

**MODELADO COMPUTACIONAL (DINÁMICA INTRÍNSECA, TOPOLOGÍA,  
FENOMENOLOGÍA, ETC.) DE PROCESOS NEUROFISIOLÓGICOS DE INDIVIDUOS  
CON CARACTERÍSTICAS COGNITIVAS DIFERENTES.**

**Eduard Gilberto Puerto Cuadros**

Universidad de los Andes  
Facultad de Ingeniería, Doctorado en Ciencias Aplicadas  
Mérida, Venezuela  
2019

**MODELADO COMPUTACIONAL (DINÁMICA INTRÍNSECA, TOPOLOGÍA,  
FENOMENOLOGÍA, ETC.) DE PROCESOS NEUROFISIOLÓGICOS DE INDIVIDUOS  
CON CARACTERÍSTICAS COGNITIVAS DIFERENTES.**

**Eduard Gilberto Puerto Cuadros**

Tesis presentada como requisito parcial para optar al título de:

**PhD. en Ciencias Aplicadas**

Director:

PhD. José Lisandro Aguilar Castro

Centro de Microelectrónica y Sistemas Distribuidos  
Universidad de los Andes  
Facultad de Ingeniería, Doctorado en Ciencias Aplicadas  
Mérida, Venezuela

2019

## Dedicatoria

## Agradecimientos

## Contenido

	<b>pág.</b>
<b>Dedicatoria</b>	<b>iii</b>
<b>Agradecimientos</b>	<b>iv</b>
<b>1. CAPITULO 1: INTRODUCCIÓN</b>	<b>14</b>
1.1 Generalidades	14
1.2 Motivación	19
1.3 Objetivos	20
1.3.1. Objetivo general	21
1.3.2 Objetivos específicos	21
1.4 Antecedentes	21
1.4.1 Trabajos sobre modelos computacionales de base neuronal de individuos con características cognitivas diferentes	22
1.4.2 Trabajos sobre modelos computacionales de base comportamental de individuos con características cognitivas diferentes (Diagnóstico)	24
1.4.2.1 Proyectos de modelos computaciones basados en la evaluación del comportamiento.	24
1.4.2.2 Proyectos de modelos computaciones basados en datos neurocientíficos.	25
1.4.3 Trabajos sobre sistemas de reconocimiento de patrones inspirados en el cerebro	26
1.5 Organización de la Tesis	27
<b>2. CAPITULO 2: MARCO TEÓRICO</b>	<b>29</b>
2.1 Aspectos Neurofisiológicos del Cerebro Humano con Características Cognitivas Diferentes	29
2.1.1 Principios Neurofisiológicos	29
2.1.2 Individuos con Características Cognitivas Diferentes: Autistas	31
2.1.2.1 Teorías sobre el Autismo.	32
2.1.2.2 Características estructurales/anatómicas del cerebro autista.	33
2.1.2.3 Modelos computacionales inspirados en la computación (natural) autista.	37
2.2 Avances en el Conocimiento y Modelado Computacional del Cerebro Humano	40
2.2.1 Modelo de Red Neuronal de Pulsos	41
2.2.2 Otros Modelos Neuronales para el Reconocimiento de Patrones	42
2.2.3 Teoría del Funcionamiento del Cerebro Humano para el Reconocimiento de Patrones PRTM	43
2.2.3.1 Una Jerarquía de Patrones.	45
2.3 Redes Neuronales Artificiales Recientes	49
2.3.1 Aprendizaje Profundo (o Deep Learning)	49
2.3.1.1 Redes Neuronales de Convolución (del inglés, Convolutional Neural Network CNN).	51

2.3.1.2 Redes de Creencias Profundas (del inglés Deep Belief Networks: DBN).	53
2.3.2 Aprendizaje de Máquina Extremo (del inglés, Extreme Learning Machine: ELM)	57
<b>3. CAPITULO 3: ESPECIFICACIÓN COMPUTACIONAL DE PROCESOS NEUROFISIOLÓGICOS DE INDIVIDUOS CON CARACTERÍSTICAS COGNITIVAS DIFERENTES EN TAREAS DE RECONOCIMIENTO DE PATRONES</b>	<b>60</b>
3.1 Un Modelo Computacional del Proceso Neurofisiológico de Reconocimiento de Patrones	60
3.1.1 Formalización matemática de Ar2p para el reconocimiento de patrones estáticos	63
3.1.1.1 Estructura formal del módulo de reconocimiento de patrones Ar2p.	65
3.1.1.2 Teoremas usado por Ar2p para el reconocimiento de un patrón estático.	66
3.1.1.3 Modelo general computacional de Ar2p.	67
3.1.1.4 Caso de prueba ilustrativo de Ar2p aplicado en el reconocimiento de un “Vehículo”.	71
3.1.2 Formalización matemática de Ar2p para el reconocimiento de patrones dinámicos	75
3.1.2.1 Formalización matemática de un patrón dinámico.	76
3.1.2.2 Descripción matemática para un módulo de reconocimiento de patrones dinámicos.	78
3.1.2.3 Teoremas extendidos usado por Ar2p para el reconocimiento patrones dinámicos.	79
3.1.2.4 Caso de reconocimiento de patrón dinámico usando Ar2p: “Caballo galopando”.	81
3.2 Mecanismos de Aprendizaje de Ar2p	85
3.2.1 Modelo de aprendizaje supervisado de Ar2p	85
3.2.1.1 Algoritmo de Aprendizaje nuevo de Ar2p	86
3.2.1.2 Algoritmo de Aprendizaje por refuerzo de Ar2p.	87
3.2.1.3 Casos con aprendizaje.	88
3.3 Aplicaciones de Ar2p en Diversos Dominios	90
3.3.1 Reconocimiento de estilos de conducción usando Ar2p	90
3.3.1.1 Definición formal del patrón de estilo de conducción de un conductor de vehículo.	92
3.3.1.2 Aplicación de Ar2p para el reconocimiento de estilos de conducción.	98
3.3.1.3 Reconocimiento de la “Ira” orientado por características (DpoC).	108
3.3.1.3 Comparación de Ar2p con otros enfoques.	110
3.3.1.3.1 Comparación cualitativa.	111
3.3.1.3.2 Comparación cuantitativa.	111
3.3.1.4 Escenario experimental.	114
3.3.1.4.1 Capacidades de razonamiento.	115
3.3.1.4.2 Capacidades de aprendizaje.	117

3.3.1.4.3 Capacidad de Comunicación.	119
3.3.1.5 Análisis y conclusiones parciales de los resultados del experimento.	120
3.3.2 Reconocimiento de emociones en robots usando Ar2p	124
3.3.2.1. Introducción al problema de reconocimiento de emociones en robots.	125
3.3.2.2. Modelo emocional de prueba para Ar2p.	128
3.3.2.3 Reconocimiento de emociones en robots usando Ar2p.	131
3.3.2.3.1 Reconocimiento de la tristeza en un robot usando Ar2p.	132
3.3.2.5 Escenario de aplicación de Ar2p en el sistema multi-robot: Recolección de Nectar.	134
3.3.2.5 Evaluación de la calidad de reconocimiento de las emociones de los robots usando Ar2p.	137
3.3.3 Reconocimiento de texto usando Ar2p	141
3.3.3.1 Proceso general de Ar2p-Text para reconocer una palabra.	142
3.3.3.2 Tratamiento de tipos de errores ortográficos usando Ar2p-Text.	147
3.3.3.2.1 Corrección de palabras por adición de letras.	148
3.3.3.2.2. Correcciones de palabras por omisión de letras.	149
3.3.3.2.3 Correcciones de palabras por letras cambiadas.	149
3.3.3.2.4 Corrección de palabras por uso de caracteres especiales.	151
3.3.3.2.5. Corrección de palabras por intercambio de letras.	152
3.3.3.2.6 Comparación más sólida, conclusiones de resultados y trabajos futuros.	152
3.4 Análisis General de Resultados de Ar2p	154
<b>4. CAPITULO 4: ARQUITECTURA DE DEEP LEARNING PARA Ar2p:</b>	
<b>Ar2p-DL</b>	<b>158</b>
4.1 Introducción	158
4.1.1 Generalidades	159
4.1.2 Ingeniería de Características	161
4.2 Arquitectura General Propuesta de Ar2p-DL	162
4.2.1 Fase de descubrimiento de descriptores de Ar2p-DL	164
4.2.1.1 Método de descubrimiento de características de Ar2p-DL.	164
4.2.2.2 Método de selección de características.	167
4.2.2 Fase de agregación de Ar2p-DL	170
4.2.3 Fase de clasificación de Ar2p-DL	171
4.3 Aplicaciones de ar2p-dl Usando Diversos Dominios	172
4.3.1 Reconocimiento de dígitos escritos a mano usando Ar2p-DL por descubrimiento de características	174
4.3.1.1 Módulo de reconocimiento del dígito “0” escrito a mano.	175
4.3.1.2 Módulo de reconocimiento del dígito “1”.	175
4.3.2 Reconocimiento de gestos usando Ar2p-DL por descubrimiento de características	176
4.3.3 Resultados of Ar2p-DL usando el enfoque de descubrimiento de características	177
4.3.4 Reconocimiento de estrellas pulsantes (HTRU2) usando Ar2p-DL por selección de características	178

4.3.5 Reconocimiento de canales de distribución (Wholesale Costumer) usando Ar2p-DL por selección de características	179
4.3.6. Reconocimiento de semillas usando Ar2p-DL por selección de características	180
4.3.7. Resultados de Ar2p-DL usando Selección de características	181
4.4. Análisis General de Resultados de Ar2p-Dl y Conclusiones	182
<b>5. CAPITULO 5: ANÁLISIS COMPUTACIONAL DE PROCESOS CONDUCTUALES EN INDIVIDUOS CON CARACTERÍSTICAS COGNITIVAS DIFERENTES</b>	<b>185</b>
5.1 Introducción	185
5.1.1 Generalidades	186
5.2 Marco Teórico	190
5.2.1 Fenómeno de estudio: Autismo (ASD) e Instrumentos de Diagnóstico ADIR y ADOS2	190
5.2.2 Modelos computacionales para predecir el ASD	192
5.2.3 Método usado por el MCFM-ASD: Mapas Cognitivos Difusos	195
5.2.3.1 Mapa Cognitivo Difuso Multicapa.	197
5.3 Modelo Mfcm-Asd	198
5.3.1 Bases del modelo MFCM-ASD	199
5.3.1.1 Descripción de los conceptos de MFCM-ASD.	199
5.3.1.2 Descripción de las relaciones entre los conceptos de la capa ADIR.	203
5.3.1.3 Descripción de las relaciones entre los conceptos de la capa ADOS.	206
5.3.2 Reglas seguidas por el modelo MFCM-ASD en el proceso de toma de decisión	208
5.4 Simulación y Resultados	210
5.4.1 Data experimental	210
5.4.2 Experimentos con casos diagnosticados con ADIR	211
5.4.3 Experimentos con casos de autismo aplicando ADOS2	215
5.4.4 Experimentos con ADIR y ADOS2	219
5.5 Comparación con Otros Métodos	221
5.5.1 Comparación Cuantitativa	221
5.5.2 Comparación Cualitativa	226
5.5.3 Comparación con otros algoritmos de aprendizaje de máquina	228
5.6 Conclusiones y Trabajos Futuros	229
5.7. Consideraciones Éticas	231
<b>6. CAPITULO 6: CONCLUSIONES GENERALES Y TRABAJOS FUTUROS</b>	<b>232</b>
<b>Bibliografía</b>	<b>236</b>

## Lista de Figuras

	<b>pág.</b>
Figura 1.2. Modelo cognitivo para cálculo de fechas de calendario	18
Figura 1.3. Producto de dos números Primos	18
Figura 2.1. Modelo de Neurona Biológica	30
Figura 2.2. Características estructurales del cerebro de Albert Einstein	34
Figura 2.3. Características estructurales del cerebro de Kim Peek	34
Figura 2.4. Características estructurales del cerebro de Rüdiger Gamm	35
Figura 2.5. Características estructurales de las fibras de asociación del cerebro de Temple Grandin	36
Figura 2.6. Características estructurales del ventrículo del cerebro de Temple Grandin	37
Figura 2.7. Modelo de razonamiento de la Teoría de la Mente	38
Figura 2.8. Caras de famosos distorsionadas	44
Figura 2.9. Imagen ambigua	44
Figura 2.10. Diagrama del Módulo de reconocimiento de patrones neocortical	46
Figura 2.11. Jerarquía de módulos de reconocimiento para palabras, letras y líneas	48
Figura 2.12. Modelo base de las RNA basadas en el aprendizaje profundo	50
Figura 2.13. Representación gráfica de CNN para reconocimiento de una imagen	51
Figura 2.14. Esquema de aprendizaje con Máquinas de Boltzmann Restringidas (RBM, en Ingles)	54
Figura 2.15. La estructura de una Máquina de Boltzmann Restringida	54
Figura 2.16. Red de creencias profundas formadas por RBMs	56
Figura 2.17. Arquitectura de ELM	58
Figura 3.1. Esquema del sistema jerárquico de reconocimiento de patrones Ar2p	64
Figura 3.2. Algoritmo de reconocimiento de patrones Ar2p	70
Figura 3.3. Tres imágenes de perfil de un Vehículo tipo automóvil	71
Figura 3.4. Caballo Galopando	81
Figura 3.5. Dos niveles de la Jerarquía de patrones para reconocer el “caballo”	81
Figura 3.6. Algoritmo de Aprendizaje nuevo Ar2p	86

Figura 3.7. Algoritmo de Aprendizaje por refuerzo Ar2p	87
Figura 3.8. Sub-Algoritmo de Aprendizaje por refuerzo Ar2p	87
Figura 3.9. Modelo del Patrón jerárquico para estilos de conducción	98
Figura 3.10. Modelo esquemático de Ar2p para la capa de emociones (felicidad)	100
Figura 3.11. Modelo esquemático de Ar2p para la capa de Estados (relajado)	101
Figura 3.12. Modelo emocional propuesto	128
Figura 3.13. Influencia de los valores de los sub-estados en las emociones	131
Figura 3.14. Los niveles de la jerarquía de patrones para reconocer la emoción	133
Figura 3.15. 10 robots, con valores aleatorios en sus descriptores: BS, OS, SS, IS	138
Figura 3.16. Partes de una P	144
Figura 4.1. Arquitectura de Ar2p-DL	163
Figura 4.2. Ejemplos de características atómicas extraídas desde la basa de datos de dígitos escritos a mano MNIST	174
Figura 4.3. Tipos de Gestos en posición rest y hold	176
Figura 4.4. Atributos seleccionados por Ar2p-DL del dataset HTRU2	178
Figura 4.5. Atributos seleccionados por Ar2p-DI del dataset Wholesale Costumers	179
Figura 4.6. Características seleccionadas por Ar2p-DL de la dataset Seeds	181
Figura 5.1. Modelo general de MFCM-ASD para predecir el ASD	202
Figura 5.2. Conceptos y relaciones de la interacción social recíproca	204
Figura 5.3. Conceptos y relaciones de alteraciones comunicación (sujeto verbal)	204
Figura 5.4. Conceptos de alteraciones de comunicación (sujeto no verbal)	205
Figura 5.5. Conceptos del patrón de comportamiento repetitivo y estereotipado	205
Figura 5.6. Conceptos y relaciones de causalidad del autismo basada en ADIR	206
Figura 5.7. Conceptos y relaciones de la capa OUT-MX	206
Figura 5.8. Conceptos y relaciones de la capa OUT-M4	207
Figura 5.9. Conceptos y relaciones de la capa OUT-ADOS2	208
Figura 5.10. Salida general OUT-TEA	208
Figura 5.11. Inicialización del concepto A1 en 0.33	213
Figura 5.12. La curva de la izquierda es del MCFM-ASD y la curva de la derecha es del modelo	225

## Lista de Tablas

	<b>pág.</b>
Tabla 2.1. Funciones de mapeo usadas en ELM	58
Tabla 3.1. Tupla = < E, U, >	xi
Tabla 3.2. Tupla para el vehículo: vista frontal	xi
Tabla 3.3. Módulo	78
Tabla 3.4. Matriz Ed = <Caballo Galopando> al segundo 1 de la Figura 3.4	84
Tabla 3.5. Descriptores del Patrón de estilos de conducción	92
Tabla 3.6. Descriptores del Patrón de estados del conductor	93
Tabla 3.7. Descriptores del Patrón de las emociones del conductor	94
Tabla 3.8. Eventos entorno a la velocidad del carro	95
Tabla 3.9. Emoción del conductor	95
Tabla 3.10. Estilos de conducción	96
Tabla 3.11. Vista conceptual de la capa de emociones	97
Tabla 3.12. Módulo de reconocimiento del patrón felicidad	100
Tabla 3.13. Módulo de reconocimiento del patrón de estilo de conducción Normal	105
Tabla 3.14. Estructura del módulo de reconocimiento de patrón = “Agresividad”	xi
Tabla 3.15. Estructura del módulo de reconocimiento de patrón = “Ira”	xi
Tabla 3.16. Condiciones fisiológicas para la emoción de Ira de un conductor	108
Tabla 3.17. Matriz del Módulo de reconocimiento del patrón “Ira”	xi
Tabla 3.18. Matriz del Módulo de reconocimiento del patrón “Ira”	xi
Tabla 3.19. Resultados de la capacidad de razonamiento	115
Tabla 3.20. Resultados de la capacidad de aprendizaje	117
Tabla 3.21. Resultados para las capacidades de comunicación	119
Tabla 3.22. Comparación con otros métodos de reconocimiento de estilos de conducción	122
Tabla 3.23. Módulo de reconocimiento del patrón tristeza en Ar2p	133
Tabla 3.25. Capacidad de reconocimiento de emociones en un robot usando Ar2p	139
Tabla 3.26. Módulo de reconocimiento de patrón palabra “PERA”	146
Tabla 3.27. Módulo de reconocimiento de patrón “A”	147

Tabla 3.28. Resultados de los sistemas de corrección ortográfica para L1	148
Tabla 3.29. Resultados de los sistemas de corrección ortográfica para L2	149
Tabla 3.30. Resultados de los sistemas de corrección ortográfica para L3	150
Tabla 3.31. Resultados de los sistemas de corrección ortográfica para L3	151
Tabla 3.32. Resultados de los sistemas de corrección ortográfica para L3	151
Tabla 3.33. Resultados de los sistemas de corrección ortográfica para L4	152
Tabla 3.34. Resultados de los sistemas de corrección ortográfica para L4	152
Tabla 3.35. Resultados para un archivo de texto con 9000 palabras	153
Tabla 4.1. Macro-algoritmo Ar2p-DL	164
Tabla 4.2. Algoritmo de descubrimiento de características para Ar2p-DL	167
Tabla 4.3. Algoritmo de selección de características para Ar2p-DL	167
Tabla 4.4. Algoritmo de Pre-selección	169
Tabla 4.5. Sub-algoritmo del proceso de combinación	169
Tabla 4.6. Sub-algoritmo del proceso de post-selección	169
Tabla 4.7. Algoritmo de Agregación para Ar2p-DL	171
Tabla 4.8. Algoritmo Bottom-Up de clasificación	172
Tabla 4.9. Datasets usadas para el descubrimiento de características	173
Tabla 4.10. Datasets para validar Ar2p-DL por selección de características	174
Tabla 4.11. Módulo de reconocimiento $\Gamma="0"$	175
Tabla 4.12. Módulo de reconocimiento $\Gamma="1"$	176
Tabla 4.13. F-measure (%) de Ar2p-DL y otros algoritmos de clasificación	177
Tabla 4.14. Performance of Ar2p-DL y otros modelos DL	178
Tabla 4.15. Módulo de reconocimiento de estrellas No-Pulsantes (NP). $\Gamma="0"$	179
Tabla 4.16. Módulo de reconocimiento $\Gamma="Horeca"$	180
Tabla 4.17. Resultados de F-measure	181
Tabla 4.18. Resultados de Accuracy	181
Tabla 4.19. Error relativo entre los más altos valores de las métricas y Ar2p-DL	182
Tabla 5.1. Conceptos extraídos de ADIR usados en la primera capa de MFCM-ASD	200
Tabla 5.2. Conceptos de ADOS2 usados en la segunda capa del modelo MFCM	202

Tabla 5.3. Combinaciones de activación del concepto OUTADIR	206
Tabla 5.4. Valores de MFCM-ASD para el caso ADIR-1 de CA1	213
Tabla 5.5. Valores de MFCM-ASD para el caso ADIR-2 de CA2	214
Tabla 5.6. Valores de MFCM-ASD para el caso ADIR-3 de CA3	214
Tabla 5.7. Caso Módulo 1. Módulo 1 total puntos de corte	216
Tabla 5.9. Valores del modelo MFCM-ASD para el caso ADOS2-MODULO 1	217
Tabla 5.8. Calibración de la severidad de los síntomas Módulo 1, Sin palabras	217
Tabla 5.10. Valores del modelo MFCM-ASD para el caso ADOS2-MODULO 2	218
Tabla 5.11. Valores del modelo MFCM-ASD para el caso ADOS2-MODULO 3	219
Tabla 5.12. Resultados de clasificación del MFCM-ASD en los casos ADIR	220
Tabla 5.13. Resultados de clasificación de MFCM-ASD en los casos ADOS2	220
Tabla 5.14. Conceptos del FCM propuesto en (Kannappan, Tamilarasi y Papageorgiou 2011) y su equivalencia aproximada en MFCM-ASD	222
Tabla 5.15. Comparación de los resultados de diagnóstico	223
Tabla 5.16. Comparación con otros algoritmos de aprendizaje de máquina	228

## 1. CAPITULO 1: INTRODUCCIÓN

### 1.1 Generalidades

El cerebro es uno de los fenómenos más complejos del ser humano que ha querido ser recreado artificialmente (De Garis, Shuo, Goertzel, y Ruiting 2010; Feygin, Morris, y Yampolskiy 2018; Attendees y Attendees 2016; Cattell y Parker 2012; Markram 2013; Puerto 2016, 2017; Sandberg 2013; Sandberg y Bostrom, 2008). Actualmente, gracias al avance tecnológico, y en particular, al crecimiento exponencial en el uso de las tecnologías de la información en las neurociencias, se han logrado importantes adelantos sobre la comprensión de diferentes aspectos neurofisiológicos; por ejemplo, sobre el funcionamiento de las neuronas a nivel individual y colectivo, así como también, sobre el comportamiento del cerebro humano en acción (realizando un cálculo matemático, o moviendo un brazo) (Jamali, Azadi-Ahmadabadi, y Asadi, 2018; Liu et al 2017; Puerto, Perez y Aguilar 2014, Puerto y Aguilar, 2016a; Serruya 2017). Estos y otros adelantos se derivan de importantes iniciativas a escala mundial en la última década, dentro de las cuales se destacan: “The Brain Research through Advancing Innovative Neurotechnologies” (BRAIN), cuyo objetivo es mapear el cerebro humano. i.e., describir cómo las neuronas a nivel individual y en colectivo interactúan, tanto en el tiempo como en el espacio; y el proyecto europeo “The Human Brain Project” (HBP), cuyo objetivo es simular el funcionamiento completo del cerebro humano dentro de una computadora; entre otras propuestas vinculadas a la formalización matemática, a la simulación/emulación de procesos neurobiológicos, neurofisiológicos, estructurales y funcionales (Puerto 2016; Puerto y Aguilar 2016a).

En general, la literatura tiene una enorme cantidad de propuestas sobre el comportamiento del cerebro, y en particular, sobre el modelado matemático y computacional del mismo (Puerto y Aguilar 2016a; Brooks, Kamruzzaman, Leider y Tappert 2018). Algunos de los modelos

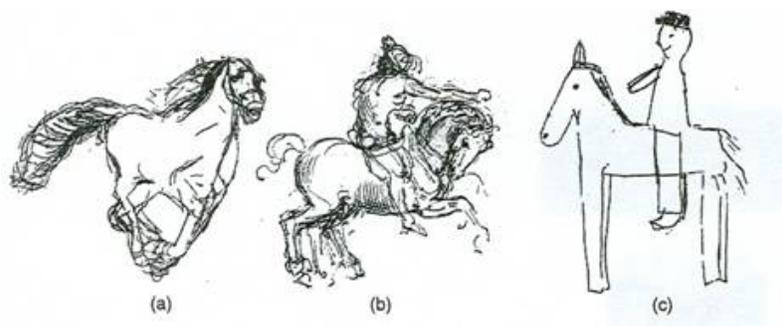
matemáticos de particular interés son los modelos computacionales de Redes Neuronales Artificiales (RNA), como los basados en el concepto de Aprendizaje Profundo (*Deep Learning*, en inglés) (Lauzon 2012; Liu et al 2017; Schmidhuber 2015), y la Teoría de la Mente Basada en el Reconocimiento de Patrones (PRTM, por sus siglas en inglés) (Castel 2015; Kurzweil 2013; Minsky, Kurzweil, y Mann 1991; Puerto 2014). Ambos han sido considerados en este trabajo por sus capacidades de aprendizaje y reconocimiento, aspectos que son claves en individuos con características cognitivas diferentes, especialmente en los Autistas (Motttron y Dawson 2013; Puerto 2017).

Los individuos con características cognitivas diferentes que están en el espectro autista, se caracterizan por dos aspectos: el primero, su gran capacidad cerebral para la resolución de problemas en dominios muy especializados, donde el aprendizaje y la capacidad de reconocimiento juegan un papel importante (Motttron y Dawson 2013; Waterhouse 1988; Waterhouse, London, y Gillberg 2016). El segundo, sus disfuncionalidades conductuales o comportamentales, las cuales limitan su interacción social. Kapur (Kapur 1996) explica en su teoría de la “facilitación funcional paradójica”, la coocurrencia de estos dos factores: la disfunción de un área del cerebro facilita/permite, “paradójicamente”, el super-funcionamiento de otra área de habilidad-capacidad del cerebro.

De acuerdo con sus conductas disfuncionales, un individuo autista puede situarse en diferentes lugares dentro del espectro autista. Dichas conductas pueden ser sociocognitivas, relacionadas con la comunicación e interacción social; o sensoriomotora, relacionadas con los patrones repetitivos y/o restringidos de comportamientos, intereses o actividades (American Psychiatric Association, 2017). El término "espectro" se refiere a la amplia gama de síntomas, habilidades y niveles de deficiencia o discapacidad, que los individuos con Autismo pueden tener. Algunos niños muestran

síntomas de discapacidad ligera, mientras que en otros la discapacidad es severa (Fernández, Mollinedo-Gajate, y Peñagarikano 2018; Mohammadi 2011).

Un caso de autismo es Nadia, una niña prodigio en el dibujo, pero con disfuncionalidad en el lenguaje verbal (Selfe 2012). En la Figura 1.1(a), se tiene el dibujo de un caballo hecho por Nadia a la edad de 5 años, tan bueno como el dibujo de la Figura 1.1 (b), hecho por Leonardo da Vinci, y bastante distante en calidad del dibujo Figura 1.1(c), hecho por un niño normal de 6 años.



**Figure 1.1.** Caballos dibujados: (a) por un autista, (b) por un artista, y (c) por un niño normal. Fuente: Ramachandran, V.S. (2012)

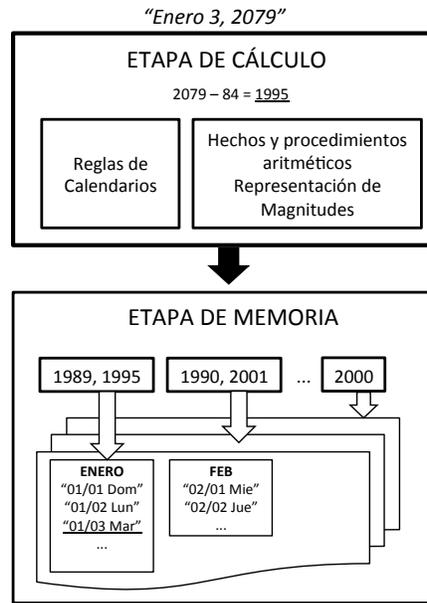
Según lo describe Selfe, el *método de resolución usado por Nadia es*: “comienza el dibujo por detalles aleatorios. En primer lugar, una pezuña, luego la melena del caballo, después su arnés, etc. Luego traza líneas que conectan estas propiedades sueltas. Y cuando lo hace, estas partes unidas se encuentran que están en la posición correcta con respecto a las otras”. Desde un punto de vista computacional, Nadia realiza un proceso jerárquico de composición de patrones crecientes en orden de complejidad, muy consistente con la descripción de la teoría PRTM (Puerto 2014).

Otro caso de habilidad autista es la capacidad de cálculo de calendario (Boddaert et al 2005; Cowan, O’Connor, y Samella 2001), la cual consiste en decir el día específico de la semana de fechas dadas, ya sean pasadas o futuras. El *método* usado por estos individuos consiste en recordar calendarios (explotando su gran capacidad de memoria), y utilizar estrategias de identificación de

patrones que explotan las regularidades del calendario. De esta manera, dada una fecha (por ejemplo, “enero 3 de 2079”, de acuerdo con (Foley 1988), el autista con esta habilidad usa su memoria para guardar las regularidades del calendario, ya que el mismo calendario se repite cada 28 años. Así, encuentra el año más próximo con la misma estructura de calendario, que le permita dar una respuesta correcta (ver Figura 1.2).

En el ejemplo anterior, determinaría que la fecha 2079 tiene el mismo calendario que 1995 que tiene almacenado en su memoria, puesto que  $1995-2079=84$ , y 84 es múltiplo de 28 (ver Figura 1.2). Cuando se ha alcanzado un año cercano al presente, y su calendario está disponible en memoria, la información del día y el mes de ese calendario pueden ser utilizadas para dar la respuesta correcta, la cual, en este caso será un día martes.

Otro caso estudiado de habilidades autistas es el cálculo o identificación de números primos (Welling 1994). Esta habilidad puede requerir cálculos complejos, dado que los algoritmos para tal fin no son sencillos. Muchos autistas, sin embargo, demuestran esta habilidad, aun cuando no tienen las capacidades matemáticas para hacer estos cálculos (multiplicación, división). En (Grandin 2009; Kunda y Goel 2011; Sousa 2006) afirman que el *método de cálculo de números primos* por parte de estos individuos, consiste en la manipulación imaginaria de imágenes mentales (pensamiento visual) que representan dicho cálculo: "Es concebible que este sistema (la manipulación de imágenes) les permita representar ciertas operaciones computacionales, o los resultados de ellas, en forma de imágenes" (ver Figura 1.3).



**Figura 1.2.** Modelo cognitivo para cálculo de fechas de calendario. Fuente. Foley, J.M. (1988)



**Figura 1.3.** Producto de dos números Primos. Fuente. Tammet, D. (2013)

La Figura 1.3 es una representación en imágenes de lo que sería una operación matemática del producto de dos números primos (53) y (131), y su resultado 6943. La imagen fue creada por el propio autista Daniel Tammet, para mostrar lo que ocurre a nivel de la manipulación de imágenes en su cerebro (Tammet 2013). Análogamente, a través de una operación de manipulación mental de imágenes, se puede distinguir un número primo de un número normal, siguiendo las mismas particularidades de pensamiento visual en números.

Es de notar que, en los tres métodos de resolución de problemas descritos, las habilidades cerebrales presentes en estos individuos están relacionadas con métodos de reconocimiento y aprendizaje de patrones, con una representación jerárquica de los mismos; entre las que se destacan, la gran capacidad de detalle, la gran destreza para asociar/relacionar información existente y nueva de manera muy rápida, la gran capacidad de memorización, la gran capacidad de aprendizaje especializado y rápido, entre otras. A estos problemas resueltos con métodos singulares de reconocimiento y aprendizaje, los hemos denominado *problemas de alta complejidad cerebral*.

Por otro lado, para simular esas capacidades singulares de resolución de dichos problemas, se requieren nuevos modelos cerebrales artificiales, que consideren una representación jerárquica de los patrones, y métodos de reconocimiento y aprendizaje basados en ella, diferentes a los que clásicamente aparecen en la literatura. Una primera hipótesis, en esta dirección, fue considerar la teoría de la mente basada en el reconocimiento de patrones, o PRTM (Castel 2015; Kurzweil 2013; Minsky, Kurzweil y Mann 1991; Puerto 2014), junto con el enfoque de aprendizaje profundo (*Deep Learning*, en inglés) (Lauzon 2012; Liu et al 2017; Schmidhuber 2015), como base para el desarrollo de un primer modelo cerebral artificial basado en esos principios. Es por esto que, en este trabajo, se analizan dichas teorías y enfoques de aprendizaje de máquina, y se propone un modelo computacional del cerebro basado en los mismos.

## **1.2 Motivación**

En el estudio de los individuos autistas, y en general, de individuos con características cognitivas diferentes, han sido apreciables los esfuerzos y enfoques propuestos por generar teorías, modelos computacionales, aplicaciones y formalismos, que busquen comprender sus comportamientos y razonamientos (Galitsky 2013, 2016; Gowen y Hamilton 2013; O’Loughlin y Thagard 2000;

Puerto 2017). Sin embargo, a nivel de su cognición, aquella que trata con la capacidad de resolución de problemas, especialmente la habilidad para representar, reconocer y aprender patrones complejos, no ha sido ampliamente estudiada.

En particular, esto abre la posibilidad de explorar una nueva aproximación computacional para incursionar en la resolución de problemas de reconocimiento y aprendizaje de patrones complejos (estáticos como dinámicos), basada en la emulación del comportamiento del cerebro de estos individuos. Por otra parte, existe además actualmente la necesidad de una evaluación automática de su comportamiento, que permita identificarlo en el espectro autista (diagnóstico). En ese sentido, es posible pensar en ofrecer una alternativa tecnológica asequible, de apoyo a padres, cuidadores o médicos, para el diagnóstico del autismo, el cual tradicionalmente se caracteriza por ser de alto costo.

Sobre la anterior base, la investigación doctoral aquí desarrollada buscó elaborar principalmente un modelo computacional de uno de los procesos neurofisiológicos básicos de representación, reconocimiento, y aprendizaje de patrones en el cerebro, basado en los métodos, teoría y enfoque antes señalados. Para ello, se creó una especificación algorítmica y computacional para modelar estos aspectos, usando la *teoría de la mente basada en el reconocimiento de patrones* y el enfoque de *aprendizaje profundo*. Además, se desarrolló un *mapa cognitivo difuso multicapa* para el diagnóstico del trastorno del espectro autista. En ambos casos, se hicieron pruebas de los modelos en diferentes dominios, se compararon con modelos similares, para determinar la calidad, y conocer sus bondades y debilidades en sus funcionamientos.

### **1.3 Objetivos**

A partir del contexto anterior y motivación, se plantearon los siguientes objetivos de la tesis:

### **1.3.1. Objetivo general**

Modelar computacionalmente la dinámica neuronal y la topología de los procesos neurofisiológicos de individuos con características cognitivas diferentes, intrínseca en la resolución de problemas de alta complejidad cerebral.

### **1.3.2 Objetivos específicos**

- Caracterizar la dinámica y topología neuronal en la resolución de problemas de alta complejidad cerebral, desde una mirada externa al cerebro.
- Caracterizar la dinámica y topología neuronal en la resolución de problemas de alta complejidad cerebral, desde una mirada interna multidisciplinar (biológica, neurofisiológica, sistémica).
- Elaborar modelos basados en los procesos descritos del comportamiento del cerebro, para la resolución de problemas de alta complejidad cerebral.
- Integrar los modelos propuestos, para representar el funcionamiento del cerebro en la resolución de problemas de alta complejidad cerebral.
- Elaborar un prototipo computacional del modelo, y hacer pruebas en problemas reales.

## **1.4 Antecedentes**

En esta sección, se resumen los trabajos y proyectos más relevantes vinculados a la tesis, que buscan modelar computacionalmente diferentes aspectos, y a diferentes escalas (parcial o total), el funcionamiento del cerebro de individuos con capacidades cognitivas diferentes. Inicialmente, en la sección 1.4.1 se cubren los trabajos de modelos computacionales de base neuronal que siguen las teorías descritas sobre el autismo (ir a sección 2.1.2 para detalles de esas teorías). Luego, en la sección 1.4.2 se presenta un segundo grupo de trabajos, que cubre los modelos computacionales

de base comportamental para el diagnóstico del autismo. Por último, en la sección 1.4.3 se presentan diversos trabajos sobre modelos computacionales neuronales de reconocimiento de patrones.

#### **1.4.1 Trabajos sobre modelos computacionales de base neuronal de individuos con características cognitivas diferentes**

Los proyectos de modelos computacionales de base neuronal, simulan diversos aspectos de procesos neurofisiológicos en el procesamiento de información autista. Por ejemplo, en (Galitsky 2016) se propone un modelo computacional de la Teoría de la Mente Autista (TM). Este modelo computacional se enfoca en resolver el problema de la Teoría de la Mente autista, para apoyar el mecanismo de razonamiento de actitudes mentales de individuos autistas. El problema de la TM consiste en, dado un estado mental arbitrario, generar un conjunto de acciones definidas (Galitsky 2016). En (O'Loughlin y Thagard 2000) se propone un modelo computacional de la teoría de coherencia central débil. Este modelo simula tal coherencia en una red neuronal. Usan la teoría de coherencia como satisfacción de restricciones, la cual consiste en maximizar la satisfacción de restricciones positivas y negativas entre elementos. Los elementos son representaciones como conceptos, proposiciones, partes de una imagen, objetivos, y acciones. La relación entre dos elementos es de causalidad o no (uno es consecuente del otro). Son consecuentes unos de otros porque facilitan, se deducen, están asociados, o explican. No lo son cuando los elementos presentan inconsistencias, incompatibilidades, o una relación negativa. Si dos elementos están restringidos positivamente entre ellos, entonces, las neuronas que los representan están conectadas por un enlace excitatorio; por el contrario, si dos elementos tienen una restricción negativa, entonces, sus neuronas están conectadas por un enlace inhibitorio. En (Carvalho 1999) se propone un modelo de red neuronal para simular el proceso de neuro-desarrollo, a través de la construcción

de mapas neuronales. Un mapa neural es una capa de neuronas cuyas propiedades están sistemáticamente relacionadas con el mundo externo (por ejemplo, espacio visual o auditivo), y representan regiones del sistema nervioso. La simulación del modelo da un indicio de qué regiones del cerebro son responsables de las representaciones de alto nivel (e.g., de sus funciones cognitivas), y cómo estas se ven afectadas en personas autistas. Estas deficiencias pueden dar cuenta de sus síntomas y capacidades especializadas.

Otro trabajo se propone en (Vattikuti y Chow 2010), el cual simula a nivel microscópico propiedades de la cinética sináptica y de neurotransmisores. Dicho modelo es aplicado al movimiento ocular autista, y proporciona una relación directa entre la función cortical cerebral y el comportamiento Autista. Duch (Duch et al 2012) usa la técnica de dinámica simbólica difusa, o FSD (por sus siglas en inglés, *Fuzzy Symbolic Dynamics*), para la simulación y visualización de redes neuronales, para evaluar la neurodinámica cerebral a largo plazo e interpretar los desórdenes del espectro autista. Rosenberg (Rosenberg, Patterson y Angelaki 2015) usa simulaciones de redes neuronales para mostrar que una reducción de las señales inhibitorias en el modelo normalizado (en la computación neuronal) puede explicar las consecuencias de la percepción autista. El modelo muestra la relación de las bases genéticas/moleculares del autismo y sus características comportamentales. El modelo se extiende a varios circuitos sensoriales y regiones del cerebro, incluyendo la representación de los olores, los efectos moduladores de la atención visual, la codificación del valor, y la integración de la información multisensorial (Carandini y Heeger 2012).

Como se observa en los trabajos anteriormente presentados, estos son modelos computacionales neuronales completamente alejados de los modelos neuronales artificiales (redes de retro-

propagación, el modelo de Kohonen, las redes neuronales aleatorias, entre otros modelos), que clásicamente han sido usados en el mundo de la inteligencia artificial.

#### **1.4.2 Trabajos sobre modelos computacionales de base comportamental de individuos con características cognitivas diferentes (Diagnóstico)**

Hay diferentes enfoques de proyectos computacionales para predecir el autismo: proyectos basados en la evaluación del comportamiento (Bone et al 2015, 2016; Crippa et al 2015; Farsi, Doctor, Petrovic, Chandran y Karyotis 2017; Oikonomou y Papageorgiou 2013; Ojeda 2015; Papageorgiou, Stylios y Groumpos 2003; Puerto 2017; Reyes, Ponce, Grammatikou y Molina 2014; Rosenberg, Patterson y Angelaki 2015), proyectos basados en datos neurocientíficos (Oikonomou y Papageorgiou 2013; Subbaraju, Sundaram, Narasimhan y Suresh 2015; Bi et al 2018) y proyectos que combinan ambos enfoques (Anirudh y Thiagarajan 2017). Estos proyectos nacen de la necesidad de acelerar el tiempo de detección, o mejorar la sensibilidad, especificidad o precisión del proceso de diagnóstico autista. A continuación, se detallan sus enfoques y métodos.

##### **1.4.2.1 Proyectos de modelos computaciones basados en la evaluación del comportamiento**

Cohen y otros (Cohen, Sudhalter, Landon-Jimenez y Keogh 1993) han propuesto una Red Neuronal Artificial para discriminar entre autismo y retardo mental, basado en el estándar de diagnóstico ABI (*Autism Behavior Interview*). La red usada en este trabajo fue una red neuronal de Backpropagation. En (Arthi y Tamilarasi 2008) han propuesto un sistema neuro-difuso que convierte las respuestas de los padres en un cuestionario en valores difusos. Estos valores son luego usados por un sistema de reglas causales “si-entonces”, y la salida difusa se convierte en la entrada para redes neuronales de *Backpropagation* predefinidas. Otro enfoque es el sistema de

screening<sup>1</sup> KBS (*Knowledge Based Screener* (KBS)), el cual es un sistema experto con conocimiento fáctico y heurístico, para analizar el desarrollo de los niños e identificar desordenes en el mismo (Farsi et al 2017). También, en (Ojeda 2015; D P Wall, Kosmicki, DeLuca, Harstad, y Fusaro 2012; Wall, Dally, Luyster, Jung y DeLuca 2012) han propuesto un clasificador basado en árboles de decisión (ADTree), y 8 preguntas del módulo 1 del Instrumento de diagnóstico ADOS (por sus siglas en inglés, de *Autism Diagnostic Observation Schedule*). El módulo 1 define la escala de observación para el diagnóstico del autismo, Hay otra versión de este enfoque, llamado ADTree, que detecta el autismo a través de 7 preguntas del instrumento ADIR (por sus siglas en inglés, *Autism Diagnostic Interview, Revised*). ADIR es una entrevista que se hace a los padres para el diagnóstico del autismo. En (Tarantino, Mazza, Valenti y De Gasperis 2016) han desarrollado una herramienta computacional, que incluye el diagnóstico, en un entorno gamificado de juegos imaginativos. En (Ojeda 2015) se ha definido un método basado en algoritmos genéticos para el diagnóstico autista. Por último, en (Bone et al 2015) han estudiado el uso de las técnicas de aprendizaje de máquina (*Machine Learning*) para la detección del autismo, usando grandes colecciones de datos. Muchos de estos métodos deben usar un gran tamaño de muestra, para poder entrenar los modelos.

#### **1.4.2.2 Proyectos de modelos computaciones basados en datos neurocientíficos**

Subbaraju y otros (Subbaraju et al 2015) ha propuesto un método de detección del autismo usando datos MRI (*Structural Magnetic Resonance Imaging*), y un clasificador neuronal funcional de base

---

<sup>1</sup> Estos métodos de screening son análisis simples que se utilizan para obtener una información rápida sobre una patología. Generalmente, si el resultado es positivo, se necesita realizar un análisis más concluyente para confirmar la enfermedad.

radial metacognitivo extendido (*Metacognitive Radial Basis Function Neural Classifier* (EMcRBFN)). En (Zhang et al 2018) se ha diseñado un método automatizado para el análisis de la conectividad de la materia blanca, basado en la tractografía guiada por la resonancia funcional cerebral, para detectar el autismo. En (Anirudh 2017) se ha definido un método que combina diferentes tipos de propiedades (comportamiento, información estructural y funcional), que actúan como bio-marcadores o bio-descriptores en un modelo predictivo para diferentes condiciones neuro-patológicas. Además, ellos desarrollaron una versión de la red neuronal convolucional G-CNNs (*Graph Convolutional Neural Networks*) para la clasificación del autismo. En (Abbas, Garberson, Glover, y Wall 2018) se ha propuesto una herramienta para la detección temprana del autismo aplicando algoritmos de Aprendizaje de Máquina. Esta herramienta combina dos métodos de screening: una evaluación estructurada orientada a padres, y otra semi-estructurada basada en el comportamiento clave de los niños, usando para ello videos caseros. Finalmente, en (Abbas et al 2018; Li, Sharma, Meng, Purushwalkam y Gowen 2017; Thabtah 2018), se emplea el aprendizaje automático en la clasificación del autismo, y se discute sus ventajas y desventajas.

### **1.4.3 Trabajos sobre sistemas de reconocimiento de patrones inspirados en el cerebro**

Conforme se van generando más conocimiento sobre el funcionamiento del cerebro, nuevos modelos de Redes Neuronales (RNAs) surgen. Uno de estos modelos son las RNAs de Pulsos (SNN, del inglés *Spiking neural network*) (Schliebs y Kasabov 2013). Estas redes se inspiran a partir de que las células del sistema nervioso generan una serie de potenciales de acción que se pueden analizar como señales en el dominio del tiempo (Markram, Gerstner, y Sjöström 2012; Shulz y Feldman 2013). En (Christophe et al 2015; Grassia, Levi, Doukkali, y Kohno 2018; Kasabov, Dhoble, Nuntalid, y Indiveri 2013; Schliebs y Kasabov 2013; Vestbøstad 2018) se describen varios modelos recientes de Redes Neuronales de Pulsos, aplicadas en varios trabajos

para el reconocimiento de patrones. En (Christophe et al 2015) se entrena una SNN para el reconocimiento de patrones (imágenes de círculos y cruz cruzada), basado en el modelo *Spike Timing Dependent Plasticity* (STDP). STDP corresponde a la forma en que las conexiones entre las neuronas cambian de acuerdo con la actividad de pulsos (spiking) en la red. Otros trabajos han entrenado SNN para la detección de caracteres y líneas (Wu, McGinnity, Maguire, Valderrama-Gonzalez, y Cai 2009).

Ahora bien, en general, existen muchos algoritmos para la resolución del problema de reconocimiento de patrones desde hace tiempo (Aguilar y Hernández 2009; Nasrabadi 2007; Michalski, Carbonell, y Mitchell 2013) Algunos de ellos, basados en modelos computaciones del cerebro, tal como la Teoría de Resonancia Adaptativa (Altamiranda, Aguilar, y Hernandez 2015) o el modelo neuronal aleatorio (Aguilar 2004). Actualmente, el aprendizaje profundo, o *Deep Learning*, ha aportado avances importantes y perfeccionado el reconocimiento de patrones (Deng y Yu 1999; Lecun, Bengio y Hinton 2015; Liu et al 2017; Schmidhuber 2015). Las arquitecturas bajo aprendizaje profundo generan modelos composicionales, donde el objeto (a analizar y reconocer) es expresado como una composición de niveles de imágenes primitivas (parecido al modelo de la teoría de la mente basado en el reconocimiento de patrones).

## **1.5 Organización de la Tesis**

A continuación, se plasma la organización de la tesis. El *capítulo 2* se basa en los trabajos (Puerto, Perez y Aguilar 2014; Puerto y Aguilar 2016a, Puerto 2016; Puerto 2017), y describe el marco teórico de la tesis, haciendo hincapié en los aspectos neurofisiológicos del cerebro de los individuos con características cognitivas diferentes; en la teoría de la mente explicada desde un enfoque sistémico y formal del funcionamiento de la misma; y en las redes neuronales avanzadas que emergen como solución al problema del reconocimiento de patrones. El *capítulo 3* se basa en

los trabajos (Aguilar et al 2017; Aguilar et al 2019; Puerto y Aguilar 2016b, 2017, 2019; Puerto Aguilar y Chávez 2017, 2018; Gil, Aguilar, Puerto y Dapena 2018, 2019). En este capítulo se especifica el modelo computacional de los procesos neurofisiológicos, inspirado en la teoría de la mente para el reconocimiento de patrones; realizándose su formalización matemática, definiendo sus mecanismos de aprendizaje, y mostrándo diversas aplicaciones en diversos dominios, como reconocimiento de estilos de conducción, reconocimiento de emociones en robots, reconocimiento de texto, entre otras aplicaciones. El *capítulo 4* se basa en los trabajos (Puerto, Aguilar, Reyes y Sarkar 2018a, 2018b; Vargas, Aguilar y Puerto 2018). En este capítulo se describe una arquitectura de *Deep learning*, que extiende las capacidades de aprendizaje y reconocimiento de patrones del modelo base (ver capítulo 3). Particularmente, se presenta la arquitectura, sus fases constitutivas, y varias aplicaciones en diferentes dominios: reconocimiento de dígitos escritos a mano, reconocimiento de gestos, reconocimiento de semillas, reconocimiento de estrellas pulsantes, entre otras aplicaciones. El *capítulo 5* se base en el trabajo (Puerto 2017; Puerto, Aguilar, López y Chávez 2018). En este capítulo se describe un sistema para el diagnóstico del autismo; su formalismo matemático, simulaciones y análisis de resultados. Por último, el capítulo 6 presenta las conclusiones y trabajos futuros.

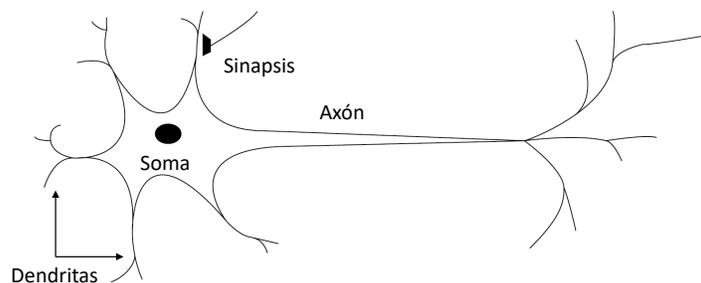
## 2. CAPITULO 2: MARCO TEÓRICO

### 2.1 Aspectos Neurofisiológicos del Cerebro Humano con Características Cognitivas

#### Diferentes

#### 2.1.1 Principios Neurofisiológicos

La *neurofisiología* es una rama de las neurociencias que estudia el funcionamiento del sistema neuronal; su actividad bio-eléctrica, su comportamiento a nivel individual y en colectivo (circuito neuronal), su capacidad de plasticidad neuronal, entre otros procesos neurofisiológicos (Chéron 2018; Carpenter, Roger y José 1998; Johnston, Daniel, y Samuel 1994; Renato 2003). Actualmente, existen diversos enfoques que tratan de explicar experimentalmente el fenómeno del procesamiento de la información en el cerebro. Desde estos experimentos, se han establecido varios principios sobre el funcionamiento del sistema neuronal (Wickens 2019; Eric, James and Thomas 2000). Un primer principio es la *ley de la polarización dinámica* de las neuronas (las neuronas vistas como células discretas (discontinuas)), postulado por Ramón y Cajal (Renato 2003). Esta ley indica que las corrientes que conducen información bioeléctrica en las células nerviosas (neuronas), fluyen desde las ramificaciones dendríticas hacia el cuerpo de la neurona, donde se procesa dicha información, y de éste hacia las ramificaciones terminales o axones, para contactar a través de la sinapsis con otra u otras neuronas. Es decir, la transmisión neuronal es unidireccional (desde las dendritas al soma, y luego a las arborizaciones del axón, ver Figura 2.1).



**Figura 2.1.** Modelo de Neurona Biológica

Otros principios están relacionados con la organización del cerebro. Por ejemplo: el principio de *organización jerárquica columnar del córtex* (Mountcastle 1997), el cual establece que el córtex se organiza en columnas y capas de células piramidales, que muestran una gran regularidad. Por ejemplo, la organización del córtex visual en animales y humanos (Shatz 2013; Hubel, David y Torsten 1959; Hubel, David y Torsten 1962; Peterson 2014). Por otro lado, el principio de *reúso neuronal* (Anderson 2010) indica que es normal que un circuito que ha sido establecido para un propósito dado, sea redefinido (re-explotado, reciclado, redistribuido), o sea utilizado para diferentes usos, sin perder sus funciones originales.

Además, otros principios están relacionados con el aprendizaje (Sompolinsky y Haim 1987; Caporale et al 2008): La *regla de Hebb* establece la base neurofisiológica del aprendizaje, describe que el valor de una conexión sináptica se incrementa si las neuronas de ambos lados de dicha sinapsis se activan repetidas veces de forma simultánea (Lowel 1992). También, el descubrimiento de circuitos corticales fundamentales encargados de realizar *cálculos en el córtex* (región del cerebro que juega un rol clave en la cognición, atención, percepción, lenguaje, entre otras funciones cognitivas), tales como el cálculo de selectividad y el cálculo recurrente de reducción de entradas externas en el área sensorial cortical (Miller y Kenneth 2016). Otro principio es el mecanismo de las *neuronas espejo*, las cuales nos permiten empatizar con otras personas o con quienes nos envuelven (Rizzolatti et al 2016).

Esos son algunos de los principios que han ido surgiendo, gracias al uso de tecnologías avanzadas para el estudio del cerebro (Churchland 2016; Puerto 2016; Eric, James and Thomas 2000). Muchos de estos principios de la neurofisiología han sido explotados por la teoría matemática y computacional, dando origen a un área de investigación denominada Redes Neuronales Artificiales (RNA) (Aguilar 2009; Aguilar y Rivas 2001; Samarasinghe 2016). Algunos de los modelos matemáticos conocidos que usan estos principios e intentan describir, normalmente parcialmente, el comportamiento del cerebro, son (Aguilar 2009; Jain et al 1996; Dayhoff 1990; Abbott 1990): Las Redes de Retropropagación (del inglés, *backpropagation*), el Perceptrón, el modelo de Hopfield, el modelo de Kohonen, el Modelo de McCulloch Pitts, las Redes con Propagación hacia delante (del inglés, *Feed-Forward Networks*), las Redes Recurrentes (del inglés *Recurrent Networks*), la Teoría de Resonancia Adaptativa (del inglés, *Adaptive Resonance Theory*), el modelo Hebbiano, el modelo de Hodgkin y Huxley, entre otros. En la sección 2.2 y 2.3 se describen varios de estos modelos computacionales fundamentales de RNA, y otros modelos más recientes.

Por otra parte, también existen diversas teorías más especializadas que buscan explicar experimentalmente los procesos neurofisiológicos de individuos con características cognitivas diferentes. A continuación, se resumen estas teorías.

### **2.1.2 Individuos con Características Cognitivas Diferentes: Autistas**

En esta sección, se describen tres bases teóricas que tratan de explicar las características cognitivas de los individuos que están en el espectro autista (Galitsky 2016), una desde la psicología, otra desde las características estructurales/anatómicas del cerebro de los mismos, y finalmente, otra basada en modelos computacionales inspirados en estas.

### 2.1.2.1 Teorías sobre el Autismo

El trastorno del espectro autista es una condición de vida sujeta a un desarrollo anormal del cerebro, que afecta el funcionamiento cognitivo, social y motor del individuo (APA, 2013, APA 2017). Existen diversas teorías que proponen explicar esta condición cerebral, desde diferentes enfoques (Puerto 2017; Fakhoury 2015; Zimmerman 2008):

- Teoría de la *Teoría de la Mente (Theory of Mind: TM)*, la cual se refiere a la habilidad para inferir y entender lo que uno mismo y otras personas están pensando (sus conocimientos, sus creencias, sus intenciones y deseos). Esta capacidad para razonar sobre actitudes mentales, se encuentra afectada en individuos que están en el espectro autista (Baron-Cohen et al 2002), lo cual, para ellos, conduce a dificultades en los comportamientos basados en el razonamiento mental, como el juego de simulación, o la capacidad de decir mentiras (Ramachandran 2007; Rizzolatti, Giacomo y Corrado 2016; Ozonoff et al 1991).
- Teoría de la *Coherencia Central Débil (Weak Central Coherence: WCC)*, la cual se refiere a la incapacidad de los individuos con autismo para procesar información del contexto, aun teniendo una remarcable capacidad para recordar detalles (Plaisted 2015). Por ejemplo, los individuos autistas tienen mayor dificultad en recordar de que se trataba una historia que otros individuos normales, aun siendo buenos en recordar cadenas de palabras sueltas (Hermelin y O'connor 1967). De acuerdo con esta teoría, el cerebro autista tiene un estilo cognitivo que favorece la segmentación. La segmentación es debida a la limitación o imposibilidad del cerebro, en procesar la información de manera más holística, i.e., los individuos con autismo son muy buenos en fijarse en los detalles, pero tiene problemas en integrar estos detalles en un todo, con sentido, en un contexto dado (Happé y Frith 2006).

- Teoría de la *Disfunción Ejecutiva (Executive Dysfunction: DF)* (Tolero et al 2015; Tsermetseli, Tabares y Kouklari 2018; Leung et al, 2016), la cual se refiere a la incapacidad de los individuos autistas de mantener un comportamiento apropiado en la resolución de problemas (Garon et al 2018; Pennington et al 1996; Hill y Elisabeth 2004; Russell 1997). Esto se entiende a partir de la definición de *Función Ejecutiva*, como un constructo cognitivo usado para describir conductas dirigidas hacia una meta, orientadas hacia el futuro, que se consideran mediadas por los lóbulos frontales. Incluyen la planificación, inhibición de respuestas prepotentes, flexibilidad, búsqueda organizada, y memoria de trabajo (Ozonoff et al 1994). Por tanto, la disfunción de esta capacidad en relación con el autismo, se debe principalmente a alteraciones del lóbulo frontal, lo cual genera estas disfunciones ejecutivas (Fisher y Happé 2005) (Ozonoff 1997).

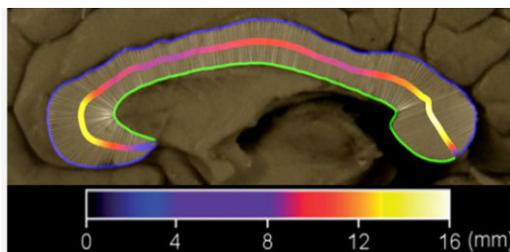
Es de notar, entonces, que muchas de estas disfuncionalidades y capacidades de los individuos autistas, se deben en parte a diversas anomalías estructurales/anatómicas y funcionales de sus cerebros.

### **2.1.2.2 Características estructurales/anatómicas del cerebro autista**

En esta sección, se describen las características estructurales/anatómicas del cerebro de cuatro individuos con características cognitivas diferentes excepcionales: Albert Einstein, Kim Peek, Rüdiger Gamm y Temple Grandin.

- *Características estructurales/anatómicas de Albert Einstein* (Witelson 1999; Men 2013)

Los hemisferios cerebrales del Albert Einstein estaban mejor conectados de lo habitual. El cuerpo calloso del cerebro de Einstein era especialmente más denso, lo que probablemente permitía mayores conexiones nerviosas (Men 2013).



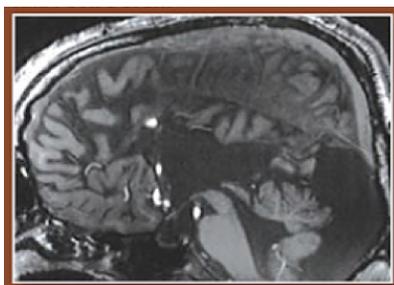
**Figura 2.2.** Características estructurales del cerebro de Albert Einstein. Fuente. Men, et al. (2013)

Los colores en la Figura 2.2 indican la variabilidad de grosos de las subdivisiones del cuerpo calloso. Además, el cerebro de Einstein tenía una alta proporción de células gliales que daban soporte a las neuronas, e intervenían activamente en el procesamiento de información (Diamond 2017).

Por otro lado, el lóbulo parietal inferior izquierdo de Einstein (la región que procesa el pensamiento matemático, la visión tridimensional, las relaciones espaciales, y otros procesos mentales) era un poco más grande que el derecho. También destacaba su corteza prefrontal de mayor tamaño.

- *Características estructurales/anatómicas de Kim Peek* (Treffert 2005)

Kim fue una persona con una extraordinaria memoria, su principal habilidad. A diferencia de Einstein, cuyo cuerpo calloso era denso, en el cerebro de Kim Peek carecía del mismo, i.e., sus hemisferios estaban desconectados (ver Figura 2.3).



**Figura 2.3.** Características estructurales del cerebro de Kim Peek. Fuente. Treffert, et al. (2005)

La Figura 6 muestra la diferencia estructural cerebral marcada del cerebro de Kim Peek: Carece del corpus calloso, y es un cerebelo no completo (le faltaban zonas responsables de ciertas funciones motoras, era más pequeño de lo normal), esto puede explicar ciertas dificultades de coordinación de Kim.

- *Características estructurales/anatómicas de Rüdiger Gamm* (Houdé 2003)

Rüdiger es un individuo capaz de calcular complejas operaciones matemáticas, como la raíz 5 de un número o su potencia 9, hacer divisiones con exactitud de muchos decimales en sus respuestas, entre otras operaciones. Los procesos de cálculo que realiza Gamm toman áreas del cerebro implicadas en la memoria episódica, incluyendo el área frontal media derecha y el giro parahipocampal. La Figura 2.4 muestra la activación de estas partes del cerebro de Gamm realizando un cálculo.



**Figura 2.4.** Características estructurales del cerebro de Rüdiger Gamm. Fuente. Houdé, (2003)

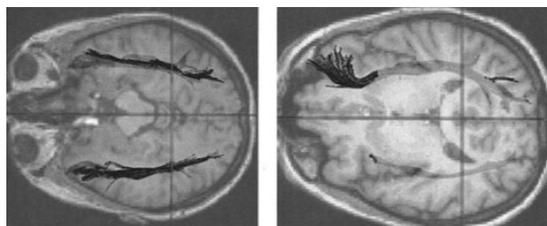
Las diferentes zonas rojas son las partes (de más) que se activaron en el cerebro de Gamm realizando un cálculo, y las zonas verdes son las áreas comunes activadas en el mismo proceso de cálculo por cualquier persona y Gamm.

Esta activación extra de zonas del cerebro de Gamm le permiten explotar la capacidad ilimitada de memoria a largo-plazo, para recordar información relevante en la solución del problema, tal como la secuencia de pasos y los resultados parciales necesarios para un cálculo complejo. A diferencia de personas normales, que usan la limitada memoria de trabajo (o memoria a corto

plazo, la que usamos al memorizar un número de teléfono mientras lo anotamos en la agenda o en nuestro celular).

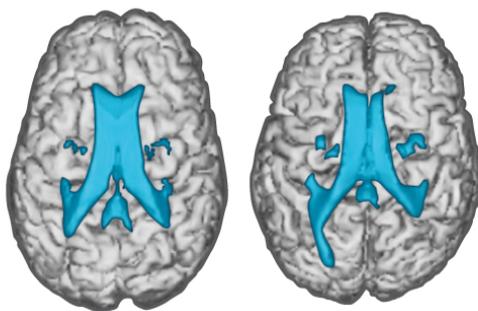
- *Características estructurales/anatómicas de Temple Grandin* (Grandin 2013)

Temple es una autista que se caracteriza por su capacidad de pensamiento visual (i.e., en imágenes). Entre sus características neurofisiológicas y estructurales cerebrales se encuentran: Muestra menos activación en respuesta a caras que lo normal. Tiene un cerebro sobre-conectado, en este caso significa que sus fascículos o fibras de asociación tienen muchas más conexiones que lo usual (ver Figura 2.5). A la izquierda se muestran las fibras de asociación mejor conectadas procesando información visual en el cerebro de Temple, que las de un cerebro normal (derecha). En la Figura 2.5, a la izquierda se observa el ventrículo izquierdo de un cerebro normal, y a la derecha el ventrículo izquierdo de Temple, mucho más grande que lo normal.



**Figura 2.5.** Características estructurales de las fibras de asociación del cerebro de Temple Grandin. Fuente. Grandin, T. (2013)

Así, tiene un ventrículo izquierdo mucho más grande que el derecho (aprox. de 57%, y normalmente es sólo del 15%). Al tener un ventrículo izquierdo mucho más grande, este invade su córtex parietal (asociado con la memoria de trabajo, útil para hacer calculo), lo cual puede explicar sus problemas para el álgebra (ver Figura 2.6).



**Figura 2.6.** Características estructurales del ventrículo del cerebro de Temple Grandin. Fuente. Grandin, T. (2013)

### 2.1.2.3 Modelos computacionales inspirados en la computación (natural) autista

Similar al origen de las RNA, muchos de las teorías sobre el fenómeno autista han sido explotados por la teoría matemática y computacional, para su simulación y desarrollo de sistemas de diagnóstico autista. A continuación, se presentan varios modelos computaciones subyacentes a las teorías arriba descritas, sobre los procesos neurofisiológicos de individuos autistas.

- *Modelo computacional de la Teoría de la Mente Autista: TM* (Galitsky 2016).

Como se describió en el Capítulo 1, este modelo computacional se enfoca en modelar la Mente autista, para definir el mecanismo de razonamiento de los individuos autistas. La Figura 2.7 presenta la estructura del modelo de razonamiento TM. Este modelo computacional de TM representa el razonamiento en términos psicológicos, y elabora su formalización en términos del modelo *creencias-deseos-intenciones* (*Belief-Desire-Intentional* (BDI)). El modelo de *creencias-deseos-intenciones* ha sido desarrollado inicialmente como un modelo para la programación de agentes inteligentes (Haddadi 1996).



**Figura 2.7.** Modelo de razonamiento de la Teoría de la Mente. Fuente. Galitsky (2016)

El modelo computacional facilita a los individuos autistas formas de aprender definiciones más complejas (estados mentales), de tal manera que estas puedan ser usadas en sus comportamientos para hacer preguntas y dar respuestas sobre lo que sabe y desea.

El modelo parte de un estado mental inicial, el cual es analizado utilizando un motor de razonamiento basado en cláusulas de DBI (las cuales son dadas), a partir del cual crea un conjunto de estados mentales consecutivos. Este conjunto de estados mentales consecutivos debe ser consistente con el conjunto de cláusulas DBI dadas, y conllevar a acciones comunicativas, emociones, entre otras (ver Figura 2.7).

- *Modelo computacional de la teoría de coherencia central débil (O’Laughlin, 2000)*

Como se describió en el Capítulo 1, este modelo simula la coherencia *central débil* a través de una red neuronal. Recientes teorías acerca de la naturaleza de las disfuncionalidades en el autismo, sugieren que los individuos autistas muestran una anormalidad en su capacidad para integrar información que les permita dar sentido a su propio entorno, contexto, o percibir “el todo”; en otras palabras, estos individuos tienen una anormal coherencia central. Así, lo que se busca con la red neuronal es la comprensión de la coherencia central débil. En el modelo ocurre esta anormalidad cuando hay una inhibición inusualmente alta en comparación con la excitación. Este

modelo conexionista simula el comportamiento autista, tanto en tareas de falsas creencias como de homografía<sup>2</sup>.

A continuación, se describe el modelo de red neuronal utilizado. La red realiza un proceso que involucra ciclos repetidos de ajuste de activación, como estrategia para calcular la coherencia (ver la Ecuación (2.1))

$$a_j(t+1) = \begin{cases} a_j(t)(1-d) + red_j(max - a_j(t)) & \text{si } red_j > 0 \\ red_j(a_j(t) - min) & \text{de lo contrario} \end{cases}, \quad (2.1)$$

Donde,  $d$  es un parámetro (por decir, 0.05) que va decrementando una unidad en cada ciclo,  $min$  es la activación mínima (-1), y  $max$  es la activación máxima (1). Basado en los pesos  $w_{ij}$  entre cada unidad  $i, j$  es posible calcular  $red_j$ , la entrada a una unidad, por la Ecuación (2.2):

En cada ciclo, la activación de todas las neuronas se ajusta mediante la Ecuación (2.2), y cada

$$red_j = \sum_i w_{ij} a_i(t) \quad (2.2)$$

neurona se actualiza sobre la base de la activación de neuronas que están conectadas por enlaces excitadores e inhibidores, por la Ecuación (2.1).

La anomalía de coherencia central débil aparece en la red neuronal, cuando la red ha sido entrenada para resolver un problema en particular (por ejemplo, falsas creencias, y homografías), y no alcance una solución de coherencia máxima, o cuando queda atrapada en un mínimo local.

- *Modelo computacional del proceso de neurodesarrollo en el autismo* (Carvalho, 1999)

En este modelo se propone una red neuronal para simular el proceso de neurodesarrollo a través de la construcción de mapas neuronales. Un mapa neural es una capa de neuronas cuyas

---

<sup>2</sup> La homografía entre palabras es cuando dos palabras de diferente significado coinciden en su escritura, es decir, tienen identidad ortográfica.

propiedades están sistemáticamente relacionadas con el mundo externo (por ejemplo, espacio visual o auditivo), y representan regiones del sistema nervioso. La simulación del modelo da un indicio de que regiones del cerebro son responsables de las representaciones de alto nivel (i.e., de sus funciones cognitivas), y como estas se ven afectadas en personas autistas. Estas deficiencias pueden dar cuenta de sus síntomas y capacidades especializadas.

Por otra parte, también hay diversos modelos computacionales para el diagnóstico del autismo que buscan apoyar a los psicólogos y terapeutas en sus actividades de diagnóstico (ver sección 1.4.2.1 del Capítulo 1). Además, existen herramientas basadas en Tecnologías de la Información y Comunicación (TIC) para la rehabilitación de disfuncionalidades (Puerto 2017). Por ejemplo, Portero (Portero 2016) elaboró una herramienta computacional de apoyo a las funciones ejecutivas, para mejorar actividades de “planeación”. Tarantino (Tarantino 2016) elaboró una herramienta computacional que incluye el diagnóstico y el aprendizaje gamificado. Galitsky (Galitsky 2013) desarrolló una herramienta computacional de apoyo a disfuncionalidades en el razonamiento. Álvarez (Álvarez 2016) elaboró una herramienta computacional para refuerzo educativo usando la metodología TEACCH (abreviación en inglés de Tratamiento y Educación de niños con Autismo y Problemas de Comunicación relacionados). Junek (Junek 2007), desarrolló una herramienta computacional para el aprendizaje de emociones (expresión facial y vocal). Esas son algunas de las herramientas que existen actualmente.

## **2.2 Avances en el Conocimiento y Modelado Computacional del Cerebro Humano**

A continuación, se describen varios modelos computacionales de Redes Neuronales Artificiales (RNA) avanzadas, y la Teoría del Funcionamiento del Cerebro Humano Basada en el Reconocimiento de Patrones (PRTM).

### 2.2.1 Modelo de Red Neuronal de Pulsos

Conforme se van generando más conocimiento sobre el funcionamiento del cerebro, nuevos modelos de Redes Neuronales (RNAs) surgen. Uno de estos modelos son las RNAs de Pulsos (SNN, del inglés, *Spiking neural network*). Estas redes se inspiran a partir de que las células del sistema nervioso generan una serie de potenciales de acción que se pueden analizar como señales en el dominio del tiempo (Maass 1997; Ghosh 2009).

Las Ecuaciones (2.3) y (2.4) modelizan el comportamiento de una neurona de pulso.

$$u_i = \sum_{t_i^{(f)} \in F_i} n_i(t - t_i^{(f)}) + h(t) \quad (2.3)$$

$$F_i = \{t | u_i(t) = \varphi \wedge u_i'(t) > 0\} \quad (2.4)$$

La Ecuación (2.4) expresa la excitación actual de la neurona de pulso, donde  $t_i^{(f)}$  caracteriza los potenciales de acción (tasa de disparo); el subíndice  $i$  indica la neurona, el superíndice (f) el número de pulso, y  $F_i$  es definido como  $F_i = \{t^1, \dots, t^n\}$ , y describe el tren de estos pulsos basado en el efecto de una influencia externa sobre el potencial de la membrana y el retardo de transmisión sináptica (ver Ecuación (2.4)),  $n_i(\cdot)$  representa el estado refractario<sup>3</sup>, y  $h(t)$  son los efectos del potencial postsináptico. La Ecuación (2.3) considera el tiempo (de activación de la neurona) en el tren del pulso  $F_i$ , en ella  $u_i$  es usado para referirse al potencial de la membrana, o el estado interno de la neurona  $i$  y  $\varphi$  es el umbral. Como se mencionó, el modelo considera los efectos del potencial postsináptico, y se modeliza a través de la Ecuación (2.5):

---

<sup>3</sup> El periodo refractario es aquel instante en que la neurona (membrana) no puede experimentar otro potencial de acción.

$$h(t) = h^{\text{ext}}(t) + \sum_{j \in \Gamma_i} \sum_{t_i^{(f)} \in F_i} w_{ij} \varepsilon_{ij}(t - t_i^{(f)}) \quad (2.5)$$

En la Ecuación (2.5),  $h^{\text{ext}}(t)$  describe el efecto de una influencia externa,  $\varepsilon_{ij}(\cdot)$  es el retardo de transmisión, y  $\Gamma_i$  es el conjunto de sinapsis entre las neuronas de pulsos.

Para este modelo existen varios algoritmos de aprendizaje (Sjöström et al 2010; Caporale et al 2008; Swiercz 2006). En (Christophe 2015) se entrena una SNN para el reconocimiento de patrones (imágenes de círculos y cruz cruzada), basado en el modelo *Spike Timing Dependent Plasticity* (STDP). STDP corresponde a la forma en que las conexiones entre las neuronas cambian de acuerdo con la actividad de pulsos (*spiking*) en la red (Vestbøstad y Ketil 2018). Otros trabajos han entrenado SNN para la detección de caracteres y líneas (Wu et al 2009). Este modelo es importante, porque muestra un nuevo paradigma de neurona artificial basada en el proceso de “integración y disparo”, diferente al modelo clásico normalmente basado en la conductancia<sup>4</sup>.

### 2.2.2 Otros Modelos Neuronales para el Reconocimiento de Patrones

En general, existen muchos algoritmos para la resolución del problema de reconocimiento de patrones desde hace tiempo (Aguilar 2009; Aguilar y Hernández 2009; Aguilar y Hernández 2015). Algunos de ellos, basados en modelos computaciones del cerebro, tal como la Teoría de

---

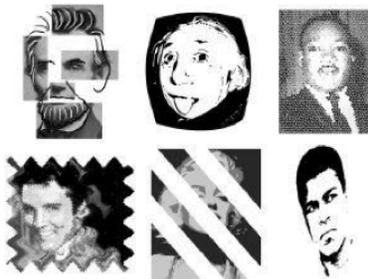
<sup>4</sup> Los modelos basados en conductancia imitan la respuesta de las neuronas biológicas modelando la dinámica de varios canales iónicos, como sodio, potasio, calcio, en la generación del potencial de acción. Ejemplo de esos modelos son el presentado por Hodgkin-Huxley (Hodgkin y Huxley, 1952) para el axón gigante el calamar, y el desarrollado por Wang y Buzsáki (Wang y Buzsáki, 1996; Wang, 2010) para explicar las propiedades dinámicas de las interneuronas de disparo rápido (*fast-spiking*) del hipocampo y la neocorteza.

Resonancia Adaptativa (Altamiranda et al 2015), o el modelo neuronal aleatorio (Aguilar 2004; Aguilar 2005).

Actualmente, el aprendizaje profundo, o *Deep Learning*, ha revolucionado y perfeccionado el reconocimiento de patrones en diversos dominios (Faust et al 2018; Deng 2014; Schmidhuber 2015). Las arquitecturas bajo aprendizaje profundo generan modelos composicionales, donde el objeto (a analizar y reconocer) es expresado como una composición de niveles de abstracciones primitivas (Bengio y Yoshua 2009), tal como se ha propuesto en la Teoría de la Mente Basado en el Reconocimiento de Patrones, presentada a continuación. Por otro lado, las principales arquitecturas de interés de redes neuronales profundas son presentadas en la siguiente sección 2.3.

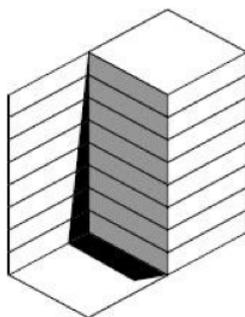
### **2.2.3 Teoría del Funcionamiento del Cerebro Humano para el Reconocimiento de Patrones PRTM**

En esta tesis, se escogió la Teoría de la Mente para el Reconocimiento de Patrones PRTM (del inglés, *Pattern Recognition Theory of Mind*), por cuanto describe aspectos claves del funcionamiento de la mente humana bajo un enfoque sistemático, plausible de ser representado en un algoritmo (Kurzweil 2013; Du Castel 2015). Por ejemplo, PRTM sugiere que nuestras memorias son secuenciales y ordenadas, y pueden ser recuperadas en el orden en que fueron aprendidas. También, sugiere que no hay imágenes, ni videos, ni sonidos, almacenados en nuestro cerebro, que las memorias son almacenadas esparcidamente como secuencias de patrones, y los patrones que no son refrescados desaparecen con el tiempo. La habilidad de reconocimiento es capaz de detectar propiedades invariantes de un patrón (características que se mantiene a pesar de las perturbaciones del mundo-real), i.e., puede reconocer un patrón, aún si sólo parte de él es percibido y/o tiene alteraciones. Este aspecto se puede apreciar en la Figura 2.8.



**Figura 2.8.** Caras de famosos distorsionadas. Fuente. Kurzweil, R. (2013)

Otro importante aspecto que sugiere es que la experiencia usada en las percepciones cambia nuestras interpretaciones. Esto se observa en la Figura 2.9, donde se puede ver una cuña oscura o una pared clara proyectando una sombra.



**Figura 2.9.** Imagen ambigua. Fuente. Kurzweil, R. (2013)

Por otra parte, considérese el siguiente experimento: “Vemos lo que queremos \_\_ \_\_ \_\_”, así, sugiere que el completar la frase dice que el cerebro constantemente está prediciendo el futuro y realizando hipótesis. Un último aspecto que sugiere con respecto al manejo de patrones del cerebro, es que nuestros procedimientos, tales como las actividades diarias (como lavarse los dientes), no son almacenados como una lista de pasos, más que eso, estos procedimientos son recordados como una jerarquía de actividades anidadas.

Con base en estos hechos, y a la luz del principio de la extraordinaria uniformidad fundamental de la estructura del neocortex (Mountcastle 1997), PRTM sostiene que tal uniformidad viene dada por módulos reconocedores de patrones, los cuales constituye el componente fundamental del

neocortex. Estos reconocedores crean conexiones entre sí en el tiempo, que se forman para reflejar los patrones que se aprenden con las vivencias, experiencias, qualías, etc., las cuales no vienen especificadas en el código genético.

Como tal, el neocortex es responsable de la percepción sensorial, del reconocimiento de objetos y conceptos abstractos, de controlar el movimiento y la orientación espacial, y del pensamiento racional y lenguaje.

### **2.2.3.1 Una Jerarquía de Patrones**

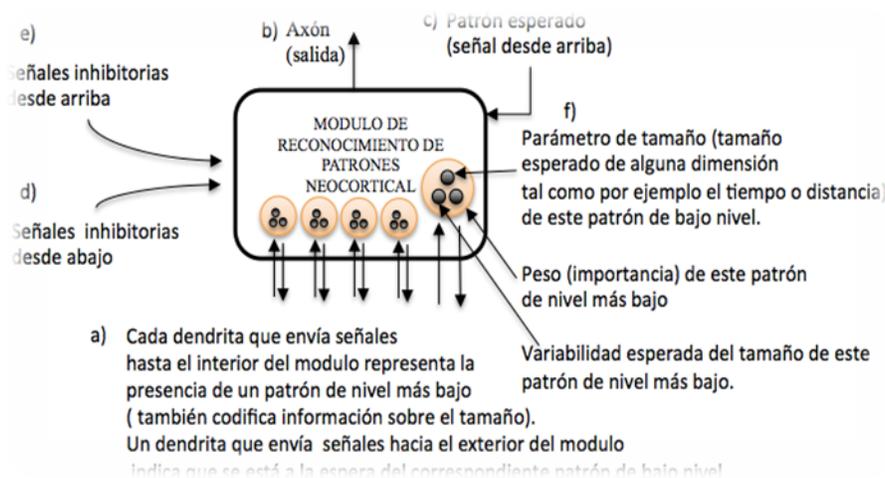
La Teoría PRTM está basada en el reconocimiento de patrones a través de módulos de reconocimiento del neocórtex. Estos patrones (y los módulos) están organizados por jerarquías (Kurzweil 2013; George 2004; Hawkins 2007).

*Definición de Patrón.* Cada patrón (el cual es reconocido por uno de los 300 millones de reconocedores de patrones en el neocórtex (Kurzweil 2013) está compuesto de tres partes:

- La parte de entrada (Input): la cual consiste en los patrones de nivel inferior que lo componen. Por ejemplo, muchos de los patrones de palabras como “CASA” o “COMA”, incluirán la letra A. Para cada uno de estos patrones, no se necesita repetir la descripción de la letra “A”, sino que se usa la misma descripción.
- La segunda parte es su nombre. En el ámbito del lenguaje, este patrón de alto nivel es simplemente la palabra “CASA” o “COMA”. En el neocórtex, el "nombre" de un patrón es el axón que emerge de cada procesador del patrón; cuando el axón se dispara, su patrón correspondiente ha sido reconocido.
- La tercera y última parte para cada patrón, es el conjunto de patrones de más alto nivel del que a su vez él forma parte. Para la letra “A”, esto significa todas las palabras que incluyen la “A”, como “CASA”, “COMA”, etc. En el neocórtex, estas conexiones están representados por dendritas físicas que fluyen entre las neuronas en cada reconocedor de

patrón cortical. Cada neurona puede recibir entradas de múltiples dendritas, y produce una única salida a través de su axón. Ese axón, sin embargo, puede a su vez transmitir a múltiples dendritas.

*Definición de Módulo de Reconocimiento de Patrón.* Un módulo de reconocimiento (constituido aproximadamente por un grupo de 100 neuronas) tiene como función reconocer un patrón. Este módulo es representado esquemáticamente por la Figura 2.10, como sigue:



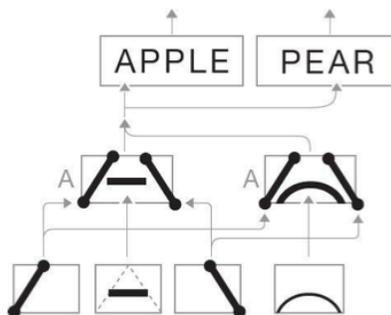
**Figura 2.10.** Diagrama del Módulo de reconocimiento de patrones neocortical. Fuente. Adaptado de Kurzweil, R. (2013).

Donde, *a)* son las dendritas que entran al módulo que representa al patrón. Las dendritas bien pueden o enviar información hacia el interior del módulo, indicando la presencia de un patrón en el nivel inferior, o hacia el exterior, indicando que se está a la espera del reconocimiento del respectivo patrón. Cada una de las dendritas está conectada a uno o más axones de reconocedores de patrones de nivel inferior que forman parte de este patrón. Para cada uno de estos patrones de entrada, puede haber muchos reconocedores de patrones de bajo nivel, que dispararían la señal que el patrón de bajo nivel ha sido reconocido. El umbral necesario para reconocer el patrón puede ser alcanzado, aún si no todas las señales de entrada están presentes. El módulo calcula la probabilidad

de que el patrón del cual es responsables este presente. Este cálculo considera los parámetros de “tamaño” e “importancia”, representados por cada punto negro en el módulo, señalados por *f*). *b*) cuando este reconocedor de patrón reconoce este patrón (basado en todas o algunas de las señales de las dendritas de entrada que se han activado), entonces el axón (que representa la salida) será disparado. A su vez, este axón puede conectarse a una red de dendritas que se conectan a muchos reconocedores de patrones de nivel superior, de los que él forma parte. *c*) Si un reconocedor de patrones de nivel superior está recibiendo una señal positiva de todos, o la mayoría de sus patrones constituyentes, excepto de este reconocedor de patrones, entonces ese reconocedor de nivel superior podría enviar una señal a este reconocedor, indicando que se espera su patrón. Una señal de este tipo haría que el reconocedor de patrones redujera su umbral, lo que significa que sería más probable que lo reconozca. *d*) representa las señales inhibitorias procedentes de bajo nivel, que hacen que sea menos probable que el reconocedor del patrón reconozca este patrón. Esto puede deberse al reconocimiento de patrones de nivel inferior, que son inconsistentes con el patrón asociado con este reconocedor de patrones (por ejemplo, el reconocimiento de unos “pezuñas” por un reconocedor de nivel inferior, haría menos probable que esta imagen sea de un “león”). *e*) representa las señales inhibitoras de alto nivel que hacen que sea menos probable que el reconocedor del patrón reconozca este patrón. Esto puede resultar de un contexto de nivel superior que es inconsistente con el patrón asociado con este reconocedor. Finalmente, *f*) para cada entrada, hay parámetros almacenados que se refieren a su importancia, tamaño esperado y variabilidad esperada del tamaño (Puerto 2014).

La Figura 2.11 ilustra el problema de reconocimiento de patrones de palabras que se realiza de manera jerárquica o taxonómica, a través de módulos de reconocimiento de distintos niveles de complejidad. El primer nivel de la jerarquía tiene el conjunto de módulos de reconocimiento de

los patrones de palabras (APPLE, PEAR), el segundo nivel tiene los reconocedores de los patrones de letras mayúsculas (A) que conforman las palabras, y finalmente, el tercer nivel o atómico contiene los módulos de reconocimiento de los patrones de líneas que constituyen las letras.



**Figura 2.11.** Jerarquía de módulos de reconocimiento para palabras, letras y líneas

. Fuente. Kurzweil, R. (2013)

Como es de notar, la salida del axón de un módulo reconocedor del patrón "A", se conecta a múltiples dendritas para cada palabra que usa el patrón "A". Ahora, si se tiene en cuenta el factor de redundancia, i.e., si hay más de un reconocedor de patrones para la letra "A" (letras "A" algo diferentes), entonces cualquiera de estos múltiples reconocedores del patrón "A" puede enviar una señal de reconocimiento hasta los reconocedores de patrones de más alto nivel que incorporan a "A".

En conclusión, un importante atributo de PRTM de interés, es cómo se hacen los reconocimientos dentro de cada módulo de reconocimiento de patrones. Almacenado en el módulo hay un peso para cada dendrita de entrada, que indica qué tan importante es esa entrada para el reconocimiento. El reconocedor de patrones tiene un umbral para la activación (lo que indica que este reconocedor de patrones ha reconocido con éxito el patrón del que es responsable). No todos los patrones de entrada tienen que estar presentes para que un reconocedor se dispare. El reconocedor aún puede dispararse, si falta una entrada con un peso bajo, pero es menos probable que se dispare si falta

una entrada de gran importancia. Cuando se dispara, un reconocedor de patrones básicamente difunde la presencia de este patrón a otros módulos de nivel superior que lo contiene.

## **2.3 Redes Neuronales Artificiales Recientes**

En esta sección, se presentan modelos computacionales de RNA que están emergiendo, basados en los conceptos de Aprendizaje Profundo (del inglés, *Deep Learning*), y Aprendizaje de Máquina Extremo.

### **2.3.1 Aprendizaje Profundo (o Deep Learning)**

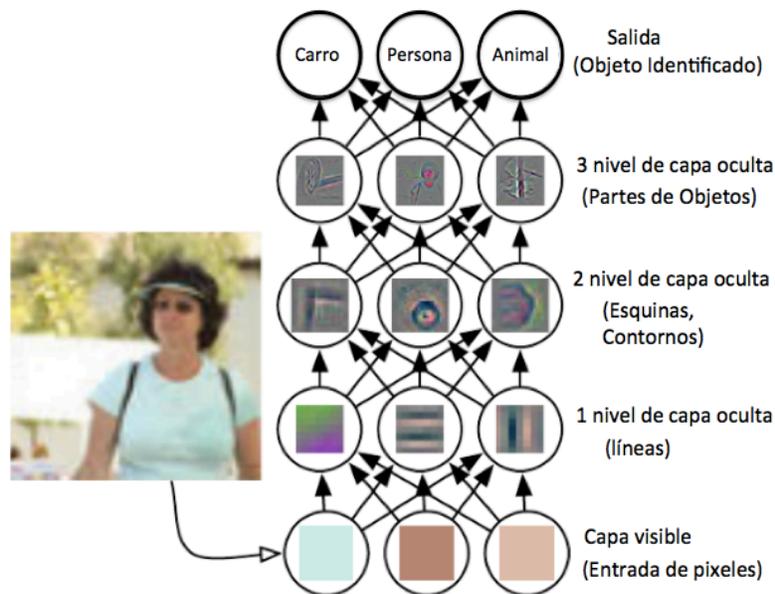
Deep Learning es una técnica de *Machine Learning*<sup>5</sup> (Aprendizaje Automático, en español) basada en redes neuronales, que trata de simular la forma que tiene de aprender el cerebro (Schmidhuber 2015). Deep Learning permite generar una representación de los datos a través de varias capas, y es relevante para la clasificación de patrones (Widrow et al 1990). En (Hornik et al 1989) se plantea que cuanto más complejo es el patrón a reconocer, más capas ocultas de neuronas son necesarias. Extrapolando esta idea, acorde con (Hinton y Geoffrey 2007), se definen mecanismos jerárquicos con varios niveles de profundidad de capas ocultas de neuronas, que permiten reconocer patrones más complejos.

Lo nuevo de este paradigma, según Hinton (Hinton et al 2006), es que plantea un aprendizaje semi-supervisado o híbrido. Es decir, la diferencia de Deep Learning con respecto a otras técnicas de Machine Learning es que los descriptores que antes se definían a mano (el diseñador los definía), ahora se aprenden de manera automática.

---

<sup>5</sup> El aprendizaje automático o aprendizaje de máquinas (del inglés, "Machine Learning") es el subcampo de las ciencias de la computación, y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

El aprendizaje en Deep Learning ocurre como un proceso de desagregación de los patrones, que permite ir descubriendo sus descriptores, y agregarlos posteriormente en las capas individuales ocultas de neuronas. Por ejemplo, la primera capa aprende características primitivas, como un borde en una imagen o la unidad más pequeña de sonido del habla. El proceso se repite en capas sucesivas, hasta que el sistema puede reconocer con seguridad cosas más complejas como fonemas, caras u otros objetos (ver la Figura 2.12).



**Figura 2.12.** Modelo base de las RNA basadas en el aprendizaje profundo. Fuente. Goodfellow (2016)

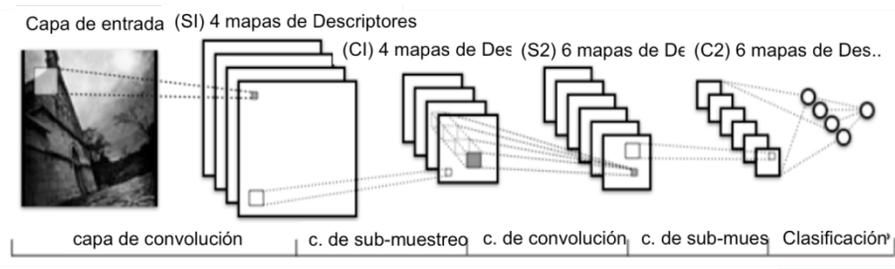
La Figura 2.12 muestra como Deep Learning va representando en cada capa de la jerarquía, abstracciones cada vez más complejas a partir de los datos. Como se dijo antes, existen varias RNA basadas en Deep Learning (Bengio y Yoshua 2009). A continuación, se presentan dos de las técnicas más comunes, las CNN y las redes de DBN.

Algunos ejemplos de estas nuevas RNA son las redes neuronales de convolución (CNN, del inglés, *Convolutional Neural Network*), o las redes de creencias profundas (DBN, del inglés, *Deep Belief Networks*), entre otras.

### 2.3.1.1 Redes Neuronales de Convolución (del inglés, Convolutional Neural Network CNN)

Las CNN (o ConvNets) son un tipo de red neuronal de la familia de Deep Learning, específicamente diseñadas para aprender y descubrir características/descriptores (features) de imágenes (LeCun et al 1995).

A nivel de su arquitectura, las CNNs son estructuras compuestas por dos etapas fundamentales; una etapa de convolución (*convolution layer*) y otra de sub-muestreo (*sub-sampling layer*), aprendiendo en cada capa descriptores de diferentes niveles de abstracción. La entrada y salida de cada una de estas etapas son conjuntos de arreglos, llamados “*features maps*”, o mapas de características. Una CNN típica puede ser construida a partir de una o varias de estas etapas, seguida de un módulo de clasificación. La Figura 2.13 muestra una representación gráfica de este modelo.



**Figura 2.13.** Representación gráfica de un modelo de CNN para reconocimiento de una imagen. Sharif et al. (2014)

En la Figura 2.13, cada plano es un *feature map*, es decir, un conjunto de unidades cuyos pesos son idénticos (este concepto se explica más adelante). Los planos (S1) y (S2) corresponden a los *feature map* de la capa de sub-muestreo, y los planos (C1) y (C2) corresponden a los *feature map* de la capa de convolución.

Las capas inferiores (línea azul) están compuestas por una capa de convolución y una capa de sub-muestreo, y corresponden al proceso de aprendizaje no supervisado. Las capas superiores

(línea roja) corresponden a la clasificación supervisada. Esta capa de alto nivel se encuentra totalmente conectada, y es manejada por un modelo tradicional MLP/Multi-Layer Perceptron.

En el proceso de aprendizaje para reconocimiento, cada pixel de la imagen en la “*input layer*” es asociado con una neurona de entrada. En la primera etapa de convolución se realiza un proceso de zonificación, que consiste en tomar regiones o campos locales de recepción (*local receptive fields: LRF*). Por cada *LRF* hay una neurona oculta responsable del mismo. Estas neuronas (ocultas) actúan como filtros locales basados en convolución<sup>6</sup>. La convolución en este contexto, es una suma de cada uno de los valores de los pixeles ponderado por el peso de las entradas a la neurona. Por ejemplo, si la zona o *LRF* es de 4x4 pixeles, la salida para cada *j, k*-ésima neurona oculta, viene dada por la Ecuación (2.6):

$$\sigma \left( b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l,k+m} \right) \quad (2.6)$$

Donde *b* es el bias<sup>7</sup> y  $w_{l,m}$  son los pesos correspondientes a las neuronas de entrada  $a_{j+l,k+m}$  dentro de la región bidimensional. Este filtro se repite localmente, y pueden definirse *K* filtros para la misma zona. En el ejemplo de la Figura 2.13, se han definido cuatro filtros para la primera capa, por tanto, se tiene cuatro (SI) 4 *features maps*. Todas la neuronas de la primera capa oculta detectan lo mismo, solo que en diferentes lugares de la imagen de entrada. En esta primera etapa de convolución ocurre el primer aprendizaje no supervisado de descriptores, consolidados en los *feature maps*.

---

<sup>6</sup> Una convolución en matemáticas es un operador matemático que transforma dos funciones *f* y *g* en una tercera función, que en cierto sentido representa la magnitud en la que se superponen *f* y una versión invertida de *g*.

<sup>7</sup> bias es un peso adicional que se contempla para cuando no hay una clasificación clara, es como un hiperplano alternativo de clasificación.

Posterior a la primera etapa de convolución, se sigue la etapa de submuestro. En esta etapa se realiza una síntesis para la siguiente fase de construcción de descriptores. Una técnica común usada para la síntesis es el max-pooling o agrupamiento. Esta técnica consiste en definir una especie de rejilla, con celdas de tamaño constante, a partir de las cuales se saca un sólo valor. La salida de esta etapa consiste en un nuevo conjunto de features maps, con lo cual se alcanza un nuevo aprendizaje, en este caso, de descriptores más abstractos.

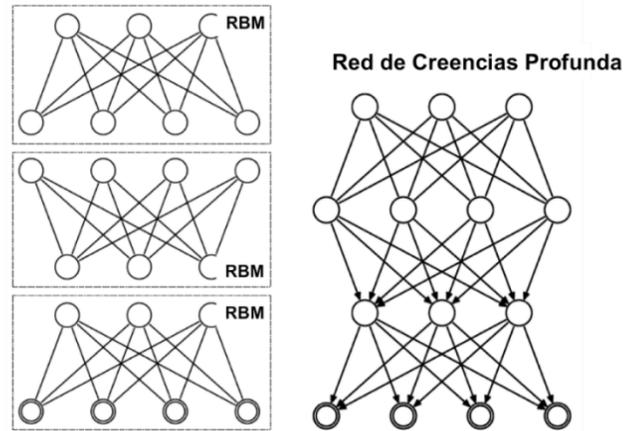
Estos dos procesos descritos conforman lo que sería un bloque básico de procesamiento a profundidad: de convolución y sub-muestreo. Es en este proceso dual donde acontece el aprendizaje automático a nivel de Deep Learning. El proceso se repite en todas las capas inferiores (línea azul) hasta alcanzar la capa superior (línea roja), donde se realiza la clasificación por métodos convencionales supervisados.

Las CNNs son importantes porque permiten aprender descriptores de manera no supervisada con niveles de complejidad alto, e incorpora en su etapa final la clasificación, todo bajo el mismo paradigma. Este paradigma es útil, por cuanto proporciona un mecanismo de aprendizaje de características jerarquizado basado en convolución, muy similar al modelo biológico de las neuronas del córtex visual (Hubel et al 1959; Hubel et al 1962).

### ***2.3.1.2 Redes de Creencias Profundas (del inglés Deep Belief Networks: DBN).***

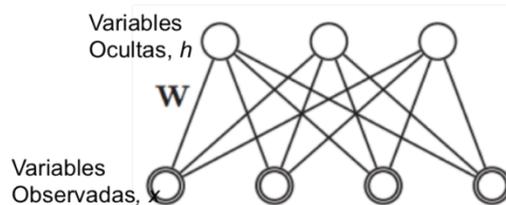
Hinton (Hinton et al 2006) introduce en el 2006 el concepto de redes de creencias profundas (DBNs, del inglés Deep Belief Networks).

Las DBNs son modelos generativos probabilísticos que contienen muchas capas ocultas, cada capa va captando correlaciones de orden superior entre las actividades de las capas inferiores. Las dos últimas capas de la DBN forman un grafo bipartito no dirigido con las capas inferiores, como se muestra en la Figura 2.14.



**Figura 2.14.** Esquema de aprendizaje con Máquinas de Boltzmann Restringidas (RBM, en Ingles) . Fuente. Fischer et al. (2012).

Así, una DBN está conformada por Redes de Creencias (BN, del inglés *Belief Networks*) y Maquinas Restringidas de Boltzmann (RBM, del inglés *Restricted Boltzmann Machines*) (Fischer et al 2012). Las RBM, de nivel inferior, se utilizan como datos de entrada para entrenar la siguiente RBM (lado izquierdo). La correspondiente red de creencias profundas se observa en el lado derecho. Las RBM se caracterizan porque tienen una capa oculta de unidades/variables ( $h$ ), con la restricción de no haber conexiones entre ellas (ni entre las unidades/variables visibles,  $x$ ), tal como se muestra en la Figura 2.15



**Figura 2.15.** La estructura de una Máquina de Boltzmann Restringida. Fuente. Fischer, et al. (2012)

Por otra parte, una red de creencias (BN, del inglés *Belief network*) es un método de razonamiento causal, basado en la teoría de probabilidad bayesiana (Hinton y Geoffrey 2009). Una BN es un

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{Pad}(X_i)) \quad (2.7)$$

grafo acíclico dirigido, el cual codifica las relaciones causales entre variables particulares, representadas en los nodos del grafo. Los nodos están conectados por relaciones o link de causalidad (arcos), que van desde los nodos principales/padres (las causas) a los nodos secundarios/hijos (efectos), los nodos hijos pueden también ser nodos padres. La probabilidad condicionada de los hijos con respecto a los padres, viene dada por la fórmula de la Ecuación (2.7). Donde, la función  $Pad(X_i)$ , representa el conjunto de nodos con enlaces causales dirigidos hacia  $X_i$ . Esta fórmula es la base o aporte de las BN a las DBNs (ver Ecuación.(2.8)). La fusión de estos dos elementos (a saber, máquinas de Boltzmann (RBM) y redes de creencias (BN)), fue realizada por Hilton (Hinton et al 2006), quien mostro que las RBMs podrían se apiladas y entrenadas

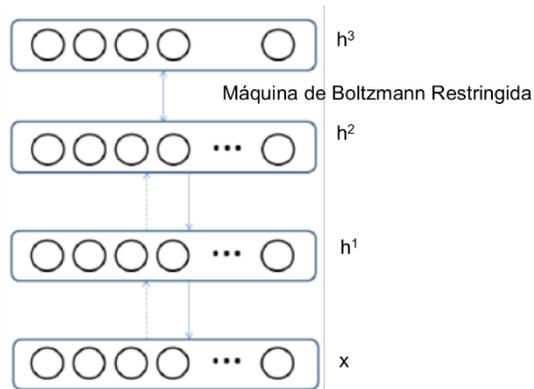
$$P(x, h^1, h^2, \dots, h^l) = \left( \prod_{k=0}^{l-2} P(h^k | h^{k+1}) \right) \cdot P(h^{l-1}, h^l) \quad (2.8)$$

(usando el paradigma de aprendizaje “*Greedy algorithm*”<sup>8</sup>), para formar las mencionadas redes de creencias profundas o DBNs. Las DBNs aprenden a extraer descriptores a partir de datos de entrenamiento. A diferencia de las CNN, que basan su técnica de aprendizaje en convolución y sub-muestreo, las DBNs basan su técnica de aprendizaje en probabilidades, específicamente, en la distribución de probabilidad conjunta entre el vector observado  $x$  y las  $l$  capas ocultas, como sigue (ver Ecuación (2.8)):

Donde,  $x = h^0$ ,  $P(h^{k-1} | h^k)$  es una distribución condicional para las unidades visibles condicionadas/limitadas sobre las unidades ocultas pertenecientes a la RBM en el nivel  $k$ , y  $P(h^{l-1}, h^l)$  es la distribución conjunta oculta-visible en la RMB del nivel superior o de salida. Esto es ilustrado en la Figura 2.16.

---

<sup>8</sup> El algoritmo voraz o *Greedy algorithm*, es aquel que, para resolver un determinado problema, sigue una heurística consistente en elegir la opción óptima en cada paso local, con la esperanza de llegar a una solución general óptima.



**Figura 2.16.** Red de creencias profundas formadas por RBMs

De esta manera, una DBN puede aprender probabilísticamente a reconstruir estas entradas. Entonces, las capas actúan como detectores de características en dicha entrada.

Después de este paso de aprendizaje no supervisado, antes de las dos últimas capas superiores de la jerarquía, una DBN puede de nuevo ser entrenada de forma supervisada para mejorar la etapa de clasificación en esas dos últimas capas. El algoritmo entrenamiento de una DBN procede de la siguiente manera: Sea  $X$  la matriz de entradas, considerada como un conjunto de vectores de característica.

- Entrenar un RBM con  $X$  para obtener su matriz de peso,  $W$ , considerada esta como la matriz de peso entre las dos capas más profundas de la red.
- Transformar  $X$  por el RBM para producir una nueva  $X'$ .
- Repetir este procedimiento con  $X \leftarrow X'$ , para el par próximo de capas, hasta las dos capas superiores.

La importancia y/o interés por las DBNs es su aporte como nuevo paradigma de aprendizaje profundo, usando una técnica de descubrimiento de características de manera probabilística no supervisada.

### 2.3.2 Aprendizaje de Máquina Extremo (del inglés, Extreme Learning Machine: ELM)

Huang (Huang et al 2006) introduce un nuevo paradigma de aprendizaje alternativo, denominado máquinas de aprendizaje extremo. Básicamente, este algoritmo se propuso para entrenar una red de propagación hacia delante con una capa oculta o tipo SLFN (del inglés, *single hidden-layer feedforward network*) (Liang et al 2006). Más específicamente, el ELM fue propuesto para “generalizar” una SLFN, donde los nodos ocultos de la capa oculta no necesariamente deben ser neuronas (i.e., no requieren estar basadas en una suma algebraica) (Huang et al 2008; Huang et al 2011). La función de salida de ELM para la red SLFM generalizada, la define la Ecuación (2.9):

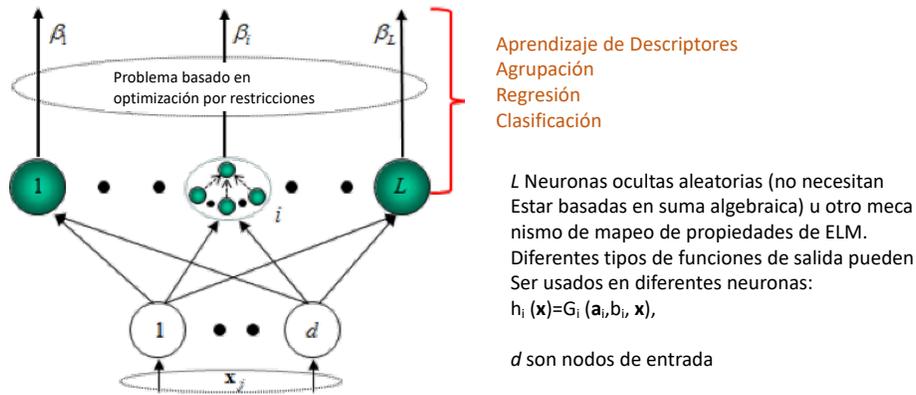
$$f_L(x) = \sum_{i=1}^L \beta_i h_i(x) = h(x)\beta \quad (2.9)$$

Donde,  $\beta = [\beta_1, \dots, \beta_n]^T$  es el vector de pesos de salida entre la capa oculta de  $L$  nodos y los  $m \geq 1$  nodos de salida,  $h(x) = [h_1(x), \dots, h_L(x)]$  es el mapeo de descriptores/propiedades no lineal de ELM (ver Figura 2.17, por ejemplo, el vector de salida (fila) de la capa oculta con respecto a la entrada  $x$ ), y  $h_i(x)$  es la  $i$ -ésima salida del nodo oculto.

Las funciones de salida de los nodos ocultos, no necesariamente deben ser únicas. Diferentes funciones de salida pueden ser usadas por diferentes neuronas ocultas. En general, la salida  $h_i(x)$  puede ser definida por la Ecuación (2.10).

$$h_i(x) = G(a_i, b_i, x) \quad (2.10)$$

Donde  $G(a, b, x)$  es una función continua por partes no lineal que satisface la propiedad o capacidad de aproximación general de ELM, y  $a, b$  son parámetros del nodo oculto (Huang et al 2008).



**Figura 2.17.** Arquitectura de ELM. Fuente. Huang, et al. (2008)

La Tabla 2.1 lista algunas de estas funciones:

**Tabla 2.1.** Funciones de mapeo usadas en ELM

Función Sigmoidea	$G(a, b, x) = \frac{1}{1 + \exp(-(a \cdot x) + b)}$
Función tangente hiperbólica	$G(a, b, x) = \frac{1 - \exp(-(a \cdot x) + b)}{1 + \exp(-(a \cdot x) + b)}$
Función Gaussiana	$G(a, b, x) = \exp(-b\ x - a\ )$
Función multicuadrática	$G(a, b, x) = (\ x - a\  + b^2)^{1/2}$
Función coseno/Bases de Fourier	$G(a, b, x) = \cos(a \cdot x + b)$
Función limite	$G(a, b, x) = \begin{cases} 1, & \text{si } a \cdot x + b \leq 0 \\ 0, & \text{en otro caso} \end{cases}$

Básicamente, ELM entrena a una SLFN a través de dos etapas principales: (1) mapeo de propiedades aleatoria y (2) resolución de parámetros lineales (Huang et al 2006). En la primera etapa, ELM inicializa de manera aleatoria la capa oculta para mapear la información de entrada en un espacio de propiedades (llamado el espacio de propiedades de ELM), a través de alguna función de mapeo no lineal (Huang et al 2011; Matias et al 2014).

En la segunda etapa del aprendizaje en ELM, las conexiones de los pesos entre la capa oculta y la capa de salida denotados por  $\beta$  se minimizan, usando la aproximación del error por mínimos cuadrados de la Ecuación (2.11).

$$\min_{\beta \in R^{L \times m}} \|H\beta - T\|^2 \quad (2.11)$$

Donde,  $T$  es la data o información de entrenamiento de la matriz objetivo;  $\|\cdot\|$  es la norma euclidiana o matriz norma<sup>9</sup>;  $H$  es la matriz de salida de la capa oculta (matriz aleatoria), definida en la Ecuación (2.12).

$$H = \begin{bmatrix} h(x_1) \\ \dots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \dots & h_L(x_1) \\ \dots & \dots & \dots \\ h_1(x_N) & \dots & h_L(x_N) \end{bmatrix} \quad (2.12)$$

A modo de conclusión, el algoritmo ELM ha sido propuesto para entrenar redes neuronales tipo feedforward con una capa oculta SLFNs. La conexión entre la capa de entrada y las neuronas ocultas se realiza de manera aleatoria, y no cambian durante el proceso de aprendizaje (i.e., no tienen que ser iterativamente afinadas). Los únicos parámetros que necesitan ser aprendidos son las conexiones (pesos) entre la capa oculta y la capa de salida. Estos pesos son afinados vía la minimización de la función de costo, por lo que la complejidad y/o costo computacional de ELM se reduce significativamente, ya que el único costo es resolver un sistema lineal. Con base en (Huang et al 2008), ELM comparado con otras técnicas tradicionales de inteligencia computacional, proporciona un mejor desempeño de generalización con un aprendizaje más rápido (aprendizaje en tiempo real, en segundos, milisegundos, incluso en microsegundos), alta precisión, con menor intervención del ingeniero de conocimiento (Huang et al 2011).

---

<sup>9</sup> La matriz normal de una matriz  $A$  de orden  $n \times m$ , es definida como la raíz cuadrada de la suma de los cuadrados absolutos de sus elementos.

### 3. CAPITULO 3: ESPECIFICACIÓN COMPUTACIONAL DE PROCESOS NEUROFISIOLÓGICOS DE INDIVIDUOS CON CARACTERÍSTICAS COGNITIVAS DIFERENTES EN TAREAS DE RECONOCIMIENTO DE PATRONES

#### 3.1 Un Modelo Computacional del Proceso Neurofisiológico de Reconocimiento de Patrones

En esta sección, se describe el modelo computacional denominado *Algoritmo Recursivo de Reconocimiento de Patrones* (en adelante *Ar2p*, por sus siglas en español), el cual está inspirado en el proceso de reconocimiento de patrones subyacente en los procesos neurofisiológicos de individuos con características cognitivas diferentes, y en las teorías sobre el funcionamiento sistemático del cerebro humano, más específicamente del neocórtex (Mountcastle 1997; Hawkins y Blakeslee 2007; Kurzweil 2013).

*Ar2p* puede reconocer patrones, tanto estáticos como dinámicos, simples y complejos. En particular, estos últimos (dinámicos/complejos) no han sido estudiados ampliamente en el área de reconocimiento de patrones (Fred y De Marisco 2015; Watanabe 2015). Para la construcción del modelo computacional *Ar2p*, se han considerado los siguientes aspectos: i) se han introducido los formalismos lógico-matemáticos de reconocimiento y aprendizaje de patrones necesarios para *Ar2p*, ii) se han especificado los algoritmos de reconocimiento y aprendizaje de patrones de *Ar2p*, iii) se han definido las estrategias usadas para el reconocimiento de patrones estáticos y dinámicos y, iv) se ha validado en varios escenarios de prueba.

*Ar2p* se caracteriza por incorporar un enfoque *Top-Down* para el reconocimiento de patrones de entrada complejos, y *Bottom-Up* para el reconocimiento de patrones de entrada simples. La especificación del modelo computacional explota la idea de recursividad, basada en la jerarquía neocortical y en la desagregación/integración del patrón en el proceso de reconocimiento (Puerto

y Aguilar 2016a, Puerto y Aguilar 2017).

Existen varios modelos computacionales para la resolución del problema de reconocimiento de patrones desde hace tiempo (Aguilar 2004; Aguilar y Hernández 2009; Puerto 2016). Algunos, basados en el cerebro, como la Teoría de Resonancia Adaptativa (Altamiranda, Aguilar y Hernández 2015), o el Neocognitron (Fukushima y Miyake 1982; Fukushima, Miyake y Ito 1983), otros basados en reglas lógicas (Collins y Bobrow 2017; Babic, Nestic y Miljkovic 2008), o basados en aprendizaje de máquina clásicos (Alpaydin 2009; Aguilar y Colmenares 1997), y modelos más recientes basados en Aprendizaje Profundo o *Deep Learning* (Goodfellow, Bengio, Courville, y Bengio 2016).

Los métodos basados en reglas aplican un principio común: las estructuras identificadas (parcialmente) a través de reglas lógicas se mapean según algún patrón en la base de conocimiento, utilizando reglas *if-then*. Y no existen dos reglas para la descripción de un mismo patrón en la base de conocimiento (Babic, Nestic, y Miljkovic 2008). Por otra parte, los métodos de aprendizaje de máquina dan al sistema la habilidad para adquirir su propio conocimiento, a través de la extracción de patrones a partir de los datos. Los métodos clásicos de aprendizaje de máquina requieren que los descriptores (*features*) de los patrones sean diseñados a mano. A diferencia de los modelos clásicos de aprendizaje de máquina, con *Deep Learning* estos descriptores son descubiertos de manera automática (Deng y Yu 2014; Schmidhuber 2015). Algunos modelos basados en *Deep Learning* son las redes neuronales convolucionadas o Convolutional Neural Network (CNN, por sus siglas en inglés) (LeCun y Bengio 1995; Razavian, Azizpour, Sullivan, y Carlsson 2014), las máquinas de Boltzmann profundas o *Deep Boltzmann Machine* (DBM, por sus siglas en inglés) (Srivastava, Salakhutdinov, y Hinton 2013), las redes de creencias profundas o *Deep Belief Networks* (DBN, por sus siglas en inglés) (Lopes y Ribeiro 2015), entre otros modelos (ver sección

2.3 del capítulo 2).

Es de notar que un enfoque parecido al nuestro (*Ar2p*) es la *Teoría de Wavelet* (Tang 2000, Hernández y Weiss 1996), la cual aplica un conjunto de filtros (los cuales pueden ser considerados equivalentes a nuestros módulos de reconocimiento de patrones), que son afinados para reconocer descriptores bien definidos (frecuencias espaciales, temporales, tamaño, etc.) del patrón original. Es una técnica que usa ventanas con regiones de tamaño variable, la cual permite descomponer un patrón (señal o imagen) en versiones escaladas desde el patrón original (Meyer 1993; Graps 1995; Bogges y Narcowich 2001; Christensen, O y Christensen, K 2004; Daubechies 1992). La teoría de Wavelet ha sido empleada en análisis y comprensión de patrones de imágenes (Prasad y Iyengar 1997), donde cada filtro extrae propiedades de mediana frecuencia en comparación al siguiente nivel. Este aspecto puede ser interpretado de alguna manera como equivalente a un proceso recursivo para entender el patrón.

Por otra parte, en algunos contextos de reconocimiento o detección de patrones de rostros, un redimensionamiento de la imagen a un nivel elemental (o re-muestreo) es aplicado paso a paso. Con esta técnica, los tamaños intrínsecos de los patrones cambian, mientras se va aplicando el mismo método de extracción de descriptores. Esto también puede considerarse como un método de extracción de descriptores recursivo.

El resto del capítulo está organizado como sigue: la sección 3.1.1 describe formalmente el modelo de reconocimiento de patrones estáticos, la sección 3.1.2 describe formalmente el modelo de reconocimiento de patrones dinámicos. La sección 3.2 describe el modelo de aprendizaje supervisado de *Ar2p*, la sección 3.3, prueba las capacidades de reconocimiento y aprendizaje de *Ar2p* en diversos dominios: reconocimiento de estilos de conducción usando *Ar2p* (sección 3.3.1); reconocimiento de emociones en robots usando *Ar2p* (sección 3.3.2); y reconocimiento de texto

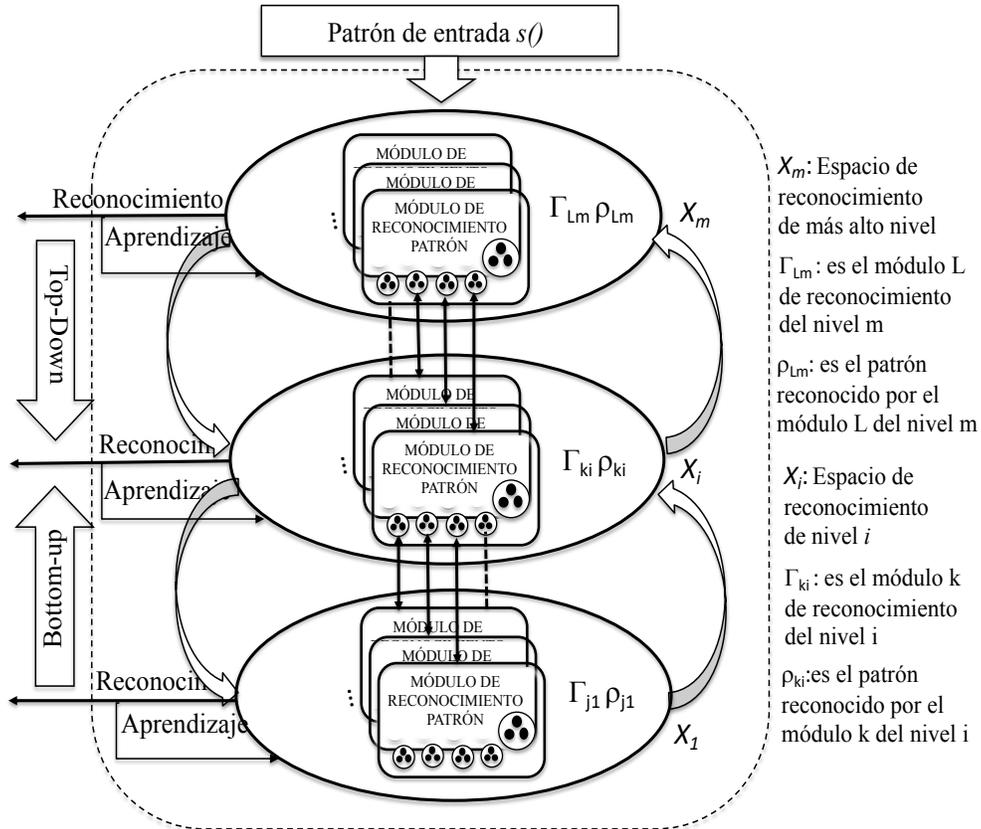
usando  $Ar2p$  (sección 3.3.3). Por último, la sección 3.4 contiene un análisis general de los resultados de  $Ar2p$ .

### 3.1.1 Formalización matemática de $Ar2p$ para el reconocimiento de patrones estáticos

El esquema general del modelo computacional de reconocimiento de patrones  $Ar2p$ , se muestra en la Figura 3.1 (Puerto y Aguilar 2016a, Puerto y Aguilar 2017). El sistema general de reconocimiento recibe como entrada un patrón  $s()$ , que viene del sistema de percepción. Entonces, el sistema toma  $s()$  e inicia su proceso de reconocimiento. Para ello, dispone de varios niveles de reconocimiento, representados como óvalos en el esquema, e identificados como  $X_i$ , desde  $i=1$  hasta  $m$ .  $X_1$  es el nivel de reconocimiento para patrones atómicos, y  $X_m$  es nivel de reconocimiento para patrones más complejos.

Cada nivel está compuesto por módulos de reconocimiento de patrones, identificados como  $\Gamma_{ji}$  (para  $j=1,2,3, \dots, \#$  de módulos en el nivel  $i$ ), que conforman el espacio de reconocimiento del nivel  $X_i$ . Cada módulo tiene la función de reconocer su patrón respectivo  $\rho_{ji}$ . esto es, el patrón reconocido por el modulo  $j$  en el nivel  $i$ . La función de cada reconocedor de patrones es la de reconocer su correspondiente patrón. Veamos con más detalle los componentes del modelo:

- La función de entrada  $s()$ , representa la presencia de un patrón a reconocer.
- Existe una relación  $v$  de composición jerárquica estructural entre los módulos  $\Gamma_{ji}$  de los diferentes niveles  $X_i$ , tal que  $v:\Gamma_{rt} \rightarrow \Gamma_{lk}$ , donde  $t < k$ , y la relación “ $\rightarrow$ ” indica que  $\Gamma_{rt}$  de  $X_t$  está contenido o hace parte de  $\Gamma_{lk}$ , que pertenece al nivel  $X_k$  de más alto nivel. En otras palabras, un  $\Gamma$  de  $X_k$  se compone de diferentes  $\Gamma$  de  $X_t$  de más bajo nivel.



**Figura 3.1.** Esquema del sistema jerárquico de reconocimiento de patrones Ar2p Fuente (Puerto y Aguilar 2016a, Puerto y Aguilar 2017)

- Cada nivel  $i$  produce una señal de salida (de reconocimiento o aprendizaje), en función de las respuestas de sus módulos.

*Postulado.* Sea  $\rho$  un patrón reconocido por el módulo  $\Gamma_{ji}$  (módulo  $j$  de reconocimiento de patrones del nivel  $i$ ). Si  $\rho$  tiene un número finito de sub-patrones que lo componen, entonces los módulos  $\Gamma$  que lo reconocen no pueden encontrarse en el mismo nivel de reconocimiento de  $i$ , sino en un nivel más abajo ( $i-1$ ) en la jerarquía. Así, las salidas de los módulos de reconocimiento de más bajo nivel se convierten en señales de reconocimiento para los módulos de más alto nivel que lo contienen, estableciendo de esta manera la relación  $v: \Gamma_{rt} \rightarrow \Gamma_{lk}$ .

### 3.1.1.1 Estructura formal del módulo de reconocimiento de patrones Ar2p.

Un módulo  $\Gamma$  de reconocimiento de patrones para *Ar2p* se define formalmente como una 3-tupla  $\Gamma = \langle E, U, S_o \rangle$  (ver Tabla 3.1), donde  $E$  es una matriz conformada por la 2-tupla  $E = \langle S, C \rangle$ , tal que  $S = \langle \text{Señal, Estado} \rangle$  es una matriz que representa el conjunto de señales que conforman al patrón reconocido por  $\Gamma$ , y sus respectivos estados. La variable de Estado es “True” cuando la señal está presente, y “False” en caso contrario. Cada señal presente en  $\Gamma$  tiene su correspondiente reconocedor del patrón de nivel más bajo, y cuando ha sido reconocido entonces su estado es “True”. El número de señales que conforman al patrón es específico para cada módulo  $\Gamma$ . Por otro lado,  $C = \langle D, V, P \rangle$  es la matriz que codifica información sobre el patrón, donde  $D$  son los descriptores/características propias del patrón en  $\Gamma$ ;  $V$  es el vector de dominio o variabilidad esperada del descriptor (por ejemplo, “de tiempo”, “de distancia”, etc.); y  $P$  es el peso de importancia del descriptor para el reconocimiento del patrón. A su vez,  $U$  es el vector de umbrales usado por el módulo  $\Gamma$  para el reconocimiento del patrón, existiendo dos tipos de umbrales:  $\Delta U1$  es el umbral de reconocimiento usado por el algoritmo para el caso del reconocimiento del patrón por señales claves, y  $\Delta U2$  es el umbral de reconocimiento usado por el algoritmo para el caso del reconocimiento del patrón por mapeo parcial y/o total (ver sección 3.1.1.2). Finalmente,  $S_o$  es la señal de salida. Cada módulo  $\Gamma$  produce una única señal de salida ( $S_o$ ) de reconocimiento del patrón hacia los niveles superiores, o señales de petición hacia niveles más bajos de reconocimiento. Cuando hay reconocimiento, esta señal  $S_o$  es la salida del módulo, y se difunde a los niveles superiores alcanzables desde ese nivel, haciendo que se modifique el estado de la señal a “True” en los patrones de más alto nivel del cual hace parte.  $S_o$  como petición se convierte en la señal de entrada  $s()$  para los módulos de reconocimiento de patrones de nivel inferior.

**Tabla 3.1.** Tupla  $\Gamma = \langle E, U, S_o \rangle$ 

E				
S		C		
Señal	Estado	Descriptor (D)	Dominio de Valores (V)	Peso de Importancia (P)
1	Falso	Descriptor1	<Dominio Descriptor1>	[0,1]
2	Falso	Descriptor2	<Dominio Descriptor2>	[0,1]
3	Falso	Descriptor3	<Dominio Descriptor2>	[0,1]
...	...	...	...	...
n	Falso	Descriptorn	<Dominio Descriptorn>	[0,1]
Umbrales de reconocimiento: $\Delta U1, \Delta U2$				

### 3.1.1.2 Teoremas usado por Ar2p para el reconocimiento de un patrón estático.

Para reconocer un patrón se han definido dos teoremas (Puerto y Aguilar 2017; Puerto y Aguilar 2016a):

*Teorema 1. Reconocimiento por señales claves.* En este caso se usan para el reconocimiento, los descriptores con mayor peso de importancia para el patrón a reconocer y el umbral  $\Delta U1$  (ambos definidos de manera supervisada). Este teorema establece que se reconoce un patrón  $\rho$  por señales claves, si la suma de los descriptores de mayor importancia reconocidos es mayor al umbral  $\Delta U1$ . Los descriptores de mayor importancia son denominados señales claves para ese patrón.

Una señal  $S$  activa en el módulo  $\Gamma$  es clave, si su peso de importancia tiene un valor mayor o igual al peso promedio de todas las señales en  $\Gamma$ . Sea  $S(\Gamma) = \{S_i \mid S_i \in \Gamma, \text{ con } i=1, \dots, n\}$ ,  $P(S_i)$  el peso de importancia de la señal particular  $S_i$ ,  $P_m(S(\Gamma))$  el peso promedio de las señales que hacen parte de  $\Gamma$ , y  $Clave\Gamma = \{S_i \mid S_i \in \Gamma \text{ y } S_i \text{ es clave}\}$  el conjunto de señales claves, la Ecuación (3.1) define una señal clave.

$$\forall S_i \in S(\Gamma) \text{ si } [P(S_i) \geq P_m(S(\Gamma))] \rightarrow S_i \in Clave\Gamma(S_i) \quad (3.1)$$

Con base en lo anterior, el Teorema 1 de reconocimiento por señales clave matemáticamente se formaliza en la Ecuación (3.2).

$$\frac{\sum_{i=1}^n | S_i \in S(\Gamma) \cap Estado(S_i=True) \cap S_i \in Clave_{\Gamma} | P(S_i)}{|Clave_{\Gamma}|} \geq \Delta U1 \rightarrow S_0 \quad (3.2)$$

*Teorema 2. Reconocimiento por mapeo() parcial.* En este caso, se usan todas las señales para el reconocimiento. Este teorema establece que se reconoce un patrón  $\rho$  por *mapeo() parcial*, si la suma de los descriptores reconocidos es mayor al umbral  $\Delta U2$ . Este teorema se formaliza en la Ecuación (3.3):

$$\frac{\sum_{i=1}^n | s_i \in S(\Gamma) \cap Estado(s_i=true) | P(S_i)}{n} \geq \Delta U2 \rightarrow S_0 \quad (3.3)$$

Estos teoremas son usados por cada módulo de cada nivel de reconocimiento  $X_i$  (desde  $X_1$  hasta  $X_m$ ) de *Ar2p*.

### 3.1.1.3 Modelo general computacional de *Ar2p*

*Ar2p* adopta las teorías sobre el comportamiento del cerebro, presentadas en la sección 2.2.3: la teoría de la mente basada en el reconocimiento de patrones (PRTM, por sus siglas en inglés) (Kurzweil 2013), los modelos de aprendizaje de patrones *On-intelligence* (Hawkins y Blakeslee 2007), y *Deep Learning* (LeCun, Bengio y Hinton 2015).

Estas teorías son complementarias y describen el proceso neurofisiológico a gran escala del reconocimiento y aprendizaje de patrones, basadas en las siguientes hipótesis (Mountcastle 1997; Kurzweil 2013): i) el proceso de reconocimiento de patrones se hace a través de módulos especializados de reconocimiento de patrones; ii) el proceso de reconocimiento es uniforme; y consiste de estructuras básicas, llamadas columnas corticales, las cuales a su vez están compuestas

de mini-columnas corticales organizadas jerárquicamente; iii) si un nivel de reconocimiento del neocórtex es incapaz de reconocer un patrón, este es enviado al nivel inferior; si ninguno de los niveles puede reconocer el patrón, este es considerado un patrón nuevo. Inspirados en lo anterior, la Figura 3.2 especifica algorítmicamente el proceso de reconocimiento de  $Ar2p$ .

*Descripción General del Algoritmo:* El mecanismo general consiste en que el módulo de reconocimiento de nivel superior invoca a los módulos de nivel inferior que los constituyen, y estos de manera recursiva hacen lo mismo. El algoritmo funciona de la siguiente manera, el sistema de reconocimiento recibe un patrón de la entrada a reconocer  $s()$  (línea 1), y genera señales de reconocimiento de  $s()$  como señal de salida  $S_o$  en diferentes casos (líneas 10, 13, 19, 31, 34, 40). Además, tiene dos pasos para aprender: uno para cuando hubo reconocimiento del patrón de entrada, en ese caso hay un aprendizaje por refuerzo (líneas 8, 29, ver sección 3.2 para más detalle); y otro cuando no hubo reconocimiento, ya que es un patrón nuevo (líneas 17, 38, ver sección 3.2 para más detalle).

Los dos mecanismos de aprendizaje son *Aprendizaje\_nuevo* y *Aprendizaje\_por\_refuerzo* (ver capítulo 3.2 para más detalle, donde se especifican los algoritmos de aprendizaje propuestos). El primero crea un nuevo módulo de reconocimiento para un nuevo patrón, y el segundo ajusta el módulo a los cambios existentes en el patrón de entrada. Recibida  $s()$ , se descompone en sus subpatrones (línea 2), y se determina el nivel  $X_i$  de la jerarquía como inicio del reconocimiento de  $s()$  (línea 3). Fijado el nivel, entonces se crea L peticiones de reconocimiento de  $(y)$  a través de  $s()$  (línea 4). Si son patrones atómicos (nivel  $X_1$ ), entonces se realiza el proceso de reconocimiento de los patrones atómicos (líneas 5-23), de lo contrario se envían L solicitudes a los niveles inferiores (línea 25), y se espera sus reconocimientos (línea 26). A continuación, se calcula el reconocimiento del patrón actual (línea 27). Si el reconocimiento fue exitoso (Ecuaciones (3.2) y (3.3)) (línea 28),

y si es el último nivel de la jerarquía ( $X_m$ , nivel donde inicio el reconocimiento), entonces se realiza un aprendizaje por refuerzo (línea 29, ver sección 3.2 para más detalle), y se genera una señal de salida  $S_0$  del módulo, que se convierte en la señal de salida del sistema “patrón reconocido” (línea 31). Si no es el tope de la jerarquía, entonces se crea y se envía la señal  $S_0$  de reconocimiento de ( $y$ ) a los niveles superiores (líneas 33-34). Si no se reconoce nada y ese módulo ha recibido varias señales atómicas (línea 37), entonces se realiza aprendizaje nuevo (línea 38, ver sección 3.2 para más detalle), y se reconoce como patrón nuevo (línea 39). Por esa razón, se envía la señal de salida  $S_0$  de reconocimiento por aprendizaje. En la línea 25 acontece la llamada recursiva del sistema. Las  $s()$  son recibidas por los módulos de reconocimiento de más bajo nivel, que pueden seguir descomponiendo el patrón (línea 2) hasta llegar al nivel  $X_1$  (línea 5), donde se inicia el proceso de reconocimiento de los patrones atómicos.

La descripción de la propuesta, en términos recursivos, es la siguiente: el algoritmo reduce progresivamente la complejidad del patrón a reconocer (ver línea 25 del algoritmo de la figura 3), y al llegar al nivel atómico (patrones base no divisibles)  $X_1$  (ver línea 5 del algoritmo), inicia el proceso de reconocimiento de patrones, desde los atómicos hasta el más complejo. Todo ese proceso se resuelve de forma recursiva, tal que a medida que se descompone se invocan los respectivos módulos de reconocimiento de patrones del siguiente nivel, hasta llegar el nivel atómico, el cual regresa sí reconoció o no (líneas 6 al 23) a los módulos de reconocimiento de su nivel superior, los cuales a su vez realizan lo mismo (regreso de la llamada recursiva, ver líneas 27 al 41). El proceso de reconocimiento de los niveles superiores se basa en los Teoremas 1 y 2. Así, vemos que el algoritmo se invoca a sí mismo (cuando realiza solicitudes de reconocimiento

(línea 4 del algoritmo), y va regresando los resultados (si reconoció o no a su patrón) de cada llamada recursiva (líneas 7 al 23 del algoritmo).

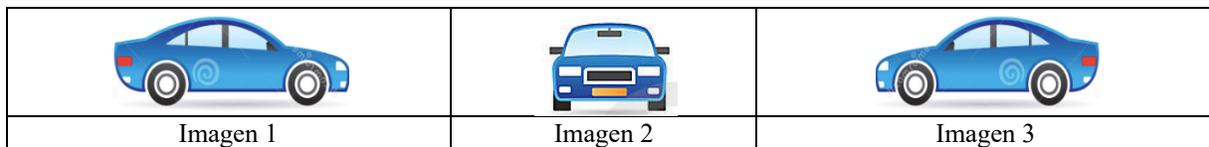
<i>Algoritmo recursivo de reconocimiento de patrones Ar2p</i>	
1.	y= s()
2.	L= descomponer(y) // extraer sub-patrones
3.	Determinar el nivel $X_i$ de la jerarquía como inicio del reconocimiento de y
4.	Crear L solicitudes de petición de reconocimiento de (y)
5.	IF y = señales de más bajo nivel $X_{i=1}$ o tiene señales atómicas reconocidas THEN //caso <i>Bottom-Up</i> o regreso de la recursión caso <i>Top-Down</i> .
6.	Calcular reconocimiento (y)
7.	IF calcular exitoso por cualquier estrategia THEN
8.	Realizar <i>Aprendizaje por refuerzo</i>
9.	IF Ultimo nivel de la Jerarquía $X_m$ , y $X_m \neq X_{i=1}$ // $X_m$ del s() inicial
10.	Enviar $S_o$ de reconocimiento (Y) salida del sistema.
11.	ELSE
12.	Crear salida $S_o$ // armar la señal a difundir
13.	Enviar la salida $S_o$ de reconocimiento de (y) a nivel superior // hay difusión.
14.	FIN IF
15.	ELSE // caso cuando no se reconoce algo nuevo
16.	IF señales atómicas reconocidas // Todas.
17.	Realizar <i>Aprendizaje nuevo</i> en ese nivel.
18.	Crear $S_o$ // armar la señal a difundir.
19.	Enviar $S_o$ de reconocimiento de (y) a nivel superior //difusión
20.	ELSE ignorar //información insuficiente para aprender nuevo patrón
21.	FIN IF
22.	FIN IF // cierre de IF línea 7
23.	FIN IF // cierre del IF de señales atómicas línea 5.
24.	ELSE // caso top-down
25.	Enviar L solicitudes de petición de reconocimiento de (y) // a los niveles inferiores. Ésta Línea genera el caso general de la recursividad.
26.	Recibir L respuestas de los niveles inferiores //
27.	Calcular reconocimiento de (y)
28.	IF calcular exitoso por cualquier estrategia THEN
29.	Realizar <i>Aprendizaje por refuerzo</i>
30.	IF Ultimo nivel de la Jerarquía $X_m$ // $X_m$ del s() inicial
31.	Enviar $S_o$ de reconocimiento (Y) salida del sistema.
32.	ELSE // seguir subiendo en el reconocimiento
33.	Crear $S_o$ // armar la señal a difundir.
34.	Enviar $S_o$ de reconocimiento de (y) a nivel superior //difusión
35.	FIN IF
36.	ELSE // caso cuando no se reconoce algo nuevo
37.	IF señales atómicas reconocidas // Todas.
38.	Realizar <i>Aprendizaje nuevo</i> en ese nivel.
39.	Crear $S_o$ // armar la señal a difundir.
40.	Enviar $S_o$ de reconocimiento de (y) a nivel superior //difusión
41.	ELSE ignorar //información insuficiente para aprender nuevo patrón
42.	FIN IF
43.	FIN IF
44.	FIN IF

**Figura 3.2.** Algoritmo de reconocimiento de patrones Ar2p. Fuente (Puerto y Aguilar 2017)

De esta manera, el caso base de la recursividad es el nivel atómico, y los procesos de descomposición y reconocimiento (basados en los Teoremas 1 y 2) que se dan en los otros niveles son los casos generales. Además, entre ellos solo se envían señales indicando si se reconoció o no el patrón respectivo. Es decir, los parámetros que se envían son simplemente señales, diciendo sí reconoció o no.

### 3.1.1.4 Caso de prueba ilustrativo de *Ar2p* aplicado en el reconocimiento de un “Vehículo”.

Un vehículo tiene múltiples vistas. Estas vistas dependen tanto del observador como del vehículo mismo. Por ejemplo, el ángulo de observación del vehículo, el estado del vehículo (chocado, deteriorado, etc.), entre otros. La Figura 3.3 muestra tres perfiles de observación de un automóvil: vista delantera (imagen 2 de la figura 3.3), y dos vistas laterales (imágenes 1 y 3 de la Figura 3.3).



**Figura 3.3.** Tres imágenes de perfil de un Vehículo tipo automóvil

Para el reconocimiento del patrón de la Figura 3.3 (en cualquiera de sus tres vistas) usando *Ar2p*, se consideran los siguientes sub-patrones: puertas, faros, ruedas, espejos, placas, etc. Se asume que, con esa información del escenario, se instancia *Ar2p* para el reconocimiento del automóvil. Se definen dos espacios de reconocimiento:  $X_{i=1}$  para patrones atómicos, que en este escenario son los sub-patrones de análisis, i.e., las partes del vehículo; placa, espejos, ruedas, etc. El otro espacio es  $X_{i=2}$ , que en este escenario consta de tres módulos de reconocimiento, uno para cada perfil del vehículo (ver Figura 3.3).

Para mostrar el algoritmo *Ar2p*, se toma como entrada la imagen 2 de la Figura 3.3. Sea  $y=s()$  = la imagen 2 de la Figura 3.3 (línea 1 del algoritmo), el siguiente paso consiste en descomponer la imagen del vehículo en los sub-patrones de análisis (línea 2). En este caso se tienen como subpatrones: espejos retrovisores laterales, espejo retrovisor central, farolas delanteras, llantas delanteras y placa, etc. El proceso de descomposición particular se logra con métodos de segmentación de imágenes (por ejemplo, por métodos de detección de bordes). Una vez se tiene las partes de la imagen del vehículo particionadas, se determina el espacio de reconocimiento  $X_i$  en la arquitectura propuesta (ver Figura 3.1), a partir del cual continuar con el proceso de reconocimiento. En esta prueba, el nivel de inicio de reconocimiento es el 2,  $X_{i=2}$ . Después, se procede a crear L solicitudes de petición de reconocimiento de las partes que conforman al vehículo o patrón y (línea 4). Así, dado que el nivel actual del proceso es  $X_{i=2}$ , para profundizar el proceso de análisis del patrón se envían L solicitudes de petición de reconocimiento de cada una de las partes del vehículo ( $y$ ) a los niveles inferiores (línea 25). Esta línea invoca el proceso recursivo, generándose nuevos procesos de reconocimiento en paralelo e independientes, para cada uno de los sub-patrones. De esta manera, el *proceso\_principal* invoca la recursividad. A continuación, se inician los nuevos procesos de reconocimiento invocados en la línea 25 por parte del *proceso\_principal*, para el respectivo reconocimiento de las partes del vehículo contenidas en el patrón original, por ejemplo: L1=sub-patrón de imagen espejo retrovisor lateral izquierdo, etc. Veamos el caso del reconocimiento de L1.

Ahora  $y=s()$ =sub-imagen de espejo retrovisor lateral izquierdo de la imagen 2 de la Figura 3.3. El siguiente paso (línea 2), consiste en descomponer la sub-imagen del vehículo en los sub-patrones de análisis, en este caso el conjunto de sub-patrones de análisis es vacío. Por cuanto es un patrón atómico, se determina que el espacio de reconocimiento  $X_i$  es el nivel 1,  $X_{i=1}$ . Ya estando

en el nivel  $X_{i=1}$  de inicio de reconocimiento, se procede a crear  $L$  solicitudes de petición de reconocimiento de las partes que conforman al vehículo o patrón y (línea 4).  $L$  contiene sólo el patrón de entrada por ser indivisible. Dado que el nivel actual del proceso es  $X_{i=1}$ , se cumple la condición para realizar el cuerpo de instrucciones entre las líneas 6 y 23. Se procede entonces a realizar el reconocimiento del patrón del espejo retrovisor lateral ( $y$ ) (línea 6), y consiste en mapear dicho patrón con la base de datos de patrones atómicos que constituye el espacio de reconocimiento  $X_{i=1}$ . Damos por hecho que el mapeo es exitoso (línea 7), por lo que se procede a aprender (Puerto y Aguilar 2016b). Se mira si el proceso está en el último nivel de la jerarquía  $X_m$ . Como no se ha alcanzado el último nivel (que es  $X_{m=2}$ ), se pasa al ELSE del condicional (línea 11), que consiste en crear la salida  $S_o$ , y enviar dicha salida  $S_o$  de reconocimiento del patrón espejo retrovisor lateral al nivel superior (difusión de una señal de reconocimiento positivo (True) al *proceso principal*). En este punto se termina este proceso interno, que fue invocado en la Línea 25, donde quedo el proceso principal. Este mismo proceso se aplica para las demás solicitudes de reconocimiento. Si asumimos que hubo reconocimiento de todas las partes, el proceso principal retoma el control del programa donde se hizo la solicitud de los  $L$  reconocimientos (línea 25), y se procede a instanciar la estructura de datos de la Tabla 3.1, con las respuestas recibidas de las solicitudes de reconocimiento (ver Tabla 3.2).

La Tabla 3.2 muestra las 9 señales que corresponden a los sub-patrones del proceso de descomposición de la imagen del vehículo, con todos los estados de los patrones reconocidos (estado="True"). En este experimento se ha asignado el peso de importancia de acuerdo a su influencia para el reconocimiento. Por ejemplo, los descriptores de las señales 1, 2, 3, 5 y 8 tienen poco peso (0.5), puesto que son partes opcionales o accesorios. En cambio, las señales 4, 6, 7 y 9,

que corresponden a los faroles, persiana, llantas y vidrios parabrisas, tiene bastante peso (1), pues son claves para el reconocimiento del vehículo.

**Tabla 3.2.** Tupla  $\Gamma$  para el vehículo: vista frontal

S		E		
Señal	Estado	Descriptor (D)	Dominio de Valores (V)	Peso (P)
1	True	Imagen espejo retrovisor lateral izq.	<Espejos Izquierdos>	0.5
2	True	Imagen espejo retrovisor lateral Der	<Espejos derechos>	0.5
3	True	Imagen espejo retrovisor central	<Espejos Centrales>	0.5
4	True	Imagen de farola	<Farolas>	1
5	True	Imagen exploradora	<Exploradoras>	0.5
6	True	Imagen persiana	<Persianas>	1
7	True	Imagen llanta	<Llantas>	1
8	True	Imagen placa	<Placas>	0.5
9	True	Imagen parabrisas delantero	<Parabrisas delanteros>	1

$\langle \Delta U1, \Delta U2 \rangle$

Continuando con el proceso de reconocimiento del proceso principal, en la línea 26 se reciben las 9 respuestas de los niveles inferiores de la solicitud L. Una vez recibidas estas señales en el módulo de reconocimiento de  $X_2$  desde donde se invocó la solicitud de reconocimiento, se procede a calcular el reconocimiento de los patrones de los módulos que las contienen (línea 27). En este caso, el módulo de interés que vamos a analizar es el que entra en reconocimiento del perfil del vehículo frontal, imagen 2 de la Figura 3.3.

Para calcular el reconocimiento, se usan los Teoremas 1 y 2. El Teorema 1 es para el reconocimiento del vehículo por señales claves. En este caso se ha definido el umbral  $\Delta U1 = 0.75$ . Aplicando la Ecuación (3.1), las señales claves son:  $= \{i=4,6,7,9\}$ . Aplicando la Ecuación (3.2) para verificar el Teorema 1, se obtiene que  $1 > 0.75$ . Por tanto, hay reconocimiento por señales claves. Así, no se aplica el Teorema 2.

Después de eso, el proceso principal pasa a chequear el resultado del cálculo (línea 29), y si el cálculo es exitoso por cualquier teorema, se procede a ejecutar el aprendizaje (línea 29). Realizado el proceso de adaptación, se chequea si el proceso de reconocimiento del patrón vehículo está en

el último nivel de la jerarquía (Línea 30). En efecto, actualmente el proceso de reconocimiento está en  $X_{m=2}$ , que es el último nivel de la jerarquía, y coincide con el nivel de reconocimiento inicial. Por tanto, se crea la señal de salida  $S_o$  de reconocimiento ( $y=s()$  inicial), correspondiente a la imagen 2 de la Figura 3.3 (Línea 31).

### 3.1.2 Formalización matemática de $Ar2p$ para el reconocimiento de patrones dinámicos

En la sección anterior (sección 3.1.1), se trató el problema de reconocimiento de patrones estáticos. En esta sección, se describe la extensión de  $Ar2p$  para el reconocimiento de patrones dinámicos (Puerto, Aguilar y Chávez 2018).

En general, los patrones pueden ser divididos en dos grandes categorías: patrones estáticos y patrones dinámicos (Kelso 2014; Chen 2005; Jesan 2004). Los patrones dinámicos están caracterizados por tener propiedades que cambian en el tiempo, tal como un peatón caminando o un vehículo en movimiento. En (Felzenszwalb, McAllester, y Ramanan 2008), el análisis de los patrones dinámicos se realiza desde tres aspectos: definición de su representación, uso del principio “divide para reconocer”, y uso de información contextual intrínseca. Otros estudios han intentado considerar otros aspectos de un patrón, por ejemplo, las partes deformables del patrón (Kelso 2014), o los cambios dinámicos en el patrón.

Este capítulo aborda el problema de reconocimiento de patrones dinámicos (Puerto, Aguilar y Chávez 2018) a partir de: i) la definición formalmente del concepto de patrón dinámico; ii) la formalización matemática de  $Ar2p$  para reconocer series de tiempo; iii) el reconocimiento de patrones dinámicos con  $Ar2p$  desde dos perspectivas: una centrada en el reconocimiento orientado por características, y la otra centrada en el reconocimiento orientado por la percepción.

*Definición 1. Patrón dinámico.* Un patrón dinámico es un patrón que cambia a nivel de sus características/descriptores, o en la percepción que se tiene del mismo, en un intervalo de tiempo, el cual es: *a)* una abstracción de un objeto espacial o temporal; *b)* una colección (posiblemente ordenada y estructurada) de descriptores que lo representan. Además, sus descriptores: *c)* son usados para el reconocimiento del patrón; *b)* son representados por variables acotadas en un dominio finito de valores; *d)* representan el conjunto de información semántica de interés; *e)* pueden ir cambiando en el tiempo (en sus dominios de posibles valores).

Un patrón dinámico es reconocido, cuando en la evaluación de sus descriptores (tanto por características como por percepción), en un intervalo de tiempo  $[t_k, t_j]$ , se detectan cambios en el dominio de sus posibles valores. En ese sentido, los patrones dinámicos son analizados desde dos perspectivas: reconocimiento de patrones dinámicos orientado/guido por características, y reconocimiento de patrones orientado/guido por la percepción.

- Patrones dinámicos orientado por características (en adelante *DpoC*, por sus siglas en inglés: Un patrón es *DpoC* cuando tiene características que cambian en el tiempo, por ejemplo, la expresión de las emociones.
- Patrones dinámicos orientado por percepción (en adelante *DpoP*, por sus siglas en inglés): un patrón es *DpoP* cuando la percepción del patrón va cambiando (vista como un todo: forma o apariencia) según lo que vamos viendo. Por ejemplo, un vehículo en movimiento.

### 3.1.2.1 Formalización matemática de un patrón dinámico.

Un patrón dinámico es formalmente definido como un 3-tupla:

$$\rho_d = \langle D_n, f_\Delta, \Delta_\tau \rangle \text{ Donde,} \quad (3.4)$$

$$D_n = [d_1, d_2, d_3 \dots d_i, d^1, d^2, d^3 \dots d^j], \quad (3.5)$$

En la Ecuación 3.5,  $d_i$  denota los descriptores por características, y  $d^j$  los descriptores por percepción. La percepción se realiza según nuestros sentidos: visual, auditivo, táctil, etc. (o una combinación de estos).

Cada descriptor tiene un dominio, que es el universo de valores posibles (ver Ecuaciones 3.6 y 3.7); cada valor, puede ser valores numéricos, etiquetas, valores continuos, etc.

Por otro lado,  $\widehat{d}_x$  es el vector de dominio  $d_x$  ( $x$  características), y tiene un rango de valores desde  $v_{1x}$  a  $v_{kx}$ , con  $x=1,2,3,\dots$ , hasta un valor arbitrario y  $k$  como el valor del descriptor en el tiempo  $t$ . (ver Ecuación. 3.6).

$$\widehat{d}_x \equiv [v_{1x}, v_{2x}, v_{3x}, \dots, v_{kx}] \quad (3.6)$$

$\widehat{d}^y$  es el vector de dominio de  $d^y$  ( $y$  percepciones), y tiene un rango de valores desde  $v^{1y}$ , hasta  $v^{ky}$ , con  $y=1,2,3,\dots$ , hasta un valor arbitrario y  $k$  como el valor del descriptor en el tiempo  $t$ . (ver Ecuación. 3.7).

$$\widehat{d}^y \equiv [v^{1y}, v^{2y}, v^{3y}, \dots, v^{ky}] \quad (3.7)$$

Finalmente,  $f_\Delta$  es el vector de funciones  $f_\Delta$ ; donde,  $gcd_i$  y  $gcd^j$  son las funciones de cambio para

$$f_\Delta = [gcd_1, gcd_2 \dots gcd_i, gcd^1, gcd^2 \dots gcd^j] \quad (3.8)$$

cada descriptor. La cardinalidad de  $|f_\Delta| = |D_n|$  (ver Ecuación. 3.8).

$\Delta_\tau$  es una serie de tiempo para cada descriptor  $d_i$ ,  $d^j$  en  $D_n$ , ordenado cronológicamente (podría ser continuo). Cada descriptor puede tener una tasa de tiempo diferente de cambio. Por ejemplo, sea  $\Delta\tau d_1$  el vector de cambio del descriptor  $d_1$ , que se muestran en la Ecuación 3.9.

$$\Delta\tau d_1 = [(t_1, v_{11}), (t_2, v_{21}), (t_3, v_{31}) \dots (t_n, v_{n1})] \quad (3.9)$$

En cada par ordenado de  $\Delta\tau d_1$ , el primer elemento  $t_\tau$  representa el tiempo de cambio, y el segundo elemento  $v_{\tau i}$  el valor obtenido. De tal manera que  $(t_1, v_{11})$  significa que  $d_1$  en el tiempo  $t_1$  tiene el valor  $v_{11}$ ,  $(t_2, v_{21})$  significa que  $d_1$  en el tiempo  $t_2$  tiene el valor  $v_{21}$ , etc. Si un descriptor permanece constante, su tasa de cambio es igual a cero (0),  $\Delta_\tau = 0$ .

### 3.1.2.2 Descripción matemática para un módulo de reconocimiento de patrones dinámicos.

Un módulo de reconocimiento de patrones dinámicos contiene la información necesaria para reconocer un patrón dinámico (descriptores, estado de descriptores, dinámica, pesos). Se usará la notación  $\Gamma\rho_d$  para representar un módulo de reconocimiento de patrones dinámico. Un  $\Gamma\rho_d$  es formalmente definido como una 3-tupla (ver Tabla 3.3):

$$\Gamma\rho_d = \langle E_d, U_d, S_d \rangle \quad (3.10)$$

Donde,  $E_d$  es un arreglo compuesto de 2-tuplas:  $E_d = \langle S_d, C_d \rangle$ .

**Tabla 3.3.** Módulo  $\Gamma\rho_d$

$S_d$		$E_d$	
		$C_d$	
Señal	Estado	Apuntador (A)*	Peso de Importancia (P)
1	False	Apuntador <sub>1</sub>	[0,1]
...	...	...	[0,1]
$n$	False	Apuntador <sub>i</sub>	[0,1]
1	False	Apuntador <sup>l</sup>	[0,1]
...	...	...	...
$m$	False	Apuntador <sup>l</sup>	[0,1]
Umbral: $\Delta U1, \Delta U2$			

\* Un apuntador a una serie de tiempo, en la cual se almacena la información

$S_d = \langle \text{Señal}, \text{Estado} \rangle$  es un arreglo que representa el conjunto de señales dinámicas que conforman al patrón a reconocido por  $\Gamma\rho_d$ , y sus respectivos estados.  $C_d = \langle A, P \rangle$ , A son apuntadores a cada una de las series de tiempo  $\Delta_\tau$ . La columna de pesos (P) es un campo que contiene el valor de importancia del descriptor, como facilitador o determinante para el reconocimiento.  $\Delta U1$ : es el vector de umbrales usado por el modulo  $\Gamma\rho_d$ , para reconocer su respectivo patrón dinámico por

señales claves.  $\Delta U2$  es el vector de umbrales usado por el módulo  $\Gamma_{\rho_d}$ , para reconocer su respectivo patrón dinámico por señales parciales.  $S_o$  es la señal de reconocimiento, o señal de petición hacia los niveles inferiores.

### 3.1.2.3 Teoremas extendidos usado por Ar2p para el reconocimiento patrones dinámicos

Esta sección describe los teoremas extendidos usados en  $\Gamma_{\rho_d}$ , para reconocer un patrón dinámico. Para reconocer un patrón dinámico, sus descriptores son evaluados en un intervalo de tiempo  $[t_i, t_j]$ . El método general de reconocimiento verifica si la información del patrón de entrada es la misma, que la información almacena en la serie temporal. Si la serie temporal es consistente, el patrón es reportado como conocido (“True”). En otro caso, es reportado como desconocido (“False”). El modelo trabaja con umbrales fijos preestablecidos.

*Teorema 1 extendido para patrones dinámicos* (Puerto, Aguilar y Chávez 2018): Estrategia de reconocimiento de patrones dinámicos por señales claves. Una señal dinámica  $S_i = (d_i \text{ or } d^j)$  es clave, si su peso de importancia tiene un valor mayor o igual al promedio de todas las señales en  $\Gamma_{\rho_d}$ . Esta estrategia usa el umbral  $\Delta U1$  (ver Ecuaciones 3.11 y 3.12)

$$\forall d_i \in S_{\Gamma_{\rho_d}}, \text{ Si } [P(d_i) \geq P_m(S_{\Gamma_{\rho_d}})] \rightarrow \text{Clave}_{\Gamma_{\rho_d}}(d_i) \quad (3.11)$$

$$\forall d^j \in S_{\Gamma_{\rho_d}}, \text{ Si } [P(d^j) \geq P_m(S_{\Gamma_{\rho_d}})] \rightarrow \text{Clave}_{\Gamma_{\rho_d}}(d^j) \quad (3.12)$$

*Definición 2. Reconocimiento de patrón dinámico por señales claves de características.* Un patrón dinámico es reconocido por señales dinámicas claves de características, si se cumple que (ver la Ecuación (3.13)):

$$\frac{\sum_{r=t_k}^{t_j} \sum_{i=1}^n \mathbb{1}_{\text{estado}(d_i=true) \cap d_i \in \text{Clave}_{\Gamma\rho_d}} P(d_{ir})}{|\text{Clave}_{\Gamma\rho_d}|} \geq \Delta U1 \rightarrow S_o \quad (3.13)$$

Si los pesos alcanzan el umbral de reconocimiento de  $\Delta U1$ , entonces se genera una salida de reconocimiento del patrón que cambia su estado a “True”, en aquellos módulos de reconocimiento de patrones de alto nivel que los contiene.

*Definición 3. Reconocimiento de patrón dinámico por señales claves de percepción.* Un patrón dinámico es reconocido por señales dinámicas claves de percepción, si se cumple que (ver Ecuación 3.14):

$$\frac{\sum_{r=t_k}^{t_j} \sum_{i=1}^n \mathbb{1}_{\text{estado}(d^i=true) \cap d^i \in \text{Clave}_{\Gamma\rho_d}} P(d_r^i)}{|\text{Clave}_{\Gamma\rho_d}|} \geq \Delta U1 \rightarrow S_o \quad (3.14)$$

*Teorema 2 extendido para patrones dinámicos* (Puerto, Aguilar y Chávez 2018): Estrategia de reconocimiento de patrón dinámico por mapeo parcial de características (ver Ecuación 3.15). Esta estrategia consiste en validar que un número mínimo de señales presentes en  $\Gamma\rho_d$ , sea superior al umbral de reconocimiento  $\Delta U2$ .

$$\frac{\sum_{r=t_k}^{t_j} \sum_{i=1}^n \mathbb{1}_{\text{estado}(d_i=true)} P(d_{ir})}{n} \geq \Delta U2 \rightarrow S_o \quad (3.15)$$

Si los pesos alcanzan el umbral  $\Delta U2$  de reconocimiento, entonces se genera una salida de reconocimiento del patrón que cambia su estado a “True”, en aquellos módulos de reconocimiento de patrones de alto nivel que los contiene.

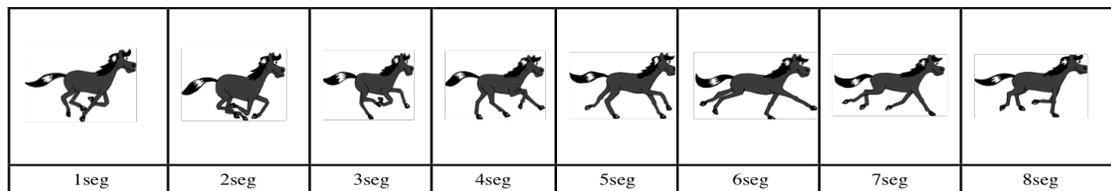
*Definición 4. Reconocimiento de patrón dinámico por mapeo parcial de percepción.*

Un patrón dinámico es reconocido por mapeo parcial, si la Ecuación 3.16 se cumple:

$$\frac{\sum_{r=t_j}^{t_k} \sum_{i=1}^n \cap \text{estado}(d^i=\text{true}) P(d_r^i)}{n} \geq \Delta U_{2 \rightarrow S_o} \quad (3.16)$$

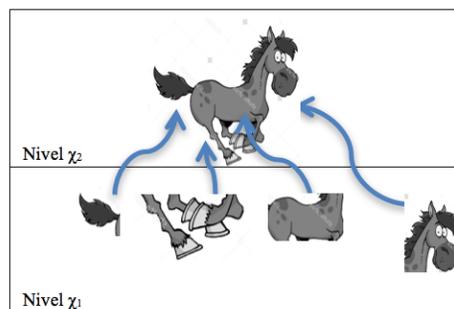
### 3.1.2.4 Caso de reconocimiento de patrón dinámico usando Ar2p: “Caballo galopando”.

Esta subsección se basa en el trabajo (Puerto, Aguilar y Chávez 2017; Puerto, Aguilar y Chávez 2018). La tarea consiste en reconocer el patrón de la Figura 3.4. Se supone que el patrón consiste en una secuencia de imágenes discretas grabadas en 8 segundos, una por segundo.



**Figura 3.4.** Caballo Galopando

Para este caso, se usará el método de reconocimiento del patrón dinámico orientado por percepción (*DpoP*), lo cual puede ser modelado por *Ar2p* con dos niveles (ver Figura 3.5).



**Figura 3.5.** Dos niveles de la Jerarquía de patrones para reconocer el patrón “caballo”

En este caso de prueba, el reconocimiento se realiza por el método orientado por percepción. A nivel de percepción, se tienen los siguientes descriptores por percepción: el descriptor de percepción de la cola, de las patas, del cuerpo y de la cabeza. La dinámica está definida por el

cambio de postura del caballo en función del tiempo (i.e. el cambio en los descriptores mencionados).

Sea la Ecuación 3.4,  $\rho_{d=caballo} = \langle D_n', f_\Delta', \Delta\tau' \rangle$ . Usando la Ecuación 3.5, se define el vector  $D_n$  con la colección de todos los  $n$  descriptores por percepción, para  $\rho_{d=caballo}$ :

$$D_n' = [d^{cola}, d^{pataIzqD}, d^{pataDerechaD}, d^{pataIzqT}, d^{pataDerechaT}, d^{Cuerpo}, d^{Cabeza}]$$

Ahora, se usa la Ecuación 3.7,  $\widehat{d^y} \equiv [v^{1y}, v^{2y}, v^{3y}, \dots, v^{ky}]$  para describir lo percibido. Cada valor representa un valor dinámico. Por ejemplo, para el descriptor de la cola, cuando  $v^{11} = 1$  significa la cola esta arriba,  $v^{12} = 2$  significa la cola va recta, y  $v^{13} = 3$  significa la cola esta abajo. Esto es similar para los otros descriptores.

$$\begin{aligned} \widehat{d^{cola}} &\equiv [1,2,3] \\ d^{pataDerechaD} &\equiv [1,2,3,4,5,6,7,8] \\ d^{pataIzqD} &\equiv [1,2,3,4,5,6,7,8] \\ d^{pataDerechaT} &\equiv [1,2,3,4,5,6,7,8] \\ d^{pataIzqT} &\equiv [1,2,3,4,5,6,7,8] \\ d^{Cuerpo} &\equiv [1,2] \\ d^{Cabeza} &\equiv [1,2,3] \end{aligned}$$

Los ocho valores  $v^{ij}$  de los descriptores de las patas representan, las ocho posibles posiciones de las patas, las cuales pueden ser descritas y representadas de diferentes maneras, por ejemplo, por ángulos internos, su distancia al cuerpo, su forma, etc. Por ejemplo,  $v^{11} = 1$  significa pata estirada,  $v^{12} = 2$  pata recogida, y así los otros. Adicional a esto, las funciones de cambio son definidas. Estas funciones modelan la transición (posibles cambios) de los valores de los descriptores (de acuerdo a sus dominios). A continuación, se definen las funciones de cambio para cada descriptor usando la Ecuación 3.8.

$$f_{\Delta}' = [fcd^{1cola}, fcd^{2pataDerechaD} \dots fcd^{8cabeza}]$$

Finalmente, cada uno de los descriptores puede tener una tasa de cambio de tiempo diferente, para ello se usa la Ecuación 3.9:

- $\Delta\tau d^{1cola} = [(t_1 = 0,5s, v^{11} = 1), (t_2 = 1s, v^{21} = 2), (t_3 = 1,05s, v^{31} = 3)]$
- $\Delta\tau d^{1pataDerechaD} = [(t_1 = 0,3s, v^{11} = 1), (t_2 = 0,4s, v^{21} = 2), (t_3 = 0,5s, v^{31} = 3), (t_4 = 0,6s, v^{31} = 4), (t_5 = 0,7s, v^{31} = 5), (t_6 = 0,8s, v^{31} = 6), (t_7 = 0,9s, v^{31} = 7), (t_8 = 10s, v^{31} = 9)]$
- $\Delta\tau d^{1patalzqD} = [(t_1 = 0,2s, v^{11} = 2), (t_2 = 0,4s, v^{21} = 3), (t_3 = 0,5s, v^{31} = 4), (t_4 = 0,6s, v^{31} = 5), (t_5 = 0,7s, v^{31} = 6), (t_6 = 0,8s, v^{31} = 7), (t_7 = 0,9s, v^{31} = 8), (t_8 = 10s, v^{31} = 1)]$
- $\Delta\tau d^{1patalzqT} = [(t_1 = 0,1s, v^{11} = 2), (t_2 = 0,4s, v^{21} = 3), (t_3 = 0,5s, v^{31} = 1), (t_4 = 0,6s, v^{31} = 5), (t_5 = 0,7s, v^{31} = 6), (t_6 = 0,8s, v^{31} = 7), (t_7 = 0,9s, v^{31} = 8), (t_8 = 10s, v^{31} = 4)]$
- $\Delta\tau d^{1pataDerechaT} = [(t_1 = 0,3s, v^{11} = 2), (t_2 = 0,4s, v^{21} = 3), (t_3 = 0,5s, v^{31} = 4), (t_4 = 0,6s, v^{31} = 5), (t_5 = 0,7s, v^{31} = 6), (t_6 = 0,8s, v^{31} = 7), (t_7 = 0,9s, v^{31} = 8), (t_8 = 10s, v^{31} = 1)]$
- $\Delta\tau d^{1cuerpo} = [(t_1 = 0,3s, v^{11} = 2), (t_2 = 0,4s, v^{21} = 1), ]$
- $\Delta\tau d^{1cabeza} = [(t_1 = 0,4s, v^{11} = 2), (t_2 = 0,6s, v^{21} = 1), (t_3 = 0,7s, v^{21} = 3)]$

A continuación, es analizado el comportamiento del reconocimiento con *Ar2p* para el caso de la Figura 3.5, usando los teoremas extendidos: En el nivel  $X_1$ , se tienen los módulos  $\Gamma_{j1}$  de reconocimiento de las partes del caballo, y de otros posibles animales (cola, cabeza, etc.); en el nivel  $X_2$  se tienen los módulos de reconocimiento de animales en movimiento (caballo, vaca, etc.)

Una vez montada la jerarquía, y sus respectivos niveles y módulos, el algoritmo recibe como entrada  $y=s()$  = "La imagen del caballo en el primer segundo de movimiento" (ver Figura 3.4). Se realiza el proceso de descomposición de la imagen en sus partes o sub-patrones <pataDerechaDelantera, pataizqDelantera, pataIzqTracera, pataDerechaTracera, cola, cuerpo,

cabeza > (ver las partes en el nivel 1 de la Figura 3.5). A continuación, se calcula el nivel  $X_i$  de partida/parada del proceso, en este caso,  $X_{i=2}$  (nivel de los animales). En este nivel ( $X_i = X_2$ ), sus módulos son cargados para cada instante de tiempo. Por ejemplo, la Tabla 3.4 muestra la estructura instanciada correspondiente al segundo 1 de la Figura 3.4: Un puntero<sup>i</sup> contiene la serie temporal correspondiente al descriptor  $i$  en ese primer segundo. Luego, se realizan o crean  $L$  peticiones de reconocimiento de  $y$ ,  $\langle y_1 = \text{pataDerechaDelantera}, y_2 = \text{pataizqDelantera}, y_3 = \text{pataIzqTracera}, y_4 = \text{pataDerechaTracera}, y_5 = \text{cola}, y_6 = \text{cuerpo}, y_7 = \text{cabeza} \rangle$  al nivel inferior. Asumiendo que se reciben  $L$  respuestas de reconocimiento de estos patrones, i.e., que estos descriptores (señales) se han activado (su estado paso a ser 'True') en los módulos del nivel donde fueron invocadas ( $X_2$ ), el siguiente paso es calcular con base en estas señales activas que módulos de ese nivel reconocen un patrón (usando los teoremas propuestos antes): como  $\Delta U_1 = 0.85$ , todas las señales están activas (ver Tabla 3.4), entonces, aplicando el método de reconocimiento por percepción ( $DpoP$ ) usando la Ecuación 3.12, las señales que son claves para reconocer este patrón son: 1, 6, y 7.

**Tabla 3.4.** Matriz  $E_d = \langle \text{Caballo Galopando} \rangle$  al segundo 1 de la Figura 3.4

$S_d$		$E_d$		
		$C_d$		
Señal	Estado	Descriptor	Puntero(P)	Peso (W)
1	T	$d^{cola}$	puntero <sup>1</sup>	0.9
2	T	$d^{pataDerechaD}$	puntero <sup>2</sup>	0.6
3	T	$d^{pataIzqD}$	puntero <sup>3</sup>	0.6
4	T	$d^{pataDerechaT}$	puntero <sup>4</sup>	0.6
5	T	$d^{pataizqT}$	puntero <sup>5</sup>	0.6
6	T	$d^{cuerpo}$	puntero <sup>6</sup>	0.8
7	T	$d^{cabeza}$	puntero <sup>7</sup>	0.9
$U_d : \langle 0.85, 0.65 \rangle$				

Un puntero<sup>i</sup>, contiene la serie temporal correspondiente al descriptor  $i$  en ese primer segundo. Luego se realizan o crean  $L$  peticiones de reconocimiento de  $y$ ,  $\langle y_1 = \text{pataDerechaDelantera}, y_2 = \text{pataizqDelantera}, y_3 = \text{pataIzqTracera}, y_4 = \text{pataDerechaTracera}, y_5 = \text{cola}, y_6 = \text{cuerpo}, y_7 = \text{cabeza} \rangle$

al nivel inferior. Asumiendo que se reciben  $L$  respuestas de reconocimiento de estos patrones, i.e., que estos descriptores (señales) se han activado (su estado paso a ser ‘True’) en los módulos del nivel donde fueron invocadas ( $X_2$ ), el siguiente paso es calcular con base en estas señales activas que módulos de ese nivel reconocen un patrón (usando los teoremas propuestos antes): como  $\Delta U_1 = 0.85$ , todas las señales están activas (ver Tabla 3.4). Entonces, aplicando el método de reconocimiento por percepción ( $DpoP$ ), usando la Ecuación 3.12, las señales que son claves para reconocer este patrón son: 1, 6, y 7. Ahora, usando la Ecuación 3.14 se puede determinar si es posible reconocer este patrón con estas percepciones. De acuerdo con el resultado ( $0.86 > 0.85$ ), se tiene un reconocimiento exitoso del patrón, y puesto que se está en el último nivel de la jerarquía  $X_2$  (el nivel donde inicio el proceso de reconocimiento), se crea una señal de salida de reconocimiento del sistema “ $S_o =$  caballo galopando”.

## 3.2 Mecanismos de Aprendizaje de Ar2p

En esta sección, se equipa a  $Ar2p$  con capacidades de aprendizaje, basado en el trabajo (Puerto y Aguilar 2016b).

### 3.2.1 Modelo de aprendizaje supervisado de Ar2p

El aprendizaje en  $Ar2p$  incluye dos mecanismos (Puerto y Aguilar 2016b): el primero, llamado *Aprendizaje nuevo*, ocurre cuando el patrón de entrada no es reconocido (i.e., no hay un módulo que lo reconozca) y crea un módulo de reconocimiento para el nuevo patrón en el modelo. El segundo mecanismo, llamado *Aprendizaje por refuerzo*, es usado para reforzar un patrón, y adapta el módulo de reconocimiento del patrón que representa a los cambios en este. Estas nuevas capacidades de aprendizaje de  $Ar2p$ , son evaluadas en varios contextos (especialmente en texto e imágenes), para analizar sus capacidades de aprendizaje y de reconocimiento (ver sección 3.3).

### 3.2.1.1 Algoritmo de Aprendizaje nuevo de Ar2p

Este mecanismo de aprendizaje ocurre cuando los patrones atómicos son reconocidos, pero no hay un módulo de reconocimiento que los reconozca como un todo. El resultado de este proceso es un nuevo módulo de reconocimiento para el patrón desconocido. El nuevo módulo es acomodado en el modelo jerárquico, en un nivel arriba de los patrones atómicos reconocidos (ver Figura 3.6).

Algoritmo de <i>Aprendizaje nuevo Ar2p</i>
Entrada: Patrón desconocido Salida: Nuevo Módulo $\Gamma$ de reconocimiento del Patrón desconocido 1. Patrón desconocido 2. IF señales atómicas reconocidas a. Establecer el nivel en la jerarquía para el nuevo patrón (Donde éste será creado) b. Crear la estructura $\Gamma$ para el nuevo módulo de reconocimiento del patrón desconocido, basado en las señales atómicas reconocidas. c. Cree el resto de los módulos necesarios en los niveles inferiores al nivel del patrón y en los niveles superiores al nivel de los patrones atómicos. d. Crear los enlaces entre los nuevos módulos 3. Fin de IF

**Figura 3.6.** Algoritmo de Aprendizaje nuevo Ar2p

La línea 1 del Algoritmo de *Aprendizaje nuevo* recibe el patrón desconocido, la línea 2 evalúa si hay un conjunto de señales atómicas reconocidas. Si es así, el algoritmo establece el nivel para el nuevo módulo en la jerarquía  $X_i$  (ver Figura 3.1), donde debe comenzar el proceso de reconocimiento de las señales atómicas (línea 2.a). Luego, en la línea 2.b, se crea la estructura específica del módulo  $\Gamma = \langle E, U, S_o \rangle$ . Posteriormente, los módulos intermedios (línea 2.c) entre el nivel del nuevo módulo para el patrón desconocido y los niveles donde son reconocidas las señales atómicas (pueden ser en diferentes niveles) son creados.

Finalmente, los enlaces entre estos nuevos módulos con los existentes son creados (línea 2.d). Si no hay señales atómicas, el algoritmo considera que el patrón está fuera del alcance de *Ar2p*, i.e., que no se tiene el conocimiento para tratar con el mismo.

### 3.2.1.2 Algoritmo de Aprendizaje por refuerzo de Ar2p.

Este mecanismo de aprendizaje ocurre cuando efectivamente hay reconocimiento, por lo cual usa como fuente de aprendizaje las señales activas reconocidas. El objetivo es actualizar la información en el módulo que refleje los cambios en el patrón de entrada, tal como por ejemplo, el peso de importancia de una señal (ver Figura 3.7).

Algoritmo de <i>Aprendizaje por refuerzo Ar2p</i> .	
Entrada: Patrón reconocido	
Salida: Módulo $\Gamma$ de reconocimiento del Patrón actualizado	
1.	Patrón reconocido
2.	Extraer información del patrón
3.	Actualizar información en el módulo
4.	Fin

**Figura 3.7.** Algoritmo de Aprendizaje por refuerzo Ar2p

A diferencia del Algoritmo de *Aprendizaje nuevo*, el algoritmo de *Aprendizaje por refuerzo* no crea un nuevo módulo, sino que actualiza la información de los descriptores, como los pesos, teniendo en cuenta las señales activas. La línea 1 recibe el patrón reconocido, la línea 2 extrae la información de las señales/descriptores presentes (i.e., cuya variable de *estado* es “True”) en el patrón de entrada reconocido. Luego, la línea 3 actualiza los pesos de importancia de las señales siguiendo el sub-programa de la Figura 3.8.

Sub-Algoritmo de <i>Aprendizaje por refuerzo Ar2p</i> .	
1.	IF $S_i$ y (estado ( $S_i$ ) = “True”) Para todo $S_i \in S$ ( $I$ ) de todos los módulos $\Gamma \in$ Patrón
1.1	$P(S_i) = P(S_i) + \alpha$ hasta alcanzar el valor máximo de 1.
2.	ELSE
2.1	$P(S_i) = P(S_i) - \alpha$ // $\alpha$ es la tasa de aprendizaje.

**Figura 3.8.** Sub-Algoritmo de Aprendizaje por refuerzo Ar2p

### 3.2.1.3 Casos con aprendizaje.

En este caso, se supone el siguiente patrón de texto  $s() = \text{"yo como carne"}$ . Además, se asume que existen en la jerarquía, los módulos para reconocer los patrones  $\Gamma = \text{"yo"}$ , y  $\Gamma' = \text{"como"}$ , pero se debe aprender el módulo de reconocimiento del patrón  $\Gamma'' = \text{"carne"}$ . Por otro lado, el peso de importante de las señales que conforman los patrones  $\Gamma$  y  $\Gamma'$  deben ser reforzadas.

Se fijan tres niveles en la jerarquía, el primer nivel  $X_1$  contiene los módulos ( $\Gamma_{j1}$ ) para el reconocimiento de los patrones atómicos de letras  $\{a, b, c, \dots, u, v, w, z\}$ ; el segundo  $X_2$  contiene los módulos ( $\Gamma_{k2}$ ) para el reconocimiento de las palabras  $\{\text{"yo"}, \text{"el"}, \text{"pollo"}, \text{"nacer"}, \text{"pescado"}, \text{"como"} \dots, \text{pero sin el módulo para reconocer "carne"}\}$ ; y el nivel  $X_3$  tiene los módulos ( $\Gamma_{l3}$ ) para el reconocimiento de patrones más complejos como oraciones  $\{\text{"yo como pollo"}, \text{"el come pescado"}, \text{"yo como carne"}, \dots\}$ .

El algoritmo recibe como entrada la sentencia  $s() = \text{"yo como carne"}$ , este la descompone en sus subcomponentes  $\langle \text{"yo"}, \text{"como"}, \text{"carne"} \rangle$  (línea 2). Luego, calcula el nivel  $X_{m=3}$  (línea 3), usando por ejemplo estrategias basadas en reglas/axiomas sintácticos/semánticas sobre la formación de las oraciones en español. Seguidamente, hace la solicitud de reconocimiento de  $y_i \langle y_1 = \text{"yo"}, y_2 = \text{"como"}, y_3 = \text{"carne"} \rangle$  (línea 4). Puesto que no son patrones del nivel  $X_{m=2}$  (ver línea 5), se envían L solicitudes (línea 25) de reconocimiento. Para cada  $y_i$  en L inicia un proceso de reconocimiento recursivo. Específicamente se cargan tres procesos recursivos de reconocimiento: uno para la palabra "yo", un segundo para la palabra "como" y un tercer proceso para "carne".

Aprendizaje por refuerzo a partir del patrón reconocido "yo", considerando una tasa de aprendizaje  $\alpha = 0.1$ . Para este primer llamado recursivo,  $y_1 = s() = \text{"yo"}$ , un nuevo proceso de reconocimiento es abierto. Si suponemos que el patrón de entrada es reconocido (línea 28), el algoritmo realiza el método de aprendizaje por refuerzo (línea 29), usando las señales (patrones)

presentes que permitieron el reconocimiento del patrón (se asume que el estado de “y” y “o” es “True”), por lo cual se actualizan sus pesos:

- $P(\text{“y”}) = P(\text{“y”}) + 0.1$
- $P(\text{“o”}) = P(\text{“o”}) + 0.1$

De esta manera, los pesos de importancia de sus patrones internos son actualizados. Esto es similar para el patrón “como”. Hecho esto, el algoritmo sigue el reconocimiento, enviando la señal  $S_o$  (señal de salida de reconocimiento de (y)) al nivel superior. Cuando en este nivel superior se reconoce un patrón (por ejemplo, una *oración*) se realiza un proceso de aprendizaje por refuerzo análogo, para actualizar el peso de los patrones de la oración reconocida: por ejemplo, “yo”, “como” y “carne”.

*Aprendizaje de un nuevo módulo.* Continuando con el escenario del caso de aprendizaje, a continuación, se describe el aprendizaje de un nuevo módulo de reconocimiento para el patrón “carne”. Recapitulando, el nivel 2 tiene los módulos de reconocimiento  $\Gamma_{k2}$  para las palabras “yo”, “pollo”, “nacer”, etc., menos un módulo para reconocer la palabra “carne”. Cuando el patrón de entrada “carne” es recibido, el algoritmo descompone este en L solicitudes hacia los niveles inferiores, y espera por el reconocimiento de cada uno (línea 26).

Como no hay un módulo que interprete o reciba todas estas señales, el patrón “carne” no es posible reconocer. Ahora bien, como todos los sub-patrones que forman el patrón desconocido “carne” fueron reconocidos (línea 37), el algoritmo aprende un nuevo módulo (línea 38). Dependiendo de las señales atómicas reconocidas del patrón “carne”, es definido el nivel de la jerarquía para el nuevo módulo (donde éste será creado). Por ser una palabra, el módulo correspondiente debe estar en el nivel de las palabras, esto es, el nivel  $X_2$ . El algoritmo procede a crear la estructura E (ver sección 3.1.1.1) para el nuevo módulo de reconocimiento responsable del

patrón “carne”, usando las señales atómicas reconocidas en el nivel inferior (los valores de los pesos de importancia  $P$  son definidos aleatoriamente).

El módulo es situado entre el nivel del patrón inicial  $s()=$ ”yo como carne” y el nivel de los patrones atómicos. Por otro lado, los enlaces entre los módulos son creados. Es de notar que el módulo “nacer” también tiene activas las señales presentes, por cuanto comparte las mismas letras del patrón “carne”. Pero este módulo de reconocimiento no es disparado como salida del sistema, o su patrón es inhibido, puesto que no cumple con la forma u orden presente con el patrón de entrada  $s()=$ ”carne”. Así, una vez recibidas las respuestas de reconocimiento de los patrones “yo”, “como”, “carne” desde  $X_2$  (Este último por el método de *Aprendizaje\_nuevo*), el algoritmo procede a calcular el reconocimiento de ( $y$  inicial), y dado que el reconocimiento es exitoso, es decir, hay un módulo de oración en el nivel  $X_3$  que da cuenta de ello, el algoritmo realiza el aprendizaje por refuerzo para ese último nivel de la jerarquía (en ese módulo de oración en específico), para terminar enviando la señal de salida  $S_0$  de reconocimiento del sistema.

### **3.3 Aplicaciones de Ar2p en Diversos Dominios**

En esta sección. se describen varias aplicaciones de *Ar2p*: para el reconocimiento de estilos de conducción (Aguilar, J., Aguilar, K., Chávez, Cordero y Puerto 2017; Aguilar, J., Aguilar, K., Chávez, Cordero y Puerto 2019), emociones en robots (Gil, Aguilar, Puerto y Dapena 2018; Gil, Aguilar, Puerto y Dapena 2019), y texto (Puerto y Aguilar 2019).

#### **3.3.1 Reconocimiento de estilos de conducción usando Ar2p**

Con la popularidad de los Sistemas Avanzados de Asistencia a la Conducción (ADAS, por sus siglas en inglés), el problema de interacción entre el conductor y los ADAS es importante, pero más importante es cómo adaptar estos mecanismos de asistencia a las características del conductor.

Para hacer que los ADAS puedan adaptarse a las necesidades de cada conductor, es necesario que ellos cuenten con sistemas adaptativos a través de los cuales puedan considerar/detectar y/o reconocer las características internas de cada conductor, como la fatiga, la falta de atención, y en este caso, su estilo de conducción (Lin et al 2014).

Existe una gran cantidad de trabajos en la literatura sobre el manejo de las emociones en contextos de conducción, por ejemplo, en (Cordero y Aguilar 2016) es propuesto un modelo de reconocimiento del estado emocional del conductor, utilizando crónicas y patrones estáticos. Por otro lado, en (Eyben et al 2010) muestran cómo las emociones son claves no solo en la interacción humano-computador, sino también en la comunicación en el vehículo. (Katsis et al 2015) presenta una revisión de los trabajos en el reconocimiento de las emociones, centrándose en aquellas emociones que más influyen en el rendimiento o manejo del conductor. En (Aypar et al 2014; Barkana y Masazade 2017) se enfocan en un mecanismo de alerta al conductor basado en el reconocimiento del estado del conductor. (Guoying y Danpan 2016) proponen un enfoque de reconocimiento de patrones para identificar el manejo o estilo del conductor. Hay mucho más trabajo sobre las emociones de los conductores de vehículos, pero en general, proponen modelos sencillos, o sólo estudian las emociones (Kolli et al 2011; Tawari y Trivedi 2010; Paschero et al 2012; Wang et al 2013).

En esta sección, se aplica *Ar2p* para el reconocimiento de estilos de conducción en un patrón jerárquico de estilos de conducción, compuesto de tres niveles (ver Sección 3.3.1.1.): un primer nivel para reconocer los estados emocionales, un segundo nivel para reconocer estados del conductor, y finalmente, el tercer y último nivel para reconocer estilos de conducción (Aguilar J, Aguilar K, Chavez y Puerto 2017). En esta propuesta por niveles, cada nivel está compuesto de diferentes descriptores, por lo cual se usa un enfoque multi-modal para ser percibidos. Los

descriptores se relacionan entre sí, por cuanto ellos son descriptores de ellos mismos, i.e., descriptores de niveles más bajos describen o constituyen descriptores de más alto nivel. En esta aplicación, *Ar2p* es comparado con un modelo basado en lógica difusa y otro modelo basado en crónicas (Aguilar J, Aguilar K, Chávez y Puerto 2017).

### 3.3.1.1 Definición formal del patrón de estilo de conducción de un conductor de vehículo.

En esa sección, se propone modelar el estilo de conducción como una jerarquía de patrones, la cual está compuesta de tres niveles (Aguilar J, Aguilar K, Chávez y Puerto 2017).

*Definición de patrón de estilos de conducción* (primer nivel). Su objetivo es modelar como los conductores manejan. En la literatura, normalmente se clasifica estos estilos de conducción en cuatro (4) tipos o categorías: *agresivo, ecológico, urbano, y normal*. Este nivel debe detectar el estilo de conducción basado en los descriptores de la Tabla 3.5.

**Tabla 3.5.** Descriptores del Patrón de estilos de conducción

Descriptor	Descripción
Tipo de Carretera	Representa el tipo de carretera, si es rural o urbana.
Estado del Conductor	Describe el estado del conductor, el cual es definido a su vez, por el segundo nivel de la jerarquía de patrones (ver la Tabla 3.6).
Emoción del Conductor	Define el estado emocional del conductor, el cual es definido por el tercer nivel de la jerarquía de patrones (ver la Tabla 3.7).
Condiciones del Ambiente	Caracteriza las condiciones actuales del ambiente, por ejemplo, lloviendo, si es un día soleado, entre otros.
Estado de la carretera	Caracteriza las condiciones actuales de la carretera: si esta pavimentada, si tiene huecos, etc.
Características del Tráfico	Define aspectos relacionados con las leyes de tránsito, y otras características de la carretera, por ejemplo, límites de velocidad, señales de tráfico, entre otras.

*Estados del Conductor* (segundo nivel). Su objetivo es modelar el estado del conductor del vehículo. En la literatura, normalmente, el estado del conductor es clasificado en distintas

categorías: despierto (o vigilante), estresado, adormecido (o letárgico), agradable (o amable, simpático), fatigado, tranquilo (o calmado), aburrido, dormido, entre otros. Este nivel debe detectar el estado del conductor basado en los descriptores de la Tabla 3.6.

**Tabla 3.6.** Descriptores del Patrón de estados del conductor

Descriptor	Descripción
Clase de Vehículo	Describe el tipo de vehículo: si es automóvil, de cargar, de pasajeros, minivan, etc.
Acción de control sobre el vehículo	Describe la acción actual del conductor del vehículo. Por ejemplo, si está acelerando, frenando, etc.
Emoción del Conductor	Define el estado emocional del conductor (ver la Tabla 3.7).
Condición del vehículo	Describe las condiciones actuales del vehículo, si tiene una falla mecánica, eléctrica, si tiene combustible, entre otras
Característica del Conductor	Define la edad del conductor o condiciones físicas, si es una persona joven, si es un adulto mayor, si tiene limitaciones físicas, entre otras.
Experiencia de Conducción	Caracteriza la experiencia del conductor si tiene poca experiencia, mediana o una mayor experiencia.
Hora de Manejo	Define la hora actual del día, por ejemplo, si es de mañana, tarde, o noche.

*Emociones del conductor* (tercer nivel). Su objetivo es modelar las emociones del conductor. Este nivel debe detectar la emoción actual del conductor. Particularmente, se usan seis emociones básicas definidas en la literatura: *felicidad, tristeza, miedo, ira, disgusto, y sorpresa*. Los descriptores que definen este patrón, se describen en la Tabla 3.7.

El principal objetivo de la jerarquía de patrones antes descrita, es reconocer el estilo de conducción. Para reconocer el estilo de conducción se necesitan varios descriptores (ver Tabla 3.5), los cuales lo describen. Particularmente, uno de los descriptores es el estado de conducción, el cual a su vez es descrito por un conjunto de descriptores (ver la Tabla 3.6).

**Tabla 3.7.** Descriptores del Patrón de las emociones del conductor

Descriptor	Descripción
Comportamiento del conductor.	Define el comportamiento actual del conductor dentro del vehículo, por ejemplo, el conductor tira la puerta, usa el cinturón de seguridad, etc.
Acción control sobre el vehículo	Ver descripción de la Tabla 3.6.
Comportamiento fisiológico del conductor	Define las condiciones fisiológicas actuales del conductor. Por ejemplo, el ritmo cardiaco, presión sanguínea, color de la cara, etc.
Condición del vehículo	Ver descripción de la Tabla 3.6.
Expresiones de voz del conductor	Caracteriza el tono de voz actual del conductor, por ejemplo, si el conductor está gritando, cantando, o hablando normalmente, etc.
Expresiones faciales del conductor	Este describe las expresiones faciales actuales del conductor, por ejemplo, si el conductor está durmiendo, o esta serio.
Expresiones corporales del conductor	Describe las expresiones corporales actuales del conductor, para ello es necesario usar un lenguaje corporal.

Otro descriptor es el estado emocional del conductor, el cual también es descrito por otro conjunto de descriptores (ver Tabla 3.7). Así, cada nivel tiene un conjunto diferente de descriptores, los cuales son percibidos de diferentes maneras (sonido, visión, etc.), lo que implica el uso de un enfoque multi-modal para su percepción. Los descriptores caracterizan varios aspectos; facial, acústico, lenguaje corporal, entre otros. El estado actual de los descriptores es determinado por los eventos que son capturados en el entorno del vehículo en un momento dado. Por eso, nosotros usamos información desde diferentes sensores en el carro, para caracterizar estos eventos. Por ejemplo, para la velocidad del carro, nosotros podemos definir el conjunto de eventos de la Tabla 3.8. Y así para el resto de descriptores de nuestro modelo multi-modal jerárquico.

**Tabla 3.8.** Eventos entorno a la velocidad del carro

Id del evento	Descripción	Velocidad
S1	Alta Velocidad	> 100 km/h (mayor a 100 Kilómetros por hora)
S2	Normal Velocidad	≥40 y ≤ 99 km/h (mayor o igual a 40 kilómetros por hora y menor o igual a 99 kilómetros por hora)
S3	Baja Velocidad	< 40 km/h (menor a 4° kilómetros por hora)

Ahora, de acuerdo con los valores actuales de los descriptores, es determinada la emoción actual del conductor, el estado actual del conductor, y finalmente, su estilo de conducción, usando el modelo multi-modal jerárquico. La Tabla 3.9 muestra un ejemplo de las posibles emociones reconocidas por el Patrón de tercer nivel de nuestro modelo multi-modal Jerárquico, de acuerdo con los valores de los descriptores de este patrón.

**Tabla 3.9.** Emoción del conductor

Id Evento	Emoción	Comportamiento del conductor	Acción control sobre el vehículo	Comportamiento fisiológico del conductor	Condición del vehículo	Expresión facial del conductor	Expresión corporal del conductor
ED1*	Ira	El conductor tira la puerta	hace presión sobre el volante	ritmo cardíaco alto, dilatación de las pupilas.	Falla mecánica o eléctricas	El conductor es serio	El conductor se mueve violentamente
ED2	Felicidad	El conductor usa el cinturón de seguridad	normal	Ritmo normal del corazón	normal	El conductor está sonriendo	El conductor reacciona calmado
ED3	Felicidad	El conductor usa el cinturón de seguridad, El conductor es calmado	normal	Ritmo normal del corazón	normal	El conductor tiene una cara calmada	La postura del conductor es calmada
ED4	Miedo	El conductor usa el cinturón de seguridad	frena	Color pálido de la cara.	Cualquiera	El conductor esta serio	

\*EDn significa el evento de la emoción del conductor n, en ingles Emotion of the Driver n.

También, ellos muestran algunas de las posibles combinaciones de los valores de los descriptores para el reconocimiento de estas emociones y estilos (e.g., la Tabla 3.9 muestra dos ejemplos de eventos (ED2 y ED3) para reconocer la emoción de “felicidad”) pero puede haber

más combinaciones de valores de los descriptores para reconocerla. Para los posibles estados del conductor (segundo nivel de nuestro modelo multi-modal jerárquico), esto es similar.

La Tabla 3.10 muestra un ejemplo de los posibles estilos de conducción, reconocidos por el Patrón de primer nivel de la jerarquía de nuestro modelo multi-modal jerárquico, de acuerdo con los valores de los descriptores en ese momento para ese patrón. Es importante resaltar que las Tablas 3.9 y 3.10 muestran algunas de las emociones y estilos que pueden ser reconocidos por *Ar2p* (ver sección 3.3.1.2).

**Tabla 3.10.** Estilos de conducción

Id Evento	Estilo del conductor	Tipo de carretera	Estado del conductor	Emoción del conductor	Condiciones del entorno	Estado de la carretera	Características de tráfico
SD1*	Agresivo	Cualquiera	Estresado	Ira	Lloviendo	Tiene huecos	Cualquiera
SD2	Ecológico	Rural	Relajado	Felicidad	Cualquiera	Cualquiera	Respetar los límites de vel.
SD3	Normal	Urbana	Relajado	Felicidad	Cualquiera	Cualquiera	Cualquiera

SDn\* significa el evento de estilo del conductor n, en inglés Style of the Driver n.

En la Tabla 3.10, cada estilo de conducción está asociado con diferentes valores de descriptores (Aguilar J, Aguilar K, Chávez y Puerto 2017). Por ejemplo, el estilo de conducción agresivo está asociado con el estrés (estado), y la ira (emoción). Adicionalmente, las condiciones del entorno de lluvia y una carretera con huecos definen también este estilo de conducción. El estilo de conducción Ecológico, está asociado con el estar relajado (estado) y estar contento (emoción). Así, esa tabla muestra algunos ejemplos de estilos de conducción. Un estilo de conducción puede tener diversos patrones (conjunto de valores de descriptores) y es posible agregar nuevos estilos de conducción usando estos descriptores. En la Tabla 3.11, la cual es usada como base para la prueba de *Ar2p* en el contexto de las emociones (ver sección 3.3.1.2), los descriptores han sido definidos como códigos, con el fin de simplificar la información representada en ellos.

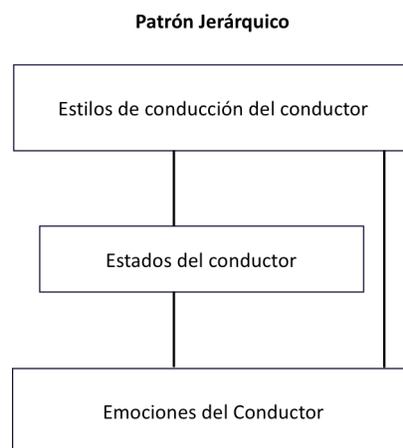
**Tabla 3.11.** Vista conceptual de la capa de emociones

Descriptor	Código	Ejemplo de Descriptor
Comportamiento del conductor	<b>XY</b> X= representa la mirada (X =1, mira fuera de la carretera; X=2, mira la carretera) Y= representa las manos en el volante (Y=1, ambas manos sobre el volante; Y=2, sólo la izquierda; Y=3, sólo la derecha; Y=4, ninguna)	21
Comportamiento o fisiológico del conductor	<b>XXXYYZZZZW</b> X = representa la temperatura corporal, Y = representa la frecuencia cardiaca, Z= representa la presión sanguínea, W= representa el nivel de alcohol	96370112800
Condición del vehículo	<b>X</b> X =representa las condiciones de los neumáticos (X=1 nuevas (<=10.000 km de uso), X=2 desgastadas (entre 10.000 y 50.000 km de uso) X=3 malas (>50.000 km de uso)	1
Acciones de control sobre el vehículo	<b>XYYYYZ</b> X = representa la luz de freno (X=1 Luz de freno encendida, X= 2 luz de freno apagada, X=3, cualquiera) Y = representa la velocidad de GPS Z = presenta el uso del pito (Z=1 uso normal del pito; Z=2 uso excesivo del pito).	31001
Expresiones faciales del conductor	<b>X</b> X = Representa el valor de la emoción de la cara (X=1, cara neutral, X=2, cara normal, X=3, cara sorprendida, X=4 cara seria, X=5 cara con una gran sonrisa, X=6 cara con una pequeña sonrisa, X=7, cara enojada, X=8, cara de repugnancia, X=9 cara con sonrisa de decepción)	1
Expresiones de voz del conductor	<b>X</b> X =Representa el valor de la emoción de la voz (X=1, voz seca y fuerte; X=2, voz suave y baja; X=3, risa; X=4, grito seco; X=5, voz neutral).	2
Emoción del conductor	<b>X</b> X representa el estado emocional del conductor (X=1, Felicidad; X=2, Sorpresa; X=3, Ira, X=4, Miedo, X=5, Tristeza)	1

### 3.3.1.2 Aplicación de Ar2p para el reconocimiento de estilos de conducción.

Con base en lo anterior, en esta sección se aplica *Ar2p*, para reconocer los estilos de conducción (Aguilar J, Aguilar K, Chávez y Puerto 2017). Específicamente, el objetivo es mostrar las capacidades de *Ar2p* para modelar el complejo patrón de estilos de conducción, donde no toda la información está disponible o no es correcta en un momento dado, ya sea por fallas de comunicación, fallas de sensores, etc. *Ar2p* es comparado con los paradigmas, basado en crónicas, y basado en la lógica difusa, seleccionados por su capacidad para procesar información parcial y ambigua.

A continuación, se describe cómo poder modelar el sistema multi-modal jerárquico usando *Ar2p*: Lo primero es establecer la base de conocimiento de *Ar2p*, que consiste en crear o instanciar los módulos de reconocimiento de patrones a partir de los tres niveles definidos en el Patrón jerárquico para estilos de conducción, que se esquematiza en la siguiente Figura 3.9.



**Figura 3.9.** Modelo del Patrón jerárquico para estilos de conducción

De acuerdo a la arquitectura jerárquica de *Ar2p* (ver Figura 3.1), la instanciación del modelo multi-modal jerárquico para reconocer los estilos de conducción (ver Figura 3.9) viene dada como sigue:

Un primer nivel  $X_1$  en la arquitectura de *Ar2p* tiene los módulos de reconocimiento de los

descriptores atómicos que constituyen las emociones, los estados y los estilos de conducción, considerados en los tres niveles del modelo multi-modal jerárquico definido (ver sección 3.3.1.1 y Figura 3.9). El segundo nivel  $X_2$  en la arquitectura de  $Ar2p$  tiene los módulos de reconocimiento de emociones consideradas en el tercer nivel del modelo multi-modal jerárquico (ver sección 3.3.1.1, ítem tercer nivel, representado en la Figura 3.9, cuadro *Emociones del conductor*). Es decir, las emociones que son consideradas en el Tercer nivel del modelo multi-modal jerárquico son reconocidas en el segundo nivel del modelo jerárquico de  $Ar2p$ . El tercer nivel  $X_3$  de la arquitectura de  $Ar2p$  tiene los módulos de reconocimiento de los estados del segundo nivel del modelo multi-modal jerárquico definido (ver sección 3.3.1.1, ítem segundo nivel, representado en la Figura 3.9, cuadro *Estados del conductor*). El cuarto y último nivel  $X_4$  de la arquitectura de  $Ar2p$  tiene los módulos de reconocimiento de estilos de conducción considerados en el primer nivel del modelo multi-modal jerárquico definido (ver sección 3.3.1.1, ítem primer nivel, representado en la Figura 3.9, cuadro *Estilos de conducción*).

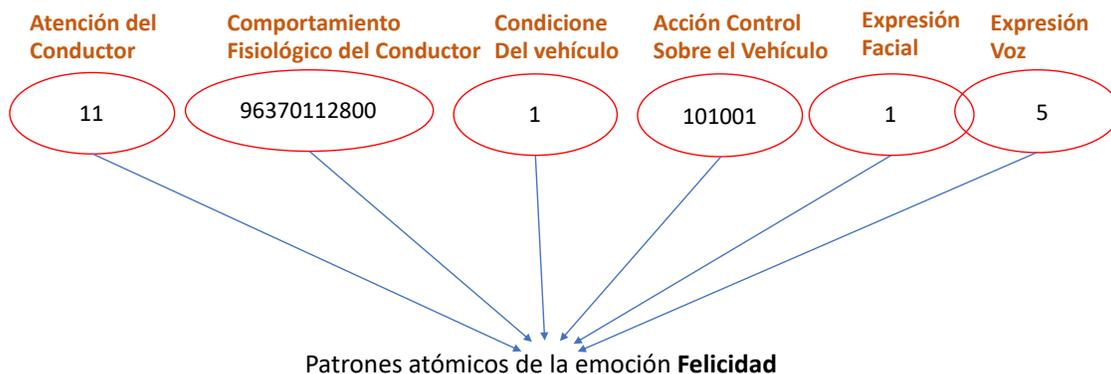
Ahora, el siguiente paso es definir el enfoque de  $Ar2p$  a utilizar, esto es, si se reconocerá utilizando el paradigma de patrones estáticos o dinámicos. Esto depende de la naturaleza y el dominio del problema. Por ejemplo, cuando se tiene una “imagen” o conjunto de valores de los descriptores en un instante, se trataría con patrones estáticos (ver sección 3.1.1); por otra parte, cuando se tiene un “video” o conjunto de valores de los descriptores en un intervalo de tiempo, como si fuese un seguimiento o persecución, se trataría con el paradigma de patrones dinámicos (ver sección 3.1.2). Asumiendo que el problema consiste en reconocer el estilo de conducción en un instante dado, la lógica de resolución con  $Ar2p$  usando el paradigma de resolución por patrones estáticos sería como sigue:

- Cada nivel tiene los módulos de reconocimiento de patrones que lo constituyen según la jerarquía. Por ejemplo, la Tabla 3.12, define la estructura de módulo de reconocimiento para la *felicidad*, que haría parte del nivel de reconocimiento de las emociones. Similares estructuras se definen para las otras emociones, estados y estilos de conducción.

**Tabla 3.12.** Módulo de reconocimiento del patrón felicidad

E				
S		C		
Señal	Estado	Descriptor(D)	Dominio	Peso
1	F	Atención del conductor	<11>	0.8
2	F	Comportamiento fisiológico del Conductor	<96370112800>	0.6
3	F	Condiciones del vehículo	<1>	0.7
4	F	Acción Control sobre el vehículo	<101001>	0.6
5	F	Expresión facial del conductor	<1>	0.9
6	F	Expresión de voz del conductor	<5>	0.9
$\Delta U$				

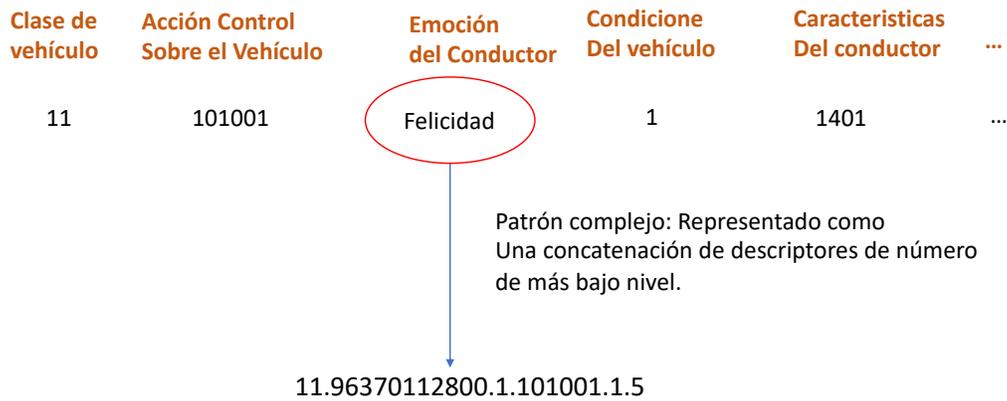
- Los descriptores son vistos como una cadena de valores numéricos que lo representa (ver la Figura 3.10, para el caso de la emoción de felicidad)



**Figura 3.10.** Modelo esquemático de Ar2p para la capa de emociones (representa felicidad)

En la Figura 3.10, los descriptores atómicos están en la parte superior (primera capa, marcados en círculos rojos), y caracterizan las condiciones del entorno, del vehículo y del conductor, consideradas para el reconocimiento de las emociones. Para este caso, la emoción de felicidad estaría representada por el patrón “11 96370112800 1 101001 1 5”.

- La capa para reconocer los estados de conducción está compuesta de los patrones de emociones y de sus propios descriptores atómicos, i.e., tiene descriptores compuestos y otros de más bajo nivel (ver la Figura 3.11 para el caso del estado de *relajado*).



**Figura 3.11.** Modelo esquemático de Ar2p para la capa de Estados (representa relajado)

En este ejemplo, el estado relajado de conducción está compuesto por el patrón de emoción (“felicidad”), cuyo patrón es “11.96370112800.1.101001.1.5”, que se describió arriba (ver Figura 3.10), y otros descriptores atómicos propios, como clase de vehículo, condición del vehículo, etc.

- La capa para reconocer los estilos de conducción está compuesta de los patrones de estado, emociones, y de sus descriptores atómicos propios.

Establecido el escenario y los supuestos de representación e interpretación de los datos/información, resta ahora, hacer la traza o seguimiento del algoritmo (ver sección 3.1.1.3 modelo general computacional de *Ar2p*) para un patrón complejo de entrada: Sea  $y = s()$  = el conjunto de valores de los descriptores dados en las Figuras 3.10 y 3.11, junto con los siguientes descriptores atómicos: *Tipo de carretera=Urbana*, *Condiciones del entorno=Día Soleado*, *Estado de la carretera=Buena*, *Características de tráfico=Respetar señales de tránsito*, que constituyen el patrón de entrada complejo (Línea 1 del algoritmo). El siguiente paso consiste en descomponer

este patrón en los sub-patrones de análisis (línea 2). En este caso, se tienen tres sub-patrones: el patrón de los descriptores de la Figura 3.10 (*emoción*), el patrón de los descriptores de la Figura 3.11 (*estado*), y los descriptores atómicos: tipo de carretera, condiciones del entorno, etc. El proceso de descomposición particular se logra con métodos supervisados (por ejemplo, con etiquetas para conjunto de descriptores, o por contendores como arreglos, listas, etc.). Una vez se tiene las partes del patrón de entrada particionadas, se determina el espacio de reconocimiento principal  $X_i$  en la arquitectura propuesta de *Ar2p* (ver Figura 3.1), a partir del cual continuar con el proceso de reconocimiento. En esta prueba, el nivel de inicio de reconocimiento es el 4,  $X_{i=4}$ . Después, se procede a crear L solicitudes de petición de reconocimiento de las partes que conforman al vehículo o patrón y (línea 4). Así, dado que el nivel actual del proceso es  $X_{i=4}$ , para profundizar el proceso de análisis del patrón se envían L solicitudes de petición de reconocimiento de cada una de las partes del patrón de entrada ( $y$ ) a los niveles inferiores (línea 25). Esta línea invoca el proceso recursivo, generándose nuevos procesos de reconocimiento en paralelo e independientes, para cada uno de los sub-patrones. De esta manera, el *proceso principal* invoca la recursividad. A continuación, se inician los nuevos procesos de reconocimiento invocados en la línea 25 por parte del *proceso principal*, para el respectivo reconocimiento de las partes contenidas en el patrón original, por ejemplo: L1=sub-patrón de Figura 3.10, L2= sub-patrones de la figura 3.11, L3= Sub-patrón atómico de carretera=Urbana, L4= Sub-patrón atómico de Condiciones del entorno=Día Soleado, etc.

Veamos el caso del reconocimiento de L1. Ahora  $y=s()$ = patrón de la Figura 3.10. El siguiente paso (línea 2), consiste en descomponer este patrón en los sub-patrones de análisis, *Atención de Conductor*, *Comportamiento fisiológico*, *Condición del vehículo*, *Acción de control*, *Expresión facial*, *Expresión de voz*. Una vez se tiene las partes del patrón de entrada particionadas, se

determina el espacio de reconocimiento principal  $X_i$  para este *sub-proceso principal*, el nivel de inicio de reconocimiento es el 2,  $X_{i=2}$  (capa de emociones). Después, se procede a crear L solicitudes de petición de reconocimiento de las partes que conforman a la emoción (línea 4). Así, dado que el nivel actual del proceso es  $X_{i=3}$ , para profundizar el proceso de análisis del patrón se envían L solicitudes de petición de reconocimiento de cada una de las partes del patrón de entrada ( $y$ ) a los niveles inferiores (línea 25). Esta línea invoca el proceso recursivo, generándose nuevos procesos de reconocimiento en paralelo e independientes, para cada uno de los sub-patrones. De esta manera, el *proceso principal* igualmente invoca la recursividad. A continuación, se inician los nuevos procesos de reconocimiento invocados en la línea 25 por parte del *Sub\_proceso\_principal*, para el respectivo reconocimiento de las partes contenidas en el patrón original, por ejemplo: L1= *Atención de Conductor*, L2=Comportamiento fisiológico, L3=Condición del vehículo, L4= Acción de control y L5= Expresión facial, Expresión de voz.

Veamos el caso interno del reconocimiento  $y=s()$ =*Atención de Conductor* (línea 1). El siguiente paso (línea 2), consiste en descomponer la sub-imagen del vehículo en los sub-patrones de análisis, en este caso el conjunto de sub-patrones de análisis es vacío. Por cuanto es un patrón atómico, se determina que el espacio de reconocimiento  $X_i$  es el nivel 1,  $X_{i=1}$ . Dado que el nivel actual del proceso es  $X_{i=1}$ , se cumple la condición para realizar el cuerpo de instrucciones entre las líneas 6 y 23. Se procede entonces a realizar el reconocimiento del patrón *Atención del Conductor* ( $y$ ) (línea 6), que consiste en mapear dicho patrón con la base de datos de patrones atómicos que constituye el espacio de reconocimiento  $X_{i=1}$ . Damos por hecho que el mapeo es exitoso (línea 7), y se procede a aprender (en este ejemplo, no se describe el aprendizaje, ver sección 3.2 para más detalles).

Se mira si el proceso actual está en el último nivel de la jerarquía  $X_m$ . Como no se ha alcanzado el ultimo nivel (que es  $X_{m=4}$ ), se pasa al ELSE del condicional (línea 11), que consiste en crear la

salida  $S_0$ , y enviar dicha salida  $S_0$  de reconocimiento del patrón *Atención del Conductor* ( $y$ ) al nivel superior (difusión de una señal de reconocimiento positivo (True) al *proceso principal*). En este punto se termina este proceso interno, que fue invocado en la Línea 25, donde quedo el proceso principal. Este mismo proceso se aplica para los demás descriptores del sub-patrón en cuestión, a saber, L2=Comportamiento fisiológico, L3=Condición del vehículo, L4=Acción de control y L5= Expresión facial, Expresión de voz. Si se asume que hubo reconocimiento de todos los descriptores del patrón de la Figura 3.10, el *proceso principal* retoma el control del programa donde se hizo la solicitud de los L reconocimientos (línea 25), y cambia, por ejemplo, el campo de estado de la estructura de datos de la Tabla 3.12 que lo contienen, con las respuestas de reconocimiento recibidas. En este caso, la columna de estados de la Tabla 3.12 cambia a “True” (las 6 señales que corresponden a los sub-patrones del proceso de descomposición del patrón felicidad han sido reconocidas). En este experimento se ha asignado el peso de importancia de acuerdo a su influencia para el reconocimiento. Por ejemplo, los descriptores de las señales 5 y 6 tienen mayor peso (0.9), pues son claves para el reconocimiento de las emociones.

Continuando con el proceso de reconocimiento del *proceso principal*, en la línea 26 se reciben las 6 respuestas de los niveles inferiores de la solicitud L. Una vez recibidas estas señales en el módulo de reconocimiento de  $X_2$  desde donde se invocó la solicitud de reconocimiento, se procede a calcular el reconocimiento de los patrones de los módulos que las contienen (línea 27). En este caso, el módulo de interés que vamos a analizar es el que entra en reconocimiento del patrón de la emoción de *felicidad*. Para calcular el reconocimiento, se usan las Ecuaciones 3.1, 3.2 y 3.3 de los teoremas 1 y 2. El Teorema 1 es para el reconocimiento del patrón por señales claves. En este caso, se ha definido el umbral  $\Delta U_1 = 0.75$ . Aplicando la Ecuación (3.1), las señales claves son:  $= \{i=1,2,5,6\}$ . Aplicando la Ecuación (3.2) para verificar el Teorema 1, se obtiene que  $0.87 > 0.75$ .

Por tanto, hay reconocimiento por señales claves. Así, no se aplica el Teorema 2. Paralelo a esto y con similar análisis, ocurre con los descriptores L2= sub-patrones de la Figura 3.11, L3= Sub-patrón atómico de carretera=Urbana, L4= Sub-patrón atómico de condiciones del entorno=Día Soleado, etc.

Si se asume, además, que hubo reconocimiento de los descriptores del patrón de la Figura 3.10 (como acabamos de ver arriba), y también de la Figura 3.11 y de los descriptores atómicos, el *proceso principal* retoma el control del programa donde se hizo la solicitud de los L reconocimientos (línea 25), y cambia el campo de estado de la estructura de datos que lo contienen, como se muestra para la Tabla 3.13 del módulo de reconocimiento del patrón *Normal*, que tiene algunas señales reconocidas (estado="True"), producto de las solicitudes de reconocimiento respectivas.

**Tabla 3.13.** Módulo de reconocimiento del patrón de estilo de conducción Normal

E				
S		C		
Señal	Estado	Descriptor(D)	Dominio	Peso
1	True	Tipo_Carretera	<1:rural, 2:urbana>	0.6
2	True	Estado_Conductor	<96370112800:Rejalado,...>	1
3	True	Emoción_Conductor	<1: Felicidad,2: tristes,..>	1
4	True	Condiciones_Ambiente	<1: Soleado, 2: lloviendo>	0.6
5	False	Estado_Carretera	<1:Buena, 2: huecos>	0.6
6	False	Características_Tráfico	<5:respeta señales, 1:cruce prohibido,...>	0.6
$\Delta U$				

Las 6 señales de la Tabla 3.13 corresponden a los sub-patrones del proceso de descomposición del patrón de estilo de conducción normal. Es de notar que en este experimento se ha asignado el peso de importancia de acuerdo a su influencia para el reconocimiento. Por ejemplo, los descriptores de las señales 2 y 3 tienen mayor peso (1.0), pues son claves para el reconocimiento del estilo de conducción.

Continuando con el proceso de reconocimiento del *proceso principal*, en la línea 26 se reciben las 6 respuestas de los niveles inferiores de la solicitud L. Una vez recibidas estas señales donde se invocó la solicitud de reconocimiento ( $X_4$ ), se procede a calcular el reconocimiento de los patrones de los módulos que las contienen (línea 27). En este caso, se supone que el módulo de interés a analizar, y que entra en reconocimiento, es el del patrón del estilo de conducción *Normal* (ver Tabla 3.13). Para calcular el reconocimiento, se aplican las Ecuaciones 3.1, 3.2 y 3.3 de los Teoremas 1 y 2. El Teorema 1 es para el reconocimiento del patrón por señales claves. En este caso, se ha definido el umbral  $\Delta U_1 = 0.85$ . Aplicando la Ecuación (3.1), las señales claves son:  $= \{i=1,2\}$ . Aplicando la Ecuación (3.2) para verificar el Teorema 1, se obtiene que  $1 > 0.85$ . Por tanto, hay reconocimiento por señales claves.

Después de eso, el proceso principal pasa a chequear el resultado del cálculo (línea 29), y si el cálculo es exitoso por cualquier Teorema, se procede a ejecutar el aprendizaje (ver sección 3.2). Se chequea si el proceso de reconocimiento del patrón vehículo está en el último nivel de la jerarquía (Línea 30). En efecto, actualmente el proceso de reconocimiento está en  $X_{m=4}$ , que es el último nivel de la jerarquía, y coincide con el nivel de reconocimiento inicial. Por tanto, se crea la señal de salida  $S_o$  de reconocimiento  $y=s()$  = “*patrón de estado de conducción Normal*”.

Por otra parte, si se definen los estilos de conducción como patrones dinámicos, se usaría el formalismo definido de *Ar2p* para el reconocimiento de patrones dinámicos (ver sección 3.1.2) considerando los siguientes cambios:

- Los módulos de reconocimiento de los patrones cambian su dominio, incluyendo las series de tiempo (ver Tabla 3.10 y 3.11)
- Las ecuación y teoremas para el reconocimiento cambian, y se usa las definidas para el paradigma de reconocimiento por patrones dinámicos (Ecuaciones 3.13, 3.14, 3.15 y 3.16)

Dos posibles estructura de módulos de reconocimiento, una para el estado de “Agresividad” y otra para la emoción “Ira”, bajo este enfoque dinámico, se pueden ver en las Tablas 3.14 y 3.15, respectivamente.

**Tabla 3.14.** Estructura del módulo de reconocimiento de patrón  $\Gamma\rho_d = \text{“Agresividad”}$

E				
S		C		
Señal	Estado	Descriptor	Dominio	Peso de Importancia
1	F	Carretera rural	Ap_carretera_rural <sup>1</sup>	0.5
2	F	Carretera urbana	Ap_carretera_urbana <sup>2</sup>	0.6
3	F	Estresado	Ap_estresado <sub>1</sub>	0.8
4	F	Ira	Ap_ira <sub>2</sub>	0.8
5	F	Lluvia	Ap_Lluvia <sup>3</sup>	0.5
6	F	Carretera mala	Ap_carretera_mala <sup>4</sup>	0.6
7	F	No respeta señales de Transito	Ap_no_s_trafico <sup>5</sup>	0.8
Umbral: $\Delta U1, \Delta U2$				

Este módulo se caracteriza por tener 5 descriptores por percepción, los cuales están definidos en sus respectivos vectores: Ap\_carretera\_rural<sup>1</sup>, Ap\_carretera\_urbana<sup>2</sup>, Ap\_Lluvia<sup>3</sup>, Ap\_carretera\_mala<sup>4</sup>, Ap\_no\_s\_trafico<sup>5</sup>, y dos descriptores por características, Ap\_estresado<sub>1</sub>, Ap\_ira<sub>2</sub>. Un ejemplo de descriptor por percepción es el apuntador Ap\_no\_s\_trafico<sup>5</sup> apunta al arreglo de eventos <numerosas multas de tráfico, conducción imprudente, conducción en estado de ebriedad, conducción intoxicado, conducción bajo la influencia de sustancias alucinógenas >. Similarmente son definidos los otros vectores. Por otra parte, la Tabla 3.15. define la estructura de un módulo de reconocimiento de patrón dinámico para el patrón de emoción del conductor  $\Gamma\rho_d = \text{“Ira”}$ .

**Tabla 3.15.** Estructura del módulo de reconocimiento de patrón  $\Gamma\rho_d = \text{“Ira”}$ 

E				
S		C		
Señal	Estado	Descriptor	Dominio	Peso
1	F	El conductor tira la puerta	<sonidos fuertes de puerta > <sup>1</sup>	0.6
2	F	Alta velocidad	< 150 Km/s $\geq$ velocidad $\leq$ 200 > <sub>1</sub>	0.8
3	F	Presionando fuerte el volante	<grados presión al volante> <sub>2</sub>	0.6
4	F	Alta dilatación de pupila	<Diámetro de dilatación de pupila entre 6 a 9 mm> <sub>3</sub>	0.6
5	F	Alto ritmo cardíaco	<de 200 a 100 latidos por min> <sub>4</sub>	0.8
6	F	Falla mecánica	<Fallas mecánicas consideradas> <sub>2</sub>	0.5
7	F	Falla eléctrica	<Fallas eléctricas consideradas> <sub>3</sub>	0.5
8	F	El conductor está gritando	<gritos considerados de conductores> <sub>5</sub>	0.8
9	F	El conductor esta serio	<estados serios de conducción> <sub>6</sub>	0.8
10	F	El conductor se mueve violentamente	<movimientos bruscos de conducción> <sub>4</sub>	0.6

Umbral:  $\Delta U1, \Delta U2$

### 3.3.1.3 Reconocimiento de la “Ira” orientado por características (DpoC).

En este caso, reconocemos los patrones emocionales de un conductor de vehículo usando patrones dinámicos orientados por características (*DpoC*). Se asumen tres descriptores fisiológicos con los valores que se muestran en la Tabla 3.16 para la emoción Ira.

**Tabla 3.16.** Condiciones fisiológicas para la emoción de Ira de un conductor

Patrones fisiológicos del conductor	Ira	Frecuencia Cardíaca	desde (0.5 to 1.0]
		Frecuencia respiratoria	desde (0.5 to 1.0]
		Presión sanguínea	desde (0.5 to 1.0]

Todos los valores están normalizados en el intervalo [0,1]. Un patrón consiste de una secuencia de valores (frecuencia cardíaca, respiratoria, presión sanguínea) del estado emocional del conductor en un intervalo de tiempo  $\Delta t$ . Para  $\Gamma\rho_d = \text{Ira}$ , usando la Ecuación (3.5), el vector  $D$  de todos los

descriptores DpoC es:  $D_n(Ira)=[d_{frecuenciaCardiaca}, d_{frecuenciaRespiratoria}, d_{presiónSanguínea}]$ . La Tabla 3.17 muestran la estructura del módulo para la emoción Ira.

**Tabla 3.17.** Matrix  $\Gamma\rho_{d=Ira}$  del Módulo de reconocimiento del patrón “Ira”

$S_d$		$E_d$	
Señal	Estado	Puntero (P)	Peso (P)
1	F	PunteroIra1frecuenciaCardica	1
2	F	PunteroIra2frecuenciaRespiratoria	0.8
3	F	PunteroIra3presiónSanguínea	0.9
$U_d: \langle \Delta U1, \Delta U2 \rangle$			

Para reconocer un patrón dinámico, sus descriptores son evaluados en el intervalo de tiempo  $[t_{i=1}, t_{j=n}]$ . Supóngase la siguiente serie de tiempo para el patrón “Ira” (ver Ecuación (3.9)):

- $\Delta t_{frecuenciaCardiaca}(Ira)=[(t1, 0.5),(t2, 0.6),(t3, 0.7),(t4, 0.8) ]$
- $\Delta t_{frecuenciaRespiratoria}(Ira)=[(t1, 0.5),(t2, 0.6),(t3, 0.7),(t4, 0.8) ]$
- $\Delta t_{presiónSanguínea}(Ira)=[(t1, 0.5),(t2, 0.6),(t3, 0.7),(t4, 0.8) ]$

Suponga la siguiente entrada  $s()=\{frecuenciaCardiaca[(t1, 0.5),(t2, 0.6),(t3, 0.7), (t4, 0.8)], frecuenciaRespiratoria [(t1, 0.5),(t2, 0.5),(t3, 0.6),(t4, 0.8) ], presiónSanguínea[(t1, 0.5),(t2, 0.6),(t3, 0.7),(t4, 0.7) ] \}$  de un conductor de vehículo.

Cada elemento de este conjunto es un patrón atómico. Cada patrón es reconocido individualmente, a través de un mapeo directo al conjunto existente de series de tiempo (vía su respectivo apuntador). De los tres patrones, sólo el patrón reconocido es *frecuenciaCardiaca*, de la emoción ‘Ira’, esto causa un cambio en el estado del módulo de reconocimiento del patrón ‘Ira’ de ser Falso a Verdadero (ver Tabla 3.18).

**Tabla 3.18.** Matriz  $\Gamma_{\rho_d=Ira}$  del Módulo de reconocimiento del patrón “Ira”

S <sub>d</sub>		E <sub>d</sub>	
S <sub>d</sub>		C <sub>d</sub>	
Señal	Estado	Puntero (P)	Weight (W)
<b>1</b>	<b>V</b>	<b>PunteroIra<sub>1</sub>frecuenciaCardica</b>	<b>1</b>
2	F	PunteroIra <sub>2</sub> frecuenciaRespiratoria	0.8
3	F	PunteroIra <sub>3</sub> presiónSanguinea	0.9
U <sub>d</sub> : <ΔU1, ΔU2>			

Sea  $\Delta U1 = 0.9$ , se procede a reconocer por características (DpoC) de acuerdo a las Ecuaciones (3.11, 3.13 y 3.15). De acuerdo a las Ecuación (3.11), las señales claves de este patrón son la primera y la tercera. Ahora, se usa la Ecuación 3.13 para determinar si el patrón puede ser reconocido con las señales reconocidas (sólo la primera). Esta señal es suficiente para superar el umbral del módulo de reconocimiento de la emoción ‘Ira’ ( $1 > 0.9$ ). El reconocimiento del patrón es exitoso, y como es el último nivel de la jerarquía (el nivel donde inicio el proceso de reconocimiento), este genera la señal de salida  $S_o$ , la cual se convierte en la señal de salida del sistema: “patrón reconocido  $S_o=Ira$ ”. Un análisis similar es posible si se incluyen descriptores orientados por percepción.

### 3.3.1.3 Comparación de Ar2p con otros enfoques.

En esta sección realizamos una comparación cualitativa y cuantitativa entre *Ar2p* con el paradigma de crónicas y lógica difusa aplicadas a este dominio. Cualitativamente, a nivel de seguridad se evalúa sus capacidades de comunicación en el escenario IoT (Internet de las Cosas), donde el intercambio de información es natural entre diferentes dispositivos, tal como dos vehículos. Cuantitativamente, se hace la comparación a nivel de tres propiedades (Huang, Fallah, Sengupta, y Krishnan 2010): i) a nivel de razonamiento, ii) a nivel de adaptación y iii) a nivel de comunicación, usando diferentes métricas.

### **3.3.1.3.1 Comparación cualitativa.**

*Crónicas.* La comunicación entre diferentes crónicas son los eventos. Por ejemplo, en un escenario el vehículo (v1) envía un evento informando a otros vehículos que el conductor se está quedando dormido. Esto es válido para el caso donde las crónicas están en el mismo carro o en carros diferentes. Estos eventos pueden incluir información específica requerida por las crónicas. Contienen información relevante cómo el conductor conduce, con el fin de generar las acciones concernientes con el estilo de conducción reconocido.

*Ar2p.* Sólo necesita enviar la señal sobre el reconocimiento de un patrón dado (por ejemplo, el estado emocional del conductor) a otros vehículos. Por ejemplo, que el vehículo (v1) envía una señal de reconocimiento de que el conductor se está quedando dormido a otros vehículos.

*Lógica difusa.* En este caso, se tiene dos posibilidades: enviar un valor discreto el cual debe ser desdifusificado en el otro vehículo (esto es, la salida del descriptor difuso debe ser desdifusificada y enviada al otro vehículo), o enviar el valor de las variables difusas (y en el otro lado el sistema difuso debe ser similar). El principal problema es que puede haber múltiples salidas (múltiples reglas activas, que pueden representar varios estilos de conducción activa), y deben enviarse a los otros vehículos para tener una idea real del contexto.

### **3.3.1.3.2 Comparación cuantitativa.**

Para realizar esta comparación se usó una base de datos artificial sobre estilos de conducción definida con datos reales, capturada de manera multimodal en diferentes proyectos que cubren los descriptores de cada nivel del modelo jerárquico de estilos de conducción (ver (Aguilar J, Aguilar K, Chavez y Puerto 2017), para una descripción detallada de la base de datos). En orden a evaluar

el sistema de reconocimiento de los paradigmas de comparación, se definieron algunos criterios de rendimiento agrupados en tres grupos:

- *Criterios relacionados con la capacidad de reconocimiento.* Consiste en detectar situaciones anómalas, como estilos de conducción negativos (agresivos, etc.), con posible información adicional sobre las causas, para informar al ADAS, de modo que pueda guiar al conductor hacia un estilo de conducción positivo. Para eso, las métricas sobre las capacidades de inferencia son importantes, lo que permite reconocer y diagnosticar. Las métricas utilizadas en este trabajo son:

*Coverage.* Es una métrica de completitud, con la cual se verifica si se cubren todas las situaciones posibles a reconocer (ver Ecuación 3.17). Específicamente, para el caso del estilo de conducción, los estilos de conducción que no pueden ser reconocidos/detectados.

$$P_{STYLES} = \frac{1}{\#Styles} \cdot \sum_{w=1}^i y_{STYLES_w} \quad (3.17)$$

Donde,  $P_{STYLES}$  es la parte de estilos de conducción que pueden ser reconocidos.  $\#Styles$  es el número de estilos de conducción, y  $y_{STYLES_w}$  es una variable binaria que es igual 1, si el estilo  $i$  es reconocido, en otro caso es 0. Esta métrica puede ser extendida para el caso de las emociones y estado del conductor.

*Compactness.* Es una métrica de densidad (ver Ecuación 3.18), definida por el número de patrones para reconocer un estado, un objeto, etc. Para el caso del patrón de estilos de conducción, esta es calculada como el número promedio de patrones usados para reconocer los diferentes estilos.

$$C_{RSTYLES} = \frac{\#Styles}{\#RStyles} \quad (3.18)$$

Donde  $C_{RSTYLES}$  es la densidad relativa de los estilos de conducción,  $\#RStyles$  es el número de

patrones usados durante la experimentación para reconocer el estilo de conducción. Esta métrica puede ser extendida para el caso de las emociones y estado del conductor.

*Tiempo de razonamiento*: es el tiempo promedio para reconocer (ver Ecuación 3.19)

$$Time_{R_t} = timeEndSimulation_t - timeStartSimulation_t \quad (3.19)$$

Donde  $Time_{R_t}$  es el tiempo promedio de la técnica t.

- *Criterios relacionados con la capacidad de adaptabilidad o aprendizaje*. A continuación, se especifican las métricas sobre la capacidad de aprendizaje de los paradigmas.

*Presicion (Mp)*. Esta métrica evalúa los casos correctos reconocidos y no otros (ver Ecuación 3.20). El resultado de esta operación está entre 0 y 1; una perfecta precisión es 1, en la cual los casos correctos son sólo reconocidos.

$$Mp = \frac{TREx}{TREx + TRFpx} \quad (3.20)$$

Tal que  $TREx$  es el total de reconocimientos exitosos (verdaderos positivos), y  $TRFpx$  es el total de respuestas fallidas (falsos positivos).

*Recall (Mr)*. Esta métrica evalúa el número de casos que son reconocidos, del total de casos que deben ser reconocidos (ver ecuación 3.21).

$$Mr = \frac{TREx}{TREx + TRFnX} \quad (3.21)$$

Donde,  $TRFnX$  es el total de respuestas fallidas (que debieron ser reconocidas) (falsos negativos).

*f-measure (Ma)*. Mide el rendimiento general de aprendizaje considerando  $Mp$  y  $Mr$  (3.22).

$$Ma = \frac{2MpMr}{Mp + Mr} \quad (3.22)$$

*Error cuadrático de aprendizaje (EAC)*: esta métrica evalúa el error cuadrático entre la salida que el paradigma da y la salida que debería dar (ver Ecuación 3.23).

$$EAC = \frac{1}{n} \sum_{i=1}^n (\widehat{Sp} - Sp)^2 \quad (3.23)$$

Donde  $\widehat{Sp}$  es un vector de n respuestas dadas por el paradigma,  $Sp$  es el vector de respuestas que debería dar.

- *Criterios relacionados con la capacidad para comunicación la información reconocida.*

Están orientados a la capacidad de comunicación de cada paradigma, considera la velocidad de transmisión, y que la información transmitida sea comprensible por los receptores. En este caso se utilizan dos métricas: *Tiempo de transmisión*. Define el tiempo para transmitir un mensaje, con la información reconocida por la técnica (ver Ecuación 3.24).

$$TT = \frac{\textit{Tiempo de transición real}}{\textit{Tiempo de transición óptimo}} \quad (3.24)$$

*Tiempo de procesamiento*. Define el tiempo para comprender los descriptores recibidos de otro sitio, con el fin de ser utilizado para reconocer una situación (ver Ecuación 3.25).

$$PT = \frac{\textit{Tiempo de respuesta actual}}{\textit{Tiempo de respuesta óptimo}} \quad (3.25)$$

### 3.3.1.4 Escenario experimental.

En esta sección se describen tres escenarios para la evaluación de  $Ar2p$ , comparándolo con lógica difusa y las crónicas: i) a nivel de razonamiento, ii) a nivel de adaptación y iii) a nivel de comunicación, usando diferentes métricas.

### 3.3.1.4.1 Capacidades de razonamiento.

Esta capacidad consiste en la posibilidad de reconocer las diferentes situaciones, y particularmente, la detección de situaciones anómalas, tal como malos estilos de conducción, de tal manera que el ADAS pueda guiar al conductor a un estilo de conducción más positivo. Los escenarios evaluados son:

- Evaluar la capacidad de reconocer la misma situación (emoción, estado o estilo) a través de diferentes patrones.
- Estudiar la capacidad de reconocer las emociones, estados o estilos básicos en los diferentes conductores.
- Verificar el correcto funcionamiento del patrón jerárquico.
- Evaluar la capacidad de reconocer diferentes emociones para el mismo conductor.

En este caso, se utilizan las siguientes métricas: *Coverage*, *compactness* y *Tiempo de razonamiento*. La Tabla 3.19 muestra los valores promedio de estas métricas para los escenarios definidos previamente.

**Tabla 3.19.** Resultados de la capacidad de razonamiento

Enfoque	Capacidad de razonamiento		
	Coverage	Compactness	Tiempo de Razonamiento (segundos)
Lógica Difusa	0.63	0.65	1,34
Crónicas	1.0	0.73	0.21
Ar2P	0.55	1.0	0.34

Con respecto a la capacidad de razonamiento, las crónicas pueden reconocer todas las situaciones posibles (*Coverage* = 1), pero requieren una gran base de datos de crónicas para reconocer todos los posibles casos. En este sentido, *Ar2p* es más eficiente (*Compactness* = 1), ya que requiere una base de datos más pequeña de módulos de reconocimiento por su esquema recursivo, los cuales reutilizan la información y mejoran su tiempo de ejecución. En general, el

rendimiento de la lógica difusa no es bueno, ya que necesita una gran base de datos de reglas, que no cubre todas las situaciones posibles. El mismo problema puede ocurrir con *Ar2p*, donde no hay patrones generales para diferentes situaciones (una ventaja de las crónicas). Otro problema es el proceso de razonamiento de la lógica difusa, que se basa en un proceso de inferencia que puede hacerlo muy lento en el tiempo de cálculo (Tiempo de razonamiento = 1.34). En general, es deseable una alta *Coverage*, lo que significa que la situación deseada fue exitosamente reconocida. Adicionalmente, una buena *Compactness* muestra la cantidad de información (patrones, descriptores, etc.) para reconocer las diferentes situaciones. Finalmente, el tiempo de respuesta es muy importante en un contexto en tiempo real, puesto que esta información es fundamental para que un ADAS pueda reaccionar rápidamente en una situación dada.

El proceso de razonamiento de las crónicas basadas en lógica temporal, describe naturalmente la situación actual. Es decir, el mecanismo de razonamiento se basa en los eventos de los descriptores y sus relaciones temporales, y maneja la incertidumbre de acuerdo a cuando ocurren los eventos. Además, las crónicas definen un diagnóstico basado en las causas detectadas, para determinar las acciones de control. El único problema es el tamaño de la base de datos de las crónicas, se requiere una gran base de datos de crónicas para reconocer las diferentes situaciones. *Ar2p*, por otra parte, puede reutilizar mucha información a través de los módulos de reconocimiento, lo cual le da una ventaja. Además, puede tratar con conocimiento incierto o no completo. Esto gracias a las estructuras de representación del patrón (es decir, los módulos de reconocimiento de patrones), que a través de sus parámetros, como el peso de importancia de los descriptores, admite diferentes formas o cambios en los descriptores de un patrón. A nivel del mecanismo de razonamiento, permite inferir una situación y navegar entre los módulos rápidamente.

Finalmente, la lógica difusa permite un razonamiento aproximado, que implícitamente puede manejar la incertidumbre, utilizando la idea de imprecisión y granularidad de la información en los descriptores difusos del modelo de patrón multimodal. El problema principal es obtener el conjunto de reglas y el tiempo de ejecución del Sistema Clasificador Multicapa Difuso utilizado MFCS (MFCS por sus siglas en inglés)(Aguilar J, Aguilar K, Chávez y Puerto 2017). El MFCS es una excelente estrategia para describir los diferentes niveles del modelo de patrón, pero introduce importantes costos de ejecución en aplicaciones en tiempo real.

### 3.3.1.4.2 Capacidades de aprendizaje.

Consiste en la capacidad de adaptación de cada paradigma, para las diferentes situaciones en el contexto vehicular, y a la personalidad del conductor. En este caso, las siguientes métricas usadas son: Precision, Recall, f-measure y error cuadrático de aprendizaje. La Tabla 3.20 muestra los valores promedio.

**Tabla 3.20.** Resultados de la capacidad de aprendizaje

Enfoque	Capacidad de aprendizaje			
	Precision	Recall	F-measure	Error Cuadrático
Lógica D	0,79	0.82	0.80	0.69
Crónicas	1	1	1	0
<i>Ar2p</i>	1	1	1	0,1

La Tabla 3.20 muestra que las crónicas dan los mejores resultados, porque obtienen la mejor *precision* y *recall* (como el algoritmo *Ar2p*), pero con un pequeño error. La lógica difusa normalmente se basa en un procedimiento de elitismo basado en expertos. El sistema clasificar difuso (Fuzzy Classifier System:FCS) permite un proceso de aprendizaje, basado clásicamente en enfoques evolutivos, lo cual no es eficiente en situaciones en tiempo real como el ADAS, mientras que el algoritmo de aprendizaje de *Ar2p* es bastante bueno, aunque converge en un error cuadrático superior a las crónicas.

De acuerdo con los resultados, la *precision* = 1 determina que los paradigmas reconocen con precisión los patrones, sin hacer otros reconocimientos inesperados. Este valor de precisión se debe a que los paradigmas aprenden situaciones muy específicas y únicas. El *recall* = 1 indica que los paradigmas pueden descubrir todos los patrones considerados que un conductor puede experimentar durante el proceso de conducción. Además, los paradigmas reconocen la misma situación (emoción, estado y / o estilo) con diferentes patrones, expresando la diversidad de contexto en la cual la misma situación puede ocurrir. Estos resultados consideran el caso del aprendizaje en línea y la customización de los patrones, cuando los patrones genéricos son construidos para cada emoción, estilo y estado (típicamente, para el caso de las crónicas y los módulos de *Ar2p*). En general, la calidad del algoritmo de aprendizaje para las crónicas y *Ar2p*, se debe a que el aprendizaje de patrones se realiza cada vez que se detecta un cambio en los descriptores.

En general, las crónicas pueden describir una misma situación (una emoción, un estado del conductor, un estilo de conducción) usando diferentes crónicas. Este enfoque requiere una robusta base de datos de crónica, la cual se aprende constantemente para adaptarse al conductor y las nuevas situaciones (Aguilar J, Aguilar K, Chávez y Puerto 2017).

Con respecto a *Ar2p*, utiliza dos estrategias de adaptación (Puerto y Aguilar 2016b), la primera, llamada *Aprendizaje nuevo*, la cual se produce cuando no se reconoce el patrón de entrada (no hay un módulo que lo reconozca). La segunda, llamada *Aprendizaje por refuerzo*, la cual se produce cuando se reconoce un patrón de entrada. Estos dos mecanismos de aprendizaje permiten una rápida adaptación al estilo de conducción del conductor. Por otro lado, *Ar2p* puede adaptar sus módulos de reconocimiento de patrones de acuerdo con los patrones reconocidos, reajustando la importancia de los pesos. Finalmente, un FCS puede aprender las reglas y las estructuras de las

variables difusas. En particular, las funciones de pertenencia de las variables difusas se pueden adaptar al contexto, y las reglas de la base de datos se pueden modificar (sus componentes antecedentes y consecuentes) (Aguilar J, Aguilar K, Chávez y Puerto 2017). Para lograr esto, el FCS requiere un proceso de modificación de las reglas, que no garantiza buenos resultados a nivel del proceso de aprendizaje ( $f\text{-measure} = 0,80$ ).

### 3.3.1.4.3 Capacidad de Comunicación.

En este último caso, se evalúa la capacidad de cada paradigma para transmitir la información de reconocimiento a otros conductores, ADASs, entre otros, de manera clara y con significado. Este caso es fundamental en el contexto de IoT (Internet de las Cosas), donde el intercambio de información es entre dispositivos, por lo que debe ser preciso, contextualizado, etc., para que sea útil. Para alcanzar este objetivo, la transmisión de la información debe ser rápida, pero adicionalmente, la información enviada debe ser útil para el receptor. En este caso, se utilizan las siguientes métricas: *tiempo de transmisión* y *tiempo de procesamiento*. La Tabla 3.21 muestra los valores promedio de estas métricas para los escenarios definidos previamente.

**Tabla 3.21.** Resultados para las capacidades de comunicación

Enfoques	Capacidades de comunicación	
	Tiempo de respuesta	Tiempo de transmisión
Lógica Difusa	0,96	0.77
Crónicas	0,12	0.063
<i>Ar2p</i>	0.093	0.081

En general, los tiempos de comunicación son mejores para *Ar2p*, ya que este simplemente envía una señal, que es reconocida por los módulos de reconocimiento de nivel superior (que pueden estar en diferentes dispositivos). En el caso de las crónicas, se envían eventos involucrados en un proceso de reconocimiento, el cual debe ser localmente interpretado. Estos eventos incluyen la información específica requerida por las crónicas (como la emoción experimentada por un

conductor), pero esta es la única información requerida. En el caso de FCS, la información correspondiente debe ser enviada, y se requiere en el sitio local, un mecanismo de razonamiento difuso para procesar las variables difusas, o la fusificación de los valores recibidos. Este tiempo adicional debe ser agregado, para descubrir la situación actual. Así, la comunicación en *Ar2p* son señales entre los módulos de reconocimiento, en las crónicas son eventos, y en FCS son variables difusas o valores que deben ser difusos.

En resumen, en las crónicas, los eventos pueden incluir información específica requerida por las crónicas, pero es la única información requerida. El modelo jerárquico de patrón de conducción comunica eventos generados por los diferentes descriptores, o el reconocimiento. *Ar2p* solo necesita enviar las señales requeridas por el módulo de reconocimiento, que corresponde a un descriptor determinado (por ejemplo, el estado emocional del conductor). Esta señal puede ser la entrada de uno de los módulos de reconocimiento en el otro lugar. Finalmente, la lógica difusa puede enviar valores discretos o difusos que deben procesarse en el otro sitio, lo cual implica más tiempo de comunicación (Tiempo de respuesta = 0,96).

### **3.3.1.5 Análisis y conclusiones parciales de los resultados del experimento.**

Con respecto a las métricas, permiten determinar la calidad de los resultados sin la necesidad de compararlos con otros trabajos. Estas métricas evalúan la calidad de las diferentes capacidades de los enfoques. Las métricas como cobertura (en el caso de razonamiento), precisión (en el caso de aprendizaje) y tiempo de comunicación, dan una idea de la calidad de estas capacidades.

Además, los patrones y el contexto experimental definidos en otros trabajos son muy diferentes de nuestro estudio, lo que hace una comparación muy compleja con trabajos anteriores. Este experimento presenta muy buenos resultados para patrones más complejos que los existentes en la literatura. Solo el modelo de lógica difusa tiene unos resultados no muy buenos, a diferencia de

*Ar2p* y las crónicas, cuyas métricas de aprendizaje y razonamiento son muy buenas, cerca de 1, lo que habla de la calidad de estos paradigmas para aprender y reconocer el patrón de estilo de conducción. Algunas observaciones importantes son: i) el modelo jerárquico es más complejo que los patrones de estilo de conducción clásicos utilizados en la literatura; ii) este modelo jerárquico incluye más descriptores, lo que hace más preciso el proceso de reconocimiento (es el principal valor agregado del modelo de modelo); iii) los métodos de reconocimiento pueden aplicarse con información parcial o ambigua sobre los patrones.

La mayoría de los métodos existentes en la literatura sobre el reconocimiento del estilo de conducción en conductores de vehículos para ADAS, utilizan diferentes enfoques, parámetros, características derivadas de datos de sensores, aspectos cognitivos como atención, juicio, toma de decisiones y estados afectivos, los cuales desempeñan un papel fundamental en el comportamiento racional, funcional e inteligente de los usuarios. La Tabla 3.22 muestra algunos criterios de comparación para evaluar los resultados presentados en esta sección, con respecto a trabajos recientes que tienen objetivos similares. Los criterios definidos son: *métodos de reconocimiento, enfoque, características discriminativas utilizadas, y clasificación de los estilos de conducción.*

Nuestro enfoque cubre y extiende algunos aspectos de otros modelos. Por ejemplo, en términos de características, nuestro modelo considera un gran número de descriptores (27 descriptores) para el reconocimiento, superando a los otros modelos. También cubre más capas del dominio en el modelo jerárquico; uno para reconocer las emociones del conductor, otro para reconocer los estados del conductor y, finalmente, el último corresponde a los estilos de conducción.

**Tabla 3.22.** Comparación con otros métodos de reconocimiento de estilos de conducción

Sistema	Método de reconocimiento	Enfoque	Propiedades discriminativas	Clasificación de estilos de conducción
<i>Nuestro modelo</i>	<i>Lógica difusa Crónicas Ar2P</i>	<i>Modelo jerárquico para el reconocimiento de estilos de conducción.</i>	<i>27 propiedades entre estados y emociones del conductor</i>	<i>ecológico normal agresivo</i>
(Han et al 2018)	Método basado en estadística: Teoría bayeciana, basado en distancia euclidiana, validación cruzada	Probabilidad bayeciana con kernel de estimación de densidad	8 propiedades: Aceleración, rotación, desplazamiento lateral, velocidad del vehículo, Angulo de dirección. Señal física, señal fisiológica	agresivo normal
(Yang et al 2018)	K-means y máquinas de soporte vectorial o (support vector machine)	Método de aprendizaje de dos capas para el reconocimiento del comportamiento de conducción utilizando datos EEG	Señales fisiológicas tal como electroencefalograma (EEG) para recolectar datos	Cinco grupos de conductas de conducción se clasificaron según estas características de conducta de conducción bidimensionales
(Dörr et al 2018)	Lógica difusa	Reconocimiento usando lógica difusa	Tipo de carretera, evento de aceleración longitudinal, evento de aceleración lateral, diferencia de velocidad.	normal confortable deportivo
(Dong et al 2016)	Aprendizaje con red neuronal convolucional: usando 1-D convolución y pooling	Aprendizaje profundo: el enfoque propuesto consta de dos componentes: transformación de datos y aprendizaje de características mediante redes profundas.	velocidad normal, diferencia de velocidad normal, aceleración normal, diferencia de aceleración normal, y velocidad angular, utilizando datos de sensores del vehículo, e.g., GPS	Patrones de conducción: desaceleración en giros bruscos, conducción a alta velocidad por carreteras rectas
(Wang et al 2017)	k-means y método de clustering	Enfoque semisupervisado de máquinas de soporte vectorial	Basados en unos pocos puntos de datos etiquetados	Agresivo Normal

A modo de conclusiones, este escenario de prueba es un modelo flexible que permite incorporar fácilmente nuevos descriptores en el modelo jerárquico, y utiliza los datos disponibles en un momento dado para reconocer. Además, está orientado a incorporar estados emocionales, estilo de conducción, entre otros, en ADAS y ACC, para brindar mayor seguridad y confort.

Particularmente, este escenario cubre los aspectos necesarios para incorporar los factores humanos en ADAS, es decir, la capacidad de reconocer los estilos de conducción usando *Ar2p*. Estos resultados pueden extrapolarse a estudios sobre la interacción hombre-máquina, dentro del área conocida como computación afectiva (ver sección 3.3.2 reconocimiento de emociones en robots).

El modelo de prueba captura la mayor cantidad de información posible sobre los estilos de conducción, los estados del conductor, o las emociones del conductor, haciendo uso de los enfoques multimodales de percepción. De esta manera, agrega una mayor cantidad de información, lo que hace posible un proceso de reconocimiento más preciso.

Se confrontó, además, *Ar2p* con dos técnicas para reconocer el estilo de conducción, una basada en lógica difusa, y otra basada en crónicas. Se comparó sus mecanismos de razonamiento, para determinar las posibles causas o detectar estados anormales, sus capacidades de adaptación, y sus capacidades de comunicación de la información reconocida, lo cual es muy importante en la IoT. Se observó que cada técnica tiene sus ventajas y desventajas, y depende del contexto real (IoT) para elegir una de ellas. Algunos de los hallazgos de estas técnicas aplicadas fueron que el único que no tiene buenos resultados es el modelo difuso. Primero, porque su proceso de aprendizaje no es eficiente y es lento; segundo, porque requiere una gran base de reglas para garantizar que todos los casos estén cubiertos; y finalmente, porque es necesario definir mensajes con información suficiente, para comprender la información generada por ellos en otros sitios. Por otro lado, se encontró que el proceso de adaptación en los otros enfoques (crónicas y *Ar2p*) permite el descubrimiento de patrones para expresar la diversidad de contextos que pueden ocurrir durante la conducción. La capacidad de razonar, en particular, en las crónicas, permite reconocer situaciones de diferentes maneras: la misma situación con diferentes patrones, diferentes situaciones,

situaciones caracterizadas por patrones atómicos (por ejemplo, solo emociones), o situaciones complejas (descritas por patrones complejos). En el caso de la comunicación, no es un problema para *Ar2p* y las crónicas, porque la transmisión es de señales o eventos que describen un cambio de valor de un descriptor, o algo relevante reconocido en un conductor (una emoción, un estado o un estilo).

Un trabajo futuro es llevar a cabo la implementación de estas técnicas en un entorno real, conectado a ADAS. Los casos de prueba definidos en el trabajo actual se desarrollaron en una base de datos artificial, utilizando datos reales. Sería importante probar el comportamiento del modelo en entornos reales con fuertes limitaciones de tiempo y grandes cantidades de flujo de datos. Para tales pruebas, el vehículo debe estar equipado con sistemas como cámara, sensor de presión arterial, sensor de temperatura, micrófono, GPS, entre otros, que permitirán percibir de manera multimodal los descriptores que componen los patrones de conducción, para capturar tanta información como sea posible sobre los estilos de conducción.

### **3.3.2 Reconocimiento de emociones en robots usando Ar2p**

En esta sección se lleva a cabo otro experimento, para evaluar las capacidades de reconocimiento de patrones de *Ar2p*, esta vez en el contexto de la robótica de enjambre (Gil, Aguilar, Puerto y Dapena, 2018, Gil, Aguilar, Puerto y Dapena 2019). La robótica de enjambre es un sistema de múltiples robots en el que surge un comportamiento colectivo, a partir de las interacciones entre los robots y el entorno. Para el experimento, se ha propuesto un modelo emocional para los robots, para permitir comportamientos emergentes. El modelo emocional utiliza cuatro emociones universales: ira, disgusto, tristeza y alegría, asignadas a cada robot, en función del nivel de satisfacción de sus necesidades básicas. Estas emociones se encuentran en un espectro en el que, dependiendo de dónde se encuentre la emoción del robot, puede afectar su comportamiento y el

de los robots vecinos. Cuanto más negativa es la emoción, más individualista se vuelve en sus decisiones. Cuanto más positivo sea el robot en su emoción, más tendrá en cuenta los objetivos globales y de grupo. Cada robot puede reconocer la emoción de otro robot en el sistema según su estado actual, utilizando *Ar2p*. El experimento aborda la influencia de las emociones en el comportamiento del sistema, a nivel individual y colectivo, y los efectos de las emociones en los comportamientos emergentes del sistema multi-robot. En la prueba se analizan dos escenarios: la recolección de néctar y el transporte de objetos, y muestra la importancia de las emociones en el comportamiento emergente en un sistema multi-robot.

### **3.3.2.1. Introducción al problema de reconocimiento de emociones en robots.**

La robótica de enjambre es un conjunto de robots autónomos que trabajan juntos para realizar una tarea. En estos sistemas, los robots coordinan sus acciones de forma descentralizada. Con la robótica de enjambre se intenta modelar comportamientos de colaboración de organismos sociales como los insectos, los cuales logran hacer tareas complejas que van más allá de la capacidad individual (tal como la construcción de un nido).

Por otro lado, las emociones (nuestros patrones de interés a reconocer usando *Ar2p*) determinan cómo un individuo percibe un estímulo, lo que le permite actuar de manera diferente en situaciones similares, dependiendo de su estado de ánimo. La inclusión de las emociones y el reconocimiento de las mismas en este contexto, busca estudiar su influencia en el comportamiento del sistema, tanto a nivel individual como colectivo. En concreto, las emociones determinan el proceso de toma de decisiones de los robots y hacen posible los fenómenos emergentes en el sistema. En particular, la representación, reconocimiento y gestión de las emociones en cada robot permite extender su comportamiento individual (un agente que solo toma decisiones inteligentes) hacia un

comportamiento más social, de modo que cada robot pueda captar señales sociales (emociones) de los otros robots, en función de las cual actúa.

En la robótica de enjambres ocurren tareas que son producto de comportamientos emergentes y autoorganizados, como el forrajeo, el agrupamiento, la agregación, la formación, entre otros (Cao, Fukunaga y Kahng, 1997; Gautam y Mohan 2012). En este experimento se estudia el efecto de las emociones en el comportamiento emergente de un sistema multi-robot, basado en su proceso de reconocimiento. La emoción define el comportamiento de un robot en un momento dado, lo que influye en el comportamiento colectivo del sistema multi-robot, generando así la emergencia y la autoorganización en el sistema.

Existen diferentes trabajos que han agregado emociones a los robots, en (Gil, Aguilar, Puerto, y Dapena, 2018, Gil, Aguilar, Puerto y Dapena 2019) se referencian varios trabajos donde se especifican emociones en sistemas multi-robot, con el objetivo de considerar la influencia de ellos en la toma de decisiones de cada robot. Por otra parte, el aprendizaje y el reconocimiento son también analizados en los sistemas multi-robots (Ren, Wang, Altmeyer y Schweizer 2014; Alexander y Smales 1997; Jose 1998). Por ejemplo, en (Jing et al 2015) se propone un modelo cognitivo emocional para robots que cuidan personas mayores, que combina el filtro de Gabor<sup>10</sup>, el algoritmo de patrón binario local (LBP) y el algoritmo del vecino más cercano (KNN), para extraer y reconocer las características emocionales faciales. En (Banik, Watanabe, Habib y Izumi 2008; Banik, Watanabe y Izumi 2007) se propone una arquitectura computacional para modelar la emoción basada en la teoría de Markov. En ese trabajo se utilizan las emociones para la ejecución de tareas cooperativas. Con la capacidad emocional, cada robot puede distinguir el entorno, puede

---

<sup>10</sup> El filtro de Gabor es un filtro lineal cuya respuesta de impulso es una función sinusoidal multiplicada por una función gaussiana.

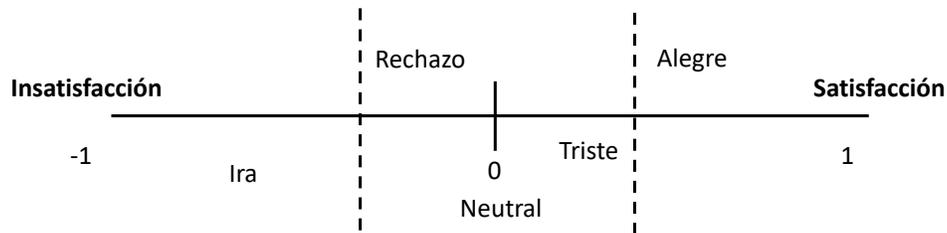
reaccionar adaptándose y, además, puede comprender el estado del robot colega. En (Fang et al 2014) se propone un modelo emocional para robots, que mide la voluntad cooperativa individual en el problema de la asignación de tareas multirobot. De esta manera, las emociones se utilizan como un factor en la asignación de tareas.

En esta aplicación de *Ar2p* bajo el modelo emocional propuesto, el valor de la "emoción" representa el estado de ánimo general de un robot, en función de su estado actual de batería / operación / seguridad / interacción, que son parámetros intrínsecos de la operación del robot, con el fin de permitir comportamientos emergentes en el sistema multi-robot. En este contexto, los robots deben reconocer las emociones y actuar de acuerdo con sus propios intereses y la emoción reconocida. En esta aplicación de *Ar2p*, los robots están equipados con un conjunto de emociones básicas que influyen en sus procesos de percepción y funcionamiento. El estado emocional en un instante  $t$  de un robot, define su comportamiento individual; por agregación, los estados emocionales de los  $n$  robots definen el estado colectivo del sistema y, por lo tanto, el comportamiento del mismo. En esta caso, no consideramos cómo se generan las emociones en los individuos del sistema, sino más bien, nos centramos en el proceso de reconocimiento.

Para este escenario de prueba, se asume que cada robot está equipado con la capacidad para reconocer emociones vía el algoritmo de reconocimiento *Ar2p*. La capacidad del robot para reconocer las emociones de los demás en un momento determinado le permite tomar mejores decisiones, facilitando procesos emergentes. Por ejemplo, si un robot necesita reclutar a otros miembros del grupo para transportar un objeto, y requiere seleccionar a otros que estén dispuestos a colaborar (suponiendo que su voluntad de colaborar esté relacionada con el estado emocional de la alegría), entonces optimizará la tarea enviando mensajes a los individuos cuyo estado emocional sea la alegría.

### 3.3.2.2. Modelo emocional de prueba para Ar2p.

En nuestra propuesta de modelo emocional de prueba, el espectro emocional se basa en cuatro emociones que varían su intensidad, de un valor altamente negativo a un valor altamente positivo, que se relacionan con un índice de satisfacción del robot (ver Figura 3.12).



**Figura 3.12.** Modelo emocional propuesto

En nuestra propuesta, utilizamos similares factores relacionados con el rendimiento de un robot utilizados por (Masuyama y Loo 2015) para modelar la personalidad, pero simplificando el espectro a un espacio unidimensional que representa el grado de satisfacción del robot (ver Figura 3.12). En nuestro modelo, los robots modifican sus acciones de acuerdo con el estado emocional que se activa en un momento dado.

En esta aplicación, el valor de la emoción representa el estado de ánimo general del robot, en función de su estado de batería / operación / seguridad / interacción. En particular, en (Perozo, Aguilar y Terán 2008) se define un conjunto de emociones, positivas y negativas, que influyen en la disposición del robot hacia comportamientos colectivos o individuales. Nuestro modelo se basa en estas ideas, y utiliza cuatro emociones básicas: ira, disgusto, tristeza y alegría, con un estado "neutral". Estas emociones están vinculadas a niveles de satisfacción del robot, y son representadas en un espacio unidimensional, en el intervalo  $[-1, 1]$ , donde tres subdivisiones definen las relaciones emoción-comportamiento, que pueden ser: reactiva, cognitiva y comportamientos imitativos, respectivamente.

El modelo busca proporcionar un espectro emocional simplificado a los robots que actúan en el sistema, con el fin de influir en la forma en que ellos manejan sus comportamientos, lo que también afecta el comportamiento colectivo. Estos tres tipos de comportamiento (imitativo, cognitivo y reactivo) están asociados a un estado emocional, como se ha propuesto en (Perozo, Aguilar y Terán 2008; Perozo, Aguilar, Terán y Molina 2013), donde las emociones negativas predisponen al individuo a la resolución del problema utilizando un enfoque individual, mientras que las emociones positivas llevan al individuo a objetivos globales, utilizando un enfoque colectivo.

Específicamente, el modelo emocional se implementa en una arquitectura a tres niveles descrita en (Pérez, Castro, Estrada, y González 2016). El primer nivel proporciona un soporte local al robot, gestiona sus procesos de acción, percepción y comunicación, así como su componente de comportamiento. El componente de comportamiento considera los aspectos reactivos, cognitivos, sociales y afectivos del robot, que influyen en su comportamiento y cómo interactúa con el entorno y con los otros robots del sistema. El segundo nivel soporta los procesos colectivos del sistema, al igual que los mecanismos de cooperación, colaboración, planificación y/o negociación, que pueden ser necesarios en cualquier momento. Este nivel de la arquitectura está basado en el concepto de coordinación emergente. El tercer nivel es responsable de la gestión del conocimiento y los procesos de aprendizaje, tanto individual como colectivamente, que se producen en el sistema.

Como ya se comentó, la emoción representa el estado actual de un robot, según su situación de batería / operación / seguridad / interacción:

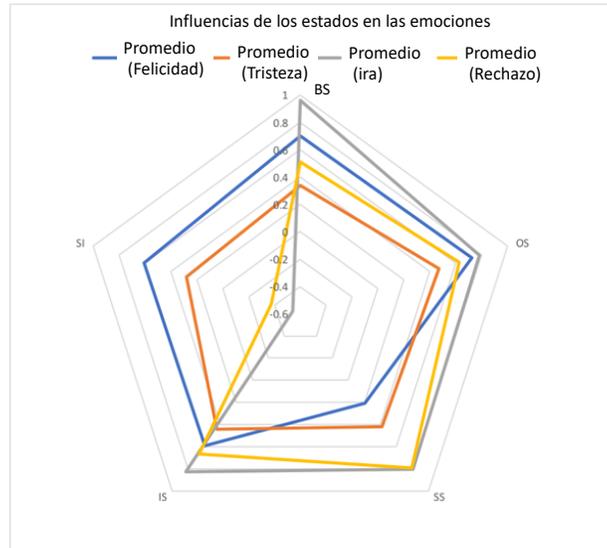
- Estado de la batería (BS): representa el nivel de energía de la batería del robot (porcentaje de batería). El nivel de carga afecta el rendimiento del robot.

- Estado de operación (OS): representa el nivel de rendimiento del robot, en relación con su participación activa en el sistema. Una participación activa en un alto porcentaje del tiempo es un buen síntoma.
- Estado de seguridad (SS): se define por el promedio de colisiones (debido a competencias u obstáculos) y fallas por minuto. Si el robot presenta pocas fallas y colisiones, entonces tiene un buen rendimiento.
- Estado de interacción (IS): se define por la capacidad social del robot, medida por el número de mensajes (enviados, recibidos) por minuto. Entre más interacciones, más se afecta positivamente al robot.

BS, OS, SS e IS se normalizan en el intervalo [0, 1]. Por otro lado, los robots comparten su estado interno utilizando el siguiente formato en el archivo "*file\_robot*" (Gil, Aguilar, Rivas, Dapena y Hernandez 2015; Kehoe, Patil, Abbeel y Goldberg 2015):

```
<file_robot_n>
  <body> sub_estado_1= valor</body>
  <body> sub_estado_2= valor </body>
  <body> sub_estado_3= valor </body>
  <body> sub_estado_4= valor </body>
</file_robot_n/>
```

Particularmente, los subestados son utilizados por el modelo emocional para calcular la intensidad emocional de los robots (ver Figura 3.13).



**Figura 3.13.** Influencia de los valores de los sub-estados en las emociones

### 3.3.2.3 Reconocimiento de emociones en robots usando Ar2p.

De acuerdo con la descripción formal *Ar2p* (ver Sección 3.1.1, y 3.1.2), en esta sección se explica cómo los robots equipados con *Ar2p*, pueden reconocer la emoción de tristeza. Para este propósito, se crea una instancia basada en el modelo emocional del sistema multi-robot propuesto en la sección anterior. Para ello, se crean las instancias de las estructuras de datos necesarias de *Ar2p* (ver sección 3.1.1), junto con las estrategias de reconocimiento:

*Suposición 1.* El conjunto de sub-estados del estado actual de un robot (BS, OS, SS, IS), son los patrones a detectar.

*Suposición 2.* Un sub-estado puede tener tres valores posibles: Bajo, Alto o Normal. De esta manera, los patrones a descubrir son:

- Patrón de batería (BS): Patrón de batería baja (LBP), Patrón de batería alta (HBP) o Patrón de batería normal (NBP).

- Patrón de operación (OS): Patrón de operación bajo (LOP), Patrón de operación alto (HOP) o Patrón de operación normal (NOP).
- Patrón de seguridad (SS): patrón de seguridad bajo (LSP), patrón de seguridad alto (HSP) o patrón de seguridad normal (NSP).
- Patrón de interacción (IS): patrón de interacción bajo (LIP), patrón de interacción alto (HIP) o patrón de interacción normal (NIP).

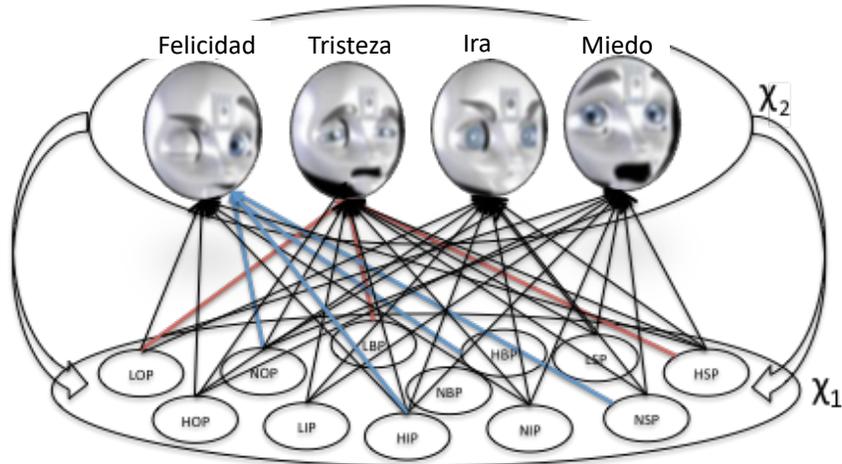
*Suposición 3.* Hay dos niveles de reconocimiento en la jerarquía del algoritmo  $Ar2p$  (ver Figura 3.1), en el primer nivel  $X_1$  están los módulos  $\Gamma_{j1}$  para el reconocimiento de patrones atómicos (LOP, HOP, NOP, LSP, HSP, etc.); y en el segundo nivel  $X_2$  están los módulos  $\Gamma_{j2}$  para los patrones emocionales (como "ira", "neutral", "tristeza", "alegría").

*Suposición 4.* El proceso de reconocimiento recibe como entrada el archivo "*file\_robot*" definido en la sección anterior, desde la nube. El archivo "*file\_robot*" contiene la información sobre los sub-estados (BS, OS, SS, IS) que determinan la emoción de cada robot en un momento dado. Veamos un par de ejemplos del comportamiento de  $Ar2p$  en el reconocimiento de las emociones (tristeza y alegría).

### 3.3.2.3.1 Reconocimiento de la tristeza en un robot usando $Ar2p$ .

En este caso,  $Ar2p$  recibe como patrón de entrada el siguiente archivo  $y=s()$  = "*file\_robot1*" (esta información se corresponde con las líneas rojas de la Figura 3.14).

```
<file_robot1>
  <body>BS= LBP</body>
  <body>OS= LOP</body>
  <body> SS= HSP</body>
  <body> IS = LIP </body>
</file_robot1/>
```



**Figura 3.14.** Los niveles de la jerarquía de patrones para reconocer la emoción.

En la Figura 3.14,  $X_1$  contiene los módulos de reconocimiento de bajo nivel y  $X_2$  los módulos de reconocimiento de alto nivel. Se supone que los módulos del nivel  $X_1$  están instanciados (ver Tabla 3.23, módulo del patrón emocional de la tristeza).

**Tabla 3.23.** Módulo de reconocimiento del patrón tristeza en Ar2p

S		E		
		C		
Señal	Estado	Descriptor	Dominio (V)	Peso (P)
1	T	BS	< LBP, HBP, NBP >	[0.8,0.1,0.5]
2	T	OS	<LOP, HOP, NOP >	[0.7,0.1,0.5]
3	T	SS	< LSP, HSP, NSP >	[0.1,0.8,0.7]
4	T	IS	< LIP, HIP, NIP >	[0.9,0.1,0.5]
U: < $\Delta U1=0.85, \Delta U2=0.75$ >				

El valor de cada descriptor dentro del arreglo del dominio (V) se mapea uno a uno con el valor del peso de la columna (P). Así, el descriptor BS (Señal 1) tiene un LBP = 0.8, un HBP = 0.1 y un NBP = 0.5. Los pesos de mayor valor, como el de LBP = 0.8, significa que este descriptor es importante para reconocer la emoción de la tristeza, a diferencia de los otros que no son tan importantes. Igual descripción para los otros descriptors OS, SS e IS.

El proceso general para el reconocimiento de la emoción de tristeza desde el patrón de entrada “file\_robot1”, es como sigue: Primero, los sub\_patrones de entrada (LBP, LOP, HSP y LPI) son

reconocidos en el nivel  $X_1$ , luego, el reconocimiento de estos descriptores son enviados al siguiente nivel en la jerarquía. Posteriormente, *Ar2p* reconoce un patrón de  $X_2$ , usando los teoremas de reconocimiento presentando en la sección 3.1.1. Específicamente, el reconocimiento en el nivel  $X_1$  coloca las señales del módulo de reconocimiento del patrón tristeza de  $X_2$  en activas, i.e cambian su estado a (“True”) (ver Tabla 3.23). Seguidamente, es aplicado el Teorema 1. De acuerdo con el teorema es necesario determinar cuales de las señales son claves, para esto se usa la Ecuación 3.1:  $P_{\text{promedio}} S(\Gamma_{\text{tristeza}}) = (0.8 + 0.9 + 0.8 + 0.7)/4 = 0.8$ .

Una vez se tiene el promedio de los pesos, se determina las señales claves. Las señales claves del patrón de entrada *<file\_robot1>* son: las señales 1, 3 y 4. Ahora, es usada la ecuación 3.2 para reconocer el patrón por señales claves, el resultado obtenido no excede el umbral establecido ( $0.83 < 0.85$ ). En este caso, el algoritmo *Ar2p* intenta reconocer aplicando la segunda estrategia de reconocimiento por señales parciales, usando la Ecuación 3.3 y el Umbral  $\Delta U_2 = 0.75$ . En este caso se tiene el resultado ( $0.8 > 0.75$ ), con lo cual, el patrón es reconocido, y como este es el último nivel de la jerarquía  $X_{i=2}$ , se produce una señal  $S_o$  que se convierte en la señal de salida del sistema “robot1(triste)”.

### 3.3.2.5 Escenario de aplicación de *Ar2p* en el sistema multi-robot: Recolección de Nectar

El propósito de esta aplicación es mostrar el comportamiento emergente en el sistema multi-robot, basado en las capacidades de los robots para reconocer las emociones de los otros robots usando *Ar2p*. Por ejemplo para reclutar, tomar decisiones, entre otras situaciones. Para llevar a cabo esta prueba, se han considerado los siguientes dos escenarios clásicos. El primero tiene que ver con la recolección de néctar de las abejas, y el segundo es el transporte de un objeto por las hormigas, ambos son ejemplos clásicos de procesos emergentes (Aguilar 2014). En ambas situaciones, se

asume que las emociones influyen en las ejecución de las tareas de los robots en el sistemas multi-robot.

Para el caso de *Recolección de Nectar*, los siguientes procesos son considerados: en la colmena, un grupo de abejas exploradoras se va en busca de fuentes de alimento que puedan ser explotadas. Cuando se localiza una fuente de alimento, las abejas exploradoras comunican a otras abejas la ubicación de la fuente y su beneficios. Al modelar este proceso en el sistema multi-robot, se definen los siguientes elementos:

- *Fuente de alimento*: en este caso, son los centros de recargados. La necesidad de obtener energía del robot está relacionada con la necesidad de alimento de la abeja. Cuando el subestado de BS explicado en la sección anterior tiene un valor bajo, el robot necesita recargarse, y esto afecta los valores de los otros subestados.
- *Abejas forrajeras fuertes o empleadoras*: están representadas por robots cuyo estado emocional tiende a ser positivo, donde sus subestados BS y OS se encuentran con valores normales o altos.
- *Abejas forrajeras débiles o empleadas*: este grupo de abejas está en busca de una fuente de alimento, y permanece en la colmena a la espera de elegir una fuente. En este caso, se considera que algunos robots observan a otros robots, si sus BS y OS están en niveles bajos. Debido a esto, sus estados emocionales son tristes, porque requieren una recarga de energía.
- *Abejas exploradoras*: son responsables de buscar fuentes de alimento. De acuerdo con nuestro modelo emocional, las emociones positivas (por ejemplo, la alegría) generan un comportamiento colectivo en los individuos como la búsqueda de alimentos. En nuestro caso, los robots exploradores pueden enviar el mensaje directamente a los débiles, si

reconocen la tristeza de la emoción en ellos. En nuestro caso, usamos Ar2p para ese reconocimiento.

En este escenario, cuando los valores de OS y BS decaen, los robots no pueden operar normalmente, bajando su rendimiento. Esto es modelado por los robots con la emoción de tristeza. De esta manera, esperan recibir un mensaje que les notifique sobre una fuente de recarga (en la tarea de recolección de néctar, las abejas exploradoras bailan para comunicarse sobre la calidad de las fuentes). En este escenario, cuando los robots descubren una fuente de recarga, entonces comunican el descubrimiento a los robots que están tristes o necesitan recargar su batería.

#### *3.3.2.4.2 Escenario de aplicación de Ar2p en el sistema multi-robot: Transporte de un Objeto.*

El transporte de un objeto es una tarea común en varias especies de insectos. Por ejemplo, las hormigas pueden transportar objetos grandes de manera colectiva. En general, el proceso comienza cuando una hormiga encuentra un objeto e intenta moverlo; si no puede moverlo, entonces trata de reclutar otras hormigas, que se forman alrededor del objeto para moverlo (Aguilar 2014).

En nuestro caso, los robots con un estado emocional de alegría (sus subestados están en niveles normales o altos) estarían dispuestos a llevar un objeto. Según nuestro modelo emocional, una emoción de alegría genera un comportamiento colectivo en los individuos. De esta manera, el robot que encuentra un objeto intenta reclutar a otros robots que se encuentran en el mismo estado emocional, para ayudarlo a mover el objeto. El reclutamiento se realiza a través de mensajes directos a los robots en el mismo estado. De esta manera, el robot debe reconocer la emoción de felicidad/alegría. De esta forma, el robot que está reclutando intenta reclutar a los robots felices. Así, el sistema multi-robot utiliza el reconocimiento de emociones en los dos escenarios emergentes. En el primer escenario es necesario reconocer la emoción de tristeza, y en el otro es

necesario reconocer la emoción de alegría. Esto facilita la aparición de comportamientos emergentes en el sistema, debido a que los estados emocionales del robot están definidos por los valores de los subestados de los robots.

### 3.3.2.5 Evaluación de la calidad de reconocimiento de las emociones de los robots usando *Ar2p*

Ahora, vamos a analizar la calidad del reconocimiento de las emociones usando *Ar2p*, y luego, el performance del comportamiento emergente generado por las emociones, desde el punto de vista cualitativo y cuantitativo. Para probar el algoritmo *Ar2p* en el contexto del sistema multi-robot se tiene:

- La emoción de un robot está representada por el vector ['BS-X', 'OS-Y', 'SS-Z', 'IS-W']. El primer elemento es el subestado (BS para batería, OS para operación, y así para el resto) y X, Y, Z y W representan los valores aleatorios asignados, normalizados en el conjunto [0,1].
- La etiqueta de los valores aleatorios (X, Y, Z, W) de los descriptores (subestados) se define como: para los valores en el intervalo [0 a 0.4], las etiquetas son LBP, LOP, LSP y LIP. Para los valores en el intervalo [0.5 a 0.6], las etiquetas son NBP, NOP, NSP y NIP. Para los valores en el intervalo [0.7 a 1], las etiquetas son HBP, HOP, HSP y HIP.

Por ejemplo: para una entrada = ['BS-0.1', 'OS-0.1', 'SS-0.7', 'IS-0.4'] significa que el robot tiene el siguiente estado: LBP, LOP, HSP y LIP, que describe el estado emocional del robot R' = [LBP, LOP, HSP, LIP]. La Tabla 3.24 muestra este proceso:

**Tabla 3.24.** Ejemplo de asignación e instanciación de un estado en un robot

Entrada= Generación aleatoria	R' = Instanciación
BS-0.1	LBP
OS-0.1	LOP
SS-0.7	HSP
IS-0.4	LIP

Con esta información de entrada, el algoritmo *Ar2p* puede ser ejecutado. Las métricas para medir el desempeño de *Ar2p* son las clásicas en problemas de reconocimiento: precisión (Mp), Recall (Mr) y F-measure (F1). Se crean 10 robots, insertando valores aleatorios en sus descriptores: BS, OS, SS, IS. La Figura 3.15 muestra un ejemplo de 10 robots creados al azar.

En total, se realizaron 80 simulaciones para el reconocimiento de emociones: 40 de emoción de tristeza y 40 de emoción de alegría usando las siguientes bases de datos (Gil, Aguilar, Puerto y Dapena 2018; Gil, Aguilar, Puerto y Dapena 2019):

```

Robot 0 = ['BS-0.8', 'OS-0.5', 'SS-0.6', 'IS-1.0']
Robot 1 = ['BS-0.9', 'OS-0.8', 'SS-0.9', 'IS-0.7']
Robot 2 = ['BS-1.0', 'OS-0.6', 'SS-0.5', 'IS-0.9']
Robot 3 = ['BS-0.5', 'OS-1.0', 'SS-1.0', 'IS-0.9']
Robot 4 = ['BS-0.9', 'OS-0.6', 'SS-1.0', 'IS-0.8']
Robot 5 = ['BS-1.0', 'OS-0.6', 'SS-0.9', 'IS-1.0']
Robot 6 = ['BS-0.9', 'OS-1.0', 'SS-1.0', 'IS-0.7']
Robot 7 = ['BS-0.9', 'OS-0.7', 'SS-1.0', 'IS-0.9']
Robot 8 = ['BS-0.9', 'OS-1.0', 'SS-1.0', 'IS-1.0']
Robot 9 = ['BS-0.7', 'OS-0.7', 'SS-0.9', 'IS-0.8']

```

**Figura 3.15.** 10 robots, con valores aleatorios en sus descriptores: BS, OS, SS, IS.

- Base de datos usada para el reconocimiento de la emoción de *alegría*:

```

{'BS-0.1', 'OS-0.1', 'SS-0.7', 'IS-0.4'}, {'BS-0.2', 'OS-0.2', 'SS-0.4', 'IS-0.4'}, {'BS-0.2', 'OS-0.2', 'SS-0.5', 'IS-0.4'}, {'BS-0.4', 'OS-0.4', 'SS-0.7', 'IS-0.2'}, {'BS-0.3', 'OS-0.4', 'SS-0.8', 'IS-0.4'}, {'BS-0.2', 'OS-0.2', 'SS-0.5', 'IS-0.4'}, {'BS-0.1', 'OS-0.4', 'SS-0.8', 'IS-0.4'}, {'BS-0.1', 'OS-0.1', 'SS-0.4', 'IS-0.1'}, {'BS-0.4', 'OS-0.4', 'SS-0.8', 'IS-0.2'}, {'BS-0.2', 'OS-0.3', 'SS-1.0', 'IS-0.4'}, {'BS-0.1', 'OS-0.3', 'SS-1.0', 'IS-0.3'}, {'BS-0.4', 'OS-0.1', 'SS-0.9', 'IS-0.1'}, {'BS-0.3', 'OS-0.1', 'SS-0.8', 'IS-0.3'}, {'BS-0.3', 'OS-0.1', 'SS-0.6', 'IS-0.1'}, {'BS-0.2', 'OS-0.3', 'SS-0.5', 'IS-0.1'}, {'BS-0.2', 'OS-0.1', 'SS-0.8', 'IS-0.2'}, {'BS-0.3', 'OS-0.4', 'SS-0.5', 'IS-0.2'}, {'BS-0.3', 'OS-0.2', 'SS-0.7', 'IS-0.3'}, {'BS-0.1', 'OS-0.1', 'SS-0.8', 'IS-0.3'}, {'BS-0.3', 'OS-0.1', 'SS-0.5', 'IS-0.2'}, {'BS-0.3', 'OS-0.1', 'SS-0.8', 'IS-0.4'}, {'BS-0.1', 'OS-0.1', 'SS-0.4', 'IS-0.4'}, {'BS-0.3', 'OS-0.2', 'SS-0.6', 'IS-0.3'}, {'BS-0.3', 'OS-0.4', 'SS-0.5', 'IS-0.1'}, {'BS-0.2', 'OS-0.3', 'SS-1.0', 'IS-0.1'}, {'BS-0.1', 'OS-0.4', 'SS-0.6', 'IS-0.2'}, {'BS-0.4', 'OS-0.3', 'SS-0.9', 'IS-0.2'}, {'BS-0.1', 'OS-0.2', 'SS-0.4', 'IS-0.1'}, {'BS-0.1', 'OS-0.2', 'SS-0.8', 'IS-0.3'}, {'BS-0.1', 'OS-0.2', 'SS-0.8', 'IS-0.1'}, {'BS-0.2', 'OS-0.4', 'SS-0.5', 'IS-0.1'}, {'BS-0.3', 'OS-0.4', 'SS-0.9', 'IS-0.3'}, {'BS-0.4', 'OS-0.3', 'SS-1.0', 'IS-0.4'}, {'BS-0.1', 'OS-0.1', 'SS-0.5', 'IS-0.1'}, {'BS-0.3', 'OS-0.3', 'SS-0.9', 'IS-0.1'}, {'BS-0.4', 'OS-0.2', 'SS-0.6', 'IS-0.4'}, {'BS-0.3', 'OS-0.4', 'SS-1.0', 'IS-0.1'}, {'BS-0.2', 'OS-0.4', 'SS-1.0', 'IS-0.4'}, {'BS-0.4', 'OS-0.4', 'SS-0.5', 'IS-0.3'}, {'BS-0.3', 'OS-0.3', 'SS-0.8', 'IS-0.3'}

```

- Base de datos usada para el reconocimiento de la emoción de *tristeza*:

{['BS-0.8', 'OS-0.8', 'SS-0.9', 'IS-0.7'], ['BS-0.5', 'OS-1.0', 'SS-0.9', 'IS-0.8'], ['BS-0.8', 'OS-0.5', 'SS-0.8', 'IS-0.7'], ['BS-0.8', 'OS-0.6', 'SS-1.0', 'IS-0.9'], ['BS-0.7', 'OS-0.5', 'SS-0.9', 'IS-0.7'], ['BS-0.5', 'OS-0.9', 'SS-0.6', 'IS-0.8'], ['BS-0.5', 'OS-0.5', 'SS-1.0', 'IS-0.9'], ['BS-1.0', 'OS-0.7', 'SS-0.6', 'IS-1.0'], ['BS-0.8', 'OS-0.7', 'SS-0.7', 'IS-0.9'], ['BS-0.7', 'OS-0.5', 'SS-0.7', 'IS-1.0'], ['BS-0.5', 'OS-0.5', 'SS-0.6', 'IS-0.9'], ['BS-0.8', 'OS-1.0', 'SS-0.5', 'IS-0.7'], ['BS-0.7', 'OS-0.5', 'SS-0.8', 'IS-1.0'], ['BS-1.0', 'OS-1.0', 'SS-0.8', 'IS-0.8'], ['BS-0.6', 'OS-1.0', 'SS-1.0', 'IS-0.9'], ['BS-0.9', 'OS-0.8', 'SS-0.7', 'IS-1.0'], ['BS-0.8', 'OS-1.0', 'SS-0.8', 'IS-1.0'], ['BS-0.5', 'OS-0.6', 'SS-0.7', 'IS-0.9'], ['BS-0.9', 'OS-0.7', 'SS-0.6', 'IS-0.8'], ['BS-0.7', 'OS-1.0', 'SS-0.7', 'IS-0.8'], ['BS-0.7', 'OS-0.6', 'SS-0.8', 'IS-0.9'], ['BS-1.0', 'OS-1.0', 'SS-0.6', 'IS-1.0'], ['BS-0.9', 'OS-0.6', 'SS-0.6', 'IS-0.8'], ['BS-0.6', 'OS-0.8', 'SS-0.6', 'IS-1.0'], ['BS-0.5', 'OS-0.6', 'SS-0.7', 'IS-0.9'], ['BS-0.6', 'OS-1.0', 'SS-0.5', 'IS-0.9'], ['BS-0.7', 'OS-0.6', 'SS-0.9', 'IS-0.7'], ['BS-0.8', 'OS-1.0', 'SS-0.8', 'IS-0.9'], ['BS-0.6', 'OS-1.0', 'SS-0.7', 'IS-0.9'], ['BS-1.0', 'OS-0.6', 'SS-0.5', 'IS-0.9'], ['BS-0.5', 'OS-0.8', 'SS-0.9', 'IS-0.8'], ['BS-0.5', 'OS-0.7', 'SS-0.5', 'IS-0.8'], ['BS-0.8', 'OS-0.9', 'SS-0.7', 'IS-0.9'], ['BS-0.6', 'OS-0.7', 'SS-0.5', 'IS-0.9'], ['BS-0.8', 'OS-0.7', 'SS-0.8', 'IS-0.7'], ['BS-1.0', 'OS-0.5', 'SS-0.7', 'IS-0.8'], ['BS-1.0', 'OS-0.8', 'SS-1.0', 'IS-0.7'], ['BS-0.8', 'OS-0.6', 'SS-0.8', 'IS-0.8'], ['BS-0.7', 'OS-0.7', 'SS-0.6', 'IS-1.0'], ['BS-0.7', 'OS-1.0', 'SS-0.5', 'IS-0.7']}

De acuerdo con los resultados (ver Tabla 3.25), la *precision* = 1 determina que *Ar2p* reconoce los patrones, sin hacer otros reconocimientos inesperados. Este valor de precisión se debe a que *Ar2p* maneja situaciones muy específicas y únicas. Esto significa que un robot puede reconocer una emoción en múltiples variaciones de los estados de las variables que la caracterizan. *Recall* = 1 indica que *Ar2p* puede descubrir todos los patrones emocionales. Esto significa que cada robot reconoce todas las emociones de otros robots. *F-measure* = 1 significa que cada robot tendrá la capacidad de reconocer con precisión cualquier emoción.

**Tabla 3.25.** Capacidad de reconocimiento de emociones en un robot usando *Ar2p*

Métrica/Emoción	Precision	Recall	F-measure
Tristeza	1	1	1
Alegría	1	1	1

De acuerdo con nuestros dos escenarios, el reconocimiento de estas emociones determina el comportamiento colectivo de los robots. Por ejemplo, en el caso del primer escenario, los robots que están tristes son informados sobre un sitio de recarga. En el segundo caso, el reconocimiento de una emoción positiva (por ejemplo, la alegría) determina que los robots pueden ser reclutados. Esto facilita los comportamientos emergentes en el sistema, porque el comportamiento de los

robots no se puede predecir a priori. El componente emocional depende del estado de un robot, que define cuándo y cómo puede actuar. Por ejemplo, un robot que está "triste" no explorará el entorno de la misma manera que un robot que está "Alegre".

En términos de la eficiencia y facilidad de uso del algoritmo *Ar2p*, se implementa en la nube, donde es invocado por cada robot. Además, los estados de los robots se almacenan en una memoria compartida, para ser utilizados por los otros robots en su proceso de reconocimiento.

Con el propósito de validar la influencia de las emociones en el comportamiento emergente del sistema multi-robot, consideramos varios métodos de verificación; uno llamado MASOES, otro con análisis a través de simulaciones, y usando una métrica de entropía (ver (Gil, Aguilar, Puerto, y Dapena, 2018) para más detalles del proceso de verificación).

A modo de conclusión, podemos decir que una de las principales características de nuestro modelo emocional para sistemas multi-robot es que permite el reconocimiento de las emociones, para generar comportamientos emergentes, lo que le da una gran flexibilidad al sistema para ejecutar diferentes tareas. La inclusión de emociones en sistemas multi-robot ayuda a su adaptación en entornos dinámicos, mejorando su toma de decisiones en un momento determinado. La emoción define la predisposición de un robot en un momento dado, lo que determina cómo puede responder a un estímulo, facilitando la emergencia y la autoorganización en el sistema.

A través de *Ar2p* podemos probar las interacciones entre los robots en función de la emoción reconocida/percibida, y la influencia en el comportamiento colectivo del sistema. Se puede usar *Ar2p* para que cada robot reconozca los estados emocionales (es una habilidad de los robots) y, de acuerdo con su percepción sobre el estado emocional de los robots a su alrededor, actuar (recluta, informa, etc.). A nivel computacional, el modelo propuesto ofrece una solución concisa, legible y elegante, basada en la recursividad.

Se han propuesto dos escenarios. El primero para informar el lugar para recargar las baterías, para los robots que necesitan recargar la batería (robots tristes); y el otro para el reclutamiento de robots para mover un objeto (robots alegres).

Los próximos trabajos se dedicarán a extender este modelo considerando más emociones, más situaciones (escenarios), etc. para probar la escalabilidad de nuestro enfoque. La extensión del modelo emocional permitirá lograr una mayor aproximación al espectro emocional humano, por lo tanto, la inclusión de estados emocionales más complejos, y así, estudiar su influencia en el comportamiento colectivo del sistema. Finalmente, como trabajo futuro, se llevarán a cabo experimentos con robots reales, que siguen la misma arquitectura propuesta en (Gil, Aguilar, Rivas, Dapena y Hernandez 2018).

### **3.3.3 Reconocimiento de texto usando Ar2p**

La siguiente prueba de *Ar2p* se enfoca en el reconocimiento automático de palabras mal escritas, específicamente, en el reconocimiento de errores sintácticos en un texto escrito en español. *Ar2p* se aplica en este escenario como un sistema corrector, que hemos denominado *Ar2p-Text*. *Ar2p-Text* determina si una palabra está bien o mal escrita (Puerto, Aguilar y Rojas 2019). Una evaluación comparativa muestra que la precisión y cobertura de *Ar2p-Text* es competitivo con el corrector ortográfico SpanishChecker®, STILUS® y el mecanismo de corrección de Microsoft Word. En general, *Ar2p-Text* obtiene una F-measure del 83%, muy por encima del 73% alcanzado por los otros correctores ortográficos.

Actualmente, hay varios sistemas que realizan análisis, extracción, anotación y corrección lingüística (basados en diccionarios o en análisis estadísticos) para realizar tareas tan diversas como la lematización, el etiquetado morfosintáctico, o el análisis sintáctico, entre otros, que

incluyen aplicaciones para el análisis de sentimientos (o minería de opinión), la extracción o anotación conceptual, etc. (Valdehíta 2016; Ferreira y Hernández 2017; Gamallo y Garcia 2017). Valdehíta (Valdehíta 2016) propone un algoritmo de corrección ortográfica y gramatical donde los posibles errores se detectan mediante el análisis estadístico, comparando combinaciones de dos palabras contra un corpus de cien millones de palabras. Ferreira (Ferreira y Hernández 2017) propone un corrector ortográfico de patrones de error complejos, palabra por palabra, y en conjunto. Por otro parte, en el corrector STILUS (Villena, González, González y Muriel 2002) se distinguen cuatro tipos de errores: gramaticales, ortográficos, semánticos y de estilo, que detecta a través de módulos específicamente dedicados a cada uno de ellos. ArText, es un sistema de apoyo automático para la escritura de textos en español en dominios especializados (da Cunha, Montané, y Hysa 2017). El sistema tiene tres módulos: un primer módulo maneja aspectos de estructura, contenido y fraseología, un segundo módulo se encarga del formato y revisión lingüística, y finalmente, un módulo realiza la revisión lingüística del texto.

Esta sección continua con los experimentos para el tratamiento de textos, los módulos de reconocimiento utilizados, las métricas de calidad, y la evaluación de desempeño. Finalmente, se presenta conclusiones y futuras líneas de investigación sobre este dominio de aplicación.

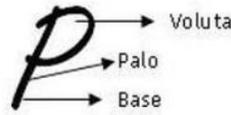
### **3.3.3.1 Proceso general de Ar2p-Text para reconocer una palabra**

A continuación, se describe el proceso y la evaluación de *Ar2p-Text* en el proceso de reconocimiento de patrones de texto. Un módulo de reconocimiento de patrones en este dominio reconoce patrones, tales como: frases, letras, palabra. Para esto, se tomaron como datos de prueba muestras aleatorias de textos escritos en blogs en español (<http://www.elmundo.es/lo-mas/blogs-mas-leidos.html>), y las métricas de *Precisión* (P), *Cobertura* (C) y *F-measure* (F) (Powers 2011).

Las palabras son reconocidas por *Ar2p-Text* a través de los módulos de reconocimiento especializados, que se activan con las señales reconocidas, usando unas estrategias de verificación /corrección/reconocimiento, tal como se describen a continuación: Sea "PERA" la palabra o patrón de entrada desconocido del texto. En este caso, las palabras, letras y formas son patrones. Las formas son tratadas como patrones atómicos. Se asumen tres niveles de reconocimiento en la jerarquía propuesta (Figura 3.1), un primer nivel  $X_1$  tiene los módulos  $\Gamma_{j1}$  de reconocimiento para patrones atómicos (líneas o formas); un segundo nivel  $X_2$  los módulos  $\Gamma_{j2}$  de reconocimiento para patrones de letras que forman las palabras del texto (por ejemplo, "P", "E", "R", "A", y en general, el alfabeto usado en el texto). Finalmente, el nivel  $X_3$  tiene los módulos de reconocimiento de patrones  $\Gamma_{j3}$  encargados de reconocer las palabras del texto (por ejemplo, "MANZANA", "PERA", y en general, el vocabulario: sustantivos, verbos, etc.). Establecido lo anterior, y partiendo del patrón de entrada  $y = s() = \text{"PERA"}$ , se inicia el proceso de reconocimiento:

Proceso *Top-Down*. El algoritmo recibe como entrada  $y = s() = \text{"PERA"}$  (Línea 1). Luego, se realiza el proceso de descomposición de la palabra (Línea 2) en sus sub-patrones  $\langle P, E, R, A \rangle$ . Posteriormente, con esta información se calcula el nivel  $X_i$  donde inicia el proceso de reconocimiento de  $y$ , en este caso  $X_{i=3}$ . Seguidamente, son creadas L solicitudes de reconocimiento de  $y \langle y_1=P, y_2=E, y_3=R, y_4=A \rangle$ . Si los patrones iniciales a reconocer son patrones atómicos (i.e., del nivel  $X_1$ ; líneas, bordes, etc.), entonces se realizará el proceso Bottom-Up (Líneas 5 a la 21). Pero como  $y$  no son patrones atómicos (es  $X_{i=3}$ ), entonces se envían las L solicitudes de reconocimiento de ( $y$ ) al nivel inferior  $X_2$  (línea 25). En este punto se inicia el proceso recursivo de reconocimiento, para cada uno de los  $y_i$  de la lista L (las señales de salida de petición de  $X_3$ , se convierten en señales de entrada para el nivel  $X_2$ ). Para el caso de reconocimiento de  $y_1=P$ ,  $s()$  es la letra ("P"), y el algoritmo carga un nuevo proceso de reconocimiento de  $s()$  para el nivel  $X_{m=2}$ .

Nuevamente, un proceso de descomposición es realizado para la letra P (línea 2) en sus subpatrones, en este caso, se realiza la división de P en tres partes para facilitar su interpretación (ver Figura 3.16).  $\langle y_1=\text{voluta}^{11} \mid, y_2=\text{palo}, y_3=\text{base} \rangle$



**Figura 3.16.** Partes de una P.

Luego, es determinado el nivel de la jerarquía donde inicia el proceso de reconocimiento para este patrón, el cual es ( $X_2$ ). El algoritmo genera L solicitudes de reconocimiento para  $y_1$ ,  $y_2$  y  $y_3$  (Línea 4), y espera (Línea 26). Cuando es reconocido  $y_1 = \text{'parte de voluta'}$ , el algoritmo determina que es un patrón atómico (en el nivel  $X_1$ ), entonces el proceso Bottom-Up es realizado. Particularmente, si el cálculo fue éxito (i.e., hay un módulo  $\Gamma$  que reconoce a éste, línea 7), y no es el último nivel de la jerarquía (línea 8), entonces el algoritmo crea y envía una  $S_0$  al nivel superior ( $X_2$ ). Esto es similar para el resto de patrones atómicos. Se asume que los patrones atómicos restantes fueron reconocidos ( $s()=\text{palo}$  y  $s()=\text{base}$ ), las respuestas ascienden en el nivel de la jerarquía, para modificar el estado de la señal correspondiente en el módulo de reconocimiento de P. Cuando estas señales son recibidas en  $X_2$  (generadas en  $X_1$  línea 26), el algoritmo calcula el reconocimiento del patrón en  $X_2$  con base a las señales ( $y$ ) reconocidas en el nivel inferior (línea 27). Se asume que el reconocimiento del patrón P fue exitoso (por alguno de los dos teoremas 3.1, o 3.2) en el nivel  $X_2$ , y puesto que no se encuentra en el último nivel de la jerarquía (es  $X_2$ ), el algoritmo crea y envía una señal de reconocimiento de "P" al nivel superior (líneas 33 y 34). En otras palabras, ocurre un proceso de difusión de la señal reconocida ("P") a los niveles superiores que la contiene. El mismo análisis de reconocimiento de "P" aplica para las demás letras "E", "R", "A". Cuando son

<sup>11</sup> la voluta es la curva que realizamos en lo alto de la letra

reconocidos los patrones de los módulos del nivel 2, i.e., las letras P,E,R,A, de las solicitudes hechas en el nivel 3, este nivel recibe las respuestas (línea 26), y calcula el reconocimiento con las señales que fueron reconocidas (línea 27). Se asume que el módulo de reconocimiento donde se activaron todas estas señales es éxito (línea 28), y dado que ahora el proceso principal se encuentra en el último nivel de la jerarquía  $X_3$  (i.e., nivel donde inicio el reconocimiento, línea 3), entonces se realiza un aprendizaje por reforzamiento (ver línea 29), y luego se genera la señal de salida  $S_o$ , que se convierte en la salida general del sistema “Patrón reconocido <PERA>”.

Veamos resumidamente el uso de los teoremas y fórmulas usados en el proceso *Top-Down* de reconocimiento, considerando una instanciación del módulo de reconocimiento de la palabra  $\Gamma' =$  “PERA” (ver Tabla 3.26), bajo algunos valores de sus pesos de importancia, umbrales y dos niveles de la jerarquía:  $X_1$  para letras (que pasará a ser el nivel atómico), y  $X_2$  para palabras, (el cual pasará a ser el nivel más alto de la jerarquía). Sea  $\Delta U_1 = 0.8$  el umbral de reconocimiento a superar por señales claves del módulo  $\Gamma'$ , y sean tres de cuatro señales reconocidas en  $X_1$  (su estado es “True”, ver Tabla 3.30, señales 1, 2 y 3). Ahora, se aplica el Teorema 1. Para ello, se necesita determinar las señales clave activas, utilizando la Ecuación (3.1).

Las señales clave de este patrón son: señales 1, 2 y 3. Luego, usando la Ecuación (3.2) para saber si con estas señales es posibles reconocer el patrón PERA relacionado al módulo  $\Gamma'$ , al aplicar esta ecuación vemos que con las señales activas son suficientes para superar el umbral  $\Delta U_1=0.8$  ( $0,83 > 0,8$ ). Por tanto, el reconocimiento del patrón es exitoso y la señal de salida del sistema es "patrón reconocido <Pera>".

**Tabla 3.26.** Módulo de reconocimiento de patrón palabra “PERA”

E				
S		C		
Señal	Estado	Descriptor (D)	Dominio (V)	Peso (P)
1	True	P	< posibles formas de P >	0.9
2	True	E	< posibles formas de E >	0.8
3	True	R	< posibles formas de R >	0.8
4	False	A	< posibles formas de A >	0.5
< $\Delta U1=0.8, \Delta U2=0.6$ >				

Proceso *Bottom-Up*. Veamos ahora el proceso en el que el patrón de entrada es un conjunto de patrones atómicos. Por ejemplo: sea  $s() = \langle \text{"r"}, \text{"-"}, \text{"\textbackslash"} \rangle$ . Si intuitivamente juntamos estas líneas de la forma “/-\” vemos como emerge el patrón “A”. En este caso,  $s()$  no es un patrón complejo que requiere descomposición, de ahí que L es cargado con el contenido de  $s()$  (Línea 2). Dependiendo del contenido de L, es determinado el nivel de inicio de reconocimiento (en este caso  $X_1$ ). Entonces el algoritmo continúa creando L peticiones de reconocimiento de  $(y)$  (línea 4). Como ellos son señales de más bajo nivel  $X_{m=1}$ , el reconociendo es calculado individualmente para cada patrón  $(y)$  (ver líneas 5 a 23). Luego, las señales de reconocimientos son difundidas o enviados al nivel superior  $X_m$ , en este caso  $X_{m=2}$ . Recibidas las señales por los módulos de reconocimiento de  $X_{m=2}$  (i.e., actualizados los estados de estas señales en los módulos de reconocimiento de este nivel que las contiene), se procede a calcular si hay o no reconocimiento por los módulos. Sea  $\Delta U1 = 0.85$  el umbral de reconocimiento a superar por señales claves, y todas las señales están activas en los módulos de  $X_2$  (sus estados son “True”, ver Tabla 3.27). Ahora, aplicamos el Teorema 1. Para ello, necesitamos determinar las señales clave activas utilizando la Ecuación (3.1).

**Tabla 3.27.** Módulo de reconocimiento de patrón letra  $\Gamma_2 = "A"$ 

E				
S		C		
Señal	Estado	Descriptor (D)	Dominio (V)	Peso (P)
1	True	/	< posibles formas de />	0.9
2	True	-	< posibles formas de - >	0.9
3	True	\	< posibles formas de \>	0.9
< $\Delta U_1=0.85, \Delta U_2=0.6$ >				

Las señales clave de este patrón son: señales 1, 2 y 3. Luego, usamos la Ecuación (3.2) para saber si con estas señales es posibles reconocer el patrón "A", al aplicar esta ecuación, se observa que con las señales activas son suficientes para superar el umbral  $\Delta U_1=0.8$  ( $0,9>0,8$ ). Por tanto, el reconocimiento del patrón es exitoso y la señal de salida del sistema es "patrón reconocido <A>".

### 3.3.3.2 Tratamiento de tipos de errores ortográficos usando Ar2p-Text

Esta sección muestra cómo funciona *Ar2P-Text* en la corrección de palabras mal escritas, ya sea por omisión de letras (por ejemplo, en lugar de escribir "limpia" se escribe "lipia"), por la agregación de letras de más (por ejemplo, en lugar de escribir "salir" se escribe "salire"), por la confusión de letras (por ejemplo, en lugar de escribir "campesinos" se cambia 'c' por 'p', y se escribe "pampecinos"), por la confusión de letras con una orientación simétrica similar (como 'd-b', 'p-q-g', por ejemplo, en lugar de escribir "Debe", se escribe "dede", o "zagato" por "zapato"), o por último, por la inversión de letras (por ejemplo, en lugar de escribir "juega" se escribe "gueja"). A continuación, se describe y se compara la capacidad de corrección de Ar2p frente a otros modelos de corrección ortográfica tratando estos tipos de errores. Como paso previo, la cadena de texto se divide en palabras para su procesamiento, y los errores reales aparecerán marcados en negrita y subrayados.

### 3.3.3.2.1 Corrección de palabras por adicción de letras

Este caso considera las palabras mal escritas que tienen letras de más en cualquier parte de la palabra, y se corrigen con *Ar2p-Text* en función de los módulos de nivel superior que se activan con las señales reconocidas. Por ejemplo, si la palabra mal escrita es "Ambrousio", donde se incluyó una letra 'u' demás, se corregirá ya sea por señales claves o parciales mediante la salida del módulo de reconocimiento  $\Gamma$ ="Ambrosio". A continuación, se revisa con los correctores ortográficos un párrafo que tiene el 5% de las palabras con ese error.

Lista (L1) = { "La **entrdada**\_de Isabel Pantoja en "Sálvame" ocurrida la semana pasada, fue la guinda a meses de silencio de la tonadillera. Una representación que merecería un Goya a una cantante que ha hecho de su vida el mejor melodrama de la historia de la **televisión** española. No es la primera vez que Isabel coge el **teléfonoo** y llama a un programa para soltar toda la bilis que lleva dentro, toda la angustia que **ssiente** cada vez que ve a su hija Chabelita pasearse por los platós, contando **laas** miserias de una familia que hace tiempo decidió que es mejor vivir..." }

De acuerdo con los resultados de la Tabla 3.28, *Ar2p-Text* y *STILUS®* detectan los errores reales, y *Microsoft Word* y *SpanishChecker* detectan incorrectamente otros errores. Por ejemplo, *Microsoft Word* detecta incorrectamente la palabra "Chabelita", y *SpanishChecker* lo hace con "Goya" y "Pantoja". *Ar2p-Text* y *STILUS®* reconocen las palabras "Chabelita", "Goya", y "Pantoja".

**Tabla 3.28.** Resultados de los diferentes sistemas de corrección ortográfica para L1

L2	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
<i>Ar2p-Text</i>	5	0	0	100%	100%	100%
<i>SpanishChecker®</i>	8	0	3	72%	100%	84%
<i>STILUS®</i>	5	0	0	100%	100%	100%
<i>Microsoft Word</i>	6	0	1	85%	100%	92%

### 3.3.3.2.2. Correcciones de palabras por omisión de letras

Este caso considera las palabras mal escritas por omisión de letras en cualquier parte de la palabra, y se corrigen con *Ar2p-Text* en función de los módulos de nivel superior que se activan con las señales reconocidas. Por ejemplo, si la palabra mal escrita es "mbrosio", donde se omitió la primera letra del nombre "A", se corregirá ya sea por señales claves o parciales, mediante la salida del módulo de reconocimiento  $\Gamma = \text{"Ambrosio"}$ . A continuación, se revisa con los correctores ortográficos un párrafo que tiene el 5% de las palabras con este tipo de error.

Lista (L2) = {“El expresidente Aznar, hoy en el **Congres**. La derecha sin complejos, efectivamente. Pero no lo que la izquierda quiere que se entienda con ello, es decir, la reacción y la caspa sin maquillajes; simplemente una **dereca** que no se acompleja cuando la izquierda pretende someterla a su habitual tratamiento de superioridad moral. El **moment** con el líder de la **podemia** fue, en este sentido, una interesante lección. Se presentó Pablo Iglesias con su aire habitual de fiscal soviético, **disiulando** la falsedad fáctica de la mayoría de sus **pregutas** con una dicción seria y reposada, más reposada aún desde la toma de hipoteca...” }

La Tabla 3.29 muestra los resultados de los diferentes correctores. *Ar2p-Text*, Microsoft Word y STILUS® detectan los errores reales, mientras que SpanishChecker detecta incorrectamente el patrón “expresidente”, “Aznar”, y omite “Congres”. SpanishChecker® incorrectamente detecta más errores.

**Tabla 3.29.** Resultados de los diferentes sistemas de corrección ortográfica para L2

L2	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
Ar2p-Text	6	0	0	100%	100%	100%
SpanishChecker®	7	1	1	87%	87%	87%
STILUS®	6	0	0	100%	100%	100%
Microsoft Word	6	0	0	100%	100%	100%

### 3.3.3.2.3 Correcciones de palabras por letras cambiadas.

Este caso considera las palabras mal escritas por el cambio de letras en cualquier parte de la palabra, las cuales se corrigen con *Ar2p-Text* en función de los módulos de reconocimiento de

nivel superior que se activan con las señales reconocidas. Por ejemplo, si la palabra mal escrita es "ambrocio", Ar2p-Text lo reconoce y corrige, ya sea por señales claves o parciales a través del módulo de reconocimiento activado  $\Gamma$ ="Ambrosio", cambiando la letra "c" por "s", y pasando la primera letra a mayúsculas. Como este error es muy frecuente, a continuación, se revisa con los correctores ortográficos un párrafo del que incrementalmente se tomará un número de errores, comenzando con el 5% de errores de este tipo, el 15% y el 20%.

Lista (L3) = { “La **nueba** pensión [donde vivía] era, mayormente, un antro de homosexuales (...) Mi única salida era bajarme a un **sine** de sesión continua y programa doble, a pasar la tarde, hasta que se fuese toda aquella gente nauseabunda. Eran tan **ginos** que me daban asco. La **homosexualidad**, por mucha literatura que se le haya puesto alrededor, a mí siempre me ha olido a mierda, y contra eso no hay nada que hacer...” }

La Tabla 3.30 muestra los resultados para los diferentes correctores, considerando sólo el 5% de los errores. En este caso, Ar2p-Text, SpanishChecker® y STILUS® detectan los errores. Microsoft Word no detecta uno de los errores.

**Tabla 3.30.** Resultados de los diferentes sistemas de corrección ortográfica para L3

L3	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
Ar2p-Text	4	0	0	100%	100%	100%
SpanishChecker®	4	0	0	100%	100%	100%
STILUS®	4	0	0	100%	100%	100%
Microsoft Word	3	1	0	100%	75%	86%

La Tabla 3.31 muestra los resultados para los diferentes correctores, considerando el 15% de los errores. En este caso, los resultados de Ar2p-Text, SpanishChecker® y STILUS® son similares a los resultados anteriores. Microsoft Word detecta incorrectamente más errores, y no detecta otros.

**Tabla 3.31.** Resultados de los diferentes sistemas de corrección ortográfica para L3

L3	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
Ar2p-Text	11	0	0	100%	100%	100%
SpanishChecker®	11	0	0	100%	100%	100%
STILUS®	11	0	0	100%	100%	100%
Microsoft Word	11	1	1	91%	91%	91%

La Tabla 3.32 muestra los resultados para los diferentes correctores, considerando el 20% de los errores. En este caso, los resultados de Ar2p-Text, SpanishChecker® y STILUS® son similares a los resultados anteriores.

**Tabla 3.32** Resultados de los diferentes sistemas de corrección ortográfica para L3

L3	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
Ar2p-Text	15	0	0	100%	100%	100%
SpanishChecker®	15	0	0	100%	100%	100%
STILUS®	15	0	0	100%	100%	100%
Microsoft Word	15	1	1	93%	93%	93%

#### 3.3.3.2.4 Corrección de palabras por uso de caracteres especiales.

Este caso considera las palabras mal escritas por uso de dígitos o caracteres especiales en cualquier parte de la palabra, por ejemplo, en cas@, C@rpinteria, P!no, etc., y se reconocen con Ar2p-Text de la misma manera que las correcciones anteriores. La Tabla 3.33 muestra los resultados de los diferentes correctores para un párrafo con el 5% de palabras con esta escritura. En este caso, solo Ar2p-Text puede detectar y reconocer estas palabras, muy similar como lo hace una persona. Los demás sistemas, SpanishChecker®, STILUS® y Microsoft Word, no tienen la capacidad para hacerlo, o los reconocen incorrectamente (STILUS®).

**Tabla 3.33.** Resultados de los diferentes sistemas de corrección ortográfica para L4

L4	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
Ar2p-Text	5	0	0	100%	100%	100%
SpanishChecker®	1	4	0	100%	20%	33%
STILUS®	1	3	1	50%	25%	33%
Microsoft Word	2	3	0	100%	40%	57%

### 3.3.3.2.5. Corrección de palabras por intercambio de letras.

Este caso contempla las palabras mal escritas por el intercambio de dos letras, por ejemplo, “Libor” por “Libro”, y se corrigen con *Ar2p-Text* de la misma manera que las correcciones anteriores. La Tabla 3.34 muestra los resultados de los diferentes correctores para un párrafo con el 5% de palabras con este error. En este caso, todos los correctores dan falsos positivos i.e., reconocen palabras que no son. Más sin embargo, Ar2p-Text solo se equivoca tres veces, mientras que los demás sistemas cometen más errores de imprecisión.

**Tabla 3.34.** Resultados de los diferentes sistemas de corrección ortográfica para L4

L4	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
Ar2p-Text	8	0	3	72%	100%	83%
SpanishChecker®	11	0	6	64%	100%	78%
STILUS®	13	0	8	61%	100%	75%
Microsoft Word	14	0	9	60%	100%	75%

### 3.3.3.2.6 Comparación más sólida, conclusiones de resultados y trabajos futuros

En anteriores pruebas, hemos predefinido los errores a detectar. Para hacer una comparación más completa, consideramos un archivo de texto con 9,000 palabras tomadas de 32 textos de la versión digital del periódico "El País" del 17 de mayo de 2001. Estos datos se usaron en STILUS®. Los resultados se resumen en la Tabla 3.35. Como se puede ver, SpanishChecker® es el sistema que

ha detectado menos errores (11/23 Errores). Microsoft Word y Stilus® son más efectivos que SpanishChecker®, ya que detectan 13/23 y 12/23 errores. Nuestro enfoque detecta todos los errores, 23/23, aunque también detecto incorrectamente 9 errores más. En general, los resultados de los otros están lejos de los resultados logrados por Ar2p-text. Particularmente, la capacidad de Ar2p-Text para reconocer palabras con caracteres especiales es que se trata de un reconocedor de patrones, y encuentra similitudes entre caracteres especiales y letras, al igual que el cerebro: @=a, E=3, S=5, 9=q, etc.

**Tabla 3.35.** Resultados obtenidos para un archivo de texto con 9000 palabras

L4	Errores detectados	Falsos Negativos	Falsos Positivos	P	C	F
Ar2p-Text	23	0	9	71.5%	100%	83%
SpanishChecker®	56	3	45	57.9%	95%	72%
STILUS®	54	1	41	58.1%	98%	73%
Microsoft Word	51	2	39	58.6%	96%	73%

En esta aplicación, presentamos la descripción general y la arquitectura de *Ar2p-Text*, un sistema de revisión ortográfica para documentos en español basado en el modelo sistemático del funcionamiento del cerebro. *Ar2p-Text* puede detectar diferentes tipos de errores ortográficos: omisión de letras, confusión de letras, cambio de letras, entre otros errores. *Ar2p-Text* utiliza dos estrategias para el proceso de corrección, una basada en señales clave y la otra basada en señales parciales, ambas utilizan umbrales de satisfacción e importancia de los pesos de las señales. El modelo recursivo permite descomponer el problema del reconocimiento de patrones en patrones más simples, lo que permite analizar palabras muy complejas.

Con respecto a las limitaciones del reconocedor de texto *Ar2p-Text*, requiere una definición supervisada de los pesos asignados a las variables utilizadas para el reconocimiento. Por ello, se

ha trabajado en un enfoque de aprendizaje no supervisado para detectar un número creciente de errores lingüísticos, con el fin de mejorar su capacidad de reconocimiento y ampliar su base de conocimientos (ver capítulo 5).

De esta manera, como trabajo futuro, la arquitectura de *Ar2p-Text* debe extenderse con mecanismos de aprendizaje no supervisados, lo que le permitirá mejorar su funcionamiento (aprender nuevas palabras). Además, debe extenderse para utilizar otros idiomas, tal vez simultáneamente. Por ejemplo, *Ar2p-Text* podría corregir simultáneamente textos escritos en inglés y en español, lo que puede ser una tarea de traducción interesante. Para eso, *Ar2p-Text* debe extenderse con más módulos de reconocimiento (su base léxica). es decir, solo necesita definir módulos especializados de reconocimiento de palabras para cada idioma.

### **3.4 Análisis General de Resultados de Ar2p**

A continuación, se describen las capacidades y debilidades computacionales de *Ar2p* con base en los resultados obtenidos en los dominios de aplicación de reconocimiento de texto, reconocimiento de emociones, y reconocimiento de estilos de conducción.

*Ar2p* modela a nivel cerebral el proceso vinculado al reconocimiento y aprendizaje de patrones, que como vimos, es una capacidad destacada en los individuos con características cognitivas diferentes.

Nuestro modelo *Ar2p* de reconocimiento de patrones es altamente recursivo y uniforme. Reconoce los patrones de entrada mediante un proceso de auto-asociación en una jerarquía de patrones. El modelo recursivo de *Ar2p* permite descomponer el problema del reconocimiento de patrones en patrones más simples, lo que permite analizar patrones de entrada sin importar su nivel de complejidad, o la naturaleza de la misma (estática o dinámica, sea imagen, texto, sonido, etc.). Esta capacidad le da robustez.

La uniformidad permite que el proceso recursivo sea el mismo en todos los niveles. Esto, dado que *Ar2p* está basado en los mecanismos funcionales del cerebro humano. Específicamente, se basa en la uniformidad y redundancia de las columnas corticales y su dinámica jerárquica de los módulos de reconocimiento de patrones (mini-columnas). Los estudios de caso analizados permiten ver la uniformidad del proceso de reconocimiento en diversos dominios: reconocimiento de emociones, estilos de conducción, textos como de imágenes.

*Ar2p* reconoce los patrones atómicos dinámicos desde dos orientaciones principales: el reconocimiento de patrones dinámicos orientados por característica, y el reconocimiento de patrones orientado por percepción. Considera para esto una lógica temporal donde está presente el tiempo, específicamente series temporales.

Además, el modelo utiliza dos estrategias para el reconocimiento; una por señales clave y otra por mapeo total o parcial. La primera explota las señales importantes que facilitan el reconocimiento, la segunda estrategia explota completamente las señales de entrada. De esta manera, puede predecir un patrón, incluso con información faltante (utilizando la estrategia de señales clave).

A nivel computacional, el modelo propuesto ofrece una solución concisa, legible y elegante (recursividad), pero define un árbol, que puede ser muy grande. Una posible optimización es usar el paralelismo para mejorar el tiempo de ejecución. Además, un problema principal es definir el nivel atómico para cada patrón (considerado en el siguiente capítulo 4).

En general, el rendimiento de *Ar2p* donde no hay patrones generales para diferentes situaciones, no es bueno (una ventaja de las crónicas), ya que al igual que la lógica difusa necesita una gran base de datos de reglas, que no cubre todas las situaciones posibles. Mas, sin embargo, *Ar2p* puede reutilizar mucha información a través de los módulos de reconocimiento, lo cual le da

una ventaja. Además, puede tratar con conocimiento incierto o no completo. Esto gracias a las estructuras de representación del patrón (es decir, los módulos de reconocimiento de patrones), que, a través de sus parámetros como el peso de importancia de los descriptores, admite diferentes formas o cambios en los descriptores de un patrón.

La versión de *Ar2p* en el contexto del reconocimiento de estilos de conducción destaca a nivel de comunicación, porque sólo necesita enviar la señal sobre el reconocimiento de un patrón dado (por ejemplo, el estado emocional del conductor) a otros vehículos.

Por otra parte, la versión *Ar2p-Text* tiene la capacidad de detectar diferentes tipos de errores ortográficos: omisión de letras, confusión de letras, cambio de letras, inclusión de caracteres especiales semejantes (a, @, etc.), etc. Además, *Ar2p* puede utilizarse para tratar con otros idiomas simultáneamente. Por ejemplo, *Ar2p-Text* podría corregir simultáneamente textos escritos en inglés y en español, lo que puede ser una tarea de traducción interesante.

La diferencia de *Ar2p* y *Wavelet* aplicada al reconocimiento de patrones es que en *Wavelet* se realiza una extracción de descriptores/características/propiedades (*features*) intrínsecos del patrón (reconocimiento de bordes, clasificación de texturas, etc.), mientras que en nuestro modelo se consideran descriptores más generales, los cuales son descompuestos recursivamente.

La importancia del algoritmo *Ar2p* frente a otros métodos de reconocimiento de patrones es que sus cálculos definidos en los teoremas son más simples y distribuidos, y dado que opera sobre una jerarquía, es susceptible de extenderse hacia un tratamiento simultáneo de los niveles (paralelismo). Así, el costo computacional puede ser mejorado con respecto a otros enfoques, además, con un uso más eficiente de la memoria, al contemplar en una sólo estructura de datos abstracta muchas posibles instancias de varios patrones de objetos.

Nuestro modelo de reconocimiento es adaptable porque puede aprender tanto los posibles cambios en los descriptores del patrón (como sus pesos de importancia), como aprender nuevos módulos (componentes) si se conocen los patrones atómicos.

Por otra parte, una limitación de *Ar2p* es que requiere una definición inicial supervisada de las variables de los módulos utilizados para el reconocimiento, en específico, el peso de importancia y el valor de los umbrales. Otra desventaja de *Ar2p* es que no incorpora mecanismos de eliminación de los módulos de reconocimiento que han caído en desuso, por lo tanto, si bien tiene mecanismo de aprendizaje que le permiten aprender, no cuenta con mecanismo que le permitan desaprender, lo que conlleva a subutilizar el espacio de memoria. Finalmente, la actual versión de reconocimiento es secuencial, con llamadas recursivas para reconocer y/o aprender patrones, y requiere definir el nivel atómico para cada patrón. De esta manera, según este árbol, la complejidad computacional puede ser muy grande. Sigüientes trabajos se dedicarán a extender este modelo considerando posibles mejoras, tanto a nivel cerebral como computacional. Por ejemplo, una posible optimización es usar el paralelismo para mejorar el tiempo de ejecución.

## 4. CAPITULO 4: ARQUITECTURA DE DEEP LEARNING PARA Ar2p: Ar2p-DL

### 4.1 Introducción

En el contexto de los procesos de reconocimiento de patrones con algoritmos de aprendizaje automático, ya sea a través de métodos supervisados, semi-supervisados o no supervisados, uno de los elementos más importantes a considerar son las características (*features* en inglés) que se utilizan para representar el fenómeno a estudiar. En este sentido, en este apartado de la tesis se propone una arquitectura de aprendizaje profundo para Ar2p, que se basa en mecanismos supervisados y no supervisados, para el descubrimiento y la selección de las características a considerar en problemas de clasificación, denominado Ar2p-DL.

Ar2p-DL se compone de tres fases: la primera fase, es la fase de *análisis de características*, la cual es soportada por dos enfoques de ingeniería de características para descubrir o seleccionar características/descriptores atómicos: el primer enfoque de ingeniería de características utilizado para el descubrimiento de las mismas, se basa en una técnica de agrupamiento llamada *K-means*; y el otro enfoque utilizado para la selección de características, se basa en la técnica de clasificación *Random Forest*. La segunda fase, llamada fase de *agregación*, crea una jerarquía de características (emergencia de descriptores) a partir de las características/descriptores atómicos descubiertos (utiliza como estrategia de agregación el algoritmo *DBSCAN*). Finalmente, la tercera fase, llamada fase de *clasificación*, la cual clasifica las entradas en función de la jerarquía de características, utilizando nuestro algoritmo *clásico Ar2p*. Esta última fase de Ar2p-DL utiliza un enfoque de aprendizaje supervisado, mientras que las dos primeras siguen un enfoque de aprendizaje híbrido (supervisado y no supervisado).

Para analizar el rendimiento de Ar2p-DL, se realizan varias pruebas con diferentes benchmarks (datasets) del repositorio de aprendizaje de máquinas de UCI, para comparar Ar2p-DL con otros métodos de clasificación.

#### 4.1.1 Generalidades

El reconocimiento de patrones es un campo en constante desarrollo. En la literatura, hay diferentes métodos de reconocimiento vistos como tareas de clasificación, como los métodos basados en árboles de decisión, los métodos basados en redes neuronales, entre otros. Un enfoque de particular interés es el "Aprendizaje Profundo" (o DL, por sus siglas en inglés), el cual se basa en la adquisición automática de conocimientos para la clasificación de patrones (LeCun, Bengio y Hinton 2015). La gran ventaja de usar DL es que no requiere una definición "a mano" de las características que permiten clasificar los patrones, sino que estas características se generan automáticamente a partir de los datos (Schmidhuber 2015).

Actualmente, hay una gran cantidad de redes neuronales artificiales para la clasificación de patrones basadas en DL: Redes Neuronales Convolucionadas (CNN, por sus siglas en inglés) (LeCun y Bengio 1995), usadas para la compresión de textos de lenguajes naturales, reconocimiento de escritura a mano, reconocimiento de voz, reconocimiento de gestos, captura de imágenes; Redes de Creencia Profunda (DBN, por sus siglas en inglés) (Hua, Guo y Hua 2015), usadas para el reconocimiento de imágenes, recuperación de imágenes, comprensión de lenguajes naturales, predicción de fallas; Modelos de memorias de largo y corto plazo (LSTM, por sus siglas en inglés) (Ester, Kriegel, Sander y Xu 1996), usadas en la compresión de textos de lenguajes naturales, reconocimiento de escritura a mano, reconocimiento de voz, reconocimiento de gestos, captura de imágenes; y Redes Neuronales Recurrentes (RNN, por sus siglas en inglés), usadas en

el reconocimiento de la voz, reconocimiento de la escritura a mano, entre otras (Aguilar 1998, 2001, 2004; Salakhutdinov y Larochelle 2010; Gelenbe y Yin 2016).

En general, las arquitecturas profundas llevan a cabo una forma de ingeniería de características, y normalmente utilizan combinaciones de aprendizaje supervisado y no supervisado en diferentes niveles. La ingeniería de características consiste en el proceso de búsqueda de características en un conjunto de datos, y se compone de los siguientes procesos: descubrimiento, extracción, construcción, reducción/fusión y/o selección de características (Guyon y Elisseeff 2006; Liu y Motoda 1998; Khalid, Khalily Nasreen 2014; Motoda y Liu 2002). Hay un número importante de trabajos que han estudiado individualmente estos procesos, proponiendo diferentes estrategias específicas para cada uno de ellos y, normalmente, se centran en áreas específicas de aplicación, como el procesamiento de imágenes (Kumar y Bhatia, 2014), análisis de tráfico de red (Pacheco, Exposito, Gineste, Baudoin, y Aguilar, 2018), o análisis urbano en ciudades (Chang, Bus y Schmitt 2017), entre otros.

Por otra parte, recapitulando lo que se describió en el capítulo anterior, Ar2p es un algoritmo de reconocimiento de patrones basado en el funcionamiento sistemático del cerebro humano (Puerto y Aguilar 2017a, 2017b), que puede reconocer patrones tanto estáticos como dinámicos (Puerto, Aguilar, y Chávez, 2018), y que incorpora mecanismos de aprendizaje supervisado (Puerto y Aguilar 2016b). Si bien estas propiedades le permiten reconocer y aprender, el hecho que sean supervisadas es también una de sus principales debilidades, puesto que restringe su capacidad para aprender nuevos patrones/características de manera automática. Es así que en este capítulo se aborda el problema de cómo extender este enfoque de aprendizaje supervisado hacia un aprendizaje híbrido o semi-automático. Para ello, se propone combinar el paradigma de

aprendizaje profundo o DL con la ingeniería de características, para diseñar un enfoque de aprendizaje híbrido para Ar2p.

#### 4.1.2 Ingeniería de Características

En el contexto del reconocimiento de patrones, ya sea a través de métodos supervisados, semi-supervisados o no supervisados, uno de los elementos más importantes a considerar son las características/descriptores que se utilizan para representar el fenómeno a estudiar. Estas características son muy importantes en el aprendizaje automático, ya que la precisión de los modelos depende de la calidad de las características que representan los objetos a reconocer. También, porque tienen un impacto en los recursos, el tiempo, y otros elementos requeridos por los modelos. En general, existen diferentes técnicas para obtener desde los datos las características/descriptores que se utilizarán en el proceso de reconocimiento. Este dominio se denomina ingeniería de características (Guyon y Elisseeff 2006; Liu y Motoda 1998; Khalid, Khalily Nasreen 2014; Motoda y Liu 2002; Brownlee 2014).

Específicamente, la ingeniería de características consiste en los procesos de descubrimiento, extracción, construcción, reducción y/o selección de características a partir de los datos (Sarkar, Bali, y Sharma 2018). Cada proceso utiliza diferentes técnicas de minería, y tiene diferentes objetivos. A continuación, se describen los principales procesos de la ingeniería de características considerados en este trabajo:

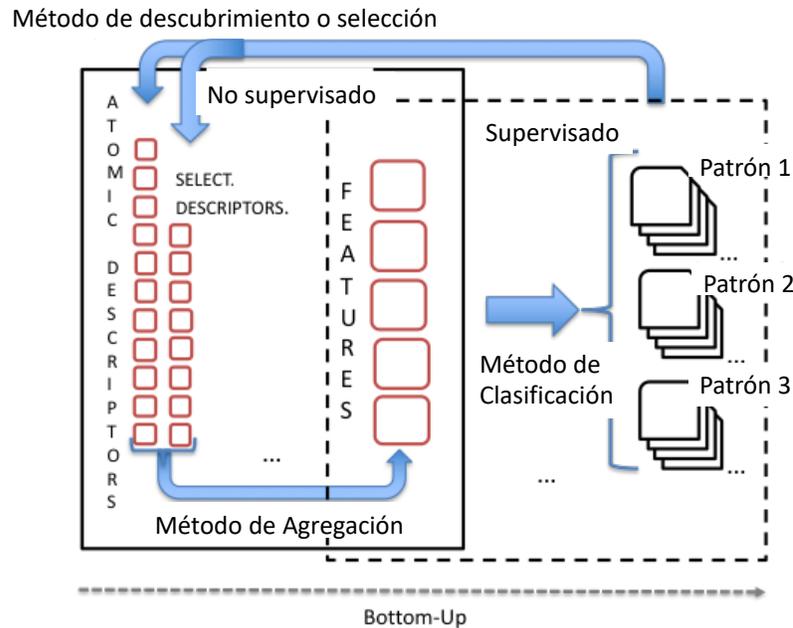
- Descubrimiento de características (*Feature Extraction: FE*). Este proceso consiste en identificar (a partir de los datos disponibles) las características que mejor describen el fenómeno estudiado, las cuales, a su vez, son útiles para construir modelos de conocimiento.
- Extracción de características (*Feature Extraction: FE*). Este proceso consiste en extraer las características identificadas del fenómeno estudiado, desde los datos.

- Construcción de características (*Feature Construction: FC*). Este proceso consiste en generar nuevas características, las cuales son usadas para complementar la información faltante en la descripción del fenómeno estudiado, o para establecer la relación entre las características que ya existen. La construcción es normalmente basada en las características que ya existen.
- Selección de características (*Feature Selection: FS*). Este proceso, a diferencia del anterior, busca reducir el conjunto de características a utilizar. En general, el FS se puede realizar ya sea por selección o agregación de las características disponibles. En el caso de selección, puede hacerse generando subconjuntos aleatoriamente, incrementalmente, etc. En el caso incremental, podría comenzar con un subconjunto vacío, y las características se agregarán una por una; o viceversa, comenzando por el conjunto completo, e ir eliminando éstas. La agregación consiste en la fusión de las características disponibles, por ejemplo, usando técnicas como la de análisis de componentes principales (PCA, por sus siglas en inglés), utilizada para describir un conjunto de datos correlacionados, y así reducir la dimensionalidad de un conjunto de datos.

## 4.2 Arquitectura General Propuesta de Ar2p-DL

La Figura 4.1 muestra la arquitectura Ar2p-DL (Puerto, Aguilar, Reyes, y Sarkar 2018). La arquitectura se compone de tres fases: la primera, denominada *fase de análisis de características*; constituida por dos métodos: un *método de descubrimiento* y un *método de selección*, a través de los cuales se descubren o se seleccionan los descriptores/características desde un dataset. La segunda, denominada *fase de agregación*, que está constituida por un *método de agregación*, el cual crea una jerarquía de características a partir de los descriptores del nivel inferior. Finalmente, la *fase de clasificación*, que está constituida por el método *Ar2p* clásico, que lleva a cabo la

clasificación utilizando la jerarquía de características de la fase de agregación. Estas fases están integradas en el macro-algoritmo de la Tabla 4.1.



**Figura 4.1.** Arquitectura de Ar2p-DL

El macro-algoritmo comienza con un proceso de exploración, caracterizado por la aplicación de un mecanismo no supervisado (línea 1, Tabla 4.1); al cual le sigue un proceso de integración, caracterizado también por la aplicación de un mecanismo no supervisado (línea 2, Tabla 4.1); hasta un proceso de clasificación y/o reconocimiento, caracterizado por la aplicación de un mecanismo supervisado (línea 3, Tabla 4.1). Así, la arquitectura sigue un enfoque de aprendizaje semi-supervisado.

El primer paso detecta/selecciona descriptores o características atómicas. La capa 1 se crea con estas características. Luego, inicia un proceso de agrupamiento jerárquico aglomerativo que combina estas propiedades atómicas, hasta que todos los descriptores de la capa previa sean combinados (línea 2, Tabla 4.1). Estos descriptores/características definen la capa 2.

**Tabla 4.1.** Macro-algoritmo Ar2p-DL

<b>Algoritmo 1.</b> Macro-algoritmo Ar2p-DL
Input: dataset Output: clasificación/reconocimiento 1. Descubrir/Seleccionar características/patrones atómicos: aplicar sub-algoritmos de descubrimientos (ver sección 4.2.1.1) o selección (ver sección 4.2.1.2) de características 2. Repetir 2.1 Agregar propiedades hasta que todos los descriptores de la capa previa sean combinados: aplicar sub-algoritmo de agregación (ver sección 4.2.2) 3. Clasificación: aplicar algoritmo de clasificación (ver sección 4.2.3)

Subsecuentemente, inicia el proceso de clasificación (método de clasificación de la Figura 4.1, y línea 3 de la Tabla 4.1), usando las características de la capa 2. Específicamente, una vez las características más representativas de los datos originales han sido identificadas en la capa 2, estas se convierten en señal de entrada para el algoritmo clásico de *Ar2p*. Los módulos de reconocimiento de *Ar2p* definen la capa 3. De esta manera, la última capa tiene los módulos de reconocimiento de cada clase a reconocer.

#### 4.2.1 Fase de descubrimiento de descriptores de Ar2p-DL

En esta fase se crea la primera capa de descriptores atómicos (o características atómicas), mediante uno de los siguientes dos métodos: o por el método de descubrimiento de características basado en técnicas de *clustering* (ver sección 4.2.1.1), o por el método de selección de características basado en técnicas de clasificación (ver sección 4.2.1.2).

##### 4.2.1.1 Método de descubrimiento de características de Ar2p-DL.

En este caso, la primera capa de descriptores/características atómicas se crea por el método basado en técnicas de *clustering*, la cual aplica dos variantes del algoritmo clásico de K-means, para descubrir los descriptores: la técnica incremental *K-means* (Pham, Dimov y Nguyen 2004), el cual determina el valor inicial de K (llamada  $M_{k_{prev}}$ ), que define el número inicial de características; y

la técnica *X-means* (Pelleg y Moore 2000), usada para descubrir/detectar más características desde las características iniciales  $M_{k_{prev}}$ .

*Definición 4.1: K-means* (MacQueen 1967). Es un algoritmo popular de aprendizaje de máquina, no supervisado, para realizar tareas de segmentación (*clustering*). Su objetivo es particionar “ $n$ ” observaciones en “ $K$ ” grupos (cluster), de tal forma que cada observación pertenezca al cluster de cuyo prototipo este más cercano. La media de las observaciones en un cluster particular, define el prototipo/centroide del cluster. Uno de sus inconvenientes es que se requiere especificar el número de cluster,  $K$ , antes de que el algoritmo sea aplicado.

*Definición 4.2: K-means secuencial incremental*. Esta técnica es usada para seleccionar  $K$  (cluster) basada en la función  $f(K)$ , la cual es definida por la Ecuación (4.1) como sigue (Pham, Dimov, y Nguyen 2004, 2005):

$$f(K) = \begin{cases} 1 & \text{if } K = 1 \\ \frac{S_K}{\alpha_K S_{K-1}} & \text{if } S_{k-1} \neq 0, \forall K > 1 \\ 1 & \text{if } S_{k-1} = 0, \forall K > 1 \end{cases} \quad (4.1)$$

$$\text{Donde } \alpha_K = \begin{cases} 1 - \frac{3}{4N_d} & \text{if } K = 2 \wedge N_d > 1 \\ \alpha_{K-1} + \frac{1-\alpha_{K-1}}{6} & \text{if } K > 2 \wedge N_d > 1 \end{cases}$$

$S_k$  es la suma de las distorsiones del *clúster* cuando el número de grupos es  $K$  (ver definición 4.3), y  $N_d$  es el número de atributos del conjunto de datos (es decir, el número de dimensiones). El término  $\alpha_k S_{k-1}$  en la Ecuación (4.1) es una estimación de  $S_k$  basada en  $S_{k-1}$ , realizada bajo el supuesto de que los datos tienen una distribución uniforme. El valor de  $f(K)$  es la relación entre la distorsión real y la distorsión estimada, y es cercano a 1 cuando la distribución de datos es uniforme. Cuando hay áreas de concentración en la distribución de datos,  $S_k$  será menor que el valor estimado, de modo que  $f(K)$  disminuye. Cuanto más pequeño es  $f(K)$ , más concentrada es la

distribución de datos. Por lo tanto, se puede considerar que los valores de  $K$  que producen una pequeño valor de  $f(K)$  proporcionan grupos bien definidos (Pham, Dimov, y Nguyen 2004, 2005).

*Definición 4.3: Distorsión de Cluster.* La distorsión de un cluster es la distancia entre los objetos de un cluster y su centroide, según la Ecuación (4.2) (Pham, Dimov, y Nguyen 2004, 2005).

$$I_j = \sum_{t=1}^{N_j} [d(x_{jt}, w_j)]^2 \quad (4.2)$$

Donde  $I_j$  es la distorsión del cluster  $j$ ,  $w_j$  es el centroide del cluster  $j$ ,  $N_j$  es el número de objetos pertenecientes al cluster  $j$ ,  $x_{jt}$  es el  $t$ th objeto perteneciente al cluster  $j$ , y  $d(x_{jt}, w_j)$  es la distancia entre el objeto  $x_{jt}$  y el centroide  $w_j$  del cluster  $j$ . Cada cluster es representado por su distorsión, y el impacto general de todos los clusters en todo el conjunto de datos se evalúa por la suma de todas las distorsiones,  $S_K$ , dada por la Ecuación (4.3) (Pham, Dimov, y Nguyen 2004).

$$S_K = \sum_{j=1}^K I_j \quad (4.3)$$

Donde  $K$  es el número de clusters. Se supone que el número de cluster  $K$  es mucho más pequeño que el número de objetos  $N$ . Cuando  $K$  aumenta, entonces  $f(K)$  debe converger a algún valor constante. Ahora bien, si para cualquier  $K$  inmediato  $f(K)$  muestra un comportamiento especial, en particular un punto mínimo, ese valor de  $K$  podría tomarse como el número deseado de clusters.

*Definición 4.4: X-means.* Determina si deben aparecer nuevos centroides dentro de un modelo actual ( $M_i$ ). La aparición de nuevos centroides se lleva a cabo dividiendo algunos cluster en 2, los cuales han sido clasificados como optimizables según el criterio de Schwarz (es un criterio para la selección de modelos entre un conjunto finito de modelos, basado en el valor BIC, ver la ecuación (4.4)) (Pelleg y Moore 2000; Kass y Wasserman 1995):

$$BIC(M_j) = \hat{t}_j(D) - \frac{p_j}{2} \cdot \log R \quad (4.4)$$

Donde,  $\hat{t}_j(D)$  es la probabilidad logarítmica de los datos en el modelo  $M_j$ ;  $p_j$  es el número de parámetros libres presentes en el modelo  $M_j$ ; y  $R$  representa el número de muestras presentes en  $D$  ( $R = |D|$ ).

X-means ha sido usado para descubrir/detectar más características a partir de una semilla inicial de características encontradas. En esencia, nuestro enfoque comienza con  $K$  igual a  $M_{k_{prev}}$ , y continúa agregando centroides usando el método X-means, que calcula el score BIC para cada cluster a través de la Ecuación (4.4). Durante este proceso se guarda el conjunto de centroides que alcanzan el mejor score, que es la salida final de esta fase. La Tabla 4.2 muestra el algoritmo de descubrimiento de características.

**Tabla 4.2.** Algoritmo de descubrimiento de características para Ar2p-DL

<b>Algoritmo 2.</b> Algoritmo de descubrimiento de características.
Input: Data set ( <i>dataset</i> )
Output: Conjunto de características descubiertas
1. Determinación inicial de $K$ ( $M_{k_{prev}}$ ), utilizando la definición 4.1.
2. Refinamiento de la estructura $M_{k_{prev}}$ , utilizando la definición 4.4.

#### 4.2.2.2 Método de selección de características.

En este caso, la primera capa de características atómicas se crea mediante el método FS (del inglés *Features Selection*), que selecciona las características a partir de un conjunto inicial predefinido.

La Tabla 4.3 especifica el sub-algoritmo de selección de características propuesto.

**Tabla 4.3.** Algoritmo de selección de características para Ar2p-DL

<b>Algoritmo 3.</b> Algoritmo de Selección de Características
Input: Data set ( <i>dataset</i> )
Output: Conjunto de características seleccionadas
1. Proceso de Pre-selección
2. Proceso de Combinación
3. Proceso de Post-selección

En general, el macro-algoritmo consiste de 3 partes:

- El *proceso de pre-selección*, en el que los primeros conjuntos de características se extraen utilizando el método RF (del inglés *Random Forest*) (Breiman 2001) y el algoritmo de análisis lineal discriminante (LDA, del inglés *Linear Discriminant Analysis*) (Balakrishnama y Ganapathiraju 1998). RF es un método de aprendizaje automático para clasificación, basado en un conjunto de árboles de decisión, construidos a partir de subconjuntos aleatorio de datos. La salida de todos los árboles de decisión se combinan para dar una solución final. Por otra parte, LDA es una técnica para la clasificación de datos y reducción de dimensionalidad, que busca la combinación lineal de características que maximicen la relación entre la variación de los datos entre clases y la variación de los datos en una misma clase, para cualquier conjunto de datos, garantizando así la máxima separabilidad.
- El *proceso de combinación*, que mezcla los conjuntos de características extraídos en la fase anterior, utilizando diferentes combinaciones o criterios de selección. En particular, el proceso de combinación utiliza las siguientes estrategias de selección: basada en la calidad de la clasificación con cada conjunto de características, basada en la combinación aleatoria de conjuntos de características, y basada en la frecuencia con que aparece cada característica en cada conjunto (*Voting Features*).
- Finalmente, el *proceso de selección posterior*, determina la precisión de los clasificadores obtenida con los conjuntos de características definidos en la fase anterior, y selecciona el conjunto con el valor más alto como el conjunto óptimo final de características.

Las Tablas de la 4.4 a la 4.6 especifican los sub-algoritmos que comprenden estos procesos.

**Tabla 4.4.** Algoritmo de Pre-selección

<b>Algoritmo 4.</b> Algoritmo de Pre-selección
<p>Input: Data set (dataset), técnica de selección (RF, LDA).  Output: Diferentes conjuntos de características seleccionadas (<i>feature set</i>)</p> <ol style="list-style-type: none"> <li>1. Elegir aleatoriamente las características en el conjunto de datos y aplicar la técnica de selección utilizada</li> <li>2. Calcular la importancia de cada característica según la calidad del resultado con cada técnica de selección utilizada.</li> <li>3. Seleccionar las características cuya importancia superan un valor de umbral</li> </ol>

**Tabla 4.5.** Sub-algoritmo del proceso de combinación

<b>Algoritmo 5.</b> Sub-algoritmo del proceso de combinación
<p>Input: Diferentes conjuntos de características generadas en el proceso de preselección.  Output: Conjunto de características combinadas</p> <ol style="list-style-type: none"> <li>1. Selección basada en la calidad de clasificación <ol style="list-style-type: none"> <li>1.1 Para cada conjunto de características obtenidas del proceso de preselección, realizar: <ol style="list-style-type: none"> <li>1.2 Un proceso de clasificación con RF. <ol style="list-style-type: none"> <li>1.2.1 Calcular el score de los resultados usando las métricas <i>f-measure</i> y <i>accuracy</i>.</li> <li>1.2.2 Seleccionar los conjuntos de características con el mejor score</li> </ol> </li> <li>1.3. Completar combinación <ol style="list-style-type: none"> <li>1.3.1 Seleccionar aleatoriamente un conjunto de características</li> <li>1.3.2 Agregar sus características en un conjunto global</li> <li>1.3.3 Finalizar con un conjunto de características que combina algunos de los conjuntos de características generados en el proceso de preselección</li> </ol> </li> </ol> </li> <li>1.4. Elección (voting) por mayoría <ol style="list-style-type: none"> <li>1.4.1 Repetir para un grupo de conjuntos de características escogido aleatoriamente</li> <li>1.4.2 Establecer un contador para cada característica encontrada</li> <li>1.4.3 Incrementar el contador para cada aparición de cada una en cada conjunto de características</li> <li>1.4.4 Finalizar con un conjunto de características que agrupa aquellas con mayor valor de contador.</li> </ol> </li> </ol> </li> </ol>

**Tabla 4.6.** Sub-algoritmo del proceso de post-selección

<b>Algoritmo 6.</b> Sub-algoritmo del proceso de post-selección
<p>Input: Conjunto de características generadas en el proceso de combinación  Output: Conjunto final de características</p> <ol style="list-style-type: none"> <li>1. Para cada conjunto de características se lleva a cabo un proceso de clasificación con RF.</li> <li>2. Calcular el score de los resultados con las métricas <i>f-measure</i> y <i>accuracy</i></li> <li>3. El conjunto de características con el mayor performance es seleccionado</li> </ol>

### 4.2.2 Fase de agregación de Ar2p-DL

En esta fase, se crea una segunda capa de características mediante un método de agrupamiento basado en densidad. Luego de la fase anterior, en esta fase se procede a chequear si las características generadas pueden agruparse utilizando el algoritmo DBSCAN (ver definición 4.5).

Definición 4.5: DBSCAN (Ester, Kriegel, Sander, y Xu 1996). DBSCAN es un algoritmo de agregación basado en dos parámetros: épsilon (“eps”) y puntos mínimos (“MinPts”). eps define el radio de vecindad alrededor de un punto  $p$ , denominado vecindad-eps del punto  $p$ , denotado por  $N_{Eps}(p)$ , y definido por la Ecuación (4.5):

$$N_{Eps}(p) = \{q \in D \mid dist(p, q) \leq Eps\} \quad (4.5)$$

Donde  $dist(p, q)$  es una función de distancia entre  $p$  y  $q$  (e.g., la distancia Euclidiana o Manhattan) que determina el tipo de vecindario, y  $Eps$  es un parámetro de vecindario (definido manualmente o automáticamente). Por otra parte, se fija el número mínimo de puntos ( $MinPts$ ) en la vecindad-eps de un punto central (core). En un enfoque de densidad sencilla se requiere que para el punto central  $q$ , el conjunto de puntos  $p$  esté dentro de la vecindad-eps y  $N_{Eps}(q)$  contenga siquiera  $MinPts$  puntos ( $|N_{Eps}(q)| \geq MinPts$ ). Un punto  $p$  es densamente alcanzable desde un punto central  $q$  si  $p \in N_{Eps}(q)$ . Entonces, una agregación de puntos para  $D$ , satisface las siguientes condiciones:

- Maximalidad:  $\forall p, q$ : si  $p \in D$  y  $q$  es un punto central de  $D$ , entonces  $p \in N_{Eps}(q)$ .
- Conectividad:  $\forall p, q$ : si  $p \in D$  y  $q$  es un punto central de  $D$ , entonces  $p$  es densamente alcanzable desde  $q$

Finalmente, el ruido (*noise*) es un conjunto de puntos en la base de datos  $D$  que no pertenecen a ningún cluster. La Tabla 4.7 muestra el algoritmo de agregación.

**Tabla 4.7.** Algoritmo de Agregación para Ar2p-DL

<b>Algoritmo 7.</b> Algoritmo de Agregación.
1. Encuentre los $\epsilon$ ( <i>eps</i> ) vecinos de cada punto e identifique los puntos centrales con más de <i>minPts</i> vecinos utilizando la Definición 4.5.
2. Encuentre los componentes conectados de los puntos centrales.
3. Asigne cada punto no central a un clúster cercano, si el clúster es un <i>eps</i> -vecino. De lo contrario, asigne este como ruido.

Nuestro algoritmo de clustering basado en densidad trabaja como sigue: Primero, para cada punto  $p$ , encuentra todos los puntos vecinos dentro de la distancia *eps* del punto  $p$  (la *eps*-vecindad de un punto  $p$  denotada por  $N_{Eps}(p)$ , ver ecuación (4.5)). Cada punto, con un contador de vecinos mayor o igual a *MinPts*, es marcado como punto central. Luego, para cada punto central, si aún no ha sido asignado a un cluster, entonces se crea un nuevo cluster.

Una vez creado el nuevo cluster, lo siguiente es encontrar recursivamente todos sus puntos que por densidad se encuentran conectados, para que sean asignados al mismo clúster que el punto central. Finalmente, se recorren los puntos restantes que no han sido visitados en el conjunto de datos. Los puntos que no pertenecen a ningún cluster se tratan como valores atípicos o ruido.

El resultado de esta fase puede ser interpretado como: todos los cluster formados por el algoritmo representan características agrupadas/fusionadas debido a sus similitudes compartidas, mientras que aquellas características que DBSCAN determina como ruido son características que no tienen relación con ningún cluster. Estos nuevos descriptores/características se convierten en la entrada de la fase de clasificación (ver sección 4.2.3).

### 4.2.3 Fase de clasificación de Ar2p-DL

Para esta fase, el proceso *Bottom-up* de *Ar2p* es usado como *clasificador*, puesto que las características atómicas descubiertas en los primeros niveles de Ar2p-DL serán tomadas como los patrones atómicos del último nivel de *Ar2p*. Por ello, la clasificación se realiza tomando como

entrada los descriptores/características establecidas en la fase de agregación. De esta manera, los módulos responsables del reconocimiento de patrones de Ar2p utilizan los resultados de las fases previas de Ar2p-DL. El algoritmo *Bottom-up* de Ar2p usado como clasificador se presenta en la Tabla 4.8.

**Tabla 4.8.** Algoritmo Bottom-Up de clasificación

<b>Algoritmo 8.</b> Proceso Bottom-Up
Input: Descriptores atómicos, 1. $y = s()$ = descriptores atómicos 2. Calcular el reconocimiento de cada módulo de reconocimiento $j$ de Ar2p usando las ecuaciones (3.1), (3.2) y (3.3) del capítulo anterior 3. Si el reconocimiento es exitoso por cualquier estrategia Entonces 4. Enviar señal $S_o$ de reconocimiento ( $\Gamma_j$ ) como salida del sistema. 5. Sino 6. Enviar señal $S_o$ de no reconocimiento ( $\Gamma_j$ ) como salida del sistema.

El proceso de clasificación comienza con las señales de las características atómicas ( $y$ ) descubiertas en las capas previas de Ar2p-DL (línea 1). Luego, el algoritmo calcula el reconocimiento de cada módulo de reconocimiento de patrones ( $j$ )  $\Gamma_j$  de Ar2p usando las ecuaciones (3.1), (3.2) o (3.3) (línea 2). Si el cálculo es exitoso (línea 3), entonces envía la señal  $S_o$  de salida de reconocimiento del patrón  $j$  (línea 4). De lo contrario, envía la señal  $S_o$  de no reconocimiento del patrón  $j$  ( $\Gamma_j$ ). Este algoritmo ha sido extendido con mecanismos de aprendizaje para reforzar los módulos de reconocimiento existentes, o para aprender nuevos módulos (ver (Puerto y Aguilar 2016) y el capítulo anterior, para más detalles).

### 4.3 Aplicaciones de ar2p-dl Usando Diversos Dominios

Para analizar el desempeño de Ar2p-DL, se presenta en esta sección un análisis comparativo entre Ar2p-DL y otros métodos de clasificación, para lo cual se usan diferentes benchmarks (*datasets*) y las métricas *F-measure* y *accuracy*. Los benchmarks de problemas de clasificación fueron tomados del repositorio de aprendizaje de máquina UCI (Asuncion y Newman 2007). Estos

benchmark/datasets tienen diferentes características, tales como el tamaño del *dataset* (número de individuos), el número de descriptores/características, el número de clases y el número de instancias para cada clase. La Tabla 4.9 reporta las características de los datasets usados para el caso de descubrimiento de características.

**Tabla 4.9.** Datasets usadas para el descubrimiento de características

Dataset	Descripción
<i>Dígitos escritos a mano MNIST</i> (Deng 2012)	10 clases codificadas de 0 a 9
Fase de Segmentación de Gestos	5 clases correspondientes a las fases de un gesto: <i>Rest</i> , <i>Preparation</i> , <i>Stroke</i> , <i>Hold</i> y <i>Retraction</i>

La Tabla 4.9 describe las dos bases de datos tomadas de UCI para analizar y/o validar la capacidad de descubrimiento de características de Ar2p-DL. La base de datos *MNIST* contiene 70,000 imágenes de diez dígitos escritos a mano, normalizados y pre-procesados, cada imagen tiene un tamaño de  $28 \times 28$  píxeles. Por otra parte, la *base de datos de gestos* está compuesta por un conjunto de características extraídas de 7 videos de personas que gesticulan, con el objetivo de estudiar la fase de segmentación de gestos (del inglés *Gesture Phase Segmentation*). Cada video está representado por dos archivos: un archivo en crudo, que contiene la posición de las manos, muñecas, cabeza y columna vertebral del usuario en cada frame; y un archivo procesado, que contiene la velocidad y la aceleración de manos y muñecas (El Kessab, Daoui, Boukhalene, y Salouan 2014).

Por otro lado, la Tabla 4.10 reporta las características de los datasets utilizados para el caso de la selección de características. La Tabla 4.10, describe el tamaño, número de características y número de instancias del problema de clasificación en las tres bases de datos en cuestión.

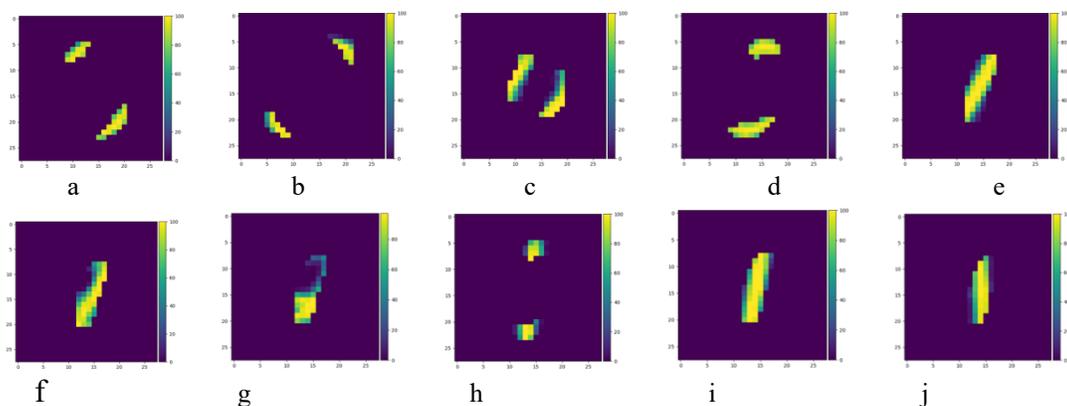
**Tabla 4.10.** Datasets para validar Ar2p-DL por selección de características

Dataset	Tamaño	Características	Número de instancias en cada clase
HTRU2 (Keith et al 2010)	17898	9	Las dos (2) clases de etiquetas usadas son: 0 (negativo) y 1(positivos). C1=1,639 casos positivos. C2=16,259 casos negativos.
Wholesale Costumers (Lyon et al 2016)	440	8	2 clases correspondientes a los tipos de medios de un distribuidor mayorista. C1 = 248; C2 = 192.
Seeds (Charytanowicz et al 2010)	201	7	3 clases correspondientes a tipos de trigo (kama, rosa y canadiense). C1 = 70; C2 = 70; C3 = 70

A continuación, se usa Ar2p-DL para reconocer diferentes patrones en dos casos de estudios. usando el enfoque por *descubrimiento de características* (secciones 4.3.1 y 4.3.2). Posteriormente, Ar2p-DL es usado para reconocer diferentes patrones en tres casos de estudio en el contexto de la *selección de características* (secciones 4.3.3, 4.3.4, 4.3.5).

### 4.3.1 Reconocimiento de dígitos escritos a mano usando Ar2p-DL por descubrimiento de características

En esta sección, se describe el proceso de reconocimiento de dígitos escritos a mano usando Ar2p-DL. La Figura 4.2 muestra 10 características atómicas extraídas de MNIST (Deng 2012), producto de aplicar los algoritmos de descubrimiento de características en las primeras capas de Ar2p-DL.



**Figura 4.2.** Ejemplos de características atómicas extraídas desde la base de datos de dígitos escritos a mano MNIST (Deng 2012)

Estos patrones se convierten en señales de entrada para la última capa de clasificación de Ar2p-DL. Con estas características es construido el conjunto de módulos de reconocimiento de Ar2p-DL para su fase de clasificación, como se muestra en la siguiente sección.

#### 4.3.1.1 Módulo de reconocimiento del dígito “0” escrito a mano.

Para la creación del módulo de reconocimiento para el dígito “0”, las diez (10) características  $\{a,b,c,d,\dots\}$ , en la Figura 4.2} son seleccionadas del conjunto. La Tabla 4.11 muestra la estructura general del módulo de reconociendo del “0”. En este caso, para reconocer el “0”, las características relevantes descubiertas de la Figura 4.2 son  $\{a,b,c,d,h\}$ , por lo que el peso del resto de señales es 0.

**Tabla 4.11.** Módulo de reconocimiento  $\Gamma=$ ”0”

S		E		
		M		
Señal	Estado	Descriptor (F)	Dominio (V)	Peso (W)
1	Falso	Fig.4.2a	<valores>	0.9
2	Falso	Fig. 4.2b	< valores >	0.9
3	Falso	Fig. 4.2c	< valores >	0.9
4	Falso	Fig. 4.2d	< valores >	0.9
...				
8	Falso	Fig. 4.2h	< valores >	0.7
...				
< $\Delta U_1=0.9, U_2=0.8$ >				

Los estados de las señales son inicialmente falsos por defecto, los valores de los pesos (W) son definidos de manera supervisada, considerando la importancia de las características atómicas.

Por ejemplo, la señal 5 tiene un valor bajo (0.7) porque esta señal es ambigua, i.e., esta puede también ser parte del dígito 1.

#### 4.3.1.2 Módulo de reconocimiento del dígito “1”

La creación del módulo de reconocimiento para el dígito “1” es definida de manera similar a la anterior (ver la Tabla 4.12). En este caso, para reconocer el “1”, las características relevantes descubiertas de la Figura 4.2 son  $\{e,f,g,h,i,j\}$ , por lo que el peso del resto de señales es 0.

**Tabla 4.12.** Módulo de reconocimiento  $\Gamma="1"$ 

E				
S		M		
Señal	Estado	Descriptor (F)	Dominio (V)	Peso (W)
...				
5	Falso	Fig.2e	<valores>	0.9
6	Falso	Fig.2f	<valores>	0.9
7	Falso	Fig.2g	<valores>	0.7
8	Falso	Fig.2h	<valores>	0.7
9	Falso	Fig.2i	<valores>	0.9
10	Falso	Fig.2j	<valores>	0.9
<U1=0.8,U2=0.7>				

Los módulos de reconocimiento para los otros dígitos son definidos de manera similar. Una vez los módulos de reconocimiento han sido creados, se aplica la última fase del algoritmo Ar2p-DL (ver algoritmo 8).

#### 4.3.2 Reconocimiento de gestos usando Ar2p-DL por descubrimiento de características

La segmentación de fase de gestos es una tarea con varias dificultades (Wagner 2014; Kahol, Tripathi, Panchanathan, y Rikakis 2003; Ding et al 2018). Primero, es una tarea subjetiva, ya que no hay un punto claro donde comienza cada fase. Además, algunas fases pueden presentar patrones similares, por ejemplo, cuando las manos permanecen quietas (posición *rest* y *hold*). La Figura 4.3 muestra esta situación.

**Figura 4.3.** Tipos de Gestos en posición rest y hold

La primera línea de la Figura 4.3 es una secuencia en *posición rest*, y la segunda línea de la misma Figura es una secuencia *hold*. La primera fase de descubrimiento de características se realiza a partir del flujo de gestos, de los cuales Ar2p-DL extrae la información: coordenadas tridimensionales ( $x$ ;  $y$ ;  $z$ ) de manos y muñecas, y la velocidad escalar y la aceleración de los movimientos de las mismas. Con esta información se crean las señales que constituirán los módulos de reconocimiento de la última capa para gestos. Una vez que se han creado y optimizado dichos módulos de reconocimiento, se aplica la última fase del algoritmo Ar2p-DL (consulte el algoritmo 8).

#### 4.3.3 Resultados of Ar2p-DL usando el enfoque de descubrimiento de características

Ar2p-DL es comparado con el modelo de perceptrón multi-capas MPL (Ruck, Rogers, y Kabrisky 1990) y K-NN (El Kessab, Daoui, Boukhalene y Salouan 2014). Para cada prueba, se aplica el proceso de validación cruzada de 10 iteraciones. En este caso, se usa la métrica *F-measure*, dado que el dataset no es balanceado. Los resultados de la métrica *F-measure* se muestran en la Tabla 4.13.

**Tabla 4.13.** F-measure (%) de Ar2p-DL y otros algoritmos de clasificación

Modelo	Dígitos escritos a mano	Gestos
MLP	0.89	0.87
Ar2p-DL	0.86	0.91
K-NN	0.84	0.97

De acuerdo con los resultados, con respecto al reconocimiento de dígitos escritos a mano, Ar2p-DL es más preciso que K-NN, pero menos preciso que MLP. Por otro parte, con respecto al reconocimiento de gestos, Ar2p-DL es más preciso que MLP, pero menos preciso que el K-NN.

Para la comparación entre los modelos DL, se utilizó la métrica de *accuracy* en el caso del reconocimiento de dígitos escritos a mano (Liu, et al 2018; Niu y Suen 2012; Nguyen, Yosinski,

y Clune 2015), y el índice Jaccard (Wu 2016; Wang 2016; Nguyen, Yosinski, y Clune 2015) en el caso del reconocimiento de gestos (ver la Tabla 4.14).

**Tabla 4.14.** Performance of Ar2p-DL y otros modelos DL

Modelo DL	Dígitos escritos a mano	Gestos
Ar2p-DL	0.96	0.92
CNN	0.97	0.75
DBM	0.99	N/A
DBN	N/A	0.77

Ar2p-DL logra un desempeño respetable en ambos casos, que en comparación con los métodos de DL en el caso de reconocimiento de dígitos escritos a mano es muy cercano a los mejores resultados (sutilmente inferior), y en el caso de reconocimiento de gestos a mano es mucho mejor.

A continuación, Ar2p-DL es usado para reconocer diferentes patrones en tres casos de estudio, en el contexto de selección de características. En un primer caso, es utilizado el dataset *HTRU2* (sección 4.3.4); luego, el dataset de *Wholesale Costumer* (sección 4.3.5), y finalmente, el dataset *seed* (sección 4.3.6).

#### 4.3.4 Reconocimiento de estrellas pulsantes (HTRU2) usando Ar2p-DL por selección de características

HTRU2 (Lyon et al) tiene originalmente 9 atributos/características, de las cuales Ar2p-DL (en sus primeras fases) propone 4 como seleccionadas (ver la Figura 4.4).

1= Mean\_i\_p  
 3= Excess\_kurtosis\_i\_p  
 4= Skewness\_i\_p  
 6= Standard\_deviation\_DM-SNR\_curve

**Figura 4.4.** Atributos seleccionados por Ar2p-DL del dataset HTRU2

Dichos descriptores se convierten en la primera capa de Ar2p-DL. Basados en estos descriptores de la primera capa, se construye el conjunto de módulos de reconocimiento para siguiente capa de

clasificación. A continuación, especificamos uno de los módulos para Ar2p-DL que permite la clasificación de estrellas Pulsantes o estrellas No-pulsantes (0) (ver la Tabla 4.15).

**Tabla 4.15.** Módulo de reconocimiento de estrellas No-Pulsantes (NP).  $\Gamma=0$

E				
S		M		
Señal	Estado	Descriptor (F)	Dominio (V)	Peso (W)
1	Falso	Mean_i_p	< 140.5625 102.5078125 103.015625...>	0.9
2	Falso	Excess_kurtosis_i_p	< -0.234571412 0.465318154 0.323328365...>	0.9
3	Falso	Skewness_i_p	< -0.699648398 -0.515087909 1.051164429...>	0.9
4	Falso	Standard_deviation_DM-SNR_curve	< 19.11042633 14.86014572 21.74466875...>	0.9
U:< $\Delta U_1=0.9$ >				

Por cuestiones de espació, el dominio (V) solo contiene los primeros tres valores de los 16259 ejemplos de estrellas No-Pulsantes. Sólo se usa el umbral U (umbral) para señales claves, por considerar a las señales seleccionadas como claves. Una vez que se han creado y ajustado los módulos de reconocimiento, se aplica la última fase del algoritmo Ar2p-DL (consulte el algoritmo 8).

#### 4.3.5 Reconocimiento de canales de distribución (Wholesale Costumer) usando Ar2p-DL por selección de características

La dataset *Wholesale Costumer* (Abreu 2011) tiene originalmente 8 características. La primera fase de Ar2p-DL selecciona 4 de ellos, a saber: las características 3,4,6 y 7 (ver Figura 4.5).

3 = Milk  
4 = Grocery  
6 = Detergents\_Paper  
7 = Delicatessen

**Figura 4.5.** Atributos seleccionados por Ar2p-DL del dataset Wholesale Costumers

Sobre la base de estas características, se construye el conjunto de módulos de reconocimiento de la capa de clasificación. La Tabla 4.16 muestra el módulo de reconocimiento para "Horeca"<sup>12</sup>.

**Tabla 4.16.** Módulo de reconocimiento  $\Gamma$ ="Horeca"

E				
S		M		
Señal	Estado	Descriptor (F)	Dominio (V)	Peso(W)
1	Falso	Milk	<1196, 3648, 1114...>	0.9
2	Falso	Grocery	<4221, 6192, 3821...>	0.9
3	Falso	Detergents_Paper	<507, 1716, 964...>	0.9
4	Falso	Delicatessen	<1788, 750, 412...>	0.9
U:< □U1=0.9>				

El dominio (V) y el Umbral (U) para este módulo se definen/muestran con iguales criterios que antes, i.e., solo se indican tres de los posibles valores, y se usa solo el Umbral para reconocimiento por señales claves. Similar proceso se realiza para otros módulos de reconocimiento, como el de "Retail".

#### 4.3.6. Reconocimiento de semillas usando Ar2p-DL por selección de características

Este dataset tiene originalmente 7 características (Charytanowicz et al 2010), de las cuales 3 son seleccionadas en las primeras fases de Ar2p-DL, a saber: las características 1, 2 y 7 (ver Figura 4.6), las cuales pasan a conformar las primeras capas de Ar2p-DL. Luego, son creados los módulos de reconocimiento que constituyen la fase de clasificación. Específicamente los módulos de reconocimiento para las semillas Kama, Rosa y Canadian.

<sup>12</sup> HORECA hace referencia a un canal de distribución que agrupa establecimientos en los que se consume alimentos y bebidas (e.g., Hoteles, Restaurantes).

1 = area A,  
 2 = perimeter P,  
 7 = length of kernel groove.

**Figura 4.6.** Características seleccionadas por Ar2p-DL de la dataset Seeds

#### 4.3.7. Resultados de Ar2p-DL usando Selección de características

En este trabajo se utiliza un procedimiento de validación cruzada de  $K$ -iteraciones ( $K$ -fold cross-validation), con  $K = 10$ . En este caso, se utilizan las métricas F-measure y Accuracy (ver las Tablas 4.17 y 4.18, respectivamente). El algoritmo con la métrica de clasificación más alta se resalta en negrita.

**Tabla 4.17.** Resultados de F-measure

Algoritmo	HTRU2	Seeds	Wholesale Costumers
LDA	N/A	<b>0,9486</b>	0,3022
NN	0,8402	0,8761	0,2950
SVM	0,8014	0,9032	0,2787
Naïve Bayes	0,7023	0,9079	0,3319
RF	<b>0,9104</b>	0,9165	<b>0,3356</b>
Ar2p-DL	0,8701	0,9156	0,3305

**Tabla 4.18.** Resultados de Accuracy

Algoritmo	HTRU2	Seeds	Wholesale Costumers
LDA	N/A	<b>0,9778</b>	0,8106
NN	0,9790	0,9016	0,6492
SVM	0,9640	0,9365	<b>0,8121</b>
Naïve Bayes	0,9390	0,9302	0,6485
RF	<b>0,9796</b>	0,9460	0,6970
Ar2p-DL	0,9665	0,9406	0,8015

De acuerdo con los resultados (ver Tablas 4.17 y 4.18), nuestro algoritmo alcanza, en general, un tercer lugar en los experimentos, tanto para datos balanceados (*Seed*) como para datos no balanceados (*HTRA2* y *Wholesale-Customers*). Presenta, además, un rendimiento muy cercano a las mejores técnicas (RF y LDA). La Tabla 4.19 muestra los errores relativos entre los resultados del mejor clasificador y Ar2p-DL.

**Tabla 4.19.** Error relativo entre los más altos valores de las métricas y Ar2p-DL

Data set	Métrica	Mejor Valor	Ar2p-DL	Error relativo
HTRU2	F-measure	0,9104	0,8701	0.0403
	Accuracy	0,9796	0,9665	0.0131
Seeds	F-measure	0,9486	0,9156	0.033
	Accuracy	0,9778	0,9406	0.0372
Wholesale	F-measure	0,3356	0,3305	0.0051
Costumers	Accuracy	0,8121	0,8015	0.0106

De acuerdo con la Tabla 4.19, los resultados de Ar2p-DL, siempre están muy próximos a los mejores resultados alcanzados, independientemente de si las clases en los dataset están balanceadas o no. Es importante resaltar que los resultados conseguidos por los otros sistemas usan diferentes técnicas para extraer características, mientras que nuestro enfoque sólo usa el enfoque por selección de características. Nuestro enfoque puede realizar de manera conjunta las dos fases (selección y clasificación), o funcionar independientemente. Esta propiedad le da versatilidad a Ar2p-DL para funcionar como seleccionador y/o clasificador o como reconocedor.

#### 4.4. Análisis General de Resultados de Ar2p-DL y Conclusiones

En este trabajo se ha propuesto una arquitectura de aprendizaje profundo, llamada Ar2p-DL. Esta arquitectura se basa en el algoritmo clásico de Ar2p, con mecanismos no supervisados para el descubrimiento y selección de características. La arquitectura Ar2p-DL tiene tres fases; la primera, llamada *Fase de análisis de características* que se apoya en dos enfoques de ingeniería de características para el descubrimiento y selección de características atómicas. Una segunda fase, llamada *Fase de Agregación*, que crea una jerarquía de características (combinación de descriptores), y una tercera fase, la *Fase de Clasificación*, que lleva a cabo la clasificación de las entradas según las características escogidas en las fases previas.

La principal diferencia entre Ar2p-DL y los otros enfoques de DL es que incluye una fase de análisis de características que puede descubrir o seleccionar características. Además, la

arquitectura Ar2p-DL tiene un diseño flexible, que permite cambios estructurales en cada una de sus fases. Por ejemplo, Ar2p-DL puede trabajar con diferentes mecanismos no supervisados para el descubrimiento o selección de descriptores/características.

Si bien Ar2p-DL reduce el tiempo de diseño, ya que las características que constituyen los módulos de reconocimiento necesarios para la fase de clasificación se definen automáticamente, y sólo usa para esta fase el mecanismo *Bottom-up* de Ar2p, Ar2p-DL tiene un tiempo de ejecución importante debido a los diferentes mecanismos incluidos en cada una de ellas.

De acuerdo con los resultados, Ar2p-DL alcanza muy buenos resultados en los problemas de selección o descubrimiento de características, tanto para datos balanceados como no balanceados. Ar2p-DL ha mejorado significativamente el rendimiento (tiempo de procesamiento, uso de memoria) en relación con los otros clasificadores, en general, con buenos resultados en comparación con los clasificadores que presentan los mejores resultados en cada experimento. Los resultados muestran que Ar2p –DL mantiene su rendimiento para diferentes dominios, lo que lo hace más estable que otros modelos. Esto es consistente con su principio de modelado Neocortical, que es uniforme e independiente de los datos.

Como trabajo futuro, debemos analizar otras técnicas no supervisadas en las primeras fases de Ar2p, para minimizar el tiempo de ejecución, que es muy importante en los problemas de reconocimiento en tiempo real. Adicional a esto, vamos a analizar la sensibilidad de cada fase en los resultados finales, y probar nuestro enfoque con otros *datasets* del repositorio de aprendizaje de máquinas de UCI.

Por otra parte, a modo de conclusiones, Ar2p-DL tiene la capacidad de reconocer y/o clasificar patrones complejos a partir del aprendizaje semi-automático de ciertos patrones atómicos

(descriptores y/o características). Específicamente, Ar2p-DL descubre o selecciona automáticamente las características utilizando varios métodos de la ingeniería de características.

La principal diferencia de nuestro enfoque Ar2p-DL comparado con los enfoques clásicos de DL, está en la fase de aprendizaje y la de análisis de características. La fase de análisis de características de Ar2p-DL se da por la selección o el descubrimiento de descriptores. Por otra parte, la fase de aprendizaje en los enfoques clásicos es más complejo (*feedforward*, *backpropagation*, optimización de funciones de costo, etc.), mientras que, en nuestro enfoque, el proceso de aprendizaje es más fácil, sin dispersión de errores, en una sola ejecución.

## 5. CAPITULO 5: ANÁLISIS COMPUTACIONAL DE PROCESOS CONDUCTUALES EN INDIVIDUOS CON CARACTERÍSTICAS COGNITIVAS DIFERENTES

### 5.1 Introducción

El trastorno del espectro autista (en adelante, ASD, por sus siglas en inglés de *Autism Spectrum Disorder*) comprende un grupo de condiciones heterogéneas del neurodesarrollo, típicamente caracterizada por una tríada de síntomas que consiste en (1) una deficiente comunicación, (2) intereses restringidos, y (3) patrones de comportamiento repetitivo y estereotipados. Un diagnóstico temprano y preciso del autismo puede proporcionar la base para un programa educativo y tratamiento adecuado. En este capítulo, se propone un modelo computacional para la predicción del ASD, denominado MFCM-ASD. MFCM-ASD utiliza un Mapa Cognitivo Difuso Multicapa (en adelante, MFCM por sus siglas en inglés de *Multilayer Fuzzy Cognitive Map*), y está basado en instrumentos estandarizados para el diagnóstico del autismo. Los dos instrumentos sobre los cuales se basa el modelo predictivo son ADOS2 y ADIR: ADOS2 es una escala de observación para el diagnóstico del autismo, en su segunda edición (ADOS-2, por sus siglas en inglés de *Autism Diagnostic Observation Schedule, Second Edition*); y ADIR es una entrevista a padres o cuidadores de niños autistas (ADIR, por sus siglas en inglés de *Autism Diagnostic Interview Revised*) (ver sección 2.1).

Los MFCM son una técnica *soft-computing* (i.e., que son tolerantes a la imprecisión, a la incertidumbre y a verdades parciales), caracterizados por sus robustas propiedades que los convierten en una técnica eficaz para los sistemas de apoyo a la toma de decisión médica, o CDSS (por sus siglas en inglés de *Clinical Decision Support System*). Para la evaluación del modelo MFCM-ASD, se han utilizado diversas *datasets* con casos reales de diagnóstico del autismo, y confrontado con otros métodos/enfoques con el mismo objetivo. Los experimentos han

demostrado que el modelo propuesto MFCM-ASD supera los mapas cognitivos difusos (FCMs por sus siglas en inglés de *Fuzzy Cognitive Maps*) clásicos o convencionales usados para el diagnóstico del ASD (Puerto, Aguilar, López y Chávez 2019).

### 5.1.1 Generalidades

El ASD comprende un grupo de condiciones heterogéneas del neuro-desarrollo, típicamente caracterizadas por una tríada de síntomas, que consisten en: (1) una comunicación deficiente, (2) intereses restringidos, y (3) patrones de comportamiento repetitivo y estereotipados (American Psychiatric Association 2013; Fakhoury 2015; Willsey 2015). El proceso de diagnóstico médico del autismo es bastante complejo; por un lado, los criterios de diagnóstico son diversos, y por otro, cambian con el tiempo (Wozniak et al 2017). El gran número de características como las antes mencionadas, además de otros biomarcadores, síntomas, información cualitativa/cuantitativa involucrados en su proceso, requiere ser elicitada y analizada, para diagnosticar la severidad de esta condición (Ecker 2017).

Por otro lado, de acuerdo con Groumpus (Groumpus 2010; Groumpus et al 2008), los FCMs, constituyen una de las técnicas más eficientes y sólidas usadas en inteligencia artificial, para el desarrollo de sistemas de apoyo a la toma de decisiones médicas (MDSSs, por sus siglas en inglés) y otros sistemas complejos. Los FCMs permiten representar conocimiento desde una perspectiva cualitativa, y han sido usados para modelar sistemas complejos donde un modelo matemático exacto no tiene lugar debido a la gran complejidad del sistema. Recientemente, se han logrado importantes avances en el modelado de la toma de decisión médica usando FCMs (Amirkhani et al 2017; Al Farsi et al 2017; Mythili y Shanavas 2016; Papageorgiou y Salmeron 2013). Mythili y Shanavas (Mythili y Shanavas 2016) desarrollaron un sistema MDDS para la predicción temprana de la presencia de desórdenes cognitivos en niños con Autismo, Dislexia o Delirio. El sistema

considera características relacionadas con el aprendizaje, interacción social, comportamiento, incomprensión de situaciones u objetos, entre otros. El método de predicción usado involucra un enfoque basado en una meta-heurísticas y un FCMs, llamado MEHECOM. MEHECOM identifica algunos desórdenes cognitivos en individuos con características cognitivas diferentes, y define un conjunto de mecanismos que permitan aliviarlos. Al Farsi (Al Farsi et al 2017) ha definido un método difuso de diagnóstico del ASD basado en un FCM, el cual permite evaluar los conceptos de causalidad y de toma de decisión del mismo. Papageorgiou y Salmeron (Papageorgiou y Salmeron 2012a) han propuesto un sistema de decisión para el diagnóstico del ASD basado en un FCM, entrenado con el algoritmo de aprendizaje-hebbiano no lineal, considerando el conocimiento y la experiencia del experto. Subbaraju (Subbaraju, Sundaram y Narasimhan 2015) estudio sistemas/métodos de detección del ASD basado en la dataset ABIDE (especialmente casos de mujeres autistas), con clasificadores basados en diferentes técnicas, tales como las redes neuronales artificiales de funciones base radiales (RBF, por sus siglas en inglés)<sup>13</sup>.

Muchos de los modelos de diagnóstico previos están basados en el estándar *Cuestionario de Autismo en la Infancia Modificado* (MCHAT, por sus siglas en inglés de *Modified Checklist for Autism in Toddlers*), un instrumento con el cual se diagnostica exclusivamente a través de las respuestas de los padres y cuidadores. Otros trabajos sobre ASD usan otras técnicas inteligentes (ver sección 5.2.2 para más detalles). Algunos de los trabajos previos son interesantes, puesto que ellos han propuesto FCMs para el modelado computacional de diferentes aspectos alrededor del

---

<sup>13</sup> Las Redes Neuronales Artificiales de Funciones Base Radiales, son un subtipo de redes neuronales progresivas de la forma  $h(x) = \sum_n w_n \phi(\|x - x^n\|)$ , donde los vectores  $x^n$  se denominan centros. y  $w_n$  pesos. Aunque inicialmente fueron introducidas para resolver el problema de interpolación exacta, el estudio de las redes RBF ha evolucionado hasta constituir una disciplina de interés independiente en el campo de las ciencias de la computación y la inteligencia artificial, con numerosas aplicaciones en área como las finanzas, medicina, biología, geología, ingeniería o física (Park y Sandberg 1993)

autismo, pero en general, dichos trabajos sólo usan un nivel de conocimiento. i.e., los FCM propuestos son monocapa, por lo que cubren sólo una ventana de observación del fenómeno del autismo, con una limitada y restringida capacidad de inferencia. Por ejemplo, Kannappan (Kannappan, Tamilarasi y Papageorgiou 2011) cubre sólo el estándar MCHAT (F-MCHAT), y Bone (Bone, 2016) se limita a dataset para la estimación clínica del diagnóstico Autista (otros ejemplos se describen en 5.2.2).

En esta parte de la tesis, se propone un enfoque multicapas para modelar al fenómeno del autismo basado en trabajos previos que incluyen: el MCFM (Aguilar 2016; Aguilar, Hidalgo, Osuna y Pérez 2016), y la herramienta FCM Designer Tool (Aguilar y Contreras 2010; Aguilar 2016; Aguilar, Hidalgo, Osuna y Pérez 2016)). Para ello, fue necesario modificar el componente en el modelo MCFM, responsable de calcular los valores de la relación entre los conceptos, para poder modelar los algoritmos subyacentes a los instrumentos ADIR y ADOS2, y de esta manera poder predecir el ASD en casos reales de autismo. La capacidad de modelado de un MFCM es mucho mayor que la de un FCM, permitiendo la caracterización de diferentes aspectos del fenómeno en cuestión, en este caso del autismo (Aguilar 2013; Papageorgiou, Stylios y Groumos 2003). MFCM-ASD realiza un diagnóstico usando dos niveles de conocimiento sobre el autismo; el primero, la percepción de los padres/cuidadores; y el segundo, la observación directa del individuo, ambos basados en los instrumentos de diagnóstico ADIR y ADOS2, respectivamente (Lord, Corsello, y Grzadzinski 2014; Stabel 2013). Estos instrumentos fueron seleccionados por ser estándares, generalmente aplicados en conjunción, para la evaluación del ASD (Filipek 2000). ADIR es una entrevista semi-estructurada<sup>14</sup> que diagnóstica el autismo, diseñada para evaluar tres

---

<sup>14</sup> Es semiestructurada ya que el entrevistador lleva una pauta o guía con los temas a cubrir, los términos a usar, y el orden de las preguntas. Además, de las preguntas estructuradas que se le hacen al progenitor o cuidador, se permite alternar con preguntas espontáneas.

aspectos claves del autismo (Lord 2012; Rutter, LeCouteur y Lord 2015): comunicación social, comportamiento repetitivo, e intereses restringidos. Por otra parte, el ADOS2, es una medida observacional diseñada para evaluar la interacción social recíproca, la comunicación, y el uso de la imaginación del individuo (Lord 2012; Papanikolaou 2009). (Gray, Tonge y Sweeney 2008; Zander, Sturm y Bölte 2015) son trabajos importantes basados en estos estándares ADOS2 y ADIR, que los han validado en niños en edad preescolar con retrasos en el desarrollo, o con posibles trastornos generalizados del desarrollo (PDD, por sus siglas en inglés). De acuerdo con los resultados, es importante combinar el uso de ADOS2 y ADIR en individuos con problemas no fácilmente detectables en su desarrollo, incluida la sospecha de ASD.

Para demostrar la calidad del modelo propuesto MCFM-ASD, se utiliza un *dataset* con casos reales de diferentes trastornos autistas, obteniendo como resultado una mejor aproximación en la predicción del ASD, en comparación con los FCM reportados en la literatura (ver sección 5).

Este capítulo continua como sigue: la sección 2, presenta el marco teórico, que proporciona los principales aspectos sobre los procesos conductuales de individuos con características cognitivas diferentes en el autismo/ASD, el método usado MCFM, y una revisión de los modelos computacionales para el diagnóstico del ASD. La sección 3 describe el modelo computación MFCM-ASD propuesto. La sección 4 presenta las simulaciones realizadas y los resultados alcanzados. La sección 5 compara MFCM-ASD con trabajos previos, la sección 6 presenta las conclusiones y trabajos futuros, y finalmente, la sección 7 algunas consideraciones éticas. Este capítulo está basado en los trabajos (Puerto, Aguilar, López y Chávez 2019; Puerto 2017).

## 5.2 Marco Teórico

En esta sección se presentan los dos modelos de evaluación comportamental ADIR y ADOS2, que son la base para el proceso de inferencia del autismo. También, el método computacional usado, específicamente, el MFCM.

### 5.2.1 Fenómeno de estudio: Autismo (ASD) e Instrumentos de Diagnóstico ADIR y ADOS2

Los pasados 30 años han sido un periodo muy activo en el desarrollo de instrumentos para el diagnóstico del autismo, que abordan una necesidad tanto en la investigación como en el dominio clínico (Wozniak et al 2017; Parisi, Di Filippo, y Roccella 2015). Hoy por hoy, se han diseñado diversos instrumentos para capturar el comportamiento de los individuos con características cognitivas diferentes; en las áreas de comunicación, interacción social, y conductas repetitivas y restringidas. Estos instrumentos intentan cuantificar los comportamientos asociados con el autismo, mediante la asignación de una puntuación (score) numérica. Estos *scores* de evaluación del comportamiento, luego son trasladados a un resumen de score para clasificar al individuo como autista o sano (Wozniak et al 2017). Los actuales instrumentos de diagnóstico incluyen cuestionarios y entrevistas a padres, así como también medidas de observación estandarizadas (Ecker 2017; Amirkhani et al 2017; Lord, Corsello, y Grzadzinski 2014; McClintock y Fraser 2011; Filipek et al 2000; Papanikolaou 2009; Charman y Gotham 2013). Dos de estos instrumentos son el ADIR y ADOS2.

El ADIR es una entrevista clínica que permite una evaluación en profundidad del autismo a través de 93 preguntas, las cuales exploran tres mayores factores alterados por el autismo (Rutter, LeCouteur, Lord 2015): la calidad de la interacción social (e.x., compartir emocionalmente, ofrecer y buscar confort, sonreír socialmente y responder a otros niños); comunicación y lenguaje (ej., expresiones estereotipadas, cambio de pronombre, uso social del lenguaje); e intereses y

comportamientos repetitivos, restringidos y estereotipados (ej., preocupaciones inusuales, gestos de manos y dedos, intereses sensoriales inusuales) (Lord et al 2000; 2012). Este instrumento se aplica a niños cuyas edades mentales están sobre los 2 años.

Esta entrevista clínica (i.e., ADIR) es sintetizada en dos algoritmos: algoritmos de comportamiento actual y de diagnóstico. Estos algoritmos usan un score para cada área (lenguaje y comunicación, interacción social, y patrones de comportamiento repetitivos y restringidos). Los algoritmos tienen un score mínimo para cada área, a partir del cual se determina un diagnóstico de autismo. Por ejemplo, el umbral de corte (*cut score*) para el dominio de lenguaje y comunicación es de 8 para individuos verbales<sup>15</sup>, y 7 para individuos no verbales. El punto de corte para el dominio de interacción social para todos los individuos, es 10, y 3 para el patrón de comportamientos restringidos y repetitivos. Las evaluaciones de área con altos scores, indican comportamientos problemáticos en esa área. De acuerdo con expertos, una clasificación de autismo es dada cuando los scores en al menos dos de las tres áreas (comunicación, interacción social, y patrones de comportamiento) se cumplen, o exceden los estos puntos de corte (Rutter, LeCouteur y Lord 2015).

Respecto a ADOS2, la versión original (Módulo 1-4) fue desarrollada en Inglés por investigadores del *Western Psychological Service* (Lord 2012), y una segunda versión (Módulo T) en (Lord y Corsello 2005). El ADOS2 es una evaluación que se hace por observación del ASD. El

---

<sup>15</sup> Se refiere al criterio de diagnóstico relacionado con el déficit en conductas comunicativas no verbales usadas en la interacción social; rango de comportamientos que, por ejemplo, va desde mostrar dificultad para integrar conductas comunicativas verbales y no verbales; a anomalías en el contacto visual y el lenguaje corporal, o déficits en la comprensión y uso de gestos, o falta total de expresividad emocional o de comunicación no verbal. Por ejemplo, es común que los niños con un trastorno del espectro autista no puedan hacer gestos (como señalar un objeto, por ejemplo) para dar significado a lo que dicen.

ADOS2 incluye cinco (5) módulos, cada módulo comprende la evaluación de una serie de actividades, utilizando materiales de simulación interactivos. Un individuo es evaluado en un solo módulo, el cual es seleccionado con base a su nivel de expresividad a nivel de lenguaje y edad cronológica:

- Modelo T (*Toddler*)- para niños entre 12 y 30 meses de edad, sin un consistente uso del lenguaje hablado.
- Módulo 1- para niños de 31 meses de edad y más, sin un consistente uso del lenguaje hablado.
- Módulo 2- para niños de cualquier edad, sin fluidez para hablar.
- Módulo 3- para niños y jóvenes adolescentes, con fluidez verbal
- Módulo 4- para adolescentes y personas mayores, con fluidez verbal.

En los módulos 1 a 4, los valores de los algoritmos son comparados con los puntos de corte para dar una de las tres clasificaciones: *autismo, dentro del espectro autista, o sin autismo*. La diferencia entre el autismo y el espectro autista es la severidad del problema, la primera indica unos síntomas más pronunciados. Recientemente, Zander y otros (Zander, Sturm y Bölte 2015) han validado la calidad de ADIR y ADOS2 en una muestra clínica de niños con edades entre 18 y 47 meses. Esta validación se realizó para cada instrumento por separado, y luego combinados, frente a un diagnóstico con consenso clínico. Este trabajo es similar al nuestro, pero no se usa una herramienta computacional.

### **5.2.2 Modelos computacionales para predecir el ASD**

Hay diferentes enfoques computacionales para predecir el ASD: métodos computacionales basados en la evaluación del comportamiento (Rosenberg, Patterson y Angelaki 2015; Puerto 2017; Crippa et al 2015; Bone et al 2015; Galitsky 2013; Reyes, Ponce, Grammatikou y Molina

2014; Kannappan, Tamilarasi y Papageorgiou 2011; Ojeda 2015; Bone et al 2016; Papageorgiou y Kannappan 2012b), métodos computacionales basados en dato neurocientíficos (en cuanto a su estructura como a nivel funcional), y métodos híbridos basados en ambas características (Anirudh y Jayaraman 2017). Algunos métodos basados en el primer enfoque (evaluación del comportamiento) son: La red neuronal artificial propuesta por Cohen (Cohen 1993), para discriminar entre retardo mental y autismo, basada en ABI (ABI por sus siglas en inglés de *Autism Behavior Interview*), el cual utiliza como técnica de aprendizaje el algoritmo de Backpropation. El sistema neuro-difuso propuesto por Arthi y Tamilarasi (Arthi y Tamilarasi 2008), que convierte las respuestas de los padres en cuestionario con valores difusos. Estos valores difusos luego son evaluados con reglas “si-entonces”, y la salida difusa se convierte en la entrada para una previa red neuronal backpropagation. El sistema experto *Knowledge Based Screener* (KBS por sus siglas en inglés), el cual es un sistema con una base de hechos y conocimiento heurístico capaz de analizar el desarrollo infantil e identificar desordenes de desarrollo. También, Wall y otros (Wall et al 2012; Wall 2012) han propuesto una herramienta basada en árboles de decisión ADTree (ADTree, por sus siglas en inglés), que trabaja como un clasificador, basado en 8 preguntas del Módulo 1 del instrumento ADOS. Otra versión del modelo ADTree detecta autistas de manera rápida a través de 7 preguntas del instrumento ADIR (Wall et al 2012). Tarantino y otros (Tarantino et al 2016) han desarrollado una herramienta basada en TIC (Tecnologías de la Información y Comunicación) para ayudar a psicólogos y terapeutas en sus actividades de diagnóstico del autismo, vía el sistema de gamificación<sup>16</sup> *Gamified TEL System*. Ojeda (Ojeda 2015) ha definido un método basado en algoritmos genéticos, para realizar el diagnóstico del ASD. Bone y otros (Bone et al 2015, Bone

---

<sup>16</sup> Es una estrategia de enseñanza-aprendizaje tecnológica, usada para motivar y despertar las emociones que faciliten el proceso de aprendizaje del individuo.

et al 2016) han estudiado el uso del aprendizaje de máquina ML (por sus siglas en inglés de *Machine Learning*) en la detección del autismo. Ellos concluyen que el ML puede ser aplicado en el diagnóstico del ASD cuando se tiene una gran base de datos certificada. La mayoría de estos métodos tratan el problema del autismo desde una única perspectiva o ventana de observación, y requieren de una gran muestra para entrenar sus modelos.

Por otra parte, Subbaraju y otros (Subbaraju 2015) han propuesto un método de detección del autismo a partir de datos de imágenes de resonancia magnética estructural MRI (por sus siglas en inglés de *Magnetic Resonance Imaging*) usando un clasificador neuronal metacognitivo extendido con funciones de base radial EMcRBFN (por sus siglas en inglés de *Extended Metacognitive Radial Basis Function Neural Classifier*). Zhang y otros (Zhang et al 2017) han diseñado un método automático de análisis de la conectividad de la materia blanca basado en MRI y tractografía por difusión, para la detección del autismo. También, Anirudh y otros (Anirudh y Jayaraman 2017) han definido un método que combina diferentes propiedades (comportamentales, estructurales y funcionales), las cuales actúan como biomarcadores, en un modelo predictivo para diferentes condiciones neuro-patológicas. Específicamente, ellos desarrollan una versión de una red neuronal convolucional gráfica G-CNNs (por sus siglas en inglés de *Graph Convolutional Neural Networks*) para la clasificación, basada en tales ideas.

Finalmente, recientemente Abbas y otros (Abbas, Garberson y Glover 2016) han propuesto una herramienta para la detección temprana del autismo, aplicando algoritmos de aprendizaje de máquina. Esta herramienta combina dos métodos de screening<sup>17</sup> en una única evaluación, por un lado, cuestionarios cortos y estructurados orientados a padres; y por otro lado, videos caseros

---

<sup>17</sup> Estos métodos de screening, son análisis simples que se utilizan para la identificación o el descarte de patologías en pacientes. Se utilizan para obtener una información rápida sobre esas patologías. Generalmente, si el resultado es positivo, se necesita realizar un análisis más concluyente para confirmar la enfermedad.

semiestructurados y etiquetados con el comportamiento del niño. Adicional a esto, se ha propuesto un framework generalizado para utilizar algoritmos de aprendizaje automático que buscan simultáneamente la presencia de muchas condiciones en el contexto.

MFCM-ASD se basa en una evaluación del comportamiento del individuo, y usa la técnica de MFCM para modelar los procesos estandarizados de toma de decisiones ADIR y ADOS2. MFCM-ASD se implementó usando la herramienta FCM Designer Tool (Aguilar y Contreras 2010), que permite definir Modelos Cognitivos Difusos (FCMs) con conceptos y relaciones que pueden cambiar en el tiempo de ejecución (Aguilar 2016), y ha sido extendido para permitir la creación de Modelos cognitivos difusos multicapa (MFCM). Con esta extensión, es posible tener diversos FCMs para el mismo problema/fenómeno, donde cada capa/modelo expresa una parte del conocimiento del sistema y las relaciones entre ellos (Aguilar 2016).

### **5.2.3 Método usado por el MCFM-ASD: Mapas Cognitivos Difusos**

Los FCMs usan una representación simbólica para la descripción y modelamiento de un fenómeno de interés. Un FCM utiliza conceptos (variables, estados, entradas, salidas) para representar diferentes aspectos del comportamiento de un sistema, los cuales a su vez se relacionan para describir la dinámica del mismo (Groumpos, 2010; Aguilar 2001; E.S. Vergini y Groumpos 2016; Aguilar 2005; Papageorgiou, y Stylios 2008). Un FCM integra la experiencia y conocimiento acumulado sobre el funcionamiento de un sistema, de expertos humanos, quienes conocen el sistema y sus comportamientos en diferentes circunstancias. Los FCMs pueden describir cualquier sistema usando un modelo basado en la causalidad (que indica relaciones positivas y negativas), las cuales, a su vez, toman valores difusos y son dinámicas (i.e., el efecto de un cambio en un concepto/nodo afecta otros conceptos, los cuales a su vez pueden afectar otros). La parte difusa

permite grados de causalidad. La estructura del FCM establece la propagación de causalidad hacia adelante o hacia atrás (Papageorgiou y Iakovidis 2013).

Los mapas cognitivos se pueden representar gráficamente, donde los conceptos se relacionan mediante arcos, o a través de una matriz de conexión. En la matriz de conexión, la  $i$ -ésima línea presenta el peso del arco. En la matriz de conexión, la línea  $i$ -enésima representa el peso de las conexiones de los arcos dirigidas hacia fuera del concepto  $C_i$ , es decir, hacia aquellos conceptos  $C_j$  afectados por  $C_i$ . La  $i$ -enésima columna lista los arcos dirigidos hacia  $C_i$ , i.e., aquellos que afectan  $C_i$  (Kosko 1986; Glykas 2010) (ver Ecuación 5.1).

$$w_{i,j} = M(C_i, C_j) \quad (5.1)$$

Por otro lado,  $M$  representa la función causal que tiene el concepto precedente  $C_i$  con el concepto consecuente  $C_j$ , y  $w_{i,j}$  es el peso de la relación entre estos dos conceptos. En general, el concepto  $C_i$  influye incrementalmente en la causalidad de la ocurrencia de  $C_j$  si  $w_{i,j} = 1$ , decrece esta influencia de causalidad si  $w_{i,j} = -1$ , y no afecta en nada si  $w_{i,j} = 0$ .

Los FCMs, fueron inicialmente presentados como mecanismos difusos, donde los conceptos y las relaciones pueden ser representados como variables difusas (expresadas en términos lingüísticos) (Aguilar 2001). En un FCM, cada concepto depende de sus predecesores en iteraciones previas, y es calculado por medio de una suma de productos normalizada, donde la relación entre un concepto y sus predecesores es modelada por un peso, de acuerdo a la siguiente Ecuación 5.2 (Aguilar 2001):

$$C_m(i + 1) = S \left[ \sum_{k=1}^N w_{m,k} \cdot C_k(i) \right] \quad (5.2)$$

Donde  $C_m(i + 1)$  indica el valor del concepto en la siguiente iteración,  $N$  indica el número de conceptos,  $w_{m,k}$  indica el valor del peso de la relación causal del concepto  $C_k$  y el concepto  $C_m$ , y  $S[\cdot]$  es una función para normalizar el valor del concepto. La inicialización de cada concepto  $C_m(0)$  se realiza mediante el establecimiento de valores específicos basados en opiniones de expertos, o en un escenario específico de predicción para el cual el FCM es creado. A continuación, se presenta la extensión del modelo FCM como modelo MCFM.

### 5.2.3.1 Mapa Cognitivo Difuso Multicapa

Para construir el MFCM se ha modificado la Ecuación (5.2), con el fin de integrar la función generada por la interfaz a partir de las reglas que describen las relaciones entre los diferentes mapas (capas). En este sentido, la nueva ecuación matemática para calcular el estado actual de los conceptos en el FCM, se muestra en la siguiente Ecuación (5.3) (Aguilar 2016; Aguilar, Hidalgo, Osuna, y Pérez 2016).

$$C_m(i + 1) = S \left[ \sum_{k=1}^N w_{m,k} \cdot C_k(i) \right] + F(mp) \quad (5.3)$$

Esta nueva fórmula mantiene el significado de la Ecuación 5.2, excepto el nuevo elemento  $F(mp)$  que indica la función de entrada generada por la interfaz del mapa multicapa. De esta manera, la función de actualización de los conceptos tiene dos partes. La primera parte es la clásica, la cual calcula el valor del concepto  $C_m$  en la iteración  $i+1$  basada en los valores de los conceptos de la iteración  $i$ . Todos estos conceptos pertenecen a la misma capa a la que pertenece el concepto “ $m$ ”. La segunda parte tiene que ver con el cálculo de la relación causal entre los conceptos en diferentes FCMs (ver (Aguilar 2016) para más detalles). Este formalismo también ha sido incluido en la herramienta FCM Designer Tool (Aguilar 2016; Aguilar y Contreras 2010).

Con esta extensión, ahora es posible tener diversos FCMs para el mismo problema o fenómeno de estudio, donde cada uno de estos modelos (FCM individuales) expresan un nivel de conocimiento diferente del sistema bajo estudio, guardando su interrelación entre ellos (Aguilar 2016; Aguilar, Hidalgo, Osuna, y Pérez 2016). Con este nuevo modelo, se pueden tener diferentes niveles de conocimiento; un primer nivel de abstracción con información específica del sistema, y luego niveles más generales. Además, las variables de un nivel dependen de las de otros niveles. Es decir, el enfoque multicapa enriquece los sistemas modelados, con un flujo de información entre capas, para obtener información sobre los conceptos involucrados en una capa, a partir de los conceptos en otras capas. En el enfoque multicapa, las relaciones entre los mapas cognitivos entre diferentes capas ( $F(mp)$ ) pueden realizarse de varias maneras (Aguilar 2016; Aguilar, Hidalgo, Osuna, y Pérez 2016): con reglas difusas, conexiones con pesos, o con ecuaciones matemáticas. En este trabajo se ha utilizado un enfoque híbrido basado en conexiones con pesos y reglas difusas. Otros avances en la teoría de FCM se pueden encontrar en (Froelich 2017; Christoforou y Andreou 2017).

### **5.3 Modelo MFCM-ASD**

En general, al diagnosticar ASD, nuestro modelo MFCM-ASD sigue el proceso de toma de decisión de los instrumentos de diagnóstico estandarizados ADIR y ADOS2. En este apartado, primeramente, damos una descripción de los componentes del modelo propuesto MFCM-ASD, i.e., cuáles son sus conceptos y relaciones. Luego, se define el conjunto de reglas que sigue el modelo MFCM-ASD para actualizar la relación entre los conceptos.

Específicamente, el modelo MFCM-ASD es un modelo multicapa, que permite expresar y modelar las diferentes dimensiones de conocimiento requeridas por los instrumentos ADIR y ADOS2, usados para el diagnóstico del autismo. Una primera dimensión modela la información

que se captura del niño afectado aplicando ADOS2, y otra dimensión modela la información que se captura de la familia o cuidador/a de niño afectado con autismo, ambas dimensiones se complementan una de la otra. De esta manera, el modelo MFCM-ASD cubre naturalmente las diferentes dimensiones estandarizadas de conocimiento subyacentes en los instrumentos ADIR y ADOS2, ambos diseñados para su uso en la investigación del autismo.

### **5.3.1 Bases del modelo MFCM-ASD**

En esta subsección, son especificados los conceptos y las relaciones entre los conceptos del modelo MFCM-ASD.

#### **5.3.1.1 Descripción de los conceptos de MFCM-ASD**

En este estudio, los conceptos usados para modelar nuestro MFCM-ASD son extraídos de las observaciones del experto y de los instrumentos de diagnóstico ADIR y ADOS2. Los conceptos relevantes extraídos para el modelo propuesto, son descritos en las Tablas 5.1 y 5.2. Los conceptos de entrada representan los síntomas y signos/señales de autismo. Los conceptos de salida representan niveles de severidad/afectación de los síntomas.

Los conceptos A1, A2, A3, y A4 son conceptos de entrada y representan las alteraciones cualitativas de interacción social recíproca. El concepto TOTALA es un concepto de salida y define el nivel de severidad en la interacción social recíproca. Los conceptos B1, B4, B2(V) y B3(V) son conceptos de entrada, y representan las alteraciones cualitativas en la comunicación en sujetos no verbales. El concepto TOTALBV es un concepto de salida, y define el nivel de severidad en la comunicación en sujetos verbales. Los conceptos B1NV y B4NV son conceptos de entrada, y representan las alteraciones cualitativas de comunicación en sujetos no verbales. El concepto TOTALBNV es un concepto de salida, y define el nivel de severidad en la comunicación en sujetos

no verbales. Los conceptos C1, C2, C3 y C4 son conceptos de entrada, y representan patrones de comportamientos repetitivos, restringidos y estereotipados. El concepto TOTALC es un concepto de salida, y define el nivel de severidad de los patrones restringidos, repetitivos y estereotipados. Finalmente, el concepto de salida OUTADIR es un concepto de salida que representa la presencia (o ausencia) de Autismo. Esta clasificación se da cuando el score de al menos dos de los tres conceptos de salida (TOTALA, TOTALBV, TOTALBNV o TOTALC), cumplen o exceden sus puntos de corte específicos.

**Tabla 5.1.** Conceptos extraídos de ADIR usados en la primera capa de MFCM-ASD

Concepto	Descripción
A1	Incapacidad para utilizar comportamientos no verbales en la regulación de la interacción social.
A2	Incapacidad para desarrollar relaciones con los compañeros
A3	Falto de alegría o placer compartido
A4	Carece de reciprocidad social o emocional.
TOTALA	Total de alteraciones cualitativas de la interacción social recíproca.
B1	Falta o retraso en el lenguaje hablado e incapacidad para compensar esta carencia con gestos, en sujetos verbales.
B4	Falta de juego imaginativo o espontáneo y de un variado juego social imitativo, en sujetos verbales.
B2(V)	Relativa incapacidad para iniciar y mantener una conversación, en sujetos verbales.
B3(V)	Discurso estereotipado, repetitivo e idiosincrásico, en sujetos verbales.
TOTABV	Total de alteraciones cualitativas de la comunicación, en sujetos verbales. Estos conceptos solo son activos en sujetos verbales.
B1NV	Falta o retraso en el lenguaje hablado e incapacidad para compensar esta debilidad mediante gestos, en sujetos no verbales.
B4NV	Falta de juego imaginativo o espontáneo y variado juego social imitativo, en sujetos no verbales. Estos conceptos solo son activos en sujetos no verbales.
TOTALNV	Total de alteraciones cualitativas de la comunicación en sujetos no verbales.
C1	Absorbente preocupación o patrón de interés circunscrito.
C2	Apego aparentemente compulsivo a rutinas o rituales no funcionales
C3	Manierismos estereotipados y repetitivos.
C4	Preocupación por partes de objetos o elementos no funcionales de materiales
TOTALC	Total de patrones de comportamiento restringidos, repetitivos y estereotipados
OUTADIR	Se da una clasificación de Autismo o No autismo, cuando las puntuaciones en al menos dos de las tres áreas (comunicación, interacción social y patrones de comportamiento), cumplen o superan los límites especificados.

Por otra parte, en ADOS2, cada módulo evalúa tres elementos que describen los principales problemas relacionados con el autismo, en edades específicas. Estos tres elementos son: problemas

de comunicación, problemas de interacción social recíproca, y problemas de comportamiento restringido y repetitivo (con la excepción del módulo 4, el cual incluye un cuarto elemento: la imaginación y creatividad).

La Tabla 5.2 describe los conceptos extraídos de ADOS2, donde los conceptos C-ADOSMX, ISR-ADOSMX, y CRR-ADOSMX representan la información de entrada del Módulo T, Módulo 1, Módulo 2 o Módulo 3. El primer concepto (C-ADOSMX) representa problemas de comunicación; el segundo concepto (ISR-ADOSMX) representa problemas de interacción social recíproca, y el tercer concepto (CRR-ADOSMX) representa problemas de comportamiento repetitivos y restringidos. OUT-MX es un concepto de salida, que determina el nivel de discapacidad social y comportamiento repetitivo y restringido, evaluado con los algoritmos del Módulo X.

Los conceptos C-ADOSM4, ISR-ADOSM4 y CRR-ADOSM4 miden los mismos problemas antes mencionados. El concepto IC-ADOSM4 representa el problema particular de creatividad e imaginación. OUT-M4 es un concepto de salida, que determina el nivel de incapacidad social y de comportamiento restringidos y repetitivos, evaluados por el algoritmo del Módulo 4. OUT-ADOS2 es el concepto de salida, que representa el valor final del diagnóstico de acuerdo con ADOS2. Finalmente, OUT-TEA es el concepto que representa una medida que combina el grado de autismo, evaluado conjuntamente por los dos instrumentos de diagnóstico, ADIR y ADOS2.

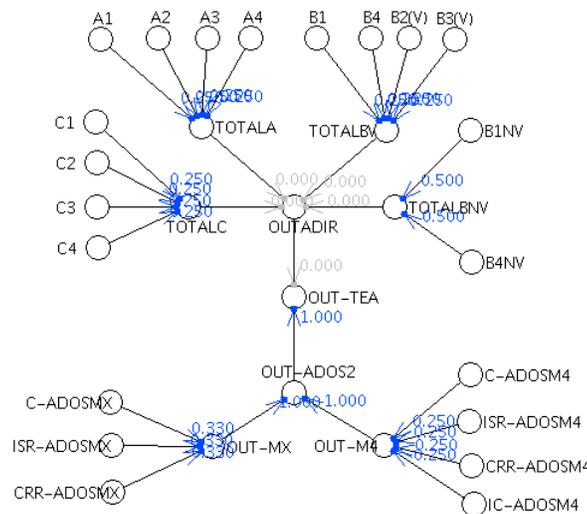
En Total, 30 conceptos constituyen el modelo MFCM-ASD; divididos en dos capas: 19 conceptos modelan el conocimiento subyacente de ADIR, y 10 conceptos entorno a ADOS2.

**Tabla 5.2.** Conceptos de ADOS2 usados en la segunda capa del modelo MFCM

Concepto	Descripción
C-ADOSMX	Problemas de comunicación, evaluados con los algoritmos del módulo X (X se refiere a los módulos T,1,2,3,4)
ISR-ADOSMX	Problema de interacción social recíproca, evaluado con los algoritmos del módulo X.
CRR-ADOSMX	Problemas de comportamiento repetitivo y restringido, evaluados con los algoritmos del módulo X.
OUT-MX	Nivel de inhabilidad social y comportamientos repetitivos y restringidos evaluados con los algoritmos del módulo X.
C-ADOSM4	Problemas de comunicación, evaluados con los algoritmos del módulo 4.
ISR-ADOSM4	Problemas de interacción social recíproca, evaluados con los algoritmos del módulo 4.
CRR-ADOSM4	Problemas de comportamiento repetitivo y restringidos, evaluados con los algoritmos del módulo 4
IC-ADOSM4	Problemas de imaginación y creatividad, evaluados con los algoritmos del módulo 4.
OUT-M4	Nivel de inhabilidad/incapacidad social y comportamientos repetitivos y restringidos, evaluados con los algoritmos del módulo 4.
OUT-ADOS2	Representa el valor de salida final del diagnóstico, de acuerdo con ADOS2
OUT-TEA	Representa el valor final de salida del diagnóstico, de acuerdo con las salidas parciales de los conceptos OUTADIR y OUT-ADOS2

Finalmente, una salida general conjuga la aplicación simultánea de ambos instrumentos. La

Figura 5.1 muestra el modelo general de MFCM-ASD usado para predecir el Autismo.



**Figura 5.1.** Modelo general de MFCM-ASD para predecir el ASD

En las sub-secciones siguientes, se describe cada parte del modelo MFCM-ASD de la Figura 5.1.

### 5.3.1.2 Descripción de las relaciones entre los conceptos de la capa ADIR

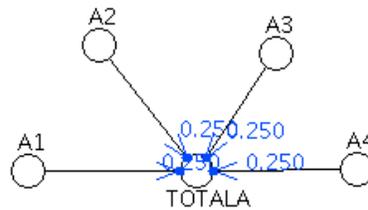
De acuerdo con Aguilar (Aguilar 2005; Aguilar 2013), hay dos maneras de establecer relaciones causales entre conceptos: (1) basada en la opinión de expertos (donde cada experto proporciona su modelo de matriz individual FCM, de acuerdo a su experiencia personal); y (2) y basada en datos históricos (datos de operación del sistema son usados como patrones de entrada para establecerlos). Para este trabajo, se tomó la primera opción (basada en la opinión de expertos), complementada con los instrumentos de diagnóstico del ASD. Los pesos son definidos basados en las opiniones de los expertos, considerando las relaciones entre los conceptos previamente definidos, de acuerdo con los instrumentos de diagnóstico ADIR y ADOS2. De acuerdo con sus opiniones, cada concepto involucrado en el dominio evaluado contribuye al diagnóstico en la misma medida (calidad de interacción social, comunicación y lenguaje, etc.). Por esa razón, cada peso de cada relación es 1 dividido por el número de conceptos de entrada del dominio.

Por otra parte, el valor de cada concepto es asignado por el experto de acuerdo al diagnóstico que realiza en cada instrumento usado, a saber, ADIR o ADOS2 (más específicamente, el algoritmo de diagnóstico empleado). Los valores de los conceptos dados por el experto son normalizados en el intervalo  $[0,1]$ . Seguidamente, se procede a describir las relaciones de cada capa que constituye el modelo general de MFCM-ASD para predecir el ASD de la Figura 5.1.

- *Descripción de las relaciones de la capa TOTALA*

Este dominio tiene 4 conceptos de entrada: A1, A2, A3, A4, y un concepto de salida TOTALA.

Cada relación tiene un peso de  $1/4=0.25$  (ver la Figura 5.2).

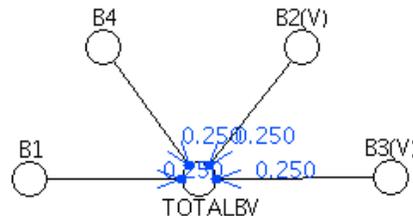


**Figura 5.2.** Conceptos y relaciones de las alteraciones cualitativas de la interacción social recíproca

El experto, de acuerdo con lo que recaba usando ADIR, asigna el valor de los pesos a los conceptos A1, A2, A3 y A4. El concepto TOTALA se activa cuando el punto de corte (*cut-off*) es alcanzado o superado. El *cut-off* para este dominio es de 0.33.

- *Descripción de las relaciones de la capa TOTALBN*

Este dominio también tiene 4 conceptos de entrada B1, B2(V), B3(V) y B4, y un concepto de salida TOTALBV. Cada relación tiene un valor de  $1/4=0.25$  (ver la Figura 5.3). El experto con base en lo encontrado con ADIR, asigna los valores de los pesos de los conceptos B1, B4, B2(V) y B3(V).



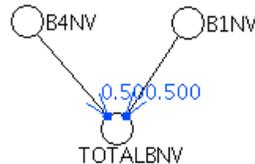
**Figura 5.3.** Conceptos y relaciones de las alteraciones cualitativas de comunicación (sujeto verbal)

El concepto TOTALBV es activado cuando el punto de corte (*cut-off*) es alcanzado o superado. El *cut-off* para este dominio es de 0.30.

- *Descripción de las relaciones de la capa TOTALBNV*

Este dominio cuenta con dos conceptos de entrada B4NV, B1NV, y un concepto de salida TOTALBNV. Cada relación tiene un peso de  $1/2=0.25$  (ver Figura 5.4). El experto, de acuerdo

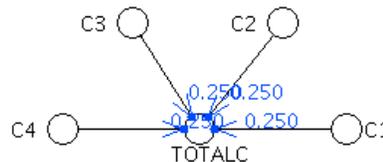
con ADIR, da los valores de los conceptos B4NV y B1NV. El concepto TOTALBNV es activado cuando el punto de corte para este dominio es superado. El punto de corte normalizado para este caso es de 0.27.



**Figura 5.4.** Conceptos de alteraciones cualitativas de comunicación (sujeto no verbal)

- *Descripción de las relaciones de la capa TOTALC*

Este dominio tiene cuatro conceptos C1, C2, C3 y C4, y un concepto de salida TOTALC. Cada relación tiene un peso de  $1/4=0.25$  (ver Figura 5.5). El experto, de acuerdo con ADIR, da los valores de los conceptos C1, C2, C3 y C4. El concepto TOTALC es activado cuando el punto de corte para este dominio es superado. El punto de corte normalizado para este caso es de 0.25.

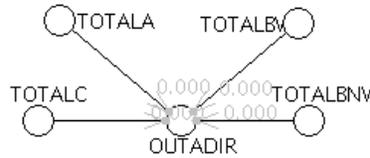


**Figura 5.5.** Conceptos del patrón de comportamiento repetitivo y estereotipado

*Descripción de las relaciones de la capa OUTADIR*

Este dominio tiene cuatro conceptos de entrada TOTALA, TOTALBV, TOTALBVN, y TOTALC, y un concepto de salida OUTADIR. Para esta capa, los pesos de todas las relaciones por defecto son iguales a cero (0) (ver Figura 5.6). En esta capa, a diferencia de las anteriores, los pesos de las relaciones son activados dinámicamente cuando el valor de los conceptos previos (TOTALA, TOTALBV, etc.,) han alcanzado el punto de corte. Si dos de tres de estos conceptos cumplen o exceden el punto de corte, entonces el concepto OUTADIR es activado. En este dominio, los

conceptos TOTALBV y TOTALBNV son excluyentes, i.e., ellos no pueden ser activados simultáneamente; cuando uno es activado el otro queda apagado.



**Figura 5.6.** Conceptos y relaciones de causalidad del autismo basada en ADIR

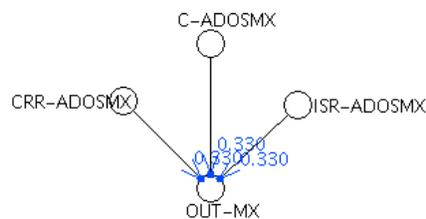
Cada relación tiene un peso de  $1/3 = 0.33$ . El umbral de activación definido para el concepto OUTADIR es de ( $\Delta=0.66$ ). Cuando un concepto excede el punto de corte, este valor pasa a ser 1 y el peso de la relación es establecido a 0.33. Las posibles combinaciones de activación del concepto OUTADIR se muestran en la Tabla 5.3.

**Tabla 5.3.** Combinaciones de activación del concepto OUTADIR

CONCEPTO	TOTALA	TOTALBV-BNV	TOTALC	OUTADIR
ACTIVATION	On	Off	Off	Off
	Off	On	Off	Off
	Off	Off	On	Off
	On	On	Off	On
	On	Off	On	On
	Off	On	On	On
	On	On	On	On

### 5.3.1.3 Descripción de las relaciones entre los conceptos de la capa ADOS.

- *Descripción de las relaciones de la capa OUT-MX*

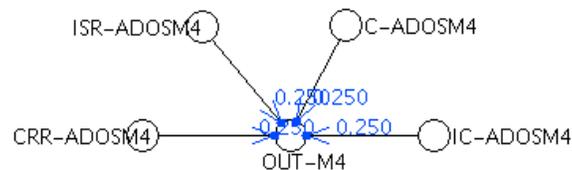


**Figura 5.7.** Conceptos y relaciones de la capa OUT-MX

Esta capa está constituida por cuatro módulos (Módulo T, Módulo 1, Módulo 2 y Módulo 3). Los cuatro módulos son agrupados en uno, puesto que todos ellos evalúan el mismo elemento (Comunicación, interacción social recíproca, y patrones de comportamiento repetitivos y estereotipados), sólo que para rangos diferentes de edades y restricciones de lenguaje. Así, este dominio tiene tres conceptos de entrada: C-ADOSMX, CRR-ADOSMX y ISR-ADOSMX. Cada relación tiene un peso de  $1/3=0.33$  (ver la Figura 5.7). Los valores de los conceptos son asignados por el experto, de acuerdo con en el instrumento de observación aplicado, y son normalizados en  $[0,1]$ .

- *Descripción de las relaciones de la capa OUT-M4*

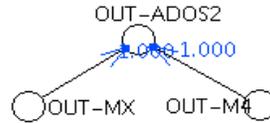
Esta capa se corresponde con el Módulo 4. Este dominio tiene cuatro conceptos de entrada: C-ADOSM4, CRR-ADOSM4, ISR-ADOSM4 y IC-ADOSM4, y un concepto de salida OUT-M4. Cada relación tiene un peso de  $1/4 = 0.25$  (ver Figura 5.8). Los valores de los conceptos son asignados por el experto, de acuerdo con el instrumento de observación aplicado, y son normalizados en  $[0,1]$ .



**Figura 5.8.** Conceptos y relaciones de la capa OUT-M4

- *Descripción de las relaciones de la capa OUT-ADOS2*

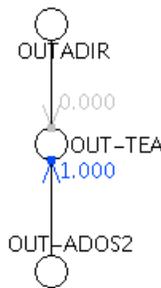
Sin importar el modelo de observación usado de ADOS2 (ya sea OUT-MX o OUT-M4), su contribución a la salida general OUT-ADOS2 es la misma (estos tienen un igual peso en la relación de (1.0)) (ver Figura 5.9). Los valores de los conceptos son asignados por el experto, de acuerdo con en el instrumento de observación aplicado, y son normalizados en  $[0,1]$ .



**Figura 5.9.** Conceptos y relaciones de la capa OUT-ADOS2

#### *Descripción de las relaciones de la capa OUT-TEA*

La conexión de los dos mapas (ADIR y ADOS2) se realiza mediante reglas de conexión directa, en las que se define el concepto de origen, el concepto de destino, y el peso de la conexión. Específicamente, la salida del concepto OUT-TEA se obtiene de uno de los dos instrumentos usados; cuando los dos instrumentos son aplicados simultáneamente, predomina la salida que viene de OUT-ADOS2 (ver Figura 5.10).



**Figura 5.10.** Salida general OUT-TEA

### **5.3.2 Reglas seguidas por el modelo MFCM-ASD en el proceso de toma de decisión**

A continuación, se especifican las reglas usadas por el modelo MFCM-ASD para el proceso de toma de decisión automático. Estas reglas se corresponden con los algoritmos de toma de decisión de los instrumentos de diagnóstico ADIR y ADOS2.

El modelo propuesto MFCM-ASD se basa en MFCM (ver sección 2.3.1). De ahí, que en la etapa de diseño del MFCM-ASD, se modificó una parte del método MCFM. El principal cambio fue en el mecanismo de cálculo del valor de las relaciones entre conceptos. Este mecanismo consiste de un conjunto de reglas con las que se calcula dicho valor. Algunas de las reglas definidas

para calcular el valor de las relaciones entre los conceptos del modelo MFCM-ASD, se especifican a continuación:

**Regla 1:**

1. IF el concepto inicial de la relación es igual a “OUTADIR” AND el concepto final de la relación es igual a “OUT-TEA” THEN
  - 1.1 IF el punto de corte de ADIR es excedido THEN el valor de la relación es 1.0.
  - 1.2 ELSE el valor de la relación es 0.0.

La línea 1 de la Regla 1 determina la relación en cuestión; la línea 1.1 determina si el valor del punto de corte general para el diagnóstico con ADIR ha sido alcanzando. Si es así, entonces es asignado un valor de 1.0 al peso de la relación (muestra signos de autismo); si no, en la línea 1.2 es asignado el valor de 0.0 a la relación (que no muestra signos de autismo).

**Regla 2:**

2. IF el concepto inicial de la relación es igual a “TOTALA” AND el concepto final de la relación es igual a “OUTADIR” THEN
  - 2.1 Obtener el valor de la relación
  - 2.2 IF el punto de corte de A es excedido THEN el valor de la relación es de 0.33.
  - 2.3 ELSE el valor de la relación es 0.0.

La línea 2 de la Regla 2 determina la relación a ser tratada. La línea 2.1 obtiene el valor actual de la relación. Si el valor que tiene excede el punto de corte o umbral de A, entonces es actualizado el peso de la relación a 0.33 (que significa que un síntoma de autismo está presente); si no, la línea 2.3 cambia el valor a 0.0 (que significa que esta dimensión no está afectada). En total, 26 reglas constituyen el núcleo del mecanismo de toma de decisiones de MFCM-ASD, que representan la lógica de diagnóstico subyacente en ADIR y ADOS2.

## 5.4 Simulación y Resultados

Esta sección presenta los diferentes experimentos realizados con el modelo MCFM-ASD. Los experimentos son clasificados en tres grupos: Un primer grupo de experimentos, con casos de autismos diagnosticados con ADIR, luego, un siguiente conjunto de experimentos analizados con ADOS2, y finalmente, un tercer grupo usando ambos instrumentos de diagnóstico.

### 5.4.1 Data experimental

En esta subsección, se describe los dataset usados para las pruebas del modelo MCFM-ASD.

*Participantes.* La muestra de estudio fue de 300 niños: 150 de un grupo clínico (diagnosticados con síndrome de asperger<sup>18</sup> o autismo), y otros 150 niños neurotípicos<sup>19</sup>, es decir, niños libres de esta condición. En el grupo clínico, el 30.2% fueron diagnosticados con autismo, y el 14.3% dentro del espectro autista. Los niños tenían entre 2 y 12 años de edad, de los cuales 76 eran niñas y 224 niños. Esta información ha sido proporcionada por la Asociación de Padres y Hermanos para el Apoyo y la defensa de los individuos con Autismo (APADA, por sus siglas en inglés) del Ecuador (ver sección 7 sobre responsabilidad ética).

*Procedimiento de muestreo.* El diagnóstico se llevó a cabo mediante la aplicación de ADIR a los padres y ADOS2 a los niños, junto con otros estudios especializados, como tomografías computarizadas, resonancias y estudios clínicos. Las pruebas se aplicaron en escenarios adecuados para proporcionar ambientes acogedores, sin distracciones y bien iluminados, con muebles y privacidad adecuados para la comodidad de los entrevistados e investigadores, y se utilizaron

---

<sup>18</sup> El síndrome de asperger forma parte de los trastornos del espectro autista.

<sup>19</sup> Neurotípico significa aquella persona que no está en el espectro ASD, o alguien neurológicamente típico.

materiales de prueba originales para cada prueba. En este estudio, los instrumentos fueron aplicados por tres profesionales certificados.

#### 5.4.2 Experimentos con casos diagnosticados con ADIR

En esta sección, se presentan y se describen los experimentos de pruebas realizados con ADIR. En nuestro sistema, los expertos dan los valores a los conceptos de entrada analizando las diferentes opciones de ADIR. Para los conceptos de entrada se ha definido el siguiente vector de entrada:  $ADI^0 = \{A1, A2, A3, A4, TOTALA, B1, B4, B2(V), B3(V), TOTALBV, B1NV, B4NV, TOTALBNV, C1, C2, C3, C4, TOTALC, OUTADIR\}$ . El concepto OUTADIR ha sido considerado por los expertos como el concepto de salida de decisión final o DOC (por sus siglas en inglés de *Decision Output Concept*), y puede ser categorizado como: *No existe evidencia de síntomas (NES, por sus siglas en inglés)*, *Moderada evidencia de síntomas (MES, por sus siglas en inglés)* y *Definitiva evidencia de síntomas (DES, por sus siglas en inglés)*. Estas tres categorías se caracterizan por sus valores  $NES = 0$ ,  $MES = 0.66$  y  $DES = 0.99$ , respectivamente. MES ocurre cuando dos de los tres elementos de diagnóstico superaron el corte. Por otra parte, DES se da cuando los tres elementos de diagnóstico superan el corte, y NES cuando ninguno o como máximo uno ha alcanzado el punto de corte. Para este último caso, es decir cuando sólo un elemento de diagnóstico está presente, nos dice que la persona tiene comportamiento anormal específico, pero que, dado que este puede ser probablemente debido a otros trastornos del desarrollo, ya sea neurológicos o psiquiátricos, se trata como NES.

Ahora, se describen diferentes casos de niños con diagnóstico de autismo, y para lo cual se ha usado el instrumento ADIR. ADIR usa dos algoritmos: *un primer algoritmo que evalúa el comportamiento actual*, y *un segundo algoritmo de diagnóstico*. Cada algoritmo considera

diferentes elementos de autismo, de acuerdo con la edad cronológica. El modelo propuesto MFCM-ASD cubre ambos algoritmos.

*CASO ADIR-1.* CA1 (el nombre de los niños ha sido cambiado para guardar su verdadera identidad) es un niño con una edad cronológica de 3 años y medio, y a quien fue aplicado el instrumento de diagnóstico ADIR. En este caso, los valores iniciales de cada concepto son:  $A1=2$ ,  $A2=4$ ,  $A3=6$ ,  $A4=7$ ,  $TOTALA=18$ ,  $B1=0$ ,  $B4=0$ ,  $B2(V)=0$ ,  $B3(V)=0$ ,  $TOTALBV=0$ ,  $B1NV=8$ ,  $B4NV=5$ ,  $TOTALBNV=13$ ,  $C1=2$ ,  $C2=0$ ,  $C3=2$ ,  $C4=2$ ,  $TOTALC=6$  y  $OUTADIR=DES$ . El vector de diagnóstico dado por el experto para este caso es:  $ADI^1 = \{ 2, 4, 6, 7, 19, 0, 0, 0, 0, 0, 8, 5, 13, 2, 0, 2, 2, 6, DES \}$ . Luego, estos valores iniciales dados por el experto son normalizados en el vector  $S^1$ , para realizar el proceso de simulación con MCFM-ASD.  $S^1 = \{ 0.33, 1, 1, 0.7, 0, 0, 0, 0, 0, 0, 1, 0.83, 0, 0.5, 0, 1, 1, 0, 0 \}$ . Los valores correspondientes a los conceptos  $TOTALA$ ,  $TOTALBV$ ,  $TOTALBNV$ ,  $TOTALC$ , y  $OUTADIR=DES$  se dejan por defecto en 0. El MFCM-ASD infiere estos valores. Haciendo uso de la herramienta FCM Designer Tool, se asignan los valores a los conceptos de entrada en nuestro modelo, como se muestra en la Figura 5.11. La ventana de la Figura 5.11 muestra el proceso de inicialización de los conceptos. Permitiendo asignar el nombre al concepto, su valor inicial, la posición y un comentario que lo describe. Una vez que todos los valores de los conceptos han sido cargados, el modelo MCFM-ASD es ejecutado, los resultados son presentados en la Tabla 5.4. El valor del concepto  $AUTADIR=0.98$  significa que los tres elementos previos están presentes, dando como resultado DES, i.e., una *Evidencia definitiva de Síntomas de Autismo* (tal como ha sido diagnosticado por el experto). El valor en la cuarta iteración corresponde al concepto de salida OUT-TEA (0.98) (ver Tabla 5.4).





punto de corte. En la segunda iteración, el valor del concepto  $OUTADIR = 0.99$  significa que los tres elementos previos están presentes, dando como resultado la *Definitiva Evidencia de Síntomas*, como ha sido especificado por el experto.

### 5.4.3 Experimentos con casos de autismo aplicando ADOS2

Similar a la sección de experimentos previa, los valores de los conceptos iniciales para ejecutar nuestro sistema de diagnóstico MCFM-ASD, son dados por el experto, analizando las diferentes opciones que le permite el instrumento de diagnóstico ADOS2.

A continuación, se describen diferentes escenarios de casos de autismo diagnosticados con ADOS2. Este caso usa distintos algoritmos: El Módulo T usa dos algoritmos, uno para niños pequeños/mayores con pocas o nulas palabras, y otro para niños mayores con algunas palabras. El Módulo 1 usa los algoritmos de pocas o nulas palabras y con algunas palabras. El Módulo 2 usa los algoritmos de niños menores de 5 años y mayores de 5 años. Los Módulos 3 y 4 tienen un solo algoritmo. Para los conceptos de entrada de ADOS2 se ha definido el siguiente vector de entrada:  $ADO^0 = \{C-ADOSMX, ISR-ADOSMX, CRR-ADOSMX, OUT-MX, C-ADOSM4, ISR-ADOSM4, CRR-ADOSM4, IC-ADOSM4, OUT-M4, OUT-ADOS2\}$ . El concepto  $OUT-ADOS2$ , determina el nivel/score de severidad de acuerdo a los Módulo de ADOS2. La Tabla 5.7 muestra el punto de corte y la clasificación estandarizada (de acuerdo con *Michigan State Department of Health and Human Services* (Application et al 2017)) para el Módulo 1.

De acuerdo con la Tabla 5.7, un niño con pocas o sin palabras es diagnosticado con autismo, cuando el total es mayor que el punto de corte para Autismo (16 o mayor). Por otro lado, un niño con pocas o sin palabra es diagnosticado con autismo o fuera del espectro, cuando el total es menor que el punto de corte para el espectro autista (10 o menos). Similares tablas son definidas para los otros módulos [67]. Los expertos han considerado a  $OUT-ADOS2$  como el concepto de salida de

decisión el cual puede ser categorizado como Fuera del Espectro o NS (por sus siglas en inglés de Non-Spectrum), Dentro del Espectro Autista o ASD (por sus siglas en inglés) y Autismo (AUT).

**Tabla 5.7.** Caso Módulo 1. Módulo 1 total puntos de corte

	POCAS/SIN PALABRAS	ALGUNAS PALABRAS
AUTISMO	16	12
AUTISMO EN EL ESPECTRO	11	8
<b>ASIGNAR LA CLASIFICACIÓN ADOS2:</b>		
AUTISMO	El total es mayor o igual que el punto de corte para el Autismo	
	<ul style="list-style-type: none"> <li>• Pocas o sin palabras-Total es 16 o mayor</li> <li>• Algunas palabras-Total es 12 o mayor</li> </ul>	
AUTISMO EN EL ESPECTRO	El total es igual o mayor que el punto de corte del espectro del autismo, pero menor que el punto de corte para el autismo.	
	<ul style="list-style-type: none"> <li>• Pocas o sin palabras- Total es de 11 a 15</li> <li>• Algunas palabras- Total es de 8 a 11</li> </ul>	
FUERA DEL ESPECTRO	El total es menor que el punto de corte del espectro autista	
	<ul style="list-style-type: none"> <li>• Pocas o sin palabras- Total es 10 o menor</li> <li>• Algunas palabras- Total es de 7 o menor</li> </ul>	

Nuestro sistema MFCM-ASD usa una escala normalizada en el rango  $[0,1]$  para inferir estas respuestas. Una descripción más detallada del rango de valores relacionados con la edad usada por nuestro sistema, se muestra en la Tabla 5.8 (similares tablas son definidas para los otros Módulos).

CASO ADOS2-MÓDULO 1. En este caso, tenemos un niño no verbal de 7 años, a quien se le aplicó el algoritmo de diagnóstico del módulo 1. El vector diagnóstico dado por el experto es:  $ADO^1 = \{4, 15, 7, 26, 0, 0, 0, 0, 0, AUT\}$ . El niño está en el espectro autista, con un alto nivel de síntomas (AUT). Posteriormente, se realiza el proceso de normalización de los datos, y los valores iniciales del proceso de simulación son  $S^1 = \{1, 0.93, 0.87, 0, 0, 0, 0, 0, 0, 0\}$ . Una vez todos los valores de los conceptos son cargados, el modelo MFCM-ASD es ejecutado. Los resultados se muestran en la Tabla 5.9.







**Tabla 5.12.** Resultados de clasificación del MCFM-ASD en los casos ADIR

	<b>NES</b>	<b>MES</b>	<b>DES</b>
<b>ADIR</b>	8/7	10/8	43/43

**Tabla 5.13.** Resultados de clasificación del modelo MCFM-ASD en los casos ADOS2

	<b>NS</b>	<b>ASD</b>	<b>AUT</b>
ADOS2- MÓDULO T	No dataset available	3/3	2/2
ADOS2-MODULE 1	8/8	7/7	5/5
ADOS2-MODULE 2	4/4	3/3	3/3
ADOS2-MODULE 3	5/5	8/7	5/5
ADOS2-MODULE 4	No dataset available	3/3	3/3

La precisión del diagnóstico de MCFM-ASD calculada para los casos con ADIR, viene dada por la ecuación 5.4:

$$Precisión\_mcfm\_ADIR = \frac{8/7 + 10/8 + 43/43}{3} = 89.2\% \quad (5.4)$$

El modelo MCFM-ASD es bastante preciso en casos donde hay una definitiva evidencia de síntomas (DES =43/43). No tanto para casos con una moderada evidencia de los síntomas (MES=10/8), o en aquellos casos que no presentan una evidencia de los síntomas (NES = 8/7). De ahí que su rendimiento general sea del 89.2 % de precisión.

Por otra parte, la precisión del diagnóstico de MCFM-ASD calculada para los casos con ADOS2, viene dada por:  $(3/3 + 2/2 + 8/8 + 7/7 + 5/5 + 4/4 + 3/3 + 3/3 + 5/5 + 8/7 + 5/5 + 3/3 + 3/3) / 13 = 99\%$ .

A diferencia de las simulaciones anteriores de ADIR, que dieron lugar a algunos casos en los que no pudo detectar casos de autismo ligero o sin autismo, las simulaciones de ADOS2 fueron muy exitosas, alcanzando el 99% de precisión. La precisión de los instrumentos evaluados por

separado es consistente con el hecho de que ADOS proporciona un mejor diagnóstico que ADIR (Zander, Sturm y Bölte 2015).

## 5.5 Comparación con Otros Métodos

En esta sección se realizan otras comparaciones cuantitativas y cualitativas con otros sistemas. Una primera comparación cuantitativa se realiza con el modelo computacional (FCM) de Kannappan (Kannappan, Tamilarasi y Papageorgiou 2011), un modelo similar al nuestro, salvo que usa otro instrumento para el diagnóstico de ASD. La comparación cualitativa se basa en la calidad del instrumento para el diagnóstico del autismo. Finalmente, la última comparación es con algoritmos de aprendizaje automático, utilizados para el diagnóstico de autismo.

### 5.5.1 Comparación Cuantitativa

En Kannappan (Kannappan, Tamilarasi y Papageorgiou 2011), se ha propuesto un modelo para el diagnóstico del autismo usando FCM, basado en el instrumento estándar MCHAT (F-MCHAT). Este modelo usa un enfoque de *soft-computing*, junto con un algoritmo de *Aprendizaje Hebbiano no Lineal* o NHL (por sus siglas en inglés de *Nonlinear Hebbian Learning*), para la estimación del ASD. Los 24 conceptos modelados en el FCM se listan en la segunda columna de la Tabla 5.14. La tercera columna de la misma tabla es la equivalente en nuestro modelo MCFM-ASD.

Esta equivalencia se hizo con el fin de usar el mismo dataset para realizar las mismas pruebas, y de esta manera compararlos. Esta comparación es importante, porque ellos usan el mismo paradigma computacional de MCFM-ASD. Así, siguiendo la misma notación para el vector de entrada definido en la sección 4.2., un vector de entrada general para el modelo propuesto en (Kannappan, Tamilarasi y Papageorgiou 2011) es:  $V = \{C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, OUTC1\}$ , donde los

primeros 23 valores corresponden a los conceptos de entrada evaluados por el experto, y el ultimo valor corresponde al Concepto de Decisión de Salida DOC = OUTC1.

**Tabla 5.14.** Conceptos del FCM propuesto en (Kannappan, Tamilarasi y Papageorgiou 2011) y su equivalencia aproximada en MCFM-ASD

#	MCHAT	MCFM-ASD (ADOS2)
C1	Disfruta siendo mecido/balanceado	CRR-ADOSMX
C2	Se interesa por otros niños	ISR-ADOSMX
C3	Le gusta subirse a los sitios	CRR-ADOSMX
C4	Disfruta Jugando	ISR-ADOSMX
C5	Simula cosas	C-ADOSMX
C6	Señala para pedir (dedo índice)	C-ADOSMX
C7	Señala para mostrar interés	ISR-ADOSMX
C8	Juega con juguetes pequeños	CRR-ADOSMX
C9	Muestra cosas	ISR-ADOSMX
C10	Mira a los ojos	C-ADOSMX
C11	Parece hipersensible al ruido	CRR-ADOSMX
C12	Sonríe como respuesta a su cara o sonrisa	ISR-ADOSMX
C13	Imita	C-ADOSMX
C14	Responde a su nombre	ISR-ADOSMX
C15	Mira a donde se le indica un objeto	C-ADOSMX
C16	Camina	CRR-ADOSMX
C17	Mira lo que ud mira	ISR-ADOSMX
C18	Hace movimientos inusuales o extraños cerca de su cara	CRR-ADOSMX
C19	Intenta atraer la atención	ISR-ADOSMX
C20	Escucha bien	CRR-ADOSMX
C21	Comprende lo que otros dicen	C-ADOSMX
C22	Se eleva mira a la nada	CRR-ADOSMX
C23	Mira a su cara para comprobar reacción cuando se enfrenta a algo extraño	ISR-ADOSMX
OUTC1	Autismo (Alto, Probable Autismo y No autismo)	OUT-MX

Kannappan (Kannappan, Tamilarasi y Papageorgiou 2011) ha usado como medida para la clasificación del autismo el siguiente baremo:

- Individuo con un claro diagnóstico de Autismo, o DA (por sus siglas en inglés de *Definite Autism*), si su valor de diagnóstico esta entre 0.41 y 1.00 ( $0.41 \leq DA \leq 1.00$ ).

- Individuo con un probable diagnóstico de Autismo, o PA (por sus siglas en inglés de *Probable Autism*), si su valor de diagnóstico esta entre 0.26 y 0.40 ( $0.26 \leq PA \leq 0.40$ ).
- Individuo sin Autismo, o NA (por sus siglas en inglés de *No Autism*), si su valor de diagnóstico esta entre 0.0 y 0.26 ( $0.0 \leq PA \leq 0.25$ ).

Para comparar los resultados de este modelo con el nuestro, realizamos un proceso de equivalencia entre los instrumentos MCHAT y ADOS2. La equivalencia se muestra en las columnas 2 y 3 de la Tabla 5.14. En nuestro modelo, el vector de entrada es reducido a tres valores  $ADO^0 = \{C\text{-ADOSMX}, \text{ISR-ADOSMX}, \text{CRR-ADOSMX} \dots\}$ , donde el valor de cada concepto en nuestro vector es el valor promedio de los conceptos equivalentes (Kannappan, Tamilarasi y Papageorgiou 2011), esto es

- $CR\text{-ADOSMX} = (C1 + C3 + C8 + C11 + C16 + C18 + C20 + C22) / 8$
- $ISR\text{-ADMX} = (C2 + C4 + C7 + C9 + C12 + C14 + C17 + C19 + C23) / 9$
- $C\text{-ADOSMX} = (C5 + C6 + C10 + C13 + C15 + C21) / 6$

Específicamente, fue seleccionado y usado el algoritmo de diagnóstico Módulo 1 de ADOS2. Aplicando nuestro modelo MCFM-ASD a los tres casos de base descrito en (Kannappan, Tamilarasi y Papageorgiou 2011), se tiene los siguientes resultados (La Tabla 5.15 muestra la comparación de los resultados).

**Tabla 5.15.** Comparación de los resultados de diagnóstico

EXPERTO	MFCM-ASD	FCM [48]
0.51 = DA	0.50 = AUT	0.73 = DA
0.41 = PA	0.28 = ASD	0.37 = PA
0.87 = DA	0.60 = AUT	0.659 = DA

*Case I:* Vector de entrada dado por el experto  $V^1 = \{0.3, 0.55, 0.6, 0.65, 0.2, 0.69, 0.73, 0.77, 0.86, 0.1, 0.57, 0.4, 0.5, 0.62, 0.6, 0.71, 0.9, 0.15, 0.25, 0.45, 0.49, 0.34, 0.62, \mathbf{0.51}\}$ . Diagnóstico:

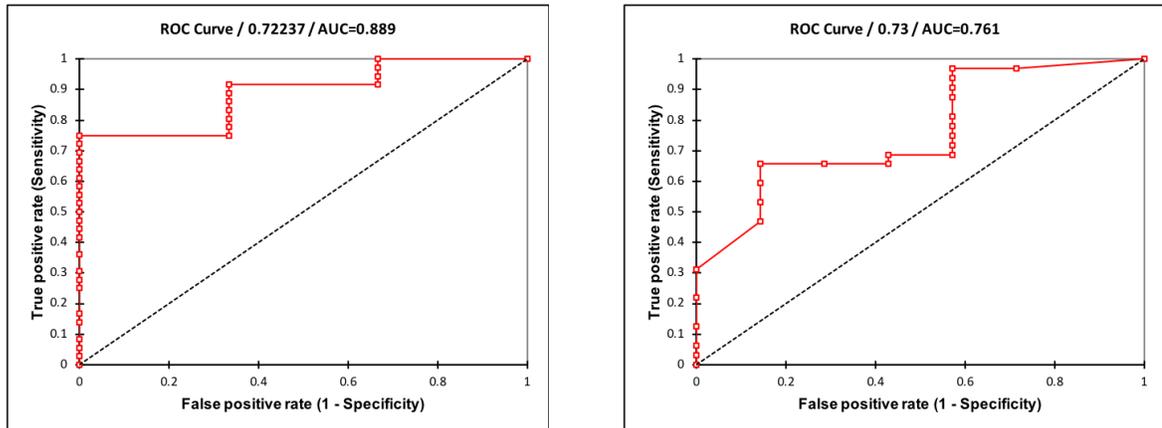
Autismo Definitivo (DA, *Definitive Autism*) (**0.51=DA**). Vector de entrada equivalente para MCFM-ASD,  $ADO^1 = \{0.430, 0.620, 0.486, 0, 0, 0, 0, 0, 0, 0\}$ . Ejecutando el MCFM-ASD, da como resultado **OUT-MX=0.50**, Diagnóstico: Autismo (AUT).

*Case 2*: Vector de entrada dado por el experto  $V^2 = \{0.17, 0.3, 0.32, 0.43, 0.2, 0.1, 0.01, 0.32, 0.41, 0.13, 0.15, 0.44, 0.28, 0.5, 0.64, 0.15, 0.25, 0.3, 0.29, 0.27, 0.31, 0.4, 0.42, \mathbf{0.41}\}$ . Diagnóstico: Probable Autismo (PA, *Probable Autism*) (**0.41=PA**). Vector de entrada equivalente para MCFM-ASD,  $ADO^1 = \{0.277, 0.339, 0.260, 0, 0, 0, 0, 0, 0, 0\}$ . Ejecutando el MCFM-ASD, da como resultado **OUT-MX=0.28**, Diagnóstico: Autismo (ASD).

*Case 3*: Vector de entrada dado por el experto  $V^3 = \{0.56, 0.72, 0.53, 0.64, 0.75, 0.66, 0.87, 0.76, 0.95, 0.45, 0.76, 0.52, 0.73, 0.44, 0.75, 0.67, 0.57, 0.48, 0.49, 0.4, 0.41, 0.42, 0.43, \mathbf{0.87}\}$ . Diagnóstico: Probable Autismo (**0.87=DA**). Vector de entrada equivalente para MCFM-ASD,  $ADO^1 = \{0.625, 0.626, 0.573, 0, 0, 0, 0, 0, 0, 0\}$ . Ejecutando el MCFM-ASD, se tiene como resultado **OUT-MX=0.60**, Diagnóstico: Autismo (AUT).

Es de notar que nuestro modelo MCFM sigue muy bien los tres diagnósticos dados por los expertos y el modelo propuesto en [48]. También, se probó con el mismo conjunto de datos usado en [48], utilizando 40 casos de diagnóstico registrados. En (Kannappan, Tamilarasi y Papageorgiou 2011), 20 de los 23 casos fueron diagnosticados como con autismo definitivo (DA), 10 de los 13 como autismo probable (PA) y 3/4 como no autista (NA). Ahora, con MFCM se reconocieron 23 de 23, 11 de 13 y 3/4, con un índice de precisión del 92.5%, un resultado mucho mejor en comparación con el 82.5% de precisión alcanzado por el FCM utilizado en (Kannappan, Tamilarasi y Papageorgiou 2011). Además, se trazan las curvas ROC (por sus siglas en inglés de *Receiver*

*Operating Characteristic*) para estos dos modelos, a fin de analizar su sensibilidad y especificidad en el proceso de diagnóstico de ASD (ver Figura. 5.12).



**Figura 5.12.** La curva de la izquierda es del MCFM-ASD y la curva de la derecha es del modelo. (Kannappan, Tamilarasi y Papageorgiou 2011)

En general, se requieren métodos de diagnóstico con alta sensibilidad y especificidad. En la curva ROC podemos calcular un área bajo la curva, llamada AUC (por sus siglas en inglés), con un valor que va de 0 a 1; el valor ideal está cerca del punto (0, 1), vértice superior izquierdo, que al mismo tiempo representa buena sensibilidad y especificidad. Esto significa que cuanto más cerca este de 1, mejor es el método de diagnóstico, lo que representa un método de diagnóstico con un discernimiento claro. En la Figura 5.12, el valor AUC alcanzado por el modelo MCFM es de 0.889, lo que indica que está cerca del borde izquierdo y superior de la curva ROC, lo que da resultados muy precisos. Por otro lado, el AUC alcanzado por (Kannappan, Tamilarasi y Papageorgiou 2011) es de 0.761, lo que indica resultados menos precisos que nuestro modelo.

### 5.5.2 Comparación Cualitativa

En la reunión internacional de 2008 sobre la investigación del autismo (IMFAR por sus siglas en inglés) en Londres, ADIR y ADOS se definieron como el estándar de oro para la investigación del autismo. ADOS es un instrumento de observación que puede aplicarse desde los 18 meses, permitiendo de esta manera un diagnóstico temprano y conocer el grado de severidad de autismo. Estudios han demostrado que ADOS tiene una sensibilidad de 0.9 (i.e., su capacidad para detectar el autismo en individuos con autismo), una especificidad de 0.8 (i.e., su capacidad para detectar la ausencia de autismo en individuos que están fuera del espectro autista), entre otras cualidades, como una excelente confiabilidad inter-rater<sup>20</sup> (0.65–0.78) y una buena confiabilidad test-retest<sup>21</sup> (o repetibilidad) (0.59–0.82) (McClintock y Fraser 2011). Por otra parte, ADIR es una entrevista dirigida a los padres o cuidadores, con el fin de diagnosticar el autismo, esta puede ser aplicada a partir de los 2 años de edad, y evalúa 4 dominios: lo social, comportamientos repetitivos y restringidos, aspectos verbales y de comunicación. ADIR tiene una sensibilidad entre 0.86 y 1.0, y una especificidad entre 0.75 y 0.96, entre otras medidas de rendimiento (McClintock y Fraser 2011).

CHAT es un cuestionario que contiene las respuestas de los padres y las observaciones de los niños evaluados rápidamente, que se puede aplicar en niños a partir de los 18 meses de edad. Sobre la base de las respuestas de los padres en el CHAT, el médico sigue los cuadros de flujo de evaluación subsecuentes para llegar a una decisión sobre el diagnóstico. Esta decisión puede ser imprecisa e intuitiva, dependiendo de la percepción y experiencia del médico. Estos

---

<sup>20</sup> Es el grado de acuerdo entre los evaluadores. Es un puntaje de cuánta homogeneidad, o consenso, hay en las calificaciones otorgadas.

<sup>21</sup> Bajo esta denominación se indica que las puntuaciones repetidas X y X' se han obtenido aplicando el mismo test a los mismos individuos y en las mismas condiciones, en dos ocasiones diferentes en el tiempo.

procedimientos también pueden llevar mucho tiempo, con un alto grado de pérdida de información en la evaluación, debido a su dependencia a entradas exactas. Este tiene una especificidad de 0.97, una sensibilidad de 0.18 y un valor predictivo de 0.58 (Johnson y Myers 2007). En el 2001, este instrumento se modificó a través de un programa de screening, tomando el nombre de MCHAT, un instrumento con el cual se diagnostica exclusivamente a través de las respuestas de los padres y cuidadores. Su sensibilidad es de 0.87, especificidad de 0.99, y capacidad de predicción de 0.80 (Johnson y Myers 2007). A este instrumento se le ha hecho una modificación adicional, llamada M-CHAT-R/F, que permite una mejor detección y reduce la tasa de falsos positivos.

Es de notar que, si bien, MCHAT tiene una alta sensibilidad y especificidad, no cuenta con una buena consistencia interna o un valor predictivo como para ubicarse en el mismo nivel de diagnóstico que ADIR y ADOS, ya que MCHAT no tiene en cuenta la complejidad del proceso de diagnóstico. Además, solo tiene en cuenta la impresión de los padres o cuidadores, cuando se ha comprobado en la práctica científica que con frecuencia los padres, en la desesperación de un diagnóstico, no tienen objetividad en sus respuestas (Ojeda 2015).

Nuestro modelo de diagnóstico integra ADOS2 y ADIR, lo que permite una mayor comprensión de la estructura de ASD. El uso de ADOS2 y ADIR hace que la herramienta sea más robusta y completa con respecto a trabajos previos basados en otros estándares (por ejemplo, MCHAT, el cual la comunidad psicosocial lo considera no tan bueno). Además, nuestro modelo MFCM-ASD permite el uso de ADIR y ADOS2 de forma separada, lo cual se logra simplemente desactivando los conceptos que no aplican durante el diagnóstico, es decir, nuestro modelo puede considerar diferentes situaciones de aplicación de ambos instrumentos. Zander (Zander, Sturm y Bölte 2015) han demostrado la utilidad de considerar el uso de estos instrumentos por separado. Además, el modelo permite quitar o agregar nuevas capas, siendo posible considerar otros aspectos del

autismo durante el diagnóstico, como la situación social, las propiedades neurofisiológicas, entre otros.

### 5.5.3 Comparación con otros algoritmos de aprendizaje de máquina

En esta sección, hemos utilizado el dataset de NSCH (por sus siglas en inglés de *National Survey of Children's Health*) para comparar nuestro modelo MFCM-ASD con los algoritmos clásicos de aprendizaje automático para las tareas de clasificación (Akshoomoff et al 2004). Más específicamente, hemos utilizado tres de los más populares algoritmos (van den Bekerom 2018): *Naive Bayes*, *Random Forest* y *Support Vector Machine*. NSCH tiene 95577 archivos de niños con 367 variables, donde se recopila información sobre la prevalencia de autismo y trastornos del espectro autista. Debido a que sólo un pequeño porcentaje de la dataset de NSCH representa a niños con autismo, se ha seleccionado una muestra aleatoria con aproximadamente el 50% de los niños con autismo y el 50% de los niños sin autismo. Se calcularon las métricas de rendimiento de *F-measure* y *Accuracy*, para comparar el modelo MCFM-ASD con estos algoritmos de aprendizaje automático. También se realizó una equivalencia entre las 367 variables del dataset NSCH y los conceptos de MCFM-ASD, con el fin de incorporar estas al modelo. En la prueba, se utilizó la técnica de validación cruzada con  $k=10$  (iteraciones), de manera que el 90% de las muestras del dataset se usaron para el entrenamiento. Se probaron 2 clases (sin ASD o ASD) con el dataset. Los resultados se muestran en la Tabla 5.16.

**Tabla 5.16.** Comparación con otros algoritmos de aprendizaje de máquina

	F-MEASURE	ACCURACY
MCFM-ASD	0.843	0.842
SVM	0.833	0.833
RANDOM FOREST	0.852	0.851
NB	0.865	0.865

Estos resultados muestran que nuestro modelo MCFM-ASD puede predecir el ASD para esta dataset, con un 83% de efectividad. El principal problema se encuentra en la definición de las

equivalencias del conjunto de variables/atributos con los conceptos de nuestro modelo, puesto que ciertos atributos pueden mapearse con diferentes conceptos del mismo. El peso de ciertos atributos en la dataset para el diagnóstico de ASD también puede ser explotado. Por ejemplo, ¿cuál es la importancia de los retrasos en el desarrollo, las discapacidades de aprendizaje, u otros problemas, para el autismo?. Este tipo de información puede ser considerada en nuestro modelo, cuando la equivalencia conceptual sea establecida. Estudios futuros pueden realizarse fácilmente con psicólogos, para analizar tales aspectos con nuestro modelo.

Nuestro objetivo con estas últimas pruebas ha sido determinar la calidad de nuestro modelo para predecir el ASD. El rendimiento alcanzado es muy cercano con respecto a los algoritmos de aprendizaje de máquina tratados. Además, nuestro modelo MFCM-ASD tiene la virtud de que, la lectura y/o la interpretación de los resultados y seguimiento del funcionamiento por parte del experto (e.g., psicólogo), para el proceso de diagnóstico, es muy fácil e intuitivo. Esta propiedad en parte compensa en gran medida su tenue diferencia de precisión, con respecto a las otras técnicas, cuyos resultados son buenos, pero no ayudan mucho a contextualizar los mismos, lo cual es muy importante para los psicólogos en sus procesos de diagnóstico.

## **5.6 Conclusiones y Trabajos Futuros**

El enfoque basado en el conocimiento utilizado en este trabajo se centra en el MFCM para el proceso de predicción de ASD. Este es el primer trabajo que propone MFCM-ASD para apoyar la identificación y clasificación de ASD. Nuestro enfoque observa el fenómeno autista utilizando dos niveles de conocimiento, definidos por ADIR y ADOS2. La utilización de múltiples capas hace que nuestro enfoque sea más robusto porque en cada nivel, podemos introducir diferentes aspectos a considerar para el diagnóstico. Específicamente, en nuestro caso, hemos integrado muy fácilmente ADOS y ADIR. Además, en nuestro modelo se pueden definir nuevas capas para

considerar neuro-imágenes, entre otras cosas. Esta extensión permitirá comparar nuestro enfoque con estudios anteriores que han utilizado la dataset ABIDE<sup>22</sup>.

MFCM-ASD lleva a cabo un proceso de inferencia difuso basado en un conjunto de reglas difusas de tipo IF-THEN, a través de las cuales representa y describe de forma natural el conocimiento detrás de los instrumentos (ADIR/ADOS2) para el diagnóstico del autismo, facilitando su utilización e interpretación a los psicólogos; un aspecto importante para darle usabilidad a esta herramienta.

Las pruebas demuestran que MFCM-ASD logra resultados de diagnósticos del autismo, que se corresponden muy bien con los resultados dados por los expertos, en los dataset de casos de diagnosticados utilizados, aplicando los estándares ADOS y ADIR. Además, los resultados obtenidos por nuestro enfoque en el estándar MCHAT, muestran una mayor versatilidad que trabajos previos.

Nuestro modelo tiene implícitamente las propias limitaciones de los modelos de diagnóstico con base comportamental, i.e., basado en entrevistas u observación, puesto que éstos no pueden señalar ninguna base biológica detrás de los síntomas conductuales observados. Finalmente, se ha comparado la capacidad predictiva de nuestro modelo en diferentes contextos y dataset, obteniendo un rendimiento muy bueno (ver 5.5.1 y 5.5.3).

Como trabajos futuros está el extender el modelo MFCM-ASD a través de la introducción de nuevas capas que cubran más dimensiones para evaluar el Autismo, como por ejemplo, el contexto social del sujeto, las variables demográficas y otras variables cognitivas, características de neuro-imágenes, etc. Además, junto con psicólogos y/o expertos, estudiar la sensibilidad y especificidad con estas nuevas dimensiones.

---

<sup>22</sup> [http://fcon\\_1000.projects.nitrc.org/indi/abide](http://fcon_1000.projects.nitrc.org/indi/abide)

También, trabajos futuros deben considerar la utilización del paradigma de aprendizaje profundo (Deep Learning) en el contexto de nuestro estudio, ya sea para el aprendizaje de nuevos conceptos, descriptores, marcadores biológicos, relaciones, reglas de diagnóstico, etc. Esta nueva información (descriptores, reglas, etc.) debe ser previamente interpretada por expertos en autismo, para que se pueda utilizarse durante el proceso de diagnóstico. Finalmente, se debe trabajar en cubrir la debilidad de explicación en sus diagnósticos, tal vez usando un paradigma lógico y/o declarativo.

### **5.7. Consideraciones Éticas**

Este trabajo se ha llevado a cabo de conformidad con el código de ética de la Asociación Médica Mundial (Declaración de Helsinki) para experimentos con seres humanos. Los datos obtenidos y utilizados en este trabajo se tomaron con el consentimiento de los miembros (padres) de la Asociación de padres y amigos para apoyar y defender los derechos de las personas con autismo, APADA. APADA es una ONG sin fines de lucro, creada el 27 de marzo de 2013 a través del Acuerdo # 0080 del Ministerio de Inclusión Social y Economía, compuesto por padres y personas dentro de este espectro (<https://goo.gl/weXhZK>). APADA ha permitido el uso de datos bajo su política de confidencialidad, por esta razón, los nombres se han cambiado para proteger las identidades de los niños. <https://goo.gl/Mzoiju>. Agradecemos a la Dra. C. López y a la Dra. M. Sandoval, por su apoyo en los conjuntos de datos utilizados.

## 6. CAPITULO 6: CONCLUSIONES GENERALES Y TRABAJOS FUTUROS

Esta tesis desarrollo varios productos. Primero que nada, logró el desarrollo de un modelo computacional a un alto nivel funcional, del proceso neurofisiológico de reconocimiento de patrones en los humanos, particularmente importante en los individuos con capacidades cognitivas diferentes. Para tal fin, se especificó un método algorítmico recursivo jerárquico de reconocimiento de patrones, inspirado en la Teoría de la Mente basada en el reconocimiento de patrones (kurzweil 2013). Para ello, se formalizó matemáticamente el esquema jerárquico de reconocimiento de patrones, en un modelo llamado Ar2p; y en particular, los módulos de reconociendo de patrones estáticos, los módulos de reconociendo de patrones dinámicos, y los teoremas usados para el reconocimiento: por señales claves y por señales parciales (Puerto y Aguilar 2017a; Puerto y Aguilar 2017b); para reconocer tanto a patrones estáticos como dinámicos (Puerto, Aguilar, y Chávez 2018). Además, la tesis definió varios mecanismos de aprendizaje para Ar2p: *Aprendizaje nuevo*, *Aprendizaje por refuerzo* y *Aprendizaje profundo*, como mecanismos naturales de adaptación del modelo (Puerto y Aguilar 2016b).

Un inconveniente del modelo propuesto a nivel computacional, es que si bien ofrece una solución concisa, legible y elegante (recursividad) al problema de reconocimiento, el define implícitamente un árbol, que puede ser muy grande. En específico, la actual versión de reconocimiento es secuencial, con llamadas recursivas para reconocer y/o aprender patrones, requiriendo definir el nivel atómico para cada patrón. Una posible optimización es usar el paralelismo para mejorar el tiempo de recorrido en el árbol. También, la versión actual de Ar2p no incorpora mecanismos de eliminación de los módulos de reconocimiento que han caído en desuso, por lo tanto, si bien tiene mecanismo de aprendizaje que le permiten aprender, no cuenta con mecanismo que le permitan

desaprender, lo que conlleva a subutilizar el espacio de memoria. Eso es otra importante mejora, para optimizar el árbol de Ar2p. Por otro lado, Ar2p requiere una definición inicial supervisada de las variables de los módulos utilizados para el reconocimiento, en específico, el peso de importancia y el valor de los umbrales.

A nivel de pruebas, el modelo logró buenos resultados en el reconocimiento de patrones en diversos dominios. Por ejemplo, a nivel de la capacidad de razonamiento, comunicación y aprendizaje, para el reconocimiento de patrones de estilos de conducción, Ar2p mostró ser más eficiente versus los otros modelos expuestos (Aguilar, J., Aguilar, K., Chávez, Cordero y Puerto 2017; Aguilar, J., Aguilar, K., Chávez, Cordero y Puerto 2019). Como trabajo futuro para este estudio, se debe probar/implementar el comportamiento del modelo en entornos reales conectado a ADAS, con fuertes limitaciones de tiempo y grandes cantidades de flujo de datos. Para tales pruebas, el vehículo deberá estar equipado con sistemas como cámara, sensor de temperatura, micrófono, GPS, entre otros, que permitirán percibir de manera multimodal los descriptores que componen los patrones de conducción, para capturar tanta información como sea posible sobre el conductor.

En otro experimento se logró evaluar las capacidades de reconocimiento de patrones de Ar2p, esta vez en el contexto de la robótica de enjambre (Gil, Aguilar, Puerto y Dapena, 2018, Gil, Aguilar, Puerto y Dapena 2019). Para este caso se propuso dos escenarios. Un primer escenario para informar el lugar para recargar las baterías de los robots que necesitan recargar (robots tristes); y el otro para el reclutamiento de robots para mover un objeto (robots alegres). A través de Ar2p se probó las interacciones entre los robots en función de la emoción reconocida/percibida, y la influencia en el comportamiento colectivo del sistema. Se usó Ar2p para que cada robot reconozca los estados emocionales de los robots en su entorno (es una habilidad de los robots) y, de acuerdo

con su percepción sobre el estado emocional de los mismos, actuar (recluta, informa, etc.). Como trabajo futuro para este caso, se propuso extender este modelo considerando más emociones, más situaciones (escenarios), etc., para probar la escalabilidad de nuestro enfoque, y en particular, estudiar su influencia en el comportamiento colectivo de un sistema (Gil, Aguilar, Rivas, Dapena y Hernandez 2018).

Otro logro fue, el diseño y construcción de una arquitectura de *aprendizaje profundo*, llamada Ar2p-DL, para el descubrimiento de patrones atómicos. Esta es una habilidad de los individuos autistas, en su proceso de reconocimiento de patrones. En particular, patrones que no son fácilmente reconocibles por parte una persona común y corriente, son fácilmente reconocibles por los niños autistas, derivado de las habilidades para determinar diferencias sutiles en patrones: frecuencias en notas musicales, órdenes de aparición en secuencia de números, etc. Esta arquitectura usa una parte de Ar2p, e incorpora mecanismos no supervisados para el descubrimiento y selección de características. Ar2p-DL alcanzó muy buenos resultados en los problemas de selección o descubrimiento de características, tanto para datos balanceados como no balanceados. Específicamente, Ar2p-DL descubre o selecciona automáticamente las características, utilizando varios métodos de la ingeniería de características (Puerto, Aguilar, Reyes, y Sarkar 2018). Como trabajo futuro, se propone analizar otras técnicas no supervisadas en las primeras fases de Ar2p-DL, para minimizar el tiempo de ejecución, que es muy importante en los problemas de reconocimiento en tiempo real. Adicional a esto, analizar la sensibilidad de cada fase en los resultados finales, y probar con otros *benchmarks*.

Un último logro derivado de los objetivos de la tesis, fue la construcción de un modelo de diagnóstico del espectro autista, denominado MFCM-ASD (Puerto, Aguilar, López y Chávez 2019). Este modelo se basó en la tecnología de mapas cognitivos difusos multicapas. MFCM-ASD

se creó como respuesta a la necesidad de caracterizar desde una mirada externa al cerebro, i.e., desde una perspectiva más cualitativa, el fenómeno autista. Este modelo representó el comportamiento del fenómeno autista utilizando dos niveles de conocimiento, basado en los estándares internacionales ADIR y ADOS2. Los dos niveles de conocimiento están vinculados a información de los padres y del comportamiento del individuo evaluado. Ahora bien, con MFCM-ASD es posible introducir otros aspectos a considerar para el diagnóstico (por ejemplo, usar neuro-imágenes). Una cualidad más de este modelo es su facilidad de utilización e interpretación para los psicólogos; un aspecto importante para darle usabilidad a esta herramienta. Como trabajos futuros se tienen: su extensión como un sistemas de diagnóstico online para el servicio de la comunidad autista, y extender el modelo MFCM-ASD a través de la introducción de nuevas capas que cubran más dimensiones para evaluar el Autismo, como por ejemplo, el contexto social del sujeto, las variables demográficas, otras variables cognitivas, neuro-imágenes, etc. Además, junto con psicólogos y/o expertos, estudiar la sensibilidad y especificidad con estas nuevas dimensiones. También, trabajos futuros deben considerar la utilización del paradigma de aprendizaje profundo (Deep Learning) en el contexto de MFCM-ASD, ya sea para el aprendizaje de nuevos conceptos, descriptores, marcadores biológicos, relaciones, reglas de diagnóstico, etc. Esta nueva información (descriptores, reglas, etc.) debe ser previamente interpretada por expertos en autismo, para que se pueda utilizarse durante el proceso de diagnóstico. Finalmente, se debe trabajar en cubrir la debilidad de explicación en sus diagnósticos, tal vez usando un paradigma lógico y/o declarativo.

## Bibliografía

- Abbott, L. F., & Kepler, T. B. (1990). Model neurons: from hodgkin-huxley to hopfield. In *Statistical mechanics of neural networks* (pp. 5-18). Springer, Berlin, Heidelberg.
- Abreu, N. G. C. F. M. (2011). *Análise do perfil do cliente Recheio e desenvolvimento de um sistema promocional* (Disertación Doctoral). ISCTE-IUL, Lisbon.
- Agirre, E., et al. (Marzo 1992). XUXEN: A spelling checker/corrector for Basque based on Two-Level morphology. In *Proc. of the third conference on Applied natural language processing. Association for Computational Linguistics*. 119-125.
- Aguilar, J. (2004). A color pattern recognition problem based on the multiple classes random neural network model. *Neurocomputing*, 61, 71-83.
- Aguilar, J y Hernández, L (2009). Diseño e Implementación de un Sistema de Reconocimiento de Patrones para Análisis de Líquidos Biológicos. *IEEE Latinoamerica Transactions*, 7(1), 12-26.
- Aguilar, J., Aguilar, K., Chávez, D., Cordero, J. y Puerto, E. Recognition of the Driving Style in Vehicle Drivers. (En Revisión).
- Aguilar, J., Chavez, D., y Cordero, J., (2016). A Dynamic Recognition Approach of Emotional States for Car Drivers. In *Proc. Technologies and Innovation: Second International Conference, CITI, Springer International Publishing, Cham*, 155–168.
- Aguilar, J., (1998). Definition of an energy function for the random neural to solve optimization problems. *Neural Networks*, 11(4), 731-737.
- Aguilar, J. (Junio 2001). A fuzzy cognitive map based on the random neural model. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 333-338. Springer, Berlin, Heidelberg.
- Aguilar, J. (2001). Learning algorithm and retrieval process for the multiple classes random neural network model. *Neural Processing Letters*, 13(1), 81-91.
- Aguilar, J. (2004). Dynamic Random Fuzzy Cognitive Maps. *Computacion y Sistemas*, 7(4) 260–270.
- Aguilar, J. (2005). A survey about fuzzy cognitive maps papers. *International journal of computational cognition*, 3(2), 27-33.
- Aguilar, J. (2005). The random neural network model on the color pattern recognition problem. *Revista Técnica de la Facultad de Ingeniería Universidad del Zulia*, 28(1), 51-58.
- Aguilar, J. (2009). *Introducción a las Redes Neuronales Artificiales*. Mérida, Venezuela: MERITEC.
- Aguilar, J. (2013). Different dynamic causal relationship approaches for cognitive maps. *Applied Soft Computing*, 13(1), 271-282.
- Aguilar, J. (2014). *Introducción a los Sistemas Emergentes*. Talleres Gráficos. Mérida, Venezuela MERITEC.
- Aguilar, J. (2016). Multilayer Cognitive Maps in the Resolution of Problems using the FCM Designer Tool. *Applied Artificial Intelligence*, 30(7), 720-743.

- Aguilar, J., y Leiss, E. (2004). *Introducción a la computación paralela*. Mérida, Venezuela: MERITEC.
- Aguilar, J., y Rivas, F. (2001). *Computación Inteligente*. Mérida, Venezuela: MERITEC.
- Aguilar, J., Aguilar, K., Chávez, D., Cordero, J. y Puerto, E., (2017). Different Intelligent Approaches for Modeling the Style of Car Driving. In *Proc. of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2, 284-291.
- Aguilar, J., Colmenares, A. (1997). Recognition algorithm using evolutionary learning on the random neural networks, In *Proc. of the IEEE International Conference on Neural Networks*, 1023-1028.
- Aguilar, J., Contrerars, J. (2010). The FCM designer tool, *Stud. Fuzziness Soft Comput.* 247, 71–87.
- Aguilar, J., Hidalgo, J., Osuna, F., y Pérez, N. (Julio 2016). Multilayer cognitive maps to model problems. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1547-1553. IEEE.
- Akshoomoff, N., Lord, C., Lincoln, A. J., Courchesne, R. Y., Carper, R. A., Townsend, J., y Courchesne, E. (2004). Outcome classification of preschool children with autism spectrum disorders using MRI brain measures. *Journal of the American Academy of Child y Adolescent Psychiatry*, 43(3), 349-357.
- Al Farsi, A., Doctor, F., Petrovic, D., Chandran, S., y Karyotis, C. (Julio de 2017). Interval valued data enhanced fuzzy cognitive maps: Towards an approach for Autism deduction in Toddlers. In *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on*, 1-6. IEEE.
- Alexander, J. R. M., y Smales, S. (1997). Intelligence, learning and long-term memory. *Personality and Individual Differences*, 23(5), 815-825.
- Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.
- Altamiranda, J., Aguilar, J., y Hernandez, L. (2015). Sistema de reconocimiento de patrones de sustancias químicas cerebrales basado en minería de datos. *Computación y Sistemas*, 19(1), 89-107.
- Álvarez, M. V. (2016). VirtuaCyL: desarrollo y validación de un sistema ubicuo basado en Android para refuerzo educativo de niños con autismo dentro de la metodología TEACCH (Tesis de Maestría), Universidad Politécnica de Valencia, Valencia, España.
- American Psychiatric Association. (2013). *Diagnostic and Statistical Manual of Mental Disorders*. (5th ed.). Washington, DC: American Psychiatric Association Press.
- American Psychiatric Association. (2017). *DSM-5 Update*. *DSM-5 Update*. Supplement to *Diagnostic and Statistical Manual of Mental Disorders*.
- Amirkhani, A., Papageorgiou, E. I., Mohseni, A., y Mosavi, M. R. (2017). A review of fuzzy cognitive maps in medicine: Taxonomy, methods, and applications. *Computer methods and programs in biomedicine*, 142, 129-145.
- Anderson, M. L. (2010). Neural reuse: A fundamental organizational principle of the brain. *Behavioral and brain sciences*, 33(4), 245-266.

- Anirudh, R., y Thiagarajan, J. J. (2017). Bootstrapping graph convolutional neural networks for autism spectrum disorder classification. *arXiv preprint arXiv:1704.07487*.
- Application, A., Purpose, E., Health, C.M., Inpatient, P., Plan, H., Services, H. State of Michigan Department of Health and Human Services Autism Application: ADOS-2 Evaluation State of Michigan Department of Health and Human Services Autism Application: *ADOS-2 Evaluation*, 7–10.
- Arthi, K., y Tamilarasi, A. (2008). Prediction of autistic disorder using neuro fuzzy system by applying ANN technique. *International journal of developmental neuroscience*, 26(7), 699-704.
- Asuncion, A., y Newman, D. (2007). UCI machine learning repository.
- Aypar, Y. C., Palaska, Y., Gokay, R., Masazade, E., Barkana, D. E., y Sarkar, N. (Septiembre 2014). Clustering of emotional states under different task difficulty levels for the robot-assisted rehabilitation system-RehabRoby. In *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 1, 34-41). IEEE.
- Babic, B., Nestic, N., y Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. *Computers in industry*, 59(4), 321-337.
- Balakrishnama, S., y Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18,1-8.
- Banik, S. C., Watanabe, K., y Izumi, K. (Septiembre 2007). Task allocation with a cooperative plan for an emotionally intelligent system of multi-robots. In *SICE, 2007 Annual Conference*. IEEE, 1004-1010.
- Banik, S. C., Watanabe, K., Habib, M. K., y Izumi, K. (Agosto 2008). An emotion-based task sharing approach for a cooperative multiagent robotic system. In *Mechatronics and Automation*, 2008. ICMA 2008. IEEE International Conference on. IEEE,77-82.
- Barkana, D. E., y Masazade, E. (2017). Classification of the Emotional State of a Subject Using Machine Learning Algorithms for RehabRoby. In *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2160-2187.
- Baron-Cohen S, et al. (2002). Development of a new screening instrument for autism spectrum disorders - the Q-CHAT. In *International Meeting for Autism Research*. Orlando, FL.
- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1), 1-127.
- Bi, X. A., Liu, Y., Jiang, Q., Shu, Q., Sun, Q., y Dai, J. (2018). The diagnosis of autism spectrum disorder based on the random neural network cluster. *Frontiers in human neuroscience*, 12, 257.
- Block, H. D. (1962). The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, 34(1),123.
- Boddaert, N., Barthélémy, C., Poline, J. B., Samson, Y., Brunelle, F., y Zilbovicius, M. (2005). Autism: Functional brain mapping of exceptional calendar capacity. *British Journal of Psychiatry*, 187(1), 83–86.
- Boggess, A., y Narcowich, F. J. (Agosto 2015). *A first course in wavelets with Fourier analysis*.

John Wiley y Sons.

- Bone, D., Bishop, S. L., Black, M. P., Goodwin, M. S., Lord, C., y Narayanan, S. S. (2016). Use of machine learning to improve autism screening and diagnostic instruments: effectiveness, efficiency, and multi-instrument fusion. *Journal of Child Psychology and Psychiatry and Allied Disciplines*, 57(8), 927–937.
- Bone, D., Goodwin, M. S., Black, M. P., Lee, C. C., Audhkhasi, K., y Narayanan, S. (2015). Applying Machine Learning to Facilitate Autism Diagnostics: Pitfalls and Promises. *Journal of Autism and Developmental Disorders*, 45(5), 1121–1136.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Brooks, T. N., Kamruzzaman, A., Leider, A., y Tappert, C. C. (Septiembre 2018). A Computer Science Perspective on Models of the Mind. In *Proc. of SAI Intelligent Systems Conference*, 759-772). Springer, Cham.
- Brownlee, J. (Septiembre 2014). Discover feature engineering, how to engineer features and how to get good at it. *Machine Learning Process*.
- Cao, Y. U., Fukunaga, A. S., y Kahng, A. (1997). Cooperative mobile robotics: *Antecedents and directions*. *Autonomous robots*, 4(1), 7-27.
- Caporale, N., y Dan, Y. (2008). Spike timing–dependent plasticity: a Hebbian learning rule. *Annu. Rev. Neurosci.*, 31, 25-46.
- Carandini, M., y Heeger, D. J. (2012). Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1), 51.
- Carpenter, R. H., y Amato Martínez, J. D. (1998). *Neurofisiología*.
- Carvalho, L. A. V., de Carvalho Ferreira, N., y Fiszman, A. (Julio de 1999). A neurocomputational model for autism. In *Proc. IV Brazilian conference on neural networks-IV*, 888-999.
- Castel, B. (2015). Pattern activation/recognition theory of mind. *Frontiers in Computational Neuroscience*, 9, 90.
- Cattell, R., y Parker, A. (2012). Challenges for Brain Emulation: Why is it so Difficult? *Natural Intelligence*, 1, 17–31.
- Chang, M. C., Bus, P., y Schmitt, G. (2017, December). Feature Extraction and K-means Clustering Approach to Explore Important Features of Urban Identity. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, 1139-1144. IEEE.
- Charman, T., y Gotham, K. (2013). Measurement Issues: Screening and diagnostic instruments for autism spectrum disorders—lessons from research and practise. *Child and adolescent mental health*, 18(1), 52-63.
- Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., y Żak, S. (2010). Complete gradient clustering algorithm for features analysis of x-ray images. In *Information technologies in biomedicine*, 15-24. Springer, Berlin, Heidelberg.
- Chen, K. (2005). On the Dynamic Pattern Analysis, Discovery and Recognition. *IEEE SMC Society eNewsletter*, no. 12.
- Chéron, G. (2018). Neurofisiología del movimiento. Oscilaciones neuronales y aprendizaje motor.

- EMC-Kinesiterapia-Medicina Física*, 39(2), 1-11.
- Christensen, O., y Christensen, K. L. (2004). *Approximation theory: from Taylor polynomials to wavelets*. Springer Science y Business Media.
- Christophe, F., Mikkonen, T., Andalibi, V., Koskimies, K., y Laukkarinen, T. (2015). Pattern recognition with Spiking Neural Networks: A simple training method. In *Proc. CEUR Workshop*. 279-288.
- Churchland, P. S., Sejnowski, T. J., y Poggio, T. A. (2016). The computational brain. MIT press. *Cognition. Physiol. Rev.*, 90, 1195-1268.
- Cohen, I. L., Sudhalter, V., Landon-Jimenez, D., y Keogh, M. (1993). A neural network approach to the classification of autism. *Journal of autism and developmental disorders*, 23(3), 443-466.
- Collins, A., y Bobrow, D. G. (Eds.). (2017). *Representation and understanding: Studies in cognitive science*. Elsevier.
- Cordero, J., y Aguilar, J. (2016). *Reconocimiento multimodal de emociones en un entorno inteligente basado en crónicas*. Gráficas El Portatítulo, 525-541.
- Cowan, R., O'Connor, N., y Samella, K. (2001). Why and how people of limited intelligence become calendrical calculators. *Infancia y Aprendizaje*, 24(1), 53–65.
- Coyer, T. (2014). *Learning in patter recognition*. Methodologies of Pattern Recognition, 111.
- Crippa, A., Salvatore, C., Perego, P., Forti, S., Nobile, M., Molteni, M., y Castiglioni, I. (2015). Use of machine learning to identify children with autism and their motor abnormalities. *Journal of autism and developmental disorders*, 45(7), 2146-2156.
- da Cunha, I., Montané, M. A., y Hysa, L. (2017). The arText prototype: An automatic system for writing specialized texts. In *Proc. of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics EACL 2017*, 57-60.
- Daubechies, I. (1992). *Ten lectures on wavelets 6l*. Siam.
- Dayhoff, J. E. (Enero 1990). *Neural network architectures: an introduction*. Van Nostrand Reinhold Co.
- de Carvalho, L. A. V., de Carvalho Ferreira, N., y Fiszman, A. (1999, Julio). A neurocomputational model for autism. In *Proc. of IV Brazilian conference on neural networks-IV*, 344-349. ITA, San José dos Campos, Brasil.
- De Garis, H., Shuo, C., Goertzel, B., y Ruiting, L. (2010). A world survey of artificial brain projects, Part I: Large-scale brain simulations. *Neurocomputing*, 74(1-3), 3-29.
- Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), 141-142.
- Deng, L., y Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7, 197-387.
- Diagnostic Interview–Revised and the Autism Diagnostic Observation Schedule: Diagnostic validity in a clinical Swedish sample of toddlers and young preschoolers. *Autism*, 19(2), 187-199.

- Diamond, M. C., Scheibel, A. B., Murphy Jr, G. M., y Harvey, T. (1985). On the brain of a scientist: Albert Einstein. *Experimental neurology*, 88(1), 198-204.
- DigitCult-Scientific Journal on Digital Cultures, 2(2), 37-50.
- Ding, Z., Yang, C., Tian, Z., Yi, C., Fu, Y., y Jiang, F. (2018). sEMG-Based Gesture Recognition with Convolution Neural Networks. *Sustainability*, 10(6), 1865.
- Dong, W., Li, J., Yao, R., Li, C., Yuan, T., y Wang, L. (Julio, 2016). Characterizing driving styles with deep learning. *arXiv preprint arXiv:1607.03611*.
- Dörr, D., Grabengiesser, D., y Gauterin, F. (Octubre, 2014). Online driving style recognition using fuzzy logic. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1021-1026. IEEE.
- Du Castel, B. (2015). Pattern activation/recognition theory of mind. *Frontiers in computational neuroscience*, 9, 90.
- Duch, W., Nowak, W., Meller, J., Osiński, G., Dobosz, K., Mikołajewski, D., y Wójcik, G. M. (2012). Computational approach to understanding autism spectrum disorders. *Computer Science*, 13, 47-61
- Ecker, C. (2017). The neuroanatomy of autism spectrum disorder: An overview of structural neuroimaging findings and their translatability to the clinical setting, *Autism* 21(1), 18–28.
- El Kessab, B., Daoui, C., Boukhalene, B., y Salouan, R. (2014). A Comparative Study between the K-Nearest Neighbors and the Multi-Layer Perceptron for Cursive Handwritten Arabic Numerals Recognition. *International Journal of Computer Applications*, 107(21).
- Ester, M., Kriegel, H. P., Sander, J., y Xu, X. (Agosto, 1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, 96 (34), 226-231.
- Eyben, F., Wöllmer, M., Poitschke, T., Schuller, B., Blaschke, C., Färber, B., Nguyen-Thien, N.,(2010). Emotion on the road—necessity, acceptance, and feasibility of affective computing in the car. *Advances in human-computer interaction*, 2010.
- Fakhoury, M. (2015). Autistic spectrum disorders: A review of clinical features, theories and diagnosis. *International Journal of Developmental Neuroscience*, 43, 70-77.
- Fang, B., Chen, L., Wang, H., Dai, S., y Zhong, Q. (2014). Research on multirobot pursuit task allocation algorithm based on emotional cooperation factor. *The Scientific World Journal*, 2014, p. 6.
- Faust, O., Hagiwara, Y., Hong, T. J., Lih, O. S., y Acharya, U. R. (Abril 2018). Deep learning for healthcare applications based on physiological signals (review). *Computer methods and programs in biomedicine*.
- Felzenszwalb, P., McAllester, D., y Ramanan, D. (Junio 2008). A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008*. IEEE Conference on. IEEE, 1-8.
- Fernández, M., Mollinedo-Gajate, I., y Peñagarikano, O. (2018). Neural Circuits for Social Cognition: Implications for Autism. *Neuroscience*. 370, 148-162
- Ferreira Cabrera, A., y Hernández Osuna, S. (2017). Diseño e implementación de un corrector ortográfico dinámico para el sistema tutorial inteligente, ELE-TUTORA. *Revista signos*,

- 50(95), 385-407.
- Feygin, Y. B., Morris, K., y Yampolskiy, R. V. (2018). Uploading Brain into Computer: Whom to Upload First?. *arXiv preprint arXiv:1811.03009*.
- Filipek, P. A., Accardo, P. J., Ashwal, S., Baranek, G. T., Cook, E. H., Dawson, G., ... y Levy, S. Fischer, A., y Igel, C. (Septiembre 2012). An introduction to restricted Boltzmann machines. In *Iberoamerican Congress on Pattern Recognition*, 14-36. Springer, Berlin, Heidelberg.
- Fisher, N., y Happé, F. (2005). A training study of theory of mind and executive function in children with autistic spectrum disorders. *Journal of autism and developmental disorders*, 35(6), 757.
- Foley, J. M. (1988). *The Exceptional Brain: Neuropsychology of Talent and Special Abilities*. Neurology. Guilford Press.
- Fred, A., De Marsico, M. *Pattern Recognition Applications and Methods*. Springer, 2015.
- Froelich, W. (2017). Towards improving the efficiency of the fuzzy cognitive map classifier. *Neurocomputing*, 232, 83-93.
- Fukushima, K., Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position, *Pattern Recognition*, 15 (6) 455-469.
- Fukushima, K., Miyake, S., y Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (5), 826-834.
- Galitsky, B. (2013). A computational simulation tool for training autistic reasoning about mental attitudes. *Knowledge-Based Systems*, 50, 25-43.
- Galitsky, B. (2016). Computational Models of Autism. In *Computational Autism*, 17-77. Springer, Cham.
- Gamallo, P., y Garcia, M. (2017). LinguaKit: a multilingual tool for linguistic analysis and information extraction. *Linguamatica*, 9(1), 19-28.
- Garon, N., Smith, I. M., y Bryson, S. E. (2018). Early executive dysfunction in ASD: Simple versus complex skills. *Autism Research*, 11(2), 318-330.
- Gautam, A., y Mohan, S. (Agosto 2012). A review of research in multi-robot systems. In *Industrial and Information Systems (ICIIS), 2012 7th IEEE International Conference on*. IEEE, 1-5.
- Gelenbe, E., and Yin, Y. (Julio 2016). Deep learning with random neural networks. In *Proc. International Joint Conference on Neural Networks (IJCNN)*, 1633-1638. Vancouver, BC, Canada.
- George, D., y Hawkins, J. (Septiembre 2004). Invariant pattern recognition using Bayesian inference on hierarchical sequences. *Technical Report*.
- Ghosh-Dastidar, S., y Adeli, H. (2009). Spiking neural networks. *International journal of neural systems*, 19(04), 295-308.
- Gil, A., Aguilar, J., Puerto, E., y Dapena, E. Emergence analysis in a Multi-robot System (Octubre 2018). In *Proc. XLIV Conferencia Latinoamericana en Informática (CLEI 2018)*, Sao Pablo Brasil.
- Gil, A., Aguilar, J., Puerto, E., y Dapena, E., (2019). Emotions in a Multi-robot System in th

- Context of Emergent Behaviors. *CLEI Electronic Journal*.
- Gil, A., Aguilar, J., Rivas, R., Dapena, E., y Hernandez, K. (Enero 2015). Architecture for multi-robot systems with emergent behavior. In *Proc. on the International Conference on Artificial Intelligence (ICAI)*. 41-47.
- Glykas, M. (Ed.). (2010). *Fuzzy cognitive maps: Advances in theory, methodologies, tools and applications 247*. Springer Science y Business Media.
- Goodfellow, I., Bengio, Y., Courville, A., y Bengio, Y. (2016). *Deep learning 1*. Cambridge: MIT press.
- Gowen, E., y Hamilton, A. (2013). Motor abilities in autism: A review using a computational context. *Journal of Autism and Developmental Disorders*. 43(2), 323-344.
- Grandin, T. (2006). *Thinking in pictures: And other reports from my life with autism*. Vintage.
- Grandin, T. (2009). How does visual thinking work in the mind of a person with autism? A personal account. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 364(1522), 1437-1442.
- Grandin, T., y Panek, R. (2013). *The autistic brain: Thinking across the spectrum*. Houghton Mifflin Harcourt.
- Graps, A. (1995). An introduction to wavelets. *IEEE computational science and engineering*, 2(2), 50-61.
- Grassia, F., Levi, T., Doukkali, E., y Kohno, T. (2018). Spike pattern recognition using artificial neuron and spike-timing-dependent plasticity implemented on a multi-core embedded platform. *Artificial Life and Robotics*. 23(2), 200-204.
- Gray, K. M., Tonge, B. J., y Sweeney, D. J. (2008). Using the Autism Diagnostic Interview-Revised and the Autism Diagnostic Observation Schedule with young children with developmental delay: evaluating diagnostic validity. *Journal of autism and developmental disorders*, 38(4), 657-667.
- Groumpos, C.D., Georgopoulos, V.C., Malandraki, G.A., y Chouliara S. (2008). Fuzzy cognitivemap architectures for medical decision support systems, *Appl. Soft Comput.* 8(3) 1243–1251.
- Groumpos, P. P. (2010). Fuzzy cognitive maps: Basic theories and their application to complex systems. In *Fuzzy cognitive maps*, 1-22. Springer, Berlin, Heidelberg.
- Guoying, C., Danpan, W., (2016). Study on Identification of Driver Steering Behaviour Characteristics Based on Pattern Recognition. *International Robotics y Automation Journal*, 1 (5).
- Guyon, I., & Elisseeff, A. (2006). An introduction to feature extraction. In *Feature extraction*, 1-25. Springer, Berlin, Heidelberg.
- Haddadi, A., y Sundermeyer, K. (1996). Belief-desire-intention agent architectures. *Foundations of distributed artificial intelligence*, 169-185.
- Ham, F. M., y Kostanic, I. (2000). *Principles of neurocomputing for science and engineering*. McGraw-Hill Higher Education.
- Han, W., Wang, W., Li, X., y Xi, J. (2018). Statistical-based approach for driving style recognition

- using Bayesian probability with kernel density estimation. *IET Intelligent Transport Systems*, 13(1), 22-30.
- Happé, F., y Frith, U. (2006). The weak coherence account: detail-focused cognitive style in autism spectrum disorders. *Journal of autism and developmental disorders*, 36(1), 5-25.
- Hawkins, J., y Blakeslee, S. (2007). *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan.
- Hermelin, B., y O'CONNOR, N. (1967). Remembering of words by psychotic and subnormal children. *British Journal of Psychology*, 58(3-4), 213-218.
- Hernández, E., y Weiss, G. (1996). *A first course on wavelets*. CRC press.
- Hertz, J., Krogh, A., y Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Addison-Wesley/Addison Wesley Longman.
- Hill, E. L. (2004). Executive dysfunction in autism. *Trends in cognitive sciences*, 8(1), 26-32.
- Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10), 428-434.
- Hinton, G. E. (2009). Deep belief networks. *Scholarpedia*, 4(5), 5947.
- Hinton, G. E., Osindero, S., y Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- Hodgkin, A. L., y Huxley, A. F. (1952). A quantitative description of membrane current and application to conductance in excitation nerve. *J. Physiol.*, (London) 117, 500-544.
- Hornik, K., Stinchcombe, M., y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359-366.
- Houdé, O., y Tzourio-Mazoyer, N. (2003). Neural foundations of logical and mathematical cognition. *Nature reviews neuroscience*, 4(6), 507.
- Hua, Y., Guo, j., y Hua, Z. (2015). Deep Belief Networks and Deep Learning. In *Proc. Int. Conf. Intel. Comput. Internet Things*, 1-4.
- Huang, C. L., Fallah, Y. P., Sengupta, R., y Krishnan, H. (2010). Adaptive intervehicle communication control for cooperative safety systems. *IEEE network*, 24(1).
- Huang, G. B., y Chen, L. (2008). Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16-18), 3460-3468.
- Huang, G. B., Wang, D. H., y Lan, Y. (2011). Extreme learning machines: a survey. *International journal of machine learning and cybernetics*, 2(2), 107-122.
- Huang, G. B., Zhu, Q. Y., y Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3), 489-501.
- Hubel, D. H., y Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3), 574-591.
- Hubel, D. H., y Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1), 106-154.
- Interneuronal Network Model. *J. Neurosci.*, 16, 6402- 6413.
- Jain, A. K., Mao, J., y Mohiuddin, K. M. (1996). Artificial neural networks: *A tutorial*. *Computer*, 29(3), 31-44.

- Jamali, H. R., Azadi-Ahmadabadi, G., y Asadi, S. (2018). Interdisciplinary relations of converging technologies: Nano–Bio–Info–Cogno (NBIC). *Scientometrics*, *116*(2), 1055–1073.
- Jesan, J. (2004). *The neural approach to pattern recognition*. Magazine, Ubiquity. 2004, 2-2.
- Jing, H., Lun, X., Dan, L., Zhijie, H., y Zhiliang, W. (2015). Cognitive emotion model for eldercare robot in smart home. *China Communications*, *12*(4), 32-41.
- Johnson, C. P., y Myers, S. M. (2007). Identification and evaluation of children with autism spectrum disorders. *Pediatrics*, *120*(5), 1183-1215.
- Johnston, D., y Wu, S. M. S. (1994). *Foundations of cellular neurophysiology*. MIT press.
- Junek, W. (2007). Mind Reading: The Interactive Guide to Emotions. *J Can Acad Child Adolesc Psychiatry*. *16*(4): 182–183.
- Kahol, K., Tripathi, P., Panchanathan, S., y Rikakis, T. (Septiembre 2003). Gesture segmentation in complex motion sequences. In *ICIP* (2), 105-108.
- Kandel, E. R., y Schwartz, J. H. (2000). *Jessell ThM: Principles of neural science*. McGraw-Hill, New York.
- Kannappan, A., Tamilarasi, A., y Papageorgiou, E. I. (2011). Analyzing the performance of fuzzy cognitive maps with non-linear hebbian learning algorithm in predicting autistic disorder. *Expert Systems with Applications*, *38*(3), 1282-1292.
- Kapur, N. (1996). Paradoxical functional facilitation in brain-behaviour research: A critical review. *Brain*, *119*(5), 1775-1790.
- Kasabov, N., Dhoble, K., Nuntalid, N., y Indiveri, G. (2013). Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition. *Neural Networks*, *41*, 188-201.
- Kashani, M. M. R., Jangjou, M., Khaefinejad, N., & Laleh, T. (2012, March). Adventurous robots equipped with basic emotions. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support* (pp. 117-120). IEEE.
- Kass, R. E., y Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the american statistical association*, *90*(431), 928-934.
- Katsis, C. D., Rigas, G., Goletsis, Y., y Fotiadis, D. I. (2015). *Emotion recognition in car industry*. Emotion Recognition: A Pattern Analysis Approach, 515-544.
- Kehoe, B., Patil, S., Abbeel, P., y Goldberg, K. (2015). A survey of research on cloud robotics and automation. *IEEE Trans. Automation Science and Engineering*, *12*(2), 398-409.
- Keith, M. J., Jameson, A., Van Straten, W., Bailes, M., Johnston, S., Kramer, M., ... y Burke-Spolaor, S. (2010). The high time resolution universe pulsar survey–i. system configuration and initial discoveries. *Monthly Notices of the Royal Astronomical Society*, *409*(2), 619-627.
- Kelso, J. S. (2014). The dynamic brain in action: coordinative structures, criticality and coordination dynamics. *Criticality in neural systems*, 67-106.
- Khalid, S., Khalil, T., y Nasreen, S. (Agosto 2014). A survey of feature selection and feature extraction techniques in machine learning. In *Science and Information Conference (SAI)*,

- 2014, 372-378. IEEE.
- Kolli, A., Fasih, A., Al Machot, F., y Kyamakya, K. (Julio 2011). Non-intrusive car driver's emotion recognition using thermal camera. In *Nonlinear Dynamics and Synchronization (INDS) y 16th Int'l Symposium on Theoretical Electrical Engineering (ISTET)*, 2011 Joint 3rd Int'l Workshop on. IEEE,1-5.
- Kosko, B. (1986). Fuzzy cognitive maps. *International journal of man-machine studies*, 24(1), 65-75.
- Kumar, G., y Bhatia, P. K. (Febrero, 2014). A detailed review of feature extraction in image processing systems. In *Advanced Computing y Communication Technologies (ACCT)*, 2014 Fourth International Conference on. 5-12 IEEE.
- Kunda, M., & Goel, A. K. (2011). Thinking in pictures as a cognitive account of autism. *Journal of autism and developmental disorders*, 41(9), 1157-1177.
- Kurzweil, R. (2013). *How to create a mind: The secret of human thought revealed*. Penguin.
- Lauzon, F. Q. (2012). An introduction to deep learning. In *2012 11th International Conference on Information Science, Signal Processing and their Applications, ISSPA 2012*, 1438–1439.
- LeCun, Y., y Bengio, Y. (1995). *Convolutional networks for images, speech, and time series*. The handbook of brain theory and neural networks, 3361(10), 255–258.
- LeCun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- Lee, J. H., Ahn, C. W., y An, J. (Julio 2013). A honey bee swarm-inspired cooperation algorithm for foraging swarm robots: An empirical analysis. In *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*. IEEE. 489-493.
- Leung, R. C., Vogan, V. M., Powell, T. L., Anagnostou, E., y Taylor, M. J. (2016). The role of executive functions in social impairment in Autism Spectrum Disorder. *Child Neuropsychology*, 22(3), 336-344.
- Li, B., Sharma, A., Meng, J., Purushwalkam, S., y Gowen, E. (2017). Applying machine learning to identify autistic adults using imitation: An exploratory study. *PLoS ONE*. 12(8).
- Liang, N. Y., Huang, G. B., Saratchandran, P., y Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on neural networks*, 17(6), 1411-1423.
- Lin, N., Zong, C., Tomizuka, M., Song, P., Zhang, Z., Li, G., 2014. An overview on study of identification of driver behavior characteristics for automotive control. *Mathematical Problems in Engineering* 2014, 15.
- Liu, H., y Motoda, H. (Eds.). (1998). *Feature extraction, construction and selection: A data mining perspective* 453. Springer Science y Business Media.
- Liu, L., Wu, Y., Wei, W., Cao, W., Sahin, S., y Zhang, Q. (Julio 2018). Benchmarking deep learning frameworks: Design considerations, metrics and beyond. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*,1258-1269. IEEE.
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., y Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11–26.
- Lopes, N., y Ribeiro, B. (2015). *Machine Learning for Adaptive Many-Core Machines-A Practical*

- Approach*. Springer.
- Lord, C., Corsello C. (2005). Diagnostic instruments in autistic spectrum disorders. in: *Handb. Autism Pervasive Dev. Disord.*, 2, 730–771.
- Lord, C., Corsello, C., y Grzadzinski, R. (2014). *Diagnostic instruments in autistic spectrum disorders*. Handbook of Autism and Pervasive Developmental Disorders, Fourth Edition.
- Lord, C., Risi, S., Lambrecht, L., Cook, E. H., Leventhal, B. L., DiLavore, P. C., ... y Rutter, M. (2000). The Autism Diagnostic Observation Schedule—Generic: A standard measure of social and communication deficits associated with the spectrum of autism. *Journal of autism and developmental disorders*, 30(3), 205-223.
- Lord, C., Rutter, M., DiLavore, P. C., Risi, S., Gotham, K., y Bishop, S. L. (2012). *Autism diagnostic observation schedule, (ADOS-2) modules 1-4*. Los Angeles, California: Western Psychological Services.
- Löwel, S. and Singer, W. (Enero 1992). Selection of Intrinsic Horizontal Connections in the Visual Cortex Correlated Neural Activity, *Science*, 255(5041), 209-212.
- Lyon, R. J., Stappers, B. W., Cooper, S., Brooke, J. M., y Knowles, J. D. (2016). Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society*, 459(1), 1104-1123.
- Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9), 1659-1671.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.
- Markram, H. (2013). Seven challenges for neuroscience. *Functional Neurology*, 28(3), 145–151.
- Markram, H., Gerstner, W., y Sjöström, P. J. (2012). Spike-timing-dependent plasticity: A comprehensive overview. *Frontiers in Synaptic Neuroscience*. 4, 2.
- Masuyama, N., y Loo, C. K. (Agosto 2015). Robotic emotional model with personality factors based on Pleasant-Arousal scaling model. In *Robot and Human Interactive Communication (ROMAN), 2015 24th IEEE International Symposium on IEEE*, 19-24.
- Matias, T., Souza, F., Araújo, R., y Antunes, C. H. (2014). Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine. *Neurocomputing*, 129, 428-436.
- McClintock, J., Fraser, J.(2011). *Diagnostic instruments for autism spectrum disorder*, no. April, 30.
- McCulloch, W. S. (2016). *Embodiments of mind*. MIT press.
- McCulloch, W. S., y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
- Men, W., Falk, D., Sun, T., Chen, W., Li, J., Yin, D., ... y Fan, M. (2014). The corpus callosum of Albert Einstein’s brain: another clue to his high intelligence?. *Brain*, 137(4), e268-e268.
- Meyer, Y. (1993). *Wavelets-algorithms and applications*. Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation., 142.

- Michalski, R. S., Carbonell, J. G., y Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science y Business Media.
- Miller, K. D. (2016). Canonical computations of cerebral cortex. *Current opinion in neurobiology*, 37, 75-84.
- Minsky, M., Kurzweil, R., y Mann, S. (1991). Society of mind: A response to four reviews. *Artificial Intelligence*, 48(3), 371-396.
- Mohammadi, Mohammad-Reza, E. (2011). A Comprehensive Book on Autism Spectrum Disorders. InTech. <http://dx.doi.org/10.5772/18613>.
- Motoda, H., y Liu, H. (2002). Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan)* 5, 67-72.
- Mottron, L., y Dawson, M. (2013). *The autistic spectrum*. Handbook of Clinical Neurology, 111, 263–271.
- Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain: a journal of neurology*, 120(4), 701-722.
- Mythili, M.S., Mohamed A.R. y Shanavas. (2016). Meta heuristic based fuzzy cognitive map approach to support towards early prediction of cognitive disorders among children (MEHECOM), *Indian J. Sci. Technol.* 9(3).
- Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4), 049901.
- Nguyen, A., Yosinski, J., y Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 427-436.
- Nighot, M. K., Patil, V. H., y Mani, G. S. (Diciembre 2012). Multi-robot hunting based on swarm intelligence. In *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*. IEEE, 203-206.
- Niu, X. X., y Suen, C. Y. (2012). A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4), 1318-1325.
- O’Laughlin, C., y Thagard, P. (2000). Autism and coherence: A computational model. *Mind y Language*, 15(4), 375-392.
- Oikonomou, P., y Papageorgiou, E. I. (2013). Particle Swarm Optimization Approach for Fuzzy Cognitive Maps Applied to Autism Classification. *Artificial Intelligence Applications and Innovations, Aiai 2013*, 412, 516–526.
- Ojeda, J. (2015). A method based on genetic algorithms to support TEA diagnosis Un método basado en algoritmos genéticos de apoyo al diagnóstico TEA. *Actas de Ingeniería*, 1, 84–93.
- Ortony, A., Clore, G. L., y Collins, A. (1990). *The cognitive structure of emotions*. Cambridge university press.
- Ozonoff, S. (1997). *Components of executive function in autism and other disorders*. In J. Russell (Ed.), *Autism as an executive disorder*, 179-211. New York, NY, US: Oxford, University Press.

- Ozonoff, S., Pennington, B. F., y Rogers, S. J. (1991). Executive function deficits in high-functioning autistic individuals: relationship to theory of mind. *Journal of child Psychology and Psychiatry*, 32(7), 1081-1105.
- Ozonoff, S., Strayer, D. L., McMahon, W. M., y Filloux, F. (1994). Executive function abilities in autism and Tourette syndrome: An information processing approach. *Journal of child Psychology and Psychiatry*, 35(6), 1015-1032.
- Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., y Aguilar, J. (2018). *Towards the deployment of Machine Learning solutions in network traffic classification: A systematic survey*. IEEE Communications Surveys y Tutorials.
- Papageorgiou, E. I., y Iakovidis, D. K. (2013). Intuitionistic fuzzy cognitive maps. *IEEE Transactions on Fuzzy Systems*, 21(2), 342-354.
- Papageorgiou, E. I., y Kannappan, A. (2012b). Fuzzy cognitive map ensemble learning paradigm to solve classification problems: Application to autism identification. *Applied Soft Computing*, 12(12), 3798-3809.
- Papageorgiou, E. I., y Salmeron, J. L. (2012a). Learning fuzzy grey cognitive maps using nonlinear hebbian-based approach. *International Journal of Approximate Reasoning*, 53(1), 54-65.
- Papageorgiou, E. I., y Salmeron, J. L. (2013). A review of fuzzy cognitive maps research during the last decade. *IEEE Transactions on Fuzzy Systems*, 21(1), 66-79.
- Papageorgiou, E. I., Stylios, C. D., y Groumpos, P. P. (2003). An integrated two-level hierarchical system for decision making in radiation therapy based on fuzzy cognitive maps. *IEEE Transactions on Biomedical Engineering*, 50(12), 1326-1339.
- Papageorgiou, E.I., Stylios, C.D. (2008). *Fuzzy cognitive maps*, in: Handbook of Granular Computing, 755-774.
- Papanikolaou, K., Paliokosta, E., Houliaras, G., Vgenopoulou, S., Giouroukou, E., Pehlivanidis, A., ... y Tsiantis, I. (2009). Using the Autism Diagnostic Interview-Revised and the Autism Diagnostic Observation Schedule-Generic for the diagnosis of autism spectrum disorders in a Greek sample with a wide range of intellectual abilities. *Journal of autism and developmental disorders*, 39(3), 414-420.
- Parisi, L., Di Filippo, T., y Roccella, M. I. C. H. E. L. E. (2015). The child with Autism Spectrum Disorders (ASDs): Behavioral and neurobiological aspects. *Acta Medica Mediterranea*, 31(6), 1187-1194.
- Park, J., y Sandberg, I. W. (1993). Approximation and radial-basis-function networks. *Neural computation*, 5(2), 305-316.
- Paschero, M., Del Vescovo, G., Benucci, L., Rizzi, A., Santello, M., Fabbri, G., y Mascioli, F. F. (Mayo 2012). A real time classifier for emotion and stress recognition in a vehicle driver. In *2012 IEEE international symposium on industrial electronics*. IEEE, 1690-1695.
- Paterson, D. J. (2014). david H. Hubel Md Frs Nobel Laureate: A man of vision. *The Journal of physiology*, 592(1), 1-1.
- Pelleg, D., y Moore, A. W. (Junio, 2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, 1, 727-734.

- Pennington, B. F., y Ozonoff, S. (1996). Executive functions and developmental psychopathology. *Journal of child psychology and psychiatry*, 37(1), 51-87.
- Pérez, Á. E. G., Castro, J. L. A., Estrada, R. D. J. R., y González, E. D. (2016). Módulo conductual inmerso en una arquitectura de control para sistemas multi-robots. *Revista Ingeniería al Día*, 2(1), 40-57.
- Perozo, N., Aguilar, J., y Terán, O and Molina., H. (2013). A Verification Method for MASOES. *IEEE Transactions on Cybernetics*, 43 (1), 64-76.
- Perozo, N., Aguilar, J., y Terán, O. (Junio 2008). Proposal for a multiagent architecture for self-organizing systems (MA-SOS). In *International Conference on Intelligence and Security Informatics Springer, Berlin, Heidelberg*, 5075, 434-439.
- Pham, D. T., Dimov, S. S., y Nguyen, C. D. (2004). An incremental K-means algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 218(7), 783-795.
- Pham, D. T., Dimov, S. S., y Nguyen, C. D. (2005). Selection of K in K-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1), 103-119.
- Plaisted, K. C. (2015). *Reduced generalization in autism: An alternative to weak central coherence*. This chapter appeared In: *The development of Autism: Perspectives from Theory and Research (2001) Eds : Jacob A. Burack, Tony Charman, Nurit Yirmiya y Philip R. Zelazo Lawrence Erlbaum Associates, Inc, Publishers.*149-169.
- Portero, P. (Junio de 2016). Automatización de las torres de Hanói: Herramienta de apoyo para el estudio de la función ejecutiva “Planificación” en niños con síndrome de Asperger. *Conferencia llevada a cabo en el congreso CIAIQ2016*, (4). Porto, Portugal.
- Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.
- Prasad, L., y Iyengar, S. S. (1997). *Wavelet analysis with applications to image processing*. CRC press.
- Puerto, E. (2017). Avances en el conocimiento y modelado computacional del cerebro autista: Una revisión de literatura. *Cuaderno Activa*, (9), 109-125.
- Puerto, E. (Octubre de 2014). Análisis de la teoría de la mente humana basada en el reconocimiento de Patrones. In *Congreso Internacional En Innovación y Apropiación de las tecnologías de la información y las Comunicaciones, (CIINATIC 2014)*. Universidad Autónoma de Bucaramanga, Colombia.
- Puerto, E., (2017). Avances en el conocimiento y modelado computacional del cerebro autista: Una revisión de literatura." *Cuaderno activa* 9, 109-125.
- Puerto, E., (Octubre de 2016). Avances en el estudio del comportamiento del cerebro en tareas de reconocimiento de patrones. Conferencia llevada a cabo en *la III Semana Internacional y XI Semana de Ciencia, Tecnología e Innovación*, 138–143, Cúcuta, Colombia.
- Puerto, E., y Aguilar, J. (Noviembre 2016a). Formal description of a pattern for a recursive process of recognition. In *Computational Intelligence (LA-CCI), 2016 IEEE Latin American*

- Conference on*. 1-2.
- Puerto, E., y Aguilar, J. (2016b). Learning algorithm for the recursive pattern recognition model. *Applied Artificial Intelligence*, 30(7), 662-678.
- Puerto, E., y Aguilar, J. (2017a). Un algoritmo recursivo de reconocimiento de patrones. *Revista Técnica de la Facultad de Ingeniería Universidad del Zulia*, 40(2), 095-104.
- Puerto, E., Aguilar, J., y Chavez D. (2018). A recursive patterns matching model for the dynamic pattern recognition problem. *Applied Artificial Intelligence*, 32 (4), 1-14.
- Puerto, E., Aguilar, J., and Chavez, D. (2017b). A new recursive patterns matching model inspired in systematic theory of human mind, *International Journal of Advancements in Computing Technology (IJACT)*, 9(1), 28-39.
- Puerto, E., Aguilar, J., López, C., y Chávez, D. (2019). Using Multilayer Fuzzy Cognitive Maps to diagnose Autism Spectrum Disorder. *Applied Soft Computing*, 75, 58-71.
- Puerto, E., Aguilar, J., Reyes, J., y Sarkar, D. (Noviembre 2018a). Deep learning architecture for the recursive patterns recognition model. In *Journal of Physics: Conference Series 1126(1)*, 012035. IOP Publishing.
- Puerto, E., Aguilar, J., Reyes, J., y Sarkar, D. (Septiembre 2018b). Deep learning architecture for the recursive patterns recognition model. Conferencia llevada a cabo en *International Meeting Applied Sciences and Engineering*, Cúcuta, Colombia.
- Puerto, E., Aguilar, J., y Rojas Juan Carlos (2019). Ar2p-Text: Automatic recognition and correction system of misspelled words in Spanish based on the systematic model of brain functioning”. *International Journal on Document Analysis and Recognition (IJDA)* (En revision).
- Puerto, E., y Aguilar J., (2016a). Estado actual de los estudios cerebrales a nivel computacional. *Revista institucional investigación Metanoia*, 3(2), 199-120.
- Puerto, E., y Aguilar, J., (2016b). Learning algorithm for the recursive pattern recognition model. *Applied Artificial Intelligence* 30 (7), 662-678.
- R.H. Wozniak, N.B. Leezenbaum, J.B. Northrup, K.L. West, J.M. Iverson, The development of autism spectrum disorders: variability and causal complexity, *Wiley Interdiscip. Rev.: Cogn. Sci.* 8 (1–2) (2017).
- Rabbani, M. (2002). JPEG2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11(2), 286.
- Ramachandran, V. S. (2012). *The tell-tale brain: A neuroscientist's quest for what makes us human*. WW Norton y Company.
- Ramachandran, V. S., y Oberman, L. M. (2006). Broken mirrors: a theory of autism. *Scientific American*, 295(5), 62-69.
- Razavian, A., Azizpour, H., Sullivan, J., y Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 806-813.
- Ren, X., Wang, T., Altmeyer, M., y Schweizer, K. (2014). A learning-based account of fluid intelligence from the perspective of the position effect. *Learning and Individual*

- Differences*, 31, 30-35.
- Renato, S. (2003). *Neurons and Synapses: The History of Its Discovery*. Brain y Mind Magazine, 17.
- Reyes, M., Ponce, P., Grammatikou, D., y Molina, A. (2014). Methodology to weight evaluation areas from autism spectrum disorder ADOS-G test with artificial neural networks and taguchi method. *Revista Mexicana de Ingenieria Biomedica*, 35(3), 223–240.
- Rizzolatti, G., y Sinigaglia, C. (2016). The mirror mechanism: a basic principle of brain function. *Nature Reviews Neuroscience*, 17(12), 757.
- Rosenberg, A., Patterson, J. S., y Angelaki, D. E. (2015). A computational perspective on autism. In *Proc. of the National Academy of Sciences*, 112(30), 9158-9165.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Ross, T. J. (2013). *Fuzzy logic with engineering applications*. John Wiley y Sons.
- Ruck, D. W., Rogers, S. K., y Kabrisky, M. (1990). Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2), 40-48.
- Russell, J. E. (1997). *Autism as an executive disorder*. Oxford University Press.
- Rutter, M., LeCouteur, A. y Lord, C. (2015). Autism Diagnostic Interview - Revised (ADIR), Statew. Agric. L. Use Baseline, 1.
- Salakhutdinov, R., y Larochelle, H. (Marzo, 2010). Efficient learning of deep Boltzmann machines. In *Proc. of the thirteenth international conference on artificial intelligence and statistics*, 693-700.
- Samarasinghe, S. (2016). *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. Auerbach publications.
- Sandberg, A. (2013). Feasibility of whole brain emulation. In *Studies in Applied Philosophy, Epistemology and Rational Ethics*, 5, 251–264.
- Sandberg, A., y Bostrom, N. (2008). *Whole brain emulation: a roadmap*. Technical Report, #2008 2003, Oxford, UK: Future of Humanity Institute, Oxford U.
- Sarkar, D., Bali, R., y Sharma, T. (2018). Feature Engineering and Selection. In *Practical Machine Learning with Python*, 177-253. Apress, Berkeley, CA.
- Schliebs, S., y Kasabov, N. (2013). Evolving spiking neural network-a survey. *Evolving Systems*. 4(2), 87-98.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Selfe, L. (2012). *Nadia revisited: A longitudinal study of an autistic savant*. Psychology Press.
- Serruya, M. D. (2017 de junio). Connecting the brain to itself through an emulation. *Frontiers in Neuroscience*, 11, 373.
- Sharif Razavian, A., Azizpour, H., Sullivan, J., y Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proc. of the IEEE conference on computer vision and pattern recognition workshops*, 806-813.
- Shatz, C. J. (2013). David Hunter Hubel (1926–2013). *Nature*. 502 (7473).

- Shulz, D. E., y Feldman, D. E. (2013). Spike Timing-Dependent Plasticity. In *Neural Circuit Development and Function in the Healthy and Diseased Brain*. 35(0), 0-0.
- Sjöström, J., y Gerstner, W. (2010). Spike-timing dependent plasticity. *Spike-timing dependent plasticity*, 35(0), 0-0.
- Sompolinsky, H. (1987). *The theory of neural networks: The Hebb rule and beyond*. In Heidelberg colloquium on glassy dynamics, 485-527. Springer, Berlin, Heidelberg.
- Sousa, D. A. (2006). *How the Brain Learns*. Corwin Press.
- Srivastava, N., Salakhutdinov, R. R., y Hinton, G. E. (2013). Modeling documents with deep boltzmann machines. *arXiv preprint arXiv:1309.6865*.
- Stabel, A., et al., (2013). Diagnostic instruments in autistic spectrum disorders, in: *Encyclopedia of Autism Spectrum Disorders*, 2, 919–926.
- Stylios, C. D., Georgopoulos, V. C., Malandraki, G. A., y Chouliara, S. (2008). Fuzzy cognitive map architectures for medical decision support systems. *Applied Soft Computing*, 8(3), 1243-1251.
- Subbaraju, V., Sundaram, S., Narasimhan, S., y Suresh, M. B. (2015). Accurate detection of autism spectrum disorder from structural MRI using extended metacognitive radial basis function network. *Expert Systems with Applications*, 42(22), 8775-8790.
- Swiercz, W., Cios, K. J., Staley, K., Kurgan, L., Accurso, F., y Sagel, S. (2006). A new synaptic plasticity rule for networks of spiking neurons. *IEEE transactions on neural networks*, 17(1), 94-105.
- Tammet, D. (2013). *Thinking in numbers: on life, love, meaning, and math*. Little, Brown.
- Tang, Y. Y. (2000). *Wavelet theory and its application to pattern recognition* 36. World Scientific.
- Tarantino, L., Mazza, M., Valenti, M., y De Gasperis, G. (2016). Towards an Integrated Approach to Diagnosis, Assessment and Treatment in Autism Spectrum Disorders via a Gamified TEL System. In *methodologies and intelligent systems for technology enhanced learning (mis4tel)*, 478, 141–149.
- Tawari, A., y Trivedi, M. (Junio 2010). Speech based emotion classification framework for driver assistance system. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 174-178.
- Thabtah, F. (2018). Machine learning in autistic spectrum disorder behavioral research: *A review and ways forward*. *Informatics for Health and Social Care*. 1-20.
- Tolero, C.G., Echeverría C.M, Sánchez Q., Morales G., Vélez A. (2015). Trastorno del Espectro Autista y la Función Ejecutiva, *Acta Neurológica Colombiana*, 31(3), 246-252.
- Treffert, D. A., y Christensen, D. D. (2005). Inside the mind of a savant. *Scientific American*, 293(6), 108-113.
- Tsermentseli, S., Tabares, J. F., y Kouklari, E. C. (2018). The role of every-day executive function in social impairment and adaptive skills in Autism Spectrum Disorder with intellectual disability. *Research in Autism Spectrum Disorders*, 53, 1-6.
- Valdehita, A. S. M. (2016). Un corpus de bigramas utilizado como corrector ortográfico y gramatical destinado a hablantes nativos de español. *Revista signos: estudios de lingüística*, (90), 94-118.

- Van den Bekerom, B. (2017). *Using Machine Learning for Detection of Autism Spectrum Disorder*. Technical Report, University of Twente, The Netherlands.
- Van den Bergh, M., y Van Gool, L. (Enero 2011). Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, 66-72. IEEE.
- Vargas, R., Aguilar, J., y Puerto, E., (2018). Ciclo Autonomo de Analisis de Datos para el Diseño de Descriptores para Algoritmos de Aprendizaje Automatico. *Revista Venezolana de Computación*, 5 (2), 1-11.
- Vattikuti, S., y Chow, C. C. (2010). A Computational Model for Cerebral Cortical Dysfunction in Autism Spectrum Disorders. *Biological Psychiatry*, 67(7), 672-678.
- Vercelli, A., Rainero, I., Ciferri, L., Boido, M., y Pirri, F. (2018). Robots in Elderly Care.
- Vergini, E. S., y Groumpos, P. P. (2016). A new conception on the fuzzy cognitive maps method. *IFAC-PapersOnLine*, 49(29), 300-304.
- Vestbøstad, K. Y. (2018). Spiking Neural Networks for Pattern Recognition (Master's thesis, University of Bergen).
- Villena, J., González, B., González, B., y Muriel, M. (2002). STILUS: Sistema de revisión lingüística de textos en castellano. *Procesamiento del lenguaje natural*, 29, 305-306.
- Vogelstein, J. T., Amunts, K., Andreou, A., Angelaki, D., Ascoli, G., Bargmann, C., ... y Church, G. (2016). Grand challenges for global brain sciences. *arXiv preprint arXiv:1608.06548*.
- W. Froelich, Towards improving the efficiency of the fuzzy cognitive map classifier, *Neurocomputing* 232 (2017) 83–93.
- Wagner, P. K., Peres, S. M., Madeo, R. C. B., de Moraes Lima, C. A., y de Almeida Freitas, F. (Mayo 2014). Gesture Unit Segmentation Using Spatial-Temporal Information and Machine Learning. In *FLAIRS Conference*, 98.
- Wall, D. P., Dally, R., Luyster, R., Jung, J. Y., y DeLuca, T. F. (2012). Use of artificial intelligence to shorten the behavioral diagnosis of autism. *PLoS ONE*, 7(8).
- Wall, D. P., Kosmicki, J., Deluca, T. F., Harstad, E., y Fusaro, V. A. (2012). Use of machine learning to shorten observation-based screening and diagnosis of autism. *Translational psychiatry*, 2(4), e100.
- Wang, J., Zhang, L., Zhang, D., y Li, K. (2013). An adaptive longitudinal driving assistance system based on driver characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 14(1), 1-12.
- Wang, P., Li, W., Liu, S., Zhang, Y., Gao, Z., y Ogunbona, P. (Diciembre 2016). Large-scale continuous gesture recognition using convolutional neural networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, 13-18. IEEE.
- Wang, W., Xi, J., Chong, A., y Li, L. (2017). Driving style classification using a semisupervised support vector machine. *IEEE Transactions on Human-Machine Systems*, 47(5), 650-660.
- Wang, X. J. (2010). Neurophysiological and computational principles of cortical rhythms in cognition. *Physiological reviews*, 90(3), 1195-1268.
- Wang, X.-J., y Buzsáki, G. (1996). Gamma Oscillation by Synaptic Inhibition in a Hippocampal

- Watanabe, S. (Ed.). *Methodologies of pattern recognition*. Academic Press, 2015.
- Waterhouse, L. (1988). Extraordinary visual memory and pattern perception in an autistic boy. *The Exceptional Brain: Neuropsychology of Talent and Special Abilities*.
- Waterhouse, L., London, E., y Gillberg, C. (2016). ASD Validity. *Review Journal of Autism and Developmental Disorders*, 3(4), 302–329.
- Welling, H. (1994). Prime number identification in idiots savants: Can they calculate them? *Journal of Autism and Developmental Disorders*, 24(2), 199–207.
- Wickens, A. P. (2019). *Key Thinkers in Neuroscience*. Routledge.
- Widrow, B., y Lehr, M. A. (1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. In *Proc. of the IEEE*, 78(9), 1415-1442.
- Willsey, A. J., y State, M. W. (2015). Autism spectrum disorders: from genes to neurobiology. *Current opinion in neurobiology*, 30, 92-99.
- Witelson, S.F., Debra L. K, y Harvey T. The exceptional brain of Albert Einstein. *The Lancet* 353, 9170 (1999): 2149-2153.
- Wozniak, R. H., Leezenbaum, N. B., Northrup, J. B., West, K. L., y Iverson, J. M. (2017). The development of autism spectrum disorders: variability and causal complexity. *Wiley Interdisciplinary Reviews: Cognitive Science*, 8(1-2), e1426.
- Wu, D., Pigou, L., Kindermans, P. J., Le, N. D. H., Shao, L., Dambre, J., y Odobez, J. M. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8), 1583-1597.
- Wu, Q., McGinnity, T. M., Maguire, L., Valderrama-Gonzalez, G. D., & Cai, J. (2009, August). Detection of straight lines using a spiking neural network model. In *2009 Fifth International Conference on Natural Computation 2*, 385-389. IEEE.
- Yang, L., Ma, R., Zhang, H. M., Guan, W., y Jiang, S. (2018). Driving behavior recognition using EEG data from a simulated car-following experiment. *Accident Analysis y Prevention*, 116, 30-40.
- Zander, E., Sturm, H., & Bölte, S. (2015). The added value of the combined use of the Autism Diagnostic Interview–Revised and the Autism Diagnostic Observation Schedule: Diagnostic validity in a clinical Swedish sample of toddlers and young preschoolers. *Autism*, 19(2), 187-199.
- Zhang, F., Savadjiev, P., Cai, W., Song, Y., Rathi, Y., Tunç, B., ... y Verma, R. (2018). Whole brain white matter connectivity analysis using machine learning: an application to autism. *Neuroimage*, 172, 826-837.
- Zadeh L.A. (eds) *Feature Extraction. Studies in Fuzziness and Soft Computing*, 207. Springer, Berlin, Heidelberg.
- Zimmerman, A. W. (Ed.). (2008). *Autism: Current theories and evidence*. Springer Science y Business Media.