



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# **Clasificación de patrones usando AMNFIS**

**Deyber Arley Valencia Gómez**

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión

Medellín, Colombia

2018



# **Clasificación de patrones usando AMNFIS**

**Deyber Arley Valencia Gómez**

Trabajo final de maestría presentado como requisito parcial para optar al título de:

**Magister en Ingeniería de Sistemas**

Director:

Ph.D. Juan David Velásquez Henao

Línea de Investigación:

Inteligencia Artificial

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión

Medellín, Colombia

2018



*A mi familia*



## **Agradecimientos**

Es importante hacer mención a las siguientes personas, que de una u otra forma estuvieron presentes durante el desarrollo del presente trabajo y fueron parte fundamental para la culminación con éxito del mismo:

- Al profesor Juan David Velásquez Henao, por su paciencia y orientación durante el desarrollo de este trabajo. Su conocimiento fue muy importante para lograr finalizar con éxito la investigación.
- A mi familia, por su apoyo incondicional en cada una de las situaciones y decisiones que se presentan en mi vida
- A Ana Isabel Fernández Correa, por ser un punto de apoyo importante en cada una de las situaciones que viví durante el desarrollo de la investigación
- Al profesional Andrés Felipe Muñeton, de quien recibí ayuda valiosa para la revisión de la investigación y quien siempre estuvo pendiente de mi avance con la misma



## Resumen

El presente trabajo introduce un enfoque novedoso para resolver problemas de clasificación no lineales usando un sistema de inferencia neuro-difuso adaptativo multidimensional (AMNFIS por sus siglas en inglés) desarrollado inicialmente en el contexto de control de procesos, y posteriormente para el pronóstico de series de tiempo no lineales. En relación a otros sistemas neuro difusos y las versiones previas de AMNFIS, el modelo presentado en este trabajo ha sido modificado para resolver problemas de clasificación binaria. El objetivo de este trabajo es determinar si el modelo AMNFIS puede ser superior a otros tipos de redes neuronales para resolver problemas de clasificación. Para ello, se evaluaron tres problemas no lineales comúnmente usados para comparación de modelos. Los resultados obtenidos para AMNFIS son comparados contra los resultados obtenidos usando diferentes tipos de redes neuronales artificiales. La evidencia empírica indica que AMNFIS es el segundo mejor modelo considerado para el primer problema mientras que para los otros dos problemas restantes, es el modelo más preciso.

**Palabras clave:** Redes neuronales artificiales; clasificación de patrones; clasificación no lineal; AMNFIS.

## Abstract

This work introduces a novel approach to solve nonlinear classification problems using an adaptive multidimensional neuro-fuzzy inference system (AMNFIS), developed originally for processes control, and later for forecasting time series. In relation to other neuro-fuzzy systems and previous versions of AMNFIS, the model presented in this work has been modified for solving binary classification problems. The aim of this work is to determine if AMNFIS is better than other types of neural networks for solving classification problems. To do that, AMNFIS is used to solve three well-known nonlinear classification problems and the results are compared against the results obtained using different types of artificial neural networks. Empirical evidences indicate that AMNFIS is the second-best model considered for first benchmark set, while is the more accurate model in the other two benchmark sets.

**Keywords:** Artificial neural networks; pattern classification; nonlinear classification, AMNFIS.

# Contenido

	Pág.
<b>Resumen</b> .....	<b>IX</b>
<b>Lista de figuras</b> .....	<b>XIII</b>
<b>Lista de tablas</b> .....	<b>XIV</b>
<b>Lista de Símbolos y abreviaturas</b> .....	<b>XV</b>
<b>1. Introducción</b> .....	<b>1</b>
1.1 Revisión de literatura y marco teórico .....	2
1.1.1 Definición matemática del problema de clasificación .....	3
1.1.2 Vecino más próximo .....	3
1.1.3 Modelo logit .....	4
1.1.4 Modelo probit .....	4
1.1.5 Máquinas de vectores de soporte (SVM).....	5
1.1.6 DAN2 .....	5
1.1.7 Árboles de clasificación.....	5
1.1.8 Métodos de ensamble .....	6
1.2 Objetivos.....	7
1.2.1 Objetivo general .....	7
1.2.2 Objetivos específicos .....	8
<b>2. Metodología Propuesta</b> .....	<b>9</b>
2.1 Arquitectura del modelo.....	9
2.2 Estimación de los valores óptimos de los parámetros del modelo .....	12
2.3 Estimación del número de reglas (conjuntos difusos multidimensionales).....	13
2.4 Implementación .....	13
<b>3. Implementación y Validación</b> .....	<b>14</b>
3.1 Instalación de paquetes .....	14
3.2 Selección del conjunto de datos.....	14
3.3 Selección de los datos de entrenamiento y validación.....	15
3.4 Entrenamiento del modelo.....	15
3.5 Validación del código .....	16
<b>4. Resultados Obtenidos y Discusión</b> .....	<b>17</b>
4.1 Conjunto de datos Johns Hopkins Ionosphere .....	17
4.2 Conjunto de datos Pima Indian Diabetes .....	18
4.3 Conjunto de datos BUPA Liver Disorders .....	20
4.4 Otras medidas de desempeño para AMNFIS .....	21

<b>5. Conclusiones y recomendaciones .....</b>	<b>23</b>
5.1 Conclusiones .....	23
5.2 Recomendaciones .....	24
<b>Bibliografía .....</b>	<b>25</b>

## Lista de figuras

	<b>Pág.</b>
Figura 2-1: Arquitectura de AMNFIS para clasificación.....	10

## Lista de tablas

	<b>Pág.</b>
<b>Tabla 3-1:</b> Desempeño de AMNFIS frente a otras técnicas para el conjunto de datos Johns Hopkins Ionosphere .....	18
<b>Tabla 3-2:</b> Desempeño de AMNFIS frente a otras técnicas para el conjunto de datos Pima Indian Diabetes .....	19
<b>Tabla 3-3:</b> Desempeño de AMNFIS frente a otras técnicas para el conjunto de datos Bupa Liver Disorders .....	20
<b>Tabla 3-4:</b> Medidas de desempeño para AMNFIS .....	21

## Lista de Símbolos y abreviaturas

### Símbolos con letras latinas

Símbolo	Término	Definición
$x_i$	Vector de características de un dato o punto	Ecuación (1.1)
$\in$	Pertenece	Ecuación (1.1)
$s_i$	Conjunto difuso multidimensional	Ecuación (1.1)
$p_i$	Valor calculado para $x_i$	Ecuación (1.1)
$f(\theta_i; x_i)$	Función lineal	Ecuación (1.1)
$n$	Atributo n	Ecuación (1.2)
$N$	Número total de atributos	Ecuación (1.2)
$c_i$	Centro de cluster	Ecuación (1.3)
$D_i^2$	Distancia	Ecuación (1.3)
$\exp$	Exponencial	Ecuación (1.4)
$L$	Cantidad de reglas o conjuntos difusos	Ecuación (1.4)
$f_i$	Función para obtener promedio ponderado	Ecuación (1.6)
$y$	Salida del modelo usando función logística	Ecuación (1.7)
$E$	Función de error	Ecuación (1.8)
$m$	Muestra actual	Ecuación (1.8)
$M$	Total de muestras	Ecuación (1.8)
$d_m$	Clase correcta para la muestra m	Ecuación (1.8)
$\log$	Logaritmo	Ecuación (1.8)
$y_m$	Clase estimada por el modelo para la muestra m	Ecuación (1.8)

### Símbolos con letras griegas

Símbolo	Término	Definición
$\Sigma$	Sumatoria	Ecuación (1.2)
$\theta$	Parámetro de función lineal	Ecuación (1.2)
$\mu_i$	Función de pertenencia	Ecuación (1.4)
$\omega_i$	Contribución de cada regla o conjunto difuso multidimensional	Ecuación (1.5)

## Abreviaturas

<b>Abreviatura</b>	<b>Término</b>
<i>AMNFIS</i>	Adaptive Multidimensional Neuro-Fuzzy Inference System
<i>BPNN</i>	Back Propagation Neural Network
<i>CMTNN</i>	Complementary Neural Network
<i>GRNN</i>	General Regression Neural Network
<i>PNN</i>	Probabilistic Neural Network
<i>RBNN</i>	Radial Basis Function Neural Network

# 1. Introducción

La CLASIFICACIÓN de patrones es una técnica fundamental en la solución de muchos problemas de la vida real, la cual consiste en clasificar un dato asignándolo a una clase o grupo que ha sido previamente definido. Dicha clasificación usualmente puede ser de dos tipos; la clasificación binaria, donde el resultado es una predicción de si el dato pertenece o no a la clase en cuestión, y la clasificación multiclase, donde se tienen más de dos posibles clases a las cuales el dato puede pertenecer. Existen múltiples áreas en las cuales se puede aplicar la clasificación de patrones, entre las cuales se destacan las siguientes: pronóstico del clima, evaluación de riesgo crediticio, procesamiento de imágenes, diagnóstico médico, reconocimiento de lenguaje natural, reconocimiento de texto manuscrito y control de la calidad [1].

Varias técnicas han sido propuestas en la literatura para abordar problemas de clasificación como las redes neuronales artificiales (ANN por su sigla en inglés) [2]–[4], máquinas de soporte vectorial (SVM por su sigla en inglés) [5], árboles de clasificación [6], regresión logística, vecino más cercano (k-NN por su sigla en inglés) [7], métodos de ensamble [8] y algunas técnicas híbridas como los sistemas neuro-difusos [9]. Estos últimos corresponden a sistemas de inferencia borrosa cuyo proceso de razonamiento está organizado en la forma de una red neuronal artificial, donde los parámetros del modelo pueden ser ajustados minimizando una función que mide el error de clasificación. Aunque muchos trabajos reportados en la literatura sobre los sistemas de inferencia neuro-difusa tienen su aplicación en el contexto de control predictivo [10], [11] y la predicción de series de tiempo no lineales [12], [13], también se encuentran algunos aplicados en el contexto de clasificación; véase las refs. [14], [15].

En [16] se detalla la arquitectura de AMNFIS (adaptive multidimensional neuro-fuzzy inference system por su sigla en inglés) aplicado hasta el momento solo en el contexto de series de tiempo, pero que podría brindar un buen desempeño en el contexto de clasificación. AMNFIS es un sistema neuro-difuso y a diferencia de otros sistemas de este tipo, hace uso de conjuntos difusos multidimensionales y presenta varias ventajas como el uso de una arquitectura muy simple en

comparación con sistemas como ANFIS [17], HyFIS [18], MUFIS [19] o DENFIS [20]; hay un número inferior de parámetros que deben ser ajustados por el algoritmo de optimización; y, finalmente, la cantidad y posición de los clusters puede ser fácilmente encontrada por el algoritmo de especificación. En AMNFIS, las funciones de pertenencia son calculadas como una función de la distancia del vector de entrada a los centros de los clusters que representan cada conjunto difuso, mientras que los consecuentes de las reglas son funciones polinomiales de orden uno.

Ya que el modelo propuesto por [16] ha sido usado únicamente en el contexto de series de tiempo, es necesario evaluar su desempeño en problemas de clasificación. Para ello, en este trabajo se presenta una versión modificada de AMNFIS para realizar clasificación binaria de patrones y determinar si este modelo es competente frente a otras técnicas propuestas en la literatura mediante la evaluación de desempeño sobre conjuntos de datos ampliamente usados como puntos de referencia en problemas de clasificación. Durante el desarrollo del mismo, se hará una exhaustiva referencia al trabajo presentado en [16] ya que i) este trabajo es una modificación a la metodología allí presentada y ii) a la fecha de revisión del estado del arte de esta investigación, febrero de 2017, no se encontraron otros trabajos reportados sobre sistemas de inferencia neuro-difusos multidimensionales.

En el resto de esta sección se presenta la revisión de literatura y se plantean los objetivos de investigación

## **1.1 Revisión de literatura y marco teórico**

En esta sección se presentan aspectos relevantes de los modelos más representativos usados desde la estadística y la inteligencia computacional. Desde la estadística se discuten los modelos: vecino más próximo y los basados en regresión. Desde la inteligencia computacional se discuten modelos basados en redes neuronales como el DAN2 y las máquinas de vectores de soporte (SVM por su traducción al inglés support vector machines); también se discuten modelos como los métodos de ensamble, dentro de los cuales se abordan el Bagging, Boosting y Random Forest. Por último, se abordan los árboles de clasificación.

### 1.1.1 Definición matemática del problema de clasificación

El problema de clasificación formalmente se define como la tarea de estimar la etiqueta o clase  $y$  para un vector de entrada  $x$  con  $k$  dimensiones, donde  $x \in X \subseteq \mathbb{R}^k$  y  $y \in Y = \{C_1, C_2, \dots, C_Q\}$ .  $X$  representa el espacio de muestras,  $Y$  representa el espacio de clases definidas para las muestras  $X$  y  $C$  indica la etiqueta o clase para una muestra particular de un conjunto de  $Q$  posibles clases. La tarea de clasificación es realizada usando una función  $g : X \rightarrow Y$  que minimiza el error en la clasificación y puede generalizar la capacidad de predicción sobre nuevas muestras [21].

### 1.1.2 Vecino más próximo

La regla del vecino más próximo le asigna a una muestra no clasificada del conjunto de datos, la misma clase de su vecino o  $k$  vecinos más próximos ya clasificados correctamente [22]. Tienen la ventaja de reconocer patrones sin la necesidad de una coincidencia exacta con algún patrón o datos almacenados. A diferencia de otras técnicas, donde la clasificación de un nuevo caso se lleva a partir de dos tareas, la inducción del modelo clasificatorio y la posterior deducción o aplicación sobre el nuevo caso, en este modelo al no existir un modelo explícito, las dos tareas anteriores se encuentran colapsadas en lo que se acostumbra a denominar transducción. En espacios de representación vectorial con producto escalar las funciones discriminantes asociadas a un clasificador basada en la regla del vecino más próximo son lineales a intervalos. Como consecuencia, se pueden obtener funciones discriminantes arbitrariamente complejas, lo que supone una ventaja clara sobre otros clasificadores (paramétricos o no) que solo producen ciertos tipos de fronteras (lineales, cuadráticas, ...). La probabilidad de error de este clasificador está acotada por el doble de la probabilidad de mínimo error de Bayes [22]–[24]:

$$p_{1-NN} \leq p_{1-NN} < p_B \left( 2 - \frac{C_{p_B}}{C-1} \right) < 2p_B$$

donde  $p_B$  es la probabilidad de mínimo error de Bayes y  $p_{1-NN}$  es la probabilidad de error del clasificador basado en la regla del vecino más próximo. A pesar de esto, se depende de la existencia de una métrica de distancia y el costo computacional puede ser alto [25].

### 1.1.3 Modelo logit

Una de las ventajas de este modelo basado en regresión es que se puede considerar como una fórmula para calcular la probabilidad de pertenencia a los grupos [26], [27], lo que conlleva a que la interpretación sea sencilla, por tratarse en términos de probabilidad. Bajo este esquema la probabilidad condicional de que una muestra pertenezca a uno de los grupos dadas  $n$  observaciones  $(y_i, x_i)$ , donde  $x_i$  es un vector con  $p$  variables y  $y_i$  es una variable aleatoria independiente multinomial, la cual indica el número de grupos, está dada por:

$$p(y = j | x_i) = p(E|x) = \frac{\exp(\alpha_{1j} + \beta'_{1j}x_i)}{1 + \sum_{k=2}^G \exp(\alpha_{1k} + \beta'_{1k}x_i)}$$

donde  $(\alpha$  y  $\beta)$  son parámetros desconocidos que se estiman de los datos y la cual asegura que la variable de respuesta es una probabilidad entre 0 y 1 [28]. Otra de las ventajas que proporciona este modelo, es que es claro y pueden usarse todos los tipos de variables [29]. Una particularidad del modelo es que funciona bien cuando la separación entre los grupos es grande. Pero, este modelo presenta la limitante de que cuando los datos no están bien estructurados o hay muy pocos, se puede presentar problemas de separación, donde el proceso de estimación de los estimadores por máxima verosimilitud de los parámetros no converge, y por lo tanto no se pueden realizar inferencias [30].

### 1.1.4 Modelo probit

Otro modelo basado en regresión es el probit desarrollado en [31], el cual está dado por la función de distribución acumulativa de la distribución normal estándar. A diferencia del modelo logit que tiene una distribución logística, la ecuación del modelo probit está dada por:

$$P(Y = 1) = L\left(\sum_{k=1}^K \beta_k x_k\right) = \frac{e^{\sum_{k=1}^K \beta_k x_k}}{1 + e^{\sum_{k=1}^K \beta_k x_k}}$$

donde  $L$  representa la distribución logística,  $x$  es la variable explicativa,  $\beta$  su coeficiente de variación y  $y$  la variable que se predice. A diferencia del modelo logit, este modelo tiene un mejor comportamiento con conjuntos de datos pequeños, pero con un conjunto de datos grande el modelo logit tiene un mejor comportamiento. Esto se debe a una mayor propagación del modelo probit sobre la curva de distribución. Véase refs [32]–[34].

### 1.1.5 Máquinas de vectores de soporte (SVM)

A diferencia de las redes neuronales artificiales que utilizan el principio de minimización de riesgo empírico (ERM) durante la fase de entrenamiento, las máquinas de vectores de soporte se basan en el principio de minimización del riesgo estructural (SRM) [3], [4], el cual ha mostrado un mejor desempeño que el ERM, ya que las SVM minimizan un límite superior al riesgo esperado a diferencia del ERM que minimiza el error sobre los datos de entrenamiento [5]. Otra de las ventajas es que se puede aplicar a problemas lineales y no lineales, a problemas con grandes conjuntos de datos complejos y con ruido, lo que para otros modelos como arboles de clasificación aumentaría la posibilidad de error en los resultados, pues resultan difíciles de cuantificar y observar. A diferencia de modelos como las redes neuronales artificiales (ANN), donde el espacio de búsqueda tiene múltiples mínimos locales, en las SVM, el espacio de búsqueda tiene un solo mínimo global, además el entrenamiento es muy eficiente [35]. Otro beneficio importante es que la complejidad del clasificador resultante es caracterizada por el número de vectores de soporte más que por la dimensionalidad del espacio de búsqueda. Como resultado de esto las SVMs tienden a ser menos propensas a problemas de sobreajuste que algunos otros métodos. A pesar de esto, una de las grandes falencias del modelo, es que no está diseñado para identificar los atributos importantes para construir la regla discriminante.

### 1.1.6 DAN2

Entre las ventajas más significativas que presenta este modelo de red neuronal están que no se requiere un modelo predefinido, ya que su construcción es dinámica [2], removiéndole la necesidad de experimentación con el modelo. Este modelo trabaja bien sobre procesos no lineales. Una de las principales diferencias con modelos comunes de redes neuronales es que DAN2 emplea una arquitectura diferente al tradicional Feed Forward Back Propagation (FFBP), dado que ésta presenta el inconveniente de ser dependiente del problema y a menudo requiere experimentación antes de que la arquitectura final sea seleccionada para el entrenamiento y pronóstico [2].

### 1.1.7 Árboles de clasificación

Los árboles de clasificación, presentan la ventaja significativa de la facilidad de interpretación, sobre modelos como las redes neuronales dado el modus operandi basado en una serie de preguntas que se van desglosando a través del árbol. Otra ventaja importante es que la clasificación es rápida [6], y posee una forma natural de incorporar conocimiento. Este modelo funciona bien cuando la

separación entre los grupos es grande. A diferencia de otros modelos, como el vecino más cercano, no necesita de una métrica definida para realizar la clasificación. Son útiles con datos no paramétricos. Una de las desventajas es que el árbol obtenido es generalmente sobreajustado por lo tanto es necesario un mecanismo de poda, la cual computacionalmente puede ser compleja y puede que no sea tan útil para problemas que tienen alto ruido y son de naturaleza estadística [29]. Además, cuando las variables tienen una correlación baja, y los tamaños de muestra son pequeños, el índice de error de la clasificación puede ser alto. También se tienen que tener en cuenta otra serie de aspectos como: la restricción de las propiedades (si deben ser valores binarios o pueden ser múltiples valores), la propiedad a ser testada en cada nodo, cuando un nodo debe ser declarado como hoja, cómo realizar la poda, si un nodo hoja es impuro, como debe ser asignada la categoría, cómo deben ser manejados los datos faltantes [6], [36]. Otro factor en contra de los árboles de clasificación es que son pobres en la inferencia de conceptos simples

### 1.1.8 Métodos de ensamble

Los ensambles son un conjunto de clasificadores que combinan las decisiones individuales de cada uno de ellos para clasificar nuevas instancias [37]. Entre las ventajas que justifican el ensamble de clasificadores, se tienen i) los datos para entrenamiento pueden no proveer suficiente información para elegir un único mejor clasificador debido a que el tamaño es pequeño [38]; ii) la combinación redundante y complementaria de clasificadores mejora la robustez, exactitud y generalidad de toda la clasificación [39]; iii) diferentes clasificadores utilizan diferentes técnicas y métodos de representación de los datos, lo que permite obtener resultados de clasificación con diferentes patrones de generalización; iv) los ensambles son frecuentemente mucho más exactos que los clasificadores individuales [37]. Los métodos de ensamble más comunes son *Bagging* [40], *Boosting* [41], [42] y *Random Forest* [43]. A continuación, se mencionan algunas características de cada uno.

***Bagging:*** Los clasificadores son del mismo tipo y cada uno se obtiene con un conjunto de muestras diferentes que se obtienen seleccionando muestras con reemplazamiento en el conjunto de muestras de entrenamiento original. Una de las ventajas es que es siempre más preciso que un clasificador simple. Además muestra mucha menos correlación, comparada con el Boosting [44].

**Boosting:** Este método, a diferencia del Bagging, produce una serie de clasificadores. El conjunto de entrenamiento usado para cada miembro de la serie es seleccionado basado en el rendimiento del clasificador anterior en la serie. La ventaja más significativa, es que podría superar en gran medida el Bagging y el clasificador simple. Sin embargo, con algunos conjuntos de datos, puede mostrar cero ganancia o incluso una disminución en el rendimiento comparado con un solo clasificador, lo cual se explica ya que puede sufrir de sobreajuste con la presencia de ruido [44]. En general el Bagging es probablemente una técnica más apropiada que el Boosting para la mayoría de problemas, pero cuando es apropiada, Boosting puede producir mayores ganancias y precisión.

**Random Forest:** Esta técnica utiliza Bagging, junto con la selección aleatoria de atributos presentada en [45] y [46]. La diferencia con el Bagging es que en cada nodo se selecciona aleatoriamente un subconjunto de los atributos disponibles en el nodo y se elige el mejor de ellos de acuerdo al criterio de división. Random Forest es un algoritmo de aprendizaje muy certero, especialmente para conjuntos de datos grandes y su tiempo de ejecución es rápido [47]. Una de las ventajas es que proporciona estimados de qué variables son importantes en la clasificación. Además, tiene un método eficaz para estimar datos perdidos y mantener la exactitud cuando una gran proporción de los datos está perdida. Puede manejar gran cantidad de variables entrantes sin excluir ninguna [47]. A pesar de esto, se ha observado que sobreajusta en ciertos grupos de datos [48]. Además, la clasificación es difícil de interpretar por el hombre y para los datos que incluyen variables categóricas con diferente número de niveles, se parcializa a favor de esos atributos con más niveles. Por consiguiente, la posición que marca la variable no es fiable para este tipo de datos [49].

## 1.2 Objetivos

### 1.2.1 Objetivo general

Determinar si el modelo AMNFIS puede ser superior a otras técnicas clásicas de clasificación, usando bases de datos propuestas en la literatura. Específicamente, se usan los conjuntos de datos binarios Pima Indians Diabetes, BUPA Liver Disorders y Johns Hopkins Ionosphere obtenidos a partir de la base de datos de Machine Learning UCI [50]. Estos conjuntos de datos fueron seleccionados ya que han sido puntos de referencia ampliamente usados en la literatura científica especializada. Ningún tipo de pre-procesamiento como selección de características relevantes,

extracción de características o aplicación métodos de imputación sobre datos faltantes es realizado sobre los datos, lo que garantiza que la comparación sea lo más transparente posible y solo difiera en el algoritmo usado para clasificar los datos.

### **1.2.2 Objetivos específicos**

1. Implementar y validar el código de un prototipo que implemente la metodología AMNFIS en problemas de clasificación.
2. Comparar contra resultados publicados en la literatura, teniendo en cuenta medidas de precisión como sensibilidad, especificidad, falsos positivos, falsos negativos.

## 2. Metodología Propuesta

La metodología usada en el presente trabajo es definida inicialmente en [16] para el contexto de series de tiempo no lineales. En esta sección se hace una descripción de la misma, aplicándola en el contexto de clasificación de patrones.

### 2.1 Arquitectura del modelo

Sea  $X$ , el conjunto de datos con 2 clases a clasificar. El modelo propuesto consiste en un conjunto de  $L$  reglas difusas, donde la  $i$ -ésima regla es expresada como:

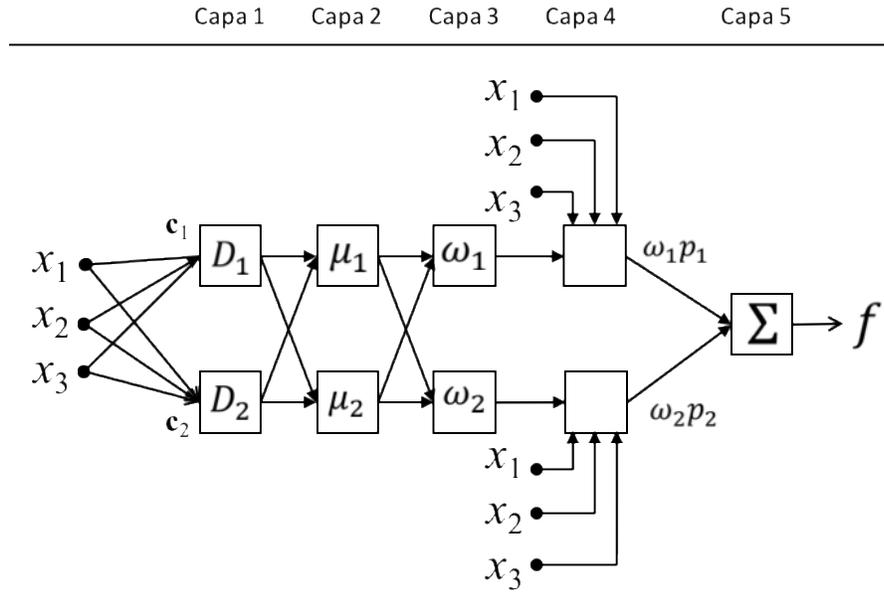
$$\text{if } \mathbf{x} \in S_i, \text{ then } p_i = f(\boldsymbol{\theta}_i; \mathbf{x}) \quad (2.1)$$

donde  $\mathbf{x} = [x_1, x_2, \dots, x_N]'$  es el vector de características de la muestra  $\mathbf{x}$ ;  $S_i$  es el  $i$ -ésimo conjunto difuso multidimensional;  $p_i$  es el valor calculado para  $\mathbf{x}$  usando la regla  $i$ ;  $f(\cdot, \cdot)$  es una función lineal que toma la forma:

$$p_i = f(\boldsymbol{\theta}_i; \mathbf{x}) = \theta_0 + \sum_{n=1}^N \theta_{i,n} x_n \quad (2.2)$$

donde  $\boldsymbol{\theta}_i$  representa los parámetros del modelo que son optimizados por el algoritmo. Nótese que la ecuación (1.2) representa un modelo de regresión lineal de  $N$  variables escalares con intercepto diferente de cero. Tal como en otros sistemas difusos, es necesario definir la función de pertenencia de  $\mathbf{x}$  al conjunto  $S_i$ . En AMNFIS, cada conjunto  $S_i$  es representado por su centro  $\mathbf{c}_i$ , y el valor de la función de pertenencia,  $\mu_i$ , es definido como una función de las distancias del punto actual  $\mathbf{x}$  al centro de los clusters o conjuntos difusos, la cual será definida más adelante en este mismo capítulo.

A modo de ejemplo, la arquitectura del modelo para un sistema con tres entradas y dos reglas difusas es presentada en la **Figura 2-1**.

**Figura 2-1:** Arquitectura de AMNFIS para clasificación

A continuación, se describe el proceso de cálculo de la salida del sistema presentado en la **Figura 2-1**:

*Capa 1.* En esta capa se realiza el cálculo del cuadrado de la distancia del punto actual  $\mathbf{x}$  a cada uno de los centros  $\mathbf{c}_i$ , como:

$$D_i^2 = \|\mathbf{x} - \mathbf{c}_i\|^2 \quad (2.3)$$

Tal como en las redes de funciones de base radial, los pesos entre las variables de entrada y los nodos de esta capa, pueden ser usados para representar las coordenadas de los centros  $\mathbf{c}_i$ ; es así como los pesos de las conexiones que entran al  $i$ -ésimo nodo corresponden a las componentes del vector  $\mathbf{c}_i$  que representa el conjunto borroso  $S_i$ .

*Capa 2.* Se calcula la función de pertenencia  $\mu_i$  del punto actual  $\mathbf{x}$  para cada uno de los  $L$  clusters con centro  $\mathbf{c}_i$ :

$$\mu_i = \exp \left[ -\frac{D_i^2}{\sum_{j=1}^L D_j^2} \right] = \exp \left[ -\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\sum_{j=1}^L \|\mathbf{x} - \mathbf{c}_j\|^2} \right] \quad (2.4)$$

Así, la función de pertenencia  $\mu_i$  es igual a la unidad cuando  $\mathbf{x} = \mathbf{c}_i$ , y decrece cuando la lejanía del punto actual  $\mathbf{x}$  al centro  $\mathbf{c}_i$  se incrementa.

*Capa 3.* Se calcula la contribución neta de cada regla,  $\omega_i$ , a partir de la normalización de los valores de pertenencia del punto actual  $\mathbf{x}$  a cada cluster:

$$\omega_i = \frac{\mu_i}{\sum_{j=1}^L \mu_j} \quad (2.5)$$

*Capa 4.* Se calcula salida  $p_i$  para  $i = 1, \dots, L$ , de todas las reglas o clusters usando la ecuación (1.2)

*Capa 5.* Se obtiene el promedio ponderado de la salida de las reglas difusas:

$$f_i = \sum_{j=1}^L \omega_j p_j \quad (2.6)$$

Hasta este punto la metodología propuesta en [16] es similar para resolver problemas de clasificación. Es bien sabido que las metodologías usadas para resolver problemas de regresión, no pueden ser aplicadas a los problemas de clasificación binaria ya que se basan en supuestos matemáticos diferentes [51].

Dado que dicha metodología fue enfocada para problemas de regresión donde la salida del sistema es un valor numérico  $\in \mathbb{R}$ , es necesario un paso adicional ya que la salida para un problema de clasificación es un valor de probabilidad en el rango  $[0, 1]$  dónde 1 significa una alta probabilidad de que la muestra  $i$  pertenezca a la clase en cuestión. En este trabajo se abordan problemas de clasificación binaria, y por lo tanto, se usa la función logística para obtener el valor de probabilidad ya que esta función es ampliamente usada en el mundo de la inteligencia computacional y la estadística [52]. La función es definida como:

$$y_i = \frac{1}{1 + \exp(-f_i)} \quad (2.7)$$

Si el problema que se aborda es de clasificación multiclase, es necesaria otra estrategia donde el sistema tiene tantas salidas como clases y por cada de una de las salidas se da un valor de probabilidad, con la restricción de que la suma de probabilidades tiene que ser igual a 1. En ese caso, la función más ampliamente usada en la literatura es softmax; véase las refs [52], [53].

## 2.2 Estimación de los valores óptimos de los parámetros del modelo

Para realizar la estimación de los valores óptimos de los parámetros del modelo se supone que las componentes del vector  $x$  (esto es, los atributos que serán considerados), y la cantidad de reglas (o clusters)  $L$  ya han sido definidas.

Los valores óptimos de los parámetros se definen como aquellos valores que permiten minimizar una función de error de la clasificación. La estrategia planteada en [16] para la optimización de los parámetros no es posible desarrollarla en esta investigación dado que el tipo de problemas es diferente. En este trabajo se abordan problemas de clasificación y para este tipo de casos la función de entropía cruzada es ampliamente usada en la literatura; dicha función se define como:

$$E = - \sum_{m=1}^M d_m * \log(y_m) + (1 - d_m) \log(1 - y_m) \quad (2.8)$$

donde  $d_m$  corresponde a la clase correcta para la muestra  $m$  y  $y_m$  corresponde a la clase estimada por el modelo.

En [16], se discute una estrategia general de optimización para la minimización del error cuadrático medio, que aprovecha la estructura línea de las reglas de los consecuentes, y se plantean varias alternativas para realizar la optimización global de los parámetros. Sin embargo, la estrategia planteada en dicho trabajo no aplica para el problema abordado en esta investigación; esto se debe a que aquí, se debe minimizar la función de entropía cruzada (definida en la Ecuación (1.8)), y este problema no puede ser resuelto usando mínimos cuadrados ordinarios. De ahí, que la única alternativa viable sea el uso de técnicas basadas en gradiente.

En este trabajo, se usa un proceso interactivo en donde los centros  $\mathbf{c}_i$  se seleccionan de acuerdo a los datos de entrenamiento que mejor contribuyan a una reducción del error. Para la elección del centro  $\mathbf{c}_i$  se evalúa el desempeño del modelo tomando cada punto del conjunto de entrenamiento como centro y se selecciona, de entre todos los puntos, aquel que proporcione el menor error en la clasificación; esto es, que minimice la función de entropía definida en Ecuación (1.8). Este proceso

continúa para la determinación de cada uno de los centros teniendo en cuenta que para hallar el centro  $\mathbf{c}_i$  se debió previamente haber hallado el centro  $\mathbf{c}_{i-1}$ .

### **2.3 Estimación del número de reglas (conjuntos difusos multidimensionales)**

Una discusión general sobre la estimación del número de reglas para AMNFIS es presentada en [16]. En esta investigación, se aplica un método constructivo similar al desarrollado allí, de la siguiente forma: se inicia con un modelo con dos reglas ( $L = 2$ ) y se adiciona una regla en cada iteración. Cada vez que se adiciona una nueva regla, los parámetros óptimos de AMNFIS son estimados usando la estrategia ya planteada y se detiene cuando no se sea posible seguir reduciendo la función de error, o cuando esta empiece a crecer.

### **2.4 Implementación**

La metodología propuesta es implementada en un paquete escrito en el lenguaje de programación R, la cual puede ser directamente instalada por el lector desde un repositorio de GitHub [54]. Se invita al lector a consultar los manuales del paquete devtools, donde se describe detalladamente como instalar aplicaciones que se encuentran en dicho depósito de software. Además, en el capítulo 3, se muestra un ejemplo de ejecución del algoritmo usando uno de los conjuntos de datos referenciados en este trabajo.

## 3. Implementación y Validación

Para la ejecución del código fuente, es necesario instalar el paquete de R *amnfisClassifier*, que se encuentra en el repositorio de GitHub [54]. Para instalar paquetes de R desde dicho repositorio, es necesario tener instalado el paquete *devtools*. Además de este paquete, es necesaria la instalación de otros paquetes, como los son el *mlbench*, que permite tener acceso al conjunto de datos, *dplyr*, con el que se puede hacer transformaciones sobre los datos con el fin de obtener el dataset en el formato apropiado para que el algoritmo de AMNFIS lo pueda procesar; y, por último, el paquete *caret*, para particionar los datos de forma aleatoria en un conjunto de entrenamiento y otro de validación

### 3.1 Instalación de paquetes

A diferencia del paquete *amnfisClassifier*, los demás paquetes se instalan desde el repositorio del CRAN. A continuación, se muestra la forma de instalar cada uno de los paquetes

```
install.packages('devtools')
install.packages('mlbench')
install.packages('dplyr')
install.packages('caret')

devtools::install_github('deybvagm/amnfisClassifier')
```

### 3.2 Selección del conjunto de datos

Para este caso, se usa el conjunto de datos *PimaIndiansDiabetes*, aunque es a gusto del lector usar otro conjunto de datos de clasificación binaria. Es importante notar que el algoritmo usado por AMNFIS solo recibe datos numéricos, por lo que se hace necesario hacer una manipulación de los datos, para convertir datos categóricos como lo es la clase a un valor numérico. En el siguiente fragmento de código, se convierte el valor del atributo diabetes, a un valor numérico de 1 si su valor es pos, ó 0 en caso contrario.

```
data("PimaIndiansDiabetes")
pima_all <- PimaIndiansDiabetes %>% mutate(diabetes = ifelse(diabetes == "pos",
1, 0))
```

### 3.3 Selección de los datos de entrenamiento y validación

Con el fin de replicar el estudio experimental realizado en esta investigación, se debe seleccionar un conjunto de entrenamiento de 614 datos y otro de validación de 154. Para esto se debe ejecutar el siguiente bloque de código:

```
pima_train_index <- createDataPartition(y = pima_all$diabetes, p = 0.7989, list
= FALSE, times = 1)
df.pima_train <- pima_all[pima_train_index, ]
mat.pima_train <- as.matrix(df.pima_train[,1:8])
vec.pima_out_train <- df.pima_train$diabetes
```

### 3.4 Entrenamiento del modelo

En este punto ya es posible realizar el entrenamiento del modelo llamando la función *fn.train\_ammfis*, usando el conjunto de datos de entrenamiento y el número de clusters que se desee usar. En este caso la investigación arrojó que el número óptimo de clusters es 8. El resultado de esta función es un objeto con los parámetros del modelo.

```
obj.res_pima <- fn.train_ammfis(df=df.pima_train, X=mat.pima_train,
d=vec.pima_out_train, formula=diabetes~., n_clusters=8)
```

Si se quiere obtener la predicción del modelo para estos parámetros, se debe ejecutar la función *fn.ammfis\_simulate* pasando el conjunto de datos de validación como parámetro.

### 3.5 Validación del código

Para validar que las funciones que definen la metodología de AMNFIS funcionen de forma correcta y así dar cumplimiento a uno de los objetivos del presente trabajo, se realizan pruebas unitarias sobre los principales componentes mencionados en la sección 2.1. De forma más específica, se realizan pruebas unitarias a la función de cálculo de distancias (`fn.get_Xi_Ci_Distances`), función de cálculo de pertenencia de los puntos a los clusters (`fn.get_membership`), función de cálculo de la contribución de los puntos a cada cluster (`fn.contrib`) y la función que agrupa las anteriores funciones para obtener una predicción de la clase (`fn.amnfis_simulate`). A continuación, se presenta la prueba unitaria realizada para la función `fn.amnfis_simulate`, ya que las pruebas para las demás funciones siguen una estructura similar y pueden ser consultadas en el repositorio de GitHub [54]:

```
test_that("forward propagation (fn.amnfis_simulate) is working properly with 2
fixed clusters, fixed phi params and fixed data points(x,y coordinates)", {
  X <- matrix(c(1,2,1,2,1,1,2,2,3,3,3,3,2,2,1,0,0,1,0,1), ncol = 2)
  C <- matrix(c(-1.6260988,0.302947,-0.5021358,0.7021099), ncol = 2)
  PHI <- matrix(c(-0.004729057,0.285950071,1.172539,0.1799518), ncol = 2)
  phi_0 <- c(0.3773669,1.8119634)
  obj <- NULL
  obj$PHI <- PHI
  obj$phi_0 <- phi_0
  expected_prediction <-
c(0.9568798251,0.9638655803,0.9271078922,0.9389457679,0.8872885183,0.8246005294
,0.8413785165,0.901940737,0.8561568117,0.9138394971)
  pred <- fn.amnfis_simulate(obj = obj, X = X, C = C)
  expect_equal(pred, expected_prediction)
  expect_that(length(pred), equals(10))
  expect_that(pred, is_a("numeric"))
})
```

Básicamente, lo que hace la prueba es definir de forma sintética 10 muestras, 2 clusters y los parámetros phi para ser pasados por la red y verificar que el resultado sea igual a uno computado manualmente. Se verifica que el resultado sea el esperado, que las dimensiones del vector resultante sean las esperadas y por último se verifica que el tipo de datos sea numérico.

## 4. Resultados Obtenidos y Discusión

En este capítulo se aplica la metodología propuesta a tres problemas benchmark de clasificación y se compara la precisión obtenida usando AMNFIS con otros clasificadores. Específicamente los clasificadores usados en esta investigación para la comparación son Back Propagation Neural Network (BPNN), Radial Basis Function Neural Network (RBFNN), General Regression Neural Network (GRNN), Probabilistic Neural Network (PNN), y Complementary Neural Network (CMTNN). Dichos clasificadores fueron usados inicialmente en [55] para realizar un análisis comparativo de la siguiente forma: cada conjunto de datos es dividido aleatoriamente 10 veces para formar diferentes datos de entrenamiento y validación con el fin de obtener resultados más confiables para realizar la comparación, lo cual implica que se tengan que realizar 10 ejecuciones del algoritmo. Para que los resultados sean comparables, se realiza exactamente el mismo experimento, primero determinando el número ideal de reglas siguiendo el enfoque presentado en la sección 2.3 y luego realizando las 10 ejecuciones con un número fijo de reglas.

### 4.1 Conjunto de datos Johns Hopkins Ionosphere

Este conjunto de datos corresponde a datos obtenidos por un sistema radar de 16 antenas de alta frecuencia en Labrador (Canadá) y cuyo objetivo es la detección de electrones libres en la ionosfera. Si se encuentra una determinada estructura en la señal se etiqueta como “g” (good, bueno) y en caso contrario se etiqueta como “b” (bad, malo). El conjunto de datos contiene 351 ejemplos con 34 atributos cada uno (no existen datos faltantes); donde el 80% (281) de los datos son usados para el entrenamiento del modelo y el 20% restante (70) para validación. La **Tabla 4-1** muestra los resultados obtenidos luego de 10 ejecuciones para un total de 3 reglas o conjuntos difusos; la última columna representa los resultados obtenidos por AMNFIS y las demás representan los resultados obtenidos por las técnicas de clasificación mencionadas al principio de este capítulo. La **Tabla 4-1** presenta el porcentaje de ejemplos bien clasificados para la muestra de validación.

Para este caso se consideró un modelo AMNFIS que usa como entradas los 34 atributos del conjunto de datos, y se evaluaron modelos que van desde dos hasta diez reglas en total. Se encontró que un modelo con tres conjuntos difusos multidimensionales presenta el mejor balance entre la precisión del ajuste para las muestras de entrenamiento y validación. Los resultados del desempeño de AMNFIS son presentados en la **Tabla 4-1**. En dicha tabla se observa que AMNFIS, con un 90.99% de precisión sobre el conjunto de validación, tiene un mejor desempeño que las técnicas BPNN, RBFNN y PNN; siendo solo superada por GRNN y CMTNN, las cuales tienen una precisión de 93.15% y 93.43% respectivamente. Así, en comparación con los demás modelos alternativos considerados, AMNFIS captura de buena forma la dinámica de los datos utilizados.

**Tabla 4-1:** Desempeño de AMNFIS frente a otras técnicas para el conjunto de datos Johns Hopkins Ionosphere

Ejecución	BPNN	GRNN	RBFNN	PNN	CMTNN	AMNFIS
1	87.14	90.00	95.71	82.86	90.00	91.42
2	94.29	94.29	95.71	85.71	95.71	92.85
3	87.14	91.43	90.00	85.71	92.86	92.85
4	94.29	95.71	82.86	90.00	95.71	91.42
5	94.29	94.29	91.43	88.57	95.71	91.42
6	82.86	94.29	91.43	85.71	88.57	88.57
7	87.14	91.43	88.57	77.14	90.00	88.57
8	88.57	94.29	85.71	91.43	94.29	91.42
9	94.29	92.86	88.57	84.29	95.71	90.00
10	92.86	92.86	91.43	84.29	95.71	91.42
Promedio	90.29	93.15	90.14	85.57	<b>93.43</b>	90.99

## 4.2 Conjunto de datos Pima Indian Diabetes

Este conjunto de datos cuenta con información de mujeres indígenas provenientes de la tribu Pima de Arizona, igual o mayores a 21 años de edad. A los indios Pima se les conoce por ser genéticamente predispuestos a padecer diabetes y entre los 8 atributos considerados en este conjunto de datos se incluye información médica como niveles de glucosa e insulina y factores de estilo de vida. El

objetivo consiste en determinar si un paciente es diabético de acuerdo a estos atributos y para esto se ha dividido el conjunto de datos de 768 registros en un conjunto de entrenamiento de 614 y uno de validación de 154. La **Tabla 4-2** muestra los resultados obtenidos luego de 10 ejecuciones para un total de 8 reglas o conjuntos difusos; la última columna representa los resultados obtenidos por AMNFIS y las demás representan los resultados obtenidos por las técnicas de clasificación mencionadas al principio de este capítulo. La tabla presenta el porcentaje de ejemplos bien clasificados para la muestra de validación.

**Tabla 4-2:** Desempeño de AMNFIS frente a otras técnicas para el conjunto de datos Pima Indian Diabetes

Ejecución	BPNN	GRNN	RBFNN	PNN	CMTNN	AMNFIS
1	77.27	74.68	79.22	74.68	77.92	78.57
2	76.62	79.87	79.22	79.87	76.62	79.22
3	70.13	70.13	74.03	70.13	72.08	77.92
4	85.71	81.82	79.22	81.82	83.77	74.67
5	75.97	75.97	77.27	75.97	75.32	77.27
6	70.78	70.13	72.08	70.13	72.08	74.67
7	75.32	72.73	76.62	72.73	75.97	76.62
8	79.22	78.57	77.27	78.57	79.22	74.67
9	74.68	74.68	76.62	74.68	75.32	78.57
10	75.97	74.03	74.03	74.03	79.62	77.92
Promedio	76.17	75.26	76.56	75.26	76.49	<b>77.01</b>

Para este caso se consideraron configuraciones con diferentes cantidades de reglas y que usan como entradas los ocho atributos del conjunto de datos. Los resultados experimentales obtenidos permitieron concluir que la mejor configuración de AMNFIS para el número de reglas es  $L = 8$ . Los resultados para el desempeño del modelo final son presentados en la **Tabla 4-2**. A diferencia del caso anterior, AMNFIS presenta un mejor desempeño en predicción respecto a todos los modelos alternativos considerados; específicamente, la precisión obtenida para AMNFIS es de 77.01% mientras que la del modelo RBFNN es de 76.56%, siendo esta técnica la que representa el segundo mejor desempeño entre todos los modelos considerados.

### 4.3 Conjunto de datos BUPA Liver Disorders

Este conjunto de datos que cuenta con 345 registros y 7 atributos; este problema consiste en predecir si un paciente masculino muestra signos de trastornos en el hígado. Para el entrenamiento del modelo se han seleccionado 276 registros (80%), mientras que para la validación se han dejado los 69 restantes (20%). La **Tabla 4-3** muestra los resultados obtenidos luego de 10 ejecuciones para un total de 4 reglas o conjuntos difusos; la última columna representa los resultados obtenidos por AMNFIS y las demás representan los resultados obtenidos por las técnicas de clasificación mencionadas al inicio de este capítulo. La **Tabla 4-3** presenta el porcentaje de ejemplos bien clasificados para la muestra de validación.

**Tabla 4-3:** Desempeño de AMNFIS frente a otras técnicas para el conjunto de datos Bupa Liver Disorders

Ejecución	BPNN	GRNN	RBFNN	PNN	CMTNN	AMNFIS
1	75.36	71.01	69.57	71.01	75.36	71.01
2	62.31	69.57	65.22	69.57	66.67	78.26
3	71.01	65.23	66.67	65.23	69.57	68.11
4	75.36	63.77	66.67	63.77	72.46	75.36
5	79.71	68.12	71.01	68.12	78.26	69.56
6	72.46	68.12	71.01	68.12	75.36	78.26
7	63.77	57.97	66.67	57.97	69.57	63.76
8	66.67	53.62	69.57	53.62	69.57	73.91
9	72.46	69.57	72.46	69.57	73.91	75.36
10	60.86	53.62	56.52	53.62	52.17	68.11
Promedio	70.00	64.06	67.54	64.06	70.29	<b>72.17</b>

Para este caso se consideraron configuraciones de reglas que van desde las dos hasta las diez, obteniendo un buen desempeño a partir de cuatro reglas. Los resultados observados en la **Tabla 4-3**, al igual que en el caso anterior, muestran que el desempeño de AMNFIS es consistentemente mejor en comparaciones con los demás modelos alternativos. Concretamente, la precisión obtenida en AMNFIS fue de 72.17%, mientras que el segundo mejor desempeño lo obtuvo CMTNN con un 70.29%.

## 4.4 Otras medidas de desempeño para AMNFIS

En los objetivos de investigación de este trabajo, se plantea hacer una comparación de AMNFIS contra otras técnicas usando medidas de desempeño como la sensibilidad, especificidad, falsos positivos y falsos negativos. La comparación de desempeño se hace tomando como referencia el trabajo de [55], el cual usa las técnicas mencionadas al principio de este capítulo, pero no se reportan otras medidas a parte del porcentaje de datos bien clasificados (*accuracy*) y por lo tanto no es posible hacer una comparación para las medidas de sensibilidad, especificidad, falsos positivos y falsos negativos. Teniendo esto en cuenta, la **Tabla 4-4** muestra estas métricas adicionales de desempeño solo para el modelo AMNFIS

**Tabla 4-4:** Medidas de desempeño para AMNFIS

Datos	% Clasific.	Falsos positivos	Falsos negativos	Sensibilidad	Especificidad
Ionosphere	87	7	2	95	75
Pima	81	13	16	61	88
BUPA	71	10	10	75	69

En la tabla anterior se reportan diferentes medidas de desempeño de AMNFIS para los tres conjuntos de datos usados. De forma más específica, se reporta el porcentaje de datos bien clasificados sobre el conjunto de datos de validación, la cantidad de falsos positivos, la cantidad de falsos negativos, el valor de sensibilidad que hace referencia a la capacidad del modelo para detectar la clase en cuestión o clase positiva, y el valor especificidad que hace referencia a la capacidad del modelo para clasificar datos de la clase negativa. Los resultados muestran que hay un buen balance en la proporción de falsos positivos y falsos negativos, aunque para el conjunto de datos Johns Hopkins Ionosphere hay una proporción de 2 a 7; también se observa que para este conjunto de datos el modelo AMNFIS tiene muy buen desempeño clasificando datos que pertenecen a la clase positiva lo cual es representado por el valor de sensibilidad.

A diferencia de los resultados reportados anteriormente, donde se realizan 10 ejecuciones con el fin de comparar contra los resultados obtenidos en [55], en este caso solo se realiza una ejecución del

modelo AMNFIS, ya que no es posible hacer comparaciones de estas métricas adicionales con la información reportada en [55].

## 5. Conclusiones y recomendaciones

### 5.1 Conclusiones

En este trabajo se presenta la arquitectura de AMNFIS, la cual fue desarrollada inicialmente en el contexto de series de tiempo no lineales, y ahora es aplicada en el contexto de clasificación de patrones. Además, se realiza la comparación de los resultados de la ejecución de AMNFIS frente a otros métodos propuestos en la literatura haciendo uso de tres conjuntos de datos benchmark bien conocidos y utilizados.

A continuación, se presentan los logros alcanzados en relación a los objetivos de investigación planteados:

1. Objetivo 1: Implementar y validar el código de un prototipo que implemente la metodología AMNFIS en problemas de clasificación.

Para el cumplimiento de este objetivo se desarrolló un paquete de R que puede ser descargado desde el repositorio de GitHub [54] como lo explica el capítulo 3; en dicho paquete se implementa la metodología AMNFIS para resolver problemas de clasificación binarios y en este trabajo se verifica su funcionamiento usando tres conjuntos de datos referentes en la literatura como lo son Pima Indian Diabetes, BUPA Liver Disorders y Johns Hopkins Ionosphere. La validación del código es presentada en la sección 3.5 y su implementación se encuentra en el repositorio de GitHub; dicha validación se hace por medio de pruebas unitarias a las principales funciones que definen la metodología. Específicamente se realizan pruebas a la función de cálculo de distancias (`fn.get_Xi_Ci_Distances`), función de cálculo de pertenencia de los puntos a los clusters (`fn.get_membership`), función de cálculo de la contribución de los puntos a cada cluster (`fn.contrib`) y la función que agrupa las anteriores funciones para obtener una predicción de la clase (`fn.amnfis_simulate`).

- Objetivo 2: Comparar contra resultados publicados en la literatura, teniendo en cuenta medidas de precisión como sensibilidad, especificidad, falsos positivos, falsos negativos.

En el capítulo 4 se presentan los resultados del estudio experimental y se compara contra los resultados obtenidos en otros trabajos. Los resultados obtenidos muestran que AMNFIS es competitivo y puede tener un desempeño superior en comparación con otras de técnicas de clasificación que basan su arquitectura en la forma de una red neuronal artificial. De forma más detallada, se podría decir que, tomando como medida de desempeño el porcentaje de datos bien clasificados, AMNFIS tiene un mejor comportamiento frente a la mayoría de las técnicas usadas para la comparación, siendo solo superada por dos técnicas para el conjunto de datos Johns Hopkins Ionosphere. En este caso se usa el porcentaje de datos bien clasificados como factor de comparación ya que es la única métrica reportada en [55] como medida de desempeño y por lo tanto, no es posible comparar contra otras medidas como la sensibilidad, especificidad, falsos positivos y falsos negativos. A pesar de esto, en el capítulo 4, sí se presentan estas medidas de desempeño para AMNFIS teniendo en cuenta los conjuntos de datos usados. De dicha información se puede concluir que AMNFIS maneja de buena forma la proporción entre falsos positivos y falsos negativos. Además, se observa que el modelo tiene buen desempeño al predecir la clase positiva para el conjunto de datos Johns Hopkins Ionosphere, reportado a través de la métrica de sensibilidad.

## 5.2 Recomendaciones

Existen dos direcciones de trabajo futuro: primero, es necesario desarrollar heurísticas para la selección inicial de los centros de los clusters basadas en el desempeño de AMNFIS que aprovechen la estructura particular que tienen los problemas de clasificación, las cuales permitan encontrar mejores puntos de mínima y mejorar la carga computacional del proceso de optimización. Y segundo, es necesario desarrollar metodologías estadísticas para la especificación del modelo (selección de entradas y cantidad de reglas).

## Bibliografía

- [1] G. P. Zhang, “Neural networks for classification: a survey,” *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.)*, vol. 30, no. 4, pp. 451–462, 2000.
- [2] M. Ghiassi and H. Saidane, “A dynamic architecture for artificial neural networks,” *Neurocomputing*, vol. 63, no. SPEC. ISS., pp. 397–413, 2005.
- [3] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” *Proc. 5th Annu. ACM Work. Comput. Learn. Theory*, pp. 144–152, 1992.
- [4] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [5] V. Vapnik, “The nature of statistical learning theory,” *Springer-Verlg*, 1995.
- [6] L. Breiman, J. H. Friedman, and R. A. Olshen, *Classification and Regression Trees*. New York: Chapman & Hall, 1993.
- [7] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” *Informatica*, vol. 31, pp. 249–268, 2007.
- [8] T. G. Dietterich, “An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees,” *Mach. Learn.*, vol. 40, pp. 139–157, 2000.
- [9] N. K. Kasabov, *Foundations of neural networks, fuzzy systems, and knowledge engineering*. 1996.
- [10] H. Alvarez and M. A. Peña, “Implicit multidimensional fuzzy sets in Takagi-Sugeno fuzzy modeling,” in *International Congress on Computational Intelligence*, 2001.
- [11] H. Alvarez and M. Peña, “Modelamiento de Sistemas de Inferencia Borrosa Tipo Takagi – Sugeno,” *Av. en Sist. e Inform.*, no. 1, pp. 1–11, 2004.
- [12] J. F. Rendón, “Modelado de series temporales con sistemas de inferencia borrosa

- multidimensionales,” Universidad Nacional de Colombia, 2009.
- [13] J. F. Rendón and J. D. Velásquez, “Una aproximación al modelado de series temporales con sistemas de inferencia borrosa multidimensionales,” 2009.
- [14] Y. Sun and F. Karray, “Hybrid soft computing techniques for heterogeneous data classification,” *Fuzzy Syst.*, pp. 1511–1516, 2002.
- [15] Y. Huang and C. Tian, “Research on Credit Risk Assessment Model of Commercial Banks Based on Fuzzy Probabilistic Neural Network,” *2008 Int. Conf. Risk Manag. Eng. Manag.*, no. 1, pp. 482–486, 2008.
- [16] J. D. Velásquez, “Adaptive Multidimensional Neuro-Fuzzy Inference System for Time Series Prediction,” *IEEE Lat. Am. Trans.*, vol. 13, no. 8, pp. 2694–2699, 2015.
- [17] J. S. R. Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System,” *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.
- [18] J. Kim and N. Kasabov, “HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems,” *Neural Networks*, vol. 12, no. 9, pp. 1301–1319, 1999.
- [19] Y. C. Hwang, Q. Song, and N. Kasabov, “MUFIS: A neuro-fuzzy inference system using multiple types of fuzzy rules,” *IEEE Int. Conf. Fuzzy Syst.*, pp. 1411–1414, 2008.
- [20] N. K. Kasabov and Q. Song, “DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, 2002.
- [21] M. Pérez-Ortiz, S. Jiménez-Fernández, P. A. Gutiérrez, E. Alexandre, C. Hervás-Martínez, and S. Salcedo-Sanz, “A Review of Classification Problems and Algorithms in Renewable Energy Applications,” *Energies*, vol. 9, no. 8, p. 607, 2016.
- [22] K. N. Cover and T. M. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Inf. Theory*, vol. I, no. 13, pp. 21–27, 1967.
- [23] E. Fix and J. L. Hodges, “Discriminatory Analysis: Nonparametric discrimination: consistency properties,” *USAF Sch. Aviat. Med.*, pp. 261–279, 1951.
- [24] E. Fix and J. L. Hodges, “Discriminatory Analysis: Nonparametric discrimination:

- small sample performance,” *USAF Sch. Aviat. Med.*, pp. 280–322, 1952.
- [25] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. New York, 1985.
- [26] R. Pearl and L. J. Reed, “On the rate of growth of the population of the United States since 1790 and its mathematical representation,” *Proc. Natl. Acad. Sci.*, vol. 6, no. 6, pp. 275–288, 1920.
- [27] M. Nerlove and J. S. Press, “Univariate and Multivariate Log-linear and Logistic Models,” *Economic Development Administration and the National Institutes of Health*, vol. RAND—R1306, no. Santa Monica, 1973. 1973.
- [28] S. L. Gortmaker, D. W. Hosmer, and S. Lemeshow, “Applied Logistic Regression.,” *Contemporary Sociology*, vol. 23, no. 1. p. 159, 1994.
- [29] S. C. Serna Pineda, “Comparacion de arboles de regresion y clasificacion y regresion logistica.”
- [30] P. Allison, *Logistic regression using the sas system: Theory and application*. 1999.
- [31] C. I. Bliss, “The Method of Probits,” *Science (80-. )*, vol. 79, no. 2037, pp. 38–39, 1934.
- [32] T. Amemiya and T. Amemiya, “Qualitative response models: A survey,” *J. Econ. Lit.*, vol. 19, no. 4, p. 1483, 1981.
- [33] G. S. Maddala, “Limited-Dependent and Qualitatives Variables in Econometrics,” vol. 95, no. 378, pp. 493–494, 1983.
- [34] S. Cakmakyapan and A. Goktas, “a Comparison of Binary Logit and Probit Models With a,” vol. 2, no. 1, pp. 1–17, 2013.
- [35] a. I. Belousov, S. a. Verzakov, and J. von Frese, “A flexible classification approach with optimal generalisation performance: support vector machines,” *Chemom. Intell. Lab. Syst.*, vol. 64, no. 1, pp. 15–25, 2002.
- [36] E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in induction*. New York: Academic Press, 1966.
- [37] S. Dzeroski and B. Zenki, “Is Combining Classifiers Better than Selecting the Best One?,” *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, 2000.
- [38] T. Dietterich, “Ensemble methods in machine learning,” *Mult. Classif. Syst.*, 2000.
- [39] S. B. Kotsiantis and P. E. Pintelas, “Selective voting,” 2004.

- [40] L. Breiman, “Stacked regressions,” *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.
- [41] Y. Freund and R. E. Schapire, “Experiments with a New Boosting Algorithm,” *Int. Conf. Mach. Learn.*, pp. 148–156, 1996.
- [42] R. E. Schapire, “The strength of weak learnability,” *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
- [43] L. Breiman, *Random Forests*. Machine Learning, 2001.
- [44] R. Maclin and D. Opitz, “Popular Ensemble Methods: An Empirical Study,” *arXiv.org*, vol. cs.AI, pp. 169–198, 2011.
- [45] Tin Kam Ho, “Random decision forests,” *Proc. 3rd Int. Conf. Doc. Anal. Recognit.*, vol. 1, pp. 278–282, 1995.
- [46] T. K. Ho, “The Random Subspace Method for Constructing Decision Forest,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.
- [47] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” *Proc. 23rd Int. Conf. Mach. Learn.*, vol. C, no. 1, pp. 161–168, 2006.
- [48] M. R. Segal, “Machine Learning Benchmarks and Random Forest Regression,” *Cent. Bioinforma. Mol. Biostat.*, 2004.
- [49] H. Deng, G. Runger, and E. Tuv, “Bias of importance measures for multi-valued attributes and solutions,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6792 LNCS, no. PART 2, pp. 293–300, 2011.
- [50] M. Lichman, “UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>],” *Irvine, CA: University of California, School of Information and Computer Science*, 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>.
- [51] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, 2000.
- [52] C. M. Bishop, *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, 2006.
- [53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

- 
- [54] D. A. Valencia, “AMNFIS classifier [https://github.com/deyvagm/amnfisClassifier.git],” *GitHub repository*, 2017. [Online]. Available: <https://github.com/deyvagm/amnfisClassifier.git>.
- [55] P. Jeatrakul and K. W. Wong, “Comparing the performance of different neural networks for binary classification problems,” *2009 Eighth Int. Symp. Nat. Lang. Process.*, pp. 111–115, 2009.