. Es decir, un patrón posible de entrada no puede tener dos activaciones = 1 en conjuntos difusos de una misma variable. Esto debido a que BCD contempla que los conjuntos difusos ya binarizados establecidos para cada una de las variables son exclusivos entre sí.

- b) Por lo anterior, el número de casos a tratar como patrones posibles se reduce significativamente ya que conceptualmente se requiere de $C_{entrada_i} \times C_{entrada_{i+1}} \times \dots \times C_{entrada_n}$ configuraciones de entrada siendo $C_{entrada_i}$ el número de conjuntos difusos establecidos para la $entrada_i$.

ALGORITMO 14. Fase 3 del modelo de extracción BCD.

Se pueden remover algunas configuraciones si se reconocen como virtualmente imposibles de presentarse en la vida real, pero por supuesto que esta estrategia de reducción depende en gran medida del problema que se esté tratando de resolver y de la información extra disponible sobre el dominio del conocimiento.

5.1.8. Procedimiento para la fase 4: Evaluación y extracción inicial

Es en esta fase en la que la primera extracción o extracción en bruto del conjunto inicial de reglas tiene lugar. Por supuesto que para poder llegar a esta extracción inicial se debe procurar haber obtenido un desempeño aceptable en la clasificación de la red. Cada patrón del conjunto de datos de entrada se emplea como insumo para crear reglas con base en las salidas correspondientes en la red entrenada.

El procedimiento para la fase 4 se describe en el algoritmo 15.

8. El conjunto de los patrones posibles se introduce en la red neuronal creada para evaluar la salida que el sistema ofrece ante cada uno de ellos.
9. Los patrones se ordenan ahora de acuerdo a la clase que la red les asocia.
10. El conjunto de patrones posibles ordenados de acuerdo a la salida

con la que la red neuronal entrenada los asocia se convierte en el *Conjunto inicial de reglas extraídas*, procesando cada patrón para volverlo regla, de la siguiente manera: Para cada patrón del conjunto de patrones posibles ya evaluados por la red neuronal:

Representétese el patrón en la forma:

$$Si[x \text{ es } M]Y[y \text{ es } N]Y\dots Y[z \text{ es } O] \rightarrow Clase_i \quad (5,6)$$

En donde x, y, z , son las variables iniciales a partir de las cuales se crearon los conjuntos difusos y M, N, O son las etiquetas asignadas a los conjuntos difusos, cuyo valor ya binarizado es = 1. Finalmente $Clase_i$ es la etiqueta de la clase asociada al nodo i que se activa en la red ante la presentación de ese patrón.

ALGORITMO 15. Fase 4 del modelo de extracción BCD.

5.1.9. Procedimiento para la fase 5: Refinamiento del conjunto inicial de reglas

En esta fase el objetivo es lograr reglas máximamente generales para lo que se hace necesario un proceso de reducción o minimización lógica de cada conjunto de reglas, en el conjunto de reglas iniciales, asociadas con cada una de las clases.

El objetivo es lograr reglas conjuntivas con el formato siguiente:

$$Si[x \text{ [no]es } M[Y[N]*]]Y[y \text{ [no]es } O[Y[P]*]]Y\dots Y[z \text{ [no]es } Q[Y[R]*]] \rightarrow Clase_i \quad (5,7)$$

En donde x, y, z , son las variables iniciales a partir de las cuales se crearon los conjuntos difusos. M, N, O, P, Q, R son las etiquetas asignadas a los conjuntos difusos, cuyo valor ya binarizado es igual a uno (o igual a cero, en los casos en los que el uso de la negación lleva a una regla más general). Finalmente, $Clase_i$ es la etiqueta de la clase asociada al nodo i que se activa en la red ante la presencia de ese patrón.

Para esta fase del procedimiento se puede echar mano de cualquier herramienta de minimización lógica o método disponible: mapas de Karnaugh, métodos algebraicos o tablas de verdad, entre otros (Taha y Gosh, 1996b).

6

Evaluación del modelo BCD. Metodología empleada

6.1. Introducción

Este capítulo tiene por objeto presentar al lector un informe de las actividades realizadas al tratar de comprobar el valor de verdad de la hipótesis general de trabajo presentada en el Capítulo 1.

La realización del diseño del modelo BCD ha comprobado que si es posible diseñar un modelo de extracción de reglas para redes neuronales difusas (ver el Capítulo 5), por lo que esta parte de la hipótesis ha sido comprobada. Sin embargo, el trabajo no termina aquí, es necesario conocer el valor de verdad de los demás elementos de la hipótesis, por lo que se hizo necesario comprobar que:

- El modelo de extracción de reglas BCD es general y escalable.
- Los conjuntos de reglas extraídos por BCD son generales, modificables y fieles al conocimiento a partir del cual fueron creadas.
- Cada regla extraída por el modelo BCD es válida, comprensible y general.
- Comprobar que el modelo BCD muestra un desempeño general competitivo.

El trabajo sobre estas partes de la hipótesis general de trabajo implicó el diseño de una metodología de evaluación para probar o refutar estos elementos.

6.2. Metodología general de la evaluación

No se puede hablar de una metodología que asegure un proceso de evaluación ideal para modelos de extracción de reglas en redes neuronales difu-

sas. De hecho, es muy poco lo que actualmente se puede encontrar sobre metodologías de investigación en este campo, incluso, en el campo de las ciencias computacionales en general este tipo de información es escaso.

La evaluación de modelos computacionales se puede realizar mediante el uso de alguna de las dos estrategias posibles o su articulación (Cohen, 1995):

- Evaluación interna, que se realiza poniendo a prueba repetidas veces el modelo a fin de obtener información acerca de su confiabilidad. A esta estrategia de evaluación se le conoce también como evaluación con base en grupos de repeticiones de corridas.¹ Implica realizar un trabajo a nivel descriptivo simple: se utilizan medidas de tendencia central y de dispersión para poder dar cuenta del estado del modelo a lo largo de un grupo de exposiciones al problema en repetidas ocasiones.
- Evaluación externa, que puede realizarse en dos modalidades: i) prueba del modelo y comparación de los resultados obtenidos con un conjunto de estándares externos, y ii) prueba del modelo y comparación de los resultados obtenidos con los resultados de un modelo similar. Esta última modalidad es conocida también como *evaluación de pares*.

Sin embargo, cualquiera de estas dos estrategias y sus modalidades exige que antes se haya definido un conjunto de criterios a partir de los cuáles se realizará la evaluación. Si existen estándares externos para realizar el proceso de evaluación, entonces el trabajo se facilita, ya que sólo restaría usar los estándares para el establecimiento de criterios. En cambio, cuando los estándares externos no existen, el investigador se tiene que dar a la tarea de definir los criterios con base en las restricciones que los problemas reales, las teorías que respaldan a los modelos y la historia previa de modelos afines (Cohen, 1995).

En esta investigación se consideraron los siguientes aspectos para el establecimiento de los criterios de evaluación, dado que no existen estándares externos para realizar una comparación:

- Se busca que un modelo de extracción de reglas sea capaz de ofrecer reglas comprensibles, fieles, escalables, máximamente generales y modificables. Por lo que se buscó implementar estrategias que consideraran estas características como criterios de evaluación.

1 *Assessing Performance in Batches of Trials*, en inglés.

- Dado que en algunos de estos aspectos, deseables para las reglas, no se cuenta aún con índices de eficacia estandarizados; la medición se realizó con base en evaluaciones internas, combinadas con *evaluación de pares*. Para este tipo de evaluación se emplearon otros modelos de extracción de reglas en redes neuronales difusas.
- Aunque es en problemas de la “vida real” donde se puede tener una idea clara acerca de la eficacia del modelo después de medir su desempeño, es común que se someta a los modelos evaluados a problemas “de laboratorio”. Este tipo de problemas ofrece la ventaja de haber sido ampliamente estudiados y ser bien conocidos, posibilitando un punto de referencia para establecer una evaluación más confiable.

Así, se definió una estrategia de evaluación compuesta por procesos de evaluación interna, cuyos resultados fueron usados en evaluación de pares. Las pruebas se realizaron durante el sometimiento de los tres modelos a la resolución de un problema de laboratorio. La estructura general del trabajo realizado en la fase de prueba se muestra en la ilustración 45.

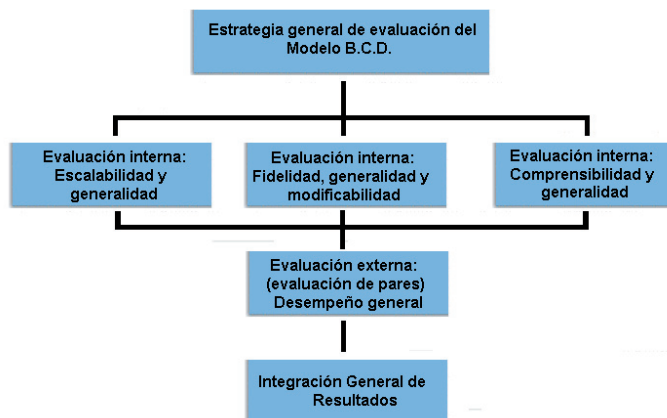


ILUSTRACIÓN 45. Estrategia general empleada para la evaluación del modelo BCD.

6.3. Metodología diseñada para Comprobar que el modelo de extracción de reglas bcd es general y escalable

Para comprobar que BCD es un modelo general y escalable, la estrategia empleada se dividió en dos fases: i) evaluar la escalabilidad del modelo, y ii) evaluar la generalidad del modelo.

6.3.1. Evaluación de la escalabilidad del modelo

El aspecto a considerar en este parámetro es la posibilidad de que el modelo pueda ser aplicado a redes neuronales difusas con gran número de unidades de entrada y unidades en las demás capas de la red.

Al evaluar la escalabilidad del Modelo BCD se consideraron no sólo las posibilidades teóricas, sino las posibilidades reales o técnicas de poder aplicar el modelo a mayores escalas. Este segundo aspecto depende de los recursos necesarios para realizar la tarea.

6.3.2. Evaluación de la generalidad del modelo

Se debe considerar la independencia del modelo con respecto a la topología de la red en la que se utiliza para aplicar el modelo, los regímenes de entrenamiento de la misma y las funciones de activación utilizadas.

Esta evaluación atiende a uno de los problemas principales en redes neuronales: la localización de los métodos o también como es conocida: la exclusividad de la solución a modelos específicos de red o tipologías específicas en un modelo de red. Es por ello que es de gran importancia conocer el grado de portabilidad o generalidad de este modelo a otras topologías.

6.4. Metodología para Comprobar que los conjuntos de reglas extraídos por bcd son generales, modificables y fieles al conocimiento a partir del cual fueron creados

Para evaluar estas características generales de las bases de reglas extraídas por el Modelo BCD, se emplearon los siguientes criterios: fidelidad, generalidad y modificabilidad, de cada conjunto de reglas extraído.

6.4.1. Evaluación de la fidelidad de los conjuntos de reglas extraídos

Para evaluar la fidelidad de los conjuntos de reglas extraídos se consideraron los elementos propuestos en la metodología de evaluación de bases de reglas extraídas, propuesta por Ismail Taha y Joydeep Ghosh (Taha y Ghosh, 1996a). Estos dos autores desarrollaron un procedimiento simple de evaluación de reglas que usa a la misma red neuronal entrenada para valorar tres aspectos en cada una de las reglas obtenidas:

- a) Número medio de los disparos correctos
Evaluar el número de veces en que una regla es disparada por el sistema en forma correcta. Con esta evaluación se puede tener una idea certera de la pertinencia de una regla para interpretar correctamente la salida que se obtendría en una red entrenada en el mismo dominio.
- b) Número de medio disparos en falso
Medición de disparos en falso, que permiten evaluar cuantas veces una regla puede llevar a una conclusión errónea (distinta a la que la misma red neuronal entrenada emitiría).
- c) Exclusividad media en los conjuntos de reglas extraídos

Evaluar el grado de exclusividad de las reglas lleva a conocer la generalidad de las mismas. Se debe entender por exclusividad de una regla, la medición del número de casos que implican el disparo de dos o más reglas al mismo tiempo; un caso que implica el disparo de más de una regla pone en evidencia a dos reglas de menor exclusividad.

6.4.2. Evaluación de la generalidad media de los conjuntos de reglas extraídos

Para evaluar la generalidad de los conjuntos de reglas, se implementó la siguiente estrategia:

- a) Conocer el número medio de casos posibles no cubiertos por las reglas.
El número de casos posibles de configuración de entrada no cubiertos en una base de reglas, es un indicador acerca de la generali-

dad de la misma, al dar cuenta de su alcance y efectividad frente a la presentación de casos nuevos. El ideal de cobertura que se propone en este documento es el 100.00% de los casos posibles cubiertos. Sin embargo, no se descarta el hecho de que puedan existir casos reales en los que configuraciones de entrada que se marcan como posibles, no se puedan presentar en el mundo real. Es por ello que se debe hacer la aclaración de que se busca una cobertura del 100.00% del total de casos posibles y viables.

En los casos en los que el número de configuraciones posibles no coincide con el número real de configuraciones que se pueden presentar, se hace necesario el uso de reglas de corte de rama o de limitación del espacio en el dominio. Es por ello que este tipo de reglas se consideran también como un factor a evaluar en este rubro.

b) Existencia de regla de corte de rama.

En problemas reales no siempre está disponible toda la información necesaria para poder realizar una *clasificación completa*, es decir, una clasificación en la que todas las clases estén contempladas. Es por ello que generalmente se usa un conjunto especial que contiene a *todo lo demás*, y reglas que permiten establecer diferencias entre este conjunto y las clases con las cuales realmente trabaja el sistema. Este tipo de reglas, que delimitan el espacio de búsqueda, representa un aspecto importante a considerar para evaluar la generalidad de una base de reglas, ya que son realmente éstas, los límites de la generalidad.

6.4.3. Evaluación de la modificabilidad de los conjuntos de reglas extraídos

Para evaluar este parámetro se debe considerar la capacidad del modelo para permitir la modificación de las reglas extraídas a partir de cambios subsecuentes en el dominio del conocimiento o en la estructura de la red que los soporta.

6.5. Metodología para Comprobar que cada regla extraída por el modelo bcd es válida, comprensible y general

6.5.1. Evaluación de la comprensibilidad de cada una de las reglas extraídas

Para la evaluación de la comprensibilidad de las reglas extraídas se consideraron los siguientes factores en cada modelo: el número de reglas extraídas, el número de premisas en cada regla y la forma en que la regla extraída es presentada al usuario.

6.5.2. Evaluación de la generalidad de cada una de las reglas extraídas

Se debe considerar que el objetivo es lograr cubrir el mayor número de casos posible con cada una de las reglas extraídas por el modelo.

6.6. Metodología para Comprobar que el modelo BCD muestra un desempeño general competitivo

Para tratar de demostrar que el modelo BCD muestra un desempeño general competitivo se diseñó una metodología de trabajo tipo evaluación de pares con base la comparación de medias (Cohen, 1995). La metodología diseñada implicó cuatro fases:

- i) Selección de los modelos de prueba,
- ii) Prueba de los modelos con un problema de laboratorio,
- iii) Evaluación de los desempeños y,
- iv) Comparación y análisis de los resultados.

En la ilustración 46 se presenta un esquema del proceso y los pasos seguidos se detallan en los apartados siguientes.

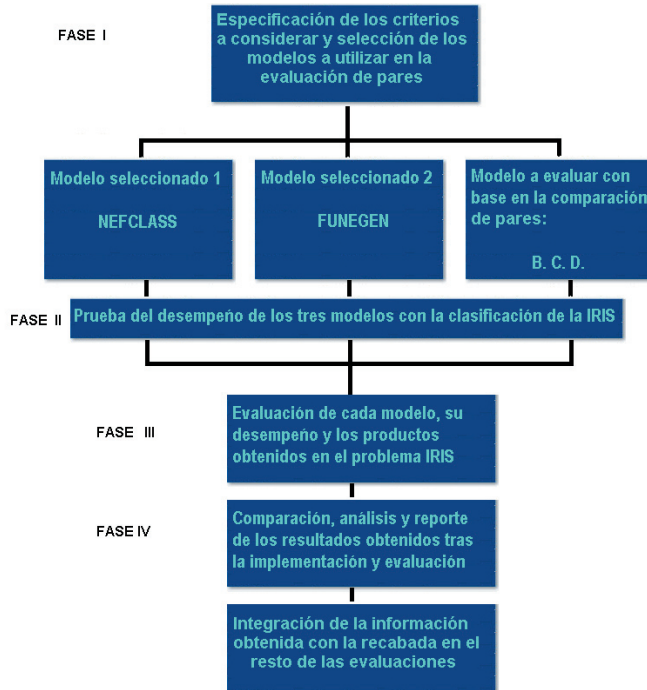


ILUSTRACIÓN 46. Estrategia empleada para la evaluación del modelo BCD.

6.6.1. Fase I: Selección

Selección de los modelos *pares* para el proceso de evaluación. En esta fase se llevó a cabo una evaluación de los modelos existentes en la actualidad que son capaces de extraer reglas de redes neuronales Difusas. A partir del conocimiento de los modelos actuales, se seleccionó:

- El Modelo FUNEGEN de Halgamuge y Glesner (Halgamuge y Glesner, 1994). Las razones para seleccionar este modelo fueron: su actualidad, sus características específicas que le posibilitan la extracción completa de reglas a partir de cero, a diferencia de otros modelos que solamente refinan conjuntos preestablecido de reglas iniciales y la importancia que ha tenido en la actualidad como el trabajo más reciente en extracción de reglas de redes neuronales difusas, y
- El Modelo NEFLCLASS de Nauck y Kruse (Nauck y Kruse, 1995; Nauck,

Klawonn y Kruse, 1995; Nauck, Nauck y Kruse, 1996; y Nauck 2003 y 2004). Que fue elegido por ser actual, representativo del *otro* enfoque de extracción de reglas basado en el agrupamiento difuso, y por ser el modelo más sobresaliente de otra de las líneas de investigación más importantes en extracción de reglas de redes neuronales difusas.

6.6.2. Fase II: Implementación y Prueba

En esta fase, tanto los modelos pares, como el modelo BCD, se sometieron a la resolución del problema de laboratorio. El problema seleccionado fue la clasificación de la planta *iris* propuesto por Fisher (ver el apartado de anexos).

6.6.3 Fase III: Evaluación de cada modelo

En esta fase se llevó a cabo la evaluación del proceso de extracción de las reglas, que cada modelo realizó para resolver el problema de la clasificación de la planta iris. También en esta fase se evaluaron los resultados obtenidos por cada modelo. La evaluación se llevó a cabo de acuerdo a tres dimensiones:

- Evaluación de las características relativas al modelo de extracción.
- Evaluación de las características de cada una de las reglas extraídas.
- Evaluación de los conjuntos de reglas extraídas.
- Evaluación de la capacidad de BCD para llegar a los mismos resultados que los otros modelos en igualdad de condiciones.

Cada modelo de extracción de reglas en redes neuronales difusas, usa sus propios cálculos y criterios para la conformación de sus funciones de membresía. Es por ello que se incluyó esta última dimensión con el afán de conocer cómo se comportaría el modelo BCD usando las mismas funciones que cada modelo par seleccionado.

6.6.3.1. Métodos empleados para la evaluación de las características relativas al modelo de extracción

6.6.3.1.1. Evaluación del desempeño

La evaluación del desempeño de cada modelo durante la fase de prueba, se realizó con base en tres características: Porcentaje de clasificación media, número de ciclos de entrenamiento usados para lograr la clasificación correcta y refinamiento de las funciones de membresía. Dado que los tres modelos implican en parte de su procedimiento entrenamiento de una red neuronal y este proceso es no determinístico, se decidió hacer varias corridas de cada modelo, de manera tal que se pudiesen tener indicadores medios acerca de su desempeño. El número necesario de corridas no podía haberse establecido *a priori* desde el inicio de la fase de prueba, sino que fue establecido con base en la dispersión de los datos arrojados en cada prueba (Cohen, 1995).

A continuación se describe cada una de las medidas empleadas:

a) Porcentaje de Clasificación Media

Se calculó la media aritmética de los porcentajes de efectividad en la clasificación lograda por cada modelo en cada una de las corridas. Esta medida da cuenta de la efectividad del modelo en la clasificación alcanzada por cada modelo en cada una de las corridas realizadas.

b) Número medio de ciclos de entrenamiento necesarios para el entrenamiento

El número de ciclos de entrenamiento necesarios es el número total de veces que un mismo patrón es alimentado a la red para su clasificación y posterior extracción de reglas. El número de ciclos necesarios para clasificación también ayuda en la evaluación de la efectividad del modelo, asumiéndose que mientras menos ciclos sean necesarios para lograr el objetivo, más eficiente puede ser el modelo de extracción de reglas que está siendo evaluado.

c) Refinamiento de las funciones de membresía

Los modelos evaluados realizan el establecimiento de funciones de membresía *a priori*, para después refinarlas con base en los mismos datos de entrada que están procesando. La evaluación de la efectividad antes y después de la readecuación de las funciones de membresía, también aporta elementos para la evaluación del desempeño general del modelo.

6.6.3.1.2. Evaluación de la escalabilidad del modelo

Este criterio ha sido definido ya en el apartado 6.3.1. de este mismo capítulo y en este caso se evaluó en cada uno de los tres modelos: FUNEGEN, NEFLCLASS y BCD.

6.6.3.1.3 Evaluación de la generalidad del modelo

Igual que el criterio anterior, la generalidad del modelo se ha sido definido ya en el apartado 6.3.2. de este mismo capítulo y en este caso, igualmente, se evaluó en cada uno de los tres modelos: FUNEGEN, NEFLCLASS y BCD.

6.6.3.2. Métodos empleados para la evaluación de las características de cada una de las reglas extraídas

6.6.3.2.1. Evaluación de la comprensibilidad de cada una de las reglas extraídas

Este criterio se ha definido ya en el apartado 6.5.1. de este Capítulo.

6.6.3.2.2. Evaluación de la generalidad de cada una de las reglas extraídas

Igual a la definición establecida en el apartado 6.5.2. de este mismo Capítulo.

6.6.3.3. Métodos empleados para la evaluación de las características de los conjuntos de reglas extraídos

Para evaluar las características generales de las bases de reglas extraídas por cada uno de los modelos, se emplearon los siguientes criterios: fidelidad, generalidad y modificabilidad, de cada conjunto de reglas extraído. No se explica en este apartado cada uno de estos criterios debido a que

fueron ya definidos en los apartados 6.4.1. , 6.4.2. y 6.4.3. de este mismo Capítulo.

6.6.3.4. Método empleado para la evaluar la capacidad de BCD para llegar a los mismos resultados que los otros modelos en igualdad de condiciones

Dado que los tres modelos a comparar emplean funciones de membresía distintas para las variables de entrada de problema de clasificación de la planta iris, se realizó la implementación del modelo BCD con las funciones de membresía que éste propone y se evaluaron su desempeño, características y producto final. Sin embargo, se consideró necesario, también, evaluar hasta qué grado es capaz BCD, de llegar a los mismos resultados que los otros dos modelos (NEFCLASS y FUNEGEN) usando las mismas funciones de membresía que éstos. Para ello, se calcularon las funciones de membresía empleadas por cada uno de estos modelos y se realizó la extracción de reglas bajo estas especificaciones, pero conforme al procedimiento establecido por BCD.

Esta prueba se propuso con una doble finalidad:

- a) Evaluar el desempeño del modelo de extracción BCD, comparándolo también con los otros modelos, bajo un problema con los mismos parámetros de membresía difusos
- b) Aportar un elemento más para la validación del conjunto de reglas extraídas por BCD, mediante esta comparación.

6.6.4. Fase IV Comparación de los resultados y análisis de los mismos

La comparación realizada entre los resultados obtenidos por cada uno de los tres modelos, permitió por un lado, conocer las ventajas y desventajas que el modelo de extracción de reglas difusas BCD presenta frente a otros modelos que tienen el mismo propósito, es decir, la extracción de reglas difusas.

El análisis y comparación de los resultados obtenidos con el problema de clasificación de la planta *iris* permitió ubicar el desempeño del modelo en la escala de desempeño de los extractores de reglas. La amplia difusión que ha tenido el problema de clasificación de la planta *iris*, posibilitó

tener acceso a una gran cantidad de literatura en la que se analiza el problema. Esto permitió saber exactamente qué debía esperarse del BCD y qué aspectos debían considerarse como indispensables en el modelo.

Finalmente, la información obtenida en esta prueba se integró a las otras evaluaciones a fin de componer el informe de investigación que el lector tiene en sus manos.

Evaluación del modelo BCD. pruebas y resultados obtenidos

El presente capítulo se divide en dos partes generales: la primera contiene los resultados de la aplicación de cada uno de los tres modelos NEFLCLASS, FUNEGEN y BCD para resolver el problema de clasificación de la planta *iris*; la segunda parte se estructura a partir de los resultados de la evaluación de cada uno de los aspectos establecidos en el Capítulo 6.

7.1. Aplicación de los tres modelos al problema de clasificación de la planta *iris*

En los apartados siguientes se presenta al lector la forma en la que los tres modelos a evaluar en esta fase del trabajo, resolvieron el problema de clasificación de la planta *iris*. Dado que el trabajo con redes neuronales implica el acercamiento de funciones en una forma no-determinista, es decir, existen n soluciones al problema y todas ellas efectivas.

7.1.1. NEFLCLASS y la Clasificación de la planta *iris*

Para el proceso de clasificación con el NEFLCLASS, el conjunto de datos fue dividido en dos partes iguales (se asignaron sus elementos al azar). Los conjuntos establecidos se usaron, uno como *conjunto de datos de entrenamiento* y el otro como *conjunto de datos de prueba*. Se estableció un tope arbitrario al sistema, de siete reglas máximas (de cualquier manera, si el sistema hubiese necesitado más reglas para lograr una clasificación correcta, las hubiese creado. Por lo que este parámetro debe ser considerado solamente como un requisito inicial).

El modelo fue probado 25 veces, con la finalidad de obtener indicadores más certeros a cerca de su comportamiento. Una de las 25 bases de

reglas fue seleccionada al azar para realizar las evaluaciones cualitativas de las reglas, individualmente, y del conjunto, en general.

La base de reglas usada para la evaluación se resultados obtenidos tras la aplicación de NEFCLASS al problema de clasificación de la planta *iris* se presentan en la tabla 2.

TABLA 2. Reglas difusas extraídas por NEFCLASS.

No	Regla
1	SI V1 es Pequeño Y V2 es Medio Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1
2	SI V1 es Pequeño Y V2 es Pequeño Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1
3	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Pequeño, Entonces: Clase 2
4	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Medio, Entonces: Clase 2
5	SI V1 es Medio Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3
6	SI V1 es Grande Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3
7	SI V1 es Grande Y V2 es Medio Y V3 es Grande Y V4 es Grande, Entonces: Clase 3

V1 = largo de tallo, V2 = Ancho de tallo, V3 = Largo de pétalo, V4 = Ancho de pétalo
Clase 1 = *Iris Setosa*, Clase 2 = *Iris Versicolor*, Clase 3 = *Iris Virginica*

La ilustración 47 muestra la arquitectura usada por NEFCLASS para la resolución del problema de clasificación de la planta *iris*. Inicialmente la red estaba compuesta por un conjunto de 19 reglas, de las cuales, después del proceso de optimización que el mismo modelo contempla, quedaron solamente las 7 unidades de regla que la ilustración presenta.

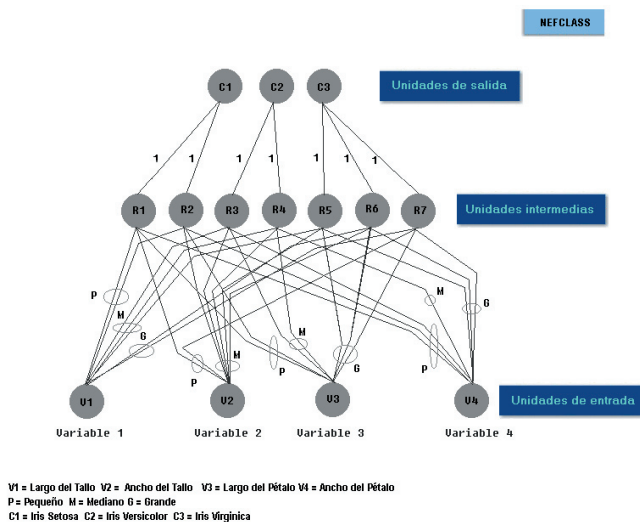


ILUSTRACIÓN 47. Arquitectura optimada usada por NEFCLASS.

Inicialmente la red comenzó sin unidades de regla, pero fueron creadas las dos primeras unidades para poder manejar la clase 1, es decir, la más fácil. Fueron creadas también dos unidades más para la clasificación correcta de la segunda clase y tres para la tercera. Después de la creación de las unidades de regla, las funciones de membresía originales fueron optimizadas para reducir el número de errores de clasificación.

A continuación se presentan las funciones de membresía calculadas por el NEFCLASS para la clasificación del problema *iris*. En las cuatro variables que se presentan en las ilustraciones 48, 49, 50 y 51, el sistema inició con tres conjuntos difusos distribuidos. Sin embargo, a lo largo del proceso de clasificación los conjuntos sufrieron adecuaciones para incrementar el desempeño del sistema.

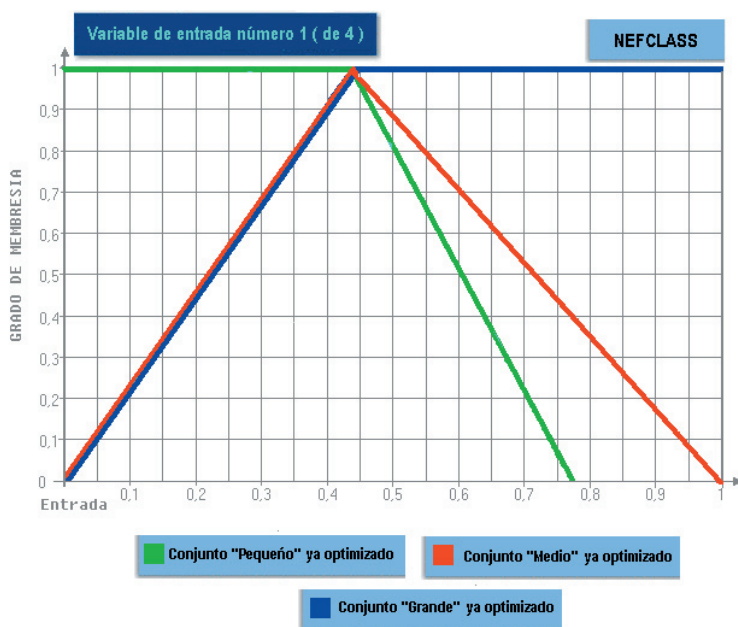


ILUSTRACIÓN 48. Funciones de membresía usadas por NEFCLASS para la variable *largo del tallo*.

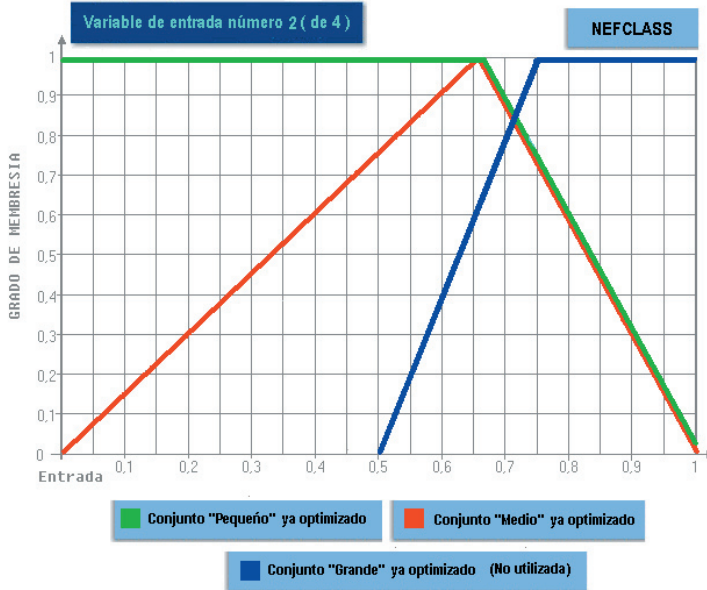


ILUSTRACIÓN 49. Funciones de membresía usadas por NEFCLASS para la variable *Ancho del tallo*.

En las dos primeras variables relacionadas con las características del tallo de la planta iris (largo y ancho), después de la optimización de los conjuntos difusos, los conjuntos etiquetados como *medio* resultaron poco útiles en el proceso de clasificación. Es por ello que la distribución de los conjuntos en las ilustraciones 48 y 49 los muestran muy cercanos a los conjuntos difusos de los extremos.

Por el contrario, en el caso de las otras dos variables, el largo y el ancho del pétalo de la planta iris, el conjunto etiquetado como *medio*, adquiere mayor significatividad en el proceso de clasificación.

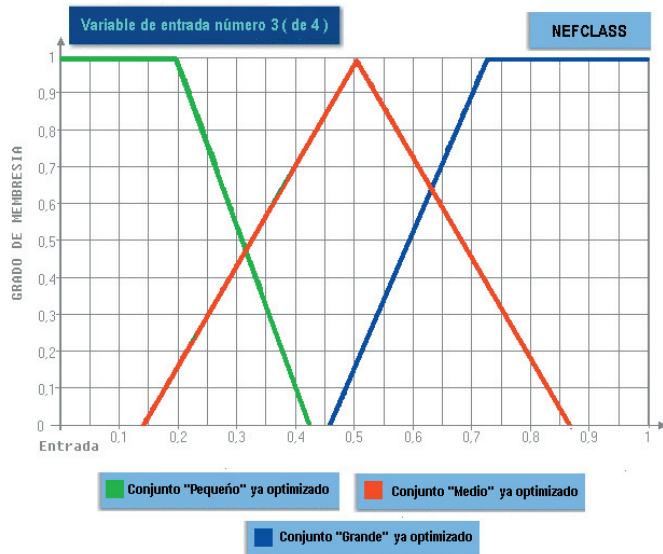


ILUSTRACIÓN 50. Funciones de membresía usadas por NEFCLASS para la variable *Largo del pétalo*.

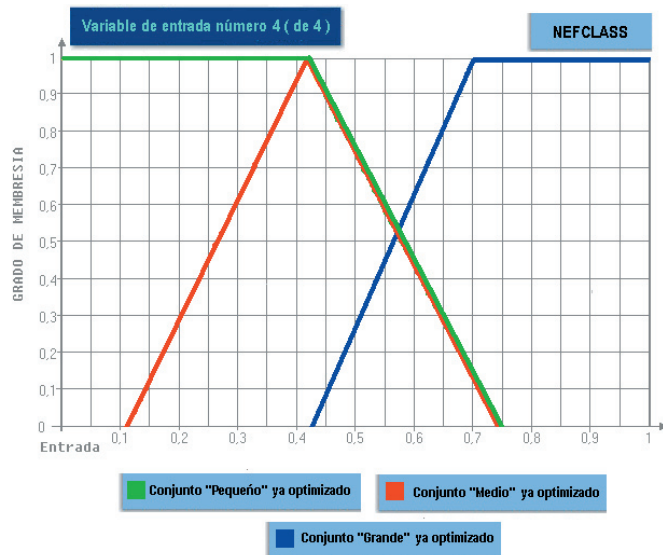


ILUSTRACIÓN 51. Funciones de membresía usadas por NEFCLASS para la variable *Ancho del pétalo*.

7.1.2. FUNEGEN y la clasificación de la planta *iris*

Para la aplicación del modelo FUNEGEN al problema de clasificación de la planta *iris*, se dividió el conjunto de datos completo en dos conjuntos llamados *conjunto de entrenamiento* y *conjunto de prueba*, cada uno de los cuales contenía 75 vectores. La asignación de los vectores a cada uno de los grupos se realizó al azar. El sistema que implementó el modelo FUNEGEN se entrenó con el conjunto de vectores de entrenamiento y su funcionamiento fue probado con el *conjunto de prueba*. Cada fase, tanto la de entrenamiento, como de prueba, se realizaron en 500 ciclos.

Las reglas extraídas por FUNEGEN durante la primera fase del proceso se muestran en la tabla 3.

TABLA 3. Reglas extraídas por FUNEGEN.

No.	Reglas
1	Sí I2 mf0 (Pequeño), Entonces: Salida 0
2	Sí I2 mf2 (Medio), Y I3 mf2 (Medio), Entonces: Salida 1
3	Sí I2 mf4 (Grande), Entonces: Salida 2
4	Sí I2 mf2 (Medio), Y I3 mf2(Medio), Entonces: No salida 2
5	Sí I3 mf4 (Grande), Entonces: No Salida 1
La Salida 0 corresponde a la planta iris <i>Setosa</i> , Salida 1 a la planta iris <i>Versicolor</i> , Salida 2 a la planta iris <i>Virginica</i>	
Las funciones de membresía mfX adjunto muestran la etiqueta dada al conjunto difuso. ¹	

Conviene resaltar que existen dos pares de reglas que son complementarias: a) la regla 4 es complemento de la regla 2, por lo que se podría considerar como redundante, dado que si en la regla 2 ya se establecen las condiciones para concluir que se trata de la “Salida 1” y solamente ésta; entonces la regla 4 es obvia. De igual manera, sería obvia la regla: “Sí I2 mf2 (Medio), Y I3 mf2 (Medio), Entonces: No salida 0”. Sin embargo, esta regla no aparece en el conjunto final de reglas extraídas. Para las reglas 3 y 5 sucede lo mismo, la regla 5 es obvia, por lo establecido en la regla 3.

La arquitectura inicial estuvo compuesta por 13 unidades de regla, es decir, por medio de 13 reglas (ver las ilustraciones 52 y 53). Después de esta fase, se realizó un proceso de reducción de la base hasta llegar a 5 reglas y 2.

Después del proceso de reducción de la base las reglas extraídas no

1 Nota: La etiqueta lingüística colocada entre paréntesis () en las premisas de las reglas no son parte del resultado original, sólo se colocaron en este trabajo solamente con el fin de facilitar la comprensión de las reglas al lector.

sufrieron modificación alguna, sin embargo, las funciones de membresía para los conjuntos difusos si se adecuaron para lograr un porcentaje más alto de clasificación de vectores.

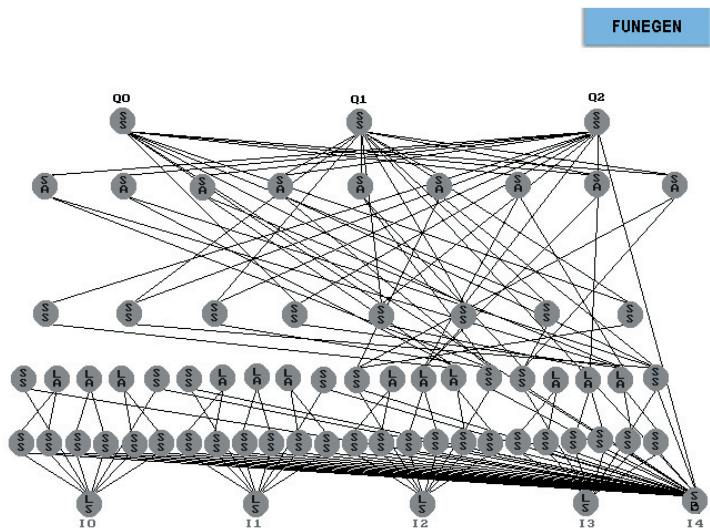


ILUSTRACIÓN 52. Configuración inicial de la red usada por FUNEEN.

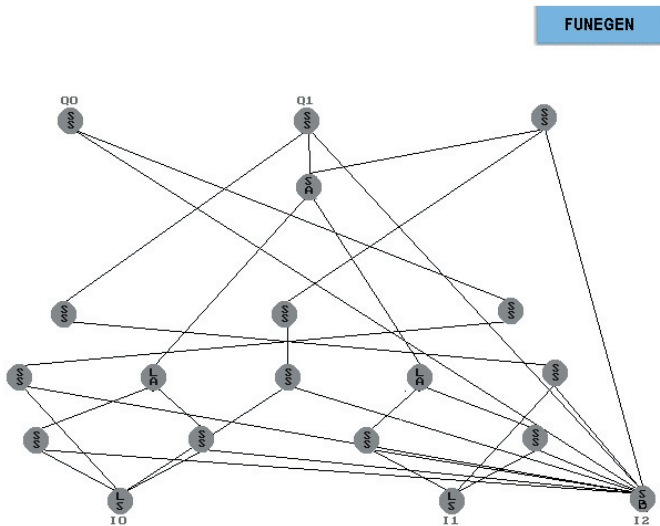


ILUSTRACIÓN 53. Configuración optimada en la primera fase por FUNEEN.

Las ilustraciones 54, 55, 56 y 57 presentan las funciones de membresía calculadas por FUNEGEN para el problema de clasificación de la planta iris.

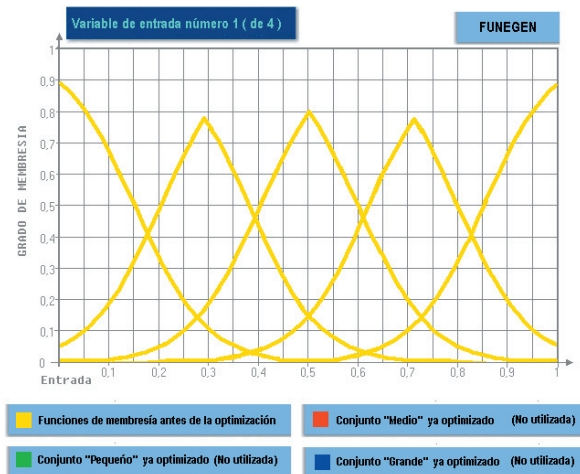


ILUSTRACIÓN 54. Funciones de membresía usadas por FUNEGEN para la variable *Largo del tallo*.

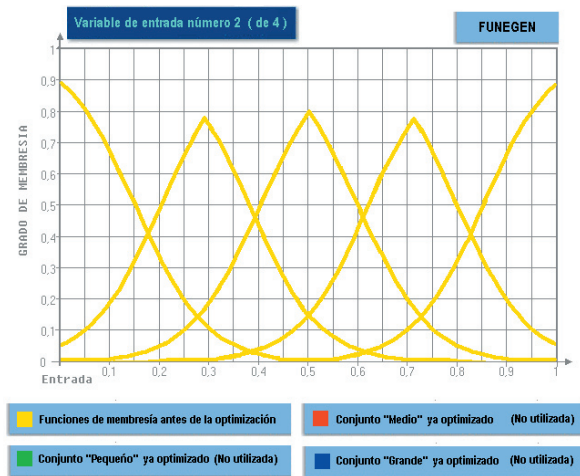


ILUSTRACIÓN 55. Funciones de membresía usadas por FUNEGEN para la variable *Ancho del tallo*.

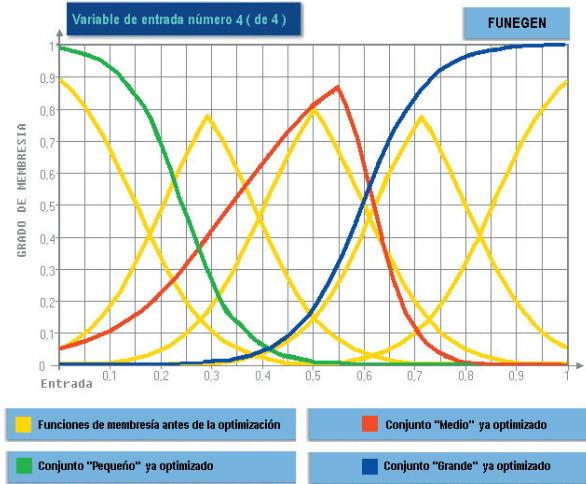


ILUSTRACIÓN 56. Funciones de membresía usadas por FUNEGEN para la variable *largo del pétalo*.

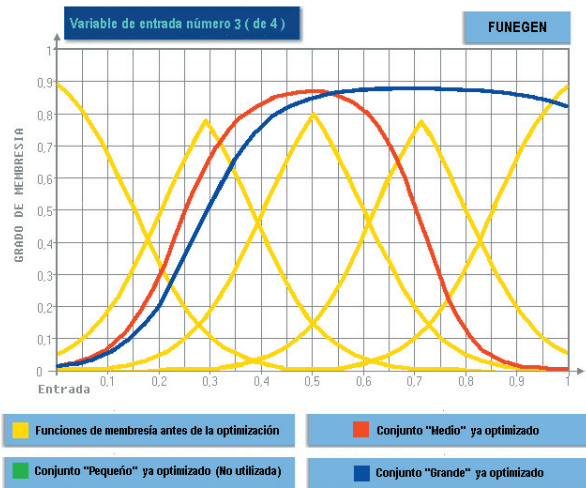


ILUSTRACIÓN 57. Funciones de membresía usadas por FUNEGEN para la variable *Ancho del pétalo*.

7.1.3. BCD y la clasificación de la planta iris

La implementación del modelo de extracción de reglas BCD, aplicado en la extracción de reglas para clasificar a la planta *iris*, se inició estableciendo tres conjuntos difusos iguales: *pequeño*, *medio*, y *grande* para las cuatro variables de entrada que plantea el problema original: V1 (Largo del tallo de la planta *iris*), V2 (Ancho del tallo de la planta *iris*), V3 (Largo del pétalo de la flor de la planta *iris*) y V4 (Ancho del pétalo de la flor de la planta *iris*). Los valores y rangos iniciales se muestran en la tabla 4.

TABLA 4. Límites iniciales para las variables establecidos por BCD.

Variable	Mínimo	Máximo	Valor	Límites
V1 Largo del tallo	4.3	7.9	Pequeño	[4.3, 5.5]
			Medio	(5.5, 6.8]
			Grande	(6.8, 7.9]
V2 Ancho del tallo	2.0	4.4	Pequeño	[2.0, 2.7]
			Medio	(2.7, 3.6]
			Grande	(3.6, 4.4]
V3 Largo del pétalo	1.0	6.9	Pequeño	[1.0, 2.9]
			Medio	(2.9, 4.9]
			Grande	(4.9, 6.9]
V4 Ancho del pétalo	0.1	2.5	Pequeño	[0.1, 0.8]
			Medio	(0.8, 1.7]
			Grande	(1.7, 2.5]

Con base en las divisiones iniciales a las variables de la tabla 5, fueron calculadas las siguientes funciones de membresía, aplicables inicialmente a cada una de las características de la planta *iris*, que el conjunto de datos del problema presenta como entrada:

$$\mu_{\text{pequeño_inicial}(x)} \begin{cases} 1 & \text{si } 0 \leq x \leq \frac{1}{6} \\ 1.5 - 3x & \text{si } \frac{1}{6} < x \leq \frac{1}{2} \\ 0 & \text{si } \frac{1}{2} < x \leq 1 \end{cases} \quad (6,4)$$

$$\mu_{\text{medio_inicial}(x)} \begin{cases} 0 & \text{si } 0 \leq x \leq \frac{1}{6} \\ 1 - 3|x - \frac{1}{2}| & \text{si } \frac{1}{6} < x \leq \frac{5}{6} \\ 0 & \text{si } \frac{5}{6} < x \leq 1 \end{cases} \quad (6,5)$$

$$\mu_{\text{grande_inicial}}(x) \begin{cases} 0 & \text{si } 0 \leq x \leq 1/2 \\ 3x - 1.5 & \text{si } 1/2 < x \leq 5/6 \\ 1 & \text{si } 5/6 < x \leq 1 \end{cases} \quad (6,6)$$

Después de la definición de estas funciones de membresía se procedió a afinarlas de acuerdo al procedimiento siguiente tomado de (Von Altrock, 1995) y modificado con base en la propuesta publicada en (Keller y Hunt, 1992):

1. Calcular las funciones de membresía (tipo *Lambda*) distribuidas equitativamente a lo largo del conjunto de datos.
2. Calcular para cada conjunto establecido la media aritmética de todos los datos que quedan integrados a este conjunto.
3. Recalcular las nuevas funciones de membresía con base en las medias aritméticas de los conjuntos iniciales en la siguiente forma:
 - a) El valor de membresía debe ser igual a 1 si el valor del dato que se analiza es igual a la media de su clase;
 - b) El valor de membresía debe ser igual a 0 si el valor del dato que se analiza es igual a la media de las clases contiguas;
 - c) Conforme un valor se aleja de la media de la clase contigua y se acerca a la media de su clase, su grado de membresía se acerca exponencialmente a 1.
 - d) Conforme un valor se aleja de la media de su clase y se acerca a la media de una clase contigua, su grado de membresía se aleja exponencialmente a 0.
 - e) Hasta este paso se tiene el calculo de las funciones de membresía para los términos intermedios, si se trata de calcular las funciones de membresía de términos extremos a la derecha, a toda la región más allá de la media se le asigna un grado de membresía de 1, y de igual manera se procede para calcular la función de membresía de un término extremo izquierdo, en el que a toda la región que queda por debajo de la media se le asigna también un grado de membresía de 1.
 - f) Finalmente se calculan las ecuaciones pertinentes para representar efectivamente las funciones establecidas en los pasos establecidos en el inciso a) al inciso e).

Las funciones de membresía resultantes después del proceso de optimización se muestran a continuación:

Para la variable 1 (Largo del tallo de la planta *iris*):

$$\mu_{\text{pequeño_Variable_1}(i)} \begin{cases} 1 & \text{si } 0 \leq x \leq .1996 \\ -3.4399725x + 1.6866185 & \text{si } .1996 < x \leq .4903 \\ 0 & \text{si } .4903 < x \leq 1 \end{cases} \quad (6,7)$$

$$\mu_{\text{medio_Variable_1}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .1996 \\ 3.4399725x - .6866185 & \text{si } .1996 < x \leq .4903 \\ -3.539823x + 2.7355752 & \text{si } .4903 < x \leq .7728 \\ 0 & \text{si } .7728 < x \leq 1 \end{cases} \quad (6,8)$$

$$\mu_{\text{grande_Variable_1}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .4903 \\ 3.539823x - 1.7355752 & \text{si } .4903 < x \leq .7728 \\ 1 & \text{si } .7728 < x \leq 1 \end{cases} \quad (6,9)$$

La ilustración 59 incluye tanto las funciones de membresía que inicialmente usó BCD para la variable 1 (largo del tallo de la planta *iris*), como las funciones de membresía optimizadas:

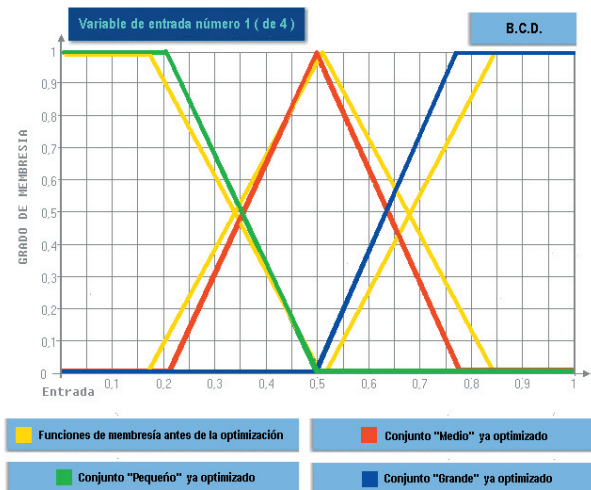


ILUSTRACIÓN 58. Funciones de membresía usadas por BCD para la variable *Largo del tallo*.

Para la variable 2 (Ancho del tallo de la planta *iris*) las funciones de membresía iniciales se modificaron para quedar de la siguiente manera:

$$\mu_{\text{pequeño_Variable_2}(i)} \begin{cases} 1 & \text{si } 0 \leq x \leq .1667 \\ -3.2175032x + 1.5363578 & \text{si } .1667 < x \leq .4775 \\ 0 & \text{si } .4775 < x \leq 1 \end{cases} \quad (6,10)$$

$$\mu_{\text{medio_Variable_2}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .1667 \\ 3.2175032x - .5363577 & \text{si } .1667 < x \leq .4775 \\ -3.4364261x + 2.6408935 & \text{si } .4775 < x \leq .7685 \\ 0 & \text{si } .7685 < x \leq 1 \end{cases} \quad (6,11)$$

$$\mu_{\text{grande_Variable_2}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .4775 \\ 3.4364261x - 1.6408935 & \text{si } .4775 < x \leq .7685 \\ 1 & \text{si } .7685 < x \leq 1 \end{cases} \quad (6,12)$$

En la ilustración 59 se muestra la adecuación que empleó BCD para las funciones de membresía iniciales en la variable 2 (ancho del tallo de la planta *iris*):

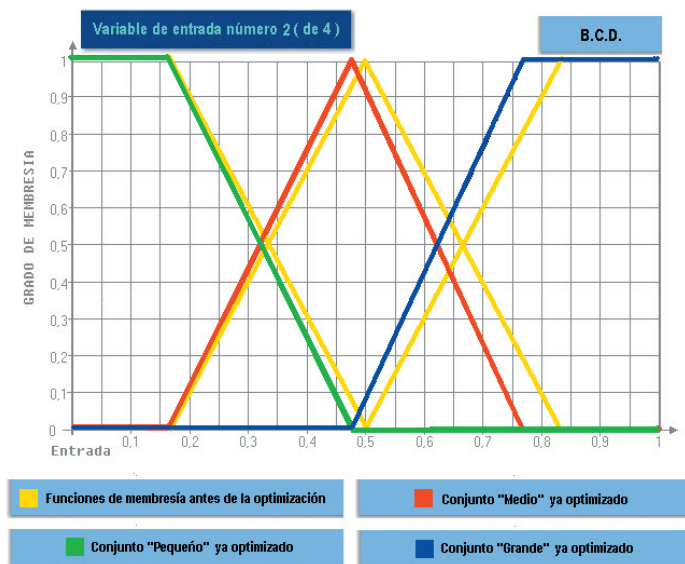


ILUSTRACIÓN 59. Funciones de membresía usadas por BCD para la variable *Ancho del tallo*.

En el caso de la variable 3 (Largo del pétalo de la flor de la planta *iris*) las adecuaciones a las funciones de membresía originales fueron las siguientes:

$$\mu_{\text{pequeño_Variable_3}(i)} \begin{cases} 1 & \text{si } 0 \leq x \leq .0763 \\ -2.0768432x + 1.1584631 & \text{si } .0763 < x \leq .5578 \\ 0 & \text{si } .5578 < x \leq 1 \end{cases} \quad (6,13)$$

$$\mu_{\text{medio_Variable_3}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .0763 \\ 2.0768432x - .1584631 & \text{si } .0763 < x \leq .5578 \\ -4.411116x + 3.4605205 & \text{si } .5578 < x \leq .7845 \\ 0 & \text{si } .7845 < x \leq 1 \end{cases} \quad (6,14)$$

$$\mu_{\text{grande_Variable_3}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .5578 \\ 4.411116x - 2.4605205 & \text{si } .5578 < x \leq .7845 \\ 1 & \text{si } .7845 < x \leq 1 \end{cases} \quad (6,15)$$

En la ilustración 60 se pueden apreciar las modificaciones que BCD realizó a las funciones de membresía iniciales en la variable 3 (largo del pétalo de la flor de la planta *iris*):

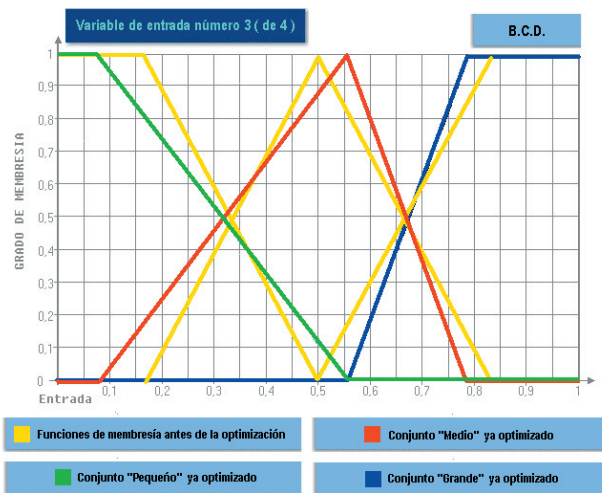


ILUSTRACIÓN 60. Funciones de membresía usadas por BCD para la variable *Largo del pétalo*.

Para la variable 4 (Ancho del pétalo de la flor de la planta *iris*) las funciones de membresía ya optimizadas fueron las siguientes:

$$\mu_{\text{pequeño_Variable_4}(i)} \begin{cases} 1 & \text{si } 0 \leq x \leq .1042 \\ -2.5265285x + 1.2632643 & \text{si } .1042 < x \leq .5000 \\ 0 & \text{si } .5000 < x \leq 1 \end{cases} \quad (6,16)$$

$$\mu_{\text{medio_Variable_4}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .1042 \\ 2.5265285x - .2632642 & \text{si } .1042 < x \leq .5000 \\ -3.2x + 2.6 & \text{si } .5000 < x \leq .8125 \\ 0 & \text{si } .8125 < x \leq 1 \end{cases} \quad (6,17)$$

$$\mu_{\text{grande_Variable_4}(i)} \begin{cases} 0 & \text{si } 0 \leq x \leq .5000 \\ 3.2x - 1.6 & \text{si } .5000 < x \leq .8125 \\ 1 & \text{si } .8125 < x \leq 1 \end{cases} \quad (6,18)$$

En la ilustración 61 se presenta el cambio entre las funciones de membresía iniciales y las funciones de membresía modificadas:

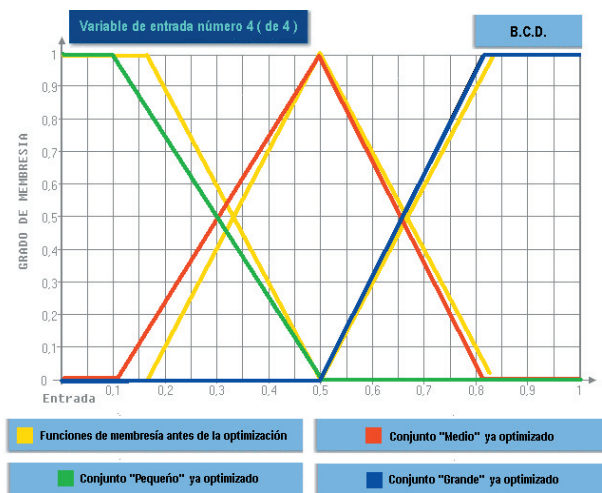


ILUSTRACIÓN 61. Funciones de membresía usadas por BCD para la variable *Ancho del pétalo*.

Posterior al establecimiento y afinación de las funciones de membresía se creó una red neuronal con 12 unidades de entrada, 12 unidades en

la capa intermedia y 3 en la capa de salida. Cada una de las unidades de entrada correspondió a cada conjunto Difuso establecido a partir de las cuatro variables de entrada originales (ver ilustración 62).

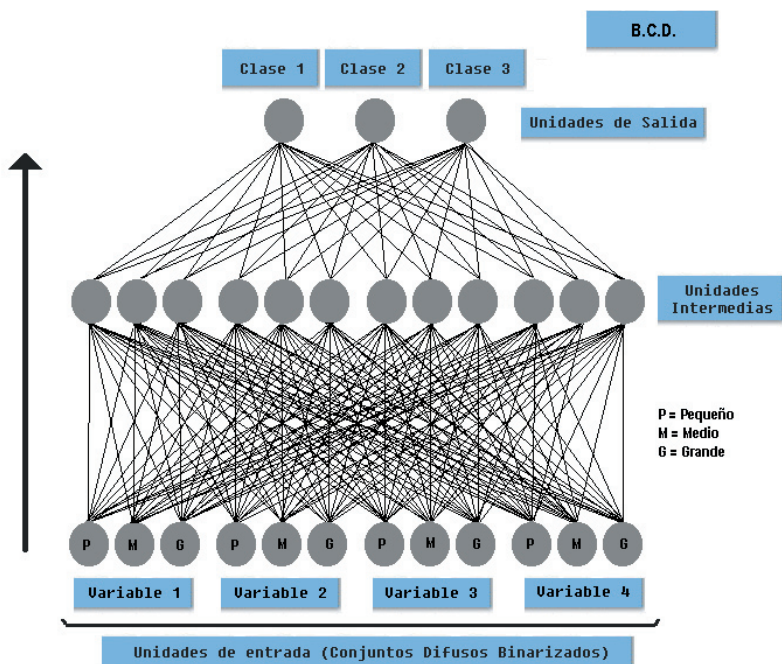


ILUSTRACIÓN 62. Arquitectura de la red usada por BCD.

Dado que se pretendía que la red pudiera clasificar vectores de datos en tres clases diferentes, se utilizaron 3 unidades de salida para lograrlo, (se buscó una mayor actividad en la unidad que representaba a la clase correcta, mientras que las dos restantes deberían mostrar una actividad menor).

La tabla 5 muestra la actividad esperada en las unidades de salida de la red de acuerdo al problema de clasificación de la planta iris.

Tabla 5. Actividad esperada en las unidades de salida de la red empleada por BCD.

Actividad en la salida 1	Actividad en la salida 2	Actividad en la salida 3	Significado
Mayor	Menor	Menor	<i>Iris Setosa</i>
Menor	Mayor	Menor	<i>Iris Versicolor</i>
Menor	Menor	Mayor	<i>Iris Virginica</i>

Para entrenar a esta red neuronal se estableció el conjunto de datos de entrenamiento a partir de los valores de entrada originales, de la siguiente manera acorde con el modelo BCD:

$$y_i \begin{cases} 1 & \text{si } \mu_i > \frac{1}{2} \\ 0 & \text{de cualquier otra manera} \end{cases} \quad (6,19)$$

En donde y_i es el valor por asignar a la unidad de entrada correspondiente y μ_i es el grado de membresía calculado de acuerdo a las funciones de membresía ya afinadas (ilustraciones 58, 59, 60 y 61).

Como conjunto de entrenamiento se usó el total de 150 vectores del problema *iris*.

Después de haber entrenado la red para la clasificación de los vectores redefinidos bajo el BCD, se procedió a aplicar la última fase del procedimiento. Se utilizaron las 81 posibles configuraciones de vectores (binarios) como datos de entrada para la red y se seleccionaron los vectores que para cada caso, la red asoció con alguna de las tres clases (tomando como parámetro criterio de decisión para ello el valor mayor de las tres unidades de salida).

La ilustración 63 muestra el número de vectores asociados por la red usada en BCD, para cada clase de *iris*.

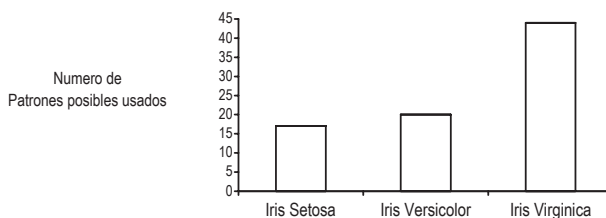


ILUSTRACIÓN 63. Distribución de los patrones posibles binarizados usados por BCD.

Después de la fase de afinación de las reglas, se conformó el conjunto final que se presenta en la tabla 6.

TABLA 6. Conjunto de reglas extraídas por el BCD usando todas las posibles configuraciones.

No	Regla
1	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1
2	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Medio Y V4 es Medio, Entonces: Clase 2
3	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Cualquiera Y V4 es Grande, Entonces: Clase 3
4	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Grande Y V4 No es grande, Entonces: Clase 3
V1: largo de tallo, V2: Ancho de tallo, V3: Largo de pétalo, V4: Ancho de pétalo. Clase 1: <i>Iris Setosa</i> , Clase 2: <i>Iris Versicolor</i> , Clase 3: <i>Iris Virginica</i>	

7.2. Resultados de las evaluaciones realizadas

7.2.1 Evaluación interna del Modelo BCD

Este apartado presenta los resultados de la evaluación del modelo BCD. Esta evaluación se realizó específicamente en tres grandes aspectos:

- i) Relativos al modelo: escalabilidad y generalidad del modelo BCD.
- ii) Relativos a los conjuntos de reglas extraídos: generalidad, modificabilidad y fidelidad a la red neuronal difusa del que fueron extraídos.
- iii) Relativos a las reglas extraídas: comprensibilidad y generalidad.

7.2.2. Evaluación de aspectos relativos al modelo

7.2.2.1. Evaluación de la escalabilidad del modelo BCD

Cuando se habla de escalabilidad se hace referencia a la posibilidad de que el modelo pueda ser aplicado a problemas de mayor escala, ya sea por el número de entradas/salidas que debe integrar en su arquitectura, o por el número de casos de entrenamiento con el que se debe trabajar.

En cuanto a este aspecto, BCD es escalable, ciertas limitaciones que atienden sobre todo a razones de equipo disponible para implementarlos, ya que conforme se incrementa el número de unidades en la capa de

entrada en la red neuronal que el modelo emplea, o incluso en el número de unidades en la capa de salida, la complejidad del proceso se ve incrementada exponencialmente.

En cuanto al número de casos en el conjunto de entrenamiento, parece que el tamaño de estos conjuntos sólo se convierte en un obstáculo cuando se habla de problemas que integran gran número de variables o implican el reconocimiento de muchas clases. De nuevo, las restricciones en este sentido atienden sobre todo, a las grandes exigencias que un proceso de este tipo implica en términos de equipo disponible para el proceso.

7.4.2.2. Evaluación de la generalidad del modelo BCD

Con la generalidad de un modelo se hace referencia a la independencia del modelo con respecto a la topología de la red en la que se utiliza para aplicarlo, así también un modelo general es independiente de los regímenes de entrenamiento y las funciones de activación utilizadas en la red.

BCD por el otro lado, es mucho más general que los dos anteriores, dado que es posible aplicar el modelo para extraer reglas de redes con todo tipo de arquitectura y/o funciones de activación. Sin embargo, una restricción en esta generalidad, que es inherente al propio BCD, es la necesidad de que los datos de entrada sean binarios, es decir, la necesidad de binarizar los datos de entrada se convierte en un obstáculo que restringe la generalidad del modelo. Sin embargo, su grado de posible generalización, aún bajo esta restricción, es mayor que el de FUNEGEN y NEFLCLASS.

Para demostrar la generalidad del modelo BCD se corrió el mismo problema para un mapa autororganizativo de Kohonen.

7.2.3. Evaluación de aspectos relativos a los conjuntos de reglas extraídos por BCD

Se evaluaron las características grupales de cada conjunto final de reglas extraídas por BCD. Es decir, se evaluó también cada *base de reglas* resultante. Las características evaluadas en esta prueba fueron: fidelidad, generalidad y modificabilidad.

7.4.3.1. Evaluación de la Fidelidad de los conjuntos extraídos por BCD

Las tres tareas consideradas para evaluar la fidelidad de las bases de reglas consistieron en la medida del número medio de disparos correctos, el número medio de disparos incorrectos o en falso y la medición de la exclusividad media de las reglas:

7.4.3.1.1. Número medio de disparos correctos en los conjuntos extraídos por BCD

Para conocer el número de disparos correctos se evaluó cada uno de los patrones de entrada del conjunto de datos de entrenamiento del problema de clasificación de la planta iris de acuerdo a la base de reglas extraída.

Los disparos correctos de las reglas extraídas por BCD se muestran en la tabla 7 y en la ilustración 64.

TABLA 7. Disparos correctos en el conjunto de reglas extraídas por el BCD.

No	Regla	Disparos Correctos
1	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Pequeño Y V1 es Pequeño, Entonces: Clase 1	50
2	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	47
3	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Cualquiera Y V4 es Grande, Entonces: Clase 3	46
4	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Grande Y V4 No es grande, Entonces: Clase 3	4
	Total	147

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo
Clase 1: *Iris Setosa*; Clase 2: *Iris Versicolor* y Clase 3: *Iris Virginica*

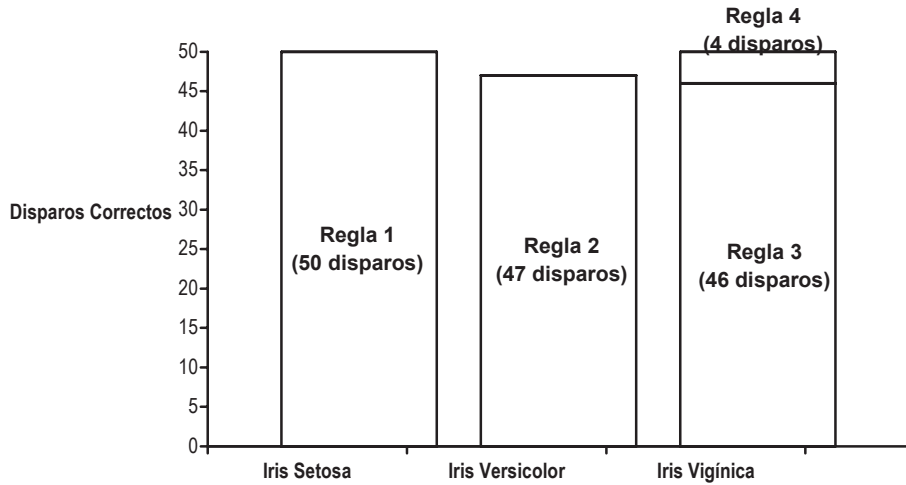


ILUSTRACIÓN 64. Distribución de las reglas de la base extraída por BCD.

La cantidad de disparos correctos es superior al 80% en todos los modelos, tomando como referencia el número de disparos totales ante en el conjunto de datos de entrenamiento, cantidad que podría llegar a ser considerada como poco aceptable, dependiendo claro del dominio del conocimiento bajo el cual se esté trabajando.

7.4.3.1.2. Número medio de disparos en falso en los conjuntos extraídos por BCD

Medición de disparos en falso, que permiten evaluar cuantas veces una regla puede llevar a una conclusión errónea (distinta a la que la misma red neuronal entrenada emitiría).

Los disparos en falso de las reglas extraídas por BCD se presentan en la tabla 8 y en la ilustración 65.

TABLA 8. Disparos en falso de las reglas extraídas por BCD.

No	Regla	Disparos en falso
1	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Pequeño Y V1 es Pequeño, Entonces: Clase 1	0
2	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	3
3	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Cualquiera Y V4 es Grande, Entonces: Clase 3	0
4	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Grande Y V4 No es grande, Entonces: Clase 3	0
Total		3

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo
 Clase 1: *Iris Setosa*; Clase 2: *Iris Versicolor* y Clase 3: *Iris Virginica*

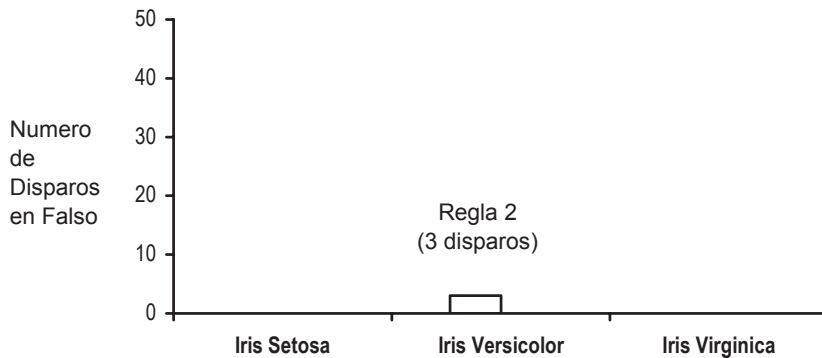


ILUSTRACIÓN 65. Disparos en falso en las reglas extraídas por BCD.

7.4.3.1.3. Exclusividad media de los conjuntos extraídos por BCD

Evaluar el grado de exclusividad de las reglas que nos lleva a conocer la generalidad de las mismas, es decir, la medición del número de casos que implican el disparo de dos o más reglas al mismo tiempo, hecho que posibilita evaluar el grado de generalidad de las reglas.

Las los disparos exclusivos de las reglas extraídas por BCD se presentan en la tabla 9.

Tabla 9. Reglas extraídas por BCD usando todas las posibles configuraciones.

No	Regla	Disparos exclusivos
1	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	100%
2	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	100%
3	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Cualquiera Y V4 es Grande, Entonces: Clase 3	100%
4	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Grande Y V4 No es grande, Entonces: Clase 3	100%
	Total	100%

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo;
Clase 1: *Iris Setosa*; Clase 2: *Iris Versicolor* y Clase 3: *Iris Virginica*

7.4.3.2. Evaluación de la generalidad de los conjuntos de reglas extraídos por BCD

Para evaluar la generalidad de los conjuntos de reglas extraídas por cada modelo, se realizó la medición de dos aspectos de las mismas, la medición de cobertura de casos posibles de configuración de entrada, y la cantidad de reglas de cierre o corte de rama.

7.4.3.2.1. Número medio de casos no cubiertos por las reglas extraídas por BCD

Las reglas extraídas por BCD cubren el 98% de los patrones que se presentan en el conjunto de datos de entrenamiento del problema de clasificación de la planta *iris*, es decir, clasifican 147 de los 150 patrones que integran el problema (ver ilustración 66).

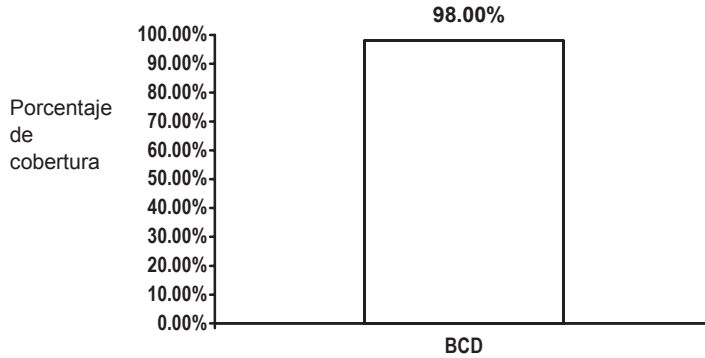


ILUSTRACIÓN 66. Porcentaje de cobertura de casos del conjunto de datos de entrenamiento.

Después de realizar esta prueba, se realizó una prueba más para evaluar la cobertura de las reglas extraídas por BCD. Después de usar las reglas para evaluar los patrones de entrada contenidos en el conjunto de datos de entrenamiento del problema de la clasificación de la planta *iris*, se evaluó la cobertura que esta base de reglas pudiese tener ante nuevos casos. Asumiendo que el conjunto de datos del problema de la planta iris integra una muestra representativa, se evaluó el desempeño de la base de reglas extraída por BCD ante las 81 posibles configuraciones de entrada (calculadas con base en las variables de entrada y los conjuntos difusos establecidos para cada variable), el resultado fue del 96.29%. de cobertura (ver ilustración 67).

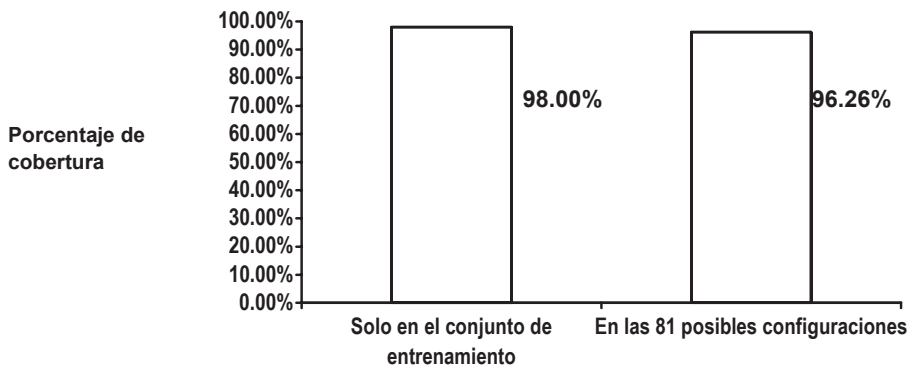


ILUSTRACIÓN 67. Comparación de los porcentajes de cobertura logrados por BCD.

En ambos casos la cobertura sobrepasa al 96%. La prueba de la base de reglas extraída por BCD ante las 81 posibles configuraciones de entrada, se realizó con el fin de tener una idea exacta de la cobertura que las reglas tendrían aún en casos no conocidos.

7.4.3.2.2. Existencia de reglas de cierre o de corte de rama } en los conjuntos extraídos por BCD

BCD no usa reglas de cierre que posibiliten establecer una delimitación del campo de búsqueda en el dominio.

Por supuesto el uso de reglas de cierre depende del dominio del conocimiento que se esté manejando, y en este caso concreto se especula solamente sobre su necesidad. Al no existir en el conjunto de datos originales algún contraejemplo o límite explícito sobre las dimensiones de las variables de los tres tipos *iris* que el problema maneja, se supone solamente que no son necesarias.

7.4.3.3. Evaluación de la modificabilidad de los conjuntos de reglas extraídos por BCD

BCD, no presenta limitaciones graves en cuanto a la modificabilidad del conjunto de reglas extraídas, dado que no implica el uso de una arquitectura de red específica o algún régimen de entrenamiento o funciones de activación determinadas. Sin embargo, aún así, si se quiere modificar el conjunto de reglas, se tiene que realizar el proceso completo de nuevo. Es decir, si se requiere modificar el conjunto de reglas extraído, se elimina el anterior y se repite el proceso con el conjunto de datos de entrada modificado.

7.2.4. Evaluación de aspectos relativos a cada una de las reglas extraídas por BCD

7.4.4.1. Evaluación de la comprensibilidad de cada una de las reglas extraídas por BCD

Para la evaluación de la comprensibilidad de las reglas extraídas se consideraron los siguientes factores en cada regla extraída por los modelos:

7.4.4.1.1. Número medio de reglas extraídas por el modelo BCD

En cuanto al número inicial de reglas extraídas antes de la fase de adecuación de las funciones de membresía, BCD extrajo 81.

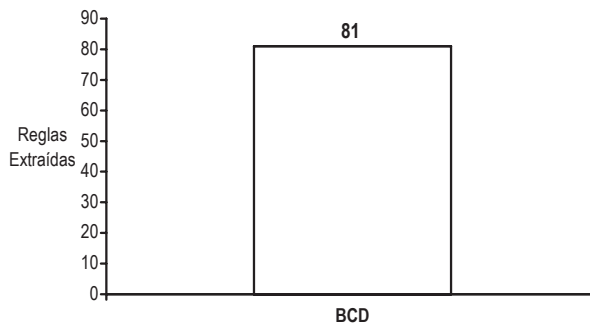


ILUSTRACIÓN 68. Número de reglas iniciales extraídas por el modelo BCD.

Después de la fase de afinación de las funciones de membresía, BCD extrajo 16 reglas finales.

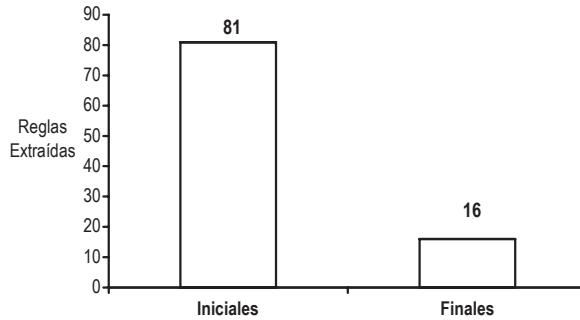


ILUSTRACIÓN 69. Número de reglas finales extraídas por el modelo BCD.

BCD redujo su conjunto inicial de reglas en un 80.25%.

7.4.2.1.2. Número medio de premisas en cada regla extraída por BCD

El número de premisas en cada regla extraída por el modelo BCD (16 reglas finales) es el mismo que el de las variables que el problema original contempla, es decir, 4 premisas.

El número de premisas usadas en cada regla indica también el grado de generalidad alcanzado en la regla. Cabría presuponerse que la ausencia de una variable en las premisas de la regla indica que no importa el valor que pueda tomar. En el caso de BCD esta generalidad se integra en la regla por medio de una premisa con la etiqueta “cualquiera”, es por ello que integra las cuatro premisas correspondientes a cada una de las variables en la regla.

7.4.2.1.3. Forma de presentación de la regla al usuario en el modelo BCD

BCD extrajo reglas conjuntivas e incluyó negaciones en parte de sus premisas o en la conclusión.

En cuanto a la accesibilidad de las reglas al usuario, BCD presentó las reglas en una forma fácilmente comprensible al usuario, por lo que no se hizo necesario un proceso de reescritura de las mismas.

Sin embargo, se debe considerar que si se usa el modelo para extraer reglas de una red neuronal cuyo paradigma de aprendizaje haya sido el no supervisado, entonces las categorías finales bien podrían ser referen-

ciadas con base en etiquetas sin significado para el humano, y dado que bajo este paradigma de aprendizaje no necesariamente se tiene que coincidir con las categorías iniciales previstas para el problema, entonces se desmerita un poco la comprensibilidad para el ser humano.

7.4.2.2. Evaluación de la generalidad de cada regla extraída por BCD

Evaluación de la generalidad de las reglas: Se debe considerar que el objetivo es lograr cubrir el mayor número de casos posible con cada una de las reglas extraídas por el modelo.

7.5. Evaluación del modelo bcd con base en evaluación de pares

Este apartado presenta al lector los resultados de la evaluación de los modelos de extracción usados en esta investigación para compararlos con el BCD y los resultados mismos de la evaluación de las características del BCD. Esta evaluación se realizó específicamente conforme a tres dimensiones básicas de los modelos de extracción de reglas: i) el desempeño del modelo, ii) la efectividad de las reglas, y iii) la efectividad de las bases de reglas.

7.5.1. Evaluación de características relativas a los modelos de extracción

A continuación se presentan los resultados de la evaluación realizada a cada modelo de extracción de reglas de acuerdo a su desempeño, escalabilidad y generalidad.

7.5.1.1. Evaluación del desempeño de los modelos usados

Tras la prueba de los tres modelos (BCD, FUNEGEN y BCD) en la resolución del problema de clasificación de la planta iris, el desempeño fue el que se muestra en los apartados siguientes.

7.5.1.1.1. Porcentaje de clasificación media en el proceso de entrenamiento

La ilustración 70 muestra el desempeño del sistema con los vectores a clasificar para el problema de la planta iris.

El sistema alcanzó un 96.6% de vectores clasificados correctamente, por lo que se puede decir que su desempeño es aceptable.

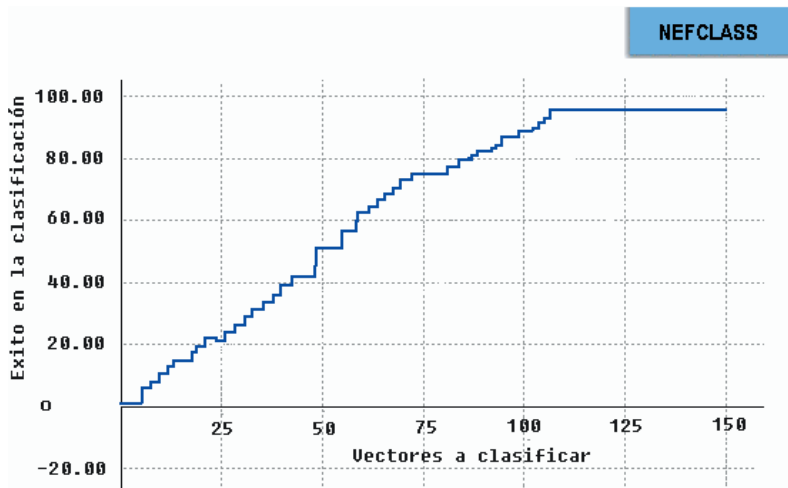


ILUSTRACIÓN 70. Éxito en la clasificación realizada por NEFCLASS.

El éxito en la clasificación de vectores se presenta en la ilustración 71. Como podrá observarse, los vectores más problemáticos en el archivo de prueba, fueron clasificados con un éxito ligeramente mayor después de la optimización de la red.

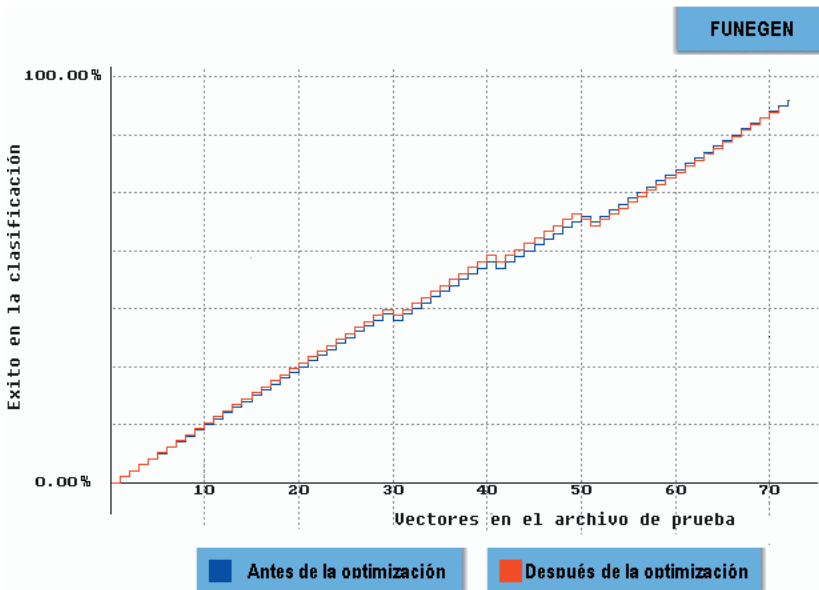


ILUSTRACIÓN 71. Éxito en la clasificación realizada por FUNEGEN.

El desempeño de la red implementada para la resolución de este problema fue del 96.7%. La trayectoria del desempeño se muestra en la ilustración 72.

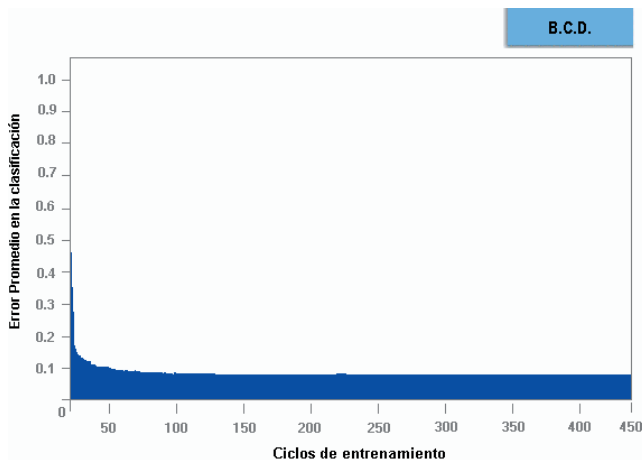


ILUSTRACIÓN 72. Éxito en la clasificación realizada por BCD.

El mayor porcentaje de patrones clasificados lo obtuvo NEFCLASS, que logró una clasificación media el 96.7% de los patrones totales del conjunto de entrenamiento (150 casos), mientras que FUNEGEN logró una media de clasificaciones correctas del 94.00%. BCD logró un nivel de clasificación media de 93.3%.

La ilustración 66 muestra gráficamente el porcentaje de clasificación media obtenido por los modelos ante el problema de clasificación de la planta iris:

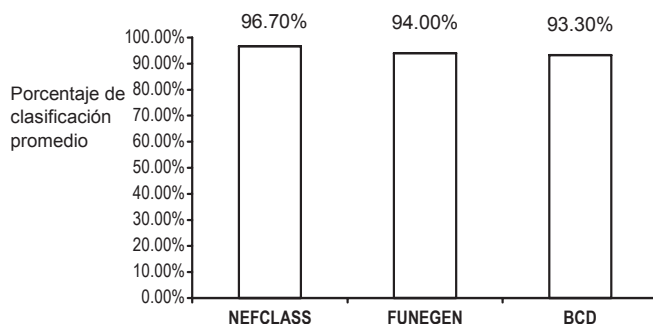


ILUSTRACIÓN 73. Clasificación media lograda por cada modelo.

BCD fue el modelo que logro una media de clasificación menor, logrando un 93.3%. Sin embargo la diferencia con respecto al nivel de clasificación alcanzado por FUNEGEN es poco significativa. Entre BCD y NEFCLASS la diferencia es del orden de 3.40%, es decir 5 patrones clasificados más.

En general el porcentaje de clasificación media logrado podría ser calificado de bueno a muy bueno, sin embargo esta calificación es subjetiva y depende de cada caso particular.

7.5.1.1.2. Número medio de ciclos de entrenamiento necesarios

En cuanto al número de ciclos de entrenamiento necesarios² los modelos usados, realizaron las clasificaciones de la siguiente manera: BCD empleó 450 ciclos para llegar a la clasificación de la planta *iris*; NEFCLASS, empleó 500 ciclos y FUNEGEN utilizó 12,100 ciclos.

² *Epochs* (épocas), en inglés.

La ilustración 74 muestra gráficamente el número de ciclos empleados por cada modelo para llegar a la clasificación de la planta iris:

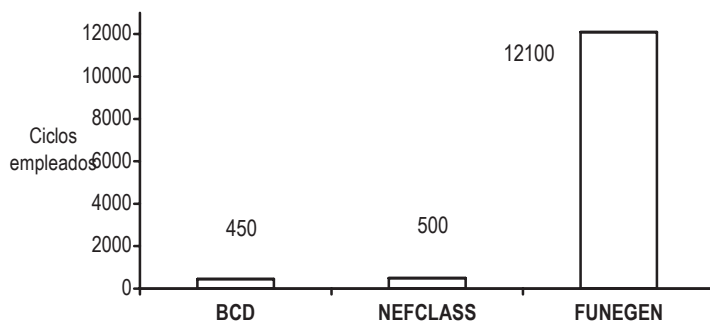


ILUSTRACIÓN 74. Número medio de ciclos empleados por cada modelo en el proceso.

7.5.1.1.3. Refinamiento de las funciones de membresía

Los tres modelos utilizaron como estrategia la readecuación de las funciones de membresía para llegar a un porcentaje más alto de clasificaciones. Sin embargo, FUNEGEN no modificó las reglas extraídas tras esta adecuación de las funciones de membresía, mientras que NEFLCLASS redujo el conjunto de reglas iniciales de 20 reglas a 7 reglas en el conjunto final. BCD redujo el conjunto de reglas iniciales de 81 a 16.

7.5.1.1.2. Evaluación de la escalabilidad de cada modelo usado

Cuando se habla de escalabilidad se hace referencia a la posibilidad de que el modelo pueda ser aplicado a problemas de mayor escala, ya sea por el número de entradas/salidas que debe integrar en su arquitectura, o por el número de casos de entrenamiento con el que se debe trabajar.

En cuanto a este aspecto, los tres modelos son escalables pero, con ciertas limitaciones que atienden sobre todo a razones de equipo disponible para implementarlos, ya que conforme se incrementa el número de unidades en la capa de entrada en cualquiera de los tres modelos, o incluso en el número de unidades en la capa de salida, la complejidad del proceso se ve incrementada exponencialmente.

En cuanto al número de casos en el conjunto de entrenamiento, aún cuando los modelos fueron probados en conjuntos relativamente pequeños (75 o 150 casos), parece que el tamaño de estos conjuntos sólo se convierte en un obstáculo cuando se habla de problemas que integran gran número de variables o implican el reconocimiento de muchas clases. De nuevo, las restricciones en este sentido atienden sobre todo, a las grandes exigencias que un proceso de este tipo implica en términos de equipo.

7.5.1.1.3. Evaluación de la generalidad de cada modelo usado

Con la generalidad de un modelo se hace referencia a la independencia del modelo con respecto a la topología de la red en la que se utiliza para aplicarlo, así también un modelo general es independiente de los regímenes de entrenamiento y las funciones de activación utilizadas en la red.

FUNEGEN y NEFCLASS no son generales bajo esta perspectiva ya que es precisamente el modelo quien, en ambos casos, determina la arquitectura especial de la red que se utilizará para implementarlo, así como también especifica las funciones de activación que se tienen que utilizar.

En este sentido, si se tiene una red neuronal entrenada y se quisiera extraer las reglas que se encuentran implicadas en la arquitectura misma de la red, sería necesario realizar alguno de estos procesos:

- a) Tomar solamente los datos del conjunto de entrenamiento que se usó para entrenar la red en cuestión. Este proceso presenta ciertos inconvenientes, dado que muchas veces se puede dar el caso en el que se tiene una red neuronal entrenada y modificada de acuerdo a necesidades especiales que van más allá de un conjunto de caso de entrenamiento iniciales, y es prácticamente imposible repetir el proceso de afinación seguido con base en la experiencia del uso del mismo sistema, ó
- b) Tratar de generar el mayor número de posibilidades de configuraciones de entrada y tras introducir las en la red entrenada, reconfigurar con base en éstas y las salidas que el sistema ofrece para cada caso, un nuevo conjunto de datos de entrenamiento a partir del cual poder implementar alguno de estos dos modelos, problema que lleva a una complejidad de proceso mucho mayor (incluso un problema *No Polynomial Duro* en algunos casos).

Lo anterior sirve de soporte para afirmar que tanto FUNE como NEFCCLASS son modelos de extracción de reglas que implican necesariamente el uso de conjuntos de datos para entrenar sistemas nuevos, y no son posibles de aplicar (bajo muy contadas excepciones) para la extracción de reglas difusas en redes neuronales ya entrenadas.

BCD por el otro lado, es mucho más general que los dos anteriores, dado que es posible aplicar el modelo para extraer reglas de redes con todo tipo de arquitectura y/o funciones de activación. Sin embargo, una restricción en esta generalidad, que es inherente al propio BCD, es la necesidad de que los datos de entrada sean binarios, es decir, la necesidad de binarizar los datos de entrada se convierte en un obstáculo que restringe la generalidad del modelo. Sin embargo, su grado de posible generalización, aún bajo esta restricción, es mayor que el de FUNEGEN y NEFCCLASS.

7.5.2. Características que fueron evaluadas en cada una de las reglas extraídas

A continuación se presentan los resultados de la evaluación de las reglas extraídas por cada uno de los modelos empleados en este trabajo. Las características que a continuación se tratarán hacen referencia a las reglas extraídas pero solamente en su dimensión individual, es decir, las características que a continuación se mencionan fueron evaluadas en cada una de las reglas obtenidas.

7.5.2.1. Evaluación de la comprensibilidad

Para la evaluación de la comprensibilidad de las reglas extraídas se consideraron los siguientes factores en cada regla extraída por los modelos:

7.5.2.1.1. Número medio de reglas extraídas por cada modelo

En cuanto al número inicial de reglas extraídas antes de la fase de adecuación de las funciones de membresía, BCD fue el que mayor número de reglas extrajo con 81, siguiendo en orden decreciente NEFCCLASS con 20. FUNEGEN estructuró su conjunto de reglas iniciales con 5.

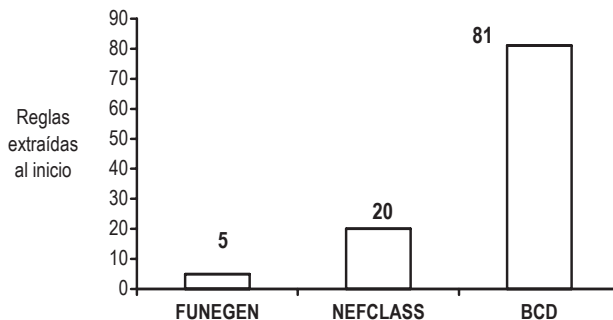


ILUSTRACIÓN 75. Número de reglas iniciales extraídas por cada modelo.

BCD extrajo 16 reglas, siendo el modelo que mayor número de reglas obtuvo, mientras que NEFLCLASS extrajo 7 reglas y FUNEGEN solamente 4 reglas.

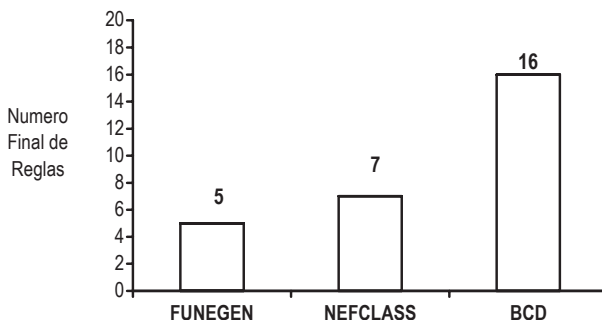


ILUSTRACIÓN 76. Número de reglas finales extraídas por cada modelo.

BCD redujo su conjunto inicial de reglas en un 80.25%, NEFLCLASS lo redujo en un 65% y FUNEGEN no redujo su conjunto inicial de reglas.

7.5.2.1.2. Número medio de premisas en cada regla extraiga

El número de premisas en cada regla extraída por el modelo NEFLCLASS (7 reglas finales) y BCD (16 reglas finales) es el mismo que el de las variables que el problema original contempla, es decir 4 premisas. En cambio FUNEGEN extrajo 3 reglas con una sola premisa y 2 reglas con 2 premisas.

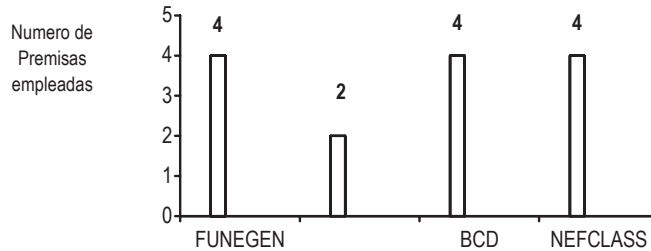


ILUSTRACIÓN 77. Número de premisas empleadas en las reglas extraídas por cada modelo.

El número de premisas usadas en cada regla indica también el grado de generalidad alcanzado en la regla. Cabría presuponerse que la ausencia de una variable en las premisas de la regla indica que no importa el valor que pueda tomar. En el caso de BCD esta generalidad se integra en la regla por medio de una premisa con la etiqueta “cualquiera”, es por ello que integra las cuatro premisas correspondientes a cada una de las variables en la regla, y a pesar de ello tener la misma generalidad que FUNEGEN.

7.5.2.1.3. Forma de presentación de la regla al usuario

Los tres modelos extrajeron reglas conjuntivas y solamente BCD y FUNEGEN incluyeron la negación en parte de sus premisas o en la conclusión.

En cuanto a la accesibilidad de las reglas al usuario NEFCLASS y BCD presentaron las reglas en una forma fácilmente comprensible al usuario, por lo que no se hizo necesario un proceso de reescritura de las mismas. Por otro lado, FUNEGEN presenta las reglas extraídas en una forma un poco más compleja, debido a que usa las etiquetas que el mismo sistema asigna a los conjuntos difusos generados, por lo que se requiere un proceso de reescritura de las reglas para que éstas sean fácilmente comprensibles por el humano.

7.5.2.2. Evaluación de la generalidad

Evaluación de la generalidad de las reglas: Se debe considerar que el objetivo es lograr cubrir el mayor número de casos posible con cada una de las reglas extraídas por el modelo.

7.5.3. Características que fueron evaluadas en los conjuntos de reglas extraídos

También se evaluaron las características grupales de cada conjunto final de reglas extraídas. Es decir, se evaluó también cada *base de reglas* resultante en los tres modelos. Las características evaluadas en esta dimensión fueron: fidelidad, generalidad y modificabilidad.

7.5.3.1. Evaluación de la Fidelidad

Las tres tareas consideradas para evaluar la fidelidad de las bases de reglas consistieron en la medida del número de disparos correctos, el número de disparos incorrectos o en falso y la medición de la exclusividad de las reglas:

7.5.3.1.1. Número medio de disparos correctos

Para conocer el número de disparos correctos se evaluó cada uno de los patrones de entrada del conjunto de datos de entrenamiento del problema de clasificación de la planta iris de acuerdo a la base de reglas extraída. Los disparos correctos de las reglas extraídas por NEFCLASS se presentan en la tabla 10.

TABLA 10. Disparos correctos de la base de reglas extraída por NEFCLASS.

No	Regla	Disparos Correctos
1	SI V1 es Pequeño Y V2 es Medio Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	10
2	SI V1 es Pequeño Y V2 es Pequeño Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	40
3	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Pequeño, Entonces: Clase 2	11
4	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	17
5	SI V1 es Medio Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	5
6	SI V1 es Grande Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	41

No	Regla	Disparos Correctos
7	SI V1 es Grande Y V2 es Medio Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	2
	Total	126

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho del pétalo

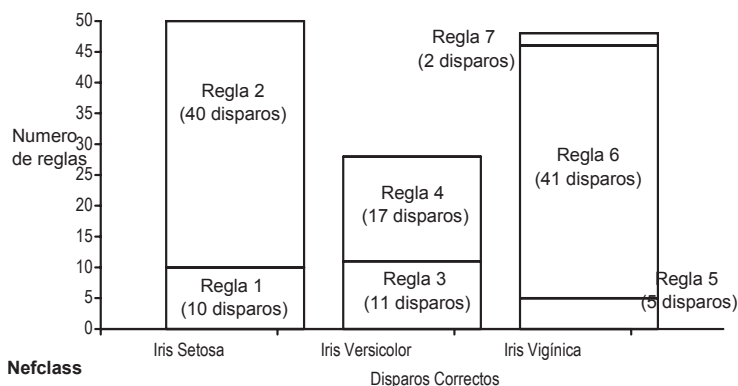


ILUSTRACIÓN 78. Distribución de las reglas en la base extraída por NEFCCLASS.

Las los disparos correctos de las reglas extraídas por FUNEGEN se muestran en la tabla 11.

TABLA 11. Disparos correctos en las reglas extraídas por FUNEGEN.

No.	Regla	Disparos Correctos
1	Si I2 es mf0 (Pequeño) Entonces Salida 0	50
2	Si I2 es mf2 (Medio) Y I3 es mf2 (Medio), Entonces Salida 1	32
3	Si I2 es mf4 (Grande), Entonces Salida 2	8
4	Si I2 es mf2 (Medio) Y I3 es mf2(Medio), Entonces No salida 2	49
5	Si I3 es mf4 (Grande), Entonces No Salida 1	1
	Total	140

Salida 0 corresponde a la planta iris *Setosa*, Salida 1 a la planta iris *Versicolor*, Salida 2 a la planta iris *Virgínica*.

Las funciones de membresía mfX muestran la etiqueta dada al conjunto difuso entre paréntesis³

3 Nota: La etiqueta lingüística colocada entre paréntesis () en las premisas de las reglas no son parte del resultado original, sino que se colocaron en este trabajo solamente con el fin de facilitar la comprensión de las reglas al lector.

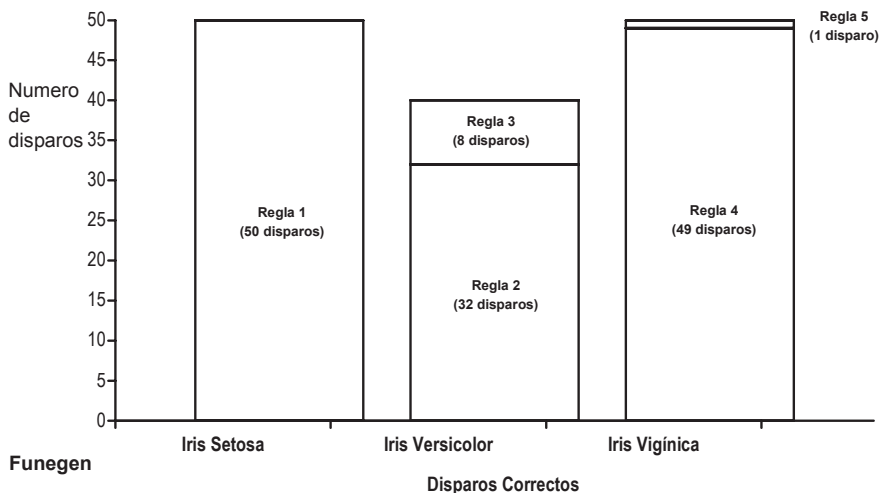


ILUSTRACIÓN 79. Disparos correctos en la base de reglas extraída por FUNEGEN.

Los disparos correctos de las reglas extraídas por BCD se muestran en la tabla 12 y la ilustración 80.

TABLA 12. Disparos correctos en el conjunto de reglas extraídas por el BCD.

No	Regla	Disparos Correctos
1	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	50
2	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	47
3	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Cualquiera Y V4 es Grande, Entonces: Clase 3	46
4	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Grande Y V4 No es grande, Entonces: Clase 3	4
Total		147

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo
 Clase 1: *Iris Setosa*; Clase 2: *Iris Versicolor* y Clase 3: *Iris Virginica*

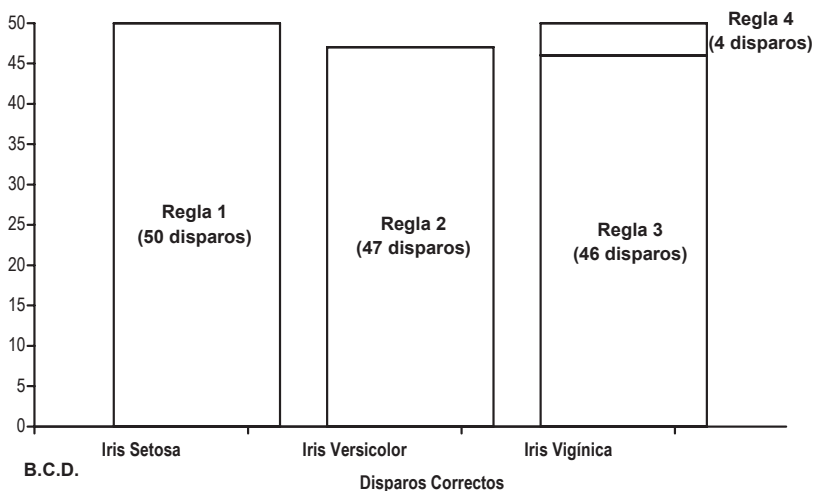


ILUSTRACIÓN 80. Distribución de las reglas de la base extraída por BCD.

La cantidad de disparos correctos es superior al 80% en todos los modelos, tomando como referencia el número de disparos totales ante en el conjunto de datos de entrenamiento, cantidad que podría llegar a ser considerada como poco aceptable, dependiendo claro del dominio del conocimiento bajo el cual se esté trabajando.

Cabe resaltar que el número de disparos correctos obtenido en la base de reglas de BCD es el mayor de todos, alcanzando un 98.00%. FUNE-GEN alcanzó un porcentaje del 93.33% y NEFCCLASS llegó apenas al 84% de disparos correctos (ver ilustración 81).

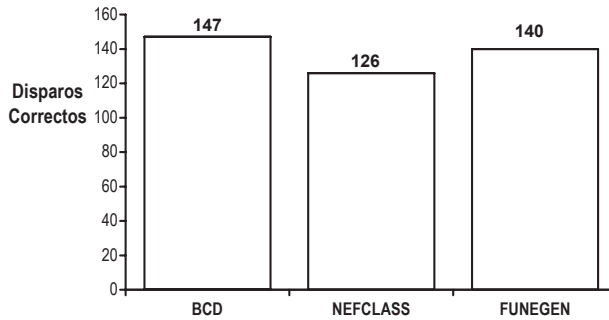


ILUSTRACIÓN 81. Disparos correctos en las bases de reglas obtenidas por los tres modelos.

7.5.3.1.2. Número medio de disparos en falso

Medición de disparos en falso, que permiten evaluar cuantas veces una regla puede llevar a una conclusión errónea (distinta a la que la misma red neuronal entrenada emitiría).

Las los disparos en falso de las reglas extraídas por NEFCLASS se presentan en la tabla 13.

TABLA 13. Disparos en falso en las Reglas difusas extraídas por NEFCLASS.

No	Regla	Disparos en falso
1	SI V1 es Pequeño Y V2 es Medio Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	0
2	SI V1 es Pequeño Y V2 es Pequeño Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	0
3	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Pequeño, Entonces: Clase 2	0
4	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	0
5	SI V1 es Medio Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	0
6	SI V1 es Grande Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	9
7	SI V1 es Grande Y V2 es Medio Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	0
	Total	9

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho del pétalo

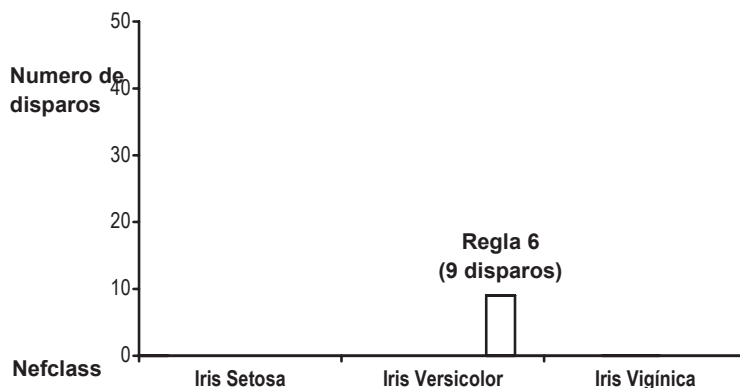


ILUSTRACIÓN 82. Disparos en falso en la base de reglas extraída por NEFCCLASS.

Las los disparos en falso de las reglas extraídas por FUNEGEN se presentan en la tabla 14 y la ilustración 83.

TABLA 14. Disparos en falso de las reglas extraídas por FUNEGEN.

No.	Regla	Disparos en falso
1	Si I2 es mf0 (Pequeño) Entonces Salida 0	0
2	Si I2 es mf2 (Medio) Y I3 es mf2 (medio), Entonces Salida 1	0
3	Si I2 es mf4 (Grande), Entonces Salida 2	14
4	Si I2 es mf2 (Medio) Y I3 es mf2(medio), Entonces No salida 2	0
5	Si I3 es mf4 (Grande), Entonces No Salida 1	14
	Total	14*

Salida 0 corresponde a la planta iris *Setosa*, Salida 1 a la planta iris *Versicolor*, Salida 2 a la planta iris *Virginica*.
 Las funciones de membresía mfX adjunto muestran la etiqueta dada al conjunto difuso.⁴
 (*) Se conserva el mismo número de disparos en falso debido a que las dos reglas implicadas son complementarias

4 Nota: La etiqueta lingüística colocada entre paréntesis () en las premisas de las reglas no son parte del resultado original, sino que se colocaron en este trabajo solamente con el fin de facilitar la comprensión de las reglas al lector.

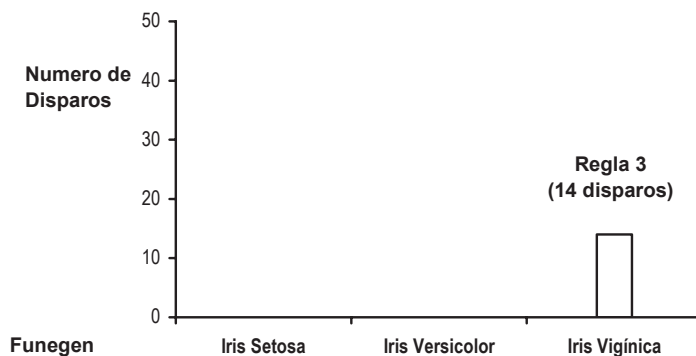


ILUSTRACIÓN 83. Disparos en falso de las reglas extraídas por FUNEGEN.

Las los disparos en falso de las reglas extraídas por BCD se presentan en la tabla 13 y la ilustración 84.

TABLA 15. Disparos en falso de las reglas extraídas por BCD.

No	Regla	Disparos en falso
1	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Pequeño Y V1 es Pequeño, Entonces: Clase 1	0
2	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	3
3	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Cualquiera Y V4 es Grande, Entonces: Clase 3	0
4	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Grande Y V4 No es grande, Entonces: Clase 3	0
Total		3

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo
 Clase 1: *Iris Setosa*; Clase 2: *Iris Versicolor* y Clase 3: *Iris Virginica*



ILUSTRACIÓN 84. Disparos en falso en las reglas extraídas por BCD.

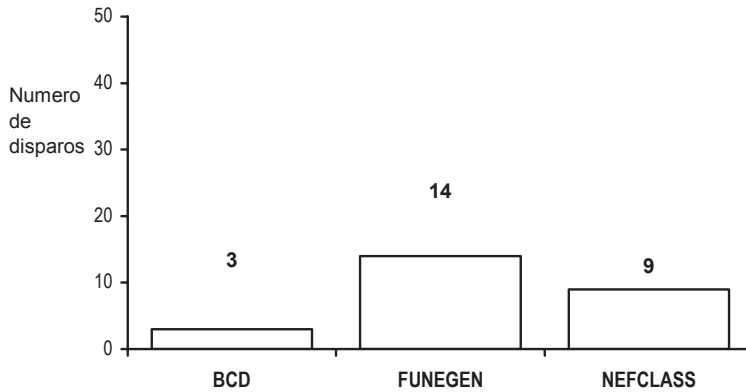


ILUSTRACIÓN 85. Disparos en falso en las reglas extraídas por los tres modelos.

7.5.3.1.3. Exclusividad media de los conjuntos extraídos

Evaluar el grado de exclusividad de las reglas que nos lleva a conocer la generalidad de las mismas, es decir, la medición del número de casos que implican el disparo de dos o más reglas al mismo tiempo, hecho que posibilita evaluar el grado de generalidad de las reglas.

Las los disparos exclusivos de las reglas extraídas por NEFLCLASS se muestran en la tabla 16.

TABLA 16. Disparos exclusivos en las reglas extraídas por NEFLCLASS.

No	Regla	Disparos Exclusivos
1	SI V1 es Pequeño Y V2 es Medio Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	100%
2	SI V1 es Pequeño Y V2 es Pequeño Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	100%
3	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Pequeño, Entonces: Clase 2	100%
4	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	100%
5	SI V1 es Medio Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	100%
6	SI V1 es Grande Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	100%
7	SI V1 es Grande Y V2 es Medio Y V3 es Grande Y V4 es Grande, Entonces: Clase 3	100%
	Total general	100%

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho del pétalo

Las los disparos exclusivos de las reglas extraídas por FUNEGEN son los que se presentan en la tabla 17.

TABLA 17. Disparos exclusivos de las reglas extraídas por FUNEGEN.

No	Regla	Disparos exclusivos
1	Si I3 es mf0 (pequeño) Entonces Salida 0	100%
2	Si I2 es mf2 (medio) Y I3 es mf2 (medio), Entonces Salida 1	50%
3	Si I2 es mf4 (grande), Entonces Salida 2	50%
4	Si I2 es mf2 (medio) Y I3 es mf2(medio), Entonces No salida 2	50%
5	Si I3 es mf4 (pequeño), Entonces No Salida 1	50%
	Total	60%

Salida 0 corresponde a la planta iris *Setosa*, Salida 1 a la planta iris *Versicolor*, Salida 2 a la planta iris *Virginica*.
Las funciones de membresía mfX adjunto muestran la etiqueta dada al conjunto difuso⁵

Cabe resaltar el hecho de que existen dos pares de reglas que son complementarias: a) la regla 4 es complemento de la regla 2, por lo que se podría considerar como redundante, dado que si en la regla 2 ya se

5 Nota: La etiqueta lingüística colocada entre paréntesis () en las premisas de las reglas no son parte del resultado original, sino que se colocaron en este trabajo solamente con el fin de facilitar la comprensión de las reglas al lector.

establecen las condiciones para concluir que se trata de la “Salida 1” y solamente ésta; entonces la regla 4 es obvia. De igual manera, sería obvia la regla: Si I2 mf2 (Medio), Y I3 mf2 (Medio), Entonces: No salida 0. Sin embargo, esta regla no aparece en el conjunto final de reglas extraídas. Para las reglas 3 y 5 sucede lo mismo, la regla 5 es obvia, por lo establecido en la regla 3.

Las los disparos exclusivos de las reglas extraídas por BCD se presentan en la tabla 18.

TABLA 18. Reglas extraídas por BCD usando todas las posibles configuraciones

No	Regla	Disparos exclusivos
1	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1	100%
2	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Medio Y V4 es Medio, Entonces: Clase 2	100%
3	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Cualquiera Y V4 es Grande, Entonces: Clase 3	100%
4	Si V1 es Cualquiera Y V2 es Cualquiera Y V3 es Grande Y V4 No es grande, Entonces: Clase 3	100%
	Total	100%

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo;
Clase 1: *Iris Setosa*; Clase 2: *Iris Versicolor* y Clase 3: *Iris Virginica*

7.5.3.2. Evaluación de la generalidad del conjunto de reglas extraído

Para evaluar la generalidad de los conjuntos de reglas extraídas por cada modelo, se realizó la medición de dos aspectos de las mismas, la medición de cobertura de casos posibles de configuración de entrada, y la cantidad de reglas de cierre o corte de rama.

7.5.3.2.1. Número medio de casos no cubiertos por las reglas

Las reglas extraídas por BCD cubren el 98% de los patrones que se presentan en el conjunto de datos de entrenamiento del problema de clasificación de la planta *iris*, es decir, clasifican 147 de los 150 patrones que integran el problema.

El modelo FUNEGEN cubrió con la base de reglas extraídas, el 93.33% de los casos, y NEFLCLASS cubre solamente el 84.00% de los casos del con-

junto de datos del problema de clasificación de la planta *iris* (ver ilustración 86).

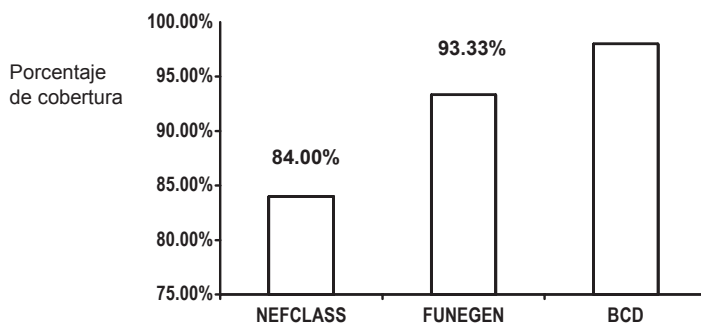


ILUSTRACIÓN 86. Porcentaje de cobertura de casos del conjunto de datos de entrenamiento.

Para el modelo BCD se realizó una prueba más para evaluar la cobertura de sus reglas: Después de usar las reglas para evaluar los patrones de entrada contenidos en el conjunto de datos de entrenamiento del problema de la clasificación de la planta *iris*, se evaluó la cobertura que esta base de reglas pudiese tener ante nuevos casos. Asumiendo que el conjunto de datos del problema de la planta *iris* integra una muestra representativa, se evaluó el desempeño de la base de reglas extraída por BCD ante las 81 posibles configuraciones de entrada (calculadas con base en las variables de entrada y los conjuntos difusos establecidos para cada variable), el resultado fue del 96.29% de cobertura (ver ilustración 80).

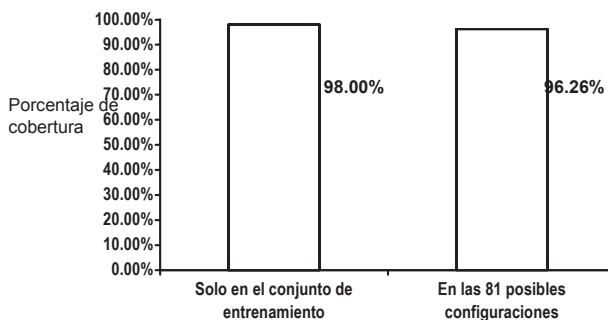


ILUSTRACIÓN 87. Comparación de los porcentajes de cobertura logrados por BCD.

En ambos casos la cobertura sobrepasa al 96%. La prueba de la base de reglas extraída por BCD ante las 81 posibles configuraciones de entrada, se realizó con el fin de tener una idea exacta de la cobertura que las reglas tendrían aún en casos no conocidos.

7.5.3.2.2. Existencia de reglas de cierre o de corte de rama

Ninguno de los tres modelos usó reglas de cierre que posibiliten establecer una delimitación del campo de búsqueda en el dominio.

Por supuesto el uso de reglas de cierre depende del dominio del conocimiento que se esté manejando, y en este caso concreto se especula solamente sobre su necesidad. Al no existir en el conjunto de datos originales algún contraejemplo o límite explícito sobre las dimensiones de las variables de los tres tipos *iris* que el problema maneja, se supone solamente que no son necesarias.

7.5.3.3. Evaluación de la modificabilidad del conjunto de reglas extraído

Tanto FUNEGEN como NEFLCLASS son modelos que implican que la extracción de reglas se lleve a cabo en una arquitectura de red y bajo condiciones de entrenamiento de la misma, especialmente diseñados para este fin. Si se hicieran modificaciones a las configuraciones de entrada bajo las cuales se entrena a la red para su posterior examen y extracción de reglas, el procedimiento tendría que volver a repetirse por completo. Es por ello que parece adecuado hablar de una nula modificabilidad de las reglas debido precisamente a que se hace necesario repetir el proceso completo ante cada modificación.

BCD, presenta un poco menos de trabajo ante esta posibilidad de modificar el conjunto de datos de entrada, dado que no implica el uso de una arquitectura de red específica o algún régimen de entrenamiento o funciones de activación determinadas. Sin embargo, el proceso aún así se tiene que realizar por completo de nuevo.

En ninguno de los modelos evaluados es posible la modificación de reglas por separado conforme el conjunto de casos de entrenamiento cambia. En los tres modelos la base de reglas se elimina y se repite el proceso.

7.5.4. Capacidad de BCD para llegar a los mismos resultados que los otros modelos en igualdad de condiciones

Para evaluar esta característica se corrió el procedimiento del modelo BCD bajo las siguientes características:

- El uso de las funciones de membresía que NEFLCLASS y FUNEGEN establecieron para la resolución del problema de clasificación de la planta iris, y
- El uso solamente del conjunto de patrones de entrenamiento para la extracción de las reglas.

Los resultados obtenidos por BCD usando las funciones de membresía establecidas por NEFLCLASS se muestran en la tabla 19.

TABLA 19. Reglas extraídas por BCD usando las mismas funciones de membresía que NEFLCLASS.

No	Regla
1	SI V1 es Pequeño Y V2 No es Grande Y V3 es Pequeño Y V4 es Pequeño, Entonces: Clase 1
2	SI V1 es Medio Y V2 es Pequeño Y V3 es Medio Y V4 No es Grande, Entonces: Clase 2
3	SI V1 No es pequeño Y V2 es Pequeño Y V3 es Grande Y V4 es Grande, Entonces: Clase 3
4	SI V1 es Grande Y V2 No es Grande Y V3 es Grande Y V4 es Grande, Entonces: Clase 3

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho del pétalo

Las reglas obtenidas por BCD bajo estas condiciones, son distintas que las reglas extraídas por NEFLCLASS, pero son equivalentes lógicamente.

Los resultados obtenidos por BCD usando las funciones de membresía establecidas por FUNEGEN se muestran en la tabla 20.

TABLA 20. Reglas extraídas por BCD usando las mismas funciones de membresía que FUNEGEN.

No.	Regla
1	Si V3 es Pequeño, Entonces: Clase 1
2	Si V3 es Medio, y V4 es Medio, Entonces: Clase 2
3	Si V3 es Grande, Entonces: Clase 3

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho del pétalo

En este caso, el conjunto final de reglas extraído por BCD se compuso solamente de 3 reglas, que son iguales a 3 de las reglas extraídas por FUNEGEN. Sin embargo, BCD no extrajo las reglas 4 y 5 del conjunto final extraído por FUNEGEN, reglas que son redundantes y que quedan cubiertas por las reglas 2 y 3.

8

Conclusiones

A manera de conclusión se presentan en este capítulo las contribuciones y limitaciones del trabajo de investigación realizado, para finalmente, poder esbozar algunas de las propuestas que se harían a trabajos futuros en esta línea.

8.1. Las contribuciones

El presente trabajo hace, principalmente, dos contribuciones al conocimiento sobre la extracción de reglas difusas en redes neuronales difusas y en sistemas expertos neuronales difusos a) propone un método efectivo y simple para extraer reglas difusas, cuyo desempeño y logros pueden ser comparados con los de otros métodos actuales, y b) se propone también una metodología de evaluación de modelos de extracción de reglas, que integra el uso de características tanto de las reglas en un sentido individual, de las reglas en su dimensión grupal y del modelo en general. Esta metodología no se presenta como “la única” ni como la “mejor”, pero sí como una integración de gran número de características que permiten establecer una evaluación más completa.

8.1.1. La propuesta de un modelo de extracción de reglas difusas efectivo y eficiente

Se logró diseñar y consolidar la propuesta de un modelo de extracción de reglas difusas que ofrece las siguientes ventajas frente dos de los modelos más sólidos en la actualidad (FUNEGEN y NEFCLASS).

8.1.1.1. BCD es más sencillo de aplicar

La sencillez de aplicación de un modelo de extracción de reglas es un indicador de ventaja frente a otros modelos, a juicio muy personal. Esta sencillez debe ser reconocida a la luz de la evaluación del desempeño y de los resultados logrados por cada modelo, ya que no basta con que un modelo sea reconocido como “simple” o “sencillo”, sino que además, debe integrar iguales o mejores resultados que modelos similares. Por esto se buscó comparar el desempeño de BCD ante problemas similares, contra otros modelos (FUNEGEN y NEFCLASS), así como también se evaluó si BCD era capaz de llegar a los mismos resultados que éstos, en igualdad de condiciones.

En ambos casos el desempeño de BCD fue similar (y en algunos casos mejor), por lo que se puede afirmar que es más sencillo y posibilita llegar a resultados similares a pesar de esta sencillez.

8.1.1.2. BCD no requiere información alguna sobre la estructura interna de la red entrenada

Esta característica se considera como una ventaja muy importante debido a que posibilita la aplicación del modelo en cualquier arquitectura, por lo que se puede aplicar tanto a sistemas expertos neuronales difusos (motivación inicial del trabajo), como a redes neuronales difusas en general.

Muchos problemas de la vida real implican el uso de arquitecturas complejas de red para poder llegar a niveles medios de clasificación aceptables; al aumentar la complejidad de la arquitectura (numero de capas, tipos de conexión, funciones de transferencia, etc.), también aumenta la complejidad de la extracción de reglas en modelos que dependen de la misma. BCD ofrece niveles de complejidad menores al no depender de estos aspectos de la red.

Incluso el modelo no se restringe solamente a su aplicación en redes neuronales, sino que es aplicable a sistemas de cualquier tipo, dado que se usan solamente los patrones de entrada y salida.

8.1.1.3. BCD no implica un régimen de entrenamiento de red determinado

La independencia del régimen de entrenamiento de la red neuronal es otra ventaja que caracteriza al modelo BCD. Usar solamente patrones de entrada y sus correspondientes salidas posibilita el uso del modelo en redes entrenadas tanto, bajo enfoques de aprendizaje supervisado, como bajo enfoques de aprendizaje no supervisado, por lo que sus posibilidades de aplicación a gran cantidad de dominios se incrementan.

8.1.1.4. BCD permite el uso de herramientas de minimización lógica

La posibilidad de uso de herramientas de minimización lógica facilita al usuario mucho del trabajo en la extracción de reglas, hecho que propicia que el modelo sea mucho más accesible y menos complicado.

El uso de este tipo de herramientas también tiene relación con la siguiente ventaja: la extracción de reglas óptimas y máximamente generales.

8.1.1.5. BCD extrae reglas óptimas y máximamente generales

La minimización lógica de los conjuntos iniciales de reglas extraídas permite obtener reglas óptimas, además de que no se hace necesario ningún tipo de simplificación adicional, por lo que tampoco requiere de algún tipo de reescritura de las mismas.

8.1.1.6. BCD extrae reglas 100% comprensibles y entendibles

El modelo extrae reglas que son presentadas al humano con las etiquetas lingüísticas que originalmente fueron asignadas a los conjuntos difusos. Además el uso de solamente conjunciones y negaciones (si es el caso) simplifica aún más la comprensión de las mismas al ofrecer al usuario sentencias determinantes.

8.1.1.7. BCD no impone limitaciones en cuanto al uso de premisas en cada regla

No existe un límite en cuanto al número de premisas en cada regla extraída, sin embargo, se reconoce que para lograr cumplir con el punto anterior, es decir, con la comprensibilidad de las reglas, la limitante es la composición misma del problema. BCD implica el uso de igual o menor número de premisas en las reglas, que variables de entrada contemple el problema a resolver.

En cuanto al tamaño de los problemas que se pueden resolver mediante la extracción de reglas BCD ofrece también ciertas ventajas ya que minimiza mucho de la complejidad implicada en problemas de más de 20 variables de entrada, ante los cuales los modelos evaluados para comparación con el BCD, exigen una cantidad de recursos de proceso, que hacen que en la práctica se considere un límite. BCD también eleva la complejidad conforme aumenta el número de variables en los patrones de entrada que se debe procesar, sin embargo, el auxilio de la lógica difusa y el tratamiento binarizado de las variables permite rangos mayores de trabajo.

8.1.1.8. BCD no impone un número limitado de reglas a extraer

Los modelos de extracción de reglas analizados tienen dos limitantes en cuanto al número de reglas en la red: a) el número de reglas iniciales a partir de las cuales el sistema debe comenzar a trabajar es un parámetro que el usuario debe determinar antes de comenzar la extracción; y b) aunque no existe un límite en cuanto a la cantidad de reglas a extraer, tanto FUNEGEN como NEFLCLASS requieren de mayor inversión en recursos conforme el número aumenta, problema que de nuevo, vuelve un límite práctico a la complejidad implicada en problemas mayores. BCD, dada la simplificación que realiza de los problemas extiende estos límites a problemas mayores.

8.1.1.9. BCD proporciona reglas válidas aún para casos no contemplados en el conjunto de entrenamiento

En BCD las reglas extraídas contemplan todas las posibilidades de configuración posibles (tomando en cuenta la actual configuración de la red, se toma el posible comportamiento de la red ante casos no contemplados en los conjuntos de entrenamiento). Por ello se puede decir que las reglas son lo más generales posibles (con la restricción claro, de asumir que el conjunto de patrones de entrada es una muestra representativa de un universo más amplio).

La consideración de todas las posibles combinaciones de reglas es un problema NP Duro, sin embargo una alternativa viable es la que ofrece precisamente este modelo BCD ya que las combinaciones que se tienen que revisar para tener contempladas todas las posibilidades son mucho menores cuando hablamos de conjuntos difusos. Aún así el número de posibilidades depende de las variables de entrada que se estén considerando para cada problema y el número de conjuntos difusos que se integren en cada una de estas variables de entrada, pero aún así la cantidad es mucho menor al conjunto de todas las posibles configuraciones de entrada usando valores continuos.

8.1.1.10. BCD puede llegar a los mismos resultados que otros modelos de extracción de reglas con una inversión menor de recursos

BCD utilizó para el tratamiento del problema de extracción de reglas de clasificación para el problema de la planta iris, funciones de membresía distintas a las que emplearon FUNEEN y NEFCCLASS (cuyas funciones de membresía también diferían entre sí), y por lo consiguiente, el conjunto de reglas que extrajo, aunque válido, fue distinto a los conjuntos extraídos por estos dos modelos.

Sin embargo, para corroborar las ventajas de BCD frente a estos otros dos modelos, había que compararlo frente a éstos en igualdad de condiciones, es decir, usando las mismas funciones de membresía que éstos usaron para lograr sus conjuntos finales y usando solamente para la extracción de reglas el conjunto de patrones de entrenamiento. BCD logró en ambos casos extraer conjuntos lógicamente equivalentes de reglas, pero con un nivel de complejidad de procesamiento mucho menor. Esta última prueba, además de que proporcionó otro elemento más para la

comparación de los modelos, también ayudó a convalidación de las reglas finales extraídas por BCD.

8.1.2. la propuesta de una metodología de evaluación integral de modelos de extracción de reglas

La otra aportación principal de esta investigación es la propuesta de una metodología de evaluación de modelos de extracción de reglas difusas (que incluso podría aplicarse con algunas modificaciones, en modelos de extracción de reglas en redes neuronales no difusas).

En realidad no existe una metodología estándar para la evaluación de modelos de extracción, reglas obtenidas y bases de reglas extraídas. La metodología aquí utilizada se diseñó y estructuró de la siguiente manera:

- a) Con base en las características que se pueden considerar como ideales en cada una de estas tres entidades y con la medición de los parámetros que a juicio personal se podían considerar como referentes empíricos de las propiedades evaluadas. Aunque la misma selección de las características que se consideran relevantes o deseables para un modelo de extracción de reglas son muy variables de un autor a otro y aún no es posible ubicar líneas de trabajo o corrientes teóricas sólidas al respecto, sino más bien propuestas. Es por ello que los aspectos de las reglas contemplados en este fueron tomados de diversos autores, tratando de articular la propuesta de cada uno de ellos en un conjunto de características lo más representativo posible del conjunto en general, y
- b) Con base en la metodología usada en estudios similares previos, tomando como ejemplo sobre todo, los estudios comparativos de modelos, y las estrategias de medición y evaluación utilizadas en éstos.

Es por estos dos aspectos que la metodología se ha considerado, también, como una aportación al campo de la evaluación de modelos de extracción de reglas, campo que a la fecha está mucho menos explorado que el de la extracción de reglas.

8. 2. Desventajas

Así como se pudieron identificar las ventajas que BCD presentó al compararlo con FUNEGEN y NEFCLASS, también fue posible identificar desventajas o puntos en contra y es precisamente el objetivo de este apartado mostrar estos aspectos, a fin de sentar las bases bajo las cuales se podría continuar y mejorar el estudio de la extracción de reglas basada en la binarización de los conjuntos difusos.

8.2.1. El número de premisas depende de las variables contempladas en la entrada de la red

Esta característica fue mencionada antes como una ventaja, pero en sí misma constituye también una desventaja: un número grande de variables de entrada en un problema. Determina la extracción de reglas por lo general, grandes - se había mencionado también que el número de premisas en las reglas es igual o menor al número de variables de entrada en el problema-, y las reglas grandes constituyen un problema ya que:

- a) Son difíciles de comprender para el ser humano, hecho que hace que se pierda de vista un objetivo legítimo en la extracción de reglas, que es el de hacer el conocimiento implicado en las redes neuronales, accesible al ser humano. Las reglas demasiado grandes se vuelven inaccesibles, y
- b) Las reglas grandes implican mayor inversión en términos de proceso en el proceso de reducción lógica de las mismas. Aunque esta tarea es mucho más sencilla que la equivalente realizada en otros modelos como los usados en este trabajo para comparación, de cualquier manera incrementa su complejidad conforme el número de premisas se incrementa y se puede llegar también a constituirse en un límite práctico.

Es por estos dos aspectos que esta característica del modelo BCD puede considerarse también una desventaja.

8.2.2. La complejidad de las reglas extraídas y la complejidad del proceso de extracción dependen del número de conjuntos difusos al interior de cada variable

Aunque esta característica fue considerada también como una ventaja frente a FUNEGEN y NEFCCLASS, también puede considerarse como un límite a la escalabilidad del modelo y por lo tanto también una desventaja ante el tratamiento de problemas grandes. Aunque la reducción de variables continuas a conjuntos discretos difusos ya de por sí es una simplificación del problema que ofrece menores niveles de complejidad, se enfrenta a dos problemas principales:

- a) No existe un método formal para establecer los conjuntos difusos y las funciones de membresía para cada uno de ellos. Este hecho impide que la tarea de extracción de reglas se pueda llegar a considerar como un proceso completamente automático, ya que se hace necesaria, de cualquier manera, la intervención de humano para fijar estas dos características. Este hecho le sigue atribuyendo aún un carácter subjetivo al proceso, aunque en menor grado, y
- b) Un número grande de conjuntos difusos en cada variable del problema implica mayores niveles de complejidad para la extracción de reglas y de nuevo, al igual que con otros factores, puede llegar a constituirse en otro límite práctico a la extracción de reglas.

Estos dos problemas que no son exclusivos de BCD, sino que atañen a la extracción de reglas, la extracción de reglas difusas, e incluso a otros campos y enfoques como el del agrupamiento y agrupamiento difuso,¹ se pueden también considerar como una desventaja de este modelo.

8.2.3. La extracción de reglas depende del conjunto de datos de entrenamiento

El comportamiento de la red depende de los patrones que se hayan seleccionado como conjunto de entrenamiento, y si estos patrones no se pueden considerar representativos, el funcionamiento de la red no será fiable. Sin embargo este no es un problema que atañe al modelo de ex-

¹ *Clustering* y *Fuzzy Clustering* en inglés, respectivamente.

tracción de reglas, sino a la elección del conocimiento inicial para entrenamiento de la red, por lo que se puede decir que no es exclusivo de este modelo sino de todos los modelos de extracción de reglas que no trabajen con todos los patrones reales sino con solo muestras (representativas o no).

Es también importante remarcar que es esta representatividad o no de los conjuntos de datos de entrenamiento y la configuración específica del dominio del conocimiento bajo el que se esté trabajando, las que determinan en gran medida, la necesidad del uso de reglas de corte de rama que ayuden a delimitar los alcances de la generalidad de la base de reglas.

8.2.4. La extracción de reglas difusas depende también de las funciones de membresía establecidas para cada conjunto difuso

El desempeño del modelo y la validez de las reglas extraídas depende también de la forma en que se establezcan las funciones de membresía para cada conjunto difuso. En este trabajo se usaron tres modelos que extrajeron, todos, conjuntos de reglas válidas, pero distintos entre sí, y las diferencias se pueden atribuir solamente a las funciones de membresía que se aplicaron en el establecimiento de cada conjunto difuso. Esto quiere decir que la extracción de reglas proporciona solamente una de las posibles soluciones válidas para un problema, solución que puede diferir de otras logradas a partir de distintas funciones de membresía.

Así, la validez de las reglas difusas para un dominio del conocimiento depende, además, de la forma en que se conceptualizan las variables lingüísticas que las mismas contienen.

Sin embargo este problema no atañe solamente a este modelo ni es exclusivo de este enfoque de extracción de reglas difusas, sino que está relacionado con todo el proceso difuso de la información.

Anexo A

La clasificación de la planta iris

Este problema se ha constituido ya en una obra clásica de prueba para muchas disciplinas y en últimos años se ha usado frecuentemente como *problema de laboratorio* para prueba de modelos de extracción de reglas.

El problema consiste en analizar una lista de valores que dan cuenta de las características correspondientes a 150 especímenes de la planta *iris*. En el campo de la extracción de reglas, el problema original se ha modificado un poco, consiste ahora en obtener un conjunto de reglas que permitan clasificar correctamente la totalidad de los casos. La extracción de reglas se debe realizar a partir del análisis de los valores proporcionados para las características de cada ejemplar y su correspondiente clase.

El conjunto de datos del problema de clasificación de la planta *iris* contiene 150 patrones correspondientes a tres diferentes clases (*Iris Setosa*, *Iris Versicolor* e *Iris Virginica*). Hay datos de 50 ejemplares para cada una de las clases. Los datos proporcionados representan cuatro características de la planta: ancho del tallo en centímetros, largo del tallo en centímetros, ancho de la hoja en centímetros y largo de la hoja en centímetros.

Dado que es un problema ampliamente estudiado, se sabe que la primera clase es linealmente separable de las otras dos. La tabla 2 presenta las características generales descriptivas del conjunto de datos.¹

¹ El conjunto de datos completos se puede revisar el anexo A de este mismo documento.

TABLA 21. Características generales de los datos del problema de clasificación de la planta *iris*.

Variable	Mínimo	Máximo	Media	σ	Correlación entre clases
Largo del tallo	4.3	7.9	5.84	0.83	0.7826
Ancho del tallo	2.0	4.4	3.05	0.43	-0.4194
Largo del pétalo	1.0	6.9	3.76	1.76	0.9490
Ancho del pétalo	0.1	2.5	1.20	0.76	0.9565

Para poder analizar los datos del problema de clasificación de la planta *iris* con una red neuronal, los valores se utilizan para conformar vectores que se alimentan a la red como patrones del conjunto de datos. Estos vectores o patrones de entrenamiento se asocian a una salida de la red, en este caso, una clase C_i asociada con cada uno de los tipos de *iris*, que es un subconjunto nítido de \mathfrak{R}^n . La intersección entre dos clases está vacía.

En redes neuronales difusas, los valores de las características se representan por medio de conjuntos difusos y la clasificación debe asociar estos conjuntos difusos con las clases nítidas.

La extracción de reglas en una red neuronal difusa, debe, entonces, obtener un conjunto de reglas lingüísticas. Si se considera que para cada característica de entrada x_i existen q conjuntos difusos, y que la base de reglas puede contener k reglas difusas, desde R_1, \dots, R_k , entonces, R_i se obtiene si x_i pertenece al conjunto difuso μ_1 , y x_j pertenece al conjunto difuso μ_j , y x_n pertenece al conjunto difuso μ_n , y se asocia el patrón (x_1, x_2, \dots, x_n) a la clase C_i .

Los datos del problema de la planta *iris*

Datos del problema de clasificación de la planta *iris* extraídos de Fisher, A. "The Use of Multiple Measurements in Axonomic Problems", en: *Annals of Eugenics*, No. 7, 179-188, 1936, por DASL, *The Data And Story Library* de Matthew Hutcheson (ProMetrics Consulting), Michael Meyer (Carnegie-Mellon University), Paul Velleman (Cornell University) y John Walker (Syracuse University). Esta base es de uso: libre y se integra con 150 casos; para los cuales se dispone de las medidas de 4 características físicas de la planta: Ancho del Pétalo en centímetros, Largo del Pétalo en centímetros, Ancho del Tallo en centímetros, Largo del Tallo en centímetros.

Nombres de las clases de planta iris a clasificar: *Iris Setosa*, *Iris Versicolor*, *Iris Virginica*

Datos para los especímenes de la planta *iris* del tipo *Iris Setosa*

TABLA 22. Conjunto de datos de los especímenes del tipo: *Iris Setosa*

No	Ancho del pétalo (cm.)	Largo del pétalo (cm.)	Ancho del tallo (cm.)	Largo del tallo (cm.)	Clase
1	5.1	3.5	1.4	0.2	<i>Iris Setosa</i>
2	4.9	3.0	1.4	0.2	<i>Iris Setosa</i>
3	4.7	3.2	1.3	0.2	<i>Iris Setosa</i>
4	4.6	3.1	1.5	0.2	<i>Iris Setosa</i>
5	5.0	3.6	1.4	0.2	<i>Iris Setosa</i>
6	5.4	3.9	1.7	0.4	<i>Iris Setosa</i>
7	4.6	3.4	1.4	0.3	<i>Iris Setosa</i>
8	5.0	3.4	1.5	0.2	<i>Iris Setosa</i>
9	4.4	2.9	1.4	0.2	<i>Iris Setosa</i>
10	4.9	3.1	1.5	0.1	<i>Iris Setosa</i>
11	5.4	3.7	1.5	0.2	<i>Iris Setosa</i>
12	4.8	3.4	1.6	0.2	<i>Iris Setosa</i>
13	4.8	3.0	1.4	0.1	<i>Iris Setosa</i>
14	4.3	3.0	1.1	0.1	<i>Iris Setosa</i>
15	5.8	4.0	1.2	0.2	<i>Iris Setosa</i>
16	5.7	4.4	1.5	0.4	<i>Iris Setosa</i>
17	5.4	3.9	1.3	0.4	<i>Iris Setosa</i>
18	5.1	3.5	1.4	0.3	<i>Iris Setosa</i>
19	5.7	3.8	1.7	0.3	<i>Iris Setosa</i>
20	5.1	3.8	1.5	0.3	<i>Iris Setosa</i>
21	5.4	3.4	1.7	0.2	<i>Iris Setosa</i>
22	5.1	3.7	1.5	0.4	<i>Iris Setosa</i>
23	4.6	3.6	1.0	0.2	<i>Iris Setosa</i>
24	5.1	3.3	1.7	0.5	<i>Iris Setosa</i>
25	4.8	3.4	1.9	0.2	<i>Iris Setosa</i>
26	5.0	3.0	1.6	0.2	<i>Iris Setosa</i>
27	5.0	3.4	1.6	0.4	<i>Iris Setosa</i>
28	5.2	3.5	1.5	0.2	<i>Iris Setosa</i>
29	5.2	3.4	1.4	0.2	<i>Iris Setosa</i>
30	4.7	3.2	1.6	0.2	<i>Iris Setosa</i>
31	4.8	3.1	1.6	0.2	<i>Iris Setosa</i>
32	5.4	3.4	1.5	0.4	<i>Iris Setosa</i>
33	5.2	4.1	1.5	0.1	<i>Iris Setosa</i>
34	5.5	4.2	1.4	0.2	<i>Iris Setosa</i>
35	4.9	3.1	1.5	0.1	<i>Iris Setosa</i>
36	5.0	3.2	1.2	0.2	<i>Iris Setosa</i>
37	5.5	3.5	1.3	0.2	<i>Iris Setosa</i>

No	Ancho del pétalo (cm.)	Largo del pétalo (cm.)	Ancho del tallo (cm.)	Largo del tallo (cm.)	Clase
38	4.9	3.1	1.5	0.1	<i>Iris Setosa</i>
39	4.4	3.0	1.3	0.2	<i>Iris Setosa</i>
40	5.1	3.4	1.5	0.2	<i>Iris Setosa</i>
41	5.0	3.5	1.3	0.3	<i>Iris Setosa</i>
42	4.5	2.3	1.3	0.3	<i>Iris Setosa</i>
43	4.4	3.2	1.3	0.2	<i>Iris Setosa</i>
44	5.0	3.5	1.6	0.6	<i>Iris Setosa</i>
45	5.1	3.8	1.9	0.4	<i>Iris Setosa</i>
46	4.8	3.0	1.4	0.3	<i>Iris Setosa</i>
47	5.1	3.8	1.6	0.2	<i>Iris Setosa</i>
48	4.6	3.2	1.4	0.2	<i>Iris Setosa</i>
49	5.3	3.7	1.5	0.2	<i>Iris Setosa</i>
50	5.0	3.3	1.4	0.2	<i>Iris Setosa</i>

Datos para los especímenes de la planta *iris* del tipo *Iris Versicolor*

Tabla 23. Conjunto de datos del problema de clasificación de la planta iris para la clase: *Iris Versicolor*.

No	Ancho del pétalo (cm.)	Largo del pétalo (cm.)	Ancho del tallo (cm.)	Largo del tallo (cm.)	Clase
1	7.0	3.2	4.7	1.4	<i>Iris Versicolor</i>
2	6.4	3.2	4.5	1.5	<i>Iris Versicolor</i>
3	6.9	3.1	4.9	1.5	<i>Iris Versicolor</i>
4	5.5	2.3	4.0	1.3	<i>Iris Versicolor</i>
5	6.5	2.8	4.6	1.5	<i>Iris Versicolor</i>
6	5.7	2.8	4.5	1.3	<i>Iris Versicolor</i>
7	6.3	3.3	4.7	1.6	<i>Iris Versicolor</i>
8	4.9	2.4	3.3	1.0	<i>Iris Versicolor</i>
9	6.6	2.9	4.6	1.3	<i>Iris Versicolor</i>
10	5.2	2.7	3.9	1.4	<i>Iris Versicolor</i>
11	5.0	2.0	3.5	1.0	<i>Iris Versicolor</i>
12	5.9	3.0	4.2	1.5	<i>Iris Versicolor</i>
13	6.0	2.2	4.0	1.0	<i>Iris Versicolor</i>
14	6.1	2.9	4.7	1.4	<i>Iris Versicolor</i>
15	5.6	2.9	3.6	1.3	<i>Iris Versicolor</i>
16	6.7	3.1	4.4	1.4	<i>Iris Versicolor</i>
17	5.6	3.0	4.5	1.5	<i>Iris Versicolor</i>
18	5.8	2.7	4.1	1.0	<i>Iris Versicolor</i>
19	6.2	2.2	4.5	1.5	<i>Iris Versicolor</i>
20	5.6	2.5	3.9	1.1	<i>Iris Versicolor</i>
21	5.9	3.2	4.8	1.8	<i>Iris Versicolor</i>
22	6.1	2.8	4.0	1.3	<i>Iris Versicolor</i>
23	6.3	2.5	4.9	1.5	<i>Iris Versicolor</i>
24	6.1	2.8	4.7	1.2	<i>Iris Versicolor</i>

No	Ancho del pétalo (cm.)	Largo del pétalo (cm.)	Ancho del tallo (cm.)	Largo del tallo (cm.)	Clase
25	6.4	2.9	4.3	1.3	<i>Iris Versicolor</i>
26	6.6	3.0	4.4	1.4	<i>Iris Versicolor</i>
27	6.8	2.8	4.8	1.4	<i>Iris Versicolor</i>
28	6.7	3.0	5.0	1.7	<i>Iris Versicolor</i>
29	6.0	2.9	4.5	1.5	<i>Iris Versicolor</i>
30	5.7	2.6	3.5	1.0	<i>Iris Versicolor</i>
31	5.5	2.4	3.8	1.1	<i>Iris Versicolor</i>
32	5.5	2.4	3.7	1.0	<i>Iris Versicolor</i>
33	5.8	2.7	3.9	1.2	<i>Iris Versicolor</i>
34	6.0	2.7	5.1	1.6	<i>Iris Versicolor</i>
35	5.4	3.0	4.5	1.5	<i>Iris Versicolor</i>
36	6.0	3.4	4.5	1.6	<i>Iris Versicolor</i>
37	6.7	3.1	4.7	1.5	<i>Iris Versicolor</i>
38	6.3	2.3	4.4	1.3	<i>Iris Versicolor</i>
39	5.6	3.0	4.1	1.3	<i>Iris Versicolor</i>
40	5.5	2.5	4.0	1.3	<i>Iris Versicolor</i>
41	5.5	2.6	4.4	1.2	<i>Iris Versicolor</i>
42	6.1	3.0	4.6	1.4	<i>Iris Versicolor</i>
43	5.8	2.6	4.0	1.2	<i>Iris Versicolor</i>
44	5.0	2.3	3.3	1.0	<i>Iris Versicolor</i>
45	5.6	2.7	4.2	1.3	<i>Iris Versicolor</i>
46	5.7	3.0	4.2	1.2	<i>Iris Versicolor</i>
47	5.7	2.9	4.2	1.3	<i>Iris Versicolor</i>
48	6.2	2.9	4.3	1.3	<i>Iris Versicolor</i>
49	5.1	2.5	3.0	1.1	<i>Iris Versicolor</i>
50	5.7	2.8	4.1	1.3	<i>Iris Versicolor</i>

Datos para los especímenes de la planta *iris* del tipo *Iris Virginica*

TABLA 24. Conjunto de datos del problema de clasificación de la planta iris para la clase: *Iris Virginica*.

No	Ancho del pétalo (cm.)	Largo del pétalo (cm.)	Ancho del tallo (cm.)	Largo del tallo (cm.)	Clase
1	6.3	3.3	6.0	2.5	<i>Iris Virginica</i>
2	5.8	2.7	5.1	1.9	<i>Iris Virginica</i>
3	7.1	3.0	5.9	2.1	<i>Iris Virginica</i>
4	6.3	2.9	5.6	1.8	<i>Iris Virginica</i>
5	6.5	3.0	5.8	2.2	<i>Iris Virginica</i>
6	7.6	3.0	6.6	2.1	<i>Iris Virginica</i>
7	4.9	2.5	4.5	1.7	<i>Iris Virginica</i>
8	7.3	2.9	6.3	1.8	<i>Iris Virginica</i>
9	6.7	2.5	5.8	1.8	<i>Iris Virginica</i>
10	7.2	3.6	6.1	2.5	<i>Iris Virginica</i>
11	6.5	3.2	5.1	2.0	<i>Iris Virginica</i>

No	Ancho del pétalo (cm.)	Largo del pétalo (cm.)	Ancho del tallo (cm.)	Largo del tallo (cm.)	Clase
12	6.4	2.7	5.3	1.9	<i>Iris Virginica</i>
13	6.8	3.0	5.5	2.1	<i>Iris Virginica</i>
14	5.7	2.5	5.0	2.0	<i>Iris Virginica</i>
15	5.8	2.8	5.1	2.4	<i>Iris Virginica</i>
16	6.4	3.2	5.3	2.3	<i>Iris Virginica</i>
17	6.5	3.0	5.5	1.8	<i>Iris Virginica</i>
18	7.7	3.8	6.7	2.2	<i>Iris Virginica</i>
19	7.7	2.6	6.9	2.3	<i>Iris Virginica</i>
20	6.0	2.2	5.0	1.5	<i>Iris Virginica</i>
21	6.9	3.2	5.7	2.3	<i>Iris Virginica</i>
22	5.6	2.8	4.9	2.0	<i>Iris Virginica</i>
23	7.7	2.8	6.7	2.0	<i>Iris Virginica</i>
24	6.3	2.7	4.9	1.8	<i>Iris Virginica</i>
25	6.7	3.3	5.7	2.1	<i>Iris Virginica</i>
26	7.2	3.2	6.0	1.8	<i>Iris Virginica</i>
27	6.2	2.8	4.8	1.8	<i>Iris Virginica</i>
28	6.1	3.0	4.9	1.8	<i>Iris Virginica</i>
29	6.4	2.8	5.6	2.1	<i>Iris Virginica</i>
30	7.2	3.0	5.8	1.6	<i>Iris Virginica</i>
31	7.4	2.8	6.1	1.9	<i>Iris Virginica</i>
32	7.9	3.8	6.4	2.0	<i>Iris Virginica</i>
33	6.4	2.8	5.6	2.2	<i>Iris Virginica</i>
34	6.3	2.8	5.1	1.5	<i>Iris Virginica</i>
35	6.1	2.6	5.6	1.4	<i>Iris Virginica</i>
36	7.7	3.0	6.1	2.3	<i>Iris Virginica</i>
37	6.3	3.4	5.6	2.4	<i>Iris Virginica</i>
38	6.4	3.1	5.5	1.8	<i>Iris Virginica</i>
39	6.0	3.0	4.8	1.8	<i>Iris Virginica</i>
40	6.9	3.1	5.4	2.1	<i>Iris Virginica</i>
41	6.7	3.1	5.6	2.4	<i>Iris Virginica</i>
42	6.9	3.1	5.1	2.3	<i>Iris Virginica</i>
43	5.8	2.7	5.1	1.9	<i>Iris Virginica</i>
44	6.8	3.2	5.9	2.3	<i>Iris Virginica</i>
45	6.7	3.3	5.7	2.5	<i>Iris Virginica</i>
46	6.7	3.0	5.2	2.3	<i>Iris Virginica</i>
47	6.3	2.5	5.0	1.9	<i>Iris Virginica</i>
48	6.5	3.0	5.2	2.0	<i>Iris Virginica</i>
49	6.2	3.4	5.4	2.3	<i>Iris Virginica</i>
50	5.9	3.0	5.1	1.8	<i>Iris Virginica</i>

Anexo B

Conjunto de reglas iniciales extraídas por el Modelo de extracción de reglas difusas basado en la Binarización de los Conjuntos Difusos (BCD)

Conjunto de reglas iniciales extraídas por BCD para la clase 1 (*Iris Setosa*)

TABLA 25. Conjunto de reglas iniciales extraídas por BCD para la clase 1.

Número	Antecedentes										Conclusión			
1	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
2	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
3	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 1
4	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
5	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 1
6	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 1
7	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 1
8	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
9	Si	V1	Medio	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
10	Si	V1	Medio	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 1
11	Si	V1	Medio	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
12	Si	V1	Medio	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 1
13	Si	V1	Medio	Y	V2	Grande	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 1
14	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
15	Si	V1	Grande	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
16	Si	V1	Grande	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Pequeño	Entonces	Clase 1
17	Si	V1	Grande	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 1

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo

Conjunto de reglas iniciales extraídas por BCD para la clase 2 (*iris Versicolor*)

TABLA 26. Conjunto de reglas iniciales extraídas por BCD para la clase 2.

No	Antecedentes										Conclusión			
1	Si	V1	Grande	Y	V2	Grande	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
2	Si	V1	Medio	Y	V2	Grande	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
3	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
4	Si	V1	Grande	Y	V2	Medio	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
5	Si	V1	Medio	Y	V2	Medio	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
6	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
7	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
8	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
9	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Medio	Entonces	Clase 2
10	Si	V1	Grande	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 2

No	Antecedentes										Conclusión			
11	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 2
12	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 2
13	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Medio	Entonces	Clase 2
14	Si	V1	Grande	Y	V2	Grande	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 2
15	Si	V1	Grande	Y	V2	Medio	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 2
16	Si	V1	Medio	Y	V2	Medio	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 2
17	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 2
18	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 2
19	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 2
20	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Pequeño	Entonces	Clase 2

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo

Conjunto de reglas iniciales extraídas por BCD para la clase 3 (*Iris Virginica*)

TABLA 27. Conjunto de reglas iniciales extraídas
por BCD para la clase 3.

No	Antecedentes										Conclusión			
1	Si	V1	Grande	Y	V2	Grande	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
2	Si	V1	Medio	Y	V2	Grande	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
3	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
4	Si	V1	Grande	Y	V2	Medio	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
5	Si	V1	Medio	Y	V2	Medio	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
6	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
7	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
8	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
9	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Grande	Entonces	Clase 3
10	Si	V1	Grande	Y	V2	Grande	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
11	Si	V1	Medio	Y	V2	Grande	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
12	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
13	Si	V1	Grande	Y	V2	Medio	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
14	Si	V1	Medio	Y	V2	Medio	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
15	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
16	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
17	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
18	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Medio	Y	V4	Grande	Entonces	Clase 3
19	Si	V1	Grande	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
20	Si	V1	Medio	Y	V2	Grande	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
21	Si	V1	Grande	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
22	Si	V1	Medio	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
23	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
24	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
25	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
26	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Pequeño	Y	V4	Grande	Entonces	Clase 3
27	Si	V1	Grande	Y	V2	Grande	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
28	Si	V1	Medio	Y	V2	Grande	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
29	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
30	Si	V1	Grande	Y	V2	Medio	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
31	Si	V1	Medio	Y	V2	Medio	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
32	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
33	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
34	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
35	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Medio	Entonces	Clase 3
36	Si	V1	Grande	Y	V2	Grande	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
37	Si	V1	Medio	Y	V2	Grande	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
38	Si	V1	Pequeño	Y	V2	Grande	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
39	Si	V1	Grande	Y	V2	Medio	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
40	Si	V1	Medio	Y	V2	Medio	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
41	Si	V1	Pequeño	Y	V2	Medio	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
42	Si	V1	Grande	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
43	Si	V1	Medio	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3
44	Si	V1	Pequeño	Y	V2	Pequeño	Y	V3	Grande	Y	V4	Pequeño	Entonces	Clase 3

V1: largo de tallo; V2: Ancho de tallo; V3: Largo de pétalo; V4: Ancho de pétalo

Bibliografía

- Ajjanagadde, Venkat y Lokenda Shastri, "Reasoning with Rules and Variables in Neural Networks" en: *Intelligent Hybrid Systems*, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 209-220.
- Alahakoon, D, S. K. Halgamuge y B. Srinivasan, "Dynamic Self Organising Maps with Controlled Growth for Knowledge Discovery", *IEEE Transactions on Neural Networks (Special Issue in Data Mining)*, 2000.
- Alexander, Jay A y Michael C. Mozer, "Template-based algorithms for Connectionist rule extraction" Research Report, Department of Computer Science and Institute for Congnitive Science, University of Colorado, 1996.
- Andrews R., Tickle A.B., & Diederich J., A Review of Techniques for Extracting Rules From Trained Artificial Neural Networks, in *Clinical Applications of Articial Neural Networks*, R.Dybowski & V.Grant (eds), Cambridge University Press, Cambridge UK, 2001.
- Andrews, R., J. Diederich y A. Tickle, "A survey and a critique of techniques for extracting rules from trained artificial neural networks", Queensland, Australia, Queensland University of Technology, Technical Report, Neurocomputing Research Center, 1995. 37 páginas.
- Andrews, R., R. Cable, J. Diederich, S. Geva, M. Golea, R. Hayward, C. Ho-Stuart y A. Tickle, An Evaluation and Comparison of Techniques for Extracting and Refining Rules From Artificial Neural Networks, Queensland, Australia, Queensland University of Technology, QUT Meritorious Project 1995 Technical Report, Neurocomputing Research Center, Marzo de 1996. 53 páginas.
- Bandemer, Hans y Siegfried Gottwald, *Fuzzy Sets, Fuzzy Logic, Fuzzy Methods, with applications*, Chichester, John Wiley & Sons, 1995.

- Berenji, H. R. "Refinement of Aproximate reasoning-based Controllers by reinforcement learning" presentado en: 8th International Machine Learning Workshop, Evaston Illinois, 1991, 5 páginas.
- Buchanan B y E. Shortliffe, Rule Based Expert Systems, Reading Massachusetts, Addison Wesley, 1984.
- Chang A.M. y L.O. Hall, "The Validation of Fuzzy Knowledge-based Systems", en: Fuzzy Logic for the Management of Uncertainty, Zadeh, Lofti. A. Y J. Kacprzyk editores, Nueva York, John Wiley & Sons. Inc., 1992, págs. 589 a 606.
- Chang B y A. Rathnaweera and S. K. Halgamuge and H. C. Watson, "Particle Swarm Optimisation for Protein Motif Discovery", Journal of Genetic Programming and Evolvable Machines, Kluwer Junio del 2004.
- Chang B y S. K. Halgamuge, "Approximate Symbolic Pattern Matching for Protein Sequence Data", International Journal of Approximate Reasoning, Elsevier, Vol 32, Febrero del 2003.
- Chang B, y S. K. Halgamuge, "Protein Motif Extraction with Neuro-Fuzzy Systems", Bioinformatics, Oxford University Press, vol 18, pp 1084-1090, 2002.
- Chang B., A. Rathnaweera, S. K. Halgamuge y H. C. Watson, "Particle Swarm Optimisation for Protein Motif Discovery", Journal of Genetic Programming and Evolvable Machines, Kluwer Junio del 2004.
- Cohen, Paul R. Empirical Methods for Artificial Intelligence, Cambridge Mass, The MIT Press, 1995.
- Craven, Mark W y Jude W. Shavlik. "Extracting Comprehensible Concept Representations from trained Neural Networks", en: Working Notes of the IJCAI 1995 Workshop on Comprehensibility in Machine Learning, 1995, 15 págs.
- Craven, Mark W y Jude W. Shavlik. "Learning symbolic rules using artificial neural networks", en: Machine Learning, Proceedings of the Tenth International Conference, 1993, San Francisco California, P.E. Utgoff editores, Morgan Kaufmann, 1993, 8 págs.
- Craven, Mark W y Jude W. Shavlik. "Using samples and queries to extract rules from trained neural networks", en: Machine Learning, Proceedings of the Eleventh International Conference, 1994, San Francisco California, Cohen and Hirsh editores, Morgan Kaufmann, 1994, 9 págs.
- Craven, Mark W. Extracting Comprehensible Models from traided Neural Networks, Madison, Universidad de Wisconsin, 1996.

- Dawson, Michael R.W. y Kevin S. Shamanski, "Connectionism, Confusion, and Cognitive Science", en: *The Journal of Intelligent Systems*, 1994, 4, 215-262.
- DeKleer, J. "An Assumption based TMS", en: *Artificial Intelligence*, Número 28, págs 127- 162.
- Dreyfus, Hubert L. "The limits of Artificial Intelligence" en: "What Computers still can't do: A critique of Artificial Reason, Cambridge, Massachusetts Institute of Technology Press, págs 285-305, 1993.
- Duda y Reboh, 1983 citado por gallant PROSPECTOR
- Feigenbaum, E. Y A. Barr, *The Handbook of Artificial Intelligence*, Nueva York, Addison Wesley, 1982.
- Fisher, R. A. Datos del problema de clasificación de la planta iris extraídos de "The Use of Multiple Measurements in Axonomic Problems", en: *Annals of Eugenics*, No. 7, 179-188, 1936, por DASL, The Data And Story Library de Matthew Hutcheson (ProMetrics Consulting), Michael Meyer (Carnegie-Mellon University), Paul Velleman (Cornell University) y John Walker (Syracuse University)
- Foltin, Craig y Linda Garceau, "Beyond Expert Systems, Neural Networks in Accounting", en *National Public Accountant (NPA)*, Vol 4, No 6, Junio 1996, pág 26-30.
- Foslien, Wendy y Tariq Samad "Fuzzy Controller Synthesis with Neural Networks Process Models" en: *Intelligent Hybrid Systems*, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 23-46.
- Fox, J. "Towards a reconciliation of fuzzy logic and standard logic", en: *International Journal of Man-Machine Studies*, Vol. 15, 1981, págs, 213 a 220.
- Frydenberg, M y S. Gallant, "Fuzzyness and Expert System generation" en: *Uncertainty in Knowledge-Based Systems*, Lecture Notes in Computer Science 286, de B. Bouchon y R. Yager (editores), Springer-Verlag, págs. 137-143, 1987.
- Fu. L. M. "Rule generation from Neural Networks", en: *IEEE Transactions on Systems, Man and Cybernetics*, Vol 8, No 18, págs. 1114-1124.
- Fullér, Robert, *Neural Fuzzy Systems*, Åbo, Turku Center of Computer Science, Åbo Academy, 1995.
- Gallant, Stephen I. "A Connectionist Learning Algorithm with Probable Generalization and Scaling Bounds" en: *Neural Network 3*, Mimeo, ITESM/MCC, 1990.

- Gallant, Stephen I. "Connectionist Expert Systems" en: Communications of the ACM, No 31, Mimeo, ITESM-Campus Morelos/MCC, 1990.
- Gallant, Stephen I. "Perceptrón-Based Learning Algorithms", en IEEE Transactions on Neural Networks, Vol. 1. Num. 2, págs. 179 a 191, 1990.
- Gallant, Stephen I. Neural Network Learning and Expert Systems, Cambridge, The MIT Press, 1994.
- Geissmann, J.R y R. Shultz "Verification and Validation of Expert Systems", en: AI Expert Magazine, Febrero de 1988, edición especial Expert Systems, 10 páginas.
- Glatzer, Hal, "Neural Computers go Practical", en: Software Magazine, USA, 26 de septiembre de 1988.
- Goh T.H, P.Z. Wang y H.C. Loui "Learning Algorithm for the Enhanced Fuzzy Perceptron", Technical report of Institute of Systems Science, National University of Singapore, Heng Mui Keng Terrace, Singapore, diciembre de 1991.
- Goonatilake, Suran y Sukhndev Khebbal, "Intelligent Hybrid Systems: Issues, Classifications and Future directions", en: Intelligent Hybrid Systems, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 1-22.
- Graubard, Stephen R. El nuevo debate sobre la Inteligencia Artificial. Sistemas Simbólicos y redes neuronales, Barcelona, Gedisa, 1993.
- Gruau, Frederic, "Genetic Programming of Neural Networks: Theory and Practice" en: Intelligent Hybrid Systems, de Suran Goonatilake y Sukhdev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 245-272.
- Gupta, M.M., y J. Qi, "On Fuzzy Neuron Models" en: Fuzzy Logic for the Management of Uncertainty, Zadeh, Lofti. A. Y J. Kacprzyk editores, Nueva York, John Wiley & Sons. Inc., 1992, págs. 479 a 493.
- Guru, S. M., S. Fernando, S. K. Halgamuge, y K. Chan, "Intelligent fastening with SMARTBOLT Technology and Sensor Network", Assembly Automation Journal (special issue on Mechatronics), UK, Octubre del 2004.
- Haack, Susan, "Do we need Fuzzy Logic", en: International Journal of Man-Machine Studies, 1979.
- Halgamuge S. K y M. Glesner "Neural Networks in designing of fuzzy systems for real world applications" enviado para su publicación en: Fuzzy Sets and Systems, 1994, 12 páginas.

- Halgamuge S. K, Werner Poechmueller y M. Glesner “An alternative approach for generation of membership functions and fuzzy rules based on radial and cubic basis function networks” en: International Journal of Approximate Reasoning, Nueva York, Elsevier, 1994, Número XXX,(7) 22 páginas.
- Halgamuge, S K “A Trainable Transparent Universal Approximator for Defuzzification in Mamdani Type Neuro-Fuzzy Controllers”, IEEE Transactions on Fuzzy Systems, Mayo del 1998.
- Halgamuge, S K “Self Evolving Neural Networks for Rule Based Data Processing”, IEEE Transactions on Signal Processing, Noviembre de 1997.
- Halgamuge, S K y M. Glesner, “Neural Networks in Designing Fuzzy Systems for Real World Applications”, International Journal for Fuzzy Sets and Systems, Vol 65, No. 1, pp 1-12, 1994.
- Halgamuge, S. K. “Realtime Directed Data Mining with Structure Adapting Neuro-Fuzzy Networks”, Journal on Integrated Computer Aided Engineering, IOS Press, Octubre del 2003.
- Hall, L y A. Kandel, “The Evolution from Expert Systems” en: Fuzzy Expert Systems, USA, CRC Press, Capítulo 1.
- Haugeland, John, La inteligencia Artificial, México, Editorial Siglo XXI, 1988.
- Hayashi, Y, Atsushi Imura y Katsumi Yoshida. “A Neural Expert System Under Uncertain Environments and its evaluation” en: Proceedings of the 11th. Knowledge and Intelligence System Symposium, Tokio, Japón, 1990, págs. 13-18.
- Hayashi, Y, Atsushi Imura y Katsumi Yoshida. “Fuzzy Neural Expert System and its application to medical diagnosis” en: Proceedings of the 8th. International Congress of Cybernetics and systems, Junio 11-15 de 1990, Nueva York, Constantine N. Manikopoulos (Editor) Department of Electrical and Computer Engineering, New Jersey Institute of Technology.
- Hayashi, Y. “Neural Expert System Using Fuzzy Teaching Input and its application to Medical Diagnosis”, en: Information Sciences, No. 1, 47-58(1994) Elsevier Science Inc. 1069-0115/94, 1994.
- Hayashi, Y. A. “A Neural Network Expert System with Automated Extraction of Fuzzy Expert System with Automated Extraction of fuzzy if-then rules and its application to medical diagnosis” en: Advances in Neural Information Processing Systems de D.S. Touretzky y R. Lippman (compiladores), San Mateo California, Morgan Kaufmann Publishers, Vol 3, 1991, págs 578-584.

- Haykin, Simon Neural networks. A Comprehensive fundation, Nueva York, McMillan College Publishing Company, 1994.
- Hebb, Donald O. The Organization of Behavior, Nueva York, Wiley. Introducción, Capítulo III y Capítulo IV.
- Hecht-Nielsen, Robert, "Theory of the back propagation neural network" en: Proceedings of the 1988 INNS Meeting, septiembre de 1988, págs 593-602.
- Hecht-Nielsen, Robert, Neurocomputing, Reading, Addison-Wesley Publishing Company, 1989
- Herrmann, Christoph S. "A hibrid Fuzzy-neural Expert System for Diagnósis", en International Joint Conference on Artificial Intelligence (IJCAI)., Morgan Kaufmann, 1995, págs 494-500.
- Hinton, G.E. "Learning distributed representations of concepts" en: Proceedings of the 8th, Annual Conference of the Cognitive Science Society, 1986, págs 1-12.
- Hollstein, T., Halgamuge, S.K. y Glesner, M., "Computer-aided design of fuzzy systems based on generic VHDL specifications", IEEE Transactions on Fuzzy Systems, Vol. 4, pp 403-417 Noviembre de 1996.
- Honavar, Vasant y Leonard Uhr, "Integrating Symbol Processing Systems and Connectionist Networks", en: Intelligent Hybrid Systems, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 177-208.
- Hopfield, J.J. "Neural Networks and Physical systems with emergent collective computational abilities" en: Proceedings of The National Academy of Sciences, No. 79, págs. 2554- 2558.
- Horikawa S., Fusuhashi T y Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm", publicado en: IEEE Transactions on Neural Networks, Vol 3, No. 5, (septiembre de 1992) págs. 801-806.
- Hsu A y S. K. Halgamuge, "Enhancement of Topology Preservation and Hierarchical Dynamic Self-Organising Maps for Data Visualisation", Elsevier, Vol 32, No 2-3, Febrero del 2003.
- Hsu A, S. Tang y S. K. Halgamuge, "An Unsupervised Hierarchical Dynamic Self-Organising Approach to Class Discovery and Marker Gene Identification in Microarray Data", Bioinformatics, Oxford University Press, Noviembre del 2003.
- Jackson, Peter , Introduction to Expert Systems, Wokingham, Inglaterra, Addison-Wesley Publishing Company, 1990.

- Jager, René, *Fuzzy Logic in control*, Technische Universeit Delft, ISBN 90-9008318-9, 1995.
- James A. Freeman y David M. Skapura *Neural Networks, Algorithms, Applications and Programming Techniques*, Reading massachusetts, Addison-Wesley Publishing Company, 1991.
- Kandell, E R, J H Schwartz y M Jessell, *Principles of Neuronal Science*, Mc Graw Hill, 2000 4ta Edición.
- Karr, Charles, “Genetic Algorithms and Fuzzy Logic for Adaptative Process Control” en: *Intelligent Hybrid Systems*, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 63-84.
- Kasabov, Nikola, R. Kozma y W. Duch, “Rule Extraction from Linguistic Rule Networks and from Fuzzy Neural Networks: Propositional versus Fuzzy Rules” en: *Proceedings of the Conference on Neural Networks and Their Applications NEURAP’98*, Marsella, Francia, 11 al 13 Marzo de 1998, págs 403-406.
- Kasabov, Nikola. “Fuzzy neural networks, rules extraction and fuzzy synergistic reasoning” (Mimeo) documento a aparecer en el *International Journal of Systems, Research and Information Science* (1998)
- Kaski, S. *Data exploration using self-organizing maps*, Número 82 de la serie *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series*, Finnish Academy of Technology, Espoo, 1997, 57 págs.
- Keller, James M y Douglas J. Hunt, “Incorporating Fuzzy Membership Functions in to the Perceptron Algorithm” en: *Fuzzy Models for Pattern Recognition*, editado por James C. Bezdeck y Sankar K. Pal, Nueva Jersey, IEEE Press, 1992.
- Kelly, Wallace E. III y John H. Painter, “Hipertrapezoidal Fuzzy Membership Functions”, en: *Fifth IEEE International Conference on Fuzzy Systems*, Nueva Orleans, Septiembre 8 de 1996, pp. 1279-1284.
- Kerber Randy, Brian Livesey y Evangelos Simoudis, “A Hybrid System for Data Mining”, en: *Intelligent Hybrid Systems*, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 121-142.
- Kilmasauskas, Casimer C. “Using fuzzy pre-proccesing with neural networks for chemical porcess diagnostic problems” en en: *Intelligent Hybrid Systems*, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 143-152.

- Koggalage R, y S. K. Halgamuge, “Reducing the number of training samples for Fast Support Vector Machine Classification”, *Neural Information Processing Letters and Reviews*, KAIST Press, Vol 2, No 3, 2004.
- Koggalage, R, S. K. Halgamuge y A. Hsu, “Unsupervised Class Discovery and Feature Selection using an Improved Hierarchical Dynamic Self-Organizing Map”, *Neural Information Processing Letters and Reviews*, KAIST Press, Vol 2, No 4, 2004.
- Kohonen, Teuvo, “Self Organized formation of topologically correct feature maps”, en: *Biological Cybernetics* No. 43, págs. 59-69.
- Kosko, Bart y Satoru Isaka, “Fuzzy Logic”, en: *Scientific American*, Nueva York, 204/Septiembre de 1993, págs. 60 – 65.
- Kosko, Bart, “Fuzzy Systems as universal approximators”, reprint IEEE, 0-7803-0236-2/92, 1992.
- Kosko, Bart, *Fuzzy Engineering*, Nueva Jersey, Prentice Hall, 1997.
- Kosko, Bart, *Neural Networks and Fuzzy Systems. A dynamical Systems Approach to Machine Intelligence*, Nueva Jersey, Prentice Hall, 1992.
- Lpimann, Richard P. “An introduction to computing with Neural Nets”, en: *IEEE ASSP Magazine*, USA, IEEE, Abril de 1987, págs 4 -22.
- Maravall Gómez_Allende, Darío, *Reconocimiento de formas y visión artificial*, Wilmington, Delaware, Addison-Wesley Iberoamericana/RAMA, 1994.
- Masuoka, R., N. Watanabe , A. Kamura, Y. Owada y K. Asakawa “Neurofuzzy systems – Fuzzy inference using a structured neural network” presentada en: *International Conference on Fuzzy Logic and Neural Networks*, Izuka, Japón, 1990, 5 páginas.
- McClelland, James L , David E. Rummelhart y Geoffrey Hinton “Representaciones distribuidas”, en *Filosofía de la Inteligencia Artificial*, compilación de Margaret Boden, México, Fondo de Cultura Económica, 1994.
- McClelland, James L y David E. Rummelhart, “An interactive activation model of context effects in letter perception. Part 1: An account of basic findings”, en *Psychological Review*, No. 88, 1981.
- McCulloch, Warrn S. Y Walter Pitts, “A logical Calculus of the ideas immanent in nervous activity”, en: *Bulletin of Mathematical Biophysics*, Numero 5, págs 115-133.
- McMillan, Clayton, Michael C. Mozer y Paul Smolensky, “Rule induction through integrated symbolic and subsymbolic processing”, en: *Ad-*

- vances in Neural Information Processing Systems IV, de J.E. Moody, S.J. Hanson y R.P. Lipmann (editores), 1993, San Mateo California, Morgan Kaufmann Publishers, 8 páginas.
- Medsker, Larry y Jay Liebowitz, Design and development of Expert Systemns and Neural Networks, Nueva York, Macmillan Publishing Company, 1993.
- Merrit, Dennis, Building Expert Systems in PROLOG, Nueva York, Springer-Verlag, 1987.
- Michalski, R. S “Understanding the nature of learning: Issues and research directions”, en Machine Learning. An Artificial Intelligence Approach de R.S. Michalski, J. Carbonell y T. M. Mitchell, 1986, Los Altos California, Morgan Kaufmann.
- Minh Pham, Khai, “Distributed Processing and learning” en: Intelligent Hybrid Systems, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 221-245.
- Minsky, Marvin “Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy”, en: Artificial Intelligence at MIT.Expanding Frontiers, Patrick H. Winston (editor), Vol 1, Masachussetts, MIT Press, 1990. Reimpreso en AI Magazine, 1991
- Montana, David J. “Neural Network Weight Selection Using Genetic Algorithms”, en: Intelligent Hybrid Systems, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 85-104.
- Nauck D y R Kruse. A Neuro-Fuzzy Method to Learn Fuzzy Classification Rules from Data. Fuzzy Sets and Systems, 89:277-288, 1997.
- Nauck A y R Kruse. Neuro-Fuzzy Systems for Function Approximaton. Fuzzy Sets and Systems, 101:261-271, 1999.
- Nauck D y R Kruse. NEFCLASS-X: A Neuro-Fuzzy Tool to Build Readable Fuzzy Classifiers. BT Technology Journal, 16(3):180-192, 1998.
- Nauck, D Adaptive Rule Weights in Neuro-Fuzzy Systems. Neural Computing and Applications, 9:60-70, 2000.
- Nauck, D Fuzzy Data Analysis with NEFCLASS. Int. J. Approximate Reasoning. 2002.
- Nauck, D Neuro-Fuzzy Systems for Explaining Data Sets. Proc. NAFIPS-FLINT 2002, New Orleans, 2002.
- Nauck, D y R Kruse. Obtaining Interpretable Fuzzy Classification Rules from Medical Data. Artificial Intelligence in Medicine, 16:149-169, 1999.

- Nauck, Detlef y Rudolf Kruse, “Choosing appropriate neuro-fuzzy models”, en: Proceedings of the 2d European Congress on Intelligent Techniques and Soft Computing (EUFIT’94), Aachen, Agosto, 1994, págs. 552 a 557.
- Nauck, Detlef y Rudolf Kruse, “NEFCLASS A neurofuzzy approach for the classification of data”, en: Proceedings of the 1995 ACM Symposium on applied computing, Nashville, Tennessee, ACM Press, febrero 26-28 de 1995, 5 páginas.
- Nauck, Detlef, “Beyond neuro-fuzzy: perspectives and directions”, en: Proceedings of the Third European Congress on Intelligent Techniques and Soft Computing (EUFIT’95), Aachen, Agosto 28-31 de 1995, págs 1159 a 1164.
- Nauck, Detlef, Frank Klawonn y Rudolf Kruse, “Fuzzy sets, fuzzy controllers, and neural networks” en: Scientific Journal of the Humboldt University of Berlin, serie Medicine 41, número 4, 1992, págs 99 a la 120.
- Nauck, Detlef, Frank Klawonn y Rudolf Kruse, “Generating rules from data by fuzzy and neuro-fuzzy methods”, en: Proceedings of the Third German GI-Workshop Fuzzy neurosysteme’95, Darmstad, Alemania, Noviembre 15-17 de 1995, 9 páginas.
- Nauck, Detlef, Ulrike Nauck y Rudolf Kruse, “Generating Classification rules with the neuro-fuzzy system NEFCLASS” en: Proceedings of The Biennial Conference of the North American Fuzzy Information Processing Society (NFAIPS’96), Berkeley, 1996, 6 páginas.
- Nürnbergger, A D Nauck y R Kruse. Neuro-Fuzzy Control Based on the NEFCON-Model: Recent Developments. *Soft Computing*, 2(4):168-182, 1999.
- Ojala, Tommi, *Neuro-fuzzy Systems in Control*, Tampere University of Technology, Department of Electrical Engineering, Master of Science Thesis, 1995.
- Omlin, Christian W y C. Lee Giles, “extraction of rules from discrete-time recurrent neural networks”, NEC Research Institute, documento presentado para su publicación en *Neural Networks*, 1996.
- Pagels, Heinz R. *Los sueños de la razón, el ordenador y los nuevos horizontes de las ciencias de la complejidad*, México, CONACyT/GE-DISA, 1991.
- Parker, D.B. *Learning Logic* Technical Report 47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, 1985.

- Patchararungruang, S, S.K. Halgamuge y N. Shenoy, "Optimized Rule-Based Delay Proportion Adjustment for Proportional Differentiated Services", IEEE Journal on Selected Areas in Communications (Special Issue on Intelligent Services and Applications in Next generation networks), IEEE Press, 2004.
- Pedrycz, Witold, Fuzzy Control and Fuzzy Systems, Londres, Research Studies Press Ltd., 1993.
- Piaget, Jean, El nacimiento de la inteligencia en el niño, México, Grijalbo/Consejo Nacional Para la Cultura y las Artes, 1985.
- Pollack J. M. Bedau, Husbands Ph. T. Ikegami y RA Watson, *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, USA, Bradford, 2004.
- Powell, David J. Michael M. Skolnick y Shi Shing Tong, "A Unified Approach for Engineering Design", en: Intelligent Hybrid Systems, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 107-120.
- Powell, Donna, "Network management With expert Systems" en ENGM 624 Knowledge Engineering, USA, 1996.
- Rathnaweera, A S. K. Halgamuge y H. C. Watson, "Self-Organizing Hierarchical Particle Swarm Optimizer with time varying acceleration coefficients", IEEE Transactions on Evolutionary Computation, Junio del 2004, IEEE Press.
- Rattasiri W y S. K. Halgamuge, "Computationally Advantageous and Stable Hierarchical Fuzzy Systems for Active Suspensions", IEEE Transactions on Industrial Electronics, IEEE Press, Vol 50, No 1, Febrero del 2003.
- Rich, Elaine y Kevin Knight, Artificial Intelligence, McGraw-Hill, 1991.
- Rosenblatt, Frank "The perceptron: a probabilistic model for information storage and organization in the brain" en: Psychological Review No. 65, págs 386-408.
- Rummelhart, D. E., G.E. Hinton y R.J. Williams, "Learning internal representations by error propagation" en: Paralell Distributed Processing: Exploration in the Microstructures of cognition, compilación de Rummelhart y McClelland, Cambridge, MIT Press, pp. 318-362. Vol. I,
- Rummelhart, D. E., G.E. Hinton y R.J. Williams, "Learning internal representations by back-propagating errors" en: Nature Numero 323, págs 533-536.

- Ruyer, Raymond, *La Cibernética y el origen de la información*, México, Ediciones del Fondo de Cultura Económica, Colección Popular, 1984.
- Saad R. y S. K. Halgamuge, “Stability of Hierarchical Fuzzy Systems Generated by Neuro-Fuzzy”, *Journal on Soft Computing, A Fusion of Foundations, Methodologies, and Applications*, Springer, Vol 8, No 6, 2004.
- Saito, K, y R. Nakano, “Medical Diagnostic Expert System based on PDP model”, en: *International Conference on Neural Networks, Proceedings*, San Diego, IEEE, Vol 1, págs 255-262.
- Sánchez y Beltrán, J.P., *Sistemas expertos, una metodología de programación*, México, Macrobit Editores, 1990.
- Scherer, Andreas y Gunter Schlageter, “A Multi-Agent Approach for Integration of Neural Networks and Expert Systems”, en: *Intelligent Hybrid Systems*, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 153-174.
- Skapura, David M. *Building Neural Networks*, Nueva York, Nueva York, ACM Press, 1995.
- Taber, R y M. Siegel, “Estimation of Expert Credibility Weights Using Fuzzy Cognitive Maps”, en: *Proceedings, IEEE First International Conference on Neural Nets*, San Diego, 1987.
- Taha, Ismail y Joydeep Gosh “Symbolic interpretation of artificial neural networks”, *Reporte de Investigación ARO DAAH04-94-G-0417*, University of Texas, Austin, Septiembre 20 de 1996.
- Taha, Ismail y Joydeep Gosh “Three techniques for extracting rules from feedforward networks”, en: *Intelligent Engineering Systems Through Artificial Neural Networks*, Volúmen 6, ASME Press, 1996, 6 páginas.
- Tiri, Henry “Raplacing the pattern Matcher of an expert system with a neural network” en: *Intelligent Hybrid Systems*, de Suran Goonatilake y Sukhndev Khebbal (Compiladores), Londres, University College, London /John Wiley & Sons Editors, 1995, págs. 47-62.
- Towell, Geoffrey G, y Jude W. Shavlik, “Extracting Refined Rules from Knowledge-Based Neural Networks”, en: *Machine Learning*, 11/92, 1992.
- Tracy, Bent K, Jr. , “Expert Systems in Energy Management” en *ENGM 624 Knowledge Engineering*, USA, 1996.
- Treviño Avendaño, Sandra Patricia y José Luis Aguirre Cervantes, “Aná-

- lisis del conocimiento de varios expertos utilizando un modelo de representación intermedia del conocimiento”, en : Memorias de la XI Reunión Nacional de Inteligencia Artificial, Guadalajara, SMIA/UdeG/MegaByte, Noriega Editores, Septiembre de 1994, págs 217-235.
- Turban, Efraim, *Expert Systems and Applied Artificial Intelligence*, Nueva York, McMillan Publishing Company, 1992.
- Umano, M e Y. Ezawa, “Execution of aproximate reasoning by neural network”, en *Proceedings of FAN Symposium*, 1991, en: Fullér, 1995, *Neural Fuzzy Systems*.
- Von Altrock, Constantin, *Fuzzy Logic & Neurofuzzy Applications Explained*, Nueva Jersey, Prentice Hall, 1995.
- Walker, A, M. McCord, J.F Sowa y W. Wilson, *Knowledge Systems and Prolog. Developing Expert Database, and Natural Lenguaje Systems*, Nueva York, Addison-Wesley, 1990.
- Werbos, P. *Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences*, Cambridge, Ma. 1974, citado por Hecht-Nielsen, 1989.
- Williamson, Mickey, “Neural Networks: Glamour and Glitches”, en: *Computerworld*, USA, Febrero 15 de 1988, pags, 89-92.
- Winston, Henry Patrick, *Inteligencia Artificial*, Willmington, Addison-Wesley, 1994.
- Yu, Jun, *Studies on Symbolic and Connectionist Learning*, ENEE 748D Final Project Report, Department of Electrical Engineering, University of Maryland College Park, diciembre de 1995.
- Zadeh, Lofti A.. “Fuzzy Sets” en: *Information and Control*, No 8, Mimeo, Material de trabajo ITESM/MCC.
- Zadeh, Lofti. A. Y J. Kacprzyk, *Fuzzy Logic for the Management of Uncertainty*, Nueva York, John Wiley & Sons. Inc., 1992.

*Extracción de reglas en redes neuronales difusas
Un modelo basado en la discretización de los conjuntos
difusos de entrada (propuesta del modelo BCD)*
se terminó de imprimir en febrero de 2005
en los talleres de Ediciones de la Noche,
Guadalajara, Jalisco, México.

Formación tipográfica:
Rodolfo Sánchez Gómez.

El tiraje fue de ____ ejemplares.