

**SISTEMA NEURO-FUZZY: PROSPECTIVA DE APLICACIÓN EN LA
DETECCIÓN DE FALLAS EN EQUIPOS DE SUBSUELO DE UNIDADES DE
LEVANTAMIENTO MECÁNICO**

**EDXON STIVEN MENESES CHACÓN
FREDY ALEXANDER GARAVITO ARENAS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA**

2014

**SISTEMA NEURO-FUZZY: PROSPECTIVA DE APLICACIÓN EN LA
DETECCIÓN DE FALLAS EN EQUIPOS DE SUBSUELO DE UNIDADES DE
LEVANTAMIENTO MECÁNICO**

**EDXON STIVEN MENESES CHACÓN
FREDY ALEXANDER GARAVITO ARENAS**

**Trabajo de grado para optar al título de
INGENIERO MECÁNICO**

**Director
JORGE ENRIQUE MENESES
Ingeniero Mecánico**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA**

2014

DEDICATORIA

A DIOS, dador de vida y salud.

A mi madre, Luz Marina Chacón por ser esa mujer incansable y trabajadora, merecedora de las mayores alegrías de este mundo, cada minuto de desmayo, en este camino de formación, tú me motivaste a seguir a delante.

A mi padre adoptivo, Fortunato Mejía por su apoyo incondicional.

A mi hermana, mujer que amo con toda mi alma.

A mis abuelos y tíos.

A Diana Forero por demostrarme que no solo de la academia se aprende, sino también del amor.

A mis amigos y compañeros, Fredy Gravito y Yosmer Bolaños, por los momentos compartidos, su amistad y enorme paciencia.

EDXON MENESES

DEDICATORIA

A Dios por darme el placer de pertenecer a una gran familia, a mi madre Nohora, a mi padre Ciro, y mis hermanos Edgar, Fernando, Marly y Yesica, por su paciencia y apoyado. No tengo palabras para mostrar mi admiración y gratitud, en el largo triunfo que acabo de culminar, solo puedo decir que los amo con todo mi corazón.

A mi compañero y amigo de proyecto de grado Edxon Meneses, por todos los días que se lucharon hombro a hombro para cumplir con esta meta propuesta hace algunos años atrás.

A cada uno de mis amigos que estuvieron a mi lado a Yosmer y a cada una de las personas que estuvieron pendientes de cada paso que daba durante mi proceso.

El color y el sabor de la vida esta, en la incansable lucha de ser mejor cada día.

FREDY GARAVITO

AGRADECIMIENTOS

Agradecer a cada uno de las personas que nos apoyaron en este proceso, resulta tedioso y complicado, porque fueron muchas las personas que se cruzaron por nuestro camino, dejando huellas en el sendero por el cual peregrinábamos, algunas más profundas que otras, pero no dejó de ser importante para nuestro crecimiento.

Al profesor Jorge Enrique Meneses Flores, Director de este proyecto, por todo el apoyo, sugerencias y guía ofrecida para el desarrollo de este trabajo.

A la Escuela de Ingeniería Mecánica, por los conocimientos adquiridos.

A la Universidad Industrial de Santander, por momentos y experiencias compartidas.

A los compañeros del laboratorio de Automatización Industrial, por compartir experiencias y ratos de alegría.

Edxon y Fredy

CONTENIDO

INTRODUCCIÓN	21
1. ALCANCES DEL PROYECTO.....	22
1.1 IDENTIFICACIÓN DEL PROBLEMA.....	22
1.2 JUSTIFICACIÓN.....	23
1.3 OBJETIVOS.....	24
1.3.1 Objetivo General	24
1.3.2 Objetivos Específicos	24
1.4 JUSTIFICACIÓN DE LA SOLUCIÓN.....	25
1.4.1 Planteamiento de las alternativas.....	25
1.4.2 Alternativa seleccionada	28
2. ANTECEDENTES DEL PROYECTO	30
2.1 TRABAJO UIS.	30
2.2 TRABAJO ENCONTRADO EN INTERNET.....	32
3. DESCRIPCIÓN DEL PROYECTO	33
3.1 OBTENCIÓN DE LA BASE DE CONOCIMIENTO. (FASE 1).....	35
3.1.1 Autores.....	35
3.1.2 Campo Escuela Colorado (CEC).....	36
3.1.3 Expertos.	36
3.1.4 Internet.	36
3.1.5 Herramientas Computacionales.....	37
3.2 PREPROCESAMIENTO (FASE 2).	40
3.2.1 Normalización.....	40
3.2.2 Discretización.....	42
3.3 MÁQUINA DE APRENDIZAJE (FASE 3).....	47
3.3.1 Sistemas Neuro-Fuzzy.....	47
3.3.2 Perceptrón Fuzzy.....	49
3.3.3 NEFCLASS-Q	56
3.4 MODELO FUZZY PARA DIAGNÓSTICO (FASE 4).	56

4.	OBTENCIÓN DE LA BASE DE CONOCIMIENTO.....	58
4.1	DESCRIPCIÓN GENERAL.....	58
4.1.1	Tipos de dinagramas.	59
4.2	INTERPRETACIÓN DE DINAGRAMAS	60
4.2.1	Proceso para identificar las partes de un Downhole card y sus problemas.	61
4.3	DINAGRAMAS EMPLEADOS.	62
4.3.1	Interferencia de Gas.....	63
4.3.2	Rotura varillas.	64
4.3.3	Golpe de Fluido.....	64
4.3.4	Fuga en Válvula Viajera o en Pistón.	65
4.3.5	Fuga en Válvula Fija.....	66
4.3.6	Barril de la Bomba Doblado o Pegándose.	66
4.3.7	Barril de la Bomba Gastado o Partido.....	68
4.3.8	Agujero en el barril de la bomba.....	69
4.3.9	Buen llenado tubería anclada.....	70
4.3.10	Ancla de Tubería en Mal Funcionamiento.....	71
4.4	CAMPO ESCUELA COLORADO (CEC).....	71
4.5	INFORMACIÓN OBTENIDA DE EXPERTOS.....	72
4.6	TIPIFICACIÓN DE DINAGRAMAS.	78
5.	PREPROCESAMIENTO DE DINAGRAMAS	81
5.1	NORMALIZACIÓN DE DINAGRAMAS.....	81
5.1.1	Procedimiento de Normalización.....	82
5.2	DISCRETIZACIÓN.....	84
5.2.1	Primer Método.....	84
5.2.2	Segundo Método.....	85
5.2.3	Tercer Método.....	88
6.	MÁQUINA DE APRENDIZAJE.....	92
6.1	SISTEMAS NEURO-FUZZY.....	92
6.1.1	Definición.	92
6.1.2	Métodos basados en sistemas Neuro-Fuzzy.	93
6.2	PERCEPTRÓN FUZZY.	97

6.2.1	Configuración del modelo.....	100
6.2.2	Preprocesamiento	101
6.2.3	Entrenamiento	102
6.3	NEFCLASS-Q.....	103
6.3.1	Creación de archivos.....	105
6.3.2	Organización de archivos.....	114
6.3.3	Procesos NEFCLASS-Q	116
6.3.4	Funcionamiento NEFCLASS-Q.....	135
7.	PRUEBAS Y RESULTADOS	148
7.1	CREACIÓN DE LA RED.....	148
7.2	VALIDACIÓN CON PRUNING.....	149
7.2.1	Pruning Simple.....	149
7.2.2	Pruning de las mejores reglas.....	150
7.2.3	Pruning de las mejores reglas por clase.....	151
7.3	CLASIFICACIÓN DE DINAGRAMAS REALES	152
8.	MODELO FUZZY.....	157
8.1	VARIABLES Y CONJUNTOS FUZZY.....	157
8.2	VARIABLES Y CONJUNTOS FUZZY.....	159
	CONCLUSIONES	162
	RECOMENDACIONES.....	164
	BIBLIOGRAFIA.....	165
	ANEXOS.....	167

LISTA DE FIGURAS

Figura 1. Diagnóstico por Expertos Humanos.	33
Figura 2. Fases del Proyecto.	34
Figura 3. Obtención del Conocimiento.	35
Figura 4. Necesidad de Ejemplos.	37
Figura 5. TWM (Total Well Management).	38
Figura 6. TAM (Total Asset Monitor)	38
Figura 7. TAM, Recorrido punto a punto.	39
Figura 8. Dinagrama tipificado en Excel.	39
Figura 9. Recorrido y Carga de dinagramas.	41
Figura 10 Dinagramas Normalizados.	42
Figura 11. Objetivo de Discretización.	43
Figura 12. División de dinagrama.	44
Figura 13. Amplitudes en cada segmento.	45
Figura 14. Amplitudes en carrera ascendente.	46
Figura 15. Amplitudes en Carrera Descendente.	46
Figura 16. Sistema Neuro-Fuzzy	48
Figura 17. Arquitectura Neuro-Fuzzy → Perceptrón Fuzzy.	50
Figura 18. Capa de Entrada.	50
Figura 19. Variable Fuzzy X_{26}	51
Figura 20. Antecedente de Regla.	52
Figura 21. Consecuente de la Regla.	52
Figura 22. Asignación de etiqueta lingüística.	53
Figura 23. Antecedente de la Regla 1.	54
Figura 24. Regla 1.	55
Figura 25. Capa 3 o capa de salida.	55
Figura 26. Modelo Fuzzy	57
Figura 27. Dinagrama de Superficie y Fondo	59
Figura 28. Partes del dinagrama.	61
Figura 29. Interferencia de Gas.	63
Figura 30. Rotura varillas.	64
Figura 31. Golpe de Fluido.	65
Figura 32. Fuga en Válvula Viajera o en Pistón.	65
Figura 33. Fuga en Válvula Fija.	66
Figura 34. Barril de la Bomba Doblado o Pegándose.	67
Figura 35. Barril Doblado.	67
Figura 36. Barril de la Bomba Gastado o Partido.	68
Figura 37. Barril Gastado.	69
Figura 38. Agujero en el barril de la bomba.	70
Figura 39. Buen llenado Tubería Anclada.	70

Figura 40. Ancla de Tubería Mal Funcionamiento	71
Figura 41. Golpe de Fluido con problemas en válvulas (C-03).	72
Figura 42. Golpe en carrera ascendente y descendente (C-07).	72
Figura 43. Funcionamiento Normal (Tubería anclada).	73
Figura 44. Interferencia de Gas.	73
Figura 45. Fuga en la Válvula Viajera.	73
Figura 46. Fuga en la Válvula Viajera.	74
Figura 47. Fuga en la Válvula Viajera.	74
Figura 48. Rotura de Varilla.	74
Figura 49. Rotura de Varilla.	75
Figura 50. Rotura de Varilla.	75
Figura 51. Golpe de Fluido.	75
Figura 52. Interferencia de Gas	76
Figura 53. Buen Funcionamiento (Tubería no Anclada).	76
Figura 54. Golpe de Fluido y Fuga en la Válvula viajera.	77
Figura 55. Golpe en carrera ascendente y mal funcionamiento de las válvulas. ...	77
Figura 56. Golpe en carrera ascendente y mal funcionamiento de las válvulas. ...	77
Figura 57. Recorrido Punto a Punto.	78
Figura 58. Almacenamiento del TAM a Excel.	79
Figura 59. Creación de dinagramas.	80
Figura 60. Campo petrolero.	81
Figura 61. Proceso de Normalización.	82
Figura 62. Necesidad de Discretizar.	84
Figura 63. Primera Método.	85
Figura 64. Búsqueda de Enlace.	85
Figura 65. Segundo Método.	86
Figura 66. Representación numérica del segundo Método.	86
Figura 67. Ingreso de amplitudes a la Máquina de Aprendizaje.	87
Figura 68. Efecto "Mirror".	88
Figura 69. Tercera Método.	89
Figura 70. Segmentos (L_j) y Amplitudes (X_i).	89
Figura 71. Dirección en la toma de amplitudes.	90
Figura 72. Representación numérica lineal de tercer Método.	90
Figura 73. Efecto "Mirror" tercer Método.	91
Figura 74. Diferentes modelos Neuro-Fuzzy cooperativos.	94
Figura 75. Modelo híbrido Neuro-Fuzzy.	95
Figura 76. Sistemas Neuro-Fuzzy.	96
Figura 77. Perceptrón Fuzzy.	97
Figura 78. Componentes NEFCLASS-Q.	105
Figura 79. Encabezado y Cuerpo de un archivo.	107
Figura 80. Organización de cada Patrón	107

Figura 81. Descripción numérica en un archivo de texto.	108
Figura 82. Estructura del archivo de Entrenamiento genérica.	109
Figura 83. Archivo de Entrenamiento creado y empleado.	111
Figura 84. Archivo de validación genérico.	112
Figura 85. Archivo de validación creado.	112
Figura 86. Archivo de Operación genérico.	113
Figura 87. Archivo de Operación creado.	114
Figura 88. Archivo de entrenamiento tipo 1.	115
Figura 89. Archivo de entrenamiento tipo 2.	116
Figura 90. Procesos NEFCLASS-Q.	117
Figura 91. Creación de la Red.	118
Figura 92. Variable 1 Fuzzificada.	119
Figura 93. Proceso de creación de Reglas.	120
Figura 94. Variable Fuzzy X_1	121
Figura 95. Variable Fuzzy X_2	121
Figura 96. Variable Fuzzy X_3	121
Figura 97. Variable Fuzzy X_4	122
Figura 98. Regla creada en la Red Neuro-Fuzzy.	123
Figura 99. Proceso de Entrenamiento.	125
Figura 100. Salida de la Red Neuro-Fuzzy.	130
Figura 101. Parámetros de un conjunto Fuzzy.	131
Figura 102. Función signo.	132
Figura 103. Conjunto modificado por entrenamiento.	133
Figura 104. Ejemplo entrenamiento.	133
Figura 105. Validación para un patrón de datos.	134
Figura 106. Interfaz software Nefclass-Q.	135
Figura 107. Asistente NEFCLASS-Q.	136
Figura 108. Nombre y Dirección de guardado.	136
Figura 109. Selección de archivos.	137
Figura 110. Asignación de cada tipo de archivo.	137
Figura 111. Creación de proyecto.	138
Figura 112. Inicio de Entrenamiento.	139
Figura 113. Puntos donde actúan los coeficientes de aprendizaje.	140
Figura 114. Visualizar, Reglas del Sistema.	141
Figura 115. Creación de Reglas.	142
Figura 116. Reglas a Priori.	142
Figura 117. Pruning de Reglas.	142
Figura 118. Visualizar, Entrar la Red.	143
Figura 119. Fase de entrenamiento.	144
Figura 120. Entrenamiento en curso.	144
Figura 121. Entrenamiento Terminado.	145

Figura 122. Visualizar, Validar el entrenamiento.....	145
Figura 123. Validación de Red.....	146
Figura 124. Clasificación de dinagramas.	147
Figura 125. Dinagrama no clasificado 1.....	153
Figura 126. Dinagrama no clasificado 2.....	154
Figura 127. Dinagrama no clasificado 3.....	154
Figura 128. Comparación de dinagrama aprendido al propagado en operación.	155
Figura 129. Comparación de dinagrama aprendido al propagado en operación 2.	155
Figura 130. Base de Reglas Fuzzy.	158
Figura 131. Conjuntos Fuzzy de la variable X_2	159
Figura 132. Conjuntos Fuzzy de la variable X_{12}	160
Figura 133. Conjuntos Fuzzy de la variable X_{15}	160
Figura 134. Conjuntos Fuzzy de la variable X_{40}	161
Figura 135. Conjuntos Fuzzy de la variable X_{50}	161

LISTA DE TABLAS

Tabla 1. Comparación de Control Neuronal y Control Fuzzy.....	49
Tabla 2. Caracterización de falla por zona.....	62
Tabla 3. Conjuntos que poseen mayor pertenencia.....	122
Tabla 4. Base de conocimiento.....	124
Tabla 5. Conjunto que tiene mayor a cero.	126
Tabla 6. Formación Regla A.	127
Tabla 7. Formación de la Regla B.....	127
Tabla 8. Formación de la Regla C.	128
Tabla 9. Formación de la Regla D.	128
Tabla 10. Posibles antecedentes de Reglas.....	129
Tabla 11. Activaciones de los conjuntos que componen la regla.....	129
Tabla 12. Codificación de los dinagramas.	148
Tabla 13. Tabla de resultados con pruning simple archivo tipo 1.	149
Tabla 14. Tabla de resultados con pruning simple archivo tipo 2.	149
Tabla 15. Tabla de resultados con las mejores reglas archivo tipo 1.....	150
Tabla 16. Tabla de resultados con las mejores reglas archivo tipo 2.....	150
Tabla 17. Tabla de resultados con las mejores reglas por clase archivo tipo 1. ...	151
Tabla 18. Tabla de resultados con las mejores reglas por clase archivo tipo 2. ...	152

LISTA DE ANEXOS

ANEXO A. BOMBEO MECÁNICO	168
ANEXO B. REDES NEURONALES ARTIFICIALES	173
ANEXO C. LÓGICA FUZZY	197

RESUMEN

TITULO: SISTEMA NEURO-FUZZY: PROSPECTIVA DE APLICACION EN LA DETECCION DE FALLAS EN EQUIPOS DE SUBSUELO DE UNIDADES DE LEVANTAMIENTO MECÁNICO¹

**AUTORES: EDXON STIVEN MENESES CHACÓN²
FREDY ALEXANDER GARAVITO ARENAS**

PALABRAS CLAVES: Pozo inteligente, Dinagramas, Diagnóstico automático, sistema Neuro-Fuzzy, NEFCLASS-Q.

DESCRIPCIÓN:

A medida que pasan los años se presenta con mayor necesidad el realizar diagnósticos de maquinaria con base en el estudio de variables (vibraciones, ultrasonido, dinagramas, etc.) que definen la condición del equipo. Este proceso de diagnóstico generalmente lo realiza personal calificado y experimentado en los diferentes campos. Actualmente en la UIS se ejecuta el proyecto de investigación “Desarrollo de un prototipo de pozo inteligente para Campo Escuela Colorado”, financiado por la Vicerrectoría de Investigación y Extensión, el cual busca identificar de forma precisa, rápida y automática los problemas en un pozo.

La necesidad actual del proyecto de investigación referido radica en generar una estrategia software capaz de reconocer y diagnosticar los diferentes estados del equipo de levantamiento artificial. El presente proyecto pretende resolver dicha necesidad presentando una alternativa para la interpretación de dinagramas, proporcionando agilidad y rapidez en el reconocimiento de problemas.

Después de analizar las diferentes alternativas de solución, se decidió utilizar un software basado en una arquitectura Neuro-Fuzzy, desarrollado en el laboratorio de Automatización Industrial, llamado Nefclass-Q. Para entrenar, validar y clasificar se crearon dinagramas con los diez tipos de fallas más comunes, teniendo en cuenta su forma característica, a la cual posteriormente se le realiza un tratamiento en su contenido. Por último se evalúa si el uso de esta arquitectura es viable para el diagnóstico de dinagramas, además de su implementación en el proyecto de investigación.

¹ Trabajo de Grado

² Facultad de Ingenierías Físico Mecánicas, Escuela de Ingeniería Mecánica, Director: Ing. Jorge Enrique Meneses Flórez, Universidad Industrial de Santander.

ABSTRACT

TITLE: NEURO-FUZZY SYSTEM: PROSPECTIVE OF APLICATION IN THE FAILURES DETECTION IN UNDERGROUND UNITS OF MECHANICAL LIFT³.

**AUTHORS: EDXON STIVEN MENESES CHACÓN⁴
FREDY ALEXANDER GARAVITO ARENAS**

DESCRIPTION:

In this recent years the industry needs with more necessity machinery diagnosis on study variables (vibration, ultrasound, dinagramas, etc.) that define the condition of the equipment. This diagnosis process is done generally by experienced workers or professionals.

Recently, the Santander Industrial University is executing a research project call "Prototype develop of an intelligent well for Campo Escuela Colorado" funded by Vicerectoria for Research and Extension. This project wants to identify in an accurate, fast and automatic form problems that can be found in an oil well. This project requires a software strategy that could be able to recognize and diagnose different problems or failures of artificial lift equipment.

The present project pretend to solve that necessity giving an alternative solution for reading dinagramas adding agility and quickness to failures recognition. Analyzing different alternatives of solution, everything came to use a software based on a Neuro-Fuzzy architecture. Nefclass-Q was the software used for training, validating and classification. This software was develop in the Industrial Automation Laboratory, Mechanical Engineering department.

It was created ten dinagramas for the most common failures, taking into a count the form of its spectrum which after the acquisition is necessary to make a treatment of its information.

Finally it is assessed if the uses of this architecture is viable for Dinagramas diagnosis, and if it is useful for applying in the macro project.

³ Graduate Work.

⁴ Faculty Physical-Mechanics Engineering, Mechanical Engineering School, Director: Jorge Enrique Meneses Flórez, Universidad Industrial de Santander.

INTRODUCCIÓN

El Bombeo mecánico⁵ es el más común de los métodos de levantamiento artificial, abarca cerca del 90% de todos los pozos, haciendo de este el método primario de levantamiento doméstico. Es usualmente el sistema más fácil de mantener cuando se diseña apropiadamente. En campo, una variedad de problemas técnicos pueden suceder, en especial en los equipos de fondo lo que hace indispensable identificar estos problemas en el menor tiempo posible y de forma acertada.

Para controlar correctamente la operación de estos sistemas se hace necesaria la supervisión constante de un dinamómetro⁶, que se ubica en la barra lisa en el equipo de superficie. El dinagramas⁷ es el resultado de los datos adquiridos por el dinamómetro, que relaciona la carga a la que se somete la barra lisa contra el desplazamiento del pistón de la bomba.

Con el fin de generar menores costos de operación y aumentar las tasas de producción en nuestro país, actualmente en la UIS se ejecuta un proyecto de investigación “Desarrollo de un prototipo de pozo inteligente para CEC⁸” el cual contribuye al desarrollo de un hardware y software propio desistiendo de la importación de los mismos. La necesidad actual del proyecto de investigación se basa en el desarrollo de una estrategia software que reconozca y diagnostique de forma automática los dinagramas de fondo, prescindiendo de la intervención de un experto.

Con base en lo anterior y motivados por aportar en un proyecto de investigación de tal magnitud e impacto, este trabajo de grado explora la alternativa de operar un software basado en una arquitectura Neuro-Fuzzy⁹ para la clasificación de estos dinagramas de forma efectiva.

⁵ Ver Anexo A

⁶ Instrumento usado para la toma de datos en equipos de bombeo mecánico.

⁷ Ver Capitulo 4.

⁸ Proyecto 8556. Desarrollo de un Prototipo de POZO INTELIGENTE para CEC Fase 1. Financiado por la VIE – CEC (Campo Escuela Colorado).

⁹ Ver Capitulo 6.

1. ALCANCES DEL PROYECTO

1.1 IDENTIFICACIÓN DEL PROBLEMA.

Campo Escuela Colorado (CEC) es el resultado de la unión entre La Universidad Industrial de Santander y La Empresa Colombiana de Petróleos – ECOPETROL S.A con fines Científicos y Tecnológicos, el cual tiene dentro de su misión desarrollar proyectos de investigación y apropiación tecnológica.

En CEC se encuentran perforados 75 pozos, no todos se encuentran activos pero la gran mayoría trabajan con bombeo mecánico y únicamente con la supervisión de operarios generando riesgos operacionales y en conjunto deficiencias en la producción, ya que la supervisión del operario no cubre todos los aspectos y variables con que funciona el bombeo mecánico. Además de estos problemas, el sistema de bombeo, físicamente, también se ve vulnerable a fallas en cualquiera de sus componentes por problemas que un operario no podría prever, bien sea la sarta de varillas, la bomba, el motor, la caja de engranajes, etc. y bien sea por diferentes razones como interferencias por gas, baches de arena, bajos niveles de fluido, entre otros. Impartiendo en la búsqueda de generar menores costos de operación, aumentar las tasas de producción, reducir los gastos de capital, aumentar las reservas, y haciendo que se pueda identificar de forma precisa y rápida problemas en un pozo, nació la idea de crear un pozo y campo petrolero inteligente. Con este propósito, en la UIS actualmente se ejecuta el proyecto de investigación “Desarrollo de un prototipo de pozo inteligente para CEC”, financiado por la Vicerrectoría de Investigación y Extensión. El proyecto contribuye al desarrollo de hardware y software propio (nacional), ya que la industria colombiana se ha dedicado en importar paquetes de automatización proporcionados por empresas extranjeras generando absoluta dependencia en todos sus aspectos, económica, mantenimiento, funcionamiento, etc.

En el proyecto de investigación referido, se desarrolla un sistema automático e inteligente que debe reconocer los patrones de falla del sistema de bombeo

mecánico a partir de los dinagramas de fondo, generados a partir de toda la información censada en el pozo y posteriormente tratada. Los componentes hardware y software pertinentes a la adquisición de señales y procesamiento de las mismas, ya han sido desarrollados.

La necesidad actual en el proyecto de investigación, radica en el desarrollo de una estrategia software que reconozca y diagnostique de manera automática los dinagramas de un pozo. El presente proyecto de grado pretende resolver dicha necesidad involucrando sistemas inteligentes capaces de identificar las diferentes fallas presentes en un proceso de producción, prescindiendo de la intervención de un experto.

1.2 JUSTIFICACIÓN

La justificación específica para este proyecto radicó en la agilidad y precisión con la cual se identificaron las diferentes fallas en un pozo petrolero. Ya que generalmente en estos campos se generan grandes cantidades de dinagramas de fondo, lo cual exige tiempo, precisión en la clasificación y la necesidad de un operario experimentado en el tema. También se pretendió proyectar la capacidad de desarrollo tecnológico que nuestro país puede llegar a tener, buscando independizar la importación de equipos y herramientas necesarios para mantener esta visión de campo inteligente.

La implementación del concepto de pozo y campo inteligente proporciona una visión de reacción rápida y oportuna ante acontecimientos inesperados durante la vida de un pozo, proporciona una operación autónoma, eficiente y continua, contribuyendo al aumento de la producción a través de la monitorización y gestión del depósito de crudo, además de disminuir la necesidad de intervención humana, lo cual conduce a crear valor mediante la reducción de los costos de intervención, reduciendo por ende los costos operativos.

1.3 OBJETIVOS

1.3.1 Objetivo General

- ✓ Contribuir con la misión de la Universidad Industrial de Santander de formar profesionales con capacidad de investigación e innovación, proporcionando una visión acerca de las posibilidades de aplicación de las técnicas de inteligencia artificial en la ingeniería del mantenimiento, generando alternativas de solución viables, con la intención de aportar crecimiento en el desarrollo tecnológico de la región y el país.

1.3.2 Objetivos Específicos

- ✓ Obtener o crear dinagramas de fondo de las 10 fallas más generales y frecuentes presentadas en unidades de levantamiento mecánico:
 - Buen llenado tubería anclada.
 - Interferencia de Gas.
 - Rotura varillas.
 - Agujero en el barril de la bomba.
 - Golpe de Fluido.
 - Fuga en Válvula Viajera o en Pistón.
 - Fuga en Válvula Fija.
 - Barril de la Bomba Doblado o Pegándose.
 - Barril de la Bomba Gastado o Partido.
 - Ancla de Tubería en Mal Funcionamiento.

- ✓ Desarrollar un proceso de tratamiento de la información contenida en los dinagramas de fondo, con la intención de que el software asimile su información:
 - Normalización.
 - Discretización.

- Creación de los archivos de entrenamiento, validación y clasificación.
- ✓ Aplicar la plataforma Nefclass-Q con el propósito de identificar y diagnosticar las 10 fallas anteriormente nombradas.
- ✓ Evaluar si el uso de una arquitectura Neuro-Fuzzy presenta resultados sobresalientes para su aplicación en el campo del diagnóstico automático.

1.4 JUSTIFICACIÓN DE LA SOLUCIÓN

1.4.1 Planteamiento de las alternativas

Actualmente en el mercado existen numerosas herramientas enfocadas a la clasificación de patrones de datos, que facilitan los procesos de diagnóstico automático, por lo tanto se hace necesario hacer un análisis de alternativas en busca de que el software (Arquitectura) a seleccionar tenga la calidad requerida para poder ser implementado en el ámbito industrial. Los modelos usados para estos sistemas aunque escasos son muy eficientes. Podemos encontrar aplicaciones que van desde la predicción hasta el control automático pasando por el reconocimiento y clasificación de patrones, en campos como la medicina, la industria del petróleo, la ingeniería, etc. Existen múltiples métodos usados para el desarrollo de los aspectos fundamentales de un sistema inteligente, abarcando así el aprendizaje, el conocimiento generado y la generación de respuestas lógicas. Cada uno de ellos proporciona ventajas competitivas respecto a los demás modelos, pero a su vez presentan desventajas con los mismos. Las técnicas más representativas son:

1.4.1.1 Redes Neuronales Artificiales¹⁰

Es un sistema formado por una cantidad determinada de elementos de procesamiento llamados nodos o neuronas, interconectados en una arquitectura

¹⁰ Ver Anexo B

inspirada en la estructura del cerebro. Las ventajas que se tienen con esta técnica son:

- Adquieren el conocimiento a partir de ejemplos.
- El conocimiento se almacena de una forma descentralizada, es decir, que si desaparecen algunos nodos de la Red no se verá afectado en gran medida el conocimiento almacenado en la misma.
- No requiere de una base de conocimiento, ya que esta es adquirida durante el aprendizaje.
- No requiere de modelos matemáticos para el proceso.

Por otro lado existen ciertos aspectos en forma de desventajas:

- Se comportan como una caja negra, es decir, los procesos al interior de la misma son inciertos.
- Después del entrenamiento de la Red, no se puede extraer el conocimiento.
- Si se dispusiera de la configuración final (después del entrenamiento de la Red) no sería posible por un ser humano interpretar tal información.

Si bien este tema puede prestarse para pensar que el trabajo allí realizado solo es aplicable a las ciencias computacionales, existen infinidad de aplicaciones en campos de la ingeniería, la medicina, la economía, etc., mostrando en cada uno de ellos excelentes resultados.

1.4.1.2 Sistemas basados en Lógica Fuzzy¹¹.

La lógica Fuzzy o difusa es un área de la soft computing que permite al software manejar variables con incertidumbre. Estos sistemas son compuestos por inferencias Fuzzy que contiene reglas sintácticas de la forma **SI** *predicado a* y *predicado b...* **ENTONCES** *Consecuente*, formando estas lo que se conoce como base de reglas. Los sistemas basados en Lógica Fuzzy imitan la forma en que tomas decisión los seres humanos con la ventaja de ser mucho más rápidos. Las ventajas presentes en estos sistemas son:

¹¹ Ver Anexo C

- No requiere de modelos matemáticos para el proceso.
- Permite usar conocimiento previo (Reglas de la forma SI-ENTONCES).
- Gracias al uso de las Reglas sintácticas se facilita la interpretación por el ser humano.

Aun así existen grandes problemas a la hora de generar los modelos Fuzzy.

- No pueden tener procesos de entrenamiento, es decir no pueden aprender.
- No existen métodos formales de optimización de conjunto Fuzzy.
- El sistema funciona solo para el entorno en el que fue creado

1.4.1.3 Algoritmos genéticos

El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin. Los algoritmos genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. Las ventajas son las siguientes:

- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.

Las desventajas son las siguientes:

- Pueden tardar mucho en dar una solución, o no encontrarla, dependiendo en cierta medida de los parámetros que se utilicen tamaño de la población, número de generaciones, etc.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole.

1.4.1.4 Sistemas Híbridos Inteligentes

Los sistemas híbridos inteligentes denotan a los sistemas software que emplean en paralelo una combinación de modelos de inteligencia artificial, métodos y técnicas de éstos subcampos como:

- Sistemas Neuro-Fuzzy.
- Redes Neuronales Evolutivas
- Sistemas Genetic-Fuzzy-Neural
- Sistemas difusos genéticos.
- Aprendizaje de algoritmos genéticos difusos (Genetic algorithm fuzzy reinforcement learning, GAFRL)

1.4.2 Alternativa seleccionada

Gracias a los antecedentes que se han tenido no solo en la Escuela de Ingeniería Mecánica sino en el exterior y a las ventajas que tienen los sistemas Neuro-Fuzzy, se puede decir que estos sistemas pueden proporcionar las características necesarias para su implementación en el diagnóstico de dinagramas.

1.4.2.2 Sistemas Neuro-Fuzzy

Los sistemas Neuro-Fuzzy nacen de la necesidad de solucionar los problemas que presentan los sistemas basados en Lógica Fuzzy que son:

- Requiere conocimiento previo explícito en forma de reglas SI-ENTONCES y conjuntos Fuzzy.
- Requiere una sintonización manual de los conjuntos Fuzzy, lo que lleva tiempo y es propenso al error.

Estos problemas se solucionan mediante un proceso de aprendizaje automático de las reglas y de los conjuntos Fuzzy. Este proceso de aprendizaje lo pueden dar las Redes Neuronales Artificiales ya que una de sus ventajas es el aprendizaje a partir de ejemplos. Cada sistema mejora el funcionamiento como apoyo o respaldo para uno y viceversa. Estos sistemas presentan las siguientes características:

- Acortar el tiempo de aprendizaje usando conocimiento previo.
- Habilidad de aprendizaje a partir de ejemplos por medio del entrenamiento del sistema.
- Proporciona medios para interpretar los resultados con criterios ambiguos, reglas de la forma SI-ENTONCES.
- Resultados más eficientes en períodos más cortos.

2. ANTECEDENTES DEL PROYECTO

REDES NEURONALES ARTIFICIALES EN EL DIAGNÓSTICO DE FALLAS EN EQUIPOS DE SUBSUELO.

2.1 TRABAJO UIS.

En la universidad Industrial de Santander se realizó un proyecto de grado llamado “Análisis y diagnóstico de problemas en el equipo de subsuelo de unidades de bombeo mecánico utilizando Redes Neuronales” realizado por los estudiantes Cristian Fernando Mateus Marín y Daniel Mauricio Rojas Cárdenas bajo la dirección del M.Sc. Fernando Enrique Calvete y la codirección del Ing. Elkin Rodolfo Santafé donde se realizan pruebas para diagnosticar problemas presentes en equipos de subsuelo de forma automática. Se usaron patrones (dinagramas) con problemas tales como:

- Funcionamiento normal con sarta no anclada.
- Interferencia de gas.
- Daño en la válvula fija.
- Daño en la válvula viajera.
- Golpe de fluido.
- Varillas partidas.
- Fuga en el barril.
- Golpe abajo.
- Presencia de parafinas o asfáltenos en la bomba.

Para ingresar los datos de entrada a la Red neuronal dispusieron obtener 60 puntos representativos de la forma de cada dinagrama. Con este planteamiento establecieron que la Red neuronal a ser utilizada debía contener 60 elementos de entrada que corresponden a los 60 puntos significativos extraídos y representativos de cada diagrama, y que cada elemento o neurona de salida corresponde a un problema o situación en el equipo de subsuelo de las unidades de bombeo mecánico a ser identificado.

El trabajo se realizó en cuatro etapas:

- **Etapa 1:** En ésta etapa se recolecto, y organizo la información necesaria requerida sobre cada dinagrama, establecidas de acuerdo a las entradas y salidas de la Red. Además se dispuso la información de interpretación de cada dinagrama de acuerdo a la forma como se espera que la Red identifique los problemas.
- **Etapa 2:** En esta etapa se explican los principios y procesos necesarios para adecuar la información de cada dinagrama y disponerla para el software de Redes Neuronales Artificiales.
- **Etapa 3:** Luego de obtener los dinagramas en el formato de entrada al software de RNA (Redes Neuronales Artificiales) se realizó el entrenamiento de la Red. En este punto se le proporciono además del archivo proveniente del preprocesador, características de la Red a entrenar sobre número de capas ocultas, número de neuronas en cada una de ellas. El entrenamiento se realiza utilizando el algoritmo de aprendizaje “Backpropagation”. Luego de terminado el entrenamiento, se debe observar que el error marginal sea inferior al 1% y que sean aprendidos la totalidad de los patrones de entrenamiento (aprendizaje = 100%), si los resultados no satisfacen alguno de estos dos criterios se regresa a los parámetros de entrenamiento y se realizan modificaciones pertinentes. La evaluación de la Red se realiza teniendo en cuenta la repuesta a los patrones de entrenamiento y a los patrones de prueba constituidos por dinagramas de pozos con uno o dos problemas en el equipo de subsuelo de la unidad de bombeo mecánico. Si la evaluación de la Red arroja resultados razonables en cuanto al reconocimiento de uno y dos problemas, entonces se termina el proceso de optimización de la Red, en caso contrario es necesario modificar la estructura de la Red y volver a realizar el proceso.
- **Etapa 4:** Análisis de resultados y conclusiones.

2.2 TRABAJO ENCONTRADO EN INTERNET.

Identificación de Cartas Dinagráficas de Fondo mediante el uso de Redes Neuronales¹².

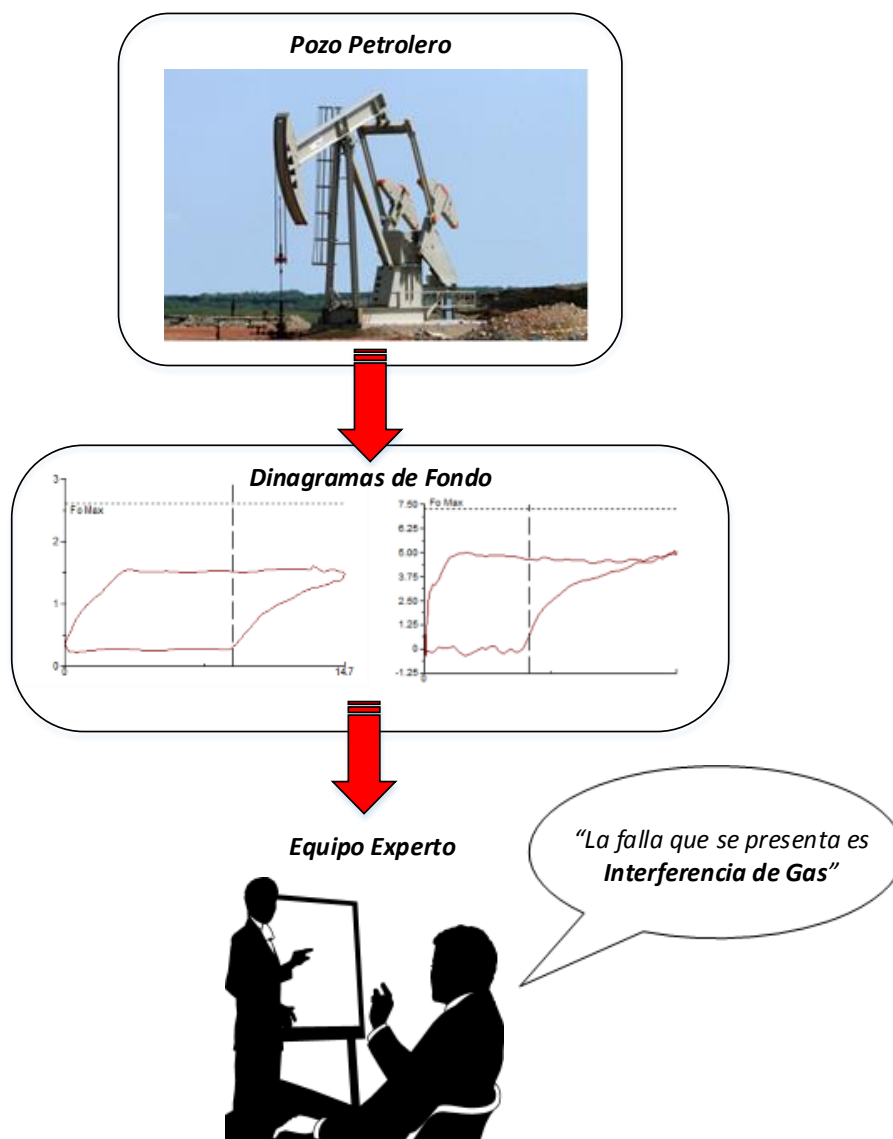
Evaluaron un conjunto de características geométricas y estadísticas de 30 dinagramas de una base de datos proporcionada por la RESERVOIR TECHNOLOGY. Mediante el cálculo de la varianza de dichas características normalizadas, de 25 seleccionaron 18, las cuales presentaron un valor grande de varianza y con las cuales no se exige demasiado gasto computacional. Para el desarrollo del software implementaron una Red tipo Backpropagation debido a que la aplicación requería de una distribución de diferentes dinagramas conocidos, relacionando cada uno con una clase que en el caso específico es un número entero iniciando desde el número uno (1) hasta el número de dinagramas clasificados. La inteligencia artificial de la Red neuronal permite una distinción de formas y a cada forma se le asigna una clase, la cual corresponde a una falla definida por la carta dinagráfica de fondo.

¹² SALGADO, José; Et al. Identificación de cartas dinagráficas de fondo mediante el uso de Redes Neuronales como herramienta de ayuda en el Bombeo Mecánico, de:
<http://es.scribd.com/doc/166720551/Identificacion-de-Cartas-Dinograficas-de-Fondo>

3. DESCRIPCIÓN DEL PROYECTO

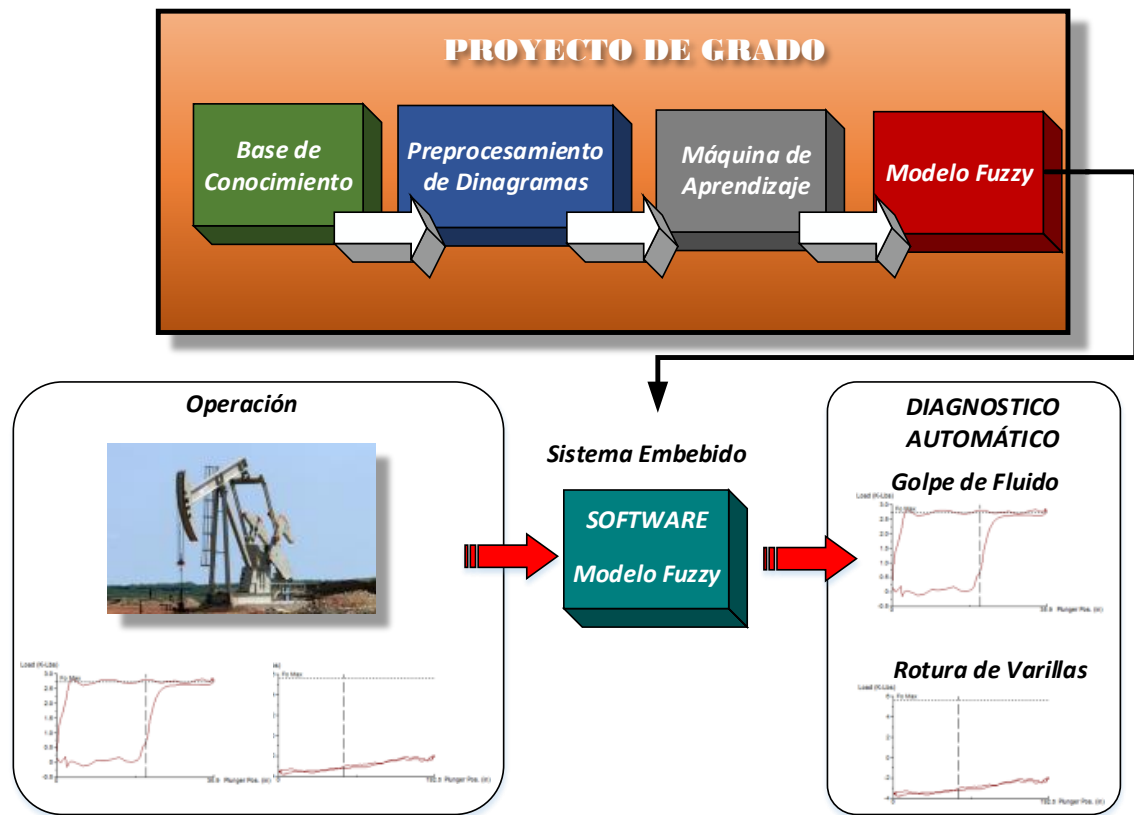
Tradicionalmente el diagnóstico de una falla en un pozo petrolero se lleva a cabo gracias al conocimiento y habilidad de un experto humano, (ver figura 1). Su metodología se basa en hallar problemas comparando las formas de las cartas dinagráficas con formas características de falla.

Figura 1. Diagnóstico por Expertos Humanos.



El producto final al que direcciona el presente proyecto de grado (ver figura 2), es un modelo Fuzzy, a partir del cual, en un sistema digital (sistema embebido), se podría diagnosticar un pozo en tiempo real y sin la intervención de un experto humano.

Figura 2. Fases del Proyecto.



Para obtener el Modelo Fuzzy que permite diagnosticar automáticamente, se desarrollaron 4 fases:

1. Obtención de la BASE DE CONOCIMIENTO.
2. PREPROCESAMIENTO de las cartas dinagráficas de la base de conocimiento.
3. MÁQUINA DE APRENDIZAJE.
4. Obtención del MODELO FUZZY.

3.1 OBTENCIÓN DE LA BASE DE CONOCIMIENTO. (FASE 1).

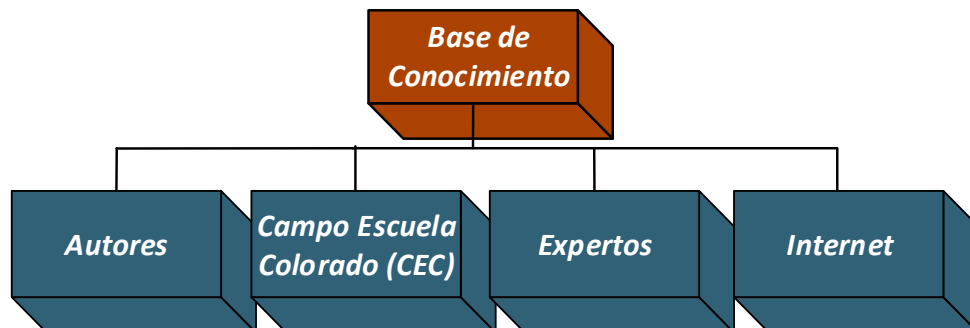
El objetivo principal de esta fase, es obtener una serie de dinagramas ya clasificados según su falla, por algún experto humano. A partir de esta base de conocimiento, en fases posteriores se sintetiza digitalmente, por medio de un aprendizaje basado en ejemplos, utilizando la arquitectura de un algoritmo Neuro-Fuzzy.

Para generar esta Base de Conocimiento se tenían 3 opciones:

1. Obtener los dinagramas de su fuente matriz (Pozo Petrolero).
2. Obtenerlos de las fuentes de información de la figura 3.
3. Tipificar sus formas, guiados por las fuentes de información de la figura 3 y con la ayuda de herramientas computacionales.

Debido a la dificultad de obtener información de fallas directamente desde campos petroleros se impartió en la búsqueda de ejemplos de fallas en diferentes fuentes de información:

Figura 3. Obtención del Conocimiento.



3.1.1 Autores

Siendo el tema, análisis de dinagramas para diagnóstico de equipos de bombeo mecánico ajeno a los intereses generales de enseñanza en la Escuela de Ingeniería Mecánica, se origina la necesidad en destinar tiempo para aprender y reconocer las fallas más comunes presentadas en equipos donde se emplea este

método de levantamiento artificial. Se emplearon textos y guías como: Optimización de Bombeo Mecánico¹³, y Libro de Bombeo Mecánico de la Especialización en Hidrocarburos de la UIS¹⁴. Estos textos poseen información teórica de formas de dinagramas y causas de falla más no ejemplos reales de los mismos.

3.1.2 Campo Escuela Colorado (CEC).

En CEC se obtuvo información pero incompleta, es decir, no se tenían suficientes ejemplos de fallas, además solo poseían dinagramas con fallas combinadas.

3.1.3 Expertos.

Se consultaron dos expertos en análisis de dinagramas de fondo. Uno de ellos, el Ingeniero Fernando Enrique Calvete González, profesor de la escuela de Ingeniería de Petróleos de la UIS, el cual dirigió un proyecto donde se diagnostica automáticamente dinagramas de fondo pero empleado la arquitectura de un algoritmo basado Redes Neuronales Artificiales (RNA). El otro experto consultado es Alexander Camacho Pardo, Ingeniero de Petróleos egresado de la UIS con amplio conocimiento en reconocimiento y diagnóstico de dinagramas. Estos expertos proporcionaron dinagramas reales, pero muy pocos ejemplos.

3.1.4 Internet.

Se encontró información de dinagramas en una base de datos de la empresa ECHOMETER¹⁵, donde se tienen varios ejemplos de fallas, pero de distinta clase. Para hacer visible la información de esta base de datos se necesita instalar el software TWM (Total Well Management), el cual permite visualizar este tipo de

¹³ THETA, Enterprises. Bombeo Mecánico Optimización: Entrenamiento Levantamiento Artificial. USA. p. 188-233.

¹⁴ MONTES, Erik. Asignatura: Bombeo Mecánico. Bucaramanga. Universidad Industrial de Santander. 2013. p. 191.

¹⁵ ECHOMETER. TWM, [Software].

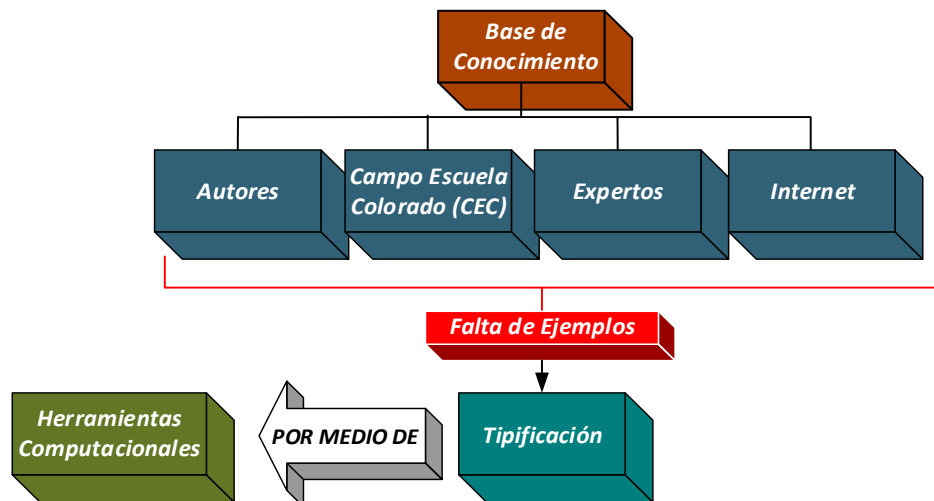
<<http://echometer.com/Software/TotalWellManagement/tabid/127/Default.aspx>>

archivos, o en su defecto el TAM (Total Well Management) que es un software que pertenece a la misma empresa.

3.1.5 Herramientas Computacionales.

Las fuentes de información nombradas anteriormente aunque importantes, no llenan el requisito indispensable de una “Base de Conocimiento”, es decir *no poseen suficientes ejemplos por cada falla*. (Figura 4).

Figura 4. Necesidad de Ejemplos.

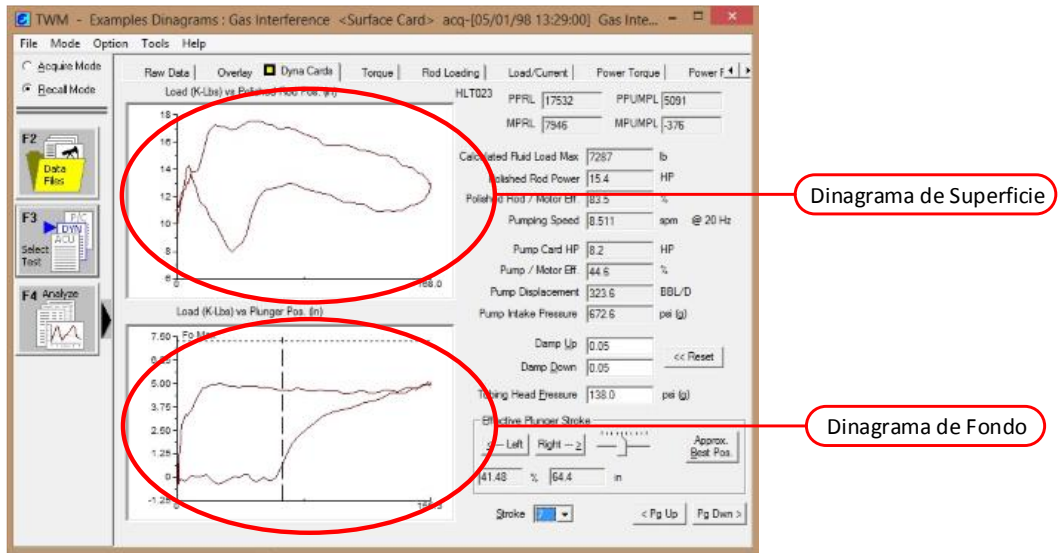


Por esta necesidad, se recurrió a partir de los ejemplos encontrados tipificar varios modelos de falla con la ayuda de herramientas computacionales. Estas fueron:

1. TWM (Total Well Management) Echometer, este software suministra suficiente información de dinagramas además de ilustrar su forma, (ver figura 5). Este se emplea con el simple objetivo de estudiar formas representativas de fallas.
2. TAM (Total Asset Monitor) Echometer, este software provee la misma información del TWM, además de simular gráficamente la falla que presenta la bomba (ver figura 6).

Figura 5. TWM (Total Well Management).

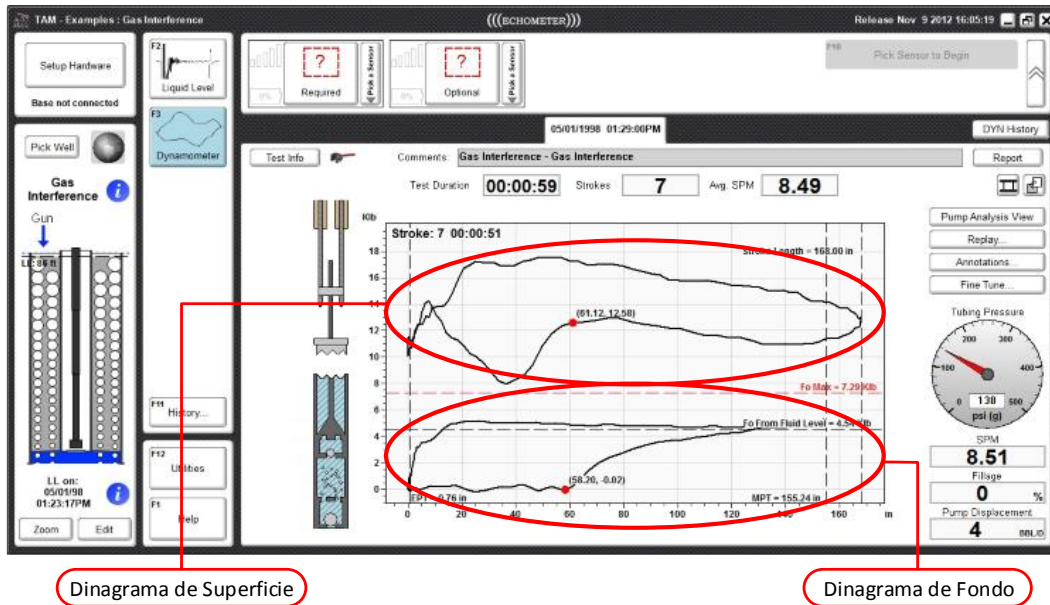
Interferencia de Gas



Fuente: Base de Datos TWM

Figura 6. TAM (Total Asset Monitor)

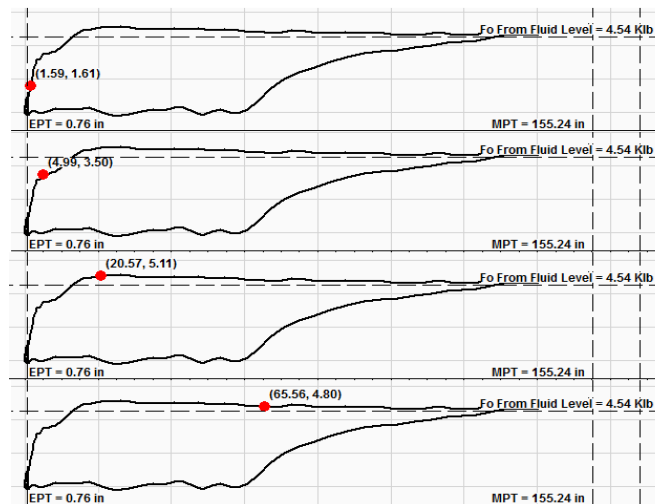
Interferencia de Gas



Fuente: Base de Datos TWM

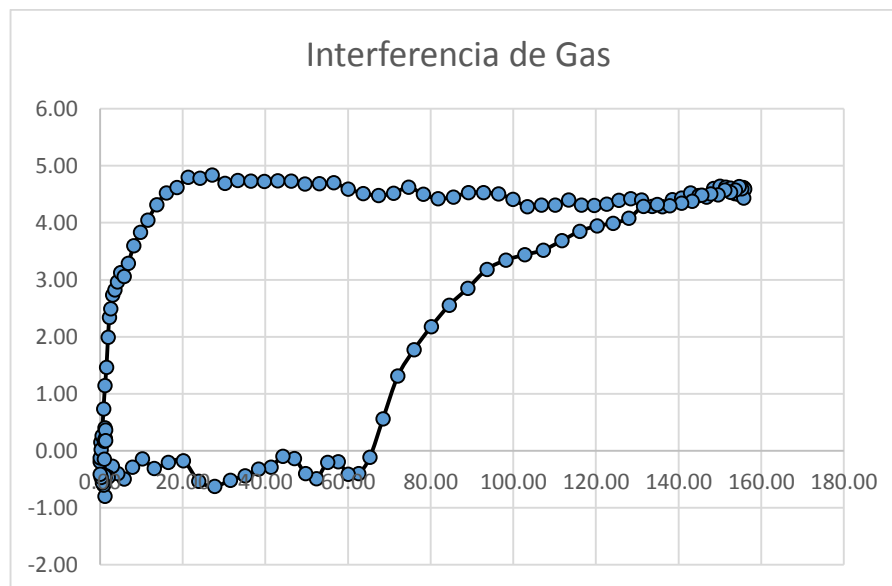
Como ventaja adicional el TAM permite recorrer el dinagrama punto a punto, como se muestra en la figura 7.

Figura 7. TAM, Recorrido punto a punto.



Asimismo, por esta utilidad se determina emplear el TAM como herramienta para almacenar en una hoja de Microsoft Excel punto a punto cada falla y simbolizarla como se observa en la figura 8, para el caso de interferencia de gas.

Figura 8. Dinagrama tipificado en Excel.



Posterior a almacenar los dinagramas base en hojas Excel, se procede a crear ruido variando aleatoriamente estos valores punto a punto, y así crear suficientes ejemplos para la base de conocimiento.

En total se crearon 220 dinagramas siguiendo la metodología descrita. Se tipificaron 10 fallas de mayor presencia en equipos de bombeo mecánico:

- Buen llenado tubería anclada.
- Interferencia de Gas.
- Rotura varillas.
- Agujero en el barril de la bomba.
- Golpe de Fluido.
- Fuga en Válvula Viajera o en Pistón.
- Fuga en Válvula Fija.
- Barril de la Bomba Doblado o Pegándose.
- Barril de la Bomba Gastado o Partido.
- Ancla de Tubería en Mal Funcionamiento.

3.2 PREPROCESAMIENTO (FASE 2).

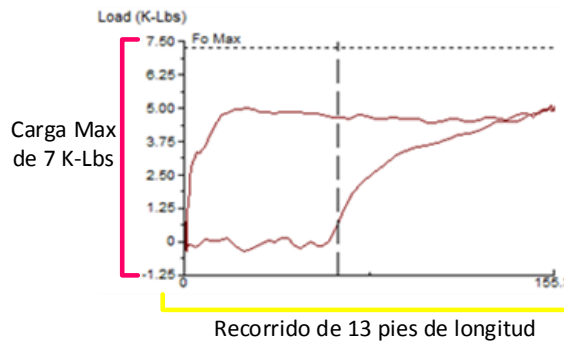
3.2.1 Normalización.

Los dinagramas que se obtienen directamente desde campos petroleros manejan distintos valores de recorrido y carga, debido a la influencia de variables como la profundidad del pozo, cargas máximas y mínimas en la barra pulida, desplazamiento de la bomba, etc. Un experto se basa en la forma del dinagrama para diagnosticar cualquier tipo de falla.

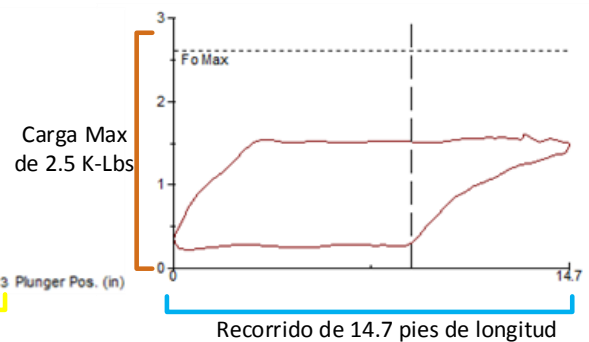
En operación, pozos diferentes (Pozo A – Pozo B) con fallas idénticas, por ejemplo “Interferencia de Gas”, generarán cartas dinagráficas con formas similares, aunque los valores de carga y recorrido sean diferentes, como lo muestra la figura 9.

Figura 9 Recorrido y Carga de dinagramas.

POZO A



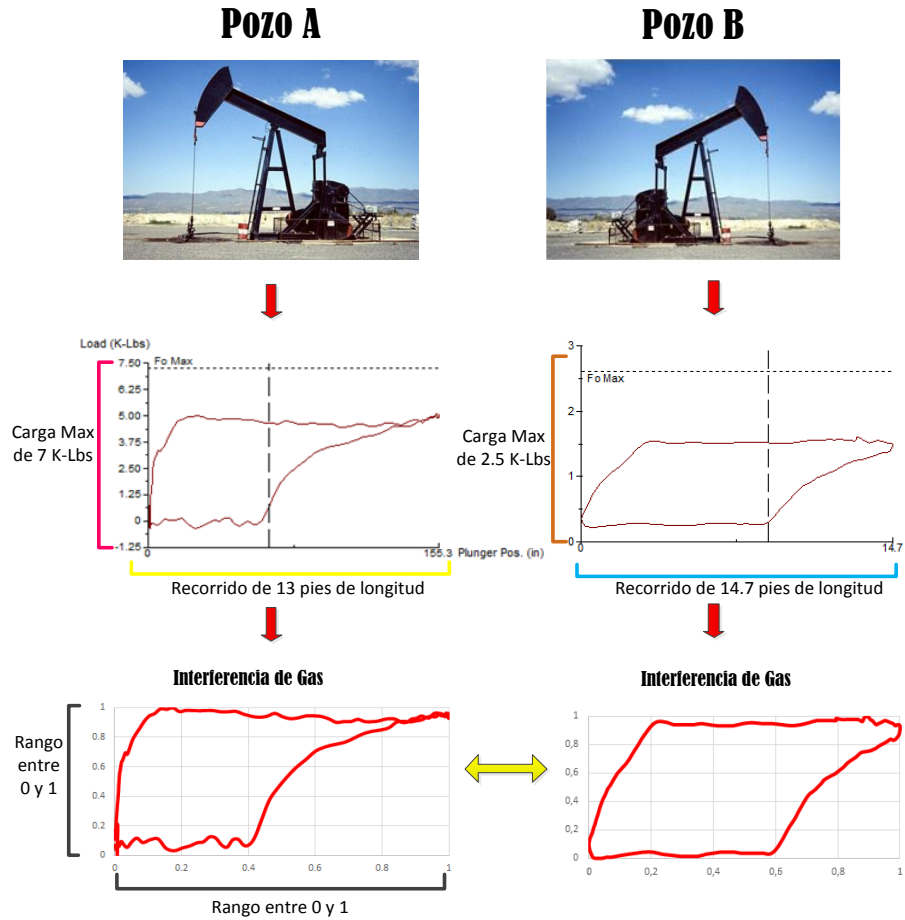
POZO B



El objetivo es obtener un modelo Fuzzy, capaz de diagnosticar cualquier falla en un pozo, independientemente de los valores específicos de carga y recorrido. Para lograrlo, los dinagramas deben ser normalizados, de tal manera que al ser presentados a la “Máquina de Aprendizaje”, se sintetice el conocimiento de los dinagramas presentados, a partir de su forma, obviando los valores específicos de carga y recorrido, pues en la normalización estos valores siempre estarán entre cero y uno, como se observa en la figura 10.

De no aplicar esta técnica, y si se deseara diagnosticar un campo petrolero, sería necesario entrenar una “Máquina de Aprendizaje” para cada pozo, además de asumir que las condiciones del mismo no varíen, lo cual no es cierto con solo analizar que, el nivel del pozo cambia con el tiempo, lo que produce cambios de carga en la barra pulida, y esto generará un dinagrama distinto en dimensiones. Por lo tanto, ya sea para un pozo o un campo petrolero siempre se deberá normalizar.

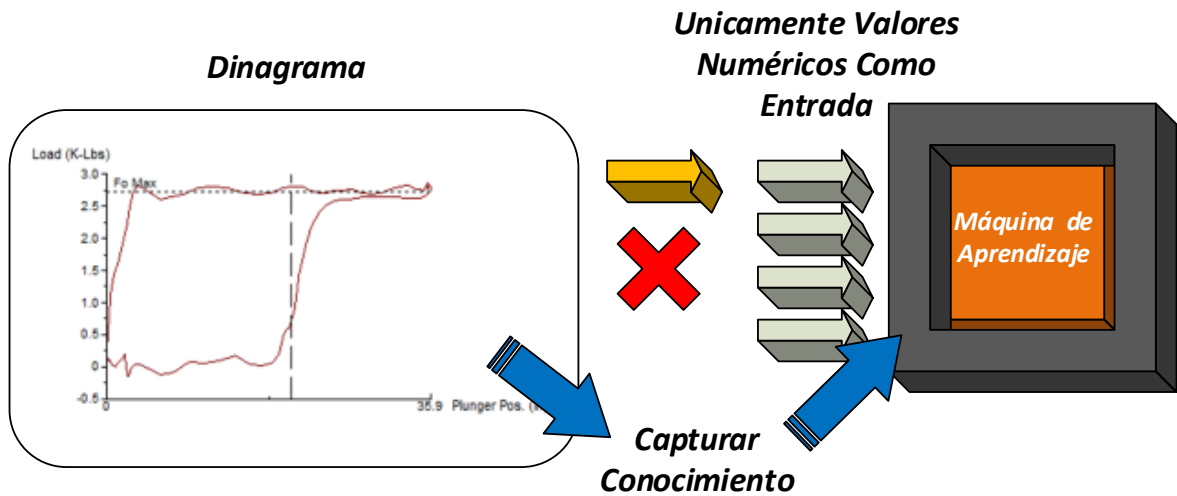
Figura 10 Dinagramas Normalizados.



3.2.2 Discretización.

Como ya ha sido expuesto, un dinagrama es una “carta gráfica”, y la Máquina de Aprendizaje empleada no acepta como entrada una gráfica, además es pertinente enfatizar la importancia de desarrollar una estrategia que permita capturar el conocimiento presente en un dinagrama, para luego ser presentado a la Máquina de Aprendizaje. (ver figura 11).

Figura 11. Objetivo de Discretización.

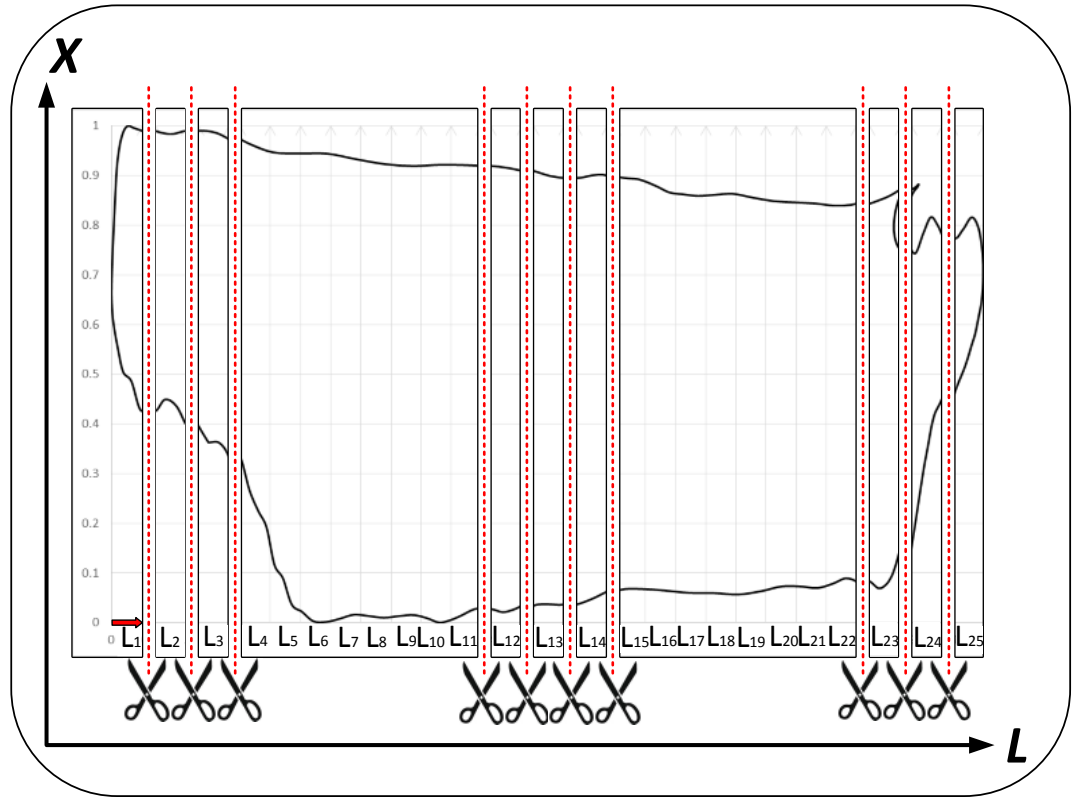


La estrategia para capturar el conocimiento presente en un dinagrama, consistió en DISCRETIZARLO. Se debe entender que “Discretizar” es la estrategia desarrollada en este proyecto, mediante la cual se transforma la gráfica de un dinagrama, en datos cuantitativos, es decir valores numéricos. Se estudiaron diferentes estrategias de como representar un dinagrama numéricamente.

Teniendo en cuenta que la Máquina de Aprendizaje solo permite cincuenta (50) valores de entrada, y que el dinagrama se presenta gráficamente en forma cerrada, la estrategia de discretización desarrollada consiste en:

1. Dividir el dinagrama normalizado en 25 partes iguales, definiendo como L_j ($j=1, j=2, j=3$ hasta $j=25$) a cada segmento de la división (ver figura 12), y a X_i ($i=1, i=2, i=3$ hasta $i=50$) como la amplitud del dinagrama, existente en cada segmento (ver figura 13).

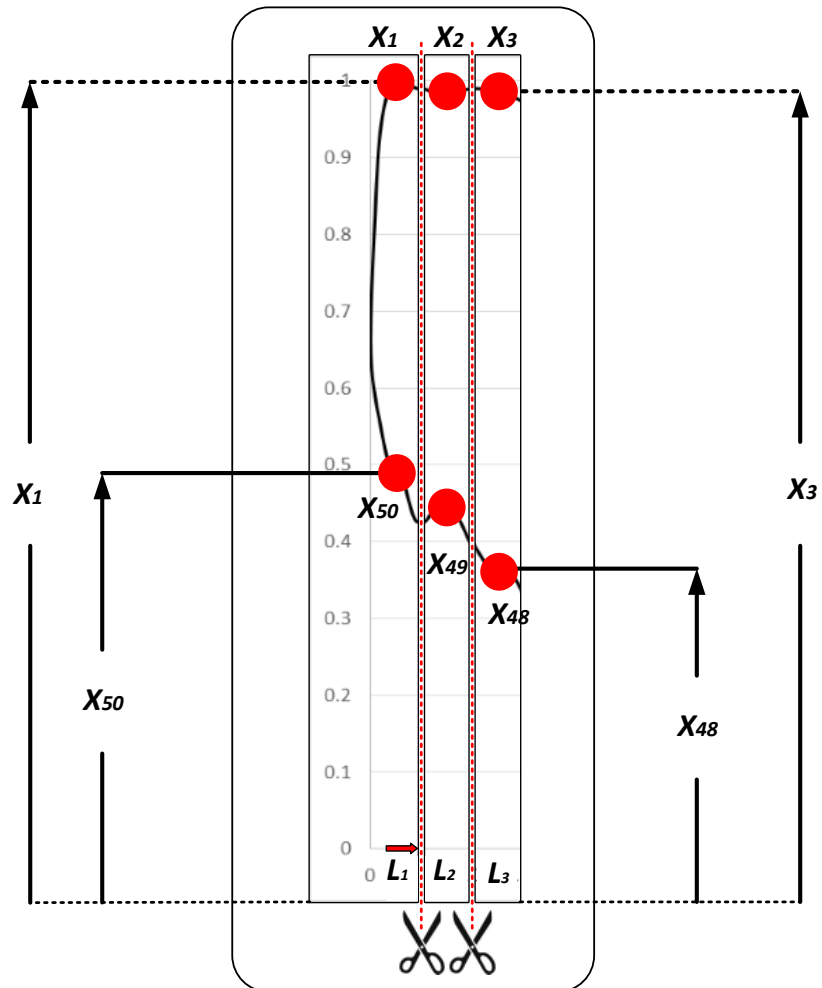
Figura 12. División de dinagrama.



Se evidencia en la figura 13 que por cada segmento (L_j), existen dos valores de amplitud (X). Uno de ellos corresponde a la carrera ascendente y el otro a la carrera descendente, respectivamente, de forma que:

- La amplitud X_1 y la amplitud X_{50} corresponden al segmento L_1 .
- La amplitud X_2 y la amplitud X_{49} corresponden al segmento L_2 .
- La amplitud X_3 y la amplitud X_{48} corresponden al segmento L_3 .
- ...
- La amplitud X_{25} y la amplitud X_{26} corresponden al segmento L_{25} .

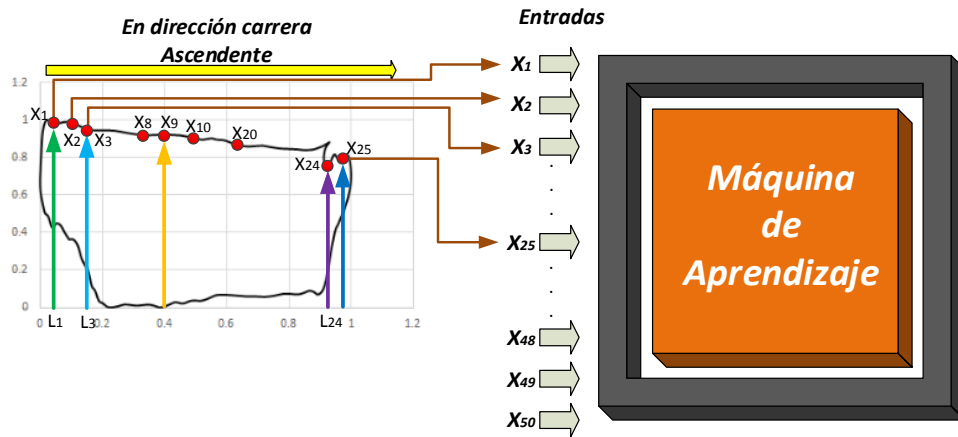
Figura 13. Amplitudes en cada segmento.



2. Tomar los 25 valores de amplitud (X), correspondientes a la carrera **ascendente**. Valores tomados en el dinagrama de izquierda a derecha de la siguiente manera, ver figura 14:

- X_1 para segmento L_1 .
- X_2 para segmento L_2 .
- X_3 para segmento L_3 .
- ...
- X_{25} para segmento L_{25} .

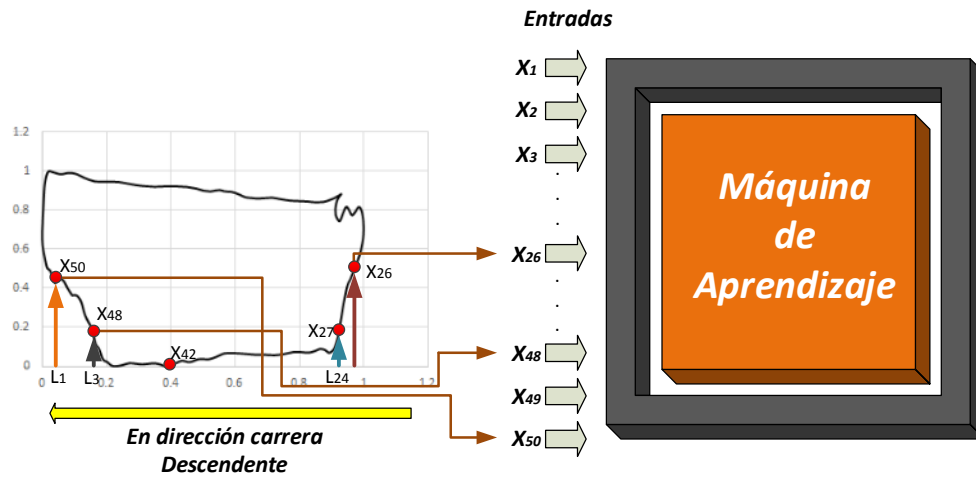
Figura 14. Amplitudes en carrera ascendente.



3. Tomar los 25 valores de amplitud (X), correspondientes a la carrera **descendente**. Valores tomados en el dinagrama de derecha a izquierda de la siguiente manera, ver figura 15.

- X_{26} para segmento L_{25} .
- X_{27} para segmento L_{24} .
- X_{28} para segmento L_{23} .
- ...
- X_{50} para segmento L_1 .

Figura 15. Amplitudes en Carrera Descendente.



3.3 MÁQUINA DE APRENDIZAJE (FASE 3).

La “Máquina de Aprendizaje” tiene como propósito generar un “Modelo Fuzzy” a partir de una serie de ejemplos (ver figura 16). Su funcionamiento está basado en el empleo de una arquitectura Neuro-Fuzzy denominada “*Perceptrón Fuzzy*”¹⁶.

NEFCLASS-Q es una herramienta software, creada en la UIS, en el laboratorio de Automatización Industrial, por los Ingenieros Fabio Caballero y Julián Salamanca dirigidos por el profesor Jorge Meneses. Esta herramienta Software utiliza un Perceptrón Fuzzy como algoritmo de aprendizaje para entrenar una Red Neuro Fuzzy, y es la “Máquina de Aprendizaje” empleada en este proyecto.

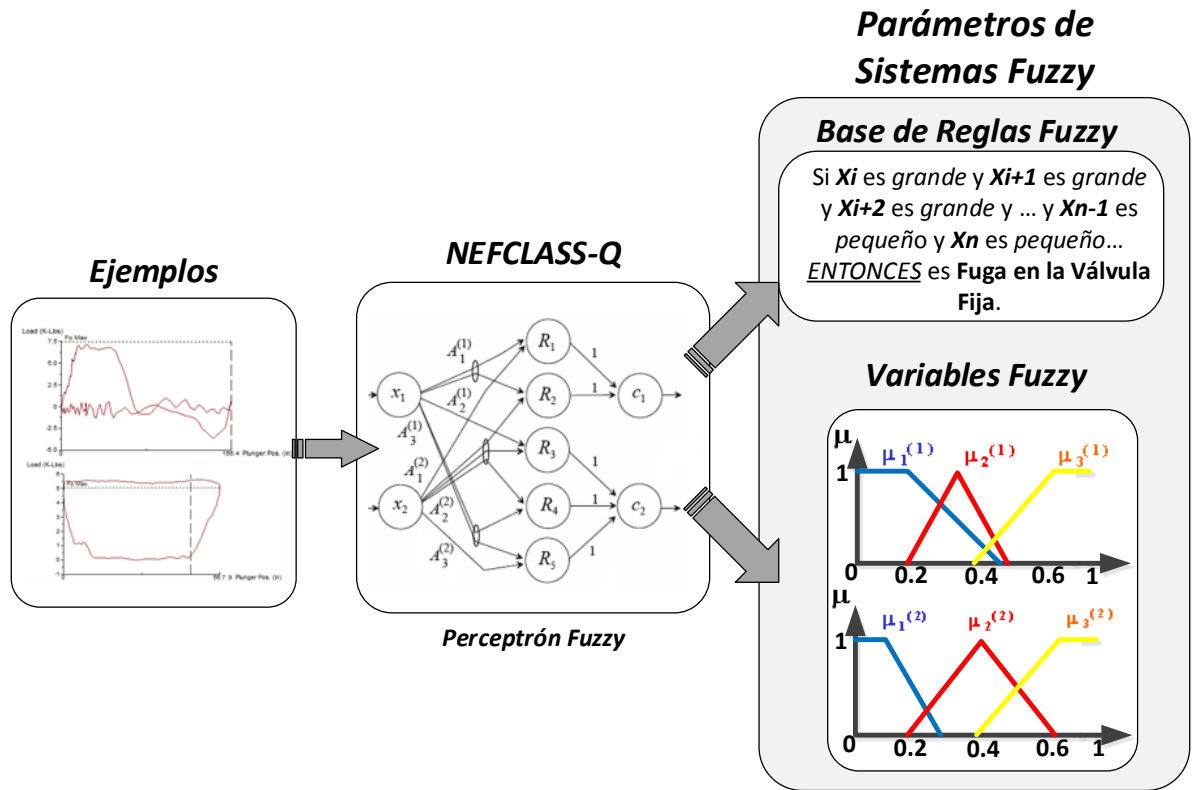
A continuación se presenta una breve descripción de los sistemas Neuro-Fuzzy y una explicación muy resumida de cada capa del Perceptrón Fuzzy.

3.3.1 Sistemas Neuro-Fuzzy.

Un Sistema Neuro-Fuzzy es una Máquina de Aprendizaje que encuentra los parámetros de un sistema Fuzzy (Reglas Fuzzy y Conjuntos Fuzzy, ver figura 16) por medio del uso de las técnicas de aproximación de las Redes Neuronales. Las Redes Neuronales Artificiales y los sistemas Fuzzy tienen características en común. Ambos pueden usarse para resolver problemas si no existe ningún modelo matemático del mismo. Por separado, presentan ventajas, y desventajas que casi desaparecen cuando se combinan ambos conceptos.

¹⁶ El Perceptrón Fuzzy de 3 capas es fruto de los estudios hechos por los Doctores Rudolf Kruse y Detleff Nauck de la Universidad de Magdeburg Alemania en el año 1994.

Figura 16. Sistema Neuro-Fuzzy



Las Redes Neuronales pueden operar si el problema es expresado por una cantidad suficiente de ejemplos, estos ejemplos son usados para entrenar *la caja negra*. Por un lado no se necesita conocimiento previo del problema, sin embargo, no es fácil extraer las reglas de esta estructura.

Por el contrario, un sistema Fuzzy demanda reglas lingüísticas en vez de ejemplos para el aprendizaje, como conocimiento previo. Además las variables de entrada y de salida deben ser descritas lingüísticamente. Si el conocimiento es incompleto, erróneo o contradictorio, el sistema Fuzzy debe ser sintonizado. Si no hay un método formal para esta sintonización, usualmente es realizada en forma heurística. Esto generalmente consume tiempo y propenso al error.

Tabla 1. Comparación de Control Neuronal y Control Fuzzy.

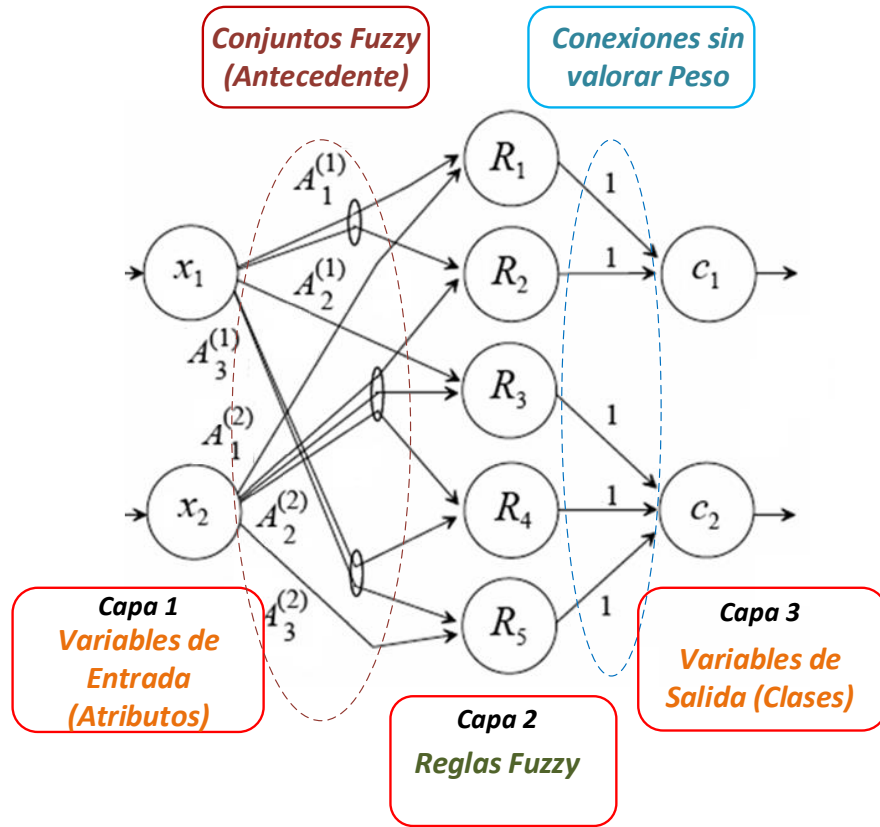
Redes Neuronales	Sistemas Fuzzy
No necesita un modelo matemático.	No necesita modelo matemático.
Aprendizaje a partir de ejemplos.	Conocimiento previo es esencial.
Varios algoritmos de aprendizaje.	No puede aprender.
Comportamiento de cada negra.	Interpretación e implementación simple

Es deseable para los sistemas Fuzzy tener un proceso adaptación automática el cual sea comparable con el proceso de las Redes Neuronales. Como se observa en la tabla 1, la combinación de ambas técnicas debe unir sus ventajas y excluir sus desventajas. La finalidad de los sistemas Neuro-Fuzzy es generar mecanismos de valoración que pueden ser interpretados en forma de reglas lingüísticas (SI-ENTONCES) y estar en capacidad de usar reglas basadas en el conocimiento previo, para que así el aprendizaje no tenga que empezar desde cero. Las ventajas de estos sistemas es que permiten acortar el tiempo de aprendizaje al poder incorporar “Conocimiento Previo” que posee un ser humano sobre una situación específica.

3.3.2 Perceptrón Fuzzy.

El Perceptrón Fuzzy es un modelo genérico de Redes Neuronales Fuzzy multicapas o un sistema Neuronal Fuzzy, respectivamente. Este modelo se sugiere para facilitar la comparación e interpretación en sistemas Neuro-Fuzzy. El Perceptrón Fuzzy fue usado para derivar el modelo NEFCLASS-Q empleado como Máquina de Aprendizaje (ver figura 17).

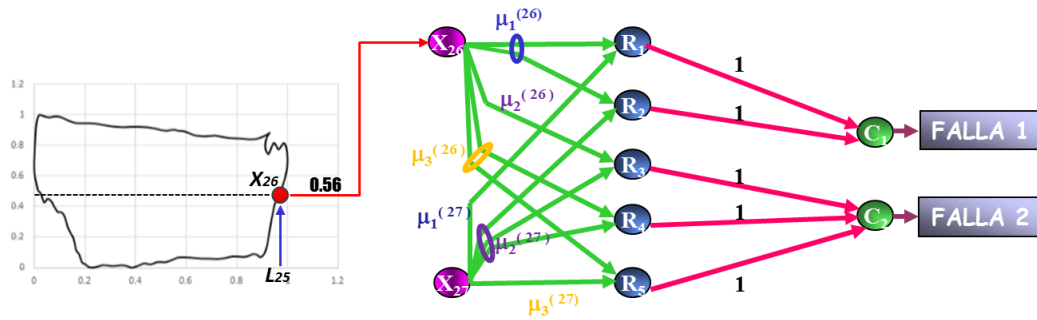
Figura 17. Arquitectura Neuro-Fuzzy → Perceptrón Fuzzy



3.3.2.1 Capa de entrada o capa 1.

Es la encargada, en este proyecto, de recibir cada unidad numérica de amplitud (X_i) de los dinagramas discretizados, ver figura 18.

Figura 18. Capa de Entrada.

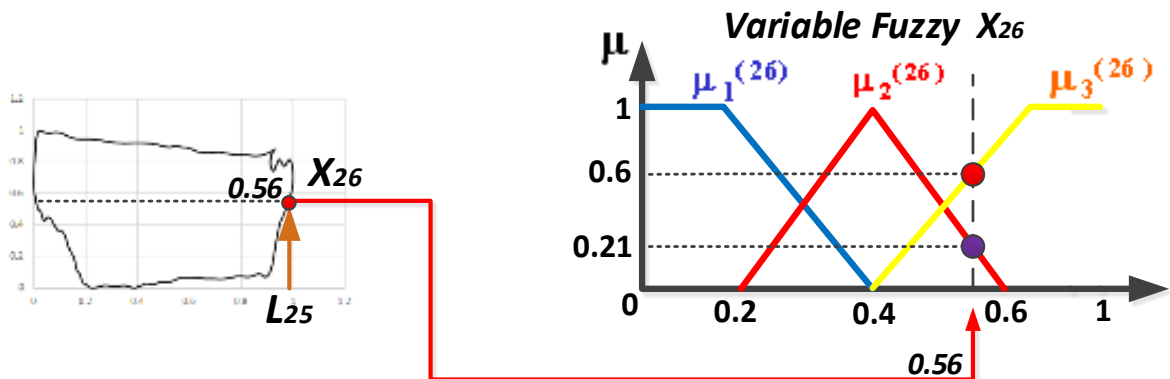


Entre la capa 1 y capa 2, se emplean conjuntos Fuzzy como pesos de conexión. Estos ponderan cada entrada numérica de amplitud del dinagrama, en valores lingüísticos catalogados mediante conjuntos Fuzzy: $\mu_1^i = \textit{Pequeña}$, $\mu_2^i = \textit{Mediana}$ y $\mu_3^i = \textit{Grande}$.

Por ejemplo (ver figura 19) si en un dinagrama, en la carrera descendente, la Variable Fuzzy X_{26} presenta un valor de 0.56 (en L_{25}), quiere decir que:

- la Variable Fuzzy X_{26} tiene un grado de pertenencia de **0** ($\mu_1^{26} = 0$) al conjunto Fuzzy *Pequeña*.
- la Variable Fuzzy X_{26} tiene un grado de pertenencia de **0.21** ($\mu_2^{26} = 0.21$) al conjunto Fuzzy *Mediana*.
- la Variable Fuzzy X_{26} tiene un grado de pertenencia de **0.6** ($\mu_3^{26} = 0.6$) al conjunto Fuzzy *Grande*.

Figura 19. Variable Fuzzy X_{26} .



La variable fuzzy X_{26} ..
 tiene una **pertenencia** de **0** al conjunto fuzzy *amplitud "Pequeña"*
 tiene una **pertenencia** de **0.21** al conjunto fuzzy *amplitud "Mediana"*
 tiene una **pertenencia** de **0.6** al conjunto fuzzy *amplitud "Grande"*

3.3.2.2 Capa de Reglas Fuzzy o capa 2.

Es la encargada de realizar las inferencias lingüísticas que se almacenan en forma de reglas del tipo SI-ENTONCES, conectando un antecedente con un consecuente.

La regla está compuesta por dos partes fundamentales:

- Un antecedente: Está formado por una serie de proposiciones relacionadas con operadores lógicos “Y” (\wedge), ver figura 20.

Figura 20. Antecedente de Regla.



- Un consecuente: Indica la clase a la que está asociada el antecedente (Tipo de falla), y esta correlacionado con su valor de verdad, ver figura 21.

Figura 21. Consecuente de la Regla.



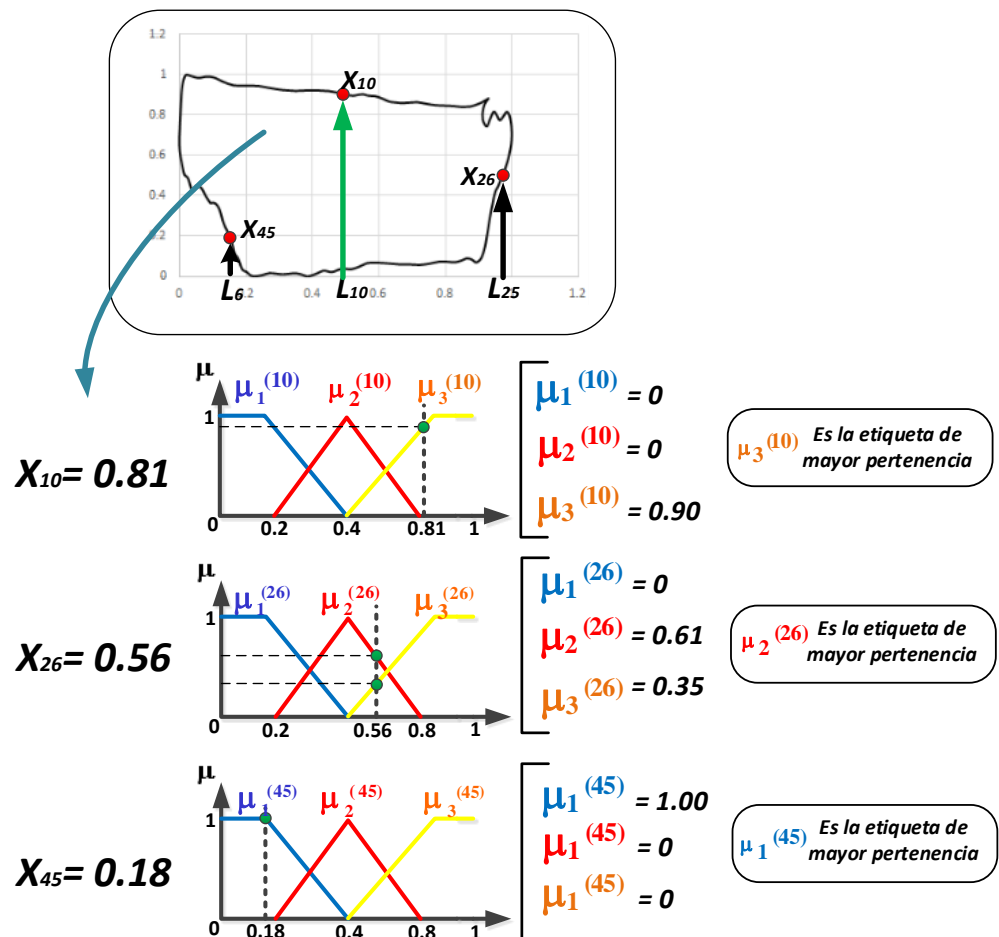
El algoritmo crea las reglas a partir de cada uno de los ejemplos del conjunto de Ejemplos de ENTRENAMIENTO (Base de Conocimiento).

Por ejemplo, asumiendo que el dinagrama mostrado en la figura 22, es el primer ejemplo del conjunto de Ejemplos de ENTRENAMIENTO, la regla 1 se formaría de la siguiente manera:

1. Las etiquetas lingüísticas (conjuntos Fuzzy) que conectan las variables Fuzzy X_{10} , X_{26} y X_{45} y la neurona de la capa oculta R_1 (Regla 1) sería:

- La Variable Fuzzy $X_{10} = 0.81$, tiene un grado de pertenencia de 0.90 ($\mu_3^{10} = 0.90$) al conjunto Fuzzy **Grande** y cero (0) a los conjuntos Fuzzy ($\mu_1^{10} = 0$) **Pequeña** y ($\mu_2^{10} = 0$) **Mediana**. La proposición sería entonces: X_{10} es **Grande**.

Figura 22. Asignación de etiqueta lingüística.



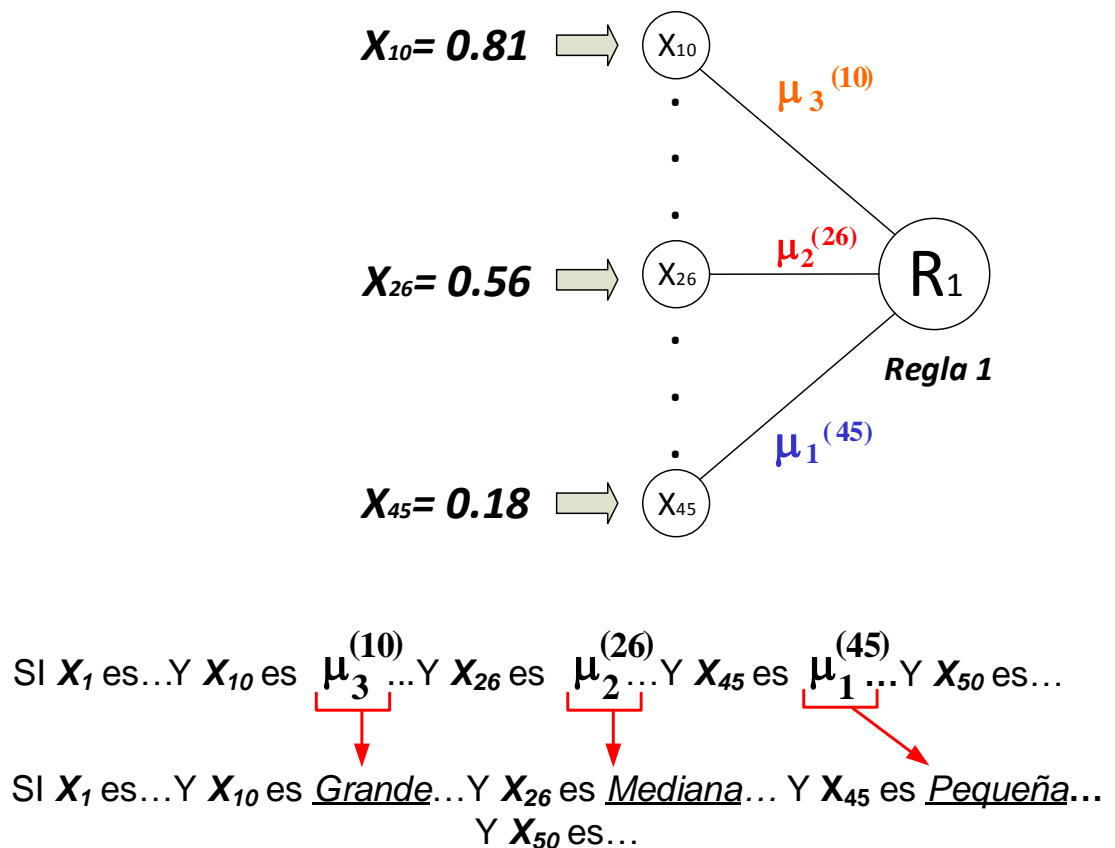
- La Variable Fuzzy $X_{26} = 0.56$, tiene un grado de pertenencia de 0.35 ($\mu_3^{26} = 0.35$) al conjunto Fuzzy **Grande**. Tiene también un grado de pertenencia de 0.61 ($\mu_2^{26} = 0.61$) al conjunto Fuzzy **Mediana** y cero (0) al conjunto Fuzzy ($\mu_1^{26} = 0$) **Pequeña**. Para esta variable Fuzzy, la etiqueta lingüística asignada a la regla, es la de mayor

pertenencia, es decir **Mediana** (μ_2^{26}). La proposición sería entonces: **X_{26} es Mediana.**

- La Variable Fuzzy **X_{45}** tiene un grado de pertenencia de 1 ($\mu_1^{45} = 1$) al conjunto Fuzzy **Pequeña** y cero (0) a los conjuntos Fuzzy ($\mu_2^{45} = 0$) **Mediana** y ($\mu_3^{45} = 0$) **Grande**. La proposición sería entonces: **X_{45} es Pequeña.**

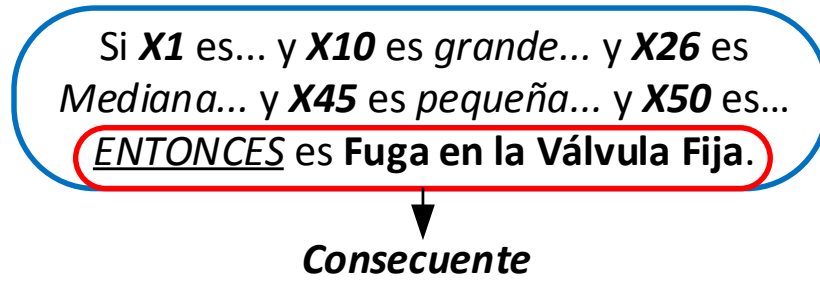
2. Según el ejemplo y posterior a la obtención de cada proposición, se procede a crear el antecedente de la regla 1 uniendo las proposiciones con operadores lógicos. (ver figura 23).

Figura 23. Antecedente de la Regla 1.



- Si no existe un antecedente igual, se define el consecuente de la regla con la clase de falla propagada (respuesta esperada), esto para evitar que se creen dos reglas iguales. Siguiendo el ejemplo, la regla 1 dictaría, ver figura 24:

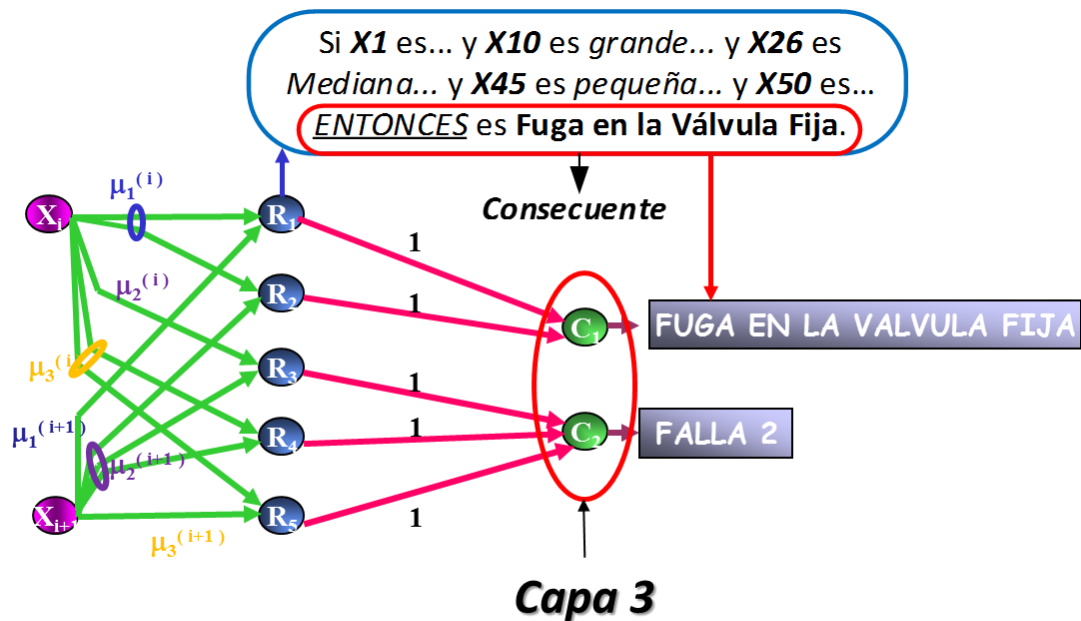
Figura 24. Regla 1.



3.3.2.3 Capa de salida o capa 3.

Es la capa que representa las diferentes opciones a la que puede pertenecer un dinagrama, y se encarga de hacer el proceso de cálculo del error. (ver figura 25)

Figura 25. Capa 3 o capa de salida.



Cada neurona de la capa 2 o capa de reglas es conectada solamente a una unidad de salida o clase. Para el caso de este proyecto la capa de salida presentó

diez (10) neuronas (Unidades de Salida), las cuales representan los diez estados o fallas que se estipularon diagnosticar.

3.3.3 NEFCLASS-Q

NEFCLASS-Q es un software desarrollado en el Laboratorio de Automatización Industrial por los Ingenieros Fabio Caballero y Julián Salamanca bajo la dirección del profesor Jorge Meneses. Esta herramienta está basada en la arquitectura Neuro-Fuzzy anteriormente descrita (Perceptrón Fuzzy) y es una aplicación similar al NEFCLASS¹⁷.

3.3.3.1 Requerimientos para el empleo del NEFCLASS-Q.

- Sistema operativo: Linux (UBUNTU).
- Compilador QT creator.
- Librerías Network, FuzzySet, Universe, Files, RuleNeuron.

Los procedimientos y requerimientos que emplea el NEFCLASS-Q de entrenamiento, validación y clasificación serán explicados en el capítulo 6.

3.4 MODELO FUZZY PARA DIAGNÓSTICO (FASE 4).

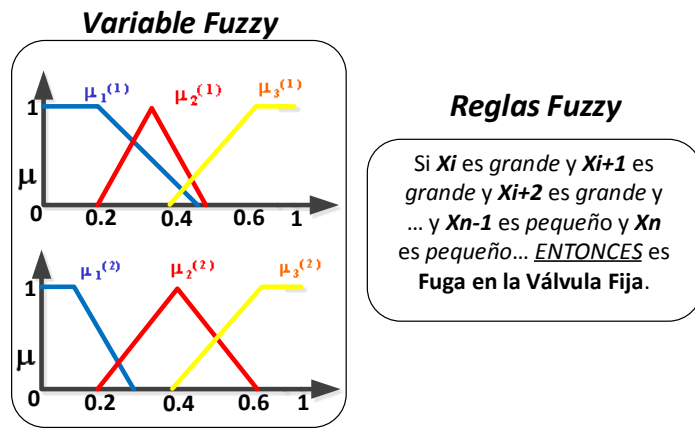
El objetivo de realizar cada fase anterior es obtener un modelo Fuzzy (ver figura 26), el cual sintetiza el conocimiento en:

1. Reglas del tipo (SI – ENTONCES).
2. Variables y conjuntos Fuzzy.

El modelo Fuzzy será implementado en posteriores proyectos, en un sistema embebido que tendrá la capacidad de diagnosticar automáticamente dinagramas de fondo sin la intervención de un experto humano.

¹⁷ Software desarrollado por Rudolf Kruse y Detlef Nauck basándose en la arquitectura del Perceptrón fuzzy simple elaborada por los mismos autores.

Figura 26. Modelo Fuzzy



4. OBTENCIÓN DE LA BASE DE CONOCIMIENTO

Para obtener una base de conocimiento consistente, se realizó como primer paso, un estudio de los fenómenos dinagráficos que se presentan en equipos de bombeo mecánico. El enfoque de este estudio apunta a la forma representativa de los dinagramas, más que las causas por las que se presenta cada falla. A continuación se presenta un breve resumen de lo estudiado, empezando por una descripción general de un dinagrama, hasta los empleados para la ejecución de este proyecto. Siguiendo a esto, se describe el aporte generado por Campo Escuela Colorado y por los expertos.

4.1 DESCRIPCIÓN GENERAL.

El dinagrama es un instrumento indispensable cuando se utiliza bombeo mecánico como levantamiento artificial. Este registra y mide de forma continua las cargas y las deformaciones que soporta el vástago pulido y como resultado muestra las cartas dinamométricas correspondientes que deben ser interpretadas para determinar los problemas que existen en el fondo del pozo y dar las soluciones pertinentes. El gráfico final representa la Carga vs Posición y se obtiene conociendo la carrera correspondiente del vástago del balancín.

Los primeros dinamómetros fueron utilizados a principios de los años veinte, desde entonces, tanto el hardware como los métodos de evaluación han mejorado considerablemente haciendo que la medición simultánea de diversos parámetros dinámicos (kilovatios de entrada, factor de potencia, torque del motor, torque de la caja reductora, velocidad, aceleración, carga, etc.) sea económicamente posible, es así que las primeras y más importantes interpretaciones cualitativas, se han convertido en métodos de análisis exactos, sofisticados y altamente confiables. Hoy en día se podría decir que el método de levantamiento artificial más utilizado es el bombeo mecánico y por tal razón es importante el estudio del dinagrama, pues es un elemento del cual se puede inferir el desempeño de un sistema de bombeo mecánico y que genera una gráfica en la que se puede visualizar la carga de la varilla lisa en función de la posición. Asimismo, tal gráfica se puede

interpretar como el trabajo hecho en superficie por la unidad de bombeo mecánico por “stroke”.

4.1.1 Tipos de dinagramas.

Existen dos tipos de dinagramas, dinagrama de superficie y dinagrama de fondo, siendo este último el más adecuado y más fácil de interpretar ya que su forma depende únicamente de lo que esté pasando en la bomba. (ver figura 27).

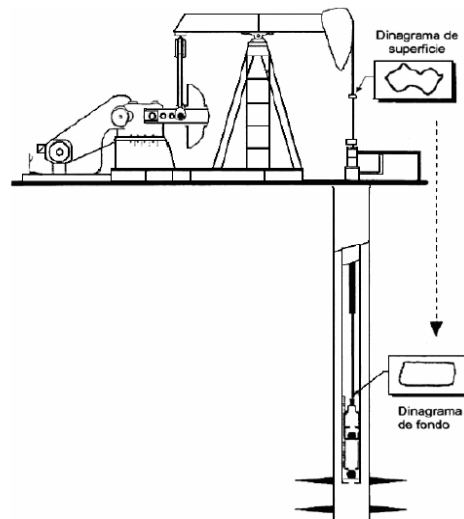
4.1.1.1 Dinagrama de superficie.

El dinagrama de superficie es un registro de cargas, sobre la barra pulida para determinar en superficie el comportamiento de la bomba durante cada ciclo de bombeo.

4.1.1.2 Dinagrama de fondo.

Para entender la forma de un dinagrama de fondo es necesario comprender las condiciones físicas en las que está operando el barril, el pistón y las válvulas en el ciclo de bombeo.

Figura 27. Dinagrama de Superficie y Fondo



Fuente: NAVARRETE, Juan Carlos. Torres, Juan Efe. Rediseño del levantamiento por bombeo mecánico mediante el análisis de dinagramas.

Un Downhole card o dinagrama de fondo es un dinagrama de Carga vs Posición de la primera varilla por encima de la bomba y es obtenido por medio de la ecuación de onda mediante la cual se construye la carta de fondo de la bomba y se determina la carga sobre los tramos de varilla.

Algunas de las diferencias de estos dos tipos de dinagramas radican en que el dinagrama de superficie depende de factores como:

- Profundidad de la bomba.
- Diseño y material de la sarta de varillas.
- Velocidad de bombeo.
- Tipo de unidad de bombeo.
- Tipo de motor.
- Condiciones de operación de la bomba.

En cambio el análisis y la forma de un dinagrama de fondo solo dependen de las condiciones de operación de la bomba.

Esto hace mucho más fácil el análisis cualitativo de la calidad o eficiencia de operación y se puede determinar cuáles son los problemas que disminuyen la producción. En la actualidad se cuenta con Dinamómetros digitales con software que calculan un dinagrama de fondo a partir de un dinagrama de superficie.

4.2 INTERPRETACIÓN DE DINAGRAMAS

Se ha determinado que la toma de las cartas dinamométricas, generan utilidad para detectar los problemas que existen en el campo cuando el sistema de bombeo no realiza trabajo. La interpretación consiste en un análisis cualitativo de la carta donde guiados por su forma se detectan fallas además de emplearse como soporte para encontrar las posibles soluciones. Por lo tanto, cualquier variación en su recorrido ascendente o descendente, informara su estado de operación.

Existe una serie de pasos que permiten realizar un diagnóstico eficiente al pozo que se estudia complementándose en la mayoría de los casos con la medición de nivel usando ecómetro.

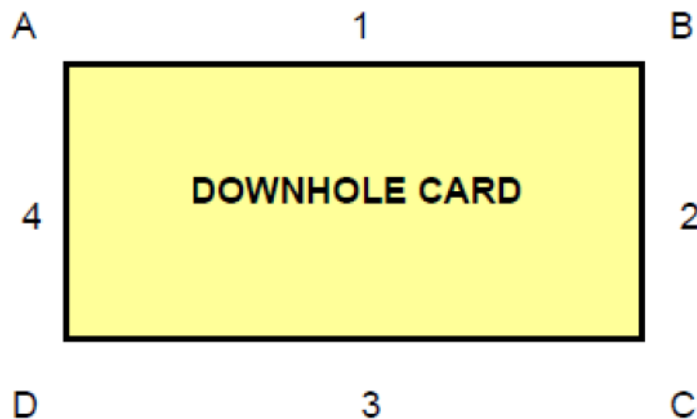
4.2.1 Proceso para identificar las partes de un Downhole card y sus problemas.

Para estar en capacidad de realizar un análisis cualitativo a un dinagrama de fondo o Downhole card existen dos maneras posibles.

1. Aprenderse todas las formas de los problemas que puede presentar la bomba de subsuelo.
2. Saber identificar las partes de un Downhole card y los problemas que pueden cambiar su forma.

Para la segunda opción se debe comprender las partes que conforman la carta (ver figura 28):






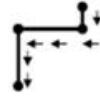








Figura 28. Partes del dinagrama.



1. Carrera ascendente. Tramo 1 de A – B.
2. Fin de la carrera ascendente e inicio de la carrera descendente. Tramo 2 de B – C.
3. Carrera descendente. Tramo 3 de C – D
4. Fin de la carrera descendente e inicio de la carrera ascendente. Tramo 4 de D-A.

En la tabla 2 se puede observar, las formas que pueden presentar los tramos anteriores (1, 2, 3 y 4 de la figura 28), y sus posibles causas.

Tabla 2. Caracterización de falla por zona.

EFICIENTE	DEFICIENTE	CAUSA
		Fuga en la válvula viajera, fricción excesiva o en algunos casos es muestra de fuga en la tubería de producción. Tramo A-B
		Un golpe o fricción que aumenta o disminuye la carga sobre el pistón. Tramo A-B
		Golpe de fluido o barril deficientemente lleno de líquido. Tramo B-C
		Golpe de gas o barril deficientemente lleno de líquido. Tramo B-C
		Fuga en la válvula fija o fricción excesiva. Tramo C-D
		Golpe o fricción en el pistón. Tramo C-D
		Fuga en la válvula viajera causada por arena o fluidos emulsificados. Tramo D-A

Fuente: MATEUS, Cristian y ROJAS Daniel. Análisis y diagnóstico de problemas en el equipo de subsuelo de unidades de bombeo mecánico utilizando redes neuronales. UIS. 2007.

4.3 DINAGRAMAS EMPLEADOS.

Para la presente tesis se trabajaron los dinagramas de mayor presencia o más conocidos en sistemas de bombeo mecánico. En total se manejaron diez de todos ellos;

- Buen llenado tubería anclada.

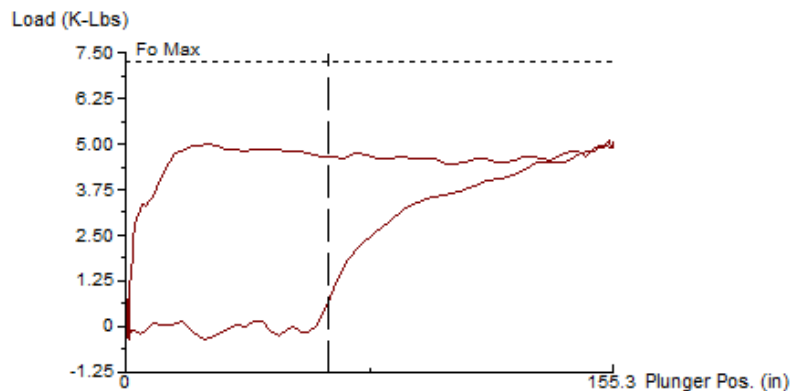
- Interferencia de Gas.
- Rotura varillas.
- Agujero en el barril de la bomba.
- Golpe de Fluido.
- Fuga en Válvula Viajera o en Pistón.
- Fuga en Válvula Fija.
- Barril de la Bomba Doblado o Pegándose.
- Barril de la Bomba Gastado o Partido.
- Ancla de Tubería en Mal Funcionamiento.

A continuación una breve explicación de cada uno y sus formas características.

4.3.1 Interferencia de Gas.

Fenómeno que se produce cuando el gas ingresa a la bomba de varilla de succión del subsuelo, ver figura 29.

Figura 29. Interferencia de Gas.



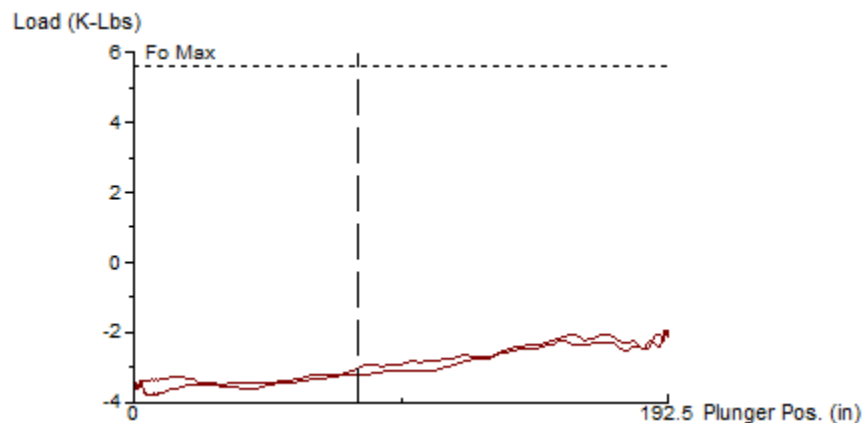
Después de que comienza la carrera descendente, el gas comprimido alcanza la presión necesaria para abrir la válvula viajera antes de que la válvula viajera llegue al líquido. La válvula viajera se abre lentamente, sin el cambio de carga drástico experimentado en los golpes de fluido. No causa fallas prematuras en el equipo pero puede indicar poca eficiencia de la bomba. Un separador de fondo de pozo o un ancla de gas pueden corregir una interferencia de gas.

4.3.2 Rotura varillas.

Puede ocurrir por 2 factores, ver figura 30:

1. Fallas Mecánicas:
 - Fallas por Tensión (Tracción).
 - Fallas por Fatiga.
2. Fallas por Corrosión:
 - Corrosión Uniforme o General.
 - Corrosión Localizada.
 - Cracking por Tensión – Corrosión.
 - Desgaste – Corrosión.
 - Corrosión Galvánica.

Figura 30. Rotura varillas.

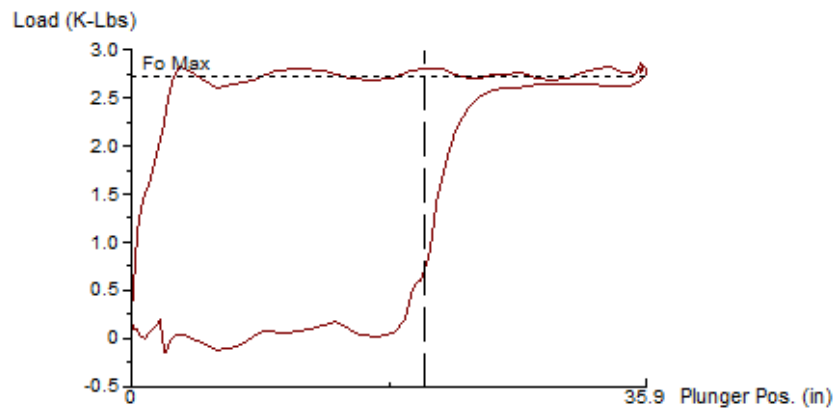


4.3.3 Golpe de Fluido.

Fenómeno que se produce cuando la tasa de la bomba de fondo de pozo supera la tasa de producción de la formación. También se puede deber a la acumulación de gas a baja presión entre las válvulas. En la carrera descendente de la bomba, se comprime el gas, pero la presión dentro del barril no abre la válvula viajera hasta que ésta golpea el líquido. Finalmente, cuando la válvula viajera se abre, el peso en la sarta de varillas puede dejar caer miles de libras en una fracción de segundo (ver figura 31). Se debe evitar esta condición porque causa grandes

esfuerzos, lo que podría generar fallas prematuras en el equipo. Una reducción de velocidad de la unidad de bombeo, el acortamiento de la longitud de la carrera o la instalación de una bomba de fondo de pozo más pequeña puede resolver este problema.

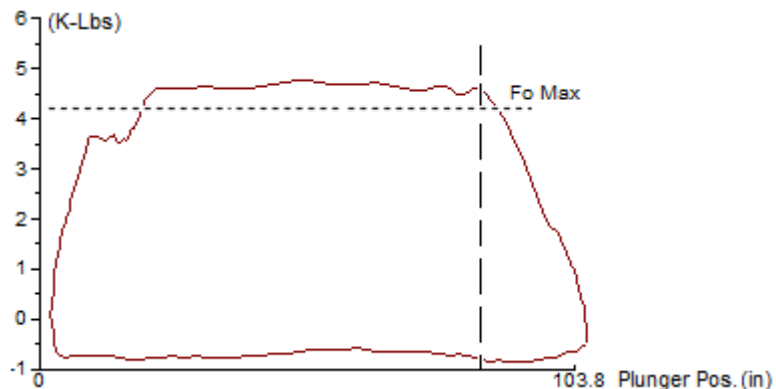
Figura 31. Golpe de Fluido.



4.3.4 Fuga en Válvula Viajera o en Pistón.

Una fuga por la válvula viajera o por el pistón son problemas muy comunes. La principal característica de la forma de esta carta dinográfica es lo redondeado en la mitad superior de la carta.

Figura 32. Fuga en Válvula Viajera o en Pistón.



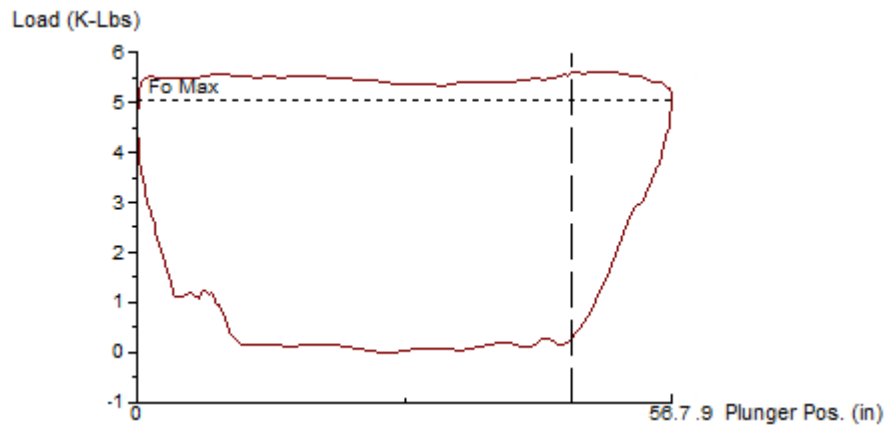
Esto sucede porque la válvula viajera o el pistón no pueden tomar completamente la carga del fluido como sucede en una carta de bomba llena. Al iniciar el pistón su carrera ascendente, toma lentamente la carga del fluido. Pero, debido a que el

fluido se está fugando hacia el barril de la bomba, la presión en el barril de la bomba no cae lo suficientemente rápido para que el pistón recoja la carga completa del fluido, ver figura 32.

4.3.5 Fuga en Válvula Fija.

Como muestra el figura 33, la forma de una carta dinagráficas de fondo correspondiente a fuga en la válvula fija es una imagen de espejo de la forma de una carta para fuga en la válvula viajera o pistón. Para comprender la forma de una fuga por válvula fija se debe recordar que una carta dinagráficas de fondo representa los cambios de carga contra posición justo sobre el pistón. En consecuencia, la forma de una fuga a través de la válvula fija muestra el efecto que tiene sobre la carga del pistón una fuga en la válvula fija.

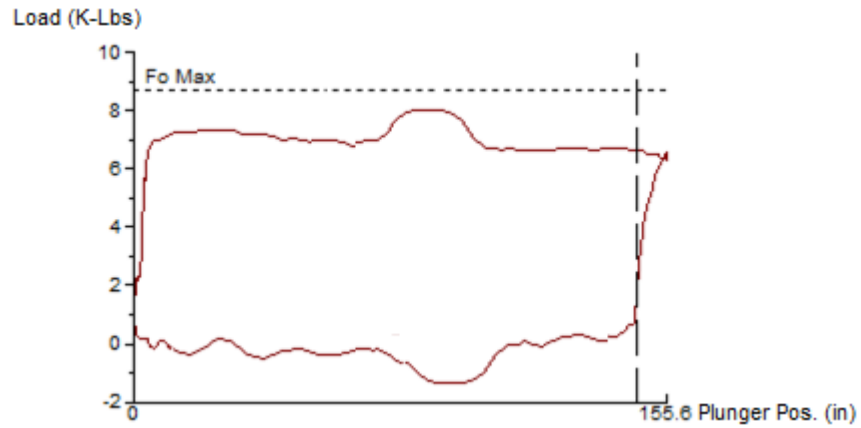
Figura 33. Fuga en Válvula Fija.



4.3.6 Barril de la Bomba Doblado o Pegándose.

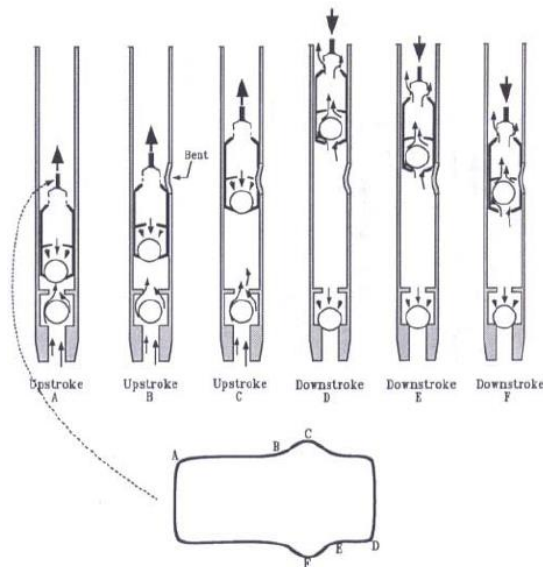
La figura 34 ayuda a explicar la forma para un barril de bomba doblado o pegándose. En un punto el pistón está por debajo de la sección doblada y la carga sobre el pistón es la misma que para bomba llena.

Figura 34. Barril de la Bomba Doblado o Pegándose.



Cuando el pistón alcanza la dobles en el punto B (ver figura 35), la carga sobre el pistón aumenta ya que el pistón debe escurrirse por esta porción del barril de la bomba. La carga del pistón llega al máximo en el punto C y luego baja mientras el pistón se aleja de la dobles.

Figura 35. Barril Doblado.



Fuente: THETA, Enterprises. Bombeo Mecánica Optimización.

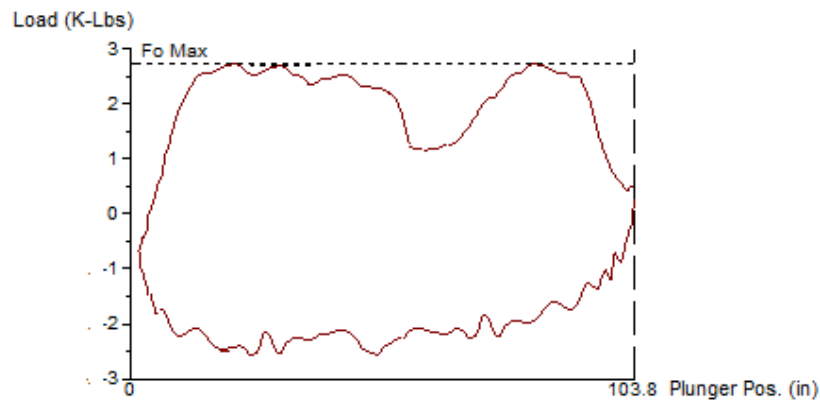
En la carrera descendente, la carga sobre el pistón es normal hasta que el pistón llega al punto malo en el barril en el punto E. Luego de este punto, la carga del

pistón decrece hasta el punto F, y luego regresa a la normalidad después que el pistón se aleja de la porción doblada del barril de la bomba.

4.3.7 Barril de la Bomba Gastado o Partido.

Cuando el barril de la bomba está gastado o partido, la carta dinagráfica de fondo puede tener una forma como la de figura 36, desde el principio de la carrera ascendente en el punto A hasta el punto gastado en el barril, las cargas son normales (ver figura 37). Cuando el pistón atraviesa la porción gastada del barril en C, una fuga de líquido pasa el pistón causando una caída de carga hasta que el pistón pasa la sección gastada y se establece de nuevo un buen sello entre el pistón y el barril.

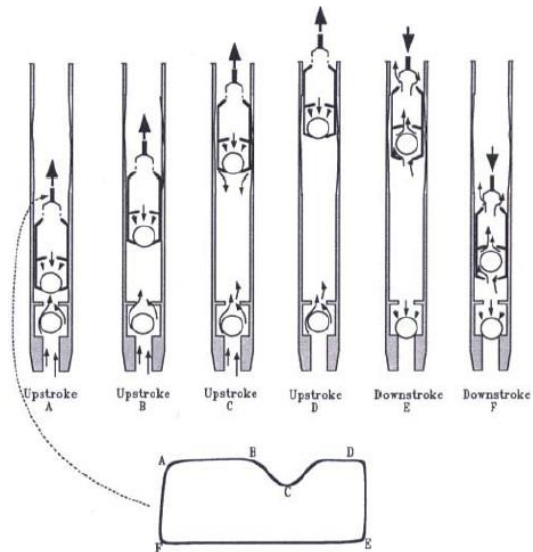
Figura 36. Barril de la Bomba Gastado o Partido.



En la carrera descendente, ya que la válvula viajera permanece abierta, la carga del pistón parece normal.

Sin embargo, si el barril de la bomba está gastado, se puede ver un aumento de carga correspondiente al mismo punto. Esto puede suceder si la sección gastada causa suficiente pérdida de presión en el barril de la bomba para que la válvula viajera comience a tomar la carga del fluido.

Figura 37. Barril Gastado.

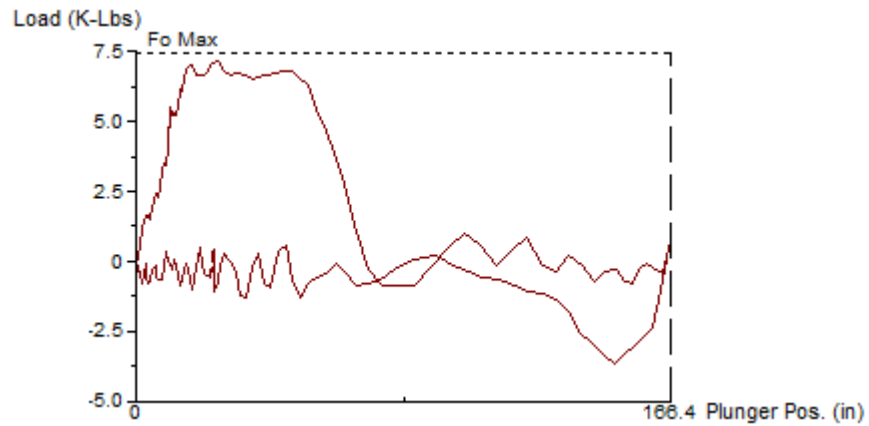


Fuente: THETA, Enterprises. Bombeo Mecánica Optimización.

4.3.8 Agujero en el barril de la bomba.

Cuando el barril de la bomba tiene un agujero, la carta dinagráfica de fondo puede tener una forma como la de figura 38, en el primer tramo de carrera ascendente su comportamiento es normal en apariencia, al llegar a 1/3 de su recorrido la carga sobre la barra pulida cae drásticamente debido a la pérdida de presión ocasionada por el agujero. En carrera descendente luego que la válvula fija está cerrada su comportamiento es apariencia normal, solo exceptuando ocasiones en las que se presentan bajonazos de carga repentinos.

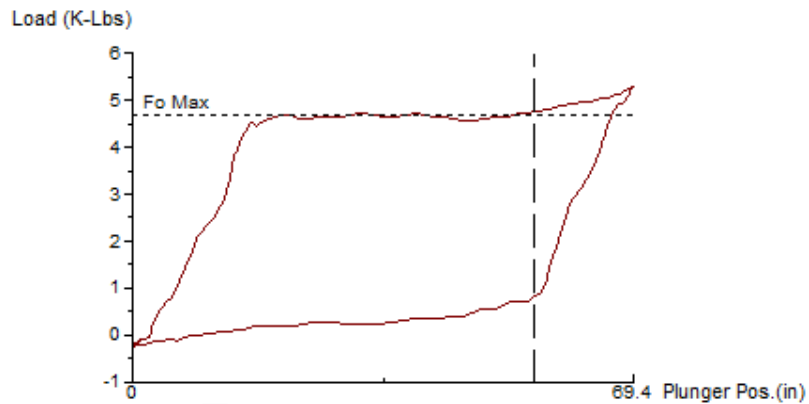
Figura 38. Agujero en el barril de la bomba.



4.3.9 Buen llenado tubería anclada.

La carta Dinagráfica mostrada en la figura 39, no es textualmente una falla. Esta se presenta cuando la bomba está en buenas condiciones de trabajo. Esta carta se muestra de forma inclinada por ser extraída de una bomba donde la tubería está anclada.

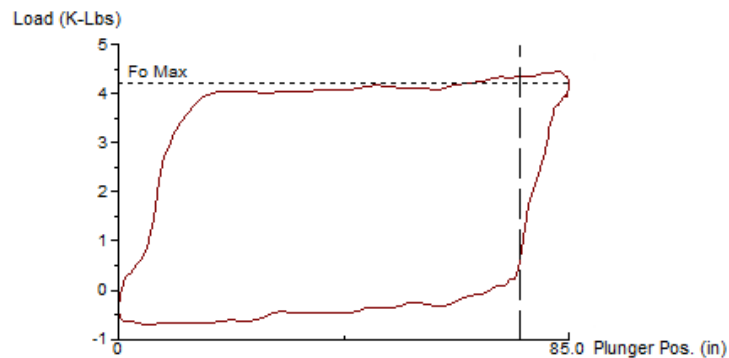
Figura 39. Buen llenado Tubería Anclada



4.3.10 Ancla de Tubería en Mal Funcionamiento.

Cuando el ancla de la tubería está en mal funcionamiento, la carta dinagráfica de fondo puede tener la forma de la figura 40, en general, se presenta con dos picos horizontes pronunciados en el inicio del ciclo de bombeo y en el final de la carrera ascendente.

Figura 40. Ancla de Tubería Mal Funcionamiento



4.4 CAMPO ESCUELA COLORADO (CEC).

Posterior al estudio teórico realizado, y al no contar con suficientes ejemplos, se imparte en la búsqueda de otras fuentes, queriendo encontrar en lo posible cartas dinagráficas reales de pozos en operación.

Por esto, la segunda fuente de información consultada fue Campo Escuela Colorado, donde no se tuvo éxito encontrando ejemplos de fallas específicas. Contaban con dinagramas obtenidos únicamente de dos pozos, el C-03 y el C-07, además de ser dinagramas con fallas combinadas. Estos dinagramas fueron:

Figura 41. Golpe de Fluido con problemas en válvulas (C-03).

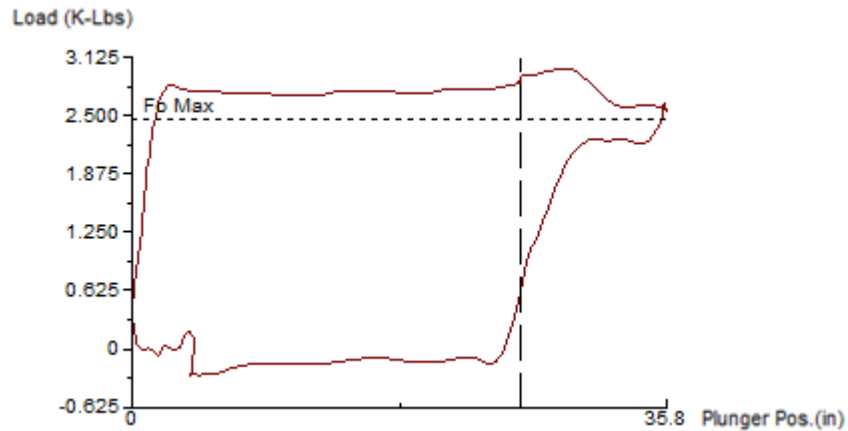
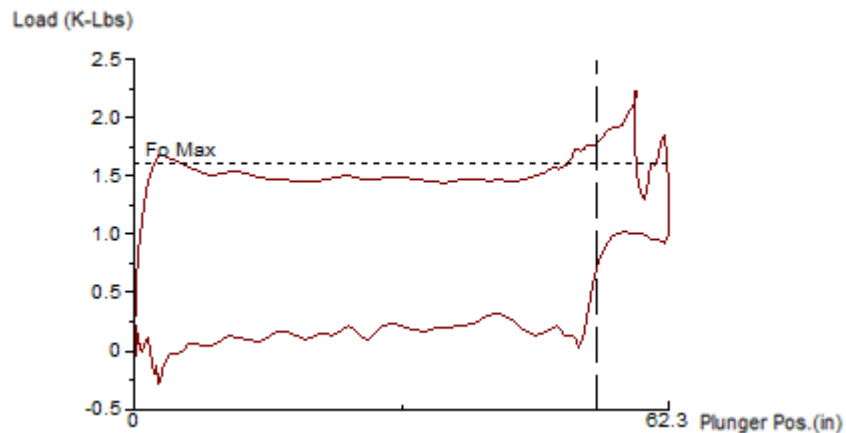


Figura 42. Golpe en carrera ascendente y descendente (C-07).



Debido al problema de no tener ejemplos de falla específicas, se recurre a la siguiente fuente de información, Los Expertos.

4.5 INFORMACIÓN OBTENIDA DE EXPERTOS.

El Ingeniero Fernando Calvete, profesor de la Escuela de Ingeniería de Petróleos facilitó dinagramas que empleó para un proyecto que él dirigió, relacionado con el diagnóstico automático de dinagramas pero empleando Redes Neuronales Artificiales.

Estos dinagramas fueron suministrados directamente en Excel, y son los siguientes (las unidades de los ejes vertical y horizontal son K-Lb y Pulgadas respectivamente):

Figura 43. Funcionamiento Normal (Tubería anclada).

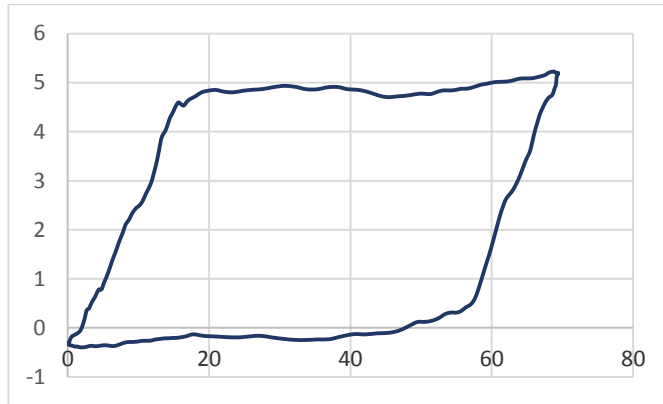


Figura 44. Interferencia de Gas.

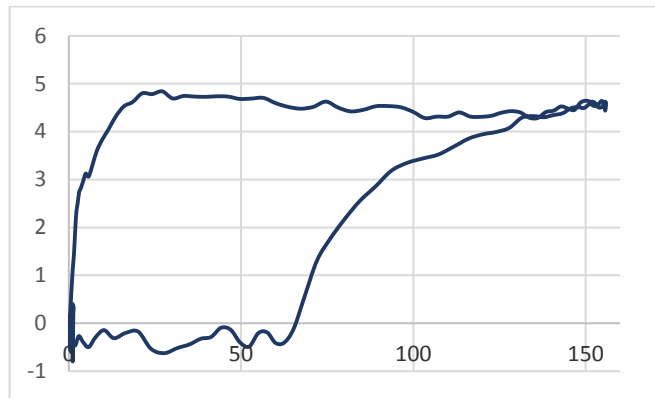


Figura 45. Fuga en la Válvula Viajera.

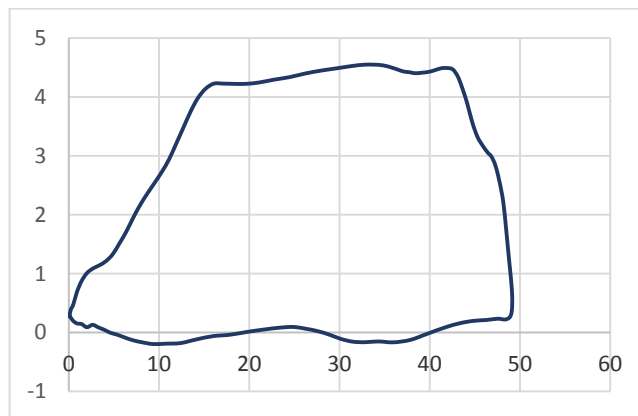


Figura 46. Fuga en la Válvula Viajera.

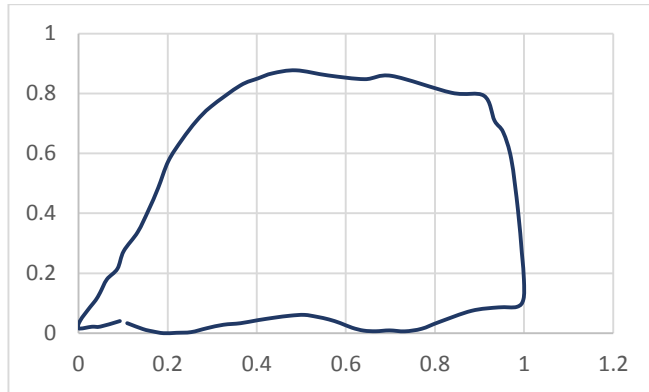


Figura 47. Fuga en la Válvula Viajera.

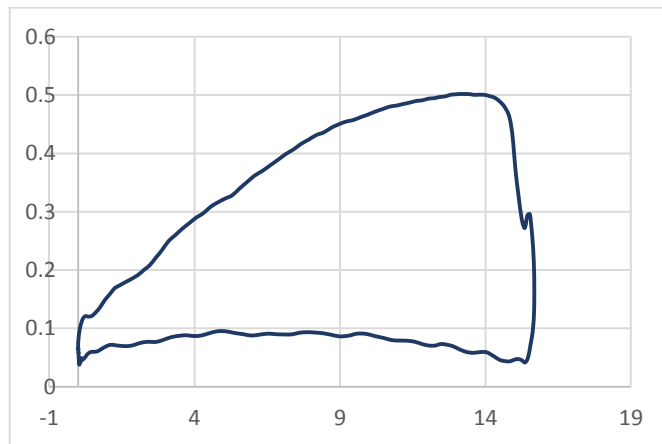


Figura 48. Rotura de Varilla.

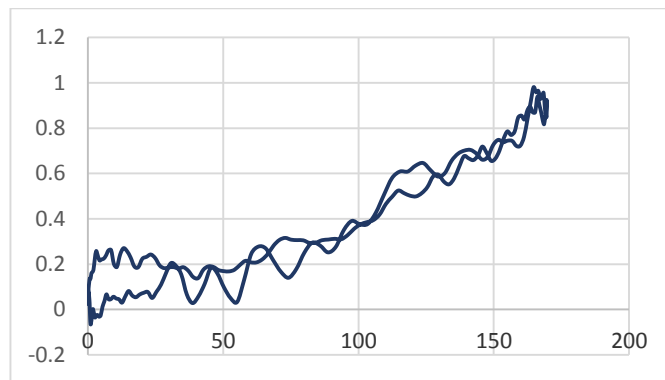


Figura 49. Rotura de Varilla.

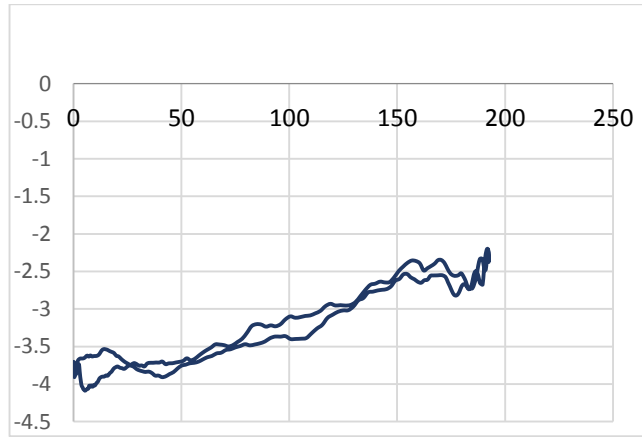


Figura 50. Rotura de Varilla.

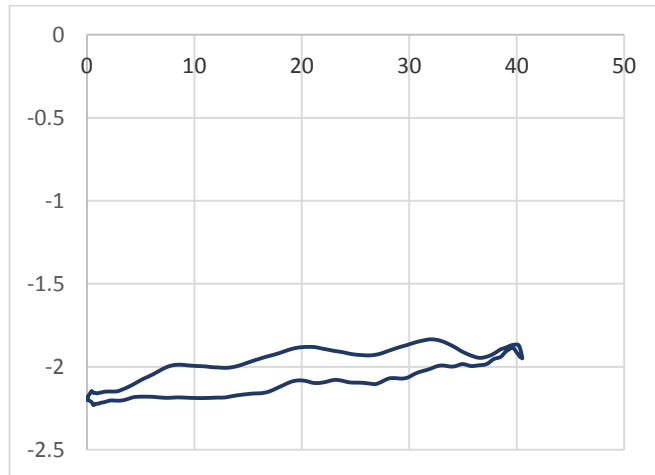
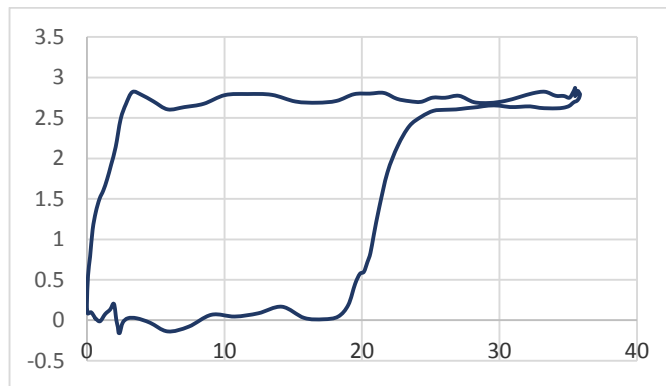


Figura 51. Golpe de Fluido.



El proyecto de investigación que dirigió el profesor Fernando Calvete, al igual que para este, no conto con suficientes ejemplos de falla, por la tanto optaron por crearlos.

Al continuar con el mismo problema, de poseer pocos ejemplos de falla, se contactó al Ingeniero Alexander Camacho Pardo, por medio de Profesor Jorge Meneses, del cual se obtuvieron los siguientes dinagramas:

Figura 52. Interferencia de Gas

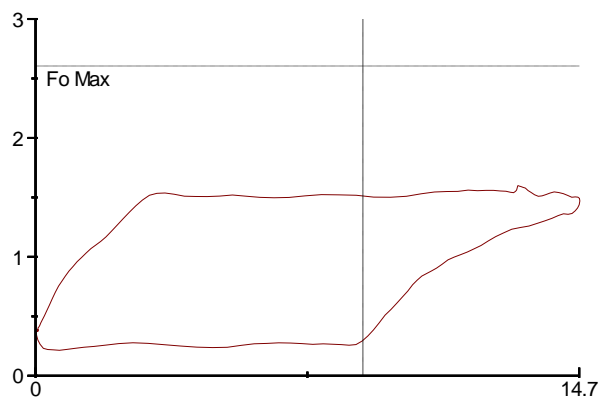
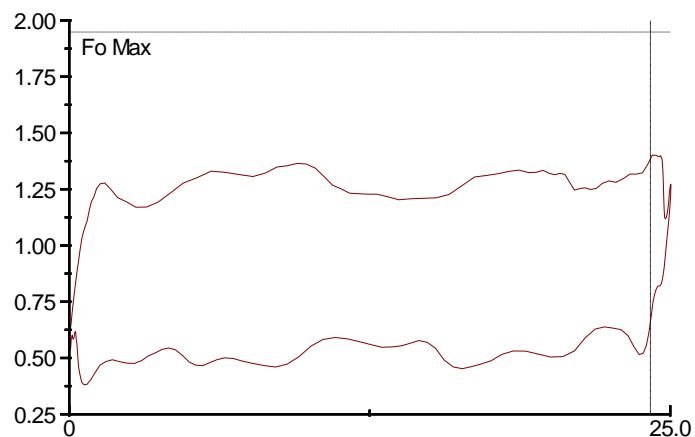


Figura 53. Buen Funcionamiento (Tubería no Anclada).



Los dinagramas proporcionados por el Ingeniero Alexander fueron cinco en total, los cuales hasta este punto son dinagramas donde se presenta una sola falla, los restantes son fallas combinadas, mostradas a continuación.

Figura 54. Golpe de Fluido y Fuga en la Válvula viajera.

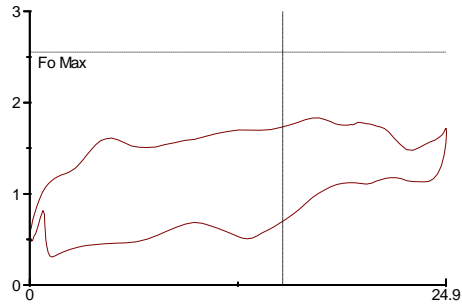


Figura 55. Golpe en carrera ascendente y mal funcionamiento de las válvulas.

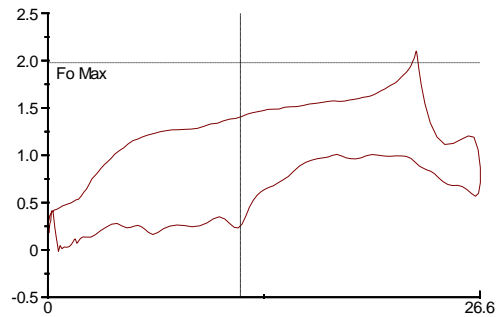
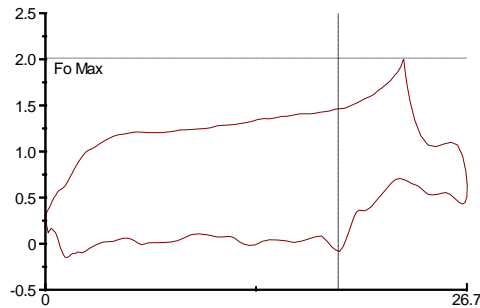


Figura 56. Golpe en carrera ascendente y mal funcionamiento de las válvulas.



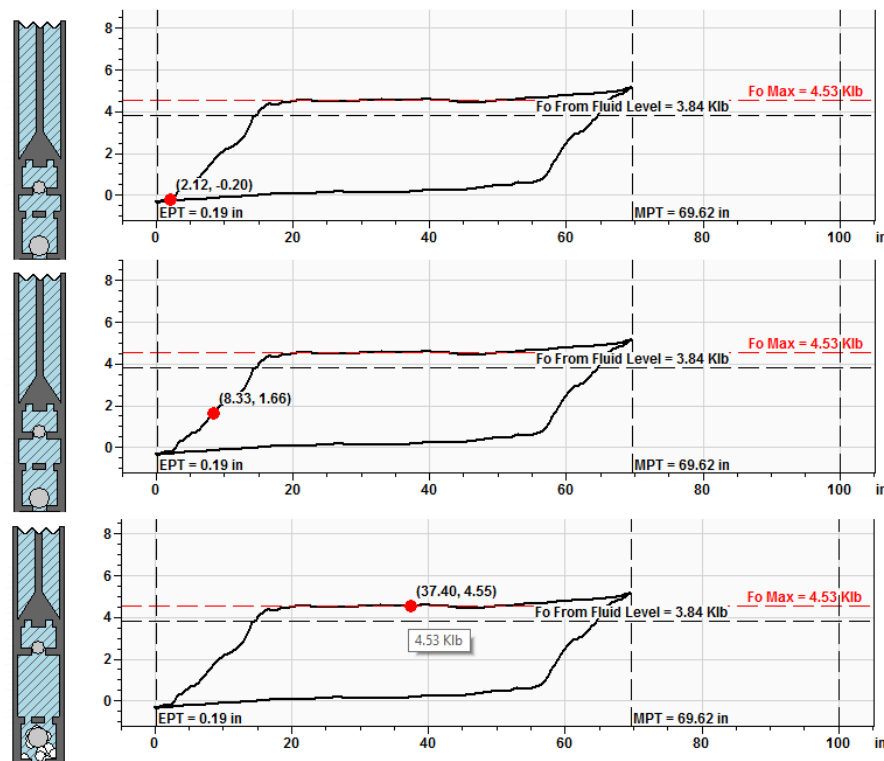
Como ya se expuso antes, la idea una “base de conocimiento” sólida, es el agrupar una serie de dinagramas diagnosticados por expertos humanos. Con la información anteriormente descrita no se cumple ese objetivo. Es por esto que nace la necesidad de tipificar estas fallas buscando obtener suficientes ejemplos como punto de partida para un entrenamiento aceptable.

Los dinagramas proporcionados por Campo Escuela Colorado no fueron empleados, debido a que son fallas combinadas, y no es el objetivo para esta investigación diagnosticarlas. Más sin embargo los dinagramas que fueron facilitados por los expertos (los de falla sencilla) se usaron solo para ser clasificados como prueba al mejor entrenamiento obtenido. Los resultados se exponen en el capítulo de pruebas y resultados.

4.6 TIPIFICACIÓN DE DINAGRAMAS.

Se decidió tipificar dinagramas debido a la falta de ejemplos de falla. El procedimiento de tipificación se explica en detalle a continuación:

Figura 57. Recorrido Punto a Punto.



Como primer paso se seleccionan las fallas a tipificar. Estas son las estudiadas en la sección 4.3 del presente capítulo. Posteriormente con la ayuda del software

TAM (Total Well Management), que permite recorrer punto a punto el dinagrama que se proyecta (ver figura 57), se da comienzo a la creación de los dinagramas BASE en una hoja de Excel, con el objetivo de emplearlos para tipificar los demás.

Entonces, se proyecta cada falla representativa de la base de datos (Usando el TAM) y se procede en almacenar cada punto del dinagrama mostrado en una hoja de Microsoft Excel. Siguiendo a este procedimiento (ver figura 58), realizado diez veces (10 Fallas), se inicia la creación de más ejemplos.

Después de almacenados en Microsoft Excel los dinagramas BASE, se procede a crear ruido tomando cada valor de punto y modificándolo de forma aleatoria, conservando su forma característica, como se muestra en la figura 59.

Figura 58. Almacenamiento del TAM a Excel.

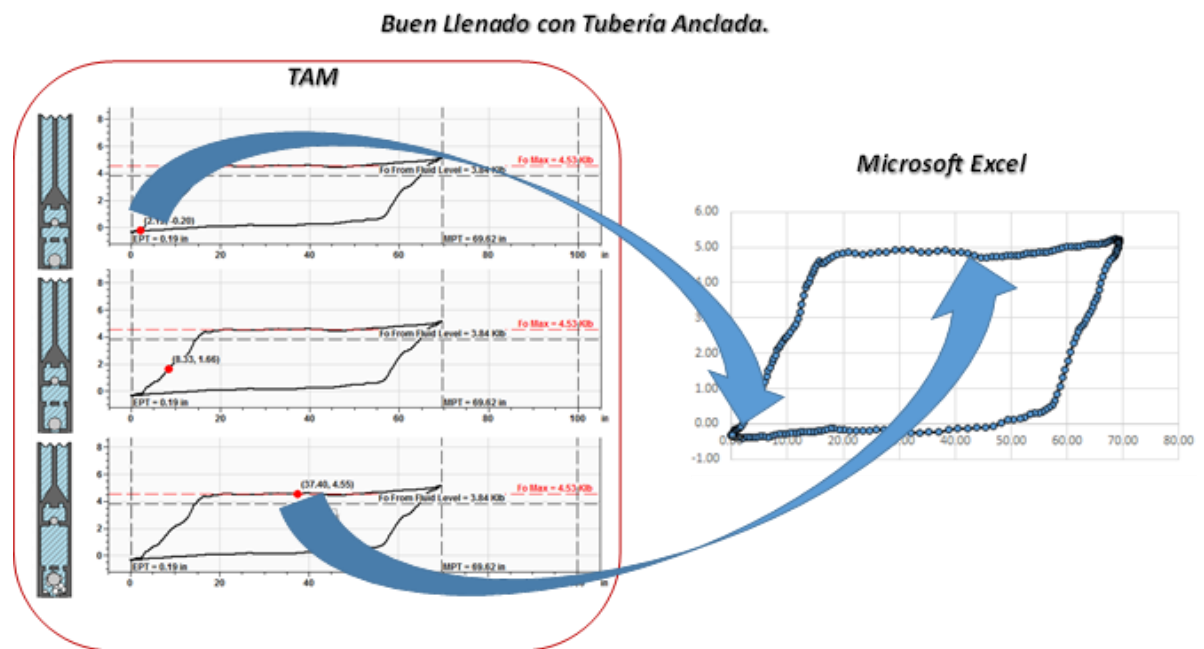
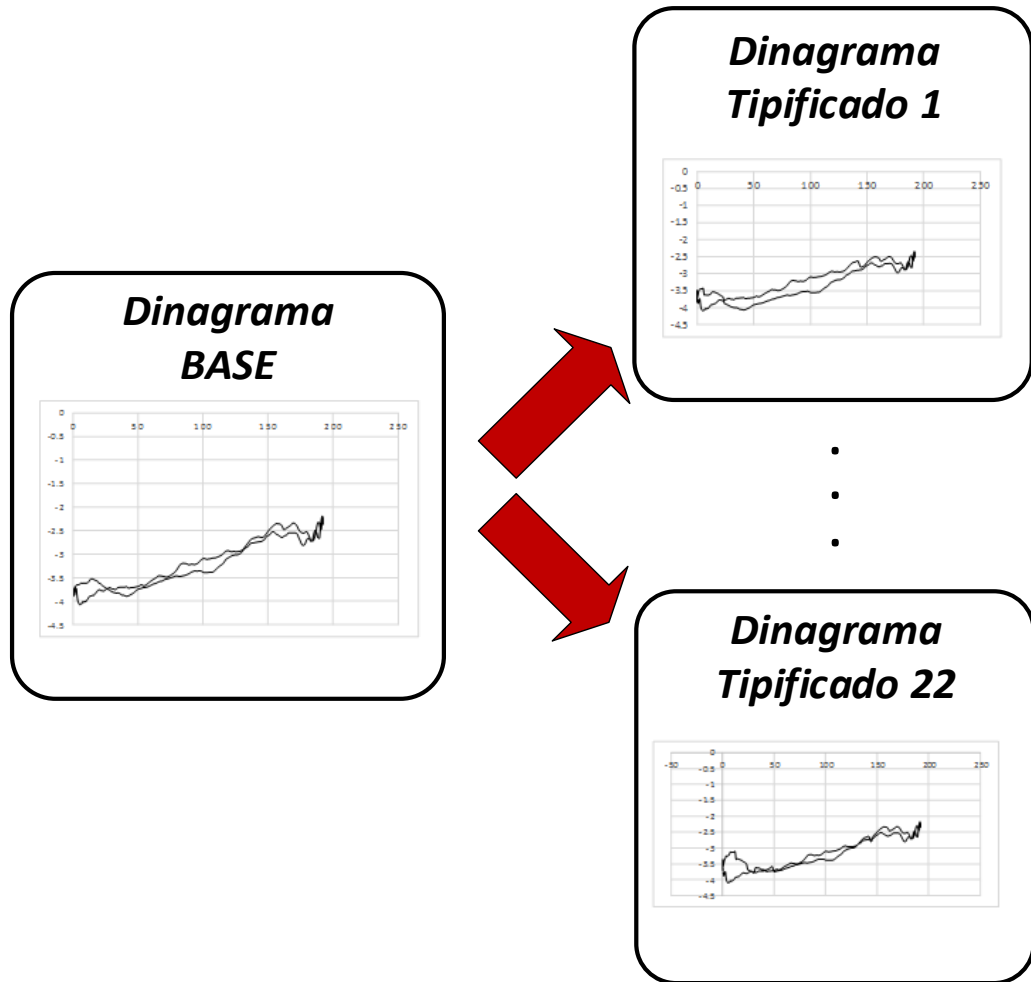


Figura 59. Creación de dinagramas.



En total se tipificaron 220 dinagramas, distribuidos 100 para entrenamiento de la Red, 60 para validación y 60 para clasificación.

5. PREPROCESAMIENTO DE DINAGRAMAS

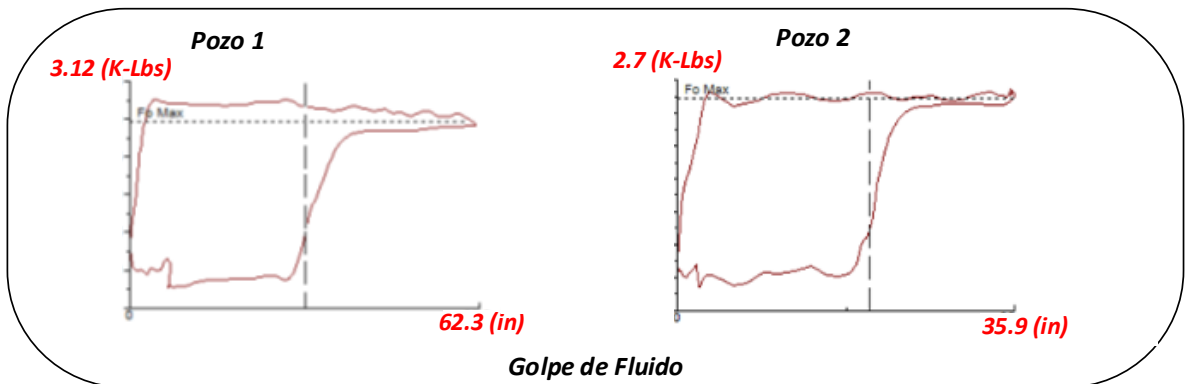
5.1 NORMALIZACIÓN DE DINAGRAMAS.

La normalización es un proceso que permite ajustar los valores de carga y recorrido, ya que las cargas máximas, mínimas y la longitud del recorrido registradas en cada carta dinamométrica dependen de varios factores específicos de cada pozo en el que se toma el registro, entre los factores más importantes se encuentra la profundidad de la bomba, el diámetro del pistón, el tipo de crudo, la cantidad de gas producido y la gravedad específica del agua de formación. En la figura 60 se puede observar un campo petrolero, donde dos pozos diferentes pueden presentar la misma falla a distintos valores de carga y recorrido.

Figura 60. Campo petrolero.



Dinagramas

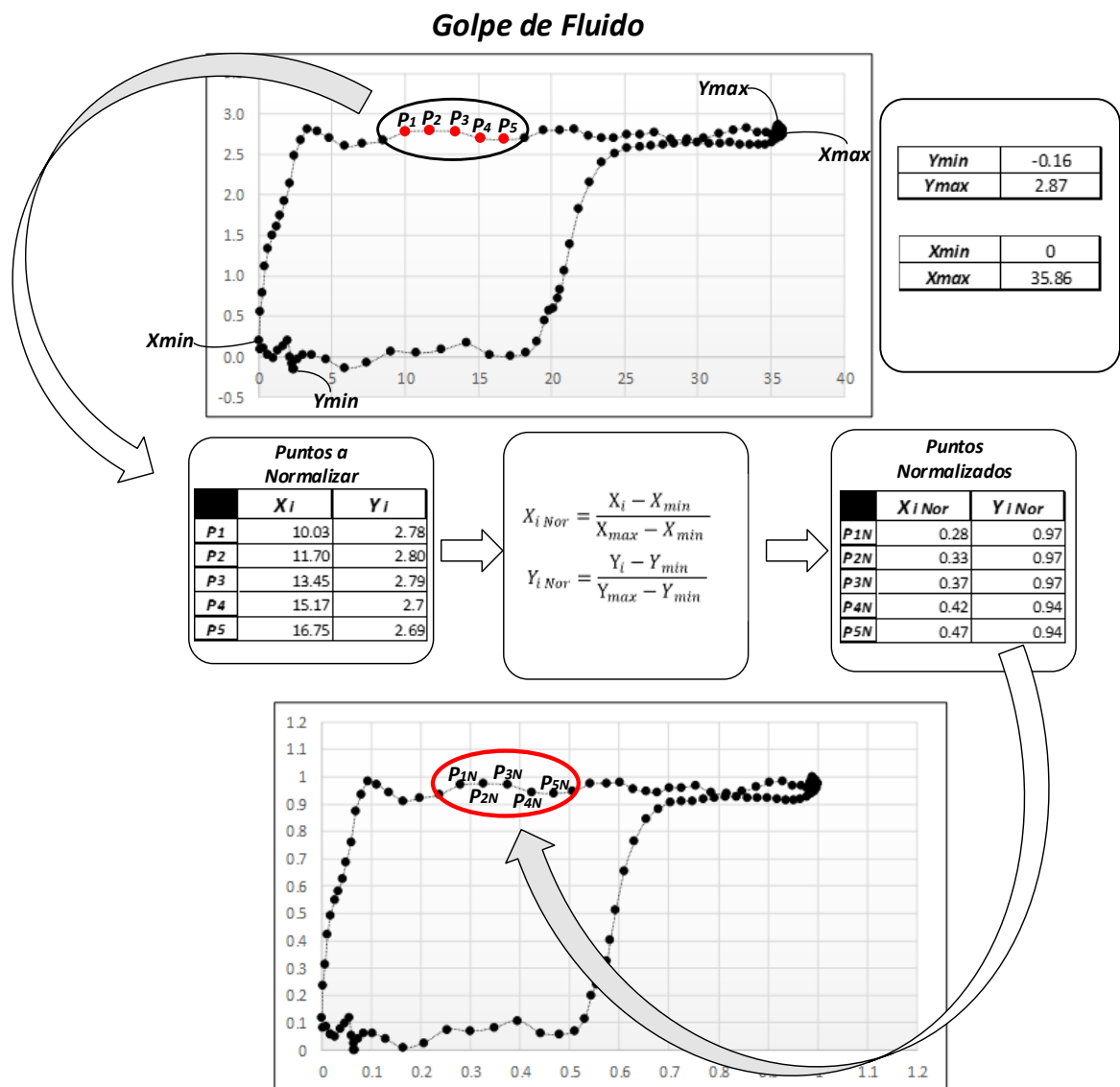


Como se planteó en el capítulo 3, el propósito de este proyecto es obtener un modelo Fuzzy que sea capaz de diagnosticar cualquier dinagrama en un campo petrolero sin importar el pozo del cual provenga.

Para lograrlo se deben ajustar los valores de carga y recorrido a una sola escala de tal manera que cuando se le presenten a la Máquina de Aprendizaje esta pueda reconocer el dinagrama a partir de su forma, mas no por sus valores específicos de carga y recorrido.

5.1.1 Procedimiento de Normalización.

Figura 61. Proceso de Normalización.



Observando la figura 61:

1. Se identifican los valores de amplitud a normalizar.
2. Se emplea un método conocido como límites naturales¹⁸. Este procedimiento transforma el valor de la evaluación de una serie de puntos a valores ubicados en un rango establecido, obedeciendo a la siguiente ecuación:

$$X_{i\text{ Nor}} = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}$$
$$Y_{i\text{ Nor}} = \frac{Y_i - Y_{\min}}{Y_{\max} - Y_{\min}}$$

Donde: X_i y Y_i son la coordenadas del punto P_j a normalizar, Y_{\max} y X_{\max} son los valores de carga y recorrido máximos respectivamente, Y_{\min} y X_{\min} son los valores de carga y recorrido mínimos respectivamente del dinagrama.

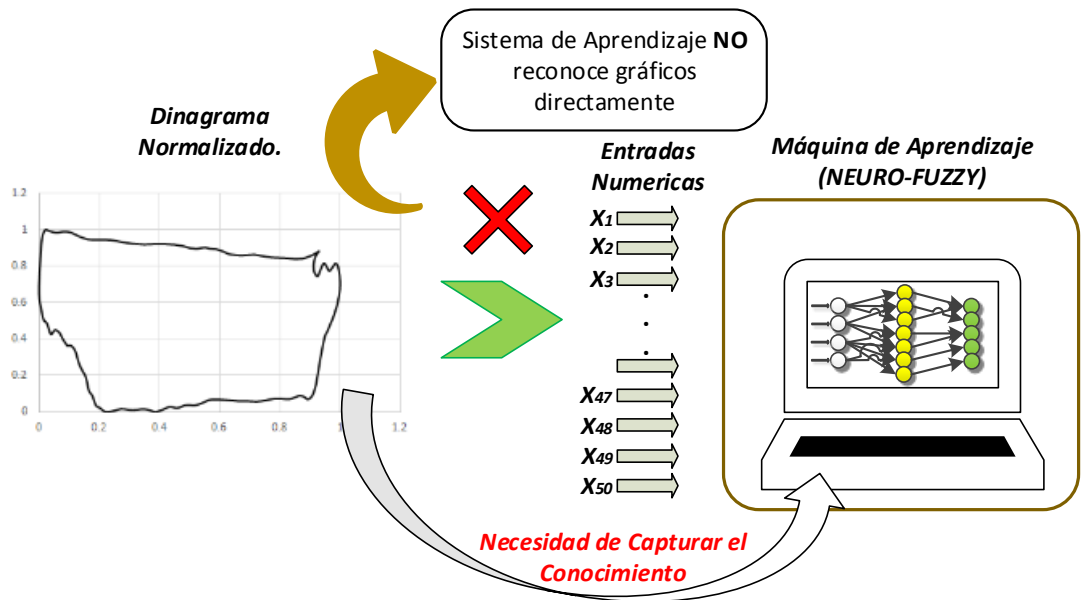
3. Aplicar la ecuación anterior a cada punto P_j del dinagrama, por ejemplo para el punto 1 (P_1) se tiene:
 - Coordenadas: $X_i = 10.03$ (in) y $Y_i = 2.78$ (K-lbs).
 - Valores Máximos: $X_{\max} = 35.86$ (in) y $Y_{\max} = 2.87$ (K-lbs).
 - Valores Mínimos: $X_{\min} = 0$ (in) y $Y_{\min} = -0.16$ (K-lbs).
 - P_1 normalizado: $X1\text{ Nor} = \frac{10.03-0}{35.86-0} = 0.28$; $Y1\text{ Nor} = \frac{2.78-(-0.16)}{2.87-(-0.16)} = 0.97$
4. Realizar el proceso para todos los puntos de cada dinagrama extraídos del TAM. Las cartas dinagráficas se normalizan a valores entre cero a uno.

¹⁸ Sistemas de evaluación, Asociación unificada de militares españoles. AUME.

5.2 DISCRETIZACIÓN.

Discretizar se define en pocas palabras, para esta investigación, como la representación numérica de un dinagrama. Este preproceso nace principalmente por la necesidad de capturar el conocimiento que posee el dinagrama para que pueda ser interpretado por la Máquina de Aprendizaje. Adicional a esto, la Máquina de Aprendizaje NO reconoce el dinagrama en su forma natural, es decir no recibe gráficos como entradas, (ver figura 62).

Figura 62. Necesidad de Discretizar.

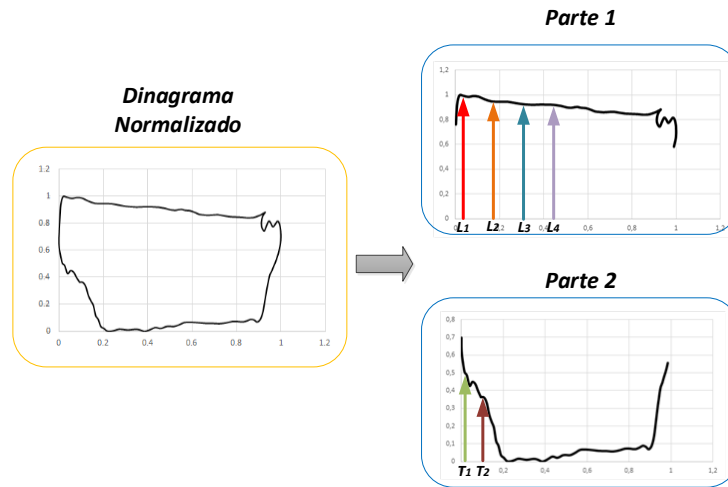


Teniendo en cuenta que la Máquina de Aprendizaje utilizada permite un máximo de cincuenta (50) entradas, y que el dinagrama se presenta en forma gráfica (cerrada), se destinó gran cantidad de tiempo en generar estrategias de cómo se podría representar un dinagrama de la mejor manera. Se estudiaron tres Métodos.

5.2.1 Primer Método.

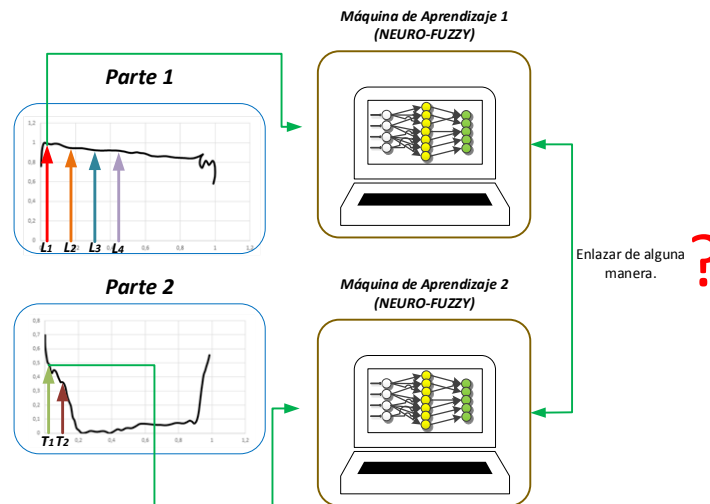
Dividir el dinagrama en dos partes y proyectar en todo su dominio líneas verticales, espaciadas cada $1/50$ de longitud, como se ve en la figura 63.

Figura 63. Primera Método.



Este proceso aunque viable por la necesidad de obtener entradas numéricas, para el momento en que se pensó, exigía un entrenamiento (de la Máquina de Aprendizaje) por cada parte (ver figura 64), y posteriormente ser enlazado de alguna manera. Este tipo de unión de entrenamientos es desconocido y no surgieron ideas aceptables para lograrlo.

Figura 64. Búsqueda de Enlace.

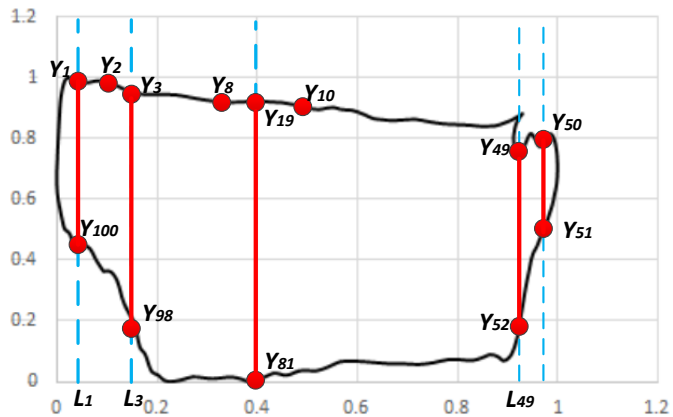


5.2.2 Segundo Método.

Proyectar 50 líneas verticales espaciadas en todo su dominio teniendo en cuenta solo las longitudes creadas dentro del dinagrama, como se observa en la figura

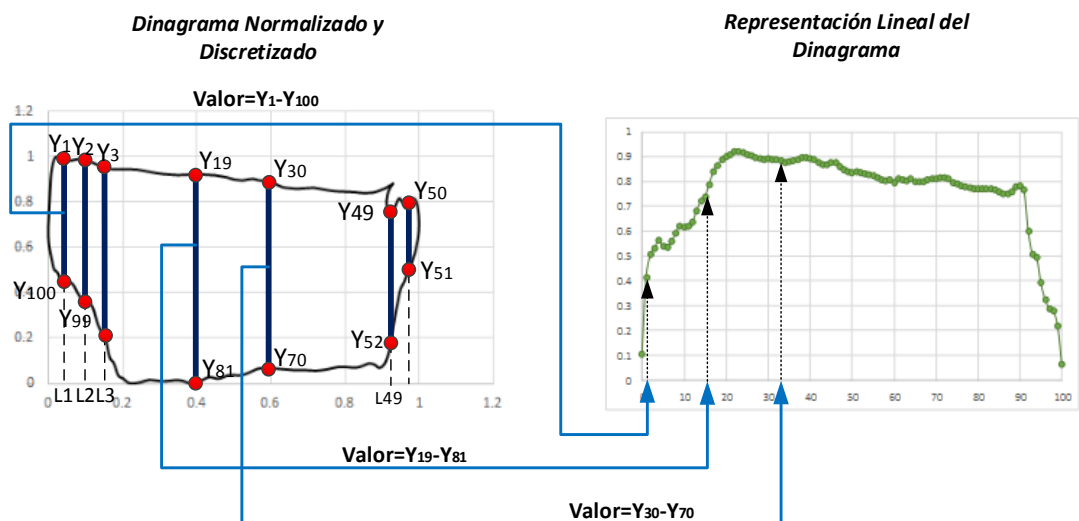
65. Es decir para las entradas de la Máquina de Aprendizaje de la figura 62, y tomando las amplitudes del dinagrama como Y_j ($j=1, j=2, j=3, \dots, j=100$), el valor X_1 para la primera entrada seria $X_1=Y_1-Y_{100}$, el valor X_3 para en la tercer entrada seria $X_3=Y_3-Y_{98}$, y así para los demás.

Figura 65. Segundo Método.



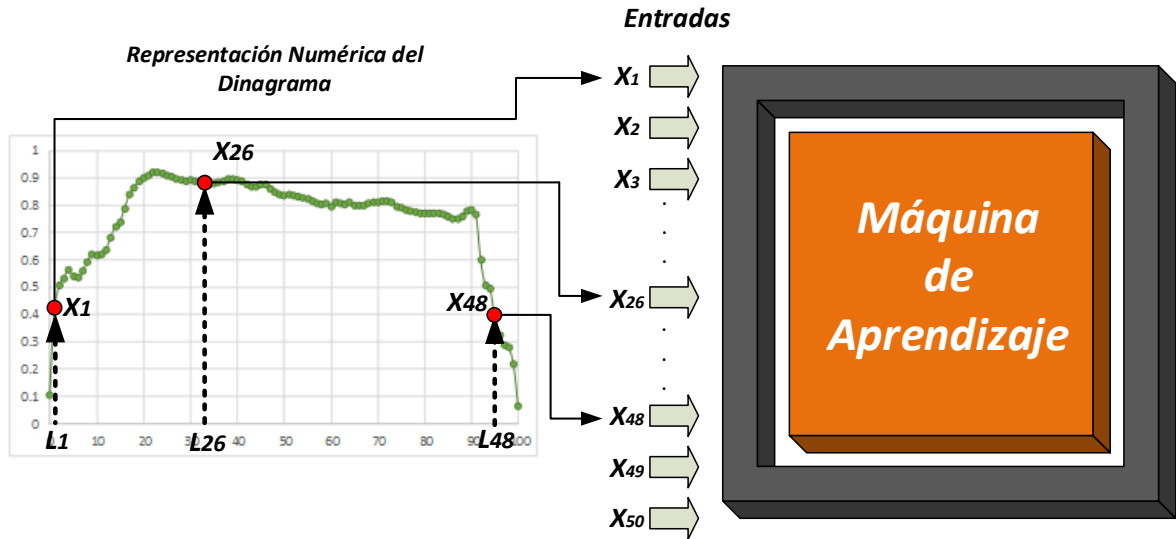
De esta manera se convierten dos valores numéricos de amplitud, en un solo valor, como se muestra en la figura 66.

Figura 66. Representación numérica del segundo Método.



Posterior a obtener el dinagrama numéricamente, se procede a ingresar las amplitudes de izquierda a derecha como lo muestra la figura 67.

Figura 67. Ingreso de amplitudes a la Máquina de Aprendizaje.



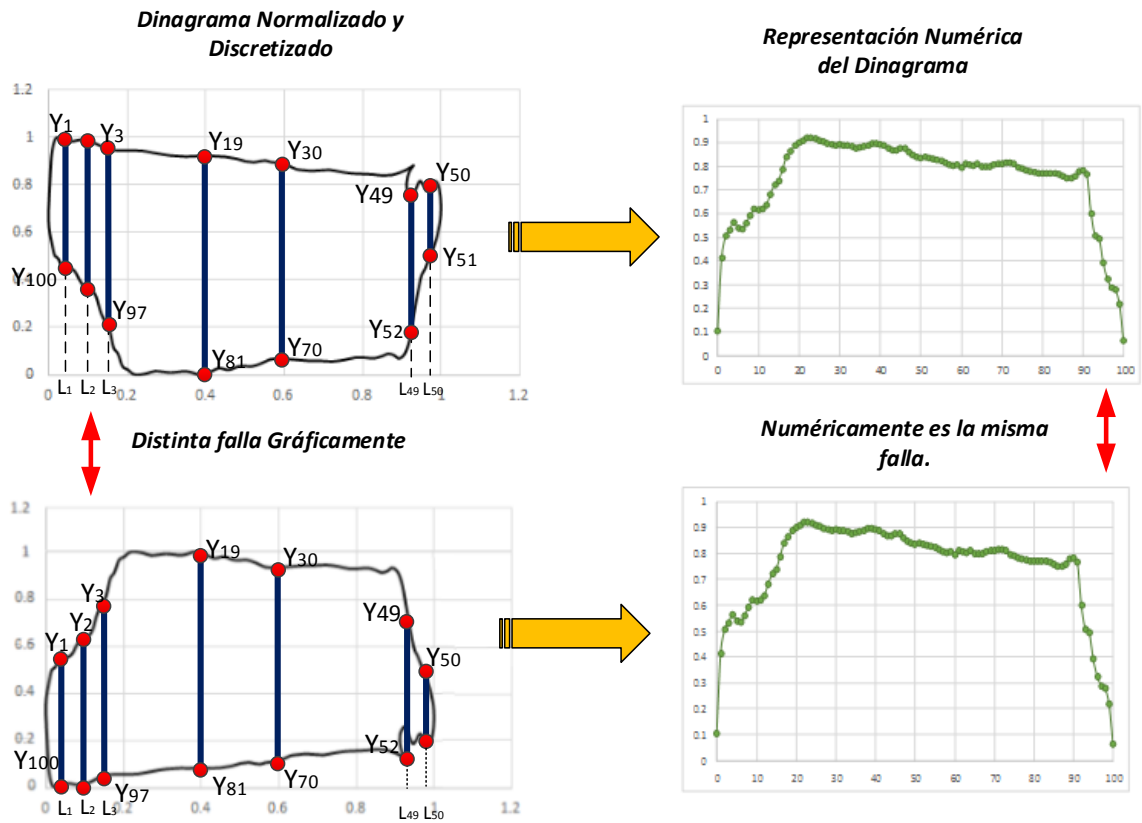
Por su viabilidad, se decidió emplear esta metodología debido a que hasta el momento no había generado inconsistencias. Se hicieron pruebas de entrenamiento y operación obteniendo resultados aceptables.

Se designó tiempo para estudiar posibles dificultades a presentarse en este método, tales como la identificación (por parte de la Máquina de Aprendizaje) de solo la falla que se está ingresando, además de que sea posible emplear este método de discretización para todos los tipos de falla que existan (falla sencilla).

La razón por la cual no se empleó esta técnica para el desarrollo total de esta investigación, se debe a la falta de referencia en las amplitudes obtenidas de cada dinagrama.

Por esta falta de referencia, se presentaba un efecto que se denominó "Mirror" el cual muestra dos fallas completamente diferentes gráficamente pero iguales numéricamente, como se observa en la figura 68.

Figura 68. Efecto "Mirror".



Por esta falta de consistencia y a pesar de que hasta este punto ya se habían realizado pruebas de entrenamiento y operación, se recurrió en concebir una nueva idea de representación numérica. Por esto, nace un tercer Método, y es el empleado para el desarrollo de esta investigación.

5.2.3 Tercer Método.

Como ya se explicó en la descripción del proyecto en esta opción se propone:

1. Dividir el dinagrama en veinticinco (25) partes iguales como se observa en la figura 69.

Figura 69. Tercera Método.

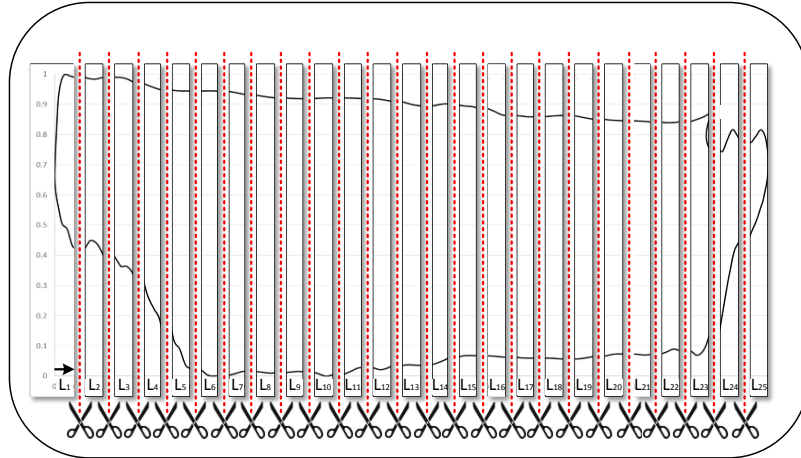
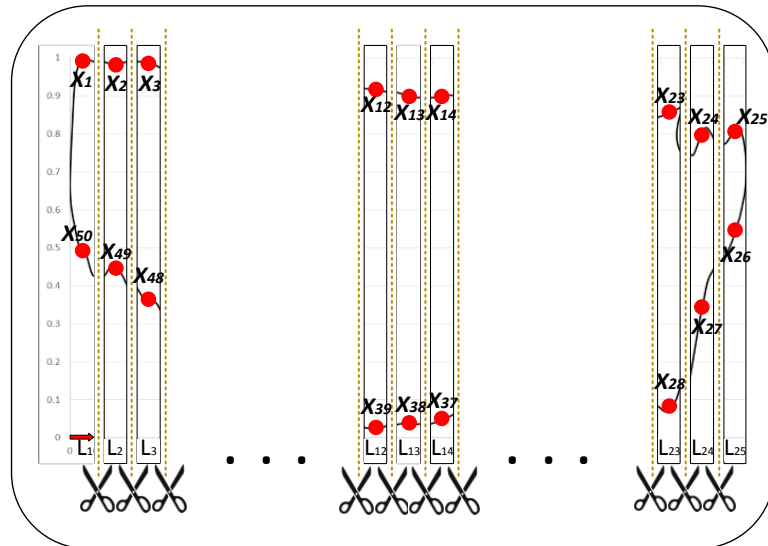


Figura 70. Segmentos (L_j) y Amplitudes (X_i).



Donde por cada $1/25$ de segmento (L_j) se obtienen dos valores de amplitud (X) del dinagrama, (ver figura 70). Se obtienen un total cincuenta (50) valores de amplitud (número de entradas máxima de la Máquina de Aprendizaje), en veinticinco (25) segmentos.

Los valores de amplitud se toman iniciando la carrera ascendente hasta el fin de la carrera descendente, ver figura 71.

Figura 71. Dirección en la toma de amplitudes.

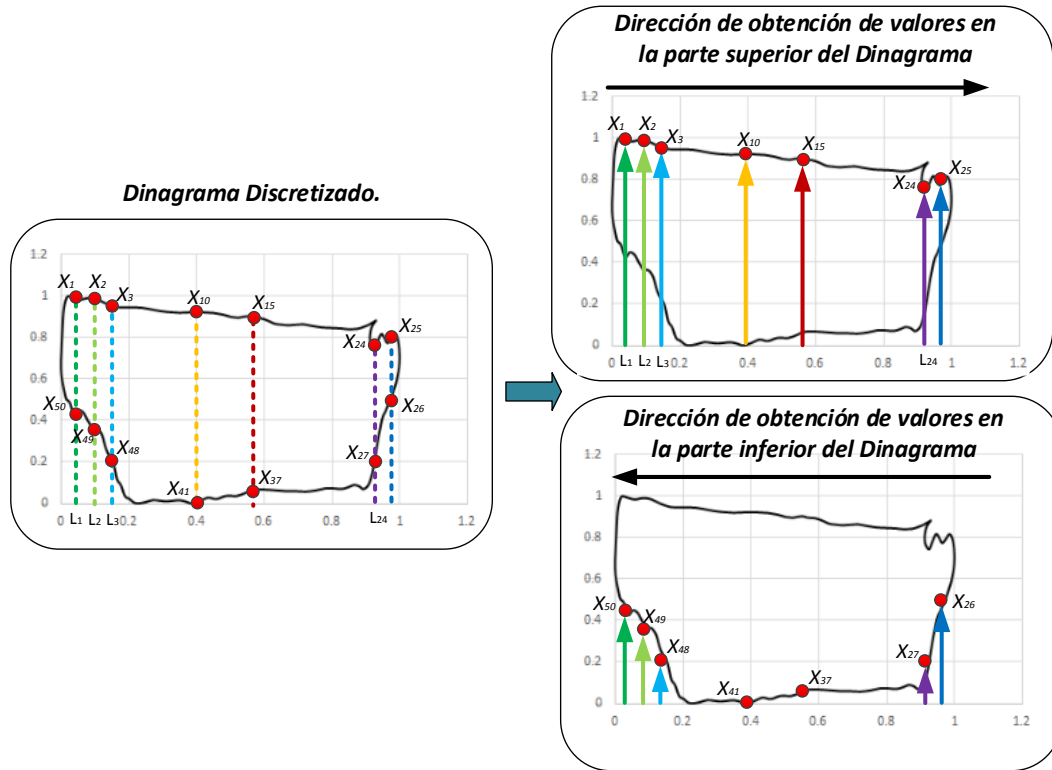
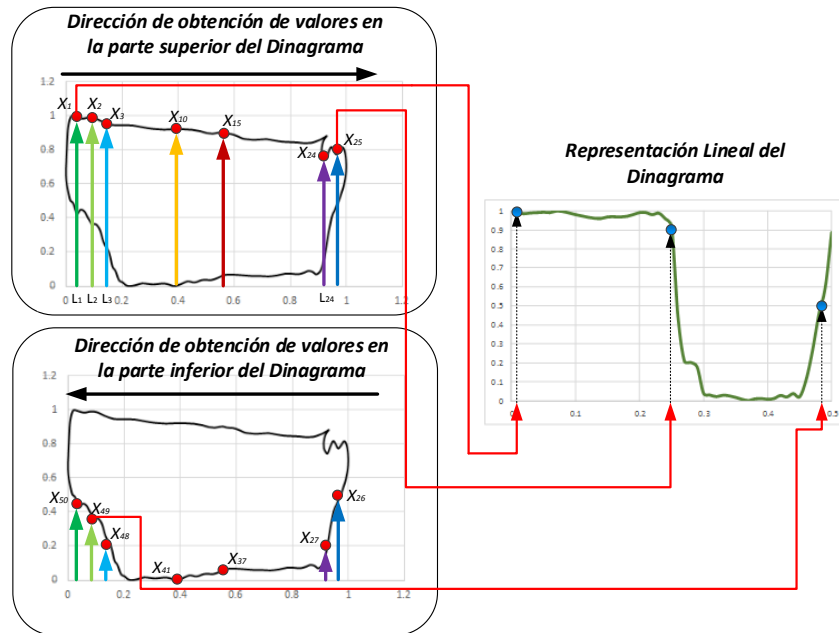
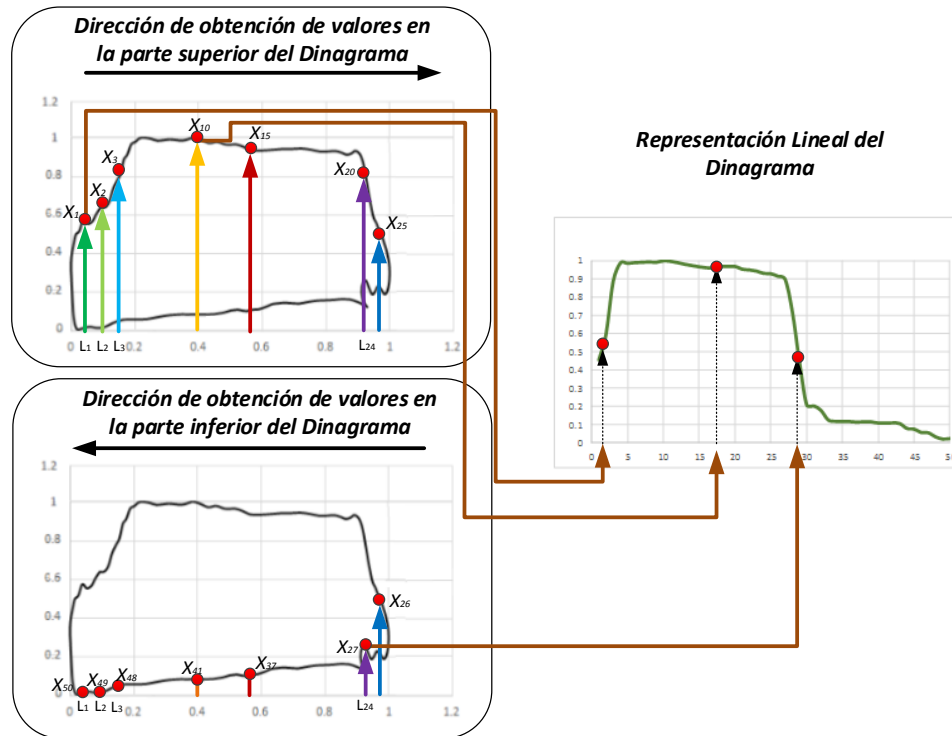


Figura 72. Representación numérica lineal de tercer Método.



Este tercer Método de representación numérica sobrepone el efecto “Mirror” por el hecho de que cada valor numérico poseerá una referencia con un orden específico.

Figura 73. Efecto “Mirror” tercer Método.



Comparando las representaciones lineales de las figuras 72 y 73, se concluye, al ser diferentes, la sobreposición del efecto (Mirror), además de un orden en la extracción de las amplitudes. Por confiabilidad y al no encontrar dificultades o inconsistencias en este Método, se decide emplearlo para Discretizar todos los dinagramas.

6. MÁQUINA DE APRENDIZAJE.

El enfoque de este proyecto apunta al mejoramiento del diagnóstico de equipos donde se emplea bombeo mecánico, a través de la evaluación de una arquitectura Neuro-Fuzzy, que para este caso particular se encuentra implementada en un software freeware denominado NEFCLASS-Q. Para cumplir este propósito es necesario familiarizarse con un tema esencial, Sistemas Neuro-Fuzzy (Lógica Fuzzy y Redes Neuronales Artificiales), además del funcionamiento del software desde la creación de los archivos hasta el producto final.

6.1 SISTEMAS NEURO-FUZZY.

6.1.1 Definición.

Un sistema Neuro-Fuzzy es una combinación de Redes Neuronales Artificiales y Sistemas Fuzzy. Las Redes Neuronales están conformadas por elementos de procesamiento simples que se comunican a través de conexiones de pesos y que están en capacidad de aproximar funciones o generar soluciones para ciertas labores o tareas específicas, por medio de aprendizaje a partir de ejemplos. También pueden usarse para resolver problemas de control y supervisión desarrollando un proceso de aprendizaje que es cimentado en la interacción de las funciones propias de la Red con los datos entregados. Este proceso de aprendizaje generalmente requiere de periodos considerables de tiempo y no garantiza un índice de confiabilidad óptimo, además presentan un inconveniente mayor al no permitir el manejo de un conocimiento previo que permita simplificar de manera considerable el tiempo necesario para llevar a cabo el proceso de aprendizaje.

La lógica Fuzzy por su parte permite manejar criterios o parámetros de estudio, con gran flexibilidad y que pueden ser modificados dentro de rangos diferentes para desarrollar una evaluación más completa y precisa de los datos e información inicialmente entrados. El manejo de las reglas fuzzy, como reglas de carácter lingüístico SI-ENTONCES, permite introducir una herramienta muy versátil al

momento de interactuar con variables que resultarían complejas de manipular desde un punto de vista de la matemática tradicional, pero que tratadas bajo la lógica Fuzzy, su interpretación resulta más práctica y simple. También cabe destacar la importancia de integrar el conocimiento previo en la obtención de resultados más flexibles que garantizan un proceso de aprendizaje más claro y preciso, al momento de evaluar los resultados obtenidos.

Una combinación de ambos enfoques (Red Neuronal Artificial y Sistemas Fuzzy), tiene la capacidad de involucrar en el proceso de aprendizaje un conocimiento a partir de las reglas Fuzzy SI-ENTONCES. Superando los problemas de compatibilidad y sintonización estructural de cada sistema y aprovechando las ventajas y eliminando las desventajas que cada sistema presenta por separado, se obtiene un sistema más eficiente, potente y versátil denominado Sistema Neuro-Fuzzy. La intención principal de un sistema Neuro-Fuzzy es crear o mejorar un sistema Fuzzy apoyado por las Redes Neuronales Artificiales o mejorar el proceso de aprendizaje de una Red neuronal artificial a partir de la lógica Fuzzy.

6.1.2 Métodos basados en sistemas Neuro-Fuzzy.

A continuación se presentan varios enfoques que combinan Redes Neuronales y sistemas Fuzzy:

6.1.2.1 Modelos concurrentes neuronales/Fuzzy.

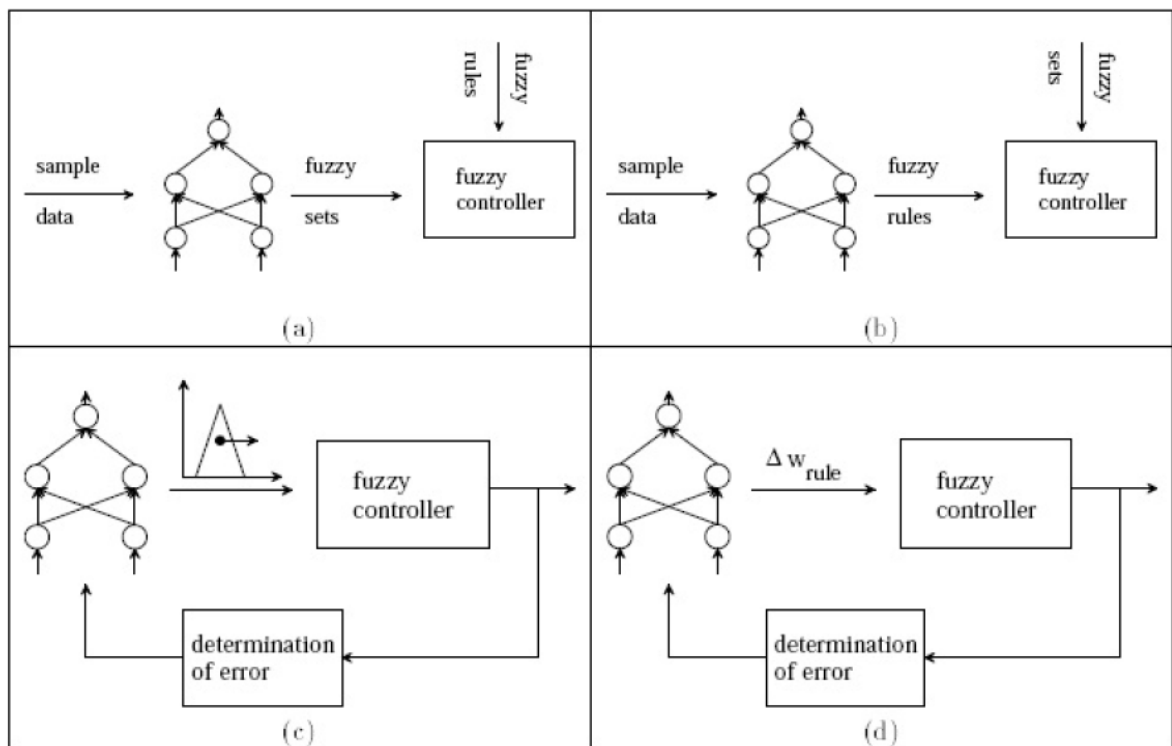
Esta es la forma más débil de establecer una combinación Neuro-Fuzzy. Una Red neuronal y un sistema Fuzzy son empleados en la misma labor. La Red neuronal puede usarse como un pre o pos procesador para el sistema Fuzzy, pero la Red neuronal presenta limitaciones al no poder inferir en el cambio de algún parámetro en el sistema Fuzzy.

6.1.2.2 Modelos cooperativos Neuro-Fuzzy.

En este caso se tiene una Red neuronal o en el caso más simple un algoritmo de aprendizaje neuronal, que es usado para aprender ciertos parámetros de los sistemas Fuzzy (conjuntos Fuzzy, reglas Fuzzy o pesos de reglas). Esta

combinación es determinada esencialmente por el hecho de que un Red neuronal este aprendiendo u optimizando varios parámetros de un sistema Fuzzy antes que el sistema Fuzzy sea operado (offline) o el proceso de aprendizaje u optimización se ejecute mientras el sistema Fuzzy opera (online). En la figura 74 se presentan cuatro enfoques de modelo cooperativos.

Figura 74. Diferentes modelos Neuro-Fuzzy cooperativos.



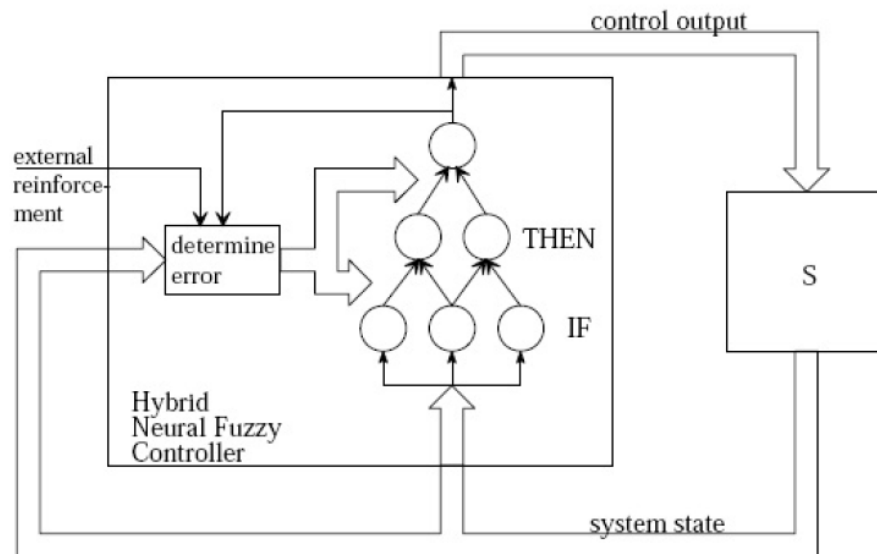
Fuente: NAUCK, Detleff; KRUSE, Rudolf. Choosing Appropriate Neuro-Fuzzy Model. Technical University of Braunschweig, Department of Computer Science, Germany.

6.1.2.3 Modelos híbridos Neuro-Fuzzy.

Este es otro tipo de combinación entre Redes Neuronales y sistemas Fuzzy. Esta combinación es un sistema Fuzzy que puede interpretarse como una Red neuronal especial o un sistema Fuzzy que emplea el procedimiento de aprendizaje de las Redes Neuronales Artificiales. En la figura 75, un modelo híbrido Neuro-Fuzzy es mostrado; la ventaja de este modelo radica en la consistencia de su

arquitectura, ya que permite establecer un puente o comunicación entre diferentes modelos. La idea de un modelo híbrido es la interpretación de las reglas base Fuzzy en términos de una Red neuronal, de esta manera los conjuntos Fuzzy pueden ser interpretados como pesos y las reglas, variables de entrada, variables de salida pueden ser representadas como neuronas. El algoritmo de aprendizaje se asemeja a una Red Neuronal, generando un cambio de arquitectura, es decir en una adaptación de los pesos y/o en la creación o eliminación de conexiones. Estos cambios pueden ser interpretados en términos de una Red Neuronal o en términos de un sistema Fuzzy. Este último aspecto es muy importantes ya que el comportamiento de caja negra de las Redes Neuronales es manejado desde un punto de vista diferente. Esto proporciona un resultado óptimo en el procedimiento de aprendizaje, ya que este se puede representar en forma de reglas base Fuzzy.

Figura 75. Modelo híbrido Neuro-Fuzzy.



Fuente: NAUCK, Detleff; KRUSE, Rudolf. Choosing Appropriate Neuro-Fuzzy Model. Technical University of Braunschweig, Department of Computer Science, Germany.

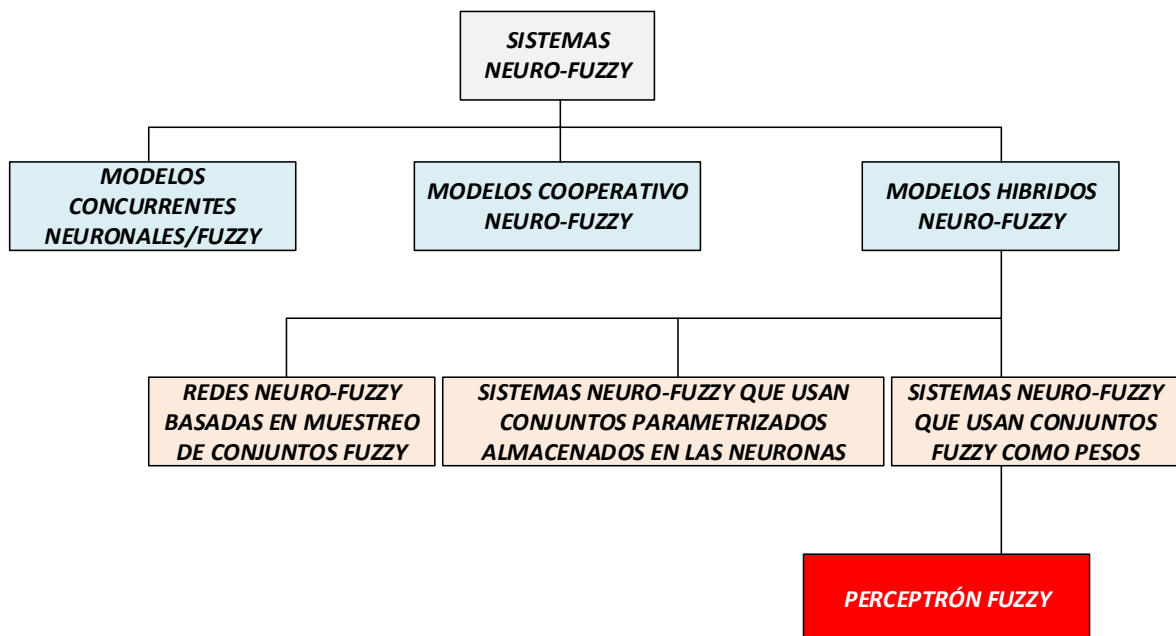
El enfoque híbrido posee una estructura que emplea conjuntos Fuzzy como pesos, en contraste con los demás enfoques, donde los conceptos de Fuzzy y Neuronal se combinan para formar una arquitectura que usa pesos con valores reales. En general se distinguen tres tipos de enfoques híbridos Neuro-Fuzzy:

- Redes Neuro-Fuzzy basadas en muestreo.
- Sistemas Neuro-Fuzzy que usan conjuntos parametrizados almacenados en las neuronas.
- Sistemas Neuro-Fuzzy que usan conjuntos Fuzzy como pesos.

Este último derivado del modelo Perceptrón Fuzzy genérico que define las propiedades básicas de este tipo de modelos Neuro-Fuzzy.

La figura 76 muestra un resumen de la clasificación de los sistemas Neuro-Fuzzy, llegando a la arquitectura empleada para la obtención del modelo Fuzzy (objetivo de este proyecto), **Perceptrón Fuzzy**.

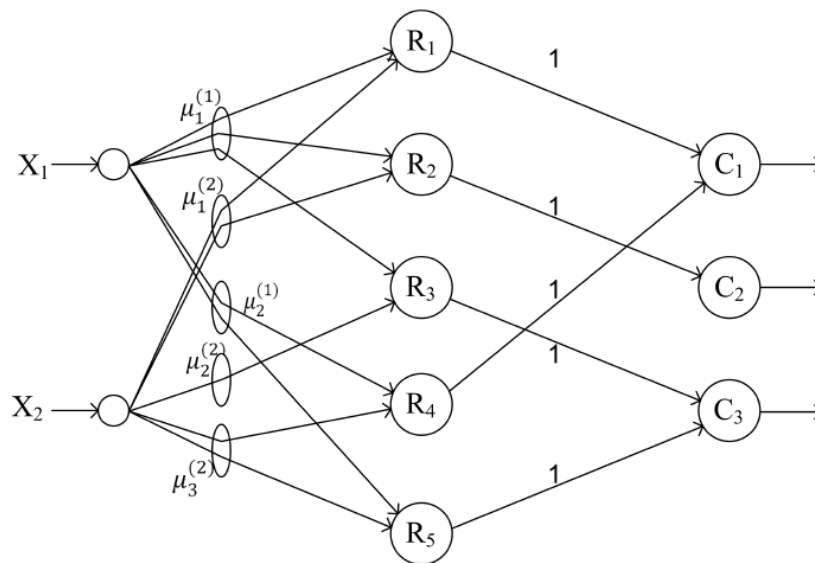
Figura 76. Sistemas Neuro-Fuzzy.



6.2 PERCEPTRÓN FUZZY¹⁹.

El Perceptrón Fuzzy tiene la arquitectura de una Red neuronal “Perceptrón multicapas”, pero los pesos son modelados como conjuntos Fuzzy y las funciones de activación, salida y propagación son modificadas de otra manera. La intención de este modelo es poder ser interpretado en forma de reglas lingüísticas (SI-ENTONCES), ser capaz de usar reglas basadas en el conocimiento previo y que pueda entrenarse usando métodos de aprendizaje neuronal, de tal forma que el aprendizaje no tenga que empezar desde cero.

Figura 77. Perceptrón Fuzzy.



Observando la figura 77 un Perceptrón Fuzzy está constituido por:

CAPA DE ENTRADA: Esta primera capa representa los nodos fuente y para fines prácticos funciona como módulos de procesamiento. Esta capa de entrada es la encargada de recibir cada uno de los patrones de datos, donde cada unidad de procesamiento de la capa de entrada recibe un dato numérico, y su activación o respuesta es igual a la entrada externa, y lo conecta a las respectivas unidades de procesamiento llamadas reglas.

¹⁹ CABALLERO, Fabio y SALAMANCA Julián. Herramienta de aplicación software para clasificación de patrones de datos implementando una arquitectura Neuro-Fuzzy. 2011. p. 36-41

CAPA OCULTA: O capa de reglas, es la encargada de realizar las inferencias lingüísticas. Almacenan las reglas, es decir conectan el antecedente con el consecuente. La activación de estos nodos es dado por un operador Fuzzy.

TERCERA CAPA: O capa de clases, es la que representa las diferentes opciones a la que puede pertenecer un patrón de datos, y se encarga de hacer el proceso de cálculo del error.

Remitiéndonos a la figura 77, μ_j^i representa los pesos sinápticos $W(X_i, R_k)$ donde i representa la variable de entrada a la unidad de procesamiento de la primera capa, y la j representa el número del conjunto para la variable i , En otras palabras, cada variable tiene un universo de discurso donde es definida de manera imprecisa por medio de los conjuntos Fuzzy.

Todos los pesos de las conexiones (conjuntos Fuzzy) que salen de una unidad de procesamiento de la capa de entrada son las que definen el lenguaje aproximado o inexacto de la correspondiente variable que representa el nodo de entrada en el universo de discurso, entonces $\mu_1^{(1)}, \mu_2^{(1)}, \mu_3^{(1)}$ son los conjuntos que están dentro del universo de discurso de la variable X_1 .

Continuando con la figura 77, R_k representa las reglas, donde a esta le puede llegar una o más conexiones de la capa de entrada, la activación o respuesta de la neurona de la capa oculta, estará dada por:

$$a_R^{(p)} = \min_{x \in U_1} \{W(x, R) (a_x^{(p)})\}$$

En otras palabras la salida de R_k está determinada por el peso de la conexión que tenga el valor más pequeño, este valor depende de la forma que tenga el conjunto Fuzzy de cada conexión y del patrón que se está propagando a través de la Red. Es sí como dependiendo de las conexiones que se tengan en la arquitectura, está implícita la regla semántica, pero hay que tener en cuenta que por cada patrón de datos que se esté propagando se puede activar más de una regla.

$W(R_k, C_m)$ es la conexión de la regla R_k a la unidad de salida C_m . Por razones semánticas los pesos de estas conexiones son valores fijos, y pueden estar entre 1 o 0 (si la conexión no existe). Cada neurona de la capa escondida es conectada solamente a una unidad de salida y la función de activación de las neuronas de salida es un operador Fuzzy. La activación de una neurona de salida está dada por la siguiente ecuación.

$$a_c^{(p)} = \max_{R \in U_2} \{a_R^{(p)}\}$$

Es decir, si la neurona de salida esta interconectada con más de una Regla, la activación está determinada por la Regla conectada a dicha neurona de salida que tenga mayor activación. La principal característica que tiene una neurona de salida es la activación, que puede ser 1 o 0, las activaciones de estas determinaran la clasificación del patrón propagado.

Una Red Neuro-Fuzzy puede ser creada a partir de conocimiento previo, o a partir de una base de reglas vacías, que es llenada en el proceso de aprendizaje con datos. Por cada variable de entrada se debe definir cuantos conjuntos Fuzzy van a constituir el universo de discurso con sus respectivos límites, igualmente se puede especificar el número máximo de reglas a crear en la capa escondida de la arquitectura, con la condición de que por cada clase debe existir al menos una regla. La función de pertenencia utilizada es de forma triangular y es definida por 3 parámetros:

$$\mu : \mathbb{R} \rightarrow \mu(x) = \begin{cases} \frac{x - a}{b - a} & \text{si } x \in [a, b) \\ \frac{c - x}{c - b} & \text{si } x \in [b, c] \\ 0 & \text{en otro caso} \end{cases}$$

Adicionalmente los conjuntos de los extremos son representados como la mitad de un trapecio (cada uno).

6.2.1 Configuración del modelo

Considérese un sistema Neuro-Fuzzy con la siguiente configuración:

1. n unidades de entrada representadas por $U_1 = \{x_1, x_2, \dots, x_n\}$. Donde U_1 representa la capa de entrada, n el número de variables de entrada y x la neurona de la capa de entrada.
2. Un número $k \leq k_{max}$ de reglas $U_2 = \{R_1, R_2, \dots, R_k\}$. Donde U_2 representa la capa escondida, k el número de reglas y R la regla o neurona de la capa escondida.
3. m unidades de salida de la capa $U_3 = \{C_1, C_2, \dots, C_m\}$. Donde U_3 representa la capa de salida, m el número de clases y C la neurona de salida.
4. Cada conexión entre las unidades $x_i \in U_1$ y $R_j \in U_2$ tiene un peso que es un conjunto Fuzzy, el cual es etiquetado con el término lingüístico $\mu_{jr}^{(i)}$ ($jr \in \{1, 2, \dots, qi\}$)
5. Los pesos de las conexiones sinápticas $W(R, C) \in \{1, 0\}$ para todo $R \in U_2$ y $C \in U_3$. Es decir los pesos entre las unidades de procesamiento de la capa escondida y la capa de salida pueden ser 1 o 0.
6. Si denotamos a $L_{x,R}$ como todas las etiquetas lingüísticas de las conexiones entre las $x \in U_1$ y $R \in U_2$ para todas $R, R' \in U_2$ se cumple que: $(V(x \in U_1)L_{x,R} = L_{x,R'}) \Rightarrow R = R'$ Esto quiere decir que si todas las etiquetas lingüísticas de las conexiones que conectan a una Regla R son iguales a las etiquetas de una Regla R' es porque $R = R'$.
7. Para todas las unidades de Regla $R \in U_2$ y todas las unidades $C, C' \in U_3$ se cumple que: $(W(R, C) = 1) \wedge (W(R, C') = 1) \Rightarrow C = C'$ Esto quiere decir que solo debe haber una conexión que salga de la unidad de Regla que conecte a una neurona de salida y que su peso sea de 1.
8. Para todas las unidades de salida $C \in U_3$, la activación $o_c = a_c = net_c$.

9. Para todas las unidades de salida $C \in U_3$, la entrada net_c es calculada como:

$$net_c = \frac{\sum_{R \in U_2} W(R, C) * o_c}{\sum_{R \in U_2} W(R, C)}$$

10. La base de reglas puede ser definida por el usuario o de forma combinada usuario-preprocesamiento. Igualmente dado un conjunto de patrones de aprendizaje $\zeta = \{(p_1, t_1)(p_2, t_2), \dots, (p_s, t_s)\}$ que consiste en un patrón de entrada $p \in \mathbb{R}^n$ y una respuesta esperada $t \in \{0,1\}^m$ el algoritmo crea k unidades de regla en los siguientes pasos.

6.2.2 Preprocesamiento

Este paso tiene como objetivo crear una base de reglas, sobre las cuales se puede fundamentar para realizar el entrenamiento y afinación de los conjuntos Fuzzy.

1. Seleccionar el siguiente patrón (p, t) de ζ , donde p representa el conjunto de datos de entrada, t representa la salida deseada de la Red cuando se propague el respectivo conjunto de datos de entrada y ζ representa el conjunto de patrones de datos.
2. Para cada neurona de entrada $x_i \in U_1$ encontrar la función de pertenencia $\mu_{jr}^{(i)}$ tal que: $\mu_{jr}^{(i)}(p_i) = \max_{j \in \{1, \dots, q_i\}} \{\mu_j^{(i)}(p_i)\}$ Es decir encontrar el conjunto que tenga mayor función de pertenencia cuando se propague el patrón.
3. Si el número de reglas existentes es menor que $kmax$ y no existe una unidad de regla con: $W(x_1, R) = \mu_{j_1}^{(1)}, \dots, W(x_n, R) = \mu_{j_n}^{(n)}$ entonces se debe crear la unidad de regla y conectarlo a la unidad de salida c_1 si $t_1 = 1$, donde t_1 es la salida esperada o clasificación del patrón propagado.
4. Si todavía hay patrones sin procesar en ζ , y $k < kmax$, entonces volver al paso (1), de lo contrario finalizar el proceso.

Determinar la base de reglas por medio de uno de los siguientes pasos:

- **Simple:** consiste en dejar las k unidades de reglas creadas y parar cuando $k < kmax$
- **Mejores Reglas:** Procesar cada uno de los patrones ζ una vez más, y acumular la activación de cada una de las reglas para cada clase cuando se propaguen los patrones. Sí una unidad de regla R muestra una mayor activación acumulada para una clase C_j que para la clase C_R especificada por la conclusión de la regla, entonces se debe cambiar la conclusión de la regla R a C_j , esto quiere decir que se debe desconectar la regla R de la unidad de salida C_R y conectarla con la unidad de salida C_j . Procesar los patrones una vez más y calcular para cada unidad de regla:

$$V_R = \sum_{p \in \zeta} a_R^{(p)} e_p, e_p = \begin{cases} 1 & \text{si es clasificado correctamente} \\ 0 & \text{otro caso} \end{cases}$$

- **Mejores reglas por clase:** Se procede de la misma forma que el método anterior, pero conservando para cada clase C_j las $\frac{k}{m}$ reglas, cuya conclusión representa la clase C_j .

6.2.3 Entrenamiento

El algoritmo de aprendizaje supervisado del sistema NEFCLASS-Q adapta los conjuntos Fuzzy corriendo los datos de entrenamiento ζ cíclicamente hasta alcanzar algún criterio, por ejemplo que sobrepase el número máximo de errores, o que el error no pueda disminuir más a futuro.

Después de que el patrón es propagado a través de la Red, el error es calculado por cada unidad de salida de forma:

$$\delta_{c_i} = t_i - o_{c_i}$$

Donde o_{ci} es la activación de la neurona de salida i y δ_{ci} el error de la neurona de salida i . Con base en este error, para cada unidad de regla activada se calcula el grado de cumplimiento de la siguiente forma:

$$\delta_R = o_R(1 - o_R) \sum_{c \in U_3} w(R, c) \delta c$$

La función de pertenencia que es responsable del grado de cumplimiento es identificada con la siguiente condición:

$$W(x', R)(a_{x'}) = \min_{x \in U_1} \{W(x, R)(o_x)\}$$

Es decir se busca el conjunto que tenga menor función de pertenencia con el patrón que se está propagando y solamente este conjunto es adaptado en función del error modificándolo con los siguientes valores:

$$\delta b = \sigma \cdot \delta R \cdot (c - a) \cdot \text{sgn}(o_{x'} - b)$$

$$\delta a = -\sigma \cdot \delta R \cdot (c - a) + \delta b$$

$$\delta c = \sigma \cdot \delta R \cdot (c - a) + \delta b$$

Un conjunto Fuzzy es modificado solamente sí este no viola ninguno de las restricciones impuestas por el usuario, como lo podrían ser:

- Los conjuntos Fuzzy no deben traslapar más de cierto grado.
- Un conjunto Fuzzy no debe sobrepasar el vecino.
- Los conjuntos Fuzzy deben ser simétricos. etc.

6.3 NEFCLASS-Q.

NEFCLASS-Q fue desarrollado teniendo como base el modelo de un Perceptrón Fuzzy de 3 capas hecho por los Doctores Rudolf Kruse y Detleff Nauck. Esta herramienta software tiene dos formas de funcionar, la primera es por medio de una interfaz, la cual facilitara el proceso de creación de la Red Neuro-Fuzzy y la

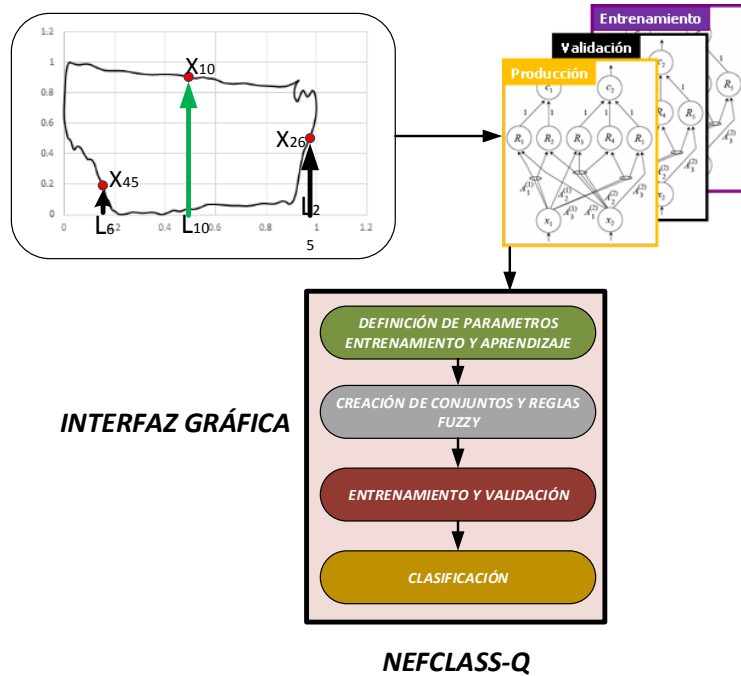
segunda es utilizar las librerías de NEFCLASS-Q para crear un agente inteligente por medio de un pequeño código en C++.

NEFCLASS-Q crea una Red Neuro-Fuzzy que es entrenada a partir de un conjunto de datos almacenados en un archivo de texto, donde cada patrón de datos pertenece a una clase determinada. El primer paso importante que hace la herramienta es la creación de Reglas Fuzzy por medio de la propagación de los datos de entrenamiento, posteriormente estas reglas son optimizadas por medio del aprendizaje de los parámetros de los conjuntos Fuzzy que conforman estas reglas. Después de que el sistema es entrenado se prosigue a la etapa de validación del entrenamiento, donde se puede evaluar el entrenamiento de la Red, en caso de que los resultados se quieran mejorar se podrá cambiar algunos parámetros de entrenamiento para obtener mejores resultados. Después de esto pasamos a la etapa de clasificación donde los patrones del archivo de clasificación son cargados y clasificados.

6.3.1 Creación de archivos.

NEFCLASS-Q no interpreta los valores de amplitud discretizados de los dinagramas directamente. Existen formatos que facilitan la lectura de esta información, ver figura 78.

Figura 78. Componentes NEFCLASS-Q



Para iniciar el proceso de entrenamiento, los dinagramas de fondo deben ser almacenados en tres archivos a saber:

- Archivo de Entrenamiento.
- Archivo de Validación.
- Archivo de Operación.

Para dar inicio a la creación de los archivos, se debe tener en cuenta la forma genérica en la cual se presentan.

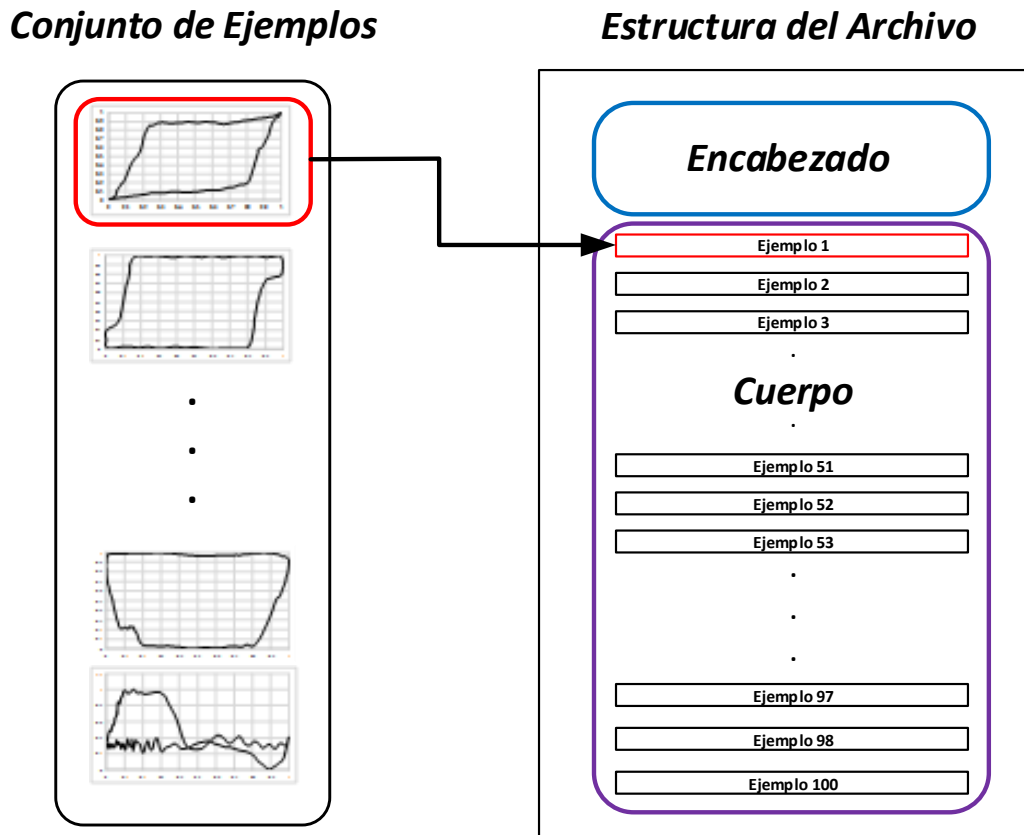
En sí, un archivo de entrenamiento, validación u operación, presenta, (ver figura 79):

1. Un encabezado. Este contiene información que describe, dependiendo de su tipo, los parámetros necesarios para cada proceso en particular (Entrenamiento, Validación y Clasificación).

Por ejemplo, como se verá más adelante, el archivo de entrenamiento en su encabezado indica la cantidad de ejemplos de dinagramas, las variables de entrada de cada dinagrama, el número de clases, entre otra información.

2. Un cuerpo. Este contiene los valores de amplitud de los dinagramas discretizados almacenados en modo de texto. En esta sección se organizan los conjuntos de ejemplos para cada proceso (Entrenamiento, Validación y Operación). El orden en el cual se ordena cada valor de amplitud es mostrado en la figura 81, siguiendo a lo estipulado en el proceso de discretización, de derecha a izquierda en carrera ascendente y de izquierda a derecha en carrera descendente.

Figura 79. Encabezado y Cuerpo de un archivo.

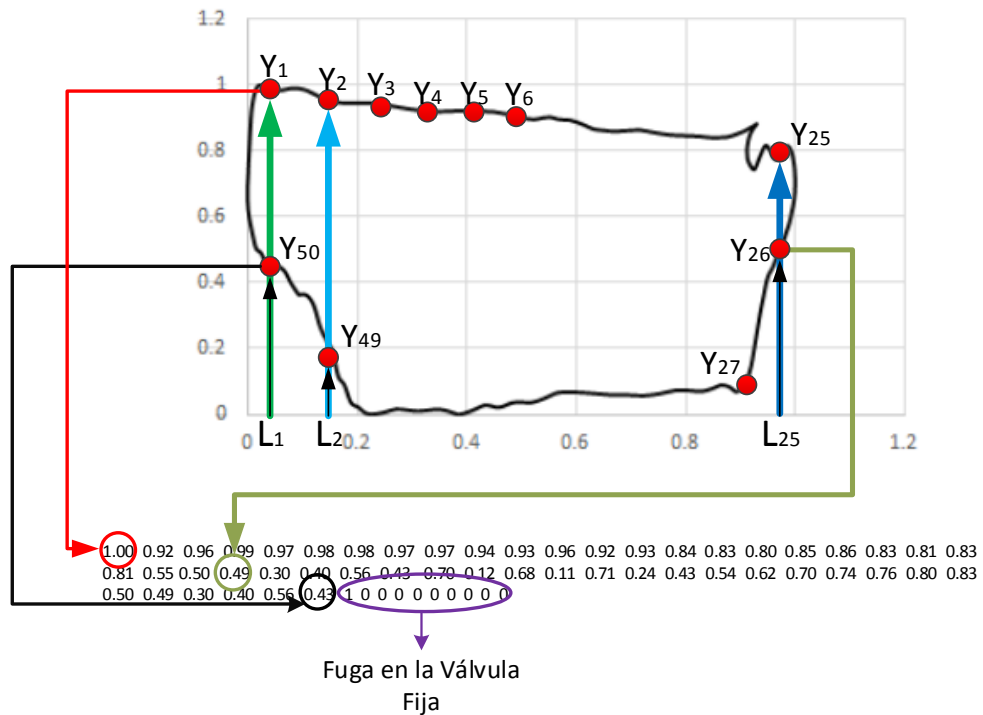


En el cuerpo, se organizan en modo de texto cada valor de amplitud es parado por dos espacios, indicando al final la clase a la que pertenece ese patrón (ver figura 80), exceptuando el archivo de Operación, ya que este solo presenta el patrón mas no la clase.

Figura 80. Organización de cada Patrón

<i>Valores Diacretizados</i>	<i>Tipo de Falla</i>
0.58 0.89 0.11 ... 0.96 0.92 0.93 0.94 ... 0.91 0.90 0.91 ... 0.92 0.91 0.96 0.96	1 0 0 0 0 0 0 0 0

Figura 81. Descripción numérica en un archivo de texto.



La codificación de las fallas al final de cada patrón se enseña en el capítulo 7 (pruebas y resultados). A continuación se presenta cada archivo, con su encabezado y cuerpo respectivamente, no sin antes describirlos muy brevemente.

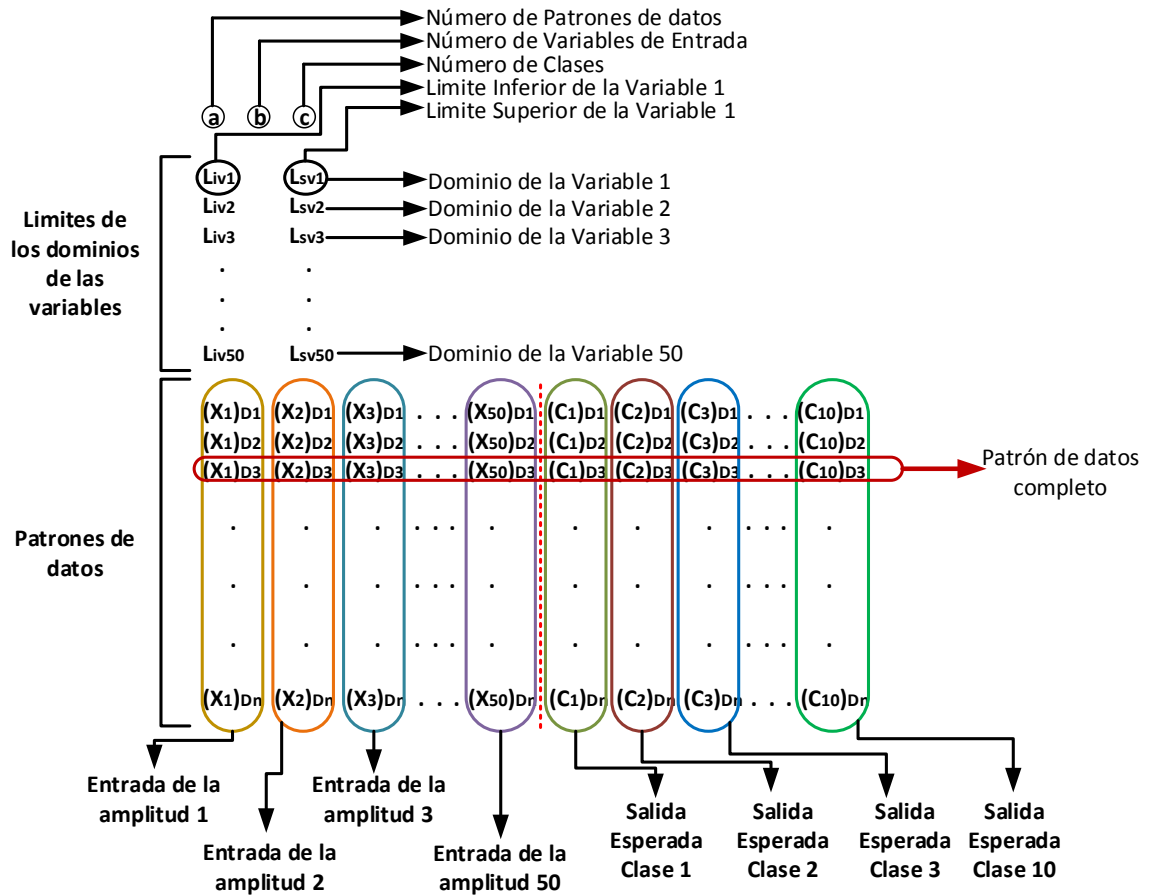
6.3.1.1 Descripción General de Cada Archivo.

El archivo de entrenamiento proporciona el conocimiento a la Red por medio de ejemplos de fallas contenidos en su estructura interna modificando así la Red del software. El archivo de validación es el soporte para evaluar el conocimiento adquirido, indica que tan buena esta estrenada la Red. Y por último el archivo de clasificación, donde se ubican patrones que simulan fallas obtenidas en producción.

Se crearon 10 dinagramas por cada falla para entrenamiento. Posteriormente para validar el aprendizaje se crearon 6 dinagramas por falla y por último para simular operación en línea se crearon otros 6 dinagramas por falla.

6.3.1.2 ARCHIVO DE ENTRENAMIENTO.

Figura 82. Estructura del archivo de Entrenamiento genérica.



La figura 82 muestra de forma general como deben ordenar de los datos de entrenamiento para suministrarlos a NEFCLASS-Q mediante un archivo de texto.

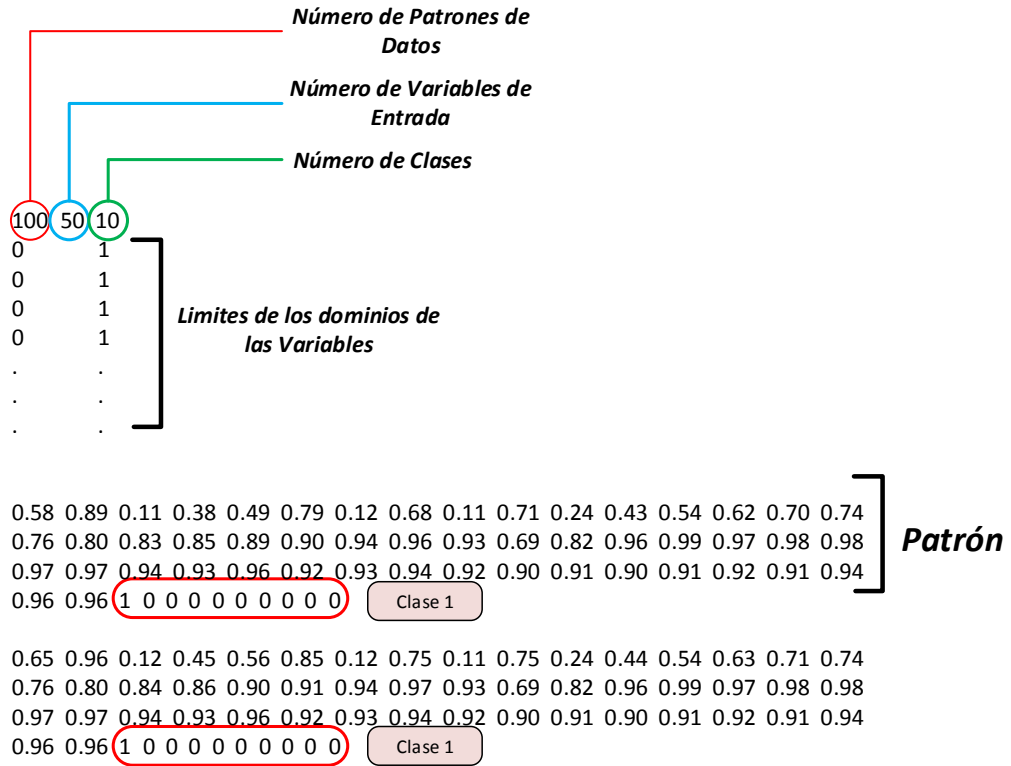
- Encabezado:** En la primera línea el valor **a** indica el número de patrones de datos que hay en el archivo, para el caso de este proyecto, en entrenamiento se crearon 100 dinagramas es decir $a = 100$. El valor **b** indica el número de variables de entrada del problema, por lo tanto $b = 50$ (Máximo número de entradas) y el valor de **c** indica el número de clases del problema, entonces $c = 10$ obedeciendo a las 10 clases de falla. En el siguiente grupo de líneas deben ir los dominios de cada una de las

variables representados por L_{iv1} que significa límite inferior de la variable 1 y L_{sv1} que significa límite superior de la variable 1, el número de líneas de este grupo debe ser igual al valor **b**.

- **Cuerpo:** El siguiente grupo de líneas resaltadas en color, son los patrones de datos. Cada patrón de datos está dividido en dos partes, en la parte izquierda van las entradas y en la parte derecha van las salidas esperadas de cada patrón. La nomenclatura de las entradas, es por ejemplo $(X_1)_{D1}$ significa entrada del patrón 1 a la variable 1 y $(X_2)_{D1}$ significa entrada del patrón 1 a la variable 2. Por otro lado en la sección de salidas $(C_1)_{D1}$ significa salida esperada del patrón 1 en la clase 1, estas salidas deben tener valores de 1 o 0, la sección de salida debe tener tantas columnas como diga el valor **c**.

A continuación se muestra parte del archivo de entrenamiento creado para entrenar la Red, en esta investigación.

Figura 83. Archivo de Entrenamiento creado y empleado.

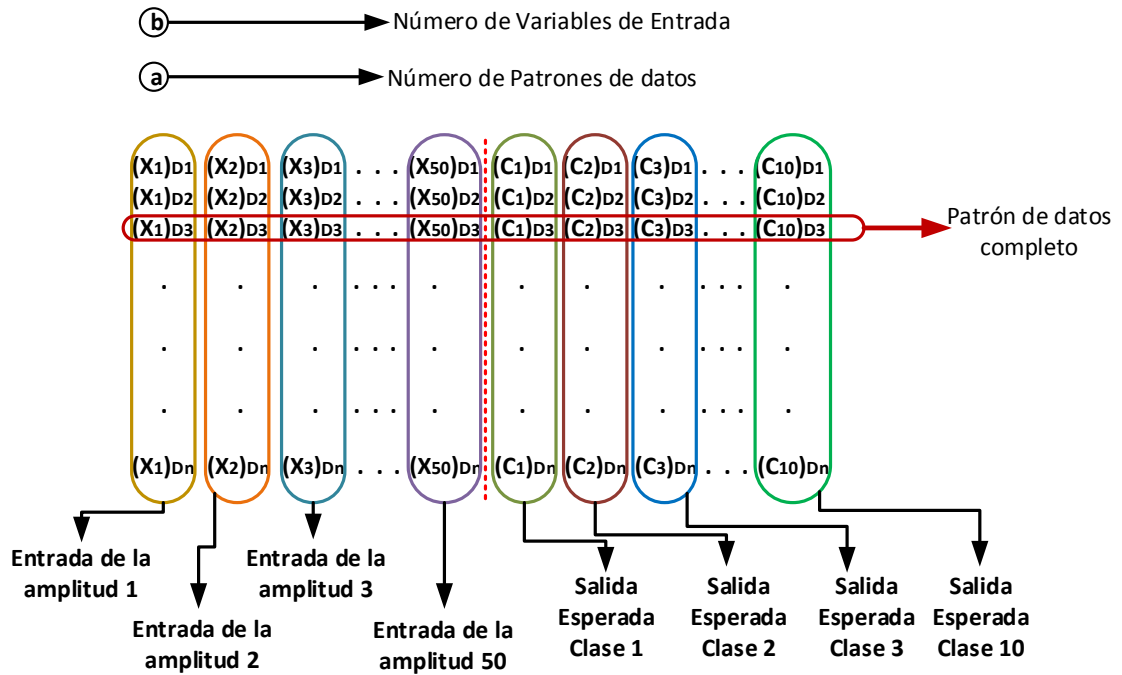


En la figura 83 no se muestra la totalidad del archivo por su extensión, pero se muestra las partes fundamentales explicadas en el archivo de entrenamiento genérico.

6.3.1.3 ARCHIVO DE VALIDACIÓN.

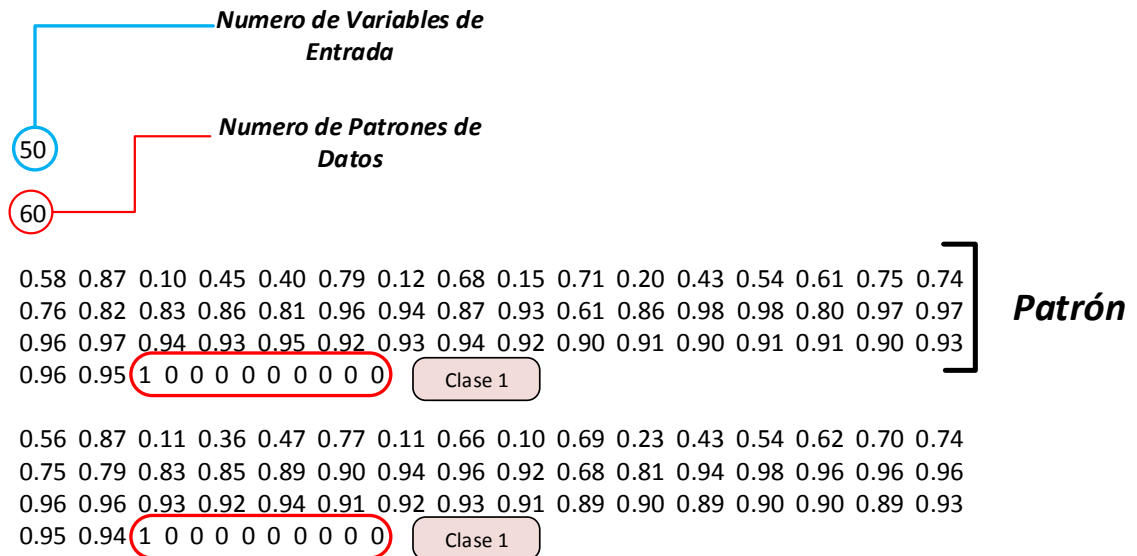
Este archivo es similar al de entrenamiento y tiene la estructura de la figura 84. En esta figura la primera línea contiene el valor **b** (50) y determina el número de variables de entrada, en la segunda línea contiene el valor **a** (100) y determina el número de patrones. En las siguientes líneas se sitúan los patrones de datos que tiene el mismo orden que en el archivo de entrenamiento.

Figura 84. Archivo de validación genérico.



A continuación se muestra parte del archivo de validación creado para validación de la Red entrenada.

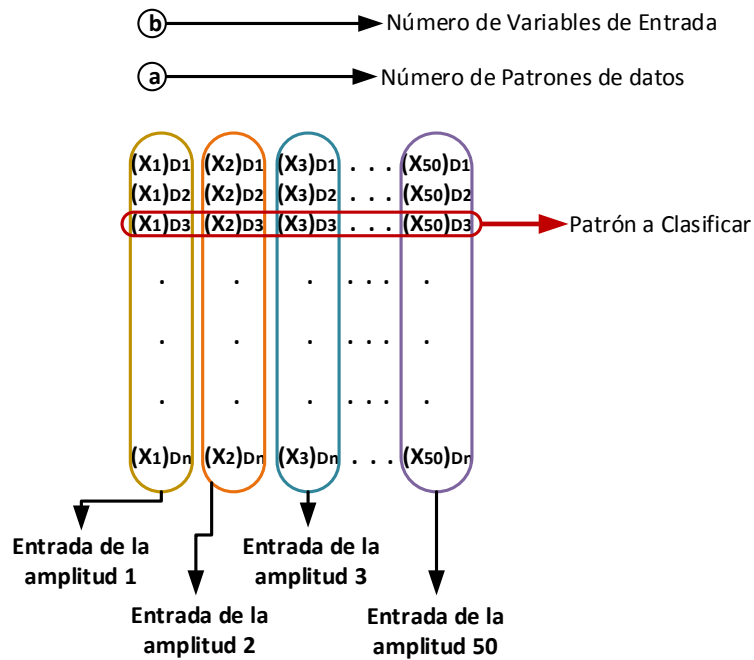
Figura 85. Archivo de validación creado.



6.3.1.4 ARCHIVO DE CLASIFICACIÓN U OPERACIÓN

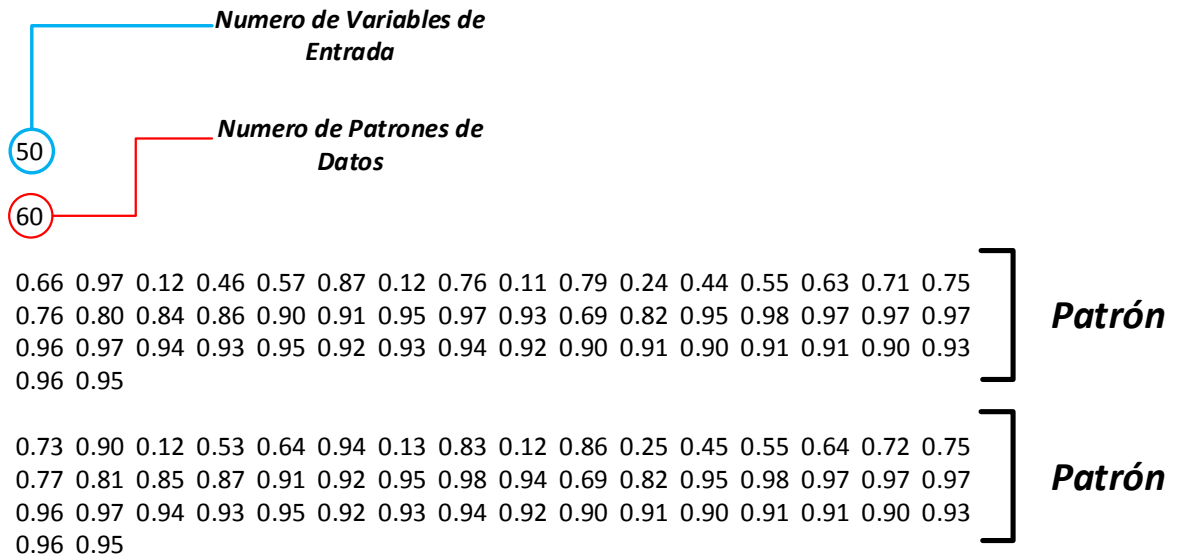
El último archivo es el de operación, en este archivo se ubican los patrones de datos que se quieren clasificar, este archivo tiene la estructura de la figura 86.

Figura 86. Archivo de Operación genérico.



Como podemos ver, la única diferencia que tiene con respecto al archivo de validación de la figura 84, es que los patrones de datos no tienen designada la clase. A continuación se muestra parte del archivo de clasificación creado.

Figura 87. Archivo de Operación creado.



6.3.2 Organización de archivos.

En total se crearon 2 tipos de archivos para entrenamiento, validación y operación, queriendo averiguar si el orden en el cual se organizan los patrones afecta su desempeño, sobre todo en entrenamiento.

6.3.2.1 Archivo Tipo 1.

Los patrones se organizaron en orden ascendente, es decir, patrón 1 pertenece a clase 1, patrón 2 pertenece a clase 2, patrón 3 pertenece a clase 3,..., etc.

Figura 88. Archivo de entrenamiento tipo 1.

100 50 10	
0 1	
0 1	
0 1	
0 1	
0 1	
.	.
.	.
.	.
0 1	
0 1	
0.58 0.89 0.11 0.38 0.49 0.79 0.12 0.68 0.11 0.71 0.24 0.43 0.54 0.62 0.70 0.74	} Patrón 1
0.76 0.80 0.83 0.85 0.89 0.90 0.94 0.96 0.93 0.69 0.82 0.96 0.99 0.97 0.98 0.98	
0.97 0.97 0.94 0.93 0.96 0.92 0.93 0.94 0.92 0.90 0.91 0.90 0.91 0.92 0.91 0.94	
0.96 0.96 1 0 0 0 0 0 0 0 0 0	
Clase 1	
0.20 0.36 0.32 0.35 0.36 0.46 0.42 0.45 0.41 0.48 0.24 0.44 0.54 0.63 0.71 0.74	} Patrón 2
0.76 0.80 0.84 0.86 0.90 0.91 0.94 0.97 0.93 0.69 0.82 0.96 0.99 0.97 0.98 0.98	
0.97 0.97 0.44 0.43 0.46 0.42 0.43 0.54 0.52 0.50 0.51 0.50 0.51 0.42 0.41 0.44	
0.46 0.46 0 1 0 0 0 0 0 0 0 0	
Clase 2	
0.93 0.94 0.91 0.93 0.54 0.94 0.12 0.73 0.91 0.76 0.24 0.44 0.54 0.63 0.71 0.74	
0.76 0.80 0.84 0.86 0.90 0.91 0.94 0.97 0.93 0.70 0.83 0.96 0.99 0.98 0.98 0.98	
0.97 0.98 0.25 0.24 0.26 0.23 0.24 0.25 0.22 0.21 0.22 0.21 0.22 0.22 0.31 0.34	
0.36 0.36 0 0 1 0 0 0 0 0 0 0	
Clase 3	
0.15 0.16 0.10 0.15 0.36 0.16 0.10 0.55 0.95 0.58 0.22 0.42 0.52 0.61 0.69 0.72	
0.74 0.78 0.82 0.84 0.88 0.89 0.92 0.95 0.91 0.70 0.83 0.96 0.99 0.98 0.98 0.98	
0.97 0.98 0.15 0.14 0.16 0.13 0.14 0.25 0.22 0.21 0.22 0.21 0.22 0.12 0.21 0.00	
0.16 0.16 0 0 0 1 0 0 0 0 0 0	
Clase 4	
0.46 0.47 0.41 0.46 0.47 0.77 0.11 0.66 0.10 0.69 0.23 0.43 0.54 0.62 0.70 0.74	
0.75 0.79 0.83 0.85 0.89 0.90 0.94 0.96 0.92 0.69 0.82 0.96 0.99 0.97 0.98 0.98	
0.97 0.97 0.98 0.73 0.86 0.82 0.83 0.74 0.82 0.80 0.81 0.80 0.71 0.72 0.81 0.74	
0.76 0.76 0 0 0 0 1 0 0 0 0 0	
Clase 5	
0.88 0.89 0.82 0.88 0.89 0.89 0.13 0.78 0.12 0.81 0.25 0.44 0.55 0.63 0.71 0.75	
0.77 0.81 0.84 0.86 0.90 0.91 0.95 0.97 0.94 0.70 0.83 0.97 1.00 0.98 0.99 0.99	
0.98 0.98 0.55 0.54 0.67 0.63 0.64 0.55 0.53 0.51 0.62 0.41 0.52 0.55 0.42 0.45	
0.57 0.57 0 0 0 0 0 1 0 0 0 0	
Clase 6	

6.3.2.2 Archivo Tipo 2.

Los patrones se organizaron en forma grupal, es decir, patrón 1, patrón 2, patrón 3,... hasta patrón 5, pertenece a clase 1, patrón 1, patrón 2, patrón 3,... hasta patrón 5, pertenece a clase 2, y así con los demás.

Figura 89. Archivo de entrenamiento tipo 2.

100 50 10	
0 1	
0 1	
0 1	
0 1	
0 1	
.	.
.	.
.	.
0 1	
0 1	

0.58 0.89 0.11 0.38 0.49 0.79 0.12 0.68 0.11 0.71 0.24 0.43 0.54 0.62 0.70 0.74	} Patrón 1
0.76 0.80 0.83 0.85 0.89 0.90 0.94 0.96 0.93 0.69 0.82 0.96 0.99 0.97 0.98 0.98	
0.97 0.97 0.94 0.93 0.96 0.92 0.93 0.94 0.92 0.90 0.91 0.90 0.91 0.92 0.91 0.94	
0.96 0.96 1 0 0 0 0 0 0 0 0 0	

Clase 1

0.65 0.96 0.12 0.45 0.56 0.86 0.12 0.75 0.11 0.78 0.24 0.44 0.54 0.63 0.71 0.74	} Patrón 2
0.76 0.80 0.84 0.86 0.90 0.91 0.94 0.97 0.93 0.69 0.82 0.96 0.99 0.97 0.98 0.98	
0.97 0.97 0.94 0.93 0.96 0.92 0.93 0.94 0.92 0.90 0.91 0.90 0.91 0.92 0.91 0.94	
0.96 0.96 1 0 0 0 0 0 0 0 0 0	

Clase 1

0.63 0.94 0.11 0.43 0.54 0.84 0.12 0.73 0.11 0.76 0.24 0.44 0.54 0.63 0.71 0.74	
0.76 0.80 0.84 0.86 0.90 0.91 0.94 0.97 0.93 0.70 0.83 0.96 0.99 0.98 0.98 0.98	
0.97 0.98 0.95 0.94 0.96 0.93 0.94 0.95 0.92 0.91 0.92 0.91 0.92 0.92 0.91 0.94	
0.96 0.96 1 0 0 0 0 0 0 0 0 0	Clase 1

0.45 0.76 0.10 0.25 0.36 0.66 0.10 0.55 0.95 0.58 0.22 0.42 0.52 0.61 0.69 0.72	
0.74 0.78 0.82 0.84 0.88 0.89 0.92 0.95 0.91 0.70 0.83 0.96 0.99 0.98 0.98 0.98	
0.97 0.98 0.95 0.94 0.96 0.93 0.94 0.95 0.92 0.91 0.92 0.91 0.92 0.92 0.91 0.94	
0.96 0.96 1 0 0 0 0 0 0 0 0 0	Clase 1

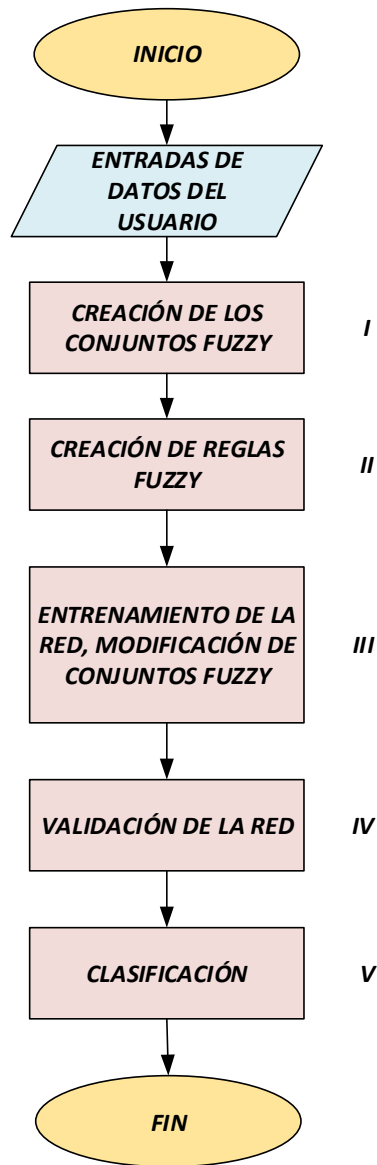
0.56 0.87 0.11 0.36 0.47 0.77 0.11 0.66 0.10 0.69 0.23 0.43 0.54 0.62 0.70 0.74	
0.75 0.79 0.83 0.85 0.89 0.90 0.94 0.96 0.92 0.69 0.82 0.96 0.99 0.97 0.98 0.98	
0.97 0.97 0.94 0.93 0.96 0.92 0.93 0.94 0.92 0.90 0.91 0.90 0.91 0.92 0.91 0.94	
0.96 0.96 1 0 0 0 0 0 0 0 0 0	Clase 1

0.68 0.99 0.12 0.48 0.59 0.89 0.13 0.78 0.12 0.81 0.25 0.44 0.55 0.63 0.71 0.75	
0.77 0.81 0.84 0.86 0.90 0.91 0.95 0.97 0.94 0.70 0.83 0.97 1.00 0.98 0.99 0.99	
0.98 0.98 0.95 0.94 0.97 0.93 0.94 0.95 0.93 0.91 0.92 0.91 0.92 0.93 0.92 0.95	
0.97 0.97 0 1 0 0 0 0 0 0 0 0	Clase 2

6.3.3 Procesos NEFCLASS-Q

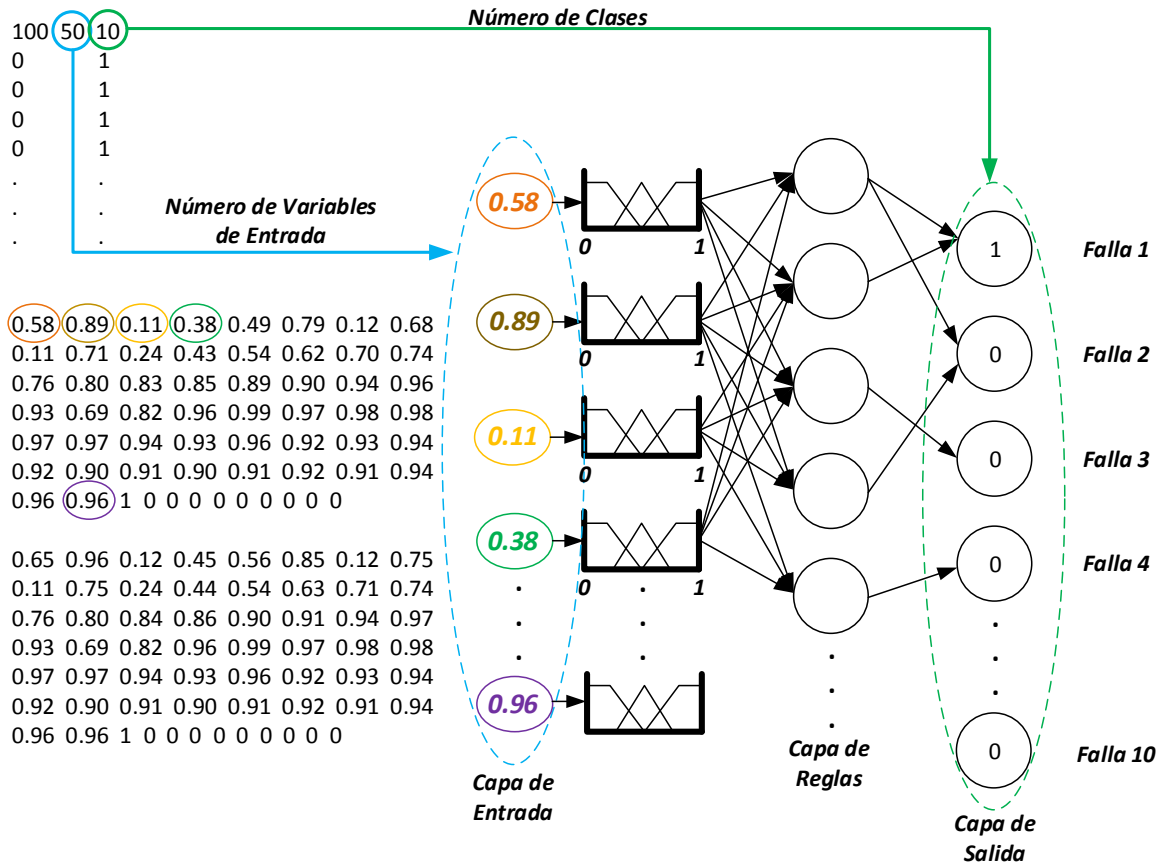
Con el fin de explicar cada uno de los procesos que realiza NEFCLASS-Q en detalle, se va utilizar el archivo de entrenamiento creado Tipo 1, figura 88, como ejemplo de clasificación.

Figura 90. Procesos NEFCLASS-Q.



Cuando un archivo de entrenamiento es cargado, automáticamente se definen el número de neuronas de entrada que representan el número de variables en el archivo, igualmente se definen el número de neuronas de salida que representan el número de clases en el archivo como se puede ver en la figura 91.

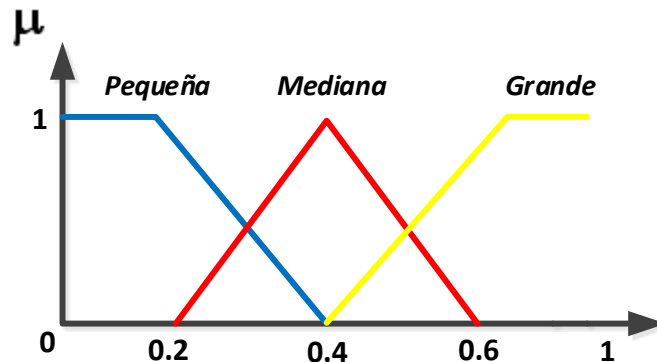
Figura 91. Creación de la Red.



6.3.3.1 Creación de los conjuntos Fuzzy (I)

Los pesos sinápticos entre la capa de entrada y la capa oculta son representados por los conjuntos Fuzzy, ellos permiten definir los valores que se están propagando en términos lingüísticos. El número de conjuntos Fuzzy para cada unidad de entrada es definido por el usuario, y la forma de estos es definida inicialmente por NEFCLASS-Q de tal manera que queden equitativamente distribuidos en el dominio de cada variable en el archivo de entrenamiento. Por ejemplo vamos a tomar el archivo de entrenamiento de la figura 88, y definiremos los conjuntos de la variable 1 (Amplitud X_1), cuyo dominio se encuentra entre 0 y 1, y suponemos que el número de conjuntos definidos por el usuario fue de 3, teniendo esto NEFCLASS-Q puede representar los conjuntos Fuzzy para la variable 1 como en la figura 92.

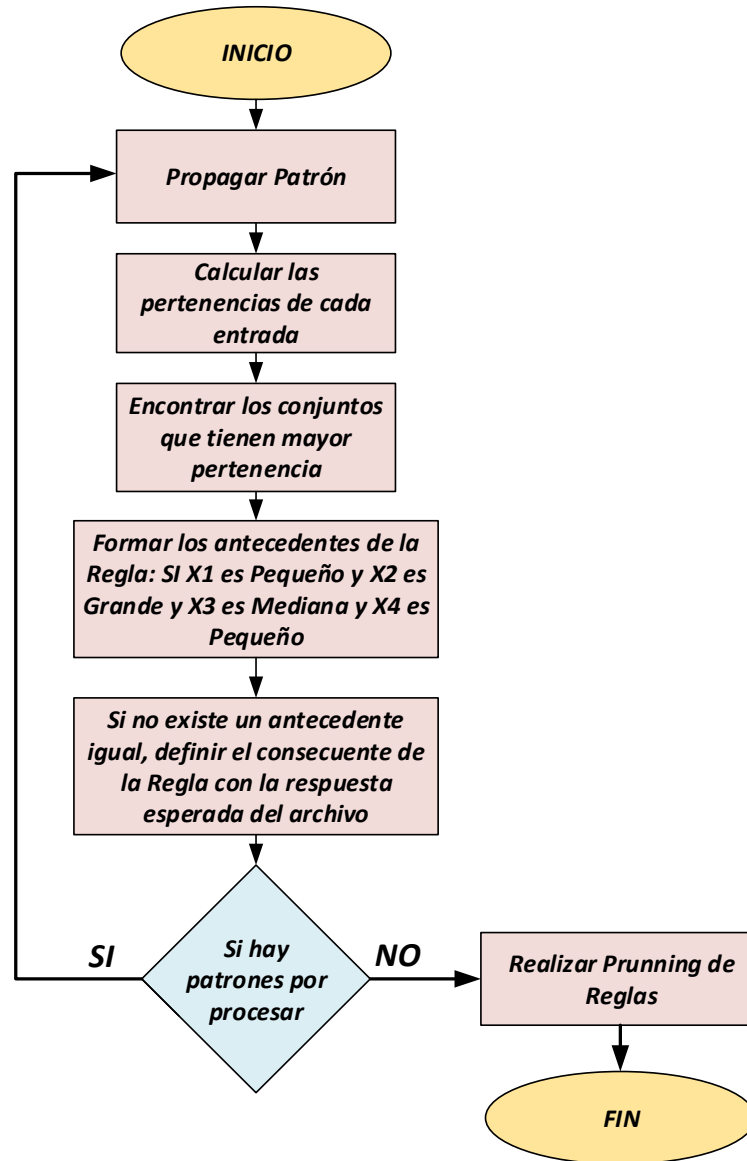
Figura 92. Variable 1 Fuzzificada.



6.3.3.2 Creación de Reglas (II).

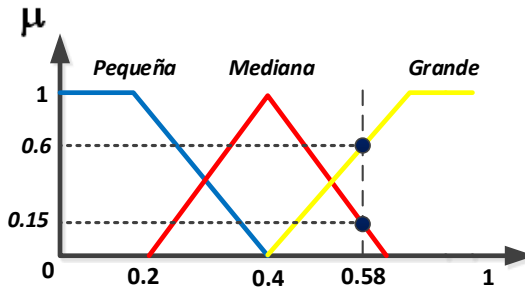
Para definir el número de neuronas de la capa oculta que representan las Reglas Fuzzy, NEFCLASS-Q tiene un proceso de creación de reglas que está explicada en la figura 93. El número de neuronas de la capa oculta va a ser igual al número de reglas creadas en el proceso o un valor máximo que es definido por el usuario. Este proceso es muy importante, ya que es en el que la Red Neuro-Fuzzy adquiere el conocimiento haciendo propagar los patrones de datos del archivo de entrenamiento. Cuando NEFCLASS-Q propaga un patrón, calcula las pertenencias de cada una de las variables Fuzzy a sus respectivos conjuntos, cuando termina de calcular todas las pertenencias, NEFCLASS-Q escoge un conjunto por cada variable, teniendo en cuenta que el conjunto tenga la mayor pertenencia. Cuando ya se han etiquetado lingüísticamente cada variable Fuzzy, estas etiquetas se emplean para formar el antecedente de la regla, por último se verifica que este antecedente exista, de no ser así, se le asigna el consecuente con la respuesta esperada y esta regla es guardada en la base de conocimiento de la Red Neuro-Fuzzy, este proceso NEFCLASS-Q lo realiza cuando está propagando cada patrón de datos del archivo de entrenamiento.

Figura 93. Proceso de creación de Reglas.



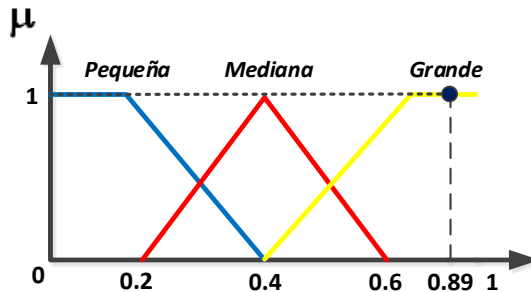
Para mayor claridad de cómo se crea una Regla, se va a propagar el patrón número uno (las primeras cuatro entradas) de la figura 88. Los valores de amplitud del patrón son 0.58, 0.89, 0.11, 0.38 que ingresan a las neuronas de entrada como se ve en la figura 91. El siguiente paso es calcular las pertenencias a los conjuntos de las variables, para la variable Fuzzy 1 el valor de amplitud ingresado es 0.58, la pertenencia a *Grande* es 0.6, a *Mediana* es 0.15, a *Pequeña* es 0 como se puede ver en la figura 94.

Figura 94. Variable Fuzzy X_1 .



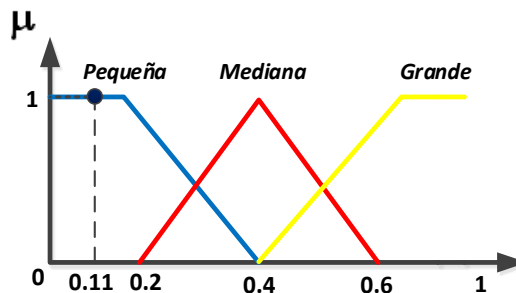
Para la variable Fuzzy 2 el valor de amplitud ingresado es 0.89, la pertenencia a *Pequeña* es 0, a *Mediana* es 0, a *Grande* es 1, como se puede ver en la figura 95.

Figura 95. Variable Fuzzy X_2 .



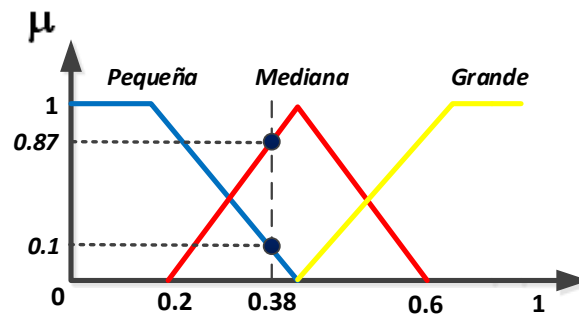
Para la variable Fuzzy 3 el valor de amplitud ingresado es 0.11, la pertenencia a *Pequeño* es 1.0, a *Mediana* es 0 y a *Grande* es 0, como se puede ver en la figura 96.

Figura 96. Variable Fuzzy X_3 .



Para la variable Fuzzy 4 el valor de amplitud ingresado es 0.38, la pertenencia a *Pequeña* es 0.1, a *Mediana* es 0.87 y a *Grande* es 0, como se puede ver en la figura 97.

Figura 97. Variable Fuzzy X_4 .



El siguiente paso es seleccionar los conjuntos que tiene mayor pertenencia en cada dominio, para mejor entendimiento se presenta la tabla 3 que tiene las pertenencias a los conjuntos de todas las variables y presenta las mayores pertenencias de cada dominio.

Tabla 3. Conjuntos que poseen mayor pertenencia.

PERTENENCIAS A LOS CONJUNTOS			
VARIABLES	PEQUEÑA	MEDIANA	GRANDE
X_1	0	0.15	0.6
X_2	0	0	1
X_3	1	0	0
X_4	0.1	0.87	0
.	.	.	.
.	.	.	.
.	.	.	.

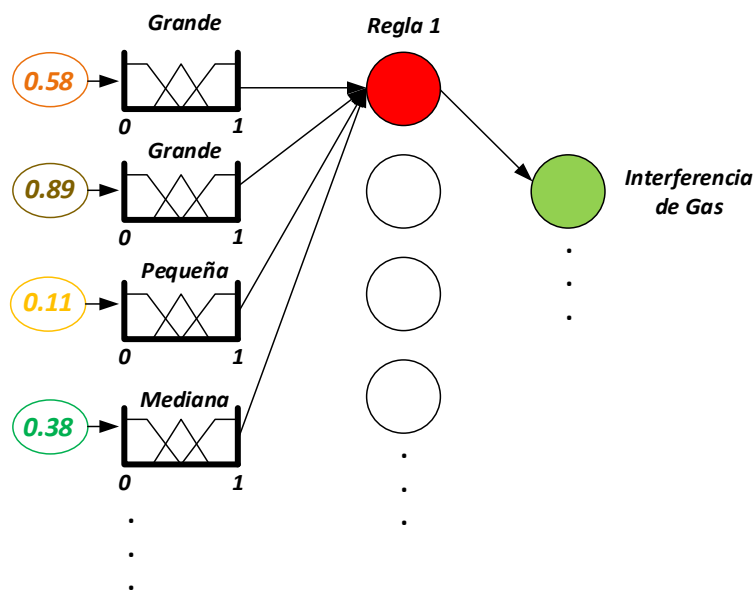
Como se puede ver en la tabla 3, el conjunto que tiene mayor pertenencia para la variable 1 (X_1) es *Grande*, para la variable 2 (X_2) es *Grande*, para la variable 3 (X_3) es *Pequeña*, para la variable 4 (X_4) es *Mediana*. Con esto se puede formar el antecedente de la regla que queda de la siguiente forma:

Si X_1 es *Grande* Y X_2 es *Grande* Y X_3 es *Pequeña* Y X_4 es *Mediana* Y...

Como esta regla es la primera que se crea y no existe, entonces se define el consecuente que está en la clase esperada del archivo de entrenamiento, en este caso es clase 1 que indica “*Interferencia de Gas*”. La regla queda finalmente como:

SI *X1* es *Grande* Y *X2* es *Grande* Y *X3* es *Pequeña* Y *X4* es *Mediana* Y...
ENTONCES es *Interferencia de Gas*.

Figura 98. Regla creada en la Red Neuro-Fuzzy.



De esta forma se crea una Regla Fuzzy, así mismo se propagan las variables y patrones del archivo de entrenamiento y se crean todas las Reglas y queda definida completamente la *base de conocimiento* de la Red Neuro-Fuzzy.

Con la finalidad de que el conocimiento sea más concreto para el usuario, se hace un proceso de *Pruning* de reglas, en el cual, puede dejar una base de reglas que contenga las reglas más importantes, para esto hay 3 métodos:

- El método de Todas las Reglas o Simple que deja la base de conocimiento tal como se creó, con el mismo número de reglas.

- El método de las Mejores Reglas que deja determinado número de reglas ha pedido del usuario, y se escogen las reglas que más han tenido activación cuando se propagan los patrones.
- El método de la Mejores Reglas por clase se limita a dejar las reglas que hayan tenido más activación para cada clase, donde el número de reglas por clase es definido por el usuario.

Con el *Pruning* de Reglas, se finaliza el proceso de creación de Reglas según el diagrama de flujo de la figura 93.

6.3.3.3 Entrenamiento de la Red (III).

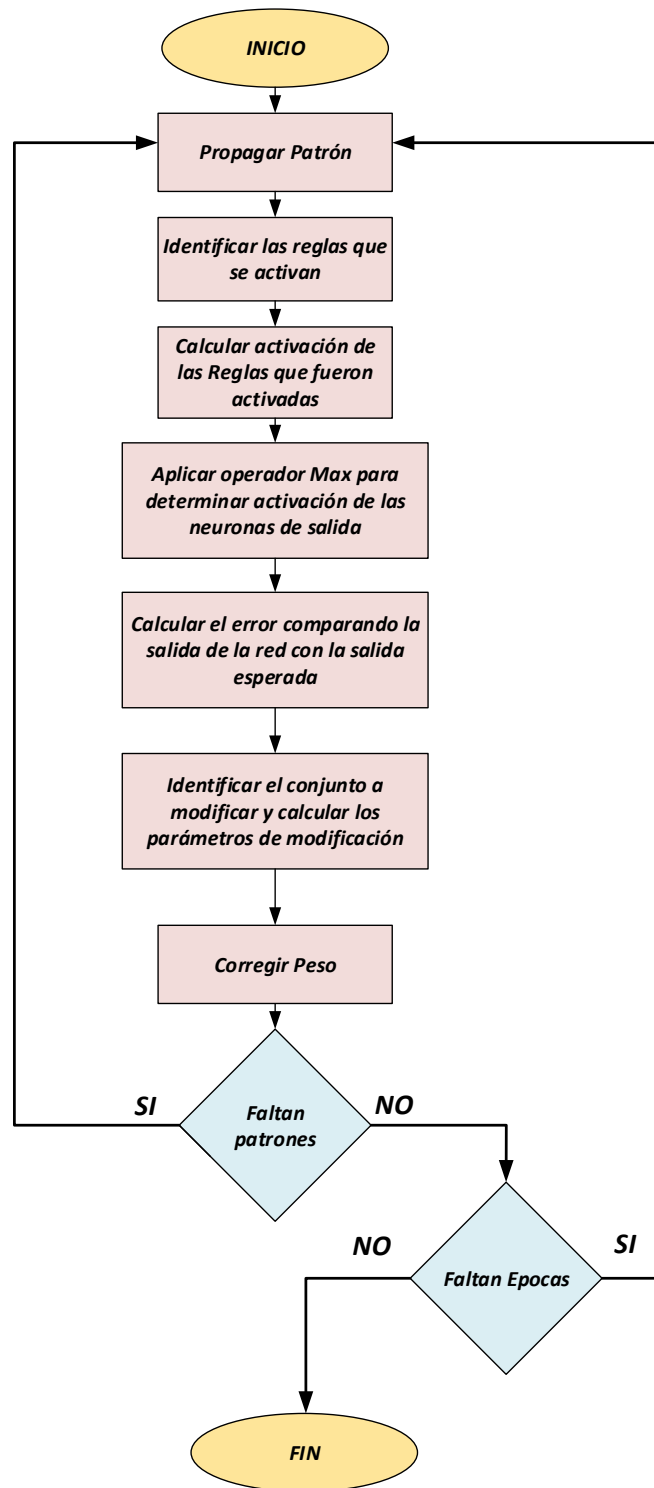
Teniendo la *base del conocimiento* definida, se prosigue a realizar el entrenamiento de la Red, en el que se hacen propagar los patrones de datos del archivo de entrenamiento para que la Red proporcione una respuesta, esta es comparada con la respuesta deseada que se encuentra en el archivo de entrenamiento para generar un error, este error es retropropagado para modificar los pesos de acuerdo a las formulas planteadas en la sección de entrenamiento 6.2.3, este proceso es explicado en la figura 99.

Para demostrar cómo se realiza el proceso de entrenamiento de la Red Neuro-Fuzzy se supone una base de conocimiento creada del proceso anterior (Creación de Reglas) como el de la tabla 4, con 4 reglas y con los mismos conjuntos Fuzzy mostrados en las figuras 94, 95, 96, 97.

Tabla 4. Base de conocimiento.

	X1	X2	X3	X4	...	Consecuente
1	Grande	Grande	Pequeña	Mediana	...	Interferencia de Gas
2	Mediana	Mediana	Grande	Grande	...	Golpe de Fluido
3	Grande	Grande	Grande	Grande	...	Fuga en la V. Fija
4	Pequeña	Pequeña	Pequeña	Mediana	...	Buen llenado
...					...	

Figura 99. Proceso de Entrenamiento.



Se prosigue a propagar nuevamente el patrón número uno del archivo de entrenamiento para calcular cada una de las pertenencias a los conjuntos, para mayor entendimiento se presenta la tabla 5, que contiene las pertenencias calculadas a los conjuntos. Como se puede ver en la tabla 5, las pertenencias que están resaltadas son las que tienen valor mayor a cero, y juegan un papel importante en este proceso de entrenamiento.

Tabla 5. Conjunto que tiene mayor a cero.

PERTENENCIAS A LOS CONJUNTOS			
VARIABLES	PEQUEÑA	MEDIANA	GRANDE
<i>X1</i>	0	0.15	0.6
<i>X2</i>	0	0	1
<i>X3</i>	1	0	0
<i>X4</i>	0.1	0.87	0
.	.	.	.
.	.	.	.
.	.	.	.

- Para la variable ***X1***: se activan los conjuntos *Mediana* y *Grande*.
- Para la variable ***X2***: se activa el conjunto *Grande*.
- Para la variable ***X3***: se activa el conjunto *Pequeña*.
- Para la variable ***X4***: se activan los conjuntos *Pequeña* y *Mediana*.

Basándose en la tabla 5 se tienen cuatro posible antecedentes de reglas. La primera regla (Regla A) se puede generar con los valores resaltados de la tabla 6.

Tabla 6. Formación Regla A.

PERTENENCIAS A LOS CONJUNTOS			
VARIABLES	PEQUEÑA	MEDIANA	GRANDE
X1	0	0.15	0.6
X2	0	0	1
X3	1	0	0
X4	0.1	0.87	0
.	.	.	.
.	.	.	.
.	.	.	.

Por lo tanto el antecedente de la Regla A quedara de la siguiente forma:
 Si X_1 es Mediana Y X_2 es Grande Y X_3 es Pequeña Y X_4 es Pequeña...

La segunda regla (Regla B) se puede generar con los conjuntos que tienen las pertenencias resaltadas de la tabla 7.

Tabla 7. Formación de la Regla B.

PERTENENCIAS A LOS CONJUNTOS			
VARIABLES	PEQUEÑA	MEDIANA	GRANDE
X1	0	0.15	0.6
X2	0	0	1
X3	1	0	0
X4	0.1	0.87	0
.	.	.	.
.	.	.	.
.	.	.	.

Por lo tanto el antecedente de la Regla B quedara de la siguiente forma:
 Si X_1 es Mediana Y X_2 es Grande Y X_3 es Pequeña Y X_4 es Mediana...

La tercera regla (Regla C) se puede generar con los conjuntos que tienen las pertenencias resaltadas de la tabla 8.

Tabla 8. Formación de la Regla C.

PERTENENCIAS A LOS CONJUNTOS			
VARIABLES	PEQUEÑA	MEDIANA	GRANDE
X1	0	0.15	0.6
X2	0	0	1
X3	1	0	0
X4	0.1	0.87	0
.	.	.	.
.	.	.	.
.	.	.	.

Por lo tanto el antecedente de la Regla C quedara de la siguiente forma:

SI X_1 es Grande Y X_2 es Grande Y X_3 es Pequeña Y X_4 es Pequeña...

La cuarta regla (Regla D) se puede generar con los conjuntos que tienen las pertenencias resaltadas de la tabla 9.

Tabla 9. Formación de la Regla D.

PERTENENCIAS A LOS CONJUNTOS			
VARIABLES	PEQUEÑA	MEDIANA	GRANDE
X1	0	0.15	0.6
X2	0	0	1
X3	1	0	0
X4	0.1	0.87	0
.	.	.	.
.	.	.	.
.	.	.	.

Por lo tanto el antecedente de la Regla D quedara de la siguiente forma:

SI X_1 es Grande Y X_2 es Grande Y X_3 es Pequeña Y X_4 es Mediana...

Teniendo los antecedentes de las Reglas A, B, C y D de la tabla 10 como las posibles Reglas que se van a activar, debemos compararlas con la base de conocimiento de la tabla 4, en el caso de que las Reglas A, B, C y D estén en la base de conocimiento se podrá decir que estas Reglas se activarán.

Tabla 10. Posibles antecedentes de Reglas.

	X_1	X_2	X_3	X_4	...
A	Mediana	Grande	Pequeña	Pequeña	...
B	Mediana	Grande	Pequeña	Mediana	...
C	Grande	Grande	Pequeña	Pequeña	...
D	Grande	Grande	Pequeña	Mediana	...

Comparando la tabla 10 con la tabla 4 se puede ver que el antecedente de la Regla D de la tabla 10 es igual al antecedente de la Regla 1 de la tabla 4, y los demás antecedentes de reglas de la tabla 10 no son igual a ningún antecedente de la base de conocimiento de la tabla 4. Por lo tanto se puede decir que la Regla 1 de la tabla 4 se activará.

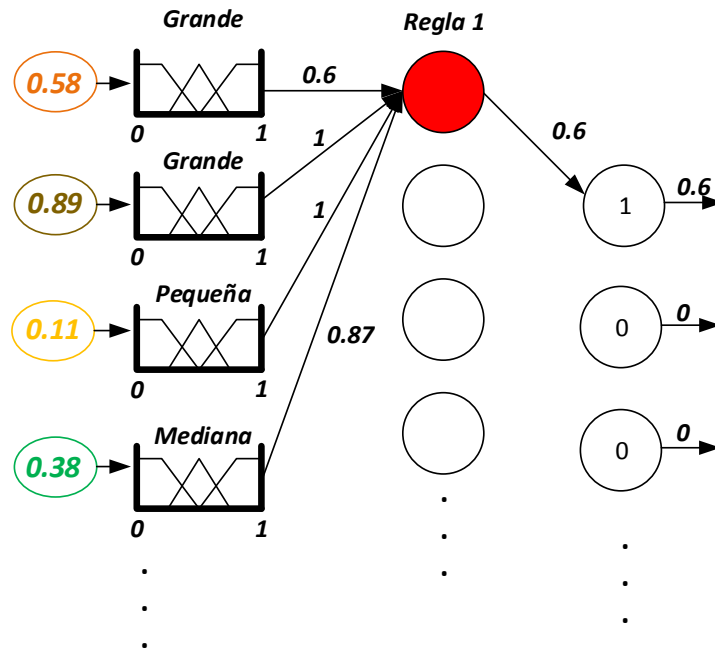
Para calcular la activación de la regla 1 se escoge el valor más pequeño de las pertenencias a los conjuntos que componen la regla 1, para mayor entendimiento se presenta la tabla 11, esta tabla se construyó a partir de los valores resaltados de las pertenencias de la Regla en la tabla 9, que fue la que se activó.

Tabla 11. Activaciones de los conjuntos que componen la regla.

	X_1	X_2	X_3	X_4	...
Conjunto	Grande	Grande	Pequeña	Mediana	...
Pertenencia	0.6	1	1	0.87	...

Como se puede ver en la tabla 11, el valor más pequeño es de 0.6 que se debe a la pertenencia al conjunto *Grande* de la variable X_1 , por ende la activación de la regla será 0.6. Como en este caso solo se activó una regla y considerando la aplicación del operador Fuzzy *max* la salida de la Red será 0.6 0 0 0 0 0 0 0 0 0 como se puede ver en la figura 100.

Figura 100. Salida de la Red Neuro-Fuzzy.



Como la salida esperada es 1 0 0 0 0 0 0 0 0 0 y debido a que son 10 neuronas de salida se calcula 10 errores con la formula $\delta_{Ci} = t_i - o_{Ci}$.

Para la neurona de salida 1 el error es: $\delta_{C1} = 1 - 0.6 = 0.4$

Para la neurona de salida 2 el error es: $\delta_{C2} = 0 - 0 = 0$.

Para la neurona de salida 3 el error es: $\delta_{C3} = 0 - 0 = 0$.

Para la neurona de salida 4 el error es: $\delta_{C4} = 0 - 0 = 0$.

Para la neurona de salida 5 el error es: $\delta_{C5} = 0 - 0 = 0$.

Y así para las demás.

A continuación se debe calcular el grado de cumplimiento para la regla activada con la formula.

$$\delta_R = O_R(1 - O_R)\delta_C$$

Donde O_R es la activación de la regla calculada anteriormente como 0.6 y δ_C es el error en la salida del consecuente de la regla, es decir en este caso la Regla 1 tiene como consecuente la clase 1 (*Interferencia de Gas*), entonces el error que se

debe utilizar es δ_{c1} , que fue calculado anteriormente con un valor de 0.4, entonces el grado de cumplimiento de la Regla 1 se calcula como:

$$\delta_R = 0.6(1 - 0.6)0.4 = 0.096$$

Ahora se pasa a escoger el conjunto Fuzzy que se va a modificar y a calcular los parámetros de modificación, en este caso es el conjunto *Grande* de la variable 1, ya que es el que tiene menor pertenencia como se puede ver en la tabla 11, por otro lado, el cálculo de los parámetros de modificaciones se hace suponiendo que el usuario ha definido un coeficiente de aprendizaje de 0.05. Para calcular los parámetros de modificación se utilizan las siguientes formulas:

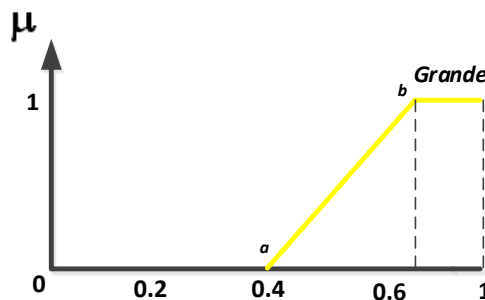
$$\delta_b = \sigma \cdot \delta_R(c - a) \cdot \text{sgn}(O_{x'} - b)$$

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

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

Donde los valores δ_a , δ_b , δ_c son las modificaciones que se la va a realizar al conjunto, σ representa el coeficiente de aprendizaje definido por el usuario, a, b, c son los parámetros del conjunto Fuzzy y están definidos en la figura 101, para el conjunto de interés (*Grande*).

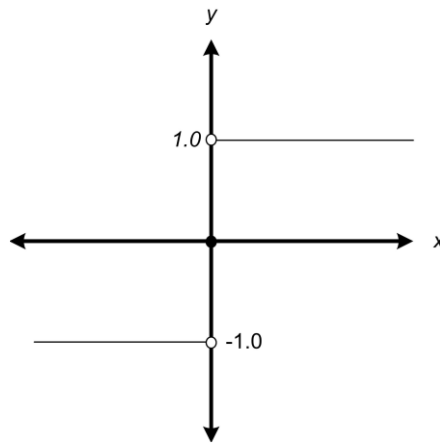
Figura 101. Parámetros de un conjunto Fuzzy.



$O_{x'}$ representa el valor del patrón de entrada a la variable que contiene el conjunto a modificar que en este caso es 0.58, la función $sgn(X)$ es la función signo y está definida como:

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$$

Figura 102. Función signo.



Y por último el valor δ_R es el grado de cumplimiento de la regla calculada anteriormente como 0.096, así los valores son calculados como:

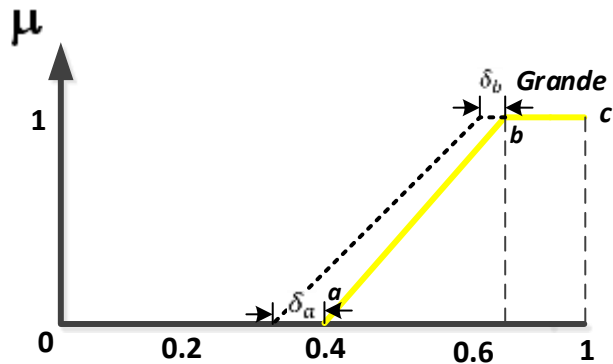
$$\delta_b = \sigma \cdot \delta_R(c - a) \cdot sgn(O_{x'} - b) = 0.05 * 0.096(1 - 0.4) * sgn(0.58 - 0.7) = -0.00288$$

$$\delta_a = -\sigma \cdot \delta_R(c - a) + \delta_b = -0.05 * 0.096(1 - 0.4) - 0.00288 = -0.00576$$

$$\delta_c = \sigma \cdot \delta_R(c - a) + \delta_b = 0.05 * 0.096(1 - 0.4) - 0.00288 = 0$$

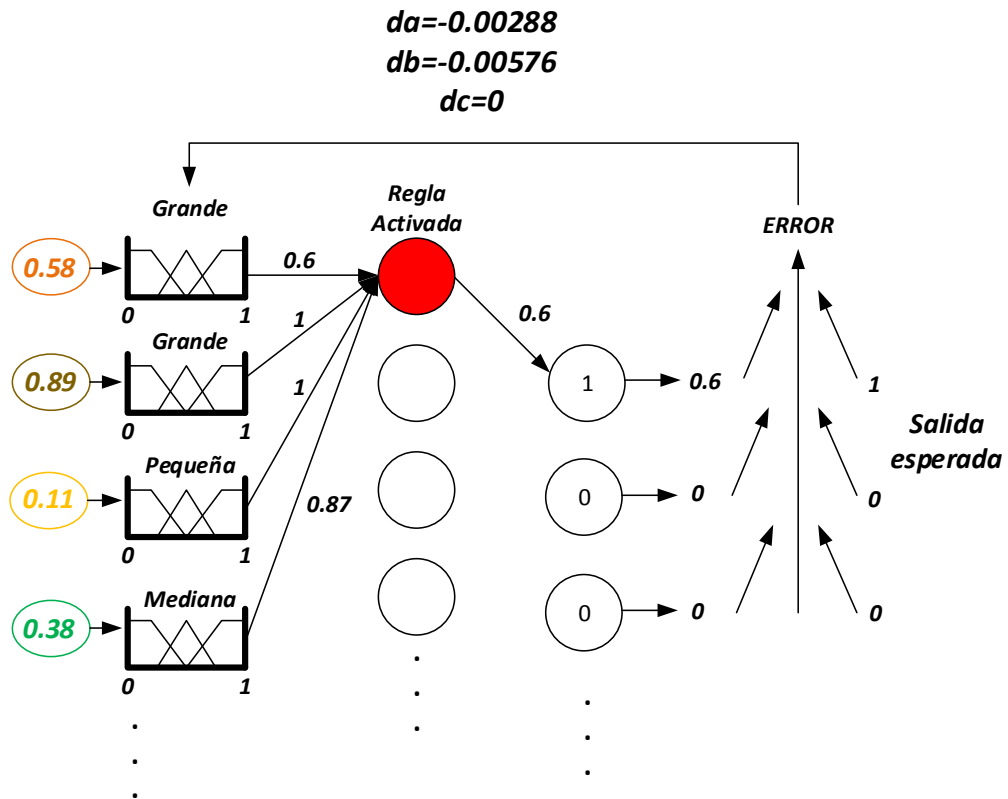
Con estos valores se pueden modificar el conjunto y queda finalmente como el de la figura 103.

Figura 103. Conjunto modificado por entrenamiento.



Un resumen de propagación de un patrón en entrenamiento se encuentra en la figura 104.

Figura 104. Ejemplo entrenamiento.



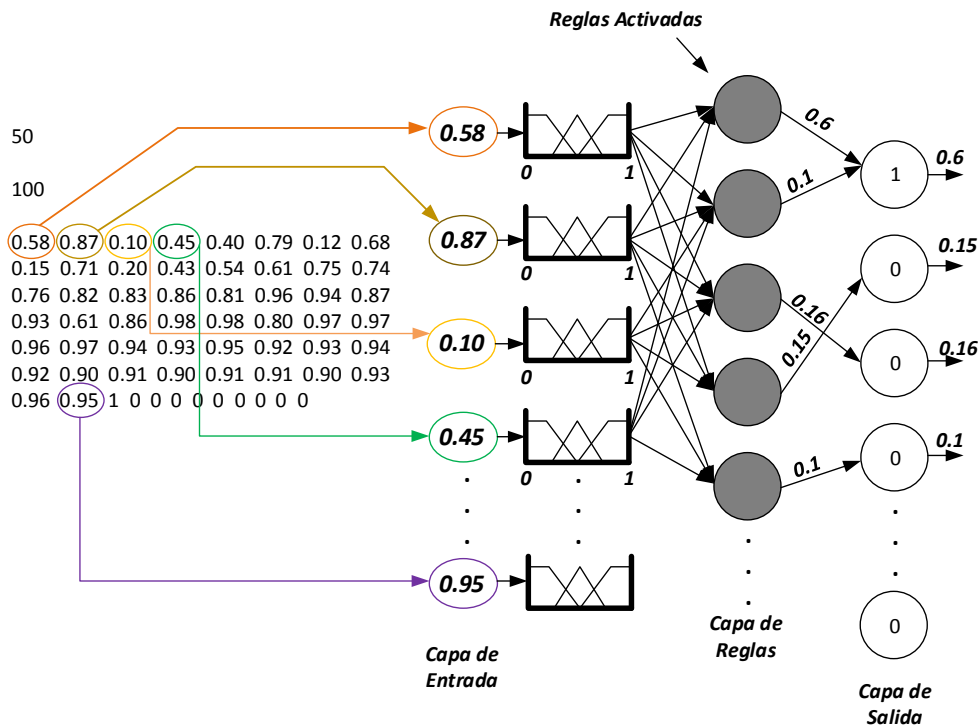
Después de que se propaguen todos los patrones el número de veces que diga el número de épocas, se finaliza el proceso de entrenamiento como muestra la figura 99. Después de que se ha entrenado la Red, se procede a realizar la validación

del entrenamiento, donde se van a obtener resultados para que el usuario decida si se necesita entrenar más la Red o si pasa al proceso de clasificación.

6.3.3.4 Validación del entrenamiento (IV).

Cuando se ha terminado de entrenar la Red, sigue el proceso de validación. En este proceso se utiliza el archivo de validación, donde se propagaran los 60 patrones de datos contenidos en este archivo de la misma forma como fue explicado en el proceso de entrenamiento con el fin de que la Red de una respuesta, esta es comparada con la respuesta esperada contenida en el archivo de validación para saber si la Red ha clasificado el patrón correctamente, después de haber propagado todos los patrones del archivo de validación, se calcula el porcentaje de los patrones clasificados correctamente. En la figura 105 se ilustra la validación para un patrón.

Figura 105. Validación para un patrón de datos.



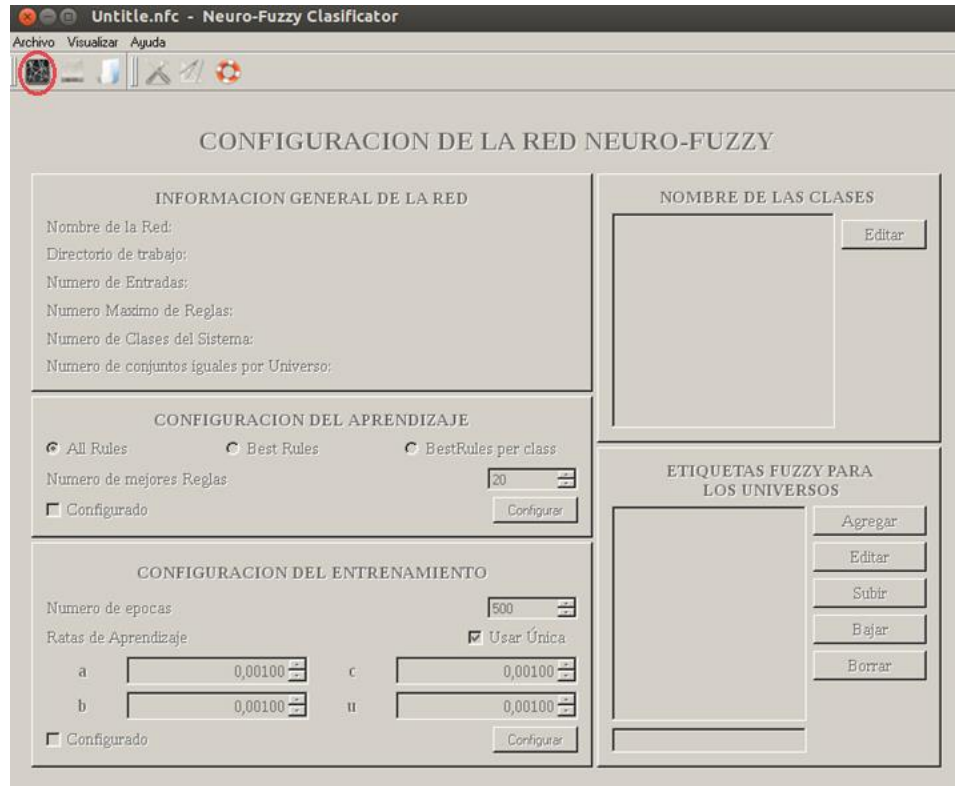
6.3.3.5 Clasificación (V)

En el proceso de clasificación es similar al de validación, donde los patrones son propagados y la Red da una respuesta para cada uno, esta respuesta es la clasificación del patrón.

6.3.4 Funcionamiento NEFCLASS-Q

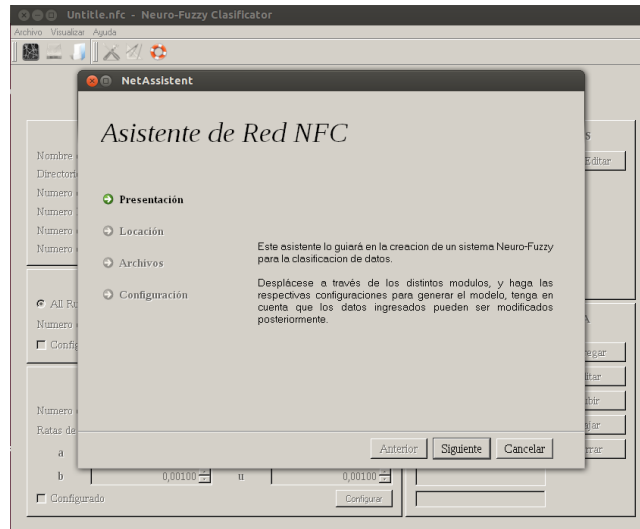
A continuación se explicara paso a paso, desde el ingreso de los archivos planos (archivos de Entrenamiento, Validación y Operación) hasta el proceso de clasificación. Luego de tener configurados los archivos de entrenamiento, validación y clasificación, procedemos en ejecutar el programa (para ejecutar el NEFCLASS-Q, se debe iniciar primero el Qt Creator, compilador empleado como plataforma de soporte) y aparecerá la ventana de la figura 106.

Figura 106. Interfaz software Nefclass-Q.



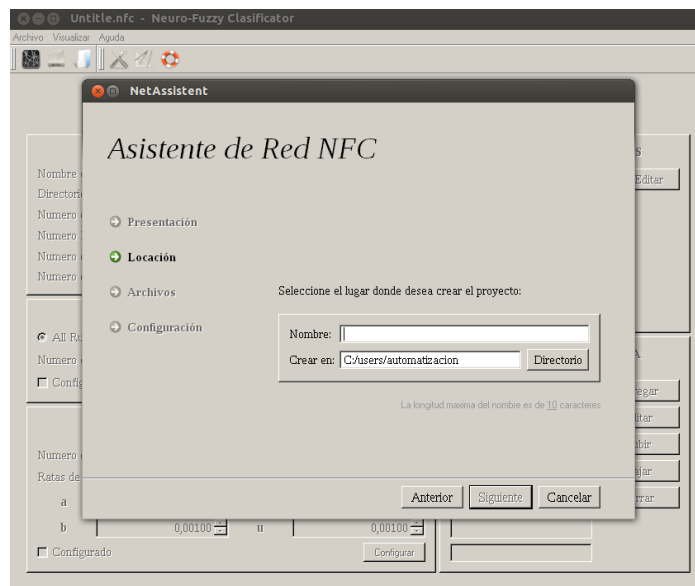
A continuación se debe crear un proyecto nuevo. Se procede a dar clic en el icono de la barra de tareas como se muestra en la figura 106. Posterior a dar clic, se genera la siguiente ventana, ver figura 107.

Figura 107. Asistente NEFCLASS-Q.



Se ejecuta el asistente para la creación de un nuevo proyecto. Se da clic en siguiente, y este nos solicita el nombre del proyecto y el lugar donde desea que se guarde, ver figura 108.

Figura 108. Nombre y Dirección de guardado.



Se da clic en siguiente nuevamente, para generar la ventana de la figura 109, donde se deben buscar los archivos de entrenamiento, validación y clasificación.

Figura 109. Selección de archivos.

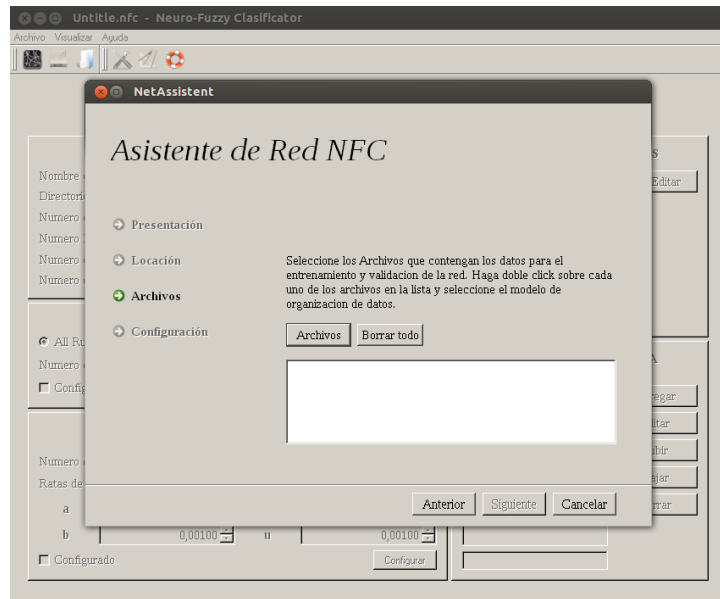
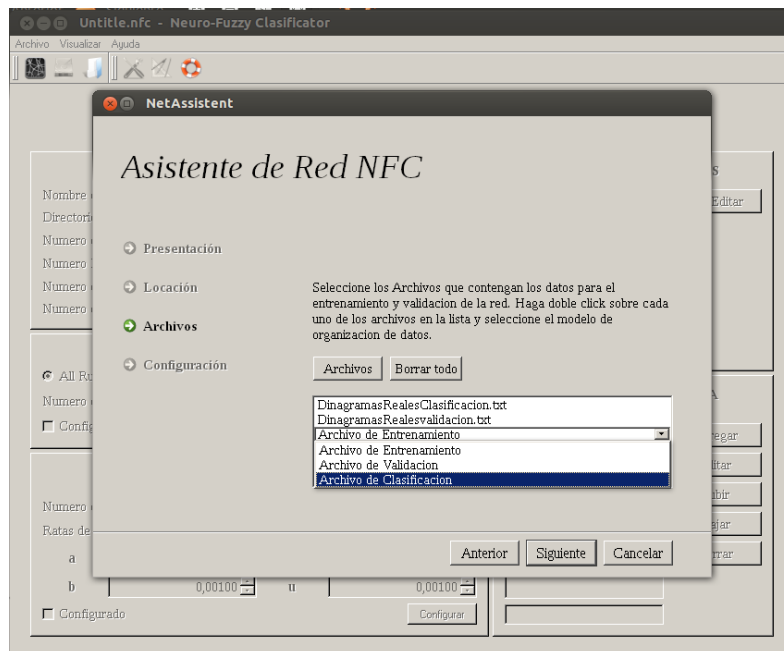
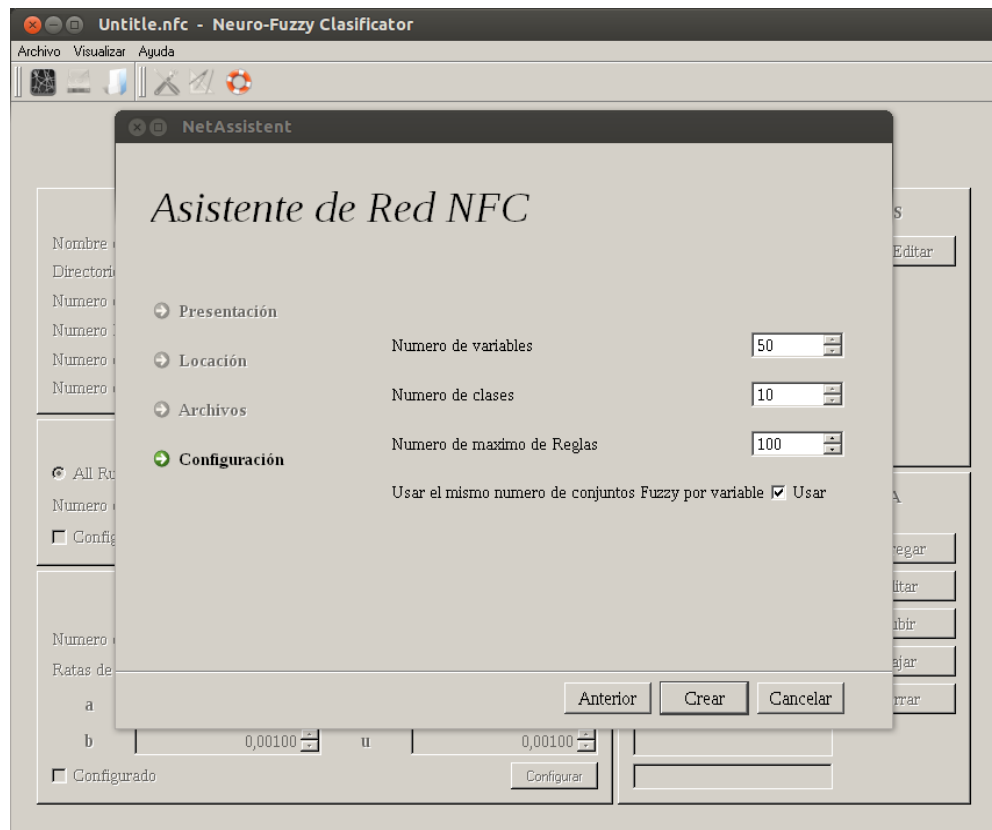


Figura 110. Asignación de cada tipo de archivo.



En la figura 110 se le indica al software que clase de archivo es cada uno, ya que todos son archivos planos y él por sí solo no identifica cual es entrenamiento, validación o clasificación.

Figura 111. Creación de proyecto.



En la figura 111 se muestra un resumen donde se indican número de variables, número de clases y número máximo de reglas, además de la opción de crear el mismo número de conjuntos Fuzzy por cada variable. Cuando se da clic en crear se muestra un mensaje que indica que el proyecto ha sido creado exitosamente. Se da clic en el botón OK, siguiendo a la ventana de configuración de parámetros de entrenamiento y aprendizaje de la Red como muestra la figura 112.

Figura 112. Inicio de Entrenamiento.

32.nfc - Neuro-Fuzzy Clasificador

Archivo Visualizar Ayuda

CONFIGURACION DE LA RED NEURO-FUZZY

INFORMACION GENERAL DE LA RED

Nombre de la Red: 32
Directorio de trabajo: C:/users/automatizacion
Numero de Entradas: 50
Numero Maximo de Reglas: 100
Numero de Clases del Sistema: 10
Numero de conjuntos iguales por Universo: SI

CONFIGURACION DEL APRENDIZAJE

All Rules Best Rules BestRules per class
Numero de mejores Reglas: 20
 Configurado

CONFIGURACION DEL ENTRENAMIENTO

Numero de epocas: 500
Ratas de Aprendizaje: 0,00100 Usar Única
a: 0,00100 c: 0,00100
b: 0,00100 u: 0,00100
 Configurado

NOMBRE DE LAS CLASES

Clase 1
Clase 2
Clase 3
Clase 4
Clase 5
Clase 6
Clase 7
Clase 8
Clase 9
Clase 10

ETIQUETAS FUZZY PARA LOS UNIVERSOS

small
medium
large

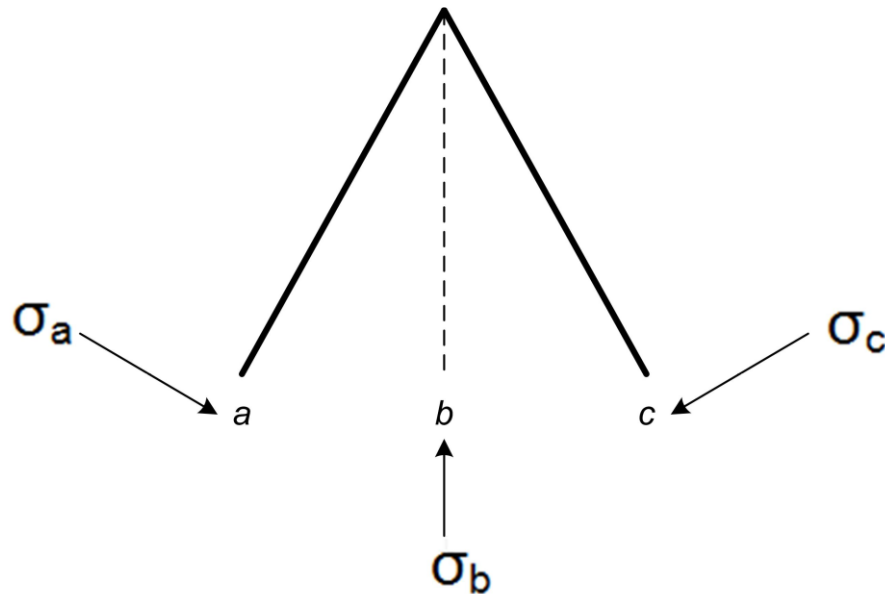
En la figura 112, se observan diferentes paneles donde se indica información general, como datos de la Red, configuración de aprendizaje, configuración de entrenamiento, nombre de las clases y etiquetas Fuzzy para los universos.

En el panel de información general de la Red, presenta toda la información acerca de la Red, para el caso de la figura 112, se entrena una Red con 50 entradas, un número máximo de reglas de 100, numero de clases que son 10, y aparece si en la creación del proyecto se seleccionó la opción de mismo número de conjuntos Fuzzy por universo, que para este caso fue SI.

En el panel de configuración de aprendizaje podemos seleccionar un método de pruning de mejores reglas o mejores reglas por clase, luego se da clic en configurar para aplicarlos.

En el panel configuración del entrenamiento, se define el número de épocas, que son el número de ciclos que realizara el algoritmo, además de las ratas de aprendizaje de los pesos, aquí se puede aplicar el mismo valor de la rata para los 3 puntos (ver figura 113) del conjunto o definirlos de manera independiente. Igualmente se debe dar clic en el botón configurar para aplicarlos a la Red.

Figura 113. Puntos donde actúan los coeficientes de aprendizaje.



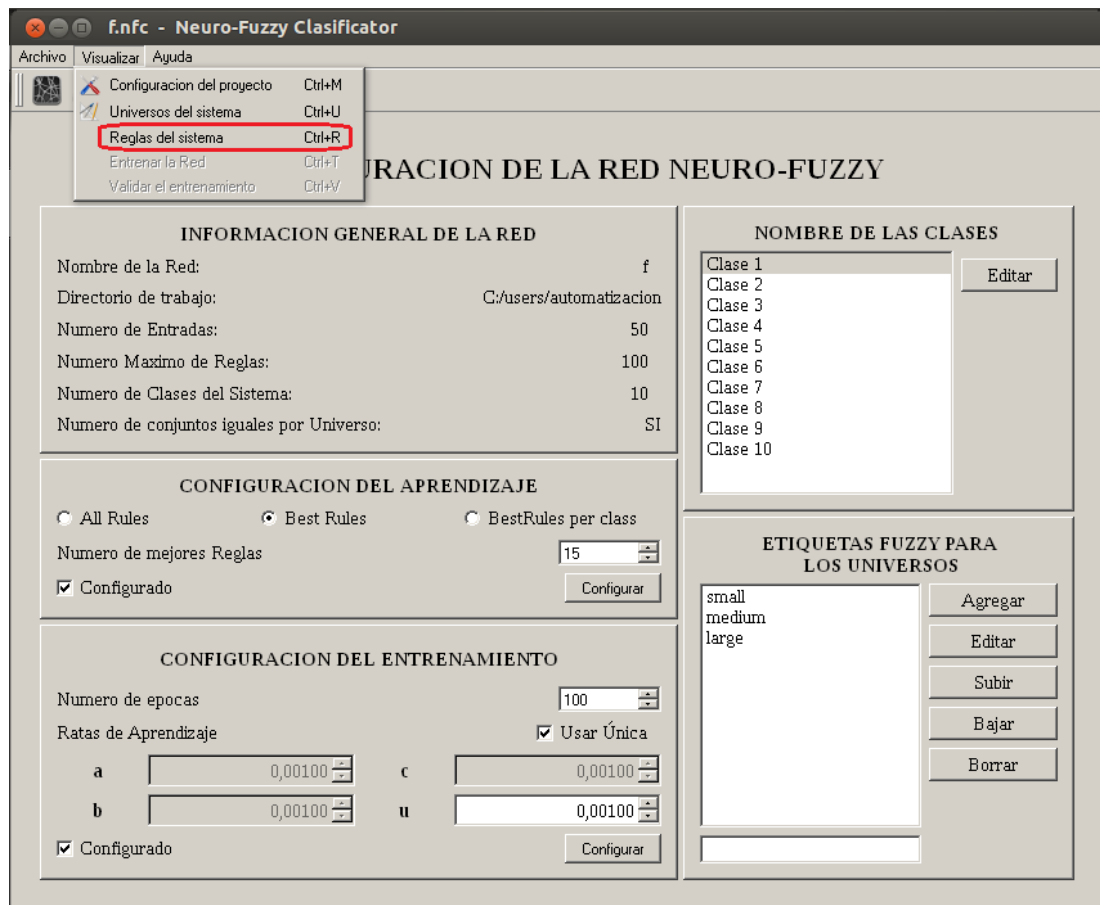
6.3.4.1 Inicio de aprendizaje.

Se procede a configurar los parámetros de aprendizaje, donde seleccionamos si queremos un aprendizaje donde se tomen todas las reglas creadas, las mejores o las mejores activadas por clase, para aplicar los cambios se clic en el botón “configurar”

Aquí se debe tener en cuenta que, si se desea mejores reglas o mejores reglas por clase, este número de reglas debe ser menor al número máximo de reglas anteriormente estipuladas, es decir, si se ingresa un número máximo de reglas de 30, el número de mejores reglas debe ser menor a 30, para poder hacer el corte respectivo. Posterior a esto se configura el entrenamiento asignando un número

de épocas y una tasa de aprendizaje, también dando clic en configurar para aplicar los cambios.

Figura 114. Visualizar, Reglas del Sistema.



Luego se debe dirigir a la pestaña ubicada en la parte superior izquierda llamada "Visualizar" y damos clic en Reglas del Sistema (ver figura 114), se presentara la ventana de la Figura 115. En esta parte se puede agregar conocimiento previo y crear reglas. Para agregar el conocimiento previo se debe seleccionar la casilla "Agregar Reglas a Priori" que está en la parte inferior izquierda como muestra la figura 116 y se colocan las reglas que se deseen agregar. Si no se tiene conocimiento previo, se generan las reglas dando clic en el botón llamado "Crear Reglas" ubicado en la parte inferior derecha (ver figura 115), y posteriormente al botón "Pruning", las reglas definitivas son mostradas como en la figura 117.

Figura 115. Creación de Reglas.

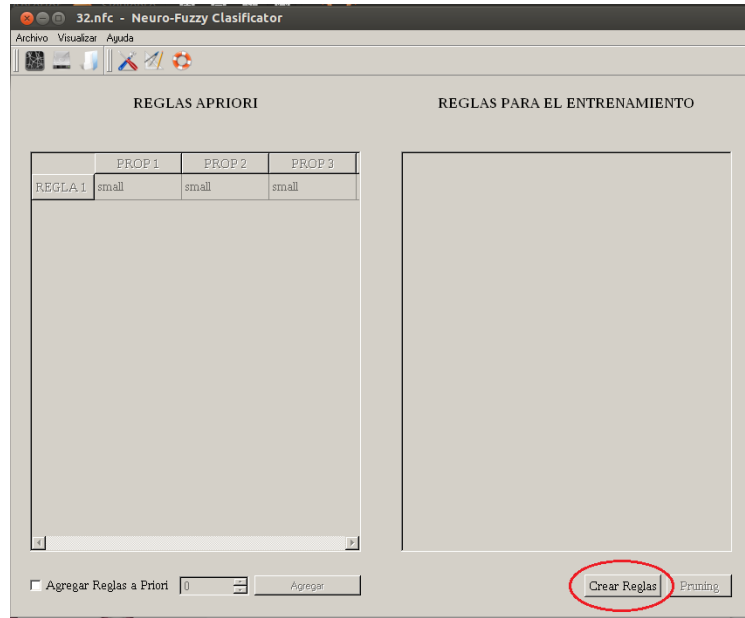
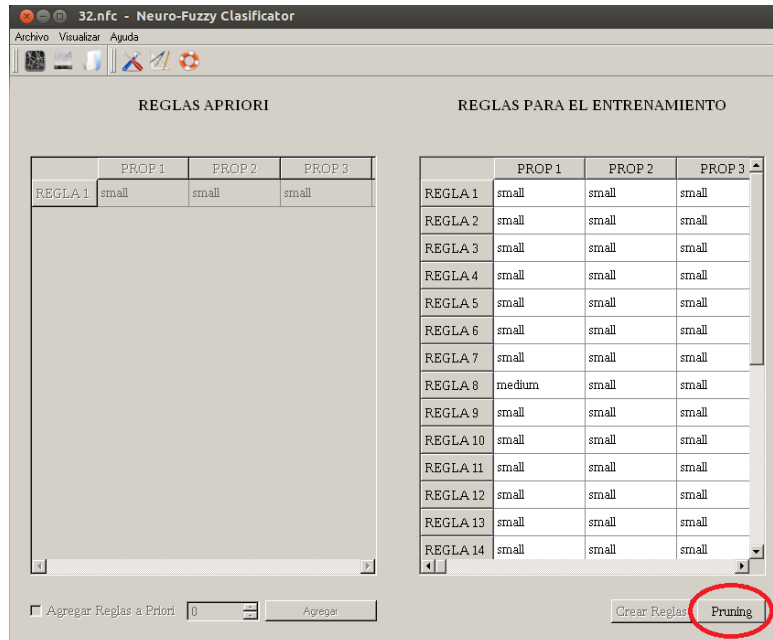


Figura 116. Reglas a Priori.

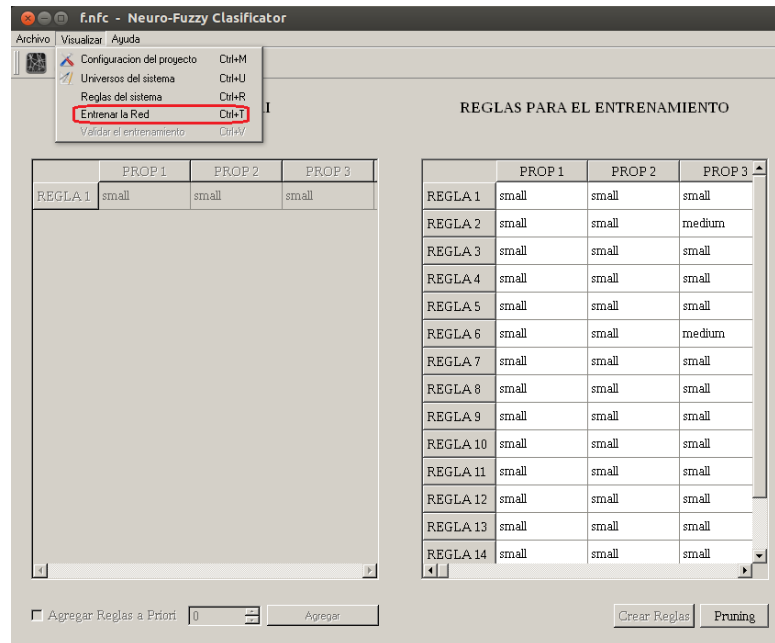


Figura 117. Pruning de Reglas.



Luego de hacer el pruning de reglas, de nuevo se regresa a la pestaña Visualizar donde damos clic en Entrenar la Red, ver figura 118.

Figura 118. Visualizar, Entrar la Red.



Donde se mostrara (ver figura 119):

- Los conjuntos Fuzzy por cada variable.
- Los universos de discurso por cada variable.
- Un resumen de: Numero de reglas, Máximo número de reglas, Número de conjuntos Fuzzy.
- Opción donde se puede definir según criterios detener el entrenamiento cuando las condiciones de porcentaje de clasificación o patrones descalificados se cumpla, para ello debemos activarlas chequeando la caja “Detener entrenamiento con”.
- Barra donde se muestra el porcentaje de entrenamiento completado en tiempo real (ver figura 120). El proceso de entrenamiento de la Red puede tardar entre 5 y 120 minutos, dependiendo del número de épocas ingresadas por el usuario. En la figura 121 se muestra la ventana donde el software indica que el proceso de entrenamiento ha sido terminado.

Figura 119. Fase de entrenamiento.

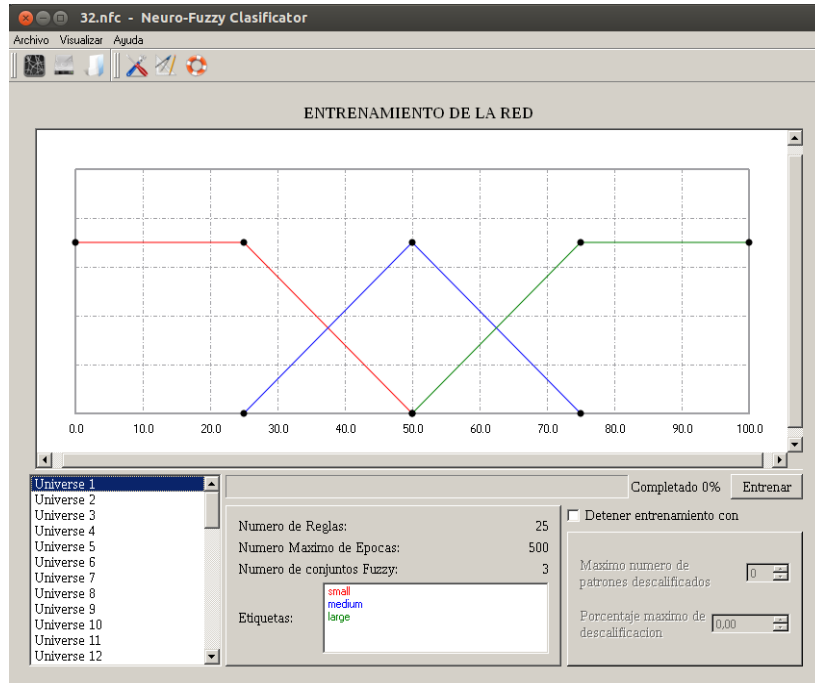


Figura 120. Entrenamiento en curso.

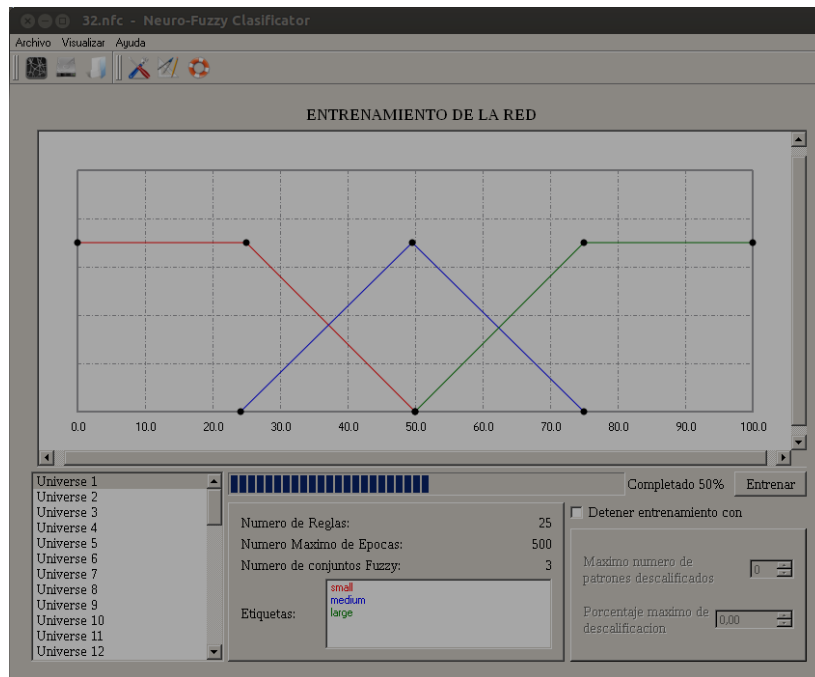
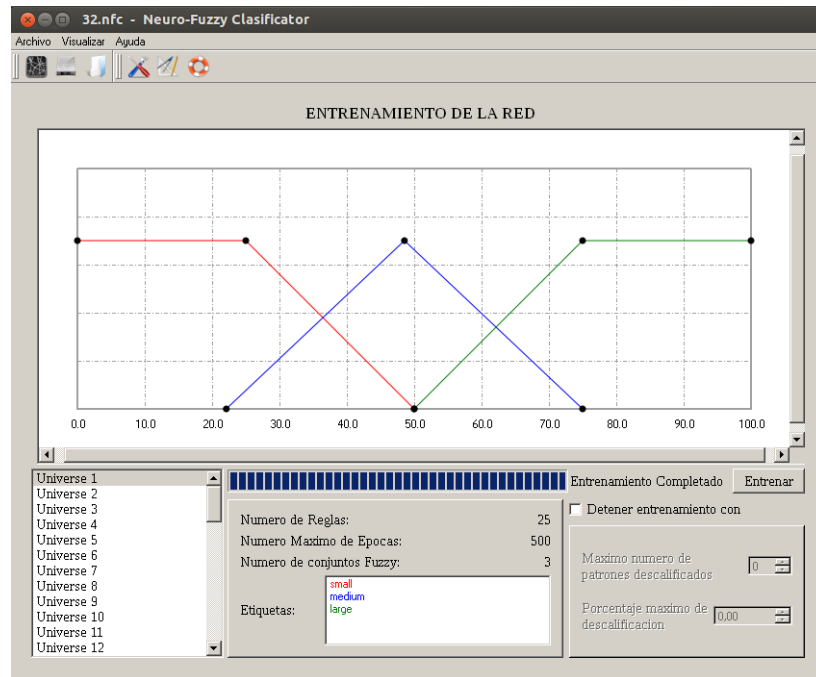
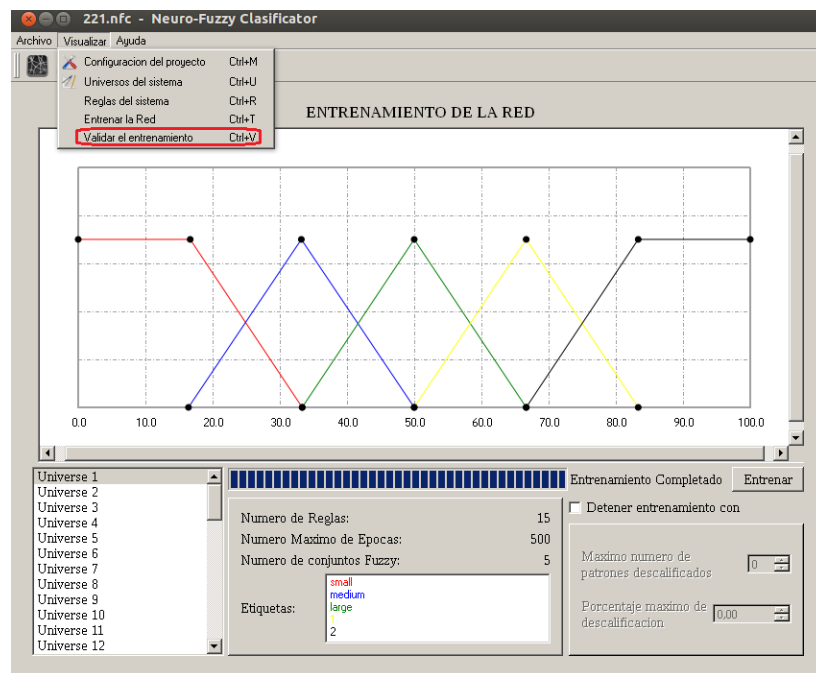


Figura 121. Entrenamiento Terminado.



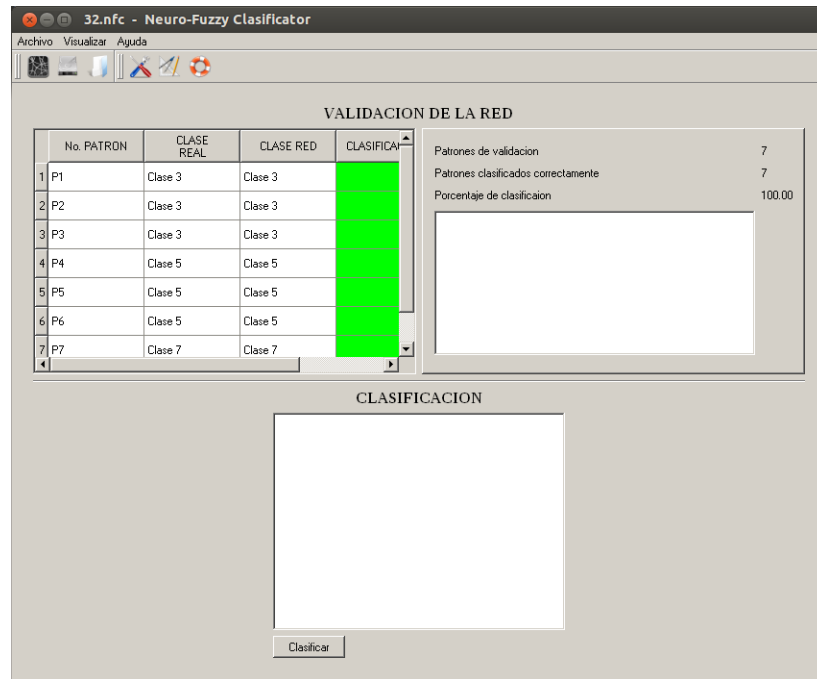
Luego de terminado el entrenamiento se debe ir a la pestaña Visualizar nuevamente y se da clic en Validar el Entrenamiento (ver figura 122).

Figura 122. Visualizar, Validar el entrenamiento.



Se presentara la ventana de la figura 123 donde se observan tres paneles distintos.

Figura 123. Validación de Red.



El primer panel muestra una tabla donde se explica la clase real de cada patrón, la clase que le asigno la Red, y por último el color, que se entiende por verde los que se clasificaron correctamente y los clasificados de forma incorrecta se identifican con color rojo.

El segundo panel indica, numero de patrones de validación, patrones clasificados correctamente, y porcentaje de clasificación.

Y el tercer panel que se observa también en la figura 124, usa el último de los archivos planos, el archivo de Operación. Después de validada la Red, se hacen pasar estos dinagramas donde se puede analizar cuantos clasifica. Cabe resaltar que no se debe esperar una clasificación perfecta por el hecho de tener un porcentaje de validación del 100%. Este porcentaje solo nos indica que tan bien estuvo el entrenamiento.

Figura 124. Clasificación de dinagramas.

32.nfc - Neuro-Fuzzy Clasificador

Archivo Visualizar Ayuda

VALIDACION DE LA RED

No. PATRON	CLASE REAL	CLASE RED	CLASIFICACION
1 P1	Clase 3	Clase 3	
2 P2	Clase 3	Clase 3	
3 P3	Clase 3	Clase 3	
4 P4	Clase 5	Clase 5	
5 P5	Clase 5	Clase 5	
6 P6	Clase 5	Clase 5	
7 P7	Clase 7	Clase 7	

Patrones de validacion: 7
 Patrones clasificados correctamente: 7
 Porcentaje de clasificacion: 100.00

CLASIFICACION

PATRON No	CLASE
7 P7	Clase 7
8 P8	Clase 10
9 P9	Clase 6
10 P10	Clase 1
11 P11	Clase 1
12 P12	Clase 1
13 P13	Clase 6

Clasificar

7. PRUEBAS Y RESULTADOS

Como se explicó en el capítulo 4, para el archivo de entrenamiento se crearon 10 dinagramas por cada falla (ver tabla 12), es decir, en total se crearon 100 dinagramas para entrenar la Red. Para el archivo de validación se crearon 6 dinagramas por cada falla, para un total de 60 dinagramas y para el archivo de operación se crearon 6 dinagramas por cada falla, para un total de 60 dinagramas. Cada patrón está codificado en 10 clases como se muestra en la tabla 12.

Tabla 12. Codificación de los dinagramas.

Clase 1 0 0 0 0 0 0 0 0 0	Interferencia de gas.
Clase 0 1 0 0 0 0 0 0 0 0	Tubería desanclada con golpe de fluido.
Clase 0 0 1 0 0 0 0 0 0 0	Rotura de varilla.
Clase 0 0 0 1 0 0 0 0 0 0	Fuga en la válvula fija.
Clase 0 0 0 0 1 0 0 0 0 0	Fuga en la válvula viajera.
Clase 0 0 0 0 0 1 0 0 0 0	Barril de la bomba doblado.
Clase 0 0 0 0 0 0 1 0 0 0	Buen llenado con tubería anclada.
Clase 0 0 0 0 0 0 0 1 0 0	Agujero en el barril de la bomba.
Clase 0 0 0 0 0 0 0 0 1 0	Ancla de tubería en mal funcionamiento.
Clase 0 0 0 0 0 0 0 0 0 1	Barril de la bomba gastado.

7.1 CREACIÓN DE LA RED.

Para crear una Red es necesario definir algunos parámetros, los cuales afectan de manera directa los resultados obtenidos en el entrenamiento (calificado por la validación) y operación. Estos parámetros son:

- Número de conjuntos Fuzzy en cada universo de discurso.
- Número máximo de reglas.
- Tipo de pruning que se le desea aplicar a las reglas.

- Numero de épocas de entrenamiento.
- Ratias de aprendizaje.

Debido a que no existe una metodología establecida de cual o cuales de los parámetros se deben variar para que genere los mejores resultados al entrenar de la Red, se decide variar algunos de ellos y posterior a esto analizar los resultados.

7.2 VALIDACIÓN CON PRUNING.

7.2.1 Pruning Simple.

En este pruning se dejan todas las reglas creadas en el proceso de aprendizaje. Estos resultados se obtuvieron usando los archivos organizados de forma ascendente (Archivo tipo 1 mostrado en la figura 88).

Tabla 13. Tabla de resultados con pruning simple archivo tipo 1.

Coef. Aprendiziza	Núm. Conjuntos	Épocas	Validación	Reglas creadas
0.001	3	500	96 %	23
0.01	5	1000	98 %	32
0.001	7	500	93 %	44
0.1	3	3000	96 %	23
0.001	5	2000	95 %	32
0.01	7	1000	96 %	44
0.01	3	4000	96 %	23
0.1	5	500	95 %	32
0.1	7	500	95 %	44

Resultados obtenidos con los archivos organizados de forma grupal (Archivo tipo 2 mostrado en la figura 89).

Tabla 14. Tabla de resultados con pruning simple archivo tipo 2.

Coef. Aprendiziza	Núm. Conjuntos	Épocas	Validación	Reglas creadas
0.001	3	500	94 %	23
0.01	5	1000	95 %	32

0.001	7	500	93 %	44
0.1	3	3000	96 %	23
0.001	5	2000	93 %	32
0.01	7	1000	95 %	44
0.01	3	4000	96 %	23
0.1	5	500	93 %	32
0.1	7	500	95 %	44

7.2.2 Pruning de las mejores reglas.

En este pruning se dejan las reglas que tengas mayor activación. Estos resultados se obtuvieron usando los archivos organizados de forma ascendente (Archivo tipo 1 mostrado en la figura 88).

Tabla 15. Tabla de resultados con las mejores reglas archivo tipo 1.

Coef. Aprendizaje	Núm. Conjuntos	Épocas	Validación	Reglas creadas
0.001	7	100	51 %	10
0.01	7	500	90 %	20
0.1	7	1000	96 %	25
0.001	7	2000	95 %	30
0.01	7	3000	96 %	35
0.1	7	4000	96 %	40
0.001	5	100	68 %	15
0.01	5	500	83 %	20
0.1	5	1000	81 %	25
0.01	5	2000	85 %	25
0.1	5	4000	96 %	30
0.001	3	100	76 %	10
0.01	3	500	96 %	20
0.01	3	500	76 %	10
0.1	3	1000	86 %	15

Resultados obtenidos con los archivos organizados de forma grupal (Archivo tipo 2 mostrado en la figura 89).

Tabla 16. Tabla de resultados con las mejores reglas archivo tipo 2.

Coef. Aprendizaje	Núm. Conjuntos	Épocas	Validación	Reglas creadas
0.001	7	100	36 %	10

0.01	7	500	48 %	20
0.1	7	1000	68 %	25
0.001	7	2000	78 %	30
0.01	7	3000	90 %	35
0.1	7	4000	93 %	40
0.001	5	100	68 %	15
0.01	5	500	79 %	20
0.1	5	1000	83 %	25
0.01	5	2000	83 %	25
0.1	5	4000	95 %	30
0.001	3	100	48 %	10
0.01	3	500	88 %	20
0.01	3	500	48 %	10
0.1	3	1000	66 %	15

7.2.3 Pruning de las mejores reglas por clase.

Este pruning tiene como objetivo dejar las reglas que tenga la mejor activación acumulada por clase. Estos resultados se obtuvieron usando los archivos organizados de forma ascendente (Archivo tipo 1 mostrado en la figura 88).

Tabla 17. Tabla de resultados con las mejores reglas por clase archivo tipo 1.

Núm. Mejo. Reglas por clase	Coef. Aprendiziza	Núm. Conjuntos	Épocas	Validación	Reglas creadas
10	0.001	3	500	96 %	23
20	0.01	3	1000	96 %	23
15	0.1	3	3000	96 %	23
5	0.001	5	2000	95 %	32
8	0.01	5	1000	95 %	32
13	0.1	5	2000	95 %	32
10	0.001	7	500	93 %	44
8	0.01	7	1000	96 %	44
13	0.1	7	2000	96 %	44

Resultados obtenidos con los archivos organizados de forma grupal (Archivo tipo 2 mostrado en la figura 89).

Tabla 18. Tabla de resultados con las mejores reglas por clase archivo tipo 2.

Núm. Mejo. Reglas por clase	Coef. Aprendiziza	Núm. Conjuntos	Épocas	Validación	Reglas creadas
10	0.001	3	500	92 %	23
20	0.01	3	1000	92 %	23
15	0.1	3	3000	92 %	23
5	0.001	5	2000	91 %	32
8	0.01	5	1000	91 %	32
13	0.1	5	2000	91 %	32
10	0.001	7	500	93 %	44
8	0.01	7	1000	96 %	44
13	0.1	7	2000	96 %	44

De todos los entrenamientos realizados, la prueba que presento mejores resultados en validación es aquella en donde se aplicó *pruning* simple con 5 conjuntos Fuzzy, un coeficiente de aprendizaje de 0.01, realizada con el archivo de entrenamiento organizado de forma ascendente (archivo tipo 1), arrojando un porcentaje de validación del 98% clasificando mal 2 dinagramas. Gracias a que la validación arrojó buenos resultados, se concluye que el entrenamiento fue un éxito, dando paso libre al proceso de clasificación.

7.3 CLASIFICACIÓN DE DINAGRAMAS REALES

Se determinó hacer una prueba extra con los dinagramas proporcionados por los expertos, enfocados al proceso de operación. La idea principal, es comprobar la veracidad del mejor entrenamiento obtenido en las pruebas anteriores. Se empleara el conocimiento adquirido para el entrenamiento que arrojó el porcentaje de validación más alto, es decir, del 98%.

Se hicieron dos pruebas la primera empleando los dinagramas que presentan una sola falla, y la segunda empleando los dinagramas de falla combinada.

En total se emplearon nueve (9) dinagramas donde se presenta una sola falla, los cuales son:

- Buen Funcionamiento. (ver figura 43).
- Válvula Viajera. (ver figura 45).
- Válvula Viajera. (ver figura 46).
- Válvula Viajera. (ver figura 47).
- Varilla Rota. (ver figura 48).
- Varilla Rota. (ver figura 49).
- Varilla Rota. (ver figura 50).
- Interferencia de Gas. (ver figura 52).
- Buen Funcionamiento. (ver figura 53).

Realizado en este proceso se obtuvo el siguiente resultado:

De los nueve (9) dinagramas empleados, solo se clasificaron 6, es decir un porcentaje de clasificación de 66.66% en operación o clasificación.

Los tres (3) dinagramas que no se clasificaron se muestran a continuación.

Figura 125. Dinagrama no clasificado 1.

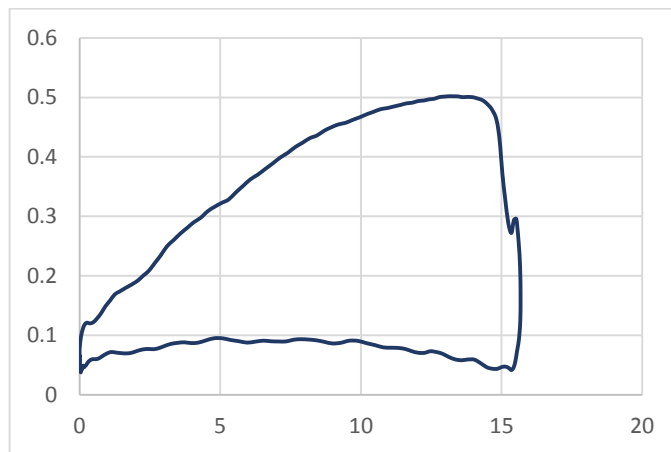


Figura 126. Dinagrama no clasificado 2.

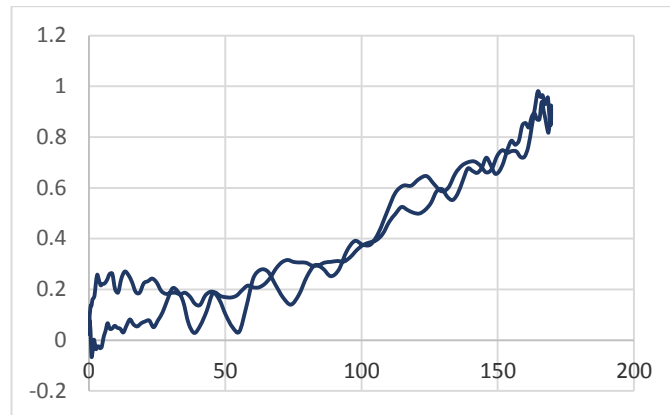
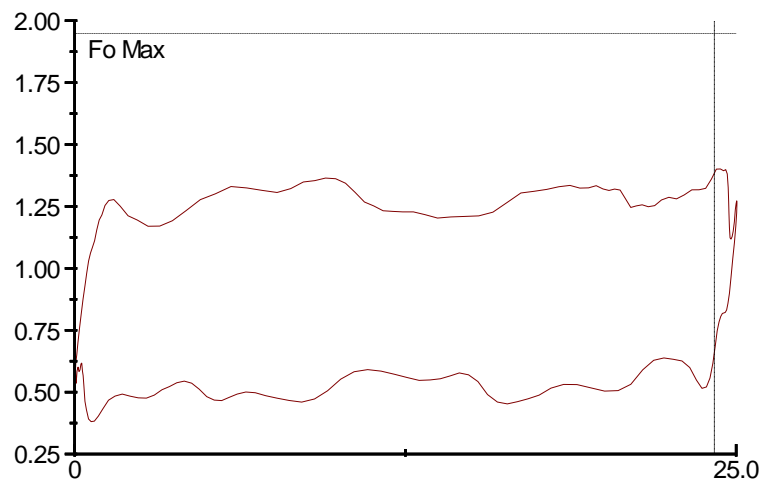
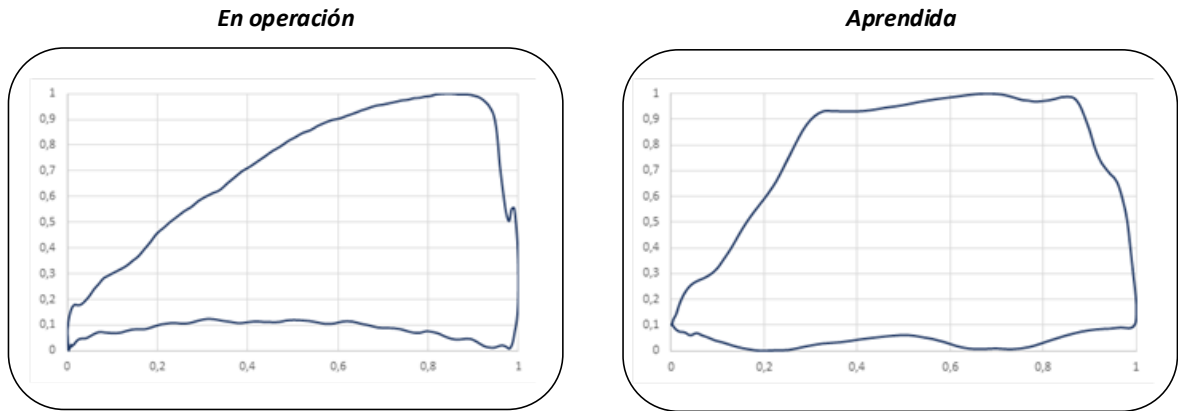


Figura 127. Dinagrama no clasificado 3.



La posible razón por la cual no clasificó, la falla de la figura 125, se debe a la diferencia en formas del dinagrama propagado al aprendido. Por ejemplo la carrera ascendente del dinagrama propagado en operación presenta mayor inclinación en comparación con la aprendida, además finalizando la carrera descendente el dinagrama de operación posee valores diferentes en carga, ver figura 128.

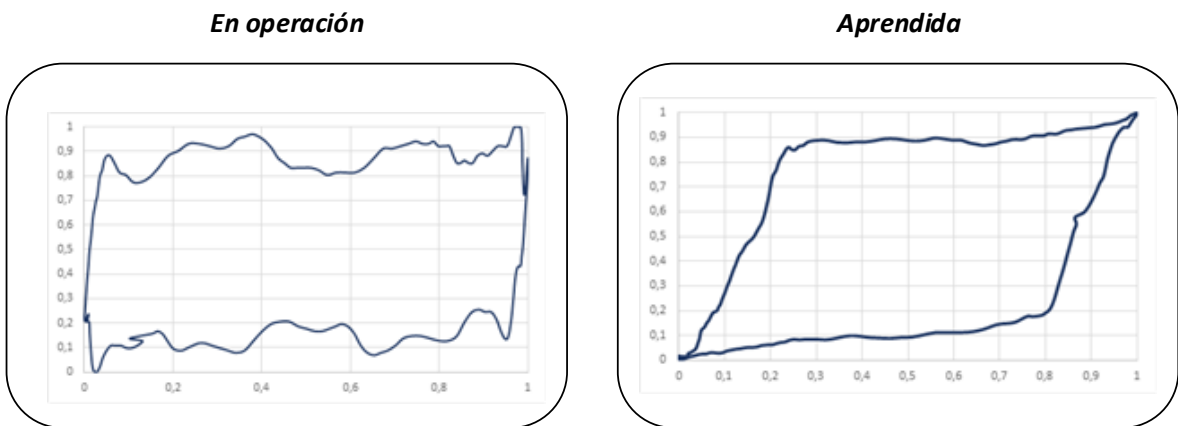
Figura 128. Comparación de dinagrama aprendido al propagado en operación.



Observando la figura 126, se presume que la razón por la cual no clasificó, se debe a que este dinagrama (rotura de varilla) presenta múltiples intersecciones entre su carrera ascendente y descendente, causando cambios en sus valores de amplitud, generando un dinagrama distinto a los aprendidos. La solución a este problema es ingresarle más ejemplos de este tipo de falla a la Red.

Por último el dinagrama de la figura 127 (Buen llenado de Bomba), no clasifica debido a la diferencia en formas, del aprendido al que se está propagando, ver figura 129.

Figura 129. Comparación de dinagrama aprendido al propagado en operación 2.



Para el caso de los dinagramas con falla combinada no se clasificó ninguno, debido a que la Red no posee ese conocimiento. La intención de esta prueba (fallas combinadas), era experimentar si al menos se identificaba una de fallas presentes en estos dinagramas, por los resultados obtenidos se concluye que no. Cabe resaltar que se estudió la forma de cada dinagrama aportado, comparando cada uno con las formas base de falla, además de contar con la evaluación del experto que las proporcionó.

8. MODELO FUZZY.

Un modelo Fuzzy sintetiza el conocimiento en:

1. Reglas del tipo SI – ENTONCES.
2. Variables y conjuntos Fuzzy.

8.1 VARIABLES Y CONJUNTOS FUZZY.

Las reglas Fuzzy generadas en el entrenamiento de mejor validación (98%) de la Red Neuro-Fuzzy se describen en la figura 130. Los antecedentes de la regla Fuzzy se escriben en forma simplificada:

- VS = Very Short (Muy Pequeño). Para el primer conjunto Fuzzy.
- S = Short (Pequeño). Para el segundo conjunto Fuzzy.
- M = Medium (Medio). Para el tercer conjunto Fuzzy.
- L = Large (Grande). Para el cuarto conjunto Fuzzy.
- VL = Very Large (Muy Grande). Para el quinto conjunto Fuzzy.

Las reglas Fuzzy que describen por ejemplo, SI (vs, vs, vs,..., vl), quiere decir que:

SI X_1 es Muy Pequeño Y X_2 es Muy Pequeño... Y X_{50} es Muy Grande ENTONCES el dinagrama pertenece a la clase 1 (Interferencia de Gas). En la figura 130 se muestra el conocimiento en forma de reglas (32 Reglas) para ser empleada en un sistema embebido.

8.2 VARIABLES Y CONJUNTOS FUZZY.

Como se explicó en el proceso de entrenamiento, el conocimiento de la Red Neuro-Fuzzy está inmerso en los pesos de conexión de la capa de entrada con la capa de reglas. A medida que la Red se entrena los conjuntos Fuzzy asignados a cada variable Fuzzy, cambian de posición llegando a un estado final cuando la salida de la Red, es la esperada. A continuación se presentan algunas variables con sus respectivos conjuntos ya modificados.

Figura 131. Conjuntos Fuzzy de la variable X_2 .

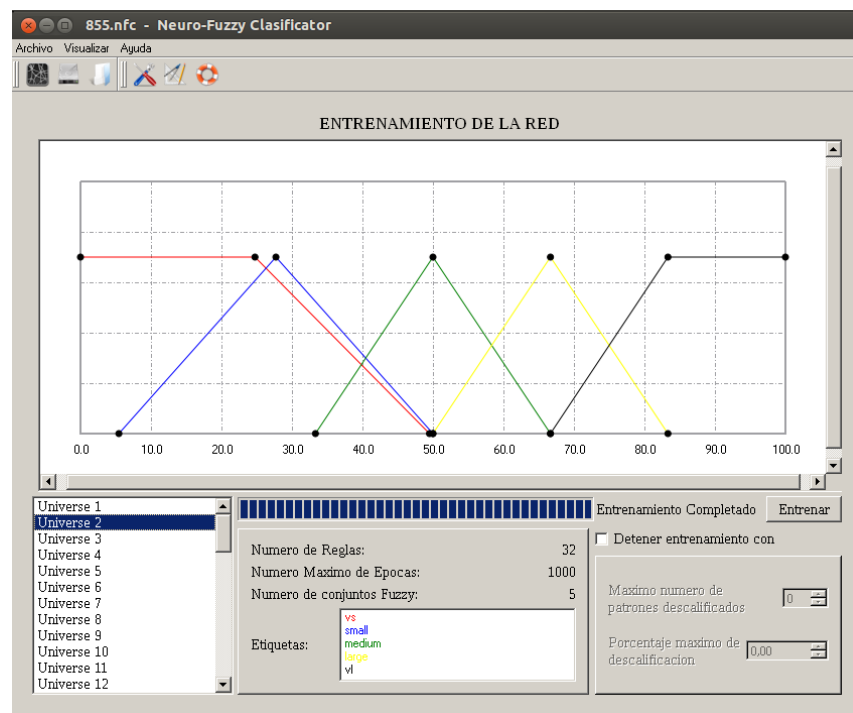


Figura 132. Conjuntos Fuzzy de la variable X_{12} .

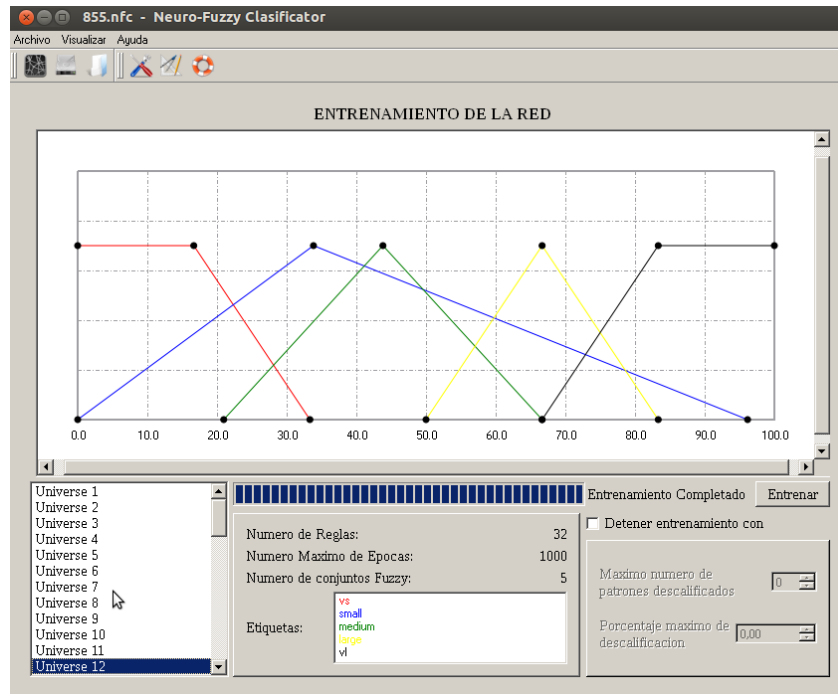


Figura 133. Conjuntos Fuzzy de la variable X_{15} .

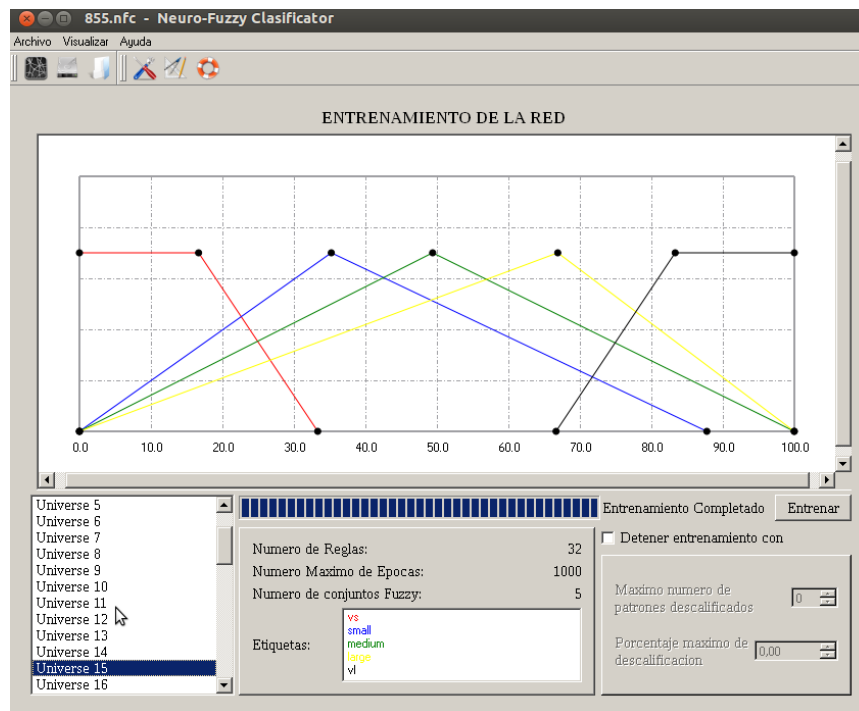


Figura 134. Conjuntos Fuzzy de la variable X_{40} .

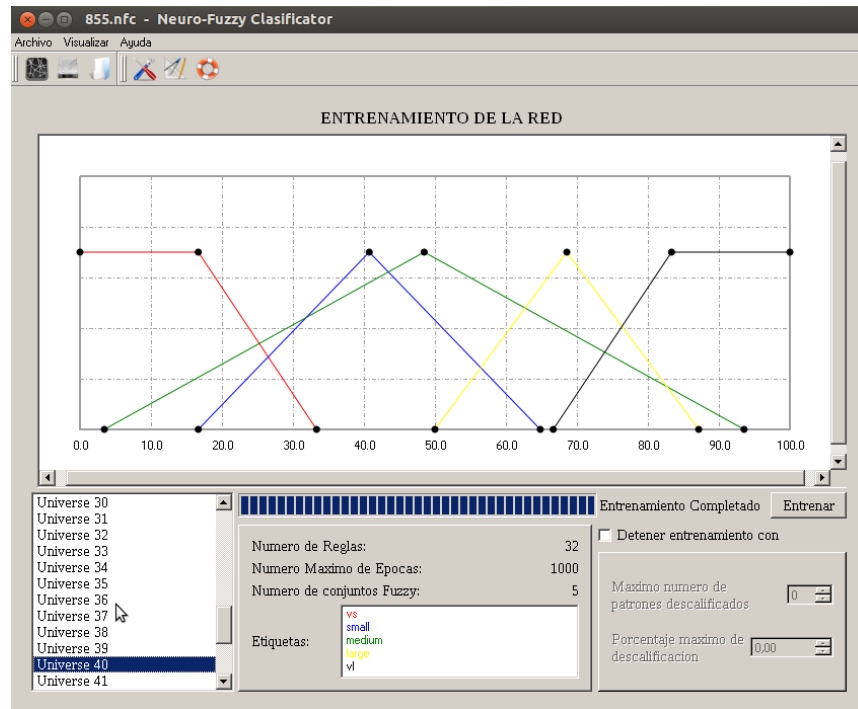
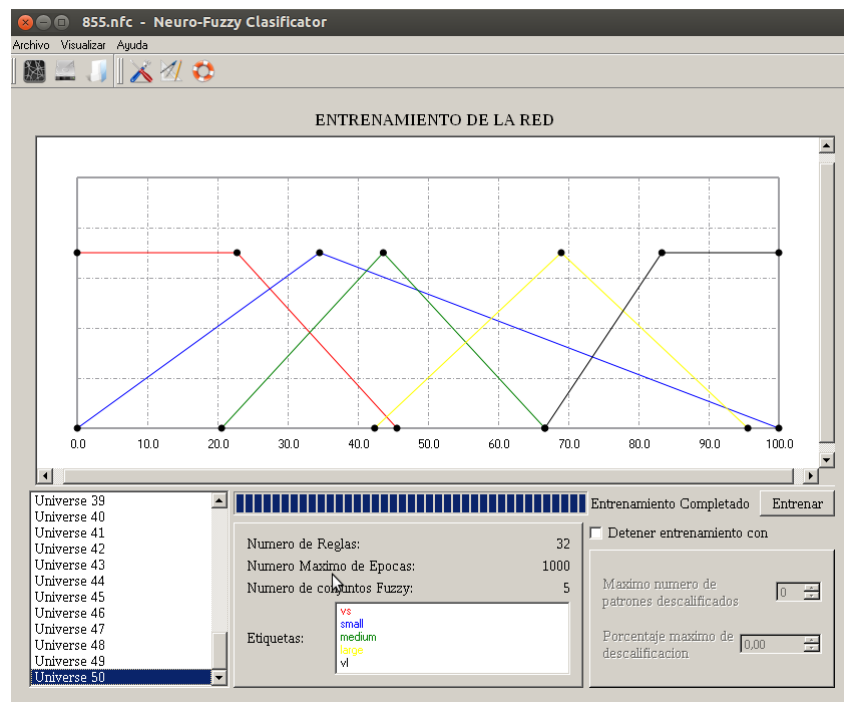


Figura 135. Conjuntos Fuzzy de la variable X_{50} .



CONCLUSIONES

Conclusión general:

- Al desarrollar la evaluación del sistema Neuro-Fuzzy, aplicado a la clasificación de dinagramas de fondo, se concluyó a partir de los resultados obtenidos; que las perspectivas que brinda esta tecnología son muy satisfactorias y merecen ser considerados en el proyecto de investigación “Desarrollo de un prototipo de pozo inteligente para Campo Escuela Colorado”.

Conclusiones específicas:

- Se cumplieron cabalmente los objetivos propuestos en el plan de proyecto, comprobando la factibilidad de utilizar una arquitectura Neuro-Fuzzy en el diagnóstico de falla de equipos de levantamiento artificial.
- La forma de identificación de un patrón cualquiera al software Nefclass-Q, va de acuerdo la facilidad de su discretización. Es decir, la forma del espectro no es tan relevante, lo verdaderamente importante es la organización y el orden de los patrones en los archivos de entrenamiento, clasificación y validación.
- Una arquitectura Neuro-Fuzzy presenta resultados sobresalientes para su aplicación en el campo del diagnóstico automático, basados en las pruebas realizadas donde obtuvieron validaciones hasta del 98% precisión y un 98,33% en porcentaje de operación o clasificación.
- Para la creación de los archivos de entrenamiento, validación y clasificación se debe tener en cuenta la rigidez con la que se deben organizar los patrones, ya que por un solo espacio que se pasa por alto, o una mala posición de alguno de los datos en el archivo, hace que el archivo fuese irreconocible por el software.
- El método de discretización usado es apropiado para hacer legible la información contenida en los dinagramas al software Nefclass-Q.

- Se clasificaron dinagramas reales obtenidos por expertos, obteniendo porcentajes de clasificación bastante aceptables, de 9 propagados, se diagnosticaron 6 generando un porcentaje de clasificación del 66.66% para su aplicación en pozos inteligentes posteriores a crear.
- El modo en que se organizan los patrones afecta el resultado de validación de la Red creada, como se observa en las tablas del capítulo 5, siendo los archivos organizados de forma escalonada los que arrojan mejores resultados.
- Cuando se utiliza un número grande de conjuntos Fuzzy en el pruning simple, se generan un gran número de reglas y por ende el resultado de la validación no es bueno.
- El valor adecuado del coeficiente de aprendizaje depende del problema (el dominio de las variables), pero cuando se utiliza un coeficiente de aprendizaje pequeño el entrenamiento de la red es menos confiable. Por en pruning de las mejores reglas, entre más pequeño es que coeficiente de aprendizaje, el porcentaje de validación disminuye notoriamente.

RECOMENDACIONES

- La utilización de una arquitectura Neuro-Fuzzy, debe conducir a sistemas unificados que faciliten el mantenimiento predictivo en línea, integrando directamente a la captación de la señal y la clasificación de estas señales. Es por esto que se propone la extracción del mejor entrenamiento en modo de reglas (Sistema Fuzzy) para su conversión en código c y así crear todo un sistema embebido.
- Ampliar el estudio presentado en este trabajo a espectros dinagráficos donde se presentan dos fallas o más, que es en realidad como se presenta en el campo.
- Aumentar el número de fallas en el entrenamiento y así abarcar un rango más amplio de reconocimiento y diagnóstico.
- Mejorar el proceso de aprendizaje, reutilizando los dinagramas clasificados en el proceso de diagnóstico con el objetivo de reforzar los ejemplos de fallas en entrenamiento y generar una base de conocimiento más consistente.

BIBLIOGRAFIA

- ✓ ALEGRE, Lideniro; DA ROCHA, A.F. y MOROOKA, C.K. Intelligent Approach of Rod Pumping Problems. Paper 26253. Presentando en Computer Conference, 11-14 Julio, New Orleans, Louisiana 1993.
- ✓ BEZERRA, Marco. SCHNITMAN, Leizer. BARRETO, Filho and SOUZA, Felipe. Pattern recognition for downhole dynamometer card in oil rod pump system using artificial neural networks, submitted to the 11th International Conference on Enterprise Information Systems (ICEIS'09) to be held in Milan, Italy, May 2009.
- ✓ CABALLERO ESTEBAN, Fabio Andrés; SALAMANCA ESPINOSA, Julián Sebastián. Herramienta de aplicación software para clasificación de patrones de datos implementando una arquitectura Neuro-Fuzzy. Bucaramanga, 2011, 222 p. Trabajo de grado (Ingeniero Mecánico). Universidad Industrial de Santander.
- ✓ DEL BRÍO Martín, Bonifacio; SANZ Molina, Alfredo. Redes Neuronales y Sistemas Borrosos. 3 ed. Alfaomega Ra-Ma, 2007. 404 p.
- ✓ GUERRERO TORRES, Joaquín Enrique; RODRIGUEZ MARTINEZ, Carlos Eduardo. Sistema Neuro-Fuzzy: prospectivas de aplicación en la detección de fallas en equipos industriales rotativos. Bucaramanga, 1998, 160 p. Trabajo de grado (Ingeniero Mecánico). Universidad Industrial de Santander.
- ✓ KOLMAN, Eyal; MARGALOT, Michael. Knowledge-Based Neurocomputing: A Fuzzy Logic Approach. Springer. Poland. p. 1-28. 2008.

- ✓ KONAR, Amit. Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain. CRC Press. London. 2000.
- ✓ LUI, Shuping. LUI, Yintao. BALOGUN, Oluwafemi. OLABINJO Lanre, SOMA, Ram, IVANHOE, Jared, SMITH, Byron. SEREN, Burcu. LENZ, Tracy. CINNAPPARAJA, Dineshbabu. ERSHAGHI, Iraj; Dinesh Babu C. G., Automatic Early Fault Detection for Rod Pump Systems, SPE 146038, 2011.
- ✓ MATEUS MARIN, Cristian Fernando; ROJAS CÁRDENAS, Daniel Mauricio. Análisis y diagnóstico de problemas en el equipo de subsuelo de unidades de bombeo mecánico utilizando redes neuronales. Bucaramanga, 2007, 237 p. Trabajo de grado (Ingeniero de Petróleos). Universidad Industrial de Santander.
- ✓ PONCE CRUZ, Pedro. Inteligencia Artificial con Aplicaciones a la Ingeniería: Alfaomega, 2010. 348 p.
- ✓ THETA ENTERPRISE, INC. Optimización de Bombeo Mecánico. Bera, USA. [CD-ROM].

ANEXOS

ANEXO A

BOMBEO MECÁNICO

El bombeo mecánico es el método más usado en el mundo. Consiste una bomba de subsuelo de acción reciprocante. La energía es suministrada a la bomba por medio de un motor eléctrico o de combustión interna y esta es transmitida a la bomba mediante la sarta de varillas. Tiene su mayor aplicación mundial en la producción de crudos pesados y extra pesados, aunque también se utiliza en la producción de crudos medianos y livianos.

Tabla 19. Ventajas y desventajas del bombeo mecánico.

Ventajas	Desventajas
<ul style="list-style-type: none">• El diseño es poco complejo.• Puede bombear flujos viscosos y a altas temperaturas.• Puede usar gas o electricidad como fuente de energía.• El sistema es eficiente, simple y fácil de operar.	<ul style="list-style-type: none">• Está limitado a una profundidad de 16000 ft.• El equipo de superficie es pesado y voluminoso.• Susceptible a bloqueo por excesivo gas en la bomba.

Fuente: http://industria-petrolera.blogspot.com/2008/01/mtodos-de-levantamientoartificial_12.html

Figura 136. El bombeo mecánico.



Fuente: www.bluebrain.ca

EL BOMBEO MECÁNICO Y SUS PARTES

El método de levantamiento artificial más común y antiguo del mundo es el bombeo mecánico. Debido a su simplicidad y robustez, es posible su aplicación en casi todos los tipos de pozos que requieren levantamiento. Sin embargo, existen límites físicos para la aplicación en cuanto a profundidad y caudales a levantar. El objetivo de un buen diseño en levantamiento artificial es lograr un sistema económicamente rentable, que logre el mayor Valor Presente Neto (VPN) considerando los costos asociados y la producción del pozo.

El yacimiento que ha de producir por bombeo mecánico tiene cierta presión, suficiente para que el petróleo alcance un cierto nivel en el pozo. Por tanto, el bombeo mecánico no es más que un procedimiento de succión y transferencia casi continúa del petróleo hasta la superficie. La función del sistema de bombeo mecánico por varillas es transmitir la potencia hasta la bomba de fondo para levantar los fluidos del yacimiento hasta la superficie. Los componentes del bombeo mecánico se pueden dividir básicamente en dos grupos:

- ✓ Equipo de superficie
- ✓ Equipo de sub-suelo.

Equipo de superficie.

Su función es transmitir energía del motor a la sarta de varillas. Para hacer esto, el equipo debe cambiar el movimiento rotatorio del motor, a un movimiento recíprocante en las varillas. La velocidad del motor debe ser reducida a una velocidad de bombeo adecuada como se ve en la *figura 141b*.

La reducción de velocidad se logra en el reductor de engranajes, y al resto del equipo le concierne el cambio de movimiento rotatorio en recíprocante.

Figura 137. Equipo de superficie.

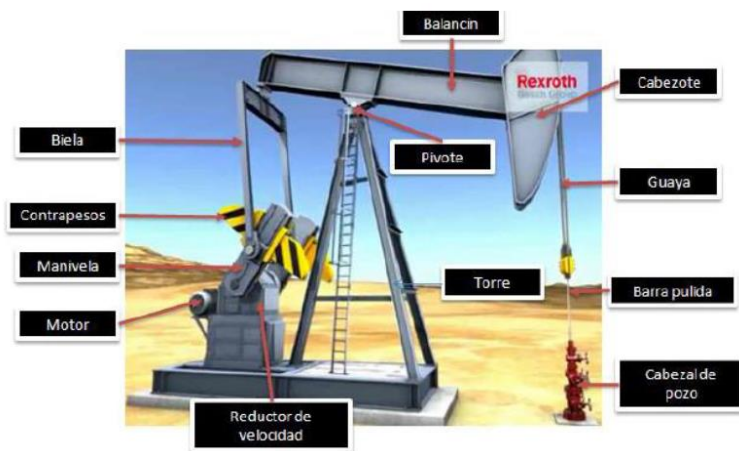


Figura 138. Movimiento rotatorio a recíprocante.



Fuente: http://www.boschrexroth.com/country_units/europe/spain/es/index.jsp.

Unidad de bombeo.

La función de la unidad de bombeo es convertir el movimiento rotatorio del motor en movimiento ascendente y descendente de la sarta de varillas. Este movimiento es denominado recorrido. Los balancines son estructuras muy fuertes y que aguantan muy bien las duras condiciones de trabajo. Toda la estructura debe ser colocada sobre una base de acero o concreto que asegure su estabilidad y alineamiento.

En el diseño apropiado de la unidad de bombeo, se debe de tener en cuenta un tamaño apropiado de la caja reductora y de la estructura, también se debe de tener una longitud de recorrido acorde con la producción de fluido que se desee.

Descripción de los componentes en la unidad de bombeo.

La Torre es el componente más fuerte ya que debe soportar las mayores cargas. Puede tener tres o cuatro patas. En el tope de la Torre está la Silla o pivote de la viga viajera (balancín). La viga debe soportar los esfuerzos de flexión que se generan en sus extremos por la carga del pozo y el empuje en los brazos. La viga es el punto de referencia para la alineación longitudinal de la unidad y la barra pulida. Si no hay una buena alineación longitudinal, la guaya (que es el elemento flexible que se encarga de mantener vertical la barra pulida, cuando el balancín realiza su movimiento), se puede ir rompiendo poco a poco con el roce de la pestaña del Cabezote y ocasionar un accidente y que el pozo se pare. Si la desalineación es muy severa, el prensa-estopas y la barra pulida se dañarán y ocasionarán derrames. La mejor forma de chequear la alineación longitudinal es colocándose a unos 20 metros detrás del balancín y observar si el centro de la viga está centralizado con la barra pulida. En el extremo de la viga del lado del pozo está el Cabezote que a través de la Guaya y Brida mueven la barra pulida. El Cabezote es curvado en su parte delantera para asegurar que solamente toque la barra pulida cuando está horizontal. Al igual que la viga, el Cabezote es la referencia para la alineación vertical de la unidad de bombeo. Esta alineación

controla la distancia entre el balancín y el cabezal del pozo. Se debe poner especial atención a este punto porque si no hay una buena alineación vertical, el Cabezote puede inducir un doblamiento en la barra pulida.

Los brazos conectan la manivela con la viga mediante los pines que transforman el movimiento rotatorio de la manivela en oscilatorio en la viga. La manivela, a su vez, está conectada al eje de baja de la caja de engranajes la cual recibe, a través de un juego de correas y poleas (Engranaje y Motor) conectadas al eje de alta la energía suministrada por el motor. Al otro extremo del eje de alta se ha colocado el tambor del freno que se conecta a través de un cable flexible a la palanca situada generalmente cerca del motor.

En la manivela están las pesas o contrapesas las cuales van montadas sobre rieles para facilitar su desplazamiento para efectos del balanceo de la unidad. Las cajas de engranajes pueden ser de simple o doble reducción, siendo estas últimas las de mayor uso. La lubricación es por salpicadura y cuando la velocidad es menor de 5 SPM es recomendable instalar un set de lubricación forzada. El balancín es un mecanismo que requiere poco mantenimiento para el trabajo que realiza y como todo aparato de grandes dimensiones, hay que moverse alrededor de él con mucho cuidado y no confiar en los frenos ni en el *switch* (interruptor) del panel.

ANEXO B

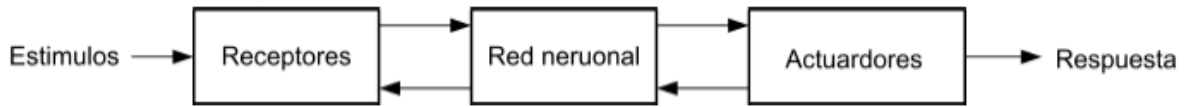
REDES NEURONALES

Una red neuronal artificial (RNA) es un modelo computacional inspirado en redes neuronales biológicas que puede ser consideradas como un sistema de procesamiento de información con características como aprendizaje a través de ejemplos adaptabilidad, robustez, capacidad de generalización y tolerancia a fallas. La RNA puede ser definida como una estructura distribuida, de procesamiento paralelo, formada de neuronas artificiales (llamadas también elementos de procesamiento o nodos), interconectados por un gran número de conexiones (sinapsis), los cuales son usados para almacenar conocimiento, con el fin de que esté disponible para poder ser usado.

El cerebro humano

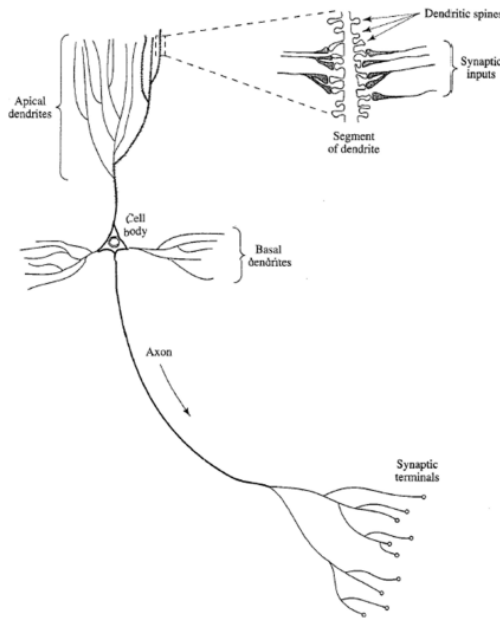
El sistema nervioso humano es visto como un sistema de tres etapas, donde el centro del sistema es el cerebro, representado por una red neuronal, la cual continuamente recibe información, la percibe, y toma decisiones apropiadas. Los receptores convierten los estímulos del cuerpo humano o del ambiente en impulsos eléctricos que transportan información a la red neuronal. Los actuadores transportan los impulsos eléctricos generados por la red neuronal o cerebro, respondiendo como un sistema de salida como se observa en la figura 153. La sinapsis son unidades funcionales y estructurales que promedian las interacciones entre las neuronas. El tipo más común de sinapsis es la química, la cual opera de la siguiente manera:

Figura 139. Sistema nervioso humano.



Un proceso pre-sináptico libera una sustancia transmisora que se difunde a través de los empalmes sinápticos entre las neuronas y luego actúan en un proceso post-sináptico. Así de este modo, una sinapsis convierte una señal eléctrica en una química, y luego de nuevo en una señal eléctrica post-sináptica. La plasticidad permite que el sistema nervioso central se adapte a sus alrededores. En un cerebro adulto, la plasticidad se puede dar por dos mecanismos. La creación de nuevas conexiones sinápticas entre neuronas y por la modificación de sinapsis existentes. Los Axones que son las líneas de transmisión, y la dendritas que son las zonas receptoras, son dos tipos de células que son distinguidas por su morfología, un axón tiene una superficie más lisa, menos ramificada y de mayor longitud que cualquier dendrita que tiene una superficie irregular y ramificada. Las neuronas vienen de una amplia variedad de formas y tamaños en diferentes partes del cerebro. La figura 154 muestra una figura de una célula en forma de pirámide. La mayoría de las neuronas codifican sus salidas como una serie de pulsos de voltaje breves, estos pulsos comúnmente conocidos como potencial de acción originados en el interior del cuerpo de las neuronas y luego se propagan a través de cada neurona individual a una velocidad y amplitud constante, la razón del uso de los potenciales

Figura 140. Célula piramidal.



De acción en la comunicación entre neuronas se basa en el aspecto físico del axón. El axón de una neuronal es largo y delgado lo cual es característico de alta resistencia y gran capacitancia, Este entonces podría ser una línea de transmisión RC.

El análisis del mecanismo de propagación en el axón revela que cuando el voltaje es aplicado en un extremo de él, este cae exponencialmente con la distancia, en cantidades insignificantes hasta alcanzar el otro extremo del axón.

Se estima que el cerebro humano tiene más de cien mil millones de neuronas y 10^{14} sinapsis en el sistema nervioso. Los estudios realizados sobre la anatomía del cerebro humano concluyen, que hay en general más de mil sinapsis por término medio en cada entrada y salida de cada neurona. Aunque el tiempo de conmutación de las neuronas biológica es casi un millón de veces mayor que en los actuales componentes de las computadoras, la conectividad de las neuronas naturales es de mil veces la de las neuronas artificiales. En conclusión el objetivo principal de las neuronas biológicas es desarrollar aplicaciones de síntesis y procesamiento de información.

Las redes neuronales artificiales son un procesador masivo distribuido en paralelo constituido por unidades simples de procesamiento (nodos), las cuales tienen una tendencia natural de almacenamiento de conocimiento experimental para tenerlo disponible para algún específico. Estas se parecen al cerebro en dos aspectos:

- El conocimiento es adquirido por la red del ambiente a través de procesos de aprendizaje.
- La fuerza de las interconexiones neuronales, conocidos como pesos sinápticos, son usados para el almacenamiento del conocimiento adquirido.

El procedimiento para lograr el aprendizaje de la red se llama algoritmo de aprendizaje, cuya función es modificar los pesos sinápticos de una manera ordenada para alcanzar el objetivo deseado del diseño.

Ventajas de las Redes Neuronales

Las redes neuronales proporcionan las siguientes utilidades y capacidades:

- No linealidad: Una neurona artificial puede ser lineal o no lineal. Una red neuronal está constituida por interconexiones de neuronas no lineales. Además la no linealidad es de un tipo especial, porque esta es distribuida a través de la red. La no linealidad es una propiedad de alta importancia, si la fuente física responsable de la generación de las señales de entrada es no lineal.
- Aprendizaje a partir de ejemplos: Un paradigma popular de aprendizaje llamado aprendizaje con un profesor o aprendizaje supervisado modifica los pesos sinápticos de una red neuronal por la aplicación de un conjunto de entrenamientos. Cada ejemplo consiste de una única señal de entrada y su correspondiente respuesta deseada. Se le presenta un ejemplo escogido al

azar de un conjunto, a la red neuronal, y los pesos sinápticos de la red son modificados para minimizar la diferencia entre la respuesta deseada y la respuesta actual de la red. El entrenamiento de la red es repetido para muchos ejemplos hasta que esta alcanza un estado seguro donde no haya un cambio significativo en los pesos sinápticos.

- **Adaptabilidad:** Una red neuronal artificial tiene la capacidad de adaptar sus pesos sinápticos cuando cambia su ambiente, en particular una red entrenada para operar en un ambiente específico, puede ser fácilmente entrenada de nuevo para operar con pequeños cambios en las condiciones de operación. Además cuando se requiere que la RNA opere en unas condiciones no estacionarias, la red puede ser diseñada para cambiar sus pesos sinápticos en tiempo real. La arquitectura natural de una RNA para clasificación de patrones de datos, procesamiento de señales, y control de procesos está relacionada con la capacidad de adaptabilidad de la red. Como regla general, se puede decir que sí el sistema tiene mayor adaptabilidad, su estabilidad será mejor.
- **Evidencia de la respuesta:** En el contexto de los patrones de clasificación, una red neuronal puede ser diseñada no solamente para proporcionar información acerca del patrón seleccionado, sino también de la confiabilidad de la decisión tomada. Esta información puede ser usada para rechazar patrones ambiguos que pueden surgir, y de este modo mejorar el rendimiento de clasificación de la red.
- **Información contextual:** El conocimiento es representado por varias estructuras y activaciones de estado de la red. Toda neurona de la red es potencialmente afectada por la activación global de las demás neuronas de la red. Consecuentemente, la información contextual es manejada de una forma natural por una RNA.
- **Tolerancia a fallas:** Una RNA implementada en hardware tiene el potencial de tolerar fallas, o capaz de manejar arquitectura de computación robusta

en el sentido que su rendimiento se degrada poco bajo condiciones adversas de operación, por ejemplo si la conexión de una neurona es dañada, la memoria es afectada en calidad, sin embargo debido a la distribución natural de la información almacenada en la red, el daño debe ser amplio para poder afectarla seriamente, mas generalmente la RNA muestra una disminución de su rendimiento en vez de una falla catastrófica.

- Implementable: El natural procesamiento en paralelo que tiene la RNA hace que sea potencialmente rápido a la hora de computar ciertas tareas.
- Uniformidad de análisis y diseño: Básicamente las redes neuronales disfrutan de universalidad como procesador de información, en el sentido que la misma notación es usada en todos los dominios de aplicación de las redes neuronales. Estas características se manifiestan de diferentes maneras:
 - ✓ Neuronas: Representan de una u otra forma un ingrediente común a todas las redes neuronales.
 - ✓ Estas cosas en común hace posible que se comparta teorías y algoritmos de aprendizaje en diferentes aplicaciones de las redes neuronales.
 - ✓ Las redes modulares pueden construirse a partir de la integración de módulos.
- Analogía con el cerebro humano: El diseño de una red neuronal está inspirado en el cerebro humano, el cual es una prueba viviente de la tolerancia a fallas del procesamiento en paralelo no solamente físicamente, sino también en rapidez y potencia.

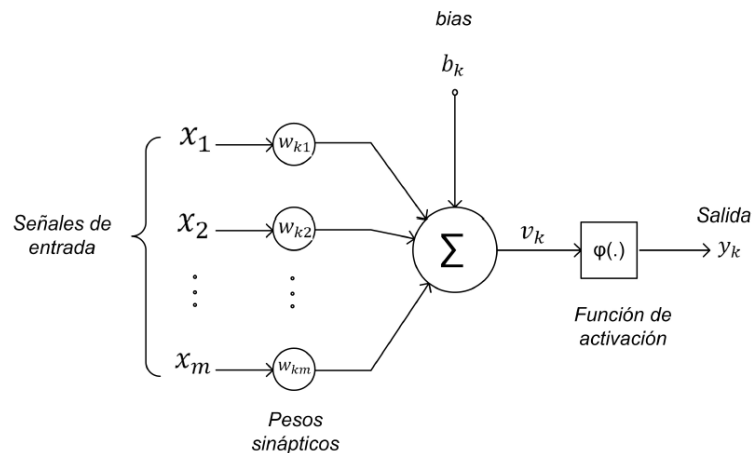
Modelo de una neurona

Una neurona es una unidad de procesamiento de información que es un fundamental en la operación de las redes neuronales. El diagrama de bloques de

la figura 115 muestra el modelo de una neurona, la cual está conformado por tres elementos básicos:

- Un conjunto de sinapsis o conexiones, donde cada una se caracteriza por un peso o fuerza de la conexión, específicamente una señal x_j a la entrada de la sinapsis j conectada a una neurona k es multiplicada por el peso sináptico w_{kj} , donde el primer subíndice se refiere a la neurona en cuestión y el segundo hace referencia a la entrada. A diferencia de las sinapsis en el cerebro humano, los pesos sinápticos de una neurona artificial puede tomar tanto valores positivos como negativos.

Figura 141. Modelo de una unidad de procesamiento.



- Un sumador que suma las señales de entrada por los pesos de la respectiva sinapsis de cada neurona.
- Una función de activación para limitar la amplitud de salida de una neurona. Esta función de activación se refiere a una squashing function que limita el rango de amplitudes permisibles de la señal de salida a un valor finito. Típicamente el rango de amplitudes normalizadas a la salida de una neurona es el intervalo cerrado $[0,1]$ o alternativamente $[-1,1]$.

El modelo de la figura también incluye la aplicación externa de *bias*, denotada como b_k . Las bias tienen un efecto de incremento o reducción de la entrada neta a la función de activación, dependiendo de si esta es positiva o negativa. En términos matemáticos se podría escribir para una neurona k lo siguiente: $u_k = \sum_{j=1}^m w_{kj}x_j$ y $y_k = \varphi(u_k + b_k)$ donde x_1, x_2, \dots, x_m son señales de entrada, $w_{k1}, w_{k2}, \dots, w_{km}$ son pesos sinápticos de la neurona k , u_k es la combinación lineal de las señales de entrada, b_k son las bias, $\varphi(\cdot)$ es la función de activación, y y_k es la señal de salida de la neurona. El uso de las bias b_k tiene el efecto de afinación de la salida u_k en el modelo.

$$v_k = u_k + b_k$$

Tipos de funciones de activación

La función de activación denotada por $\varphi(v)$, define la salida de una neurona en términos del campo local inducido, hay tres tipos básicos de función de activación.

- Función umbral. Para este tipo de función de activación, mostrada en la figura 156 tenemos:

$$\varphi(v) = \begin{cases} 1 & \text{if } v > 0, \\ 0 & \text{if } v < 0. \end{cases}$$

Donde cada neurona es referida a la literatura del modelo *Pitts McCulloch*, en reconocimiento a sus trabajos hechos en 1943. En este la salida de la neurona toma un valor de 1, si el campo local de la neurona no es negativo, en otro caso toma 0. También se conoce como el modelo del todo o nada.

- Función lineal a trozos: tenemos que:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0,5 \\ 1 & \text{if } 0,5 > v > -0,5 \\ 0 & \text{if } v \leq -0,5 \end{cases}$$

Donde el factor de ampliación influye en la región lineal, donde se supone que va

Figura 142. Función umbral.

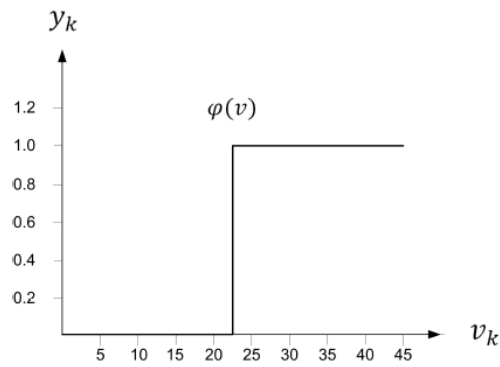
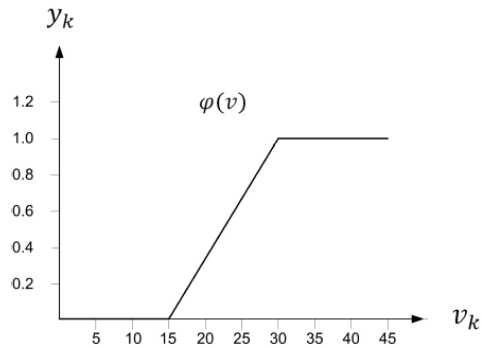


Figura 143. Función lineal a trazos.



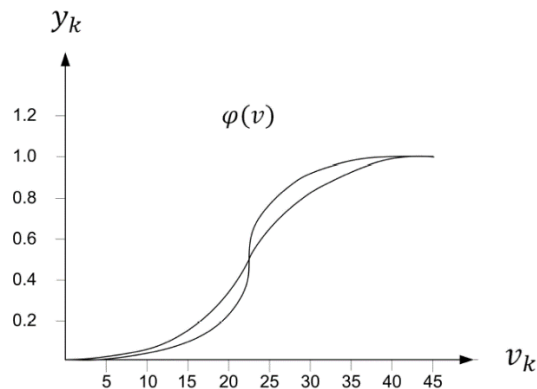
a estar la unidad. Esta función de activación puede ser vista como una aproximación a un amplificador lineal. Las siguientes situaciones pueden ser vistas como dos formas de función de activación a trozos.

- Función sigmoideal: Es la función de activación más usada en la creación de redes neuronales artificiales. Es definida como una función de incremento estricto que muestra un buen balance entre comportamiento lineal y no lineal. La figura 158 es un ejemplo de una función sigmoideal, es la función logística que se define como:

$$P = \frac{1}{1 + \exp(-av)}$$

Donde a es la pendiente de la función sigmoideal, si variamos a obtenemos funciones sigmoideales de diferente pendiente. Con valores de la pendiente límites la función sigmoideal empieza a simplificarse a una función umbral, pero mientras una función umbral da valores de 0 o 1, la función sigmoideal da un rango continuo de 0 a 1, esto quiere decir que es diferenciable.

Figura 144. Función sigmoideal.



Arquitectura de las redes neuronales

La manera con que las neuronas se organizan en la estructura de una red, está íntimamente ligado al algoritmo de aprendizaje en el entrenamiento de la red, En general se puede identificar tres tipos de arquitectura de RNA.

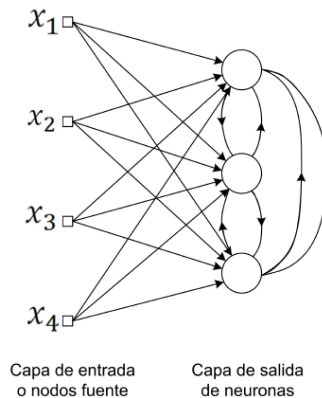
Redes de una capa con feedforward.

En una red neuronal, las neuronas están organizadas en forma de capas. En la forma más simple de una red se tiene una capa de entrada de nodos fuentes que se proyectan a las neuronas de la capa de salida, pero no viceversa, en otras palabras esta red es acíclica. Se dice que esta red es de una sola capa por su capa de salida, ya que la capa de entrada no se cuenta porque no hace ningún proceso.

Redes Neuronales multicapa

El segundo tipo de redes neuronal alimentadas hacia adelante se reconoce por la presencia de una o más capas, llamadas capas escondidas, y cuyas unidades de procesamiento son llamadas neuronas escondidas o unidades escondidas. La función de estas neuronas es intervenir entre las entradas externas y las salidas de la red de una manera indicada, por medio de la adición de una o más capas a la red, estas quedan a disposición de extraer datos estadísticos de orden más altos, esta habilidad de las unidades escondidas es muy importante cuando la cantidad de entradas a la red son grandes. Los nodos fuente de la capa de entrada suministran

Figura 145. Red Neuronal de una capa feedforward.



patrones de activación, los cuales así mismo constituyen la señal de entrada aplicada a las neuronas de la segunda capa, las salidas de esta capa son usadas como entrada a la tercera capa, y así sucesivamente para el resto de la red. Típicamente las neuronas tienen como señal de entrada las salidas de la capa anterior. El conjunto de salidas de las neuronas de la capa final o de salida, son así mismas la respuesta global de toda la red al patrón de activación suministrado por unidades de la capa de entrada.

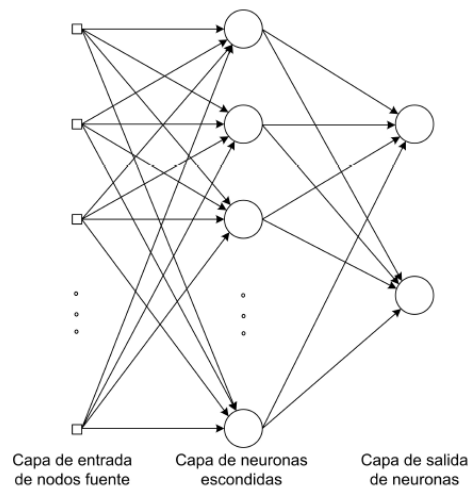
Redes neuronales recurrentes

Una red neuronal recurrente se reconoce fácilmente por su forma, ya que tiene una pequeña retro-propagación, por ejemplo una red recurrente puede consistir de una sola capa de neuronas, en la que cada salida de ellas alimenta las entradas de cada neurona, algunas de estas redes contienen self-feedback que se refiere cuando la salida de la neurona es retroalimentada a su propia entrada. La presencia del circuito de retroalimentación tiene un gran impacto en la capacidad de aprendizaje de la red y en su rendimiento, además el circuito de retroalimentación implica ramificaciones particulares, las cuales dan como resultado un comportamiento no lineal suponiendo que la red tiene unidades no lineales.

Procesos de aprendizaje

Esta es una propiedad muy importante para una red neuronal, ya que de esta depende la capacidad de aprender del ambiente y mejorar su rendimiento a través del proceso de aprendizaje. Una red neuronal aprende de su ambiente a través de procesos iterativos que ajustan los pesos sinápticos y las bias. Idealmente, la red empezara a adquirir más conocimiento con cada iteración en el proceso de aprendizaje.

Figura 146. Red neuronal multicapa.



Definición de aprendizaje

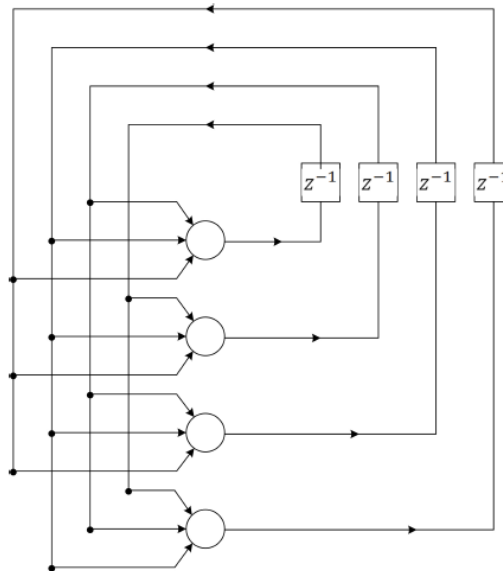
El aprendizaje es un proceso por medio del cual los parámetros libres de una red neuronal son adaptados a través de procesos de simulación del ambiente en el que la red está sometida. El tipo de aprendizaje es determinado por la manera en que los parámetros cambian (Mendel y McClaren 1970). Esta definición de proceso de aprendizaje implica la siguiente secuencia de eventos:

- La red neuronal es estimulado por el ambiente.
- La red sufre cambios en sus parámetros libres debido a la estimulación.

- La red responde al ambiente de una forma distinta debido a los cambios que ha sufrido en la estructura interna.

Un conjunto de reglas preestablecidas y bien definidas para solucionar el proceso de aprendizaje es llamado algoritmo de aprendizaje. Existen muchos algoritmos de aprendizaje para el diseño de las redes neuronales, en el que cada uno de ellos ofrece sus propias ventajas. Básicamente los algoritmos difieren el uno del otro en la forma como ajustan los pesos sinápticos de la red, otro factor muy importante es la manera de cómo están constituidas las interconexiones de las neuronas.

Figura 147. Red neuronal recurrente.



Aprendizaje de corrección del error

Para explicar este concepto, considere una red alimentada hacia adelante constituida por una unidad de procesamiento en la capa de salida como muestra la figura 160, a este neurona le llegan conexiones de una o más unidades escondidas de la capa oculta, las cuales así mismo se le aplica un vector de entrada proveniente de las neuronas fuentes de la capa de entrada. El argumento n indica el número de la iteración en el proceso de ajuste de los pesos sinápticos

de la red, la señal de salida es denotada como $y_k(n)$. Esta señal de salida representa la respuesta de la red, por lo tanto es comparada con la respuesta deseada $d_k(n)$. Por lo tanto la señal de error puede definirse como:

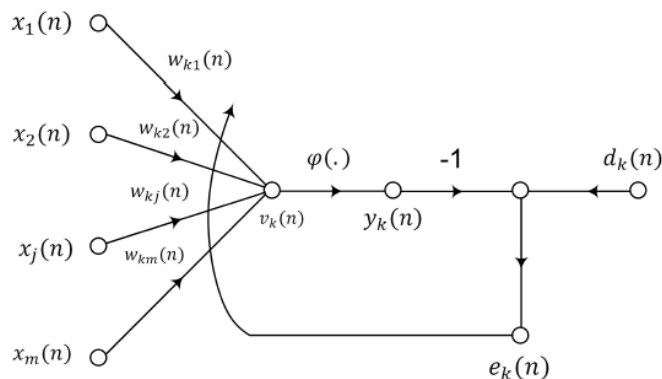
$$e_k(n) = d_k(n) - y_k(n)$$

La señal de error $e_k(n)$ actúa como un mecanismo de control, cuyo propósito es aplicar una secuencia de ajustes correctivos a los pesos sinápticos de la neurona k . El ajuste correctivo está diseñado de tal manera que la señal de salida $y_k(n)$ sea el mismo valor o aproximado a la respuesta deseada $d_k(n)$. El objetivo es minimizar la función de costo, definida en función del error como:

$$\xi(n) = \frac{1}{2} e_k^2$$

$\xi(n)$ es un valor instantáneo de la energía del error, el proceso de ajuste de los pesos continua hasta que estos se estabilizan, es decir, hasta que la variación de los pesos

Figura 148. Aprendizaje de corrección del error.



con cada iteración no cambia en un valor significativo. En particular la disminución de la función de costo guía la regla de aprendizaje comúnmente llamada como Regla delta o *Regla de Widrow-Hoff*, llamada así en honor a sus creadores. Si $w_{kj}(n)$ es el peso sináptico de la neurona k excitada por el elemento $x_j(n)$ del

vector de señal $x(n)$ en una iteración n . Siguiendo la regla delta, el ajuste del peso $\Delta w_{kj}(n)$ aplicado a los pesos sinápticos w_{kj} en la iteración n se define como:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n)$$

donde η es una constante positiva que determina la tasa de aprendizaje con la que se va a proceder de una iteración a otra en el proceso de aprendizaje, en otras palabras la regla delta puede ser definida como: El ajuste hecho a los pesos sinápticos de una neurona es proporcional al producto de la señal de error y la señal de entrada de la sinapsis en cuestión.

Teniendo calculado el ajuste del peso sináptico Δw_{kj} , la actualización del peso se da de la forma:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$$

donde los valores $w_{kj}(n)$ y $w_{kj}(n+1)$ pueden ser vistos como el valor anterior y el nuevo del peso sináptico respectivamente.

Aprendizaje de Hebbian

El aprendizaje de Hebbian es el más viejo y famoso de las reglas de aprendizaje, es llamado así en honor al neuropsicólogo HEBB. Cuando el axón de una célula A esta lo suficientemente cerca para excitar a la célula B, y se acerca en repetidas ocasiones, los cambios metabólicos toman lugar en una o ambas células, de modo que tanto la eficiencia de la célula A como la B se incrementan. HEBB propone estos cambios como base del proceso de aprendizaje a nivel celular, debido a esto se pueden deducir dos reglas:

- Si dos neuronas a cada la de la conexión o sinapsis son activadas simultáneamente, la fuerza del peso sináptico se incrementa.

- Si dos neuronas a cada lado de la conexión son activadas asincrónicamente, la sinapsis es debilitada o eliminada.

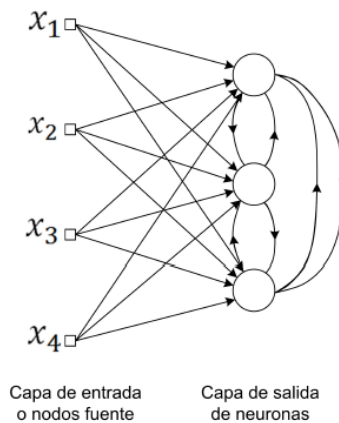
Las modificaciones del modelo matemático del aprendizaje de Hebb, considera un peso sináptico w_{kj} de una neurona k con señales de pre-sinapsis y post-sinapsis denotadas como x_j y y_k respectivamente, entonces el ajuste aplicado a los pesos sinápticos en la iteración n es expresado en forma general como:

$$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$

Aprendizaje competitivo

El aprendizaje competitivo como su nombre lo dice, es aquel donde las neuronas de salidas compiten entre ellas para llegar a ser activadas. En las redes neuronales basadas en el aprendizaje de Hebb puede haber varias neuronas de salidas que se activan al mismo tiempo, en el aprendizaje competitivo una sola neurona puede ser activada en algún tiempo. Esta característica hace que este aprendizaje sea usado en la clasificación de patrones de datos

Figura 149. Aprendizaje competitivo.



Hay tres elementos básicos en las reglas de aprendizaje competitivo.

- Un conjunto de neuronas que son parecidas excepto algunas que tienen una distribución aleatoria de los pesos y que responden de manera diferente para un patrón de entrada dado.
- Un límite impuesto a la fuerza de cada neurona.
- Un mecanismo que permite a las neuronas competir por la respuesta correcta a un subconjunto de entrada dado, y en la que una y solo una neurona será activada por grupo.

La forma más simple de aprendizaje competitivo, se da en la arquitectura con una sola capa de neuronas de salida, y en la que cada una de ellas está completamente conectada a las neuronas de entrada y la red puede incluir conexiones *feedback* como muestra la figura 163. Para una neurona k que será la neurona ganadora, su campo de inducción v_k para un patrón de entrada x será el más grande de todas las neuronas, la señal de salida y_k de la neurona ganadora es iguala 1, mientras que la salida de todas las otras neuronas perdedoras será 0.

$$y_k = \begin{cases} 1 & \text{if } v_k \geq v_j \\ 0 & \text{otro caso} \end{cases}$$

Los pesos sinápticos de las conexiones de unas neuronas están dados de tal forma que la sumatoria de cada uno de ellos de 1.

$$\sum_j w_{kj} = 1 \text{ para cada neurona } k$$

El ajuste de los pesos sinápticos Δw_{kj} para el aprendizaje competitivo está dado por:

$$\Delta w_{kj} = \begin{cases} \eta(x_j - w_{kj}) & \text{si la neurona es ganadora} \\ 0 & \text{si la neurona es perdedora} \end{cases}$$

donde η es la tasa de aprendizaje.

Aprendizaje de Boltzmann

Las reglas de aprendizaje de Boltzmann, es un algoritmo de aprendizaje estocástico derivadas de las ideas de la mecánica estadística. Una red neuronal diseñada tomando como base las reglas de aprendizaje de Boltzmann es llamada máquina de Boltzmann. Las neuronas de una máquina de Boltzmann constituyen una estructura recurrente, y operan de una manera binaria, por ejemplo, si la neurona esta activada entonces su estado es denotado por +1, o si esta desactivada su estado es denotado por -1. La máquina de Boltzmann es caracterizada por la Función de Energía, E, cuyo valor es determinado por estado particular de las neuronas de la máquina y se calcula de la siguiente manera:

$$E = -\frac{1}{2} \sum_j \sum_k w_{kj} x_k x_j$$

donde x_j es el estado de la neurona j, w_{kj} es el peso sináptico de la conexión de la neurona j con la neurona k. La máquina opera escogiendo una neurona aleatoria, por ejemplo una neurona k, en una iteración del proceso de aprendizaje, el estado de la neurona k va desde x_k a $-x_k$ a una temperatura T con la probabilidad:

$$P(x_k > -x_k) = \frac{1}{1 + \exp(-\Delta E_k / T)}$$

Las neuronas de la máquina de Boltzmann se dividen en dos grandes grupos: visibles y escondidas, las visibles proporcionan una interfaz entre la red y el ambiente, y las neuronas escondidas operan de manera libre. Hay dos formas de operar:

- Condición enlazada: en la que las neuronas visibles están enlazadas por un estado específico determinado por el ambiente.

- Operación libre: en el que todas las neuronas se les permite operar de forma libre.

Si ρ_{kj}^+ es la correlación entre el estado de la neurona j y k, con la red en condición de operación enlazada, y ρ_{kj}^- es la correlación entre estado de la neurona j y k, con la condición de operación libre, estas correlaciones son promediadas para todos los estados posibles de la maquina en el estado de equilibrio térmico. Entonces acordado con la regla de aprendizaje de Boltzmann, el ajuste del peso Δw_{kj} aplicado al peso sinápticos de la neurona j a la k es definido como:

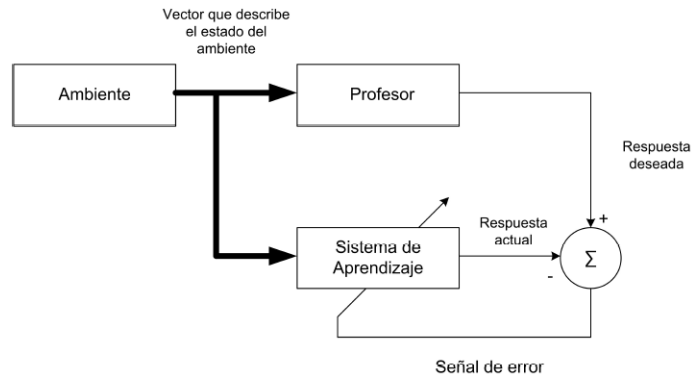
$$\Delta w_{kj} = \eta(p_{kj}^+ - p_{kj}^-) \quad j \neq k$$

Dentro de los algoritmos de aprendizaje se encuentran diversos tipos como los siguientes:

Aprendizaje con un profesor

La figura 164 muestra el proceso de aprendizaje con un profesor o supervisado, en términos conceptuales se puede pensar que el profesor es el que tiene el conocimiento del ambiente, y este conocimiento está representado como un conjunto de entradas y ejemplos de salida, el ambiente es sin embargo desconocido para la red de interés.

Figura 150. Aprendizaje supervisado.



Supongamos que el profesor y la red son expuestos a un vector de entrenamiento proporcionado por el ambiente, por virtud del conocimiento adquirido el profesor es capaz de proporcionar a la red una respuesta deseada para el vector de entrenamiento, esta respuesta deseada es la acción óptima de la red. Los parámetros de la red son ajustados bajo la influencia del vector de entrenamiento y la señal de error, donde la señal de error es definida como la diferencia entre la respuesta deseada y la respuesta actual de la red. Este ajuste es llevado en un proceso iterativo donde el objetivo es que la red emule al profesor, esta emulación se supone que es el estado óptimo en el sentido estadístico. En este camino, el conocimiento en el ambiente disponible al profesor es transferido en la mayor cantidad posible a la red neuronal.

Cuando la condición óptima es alcanzada, se puede prescindir del profesor y dejar que la red interactúe con el ambiente directamente. Para medir el rendimiento del sistema se define una función de los parámetros libres del sistema, esta función puede ser vista como un error de rendimiento superficial o error de superficie, este error es un promedio global de las posibles entradas y ejemplos de salida. En una operación dada del sistema bajo la supervisión del profesor, es representada como un punto en el error de superficie, para mejorar el rendimiento el punto de operación tiene que bajar hacia un punto donde el error de superficie es mínimo.

Aprendizaje sin profesor

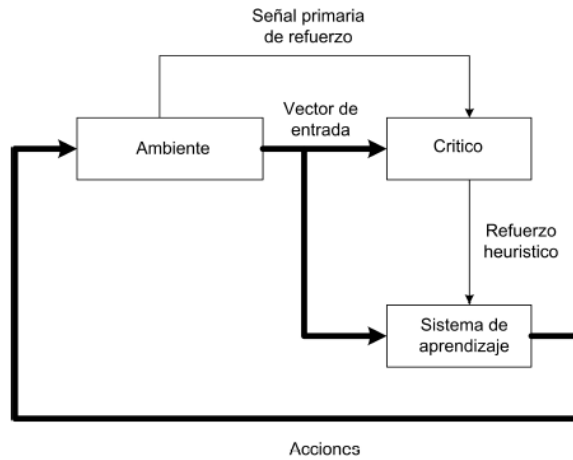
El aprendizaje supervisado se da bajo la tutoría de un profesor, sin embargo en el paradigma conocido como aprendizaje sin profesor como su nombre lo dice, no hay un profesor que supervise el proceso de aprendizaje, bajo este segundo paradigma existen dos clasificaciones:

- Aprendizaje reforzado: En el aprendizaje reforzado la asignación de entradas y salidas es realizado a través de continuas interacciones con el ambiente.

La figura 8.40 muestra un diagrama de bloques de una forma de sistema de aprendizaje reforzado construido alrededor de un elemento crítico que convierte una señal primaria reforzada recibida del ambiente a una señal de calidad más alta llamada señal heurística de refuerzo, donde ambas son escalares.

El sistema es diseñado para aprender bajo el concepto de refuerzo retardado, el cual significa que el sistema observa una secuencia temporal de estímulos, también recibidos del ambiente, y con las cuales resulta la generación de señales heurísticas reforzadas. El objetivo del aprendizaje es minimizar la función de costo, definida como la acumulación de costo de las acciones asumidas en cada paso de la iteración, en vez de un simple costo inmediato. Esto podría producir que algunas acciones tomadas en la iteración tempranamente sean las mejores de todo el sistema, la función de la máquina de aprendizaje, que constituye el segundo componente del sistema, es descubrir las acciones y alimentarlas de nuevo al ambiente.

Figura 151. Aprendizaje reforzado.

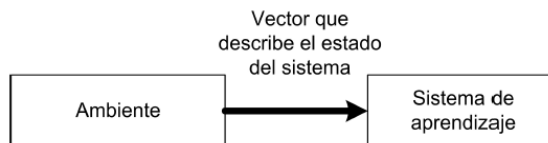


El aprendizaje reforzado en retraso es difícil de mejorar debido a dos aspectos básicos:

- ✓ No hay un profesor que suministre la respuesta deseada en cada paso del proceso de aprendizaje.
- ✓ El retraso incurre en la generación de señales primarias reforzadas y esto implica que la máquina de aprendizaje debe solucionar el problema de asignación de crédito temporal.

Aprendizaje no supervisado: En este aprendizaje no hay un profesor externo o crítico que supervise el proceso de aprendizaje, como indica la figura 166. El suministro es hecho por un medidor de tareas independiente de la calidad de la representación que la red requiere para aprender, y los parámetros libres de la red son optimizados por el medidor. Una vez que la red a llegado a afinar

Figura 152. Aprendizaje no supervisado.



Fuente: Los autores

las estadísticas regulares de la entrada de datos, esta desarrolla la habilidad de formar representaciones internas para codificar las características de la entrada y de este modo crear nuevas clases automáticamente. Para mejorar el aprendizaje no supervisado se debe usar una regla de aprendizaje competitiva, por ejemplo, se puede usar una red que consiste de dos capas, una capa de entrada y una capa competitiva, la capa de entrada recibe los datos disponibles, y la capa competitiva consiste de neuronas que compiten entre ellas por la oportunidad de responder a la entrada de datos.

ANEXO C

LÓGICA FUZZY

La lógica difusa o lógica Fuzzy es un conjunto de principios matemáticos basados en grados de membresía o pertenencia, cuya función es modelar información. Este modelado se hace con base en reglas lingüísticas que aproximan una función mediante la relación de entradas y salidas del sistema (composición). Esta lógica presenta rangos de membresía dentro de un intervalo entre 0 y 1, a diferencia de la lógica convencional, en la que el rango se limita a dos valores: el cero o el uno.

Mediante el uso de lógica difusa se puede representar la forma de la lógica humana, por ejemplo en afirmaciones como “el día es caluroso”, “el automóvil va muy rápido”, etc. En el primer caso, se sabe que hay alta temperatura, pero no se sabe a qué temperatura exactamente nos estamos refiriendo; en el segundo caso, se dice que “el automóvil va rápido”, sin embargo nunca se especifica su velocidad.

¿Por qué usar un sistema difuso? Si se requiere automatizar un proceso que controla un trabajador, el sistema difuso tendrá la tarea de emular a dicho trabajador. Además, si se toma en cuenta que el trabajador hace juicios con base en su criterio y experiencia, y que estos juicios y decisiones se realizan en forma lingüística (como “alto”, “lento”, etc.), se puede notar que un sistema convencional no maneja este tipo de entradas, mientras que el sistema difuso sí lo hace.

Otra ventaja del sistema de control basado en lógica difusa es que no es necesario conocer un modelo matemático del sistema real, pues se puede ver como una caja negra a la cual se le proporcionan entradas, y a través del sistema esta planta generará la salida deseada. En el control convencional sí es necesario conocer la planta del sistema.

Para desarrollar un control con estas características, es necesario un experto, en este caso el trabajador, del cual se tomará un registro de las situaciones que se le

presentan, así como de la solución que él les da. Esta experiencia se traduce en reglas que usan variables lingüísticas.

Variable lingüística

Una variable lingüística adopta valores con palabras que permiten describir el estado de un objeto o fenómeno; estas palabras se pueden representar mediante conjuntos difusos. Una variable numérica toma valores numéricos, por ejemplo: edad 5 65, mientras que una variable lingüística toma valores lingüísticos: edad es “viejo”. Todos los valores lingüísticos forman un conjunto de términos o etiquetas.

Para desarrollar un control con estas características, es necesario un experto, en este caso el trabajador, del cual se tomará un registro de las situaciones que se le presentan, así como de la solución que él les da. Esta experiencia se traduce en reglas que usan variables lingüísticas.

Para hacer este control es necesario tener las entradas del sistema y éstas se van mapear a variables lingüísticas. A este mapeo se le llama difusificación. Con estas variables se forman reglas, las cuales serán las que regirán la acción de control que será la salida del sistema.

La anatomía básica de un controlador difuso consta de tres partes:

Reglas: estas son reglas que dictan la acción de control que se va a tomar. Éstas se derivan de un experto. Dichas reglas tiene la estructura de relaciones. La lógica difusa se basa en relaciones, las cuales se determinan por medio de cálculo de reglas “SI-ENTONCES” (con las cuales se puede modelar aspectos cualitativos del conocimiento humano, así como los procesos de razonamiento sin la necesidad de un análisis cuantitativo de precisión). Un ejemplo de una regla sería:

Si la temperatura es alta entonces se debe de encender el ventilador.

Difusificador: es el nexo entre las entradas reales y difusas. Todas las entradas necesitan ser mapeadas a una forma en que las reglas puedan utilizarlas.

Desdifusificador: toma un valor difuso de las reglas y genera una salida real.

Diferencias entre lógica Fuzzy y Crisp

- Veracidad: En los sistemas clásicos, solo hay dos posibles valores de verdad, verdadero o falso. En la lógica fuzzy los valores de verdad pueden ser un subconjunto fuzzy de algún conjunto parcialmente ordenado, usualmente este valor se asume que está en el intervalo $[0,1]$ de un subconjunto fuzzy o simplemente un punto en este intervalo. Los llamados valores de verdad lingüísticos expresados como verdadero, muy verdadero, no muy verdadero, etc. son interpretados como etiquetas de subconjuntos fuzzy.
- Predicados: En los sistemas clásicos, los predicados son concretos, por ejemplo mortal, liso, más grande que. En la lógica fuzzy los predicados no son concretos, por ejemplo mal, pronto, veloz, mucho más grande que. Debe tenerse en cuenta que la mayoría de los predicados en el lenguaje natural no son concretos.
- Modificadores de predicado: En los sistemas clásicos es usado el modificador de predicado de negación Not. En la lógica fuzzy hay una variedad de modificadores de predicado como por ejemplo muy, más o menos, bastante, extremadamente.
Los modificadores de predicado juegan un papel importante en la generación de valores de una variable lingüística, por ejemplo muy joven, no muy joven, más o menos joven, etc.
- Cuantificadores: En la lógica clásica hay dos cuantificadores: universal y existencial. La lógica fuzzy tiene además de estas una amplia variedad de cuantificadores fuzzy, por ejemplo poco, varios, usualmente, la mayoría,

siempre, frecuentemente, etc. En la lógica fuzzy, un cuantificador fuzzy es interpretado como un número fuzzy o una proporción fuzzy.

- Probabilidades: En los sistemas de lógica clásica, es un valor numérico o intervalo numérico. En lógica fuzzy, se tiene la opción adicionalmente de emplear la lingüística o generalmente las probabilidades fuzzy, por ejemplo probable, no probable, muy probable, altamente probable. Las probabilidades fuzzy pueden ser interpretadas como números fuzzy, los cuales deben ser manipulados a través de la aritmética fuzzy.

Adicionalmente a las probabilidades fuzzy, la lógica fuzzy hace posible el manejo de los eventos fuzzy. Un ejemplo de un evento fuzzy es mañana será un día cálido, donde cálido es predicado fuzzy.

- Posibilidades: En contraste al modelo de lógica fuzzy, el concepto de posibilidad en lógica fuzzy es graduado en vez de concreto. Sin embargo como en el caso de las probabilidades, las posibilidades pueden ser tratadas como variables lingüísticas con valores como posible, poco posible, altamente posible, etc. Estos valores pueden ser interpretados como etiquetas de subconjuntos fuzzy de la línea real. Uno de los conceptos que juega un papel importante en la lógica fuzzy es la distribución de posibilidades.

Significado y representación del conocimiento

El conocimiento puede ser visto como una colección de proposiciones.

Mary es joven.

Pat es más alta que Mary.

170

La mayoría de los suecos son rubios.

Los tomates son más rojos si ellos están maduros.

Usualmente la alta calidad va acompañada de un alto precio.

La mayoría de los hombres altos no son muy ágiles.

En estas oraciones significado y representación están relacionadas entre sí. En la lógica fuzzy significado y representación están basados en el concepto de test score. Una idea fundamental del concepto test-score es que una proposición en un lenguaje natural puede ser vista como una colección de restricciones fuzzy elásticas, por ejemplo la proposición de Mary es alta representa una restricción elástica sobre la altura de Mary. Similarmente la proposición Jean es rubia representa una restricción elástica sobre el color del cabello de Jean y la proposición la mayoría de los hombres altos no son muy ágiles representa una restricción elástica sobre la proporción de los hombre que no son muy ágiles.

En términos más concretos, la representación y el significado en vuelve los siguientes pasos usando el test-score:

1. Identificación de las variables $X_1, X_2, X_3, \dots, X_n$, en la que los valores son restringidos por la proposición. Usualmente estas variables están implícitas en vez de explícitas en la proposición.
2. Identificación de las variables $C_1, C_2, C_3, \dots, C_n$ las cuales son introducidas por la proposición.
3. Caracterización de cada restricción C_i por la descripción del procedimiento de prueba que asocia C_i con un *test-score* T que representa el grado con el cual C_i es satisfecho. Usualmente T_i es expresado como un número en el intervalo $[0,1]$, sin embargo un test-score puede ser una probabilidad distribución de posibilidad en un intervalo.

4. La agregación de un test-score parcial. Los cuales son representados como un vector de test-score T_i, \dots, T_k . En la mayoría de los casos $K = 1$ entonces el test-score sería un escalar.

Cuantificación de la ambigüedad

La mayoría de los lenguajes naturales contienen ambigüedad y multiplicidad de significados, que están determinados por factores culturales y que pueden cambiar de forma brusca inclusive en comunidades vecinas. Los propósitos de los adjetivos, especialmente, no son claramente específicos, ellos son ambiguos en términos de la amplitud de su significado. Por ejemplo, si nosotros decimos “Persona alta”, no podemos determinar quién es más alto o quien es menos alto. La ambigüedad (imprecisión, incertidumbre) de “Persona vieja” proviene del adjetivo “viejo”. Las palabras son usualmente cualitativas, pero no persigue que “Alto” o “Viejo” sean percibida en conexión con cantidades asociadas a la altura o la edad.

Fundamento de la teoría de conjuntos fuzzy

Definición de conjunto fuzzy:

Un conjunto fuzzy μ de X es una función que asigna a los elementos del universo de discurso un valor dentro de un intervalo. $\mu: X \rightarrow [0; 1]$, donde $F(X)$ representa el conjunto de todos los conjuntos fuzzy de X .

El valor $\mu(x)$ representa el grado de pertenencia del elemento x al conjunto fuzzy

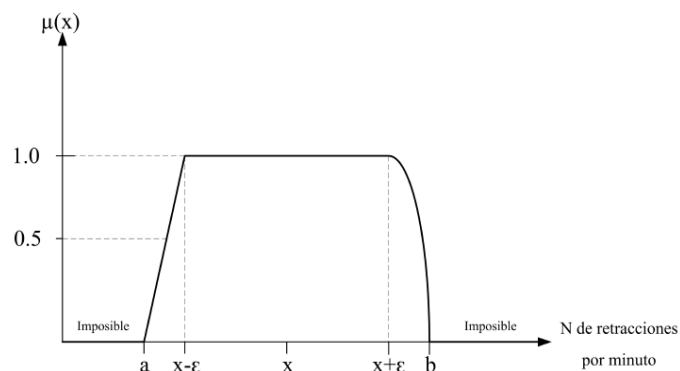
El uso de los conjuntos fuzzy es representar conceptos vagos y con frecuencia es hecho de manera intuitiva, porque en muchas ocasiones no existe un modelo que intérprete de manera clara el grado de pertenencia.

Por ejemplo sí x es el número de revoluciones por minuto del disco duro de un computador, obviamente esta información es afectada con incertidumbre, por la tanto es más real usar datos lingüísticos.

El número de rotaciones por minuto es cercano a x

El intervalo $[x - \epsilon ; x + \epsilon]$ puede ser usado para modelar, pero esto causa el problema mencionado de contar con límites concretos de $x - \epsilon$ y $x + \epsilon$. Sí los datos estadísticos no están disponibles, entonces los métodos de probabilidades no pueden ser usados, los conjuntos fuzzy son una posible solución, donde los expertos pueden especificarlos de una manera directa. En este ejemplo el experto escoge el conjunto fuzzy de la figura 139. Los valores de x menores que a o más grandes que b son considerados como imposible, mientras que los valores que están entre $x - \epsilon$ y $x + \epsilon$ ocurren con certeza. Para el resto de las áreas el experto escoge un incremento monótono y una pendiente negativa de la función respectivamente.

Figura 153. Conjunto Fuzzy que caracteriza el número de rotaciones de un disco duro.



Variable lingüística o variable fuzzy

Cuando se intenta modelar la realidad, a cada fenómeno o hecho se le asocia un nombre de variable en el modelo, para posteriormente ser descrita en términos de su espacio fuzzy, y es a esta variable la que se denomina “Variable Lingüística” por ejemplo: Temperatura, Presión, Edad, Estatura, etc.

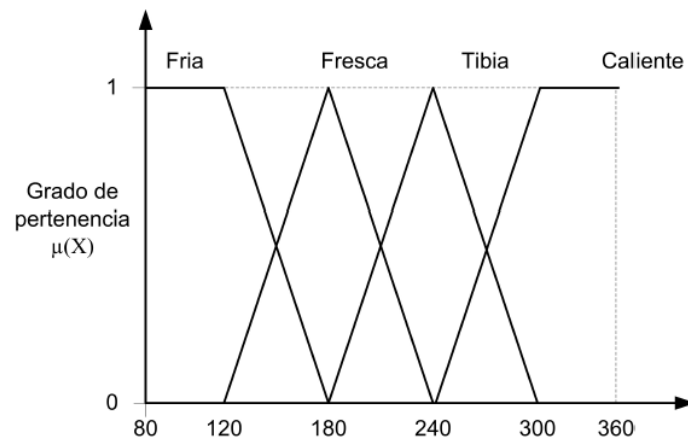
Términos lingüísticos

Si tomamos la variable lingüística “Temperatura”, es evidente que nosotros le asociamos múltiples atributos (categorías, etiquetas) semánticos para describirla en todo su espacio fuzzy; en este caso la temperatura podría ser: caliente, tibia, fresca, fría. Es a cada uno de estos atributos, lo que se denomina “Términos Lingüísticos” y como se mostró anteriormente, cada uno de ellos puede ser representado a través de un conjunto fuzzy.

Universo de discurso

Una variable en un modelo, como anteriormente se mostró, es a menudo descrita en términos

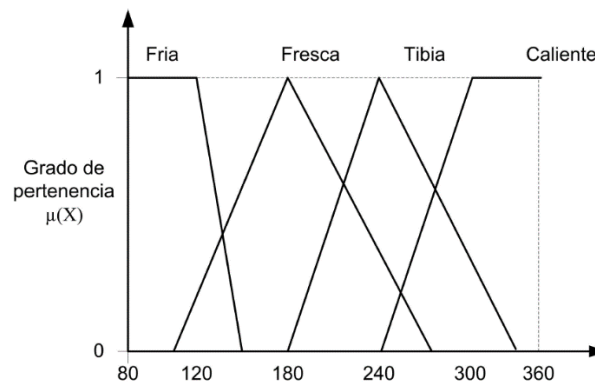
Figura 154. Universo de discurso para la temperatura de una turbina.



de su espacio fuzzy. Este espacio está generalmente compuesto de múltiples conjuntos fuzzy (términos o atributos lingüísticos) traslapados, donde cada conjunto fuzzy describe una partición semántica de la variable.

La figura 140, ilustra este concepto. La variable temperatura esta subdividida en cuatro conjuntos fuzzy: fría, fresca, tibia y caliente. El espacio total del problema, desde el valor permisible más pequeño hasta el valor permisible más grande, se denomina “Universo de discurso o de raciocinio”. El universo de discurso para la variable temperatura (Figura 140), es de 80°C a 360°C, mientras que el dominio del conjunto fuzzy tibia es de 180°C a 300°C. Los conjuntos fuzzy que describen el universo de discurso no necesariamente son simétricos pero ellos siempre deberían traslaparse en algún grado o porcentaje.

Figura 155. Universo de discurso con conjuntos asimétricos.



Representación de los conjuntos fuzzy

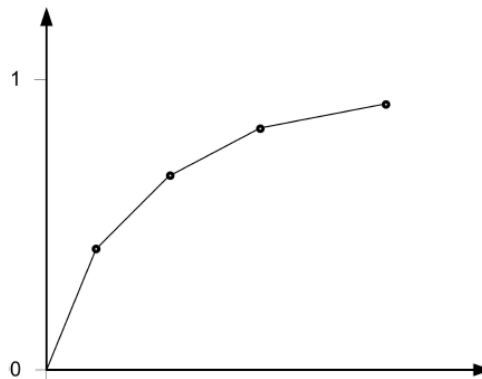
Hay varias posibilidades de representar una función de pertenencia característica de un conjunto fuzzy. Sí el universo X tiene un número finito de elementos, un conjunto fuzzy μ de X será definido por cada elemento x que pertenece a X y su grado de pertenencia $\mu(x)$, sí el número de elementos es muy grande, puede ser mejor definido por una función que use parámetros que se adapten al modelo del problema. Sí $X =$ la expresión lingüística grande puede ser interpretada por una función lineal como:

$$\mu_{a,b}(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x \geq b \end{cases} \quad \text{donde se asume que } a \leq b$$

La función mostrada en la figura 142 es una interpretación alternativa, usando las coordenadas de los puntos definidos como parámetros.

Una función de incremento exponencial

Figura 156. Conjunto Fuzzy definido por puntos.



Fuente: Los autores

$$\mu_{a,b(x)} = \begin{cases} 1 - e^{-a(x-b)} & b \leq x \\ 0 & b > x \end{cases} \text{ con } a > 0 \text{ y } b \in \mathbb{R}$$

Para interpretar expresiones lingüísticas, se usan los llamados números fuzzy, que son simples triángulos simétricos

$$\mu_{m,d(x)} = \begin{cases} 1 - \left| \frac{m-x}{d} \right| & m-d \leq x \leq m+d \\ 0 & x < m-d \vee x > m+d \end{cases} \text{ con } d > 0$$

O la función Gaussiana

$$\mu_{a,b(x)} = \exp(-a(x-b)^2) \text{ con } a > 0, b \in \mathbb{R}$$

La expresión aproximadamente entre a y b puede ser caracterizada por la función trapezoidal

$$\mu_{a,b,c,d}(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x < c \\ \frac{x-d}{c-d} & c \leq x < d \\ 0 & b < a \vee x < d \end{cases}$$

La representación de los conjuntos fuzzy usando la caracterización de la función de pertenencia es llamada representación vertical, considerando la representación gráfica de la función. Con frecuencia un experto define un conjunto fuzzy, especificando para todos los grados de pertenencias α de un subconjunto escogido, esos elementos de X que tienen al menos el grado de pertenencia α . Esta es la representación horizontal de un conjunto fuzzy, usando α -cortes.

Definición: Si $\mu \in F(X)$ y $\alpha \in [0,1]$ el conjunto

$$[\mu]_{\alpha} = \{x \in X \mid \mu(x) \geq \alpha\} \text{ es llamado } \alpha - \text{corte de } \mu.$$

Operaciones sobre conjuntos fuzzy

Los operadores básicos sobre conjuntos como unión, intersección y complemento también pueden ser definidos para conjuntos fuzzy. Es esperado que la intersección

$\mu \cap \mu'$ de dos conjuntos fuzzy $\mu, \mu' \in F(X)$ puede ser calculado elemento por elemento, existe una función $\Pi: [0,1]^2 \rightarrow [0,1]$ tal que $(\mu \cap \mu')(x) = \Pi(\mu(x), \mu'(x))$.

Para aceptar Π como un operador de intersección, tiene que cumplir con algunos axiomas, los cuales manejan la definición de la *t-norma*.

Definición: Una función $T: [0,1]^2 \rightarrow [0,1]$ es llamada *t-norma* si cumple las siguientes condiciones:

$$\begin{aligned} T(a,1) &= a \\ a \leq b &\Rightarrow T(a,c) \leq T(b,c) \\ T(a,b) &= T(b,a) \\ T(a,T(b,c)) &= T(T(a,b),c) \end{aligned}$$

Obviamente T no es decreciente en ambos argumentos y $T(a, 0) = 0$. Las *t-normas* más utilizadas son:

$$\begin{aligned} T_{\min}(a,b) &= \min(a,b) \\ T_{LUKA}(a,b) &= \max(0, a+b-1) \\ T_{prod}(a,b) &= a \cdot b \end{aligned}$$

Definición: Una función $\perp : [0,1]^2 \rightarrow [0,1]$ es llamada *t-conorma*, si \perp es conmutativa, asociativa, no decreciente en ambos argumentos y el 0 es su unidad.

Algunos ejemplos de *t-conormas* son:

$$\begin{aligned} \perp_{\min}(a,b) &= \max(a,b) \\ \perp_{LUKA}(a,b) &= \min(a+b, 1) \\ \perp_{prod}(a,b) &= a + b - ab \end{aligned}$$

Las *t-norma* y las *t-conorma* inducen conexiones para conjuntos fuzzy como:

$$(\mu \cap T \mu')(x) = T(\mu(x), \mu'(x))$$

$$(\mu \cup \perp \mu')(x) = \perp(\mu(x), \mu'(x))$$

Además de las operaciones entre conjuntos teóricos, es posible extender la asignación de la forma $\phi: X \rightarrow Y$ a la asignación: $\hat{\phi}: F(X)^n \rightarrow F(Y)$. La transición de la función ϕ a su extensión $\hat{\phi}$ es llamada principio de extensión.

Definición: Si $\phi: (X)^n \rightarrow Y$ es una asignación. La extensión de ϕ es definida por:

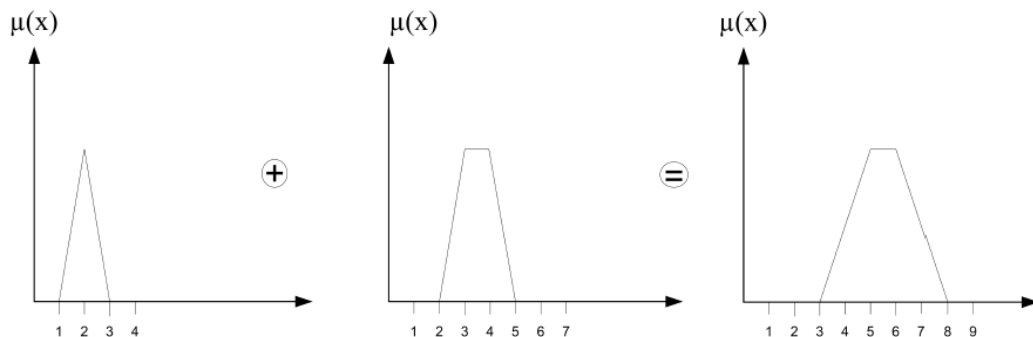
$$\hat{\phi}: F(X)^n \rightarrow F(Y) \text{ con}$$

$$\hat{\phi}(\mu_1, \dots, \mu_n)(y) = \sup \left\{ \min \{ \mu_1(x_1), \dots, \mu_n(x_n) \} \mid (x_1, \dots, x_n) \in X^n \wedge y = \phi(x_1, \dots, x_n) \right\}$$

Con la ayuda del principio de extensión podemos obtener la adición de dos conjuntos fuzzy arbitrarios $\mu, \mu' \in F(\mathbb{R})$:

$$(\mu + \mu')(t) = \sup \left\{ \min \{ \mu(x_1), \mu'(x_2) \} \mid x_1, x_2 \in \mathbb{R} \wedge x_1 + x_2 = t \right\}$$

Figura 157. Adición de conjuntos fuzzy.



Fuente: Los autores

La figura 143 muestra, la adición de dos conjuntos fuzzy, si como en este caso los conjuntos son convexos, es fácil operar sobre ellos. En general las operaciones sobre conjuntos fuzzy son mucho más complicadas, especialmente si se usa la representación vertical. Así de este modo es preferible la representación horizontal, y operar sobre los α - cortes.

Razonamiento fuzzy

A diferencia de los sistemas expertos convencionales, donde las proposiciones (reglas) son ejecutadas en serie, el protocolo principal de razonamiento detrás de la lógica fuzzy es un paradigma de procesamiento paralelo. En los sistemas convencionales basados en conocimientos se aplican algoritmos de poda y /o heurísticos para reducir el número de reglas a examinar, pero en los sistemas fuzzy todas las reglas son disparadas.

El mecanismo raíz en un modelo fuzzy son las proposiciones. Estas son reglas que establecen relaciones entre las variables del modelo y una o más regiones fuzzy.

Una serie de proposiciones fuzzy condicionales e incondicionales son evaluadas a través de su grado de verdad (pertenencia) y todas aquellas que tengan algún grado de verdad contribuyen al estado final de la salida, del conjunto variable solución. El lazo funcional entre el grado de verdad y las regiones fuzzy relacionadas, es llamado el “método de implicación”. El lazo funcional entre las regiones fuzzy y el valor esperado de un conjunto objetivo, es denominado el “método de difusificación”. Tomándolos juntos, ellos se constituyen en el sostén (columna vertebral) del razonamiento aproximado.

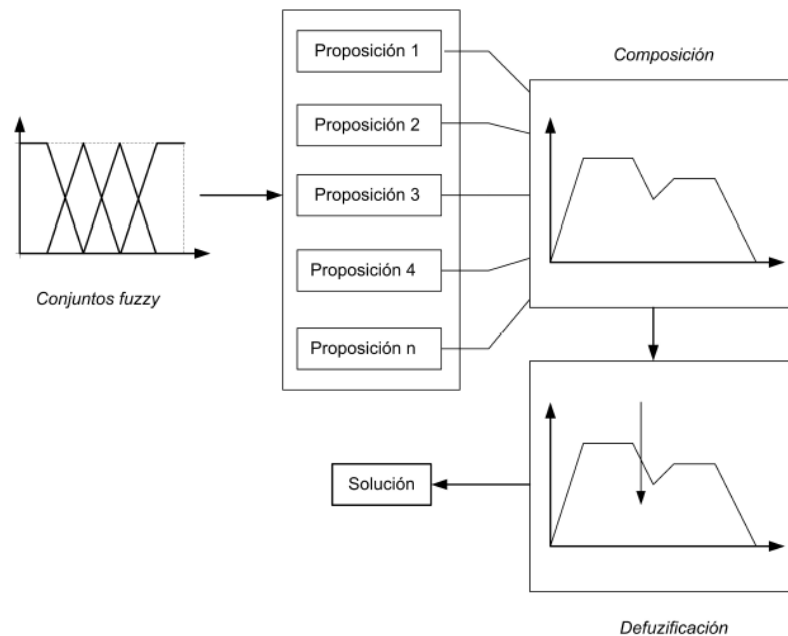
Mientras se están desarrollando soluciones a un método fuzzy, el sistema de razonamiento fuzzy convierte cada variable solución dentro de una región fuzzy provisional, así:

$$z_i \Rightarrow Z_i$$

Donde z es la variable solución (algunas veces llamada la salida) y Z es el correspondiente conjunto fuzzy solución. A medida que cada proposición es evaluada, su consecuente región fuzzy es usada para actualizar la región fuzzy solución. Véase figura 9.

Este proceso de actualización está bajo el control de una función de transferencia (g) que implementa una regla de implicación entre el estado de la región fuzzy consecuente y el estado de la salida fuzzy así:

Figura 158. Combinación y defuzzificación de proposiciones fuzzy



$$g(w_i) \Rightarrow Z_i$$

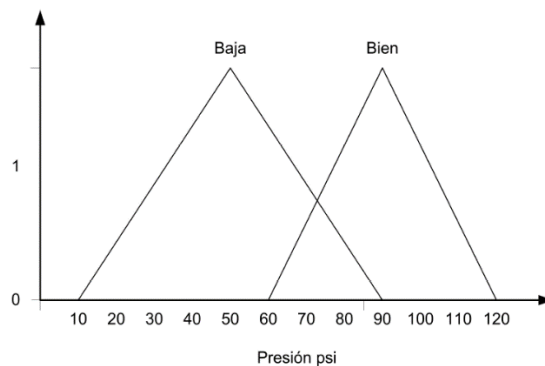
Existen muchas posibles funciones de transferencias para la implicación, pero en últimas cada una intenta correlacionar el sistema semántico del antecedente con la semántica del consecuente, y así generar una solución compatible con el significado del estado fuzzy para cada variable de salida.

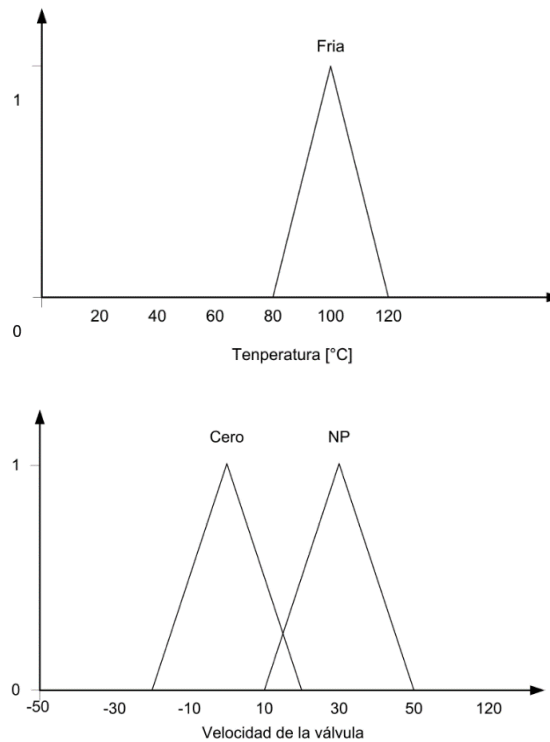
Reglas composicionales fuzzy de inferencia

A diferencia del razonamiento monótono, el espacio de implicación generado por las reglas composicionales de inferencia, se deriva de la agregación y correlación de espacios fuzzy producidos por la interacción de muchas reglas (proposiciones).

En efecto, todas las proposiciones están corriendo en paralelo para crear un espacio de salida que contiene información de todas las proposiciones. Existen dos métodos principales de inferencia en los sistemas fuzzy. Estos métodos difieren en la manera en que actualiza la representación del espacio fuzzy de salida.

Figura 159. Conjuntos fuzzy para variables de entrada y salida





Reglas MIN-MAX de implicación

La operación composicional *Min-Max* deriva su nombre del método que aplica. La región fuzzy consecuente se restringe al mínimo valor de verdad del predicado. La región fuzzy de salida es actualizada tomando el máximo de esos conjuntos fuzzy minimizados.

Reglas aditivas fuzzy

La operación composicional aditiva tiene una ligera diferencia en la forma de actualizar la región fuzzy de la derivada solución (de salida). La región fuzzy consecuente se mantiene reducida por el mínimo valor de verdad del predicado, pero la región fuzzy de salida es actualizada por una regla diferente. Esta es esencialmente una operación de adición limitada aplicada a la región fuzzy de salida. En lugar de tomar el $\max(\mu_A[x_i], \mu_B[x_i])$ en cada punto a lo largo del

conjunto fuzzy de salida, se suman los valores de funciones de pertenencia. La adición está delimitada por [1.0] de forma que el resultado de cualquier adición no pueda exceder el máximo valor de verdad de un conjunto fuzzy.

Ejemplo de implicación fuzzy con reglas composicionales

Para efectos del presente ejemplo se usa un modelo con tres reglas (proposiciones) y dos variables fuzzy de entrada: Presión y temperatura y una variable fuzzy de salida: velocidad-válvula.

Para la variable Presión se manejarán dos términos lingüísticos: *baja* y *bien*. Para la variable temperatura el término lingüístico *fría*. Para la variable velocidad-válvula los términos lingüísticos *cero* y *positivamente-moderado* (*pm*). En la figura 145, se pueden observar los conjuntos fuzzy para los anteriores conceptos. Se busca la región fuzzy de salida, cuando la presión sea de 70 psi y la temperatura 120°C las tres reglas que se aplicaran son:

1. Sí la presión es *baja* y la temperatura es *fría* entonces velocidad-válvula es *pm*.
2. Sí la presión es *bien* entonces velocidad-válvula es *pm*.
3. Sí la presión es *bien* y la temperatura es *fría* entonces la velocidad-válvula es *cero*.

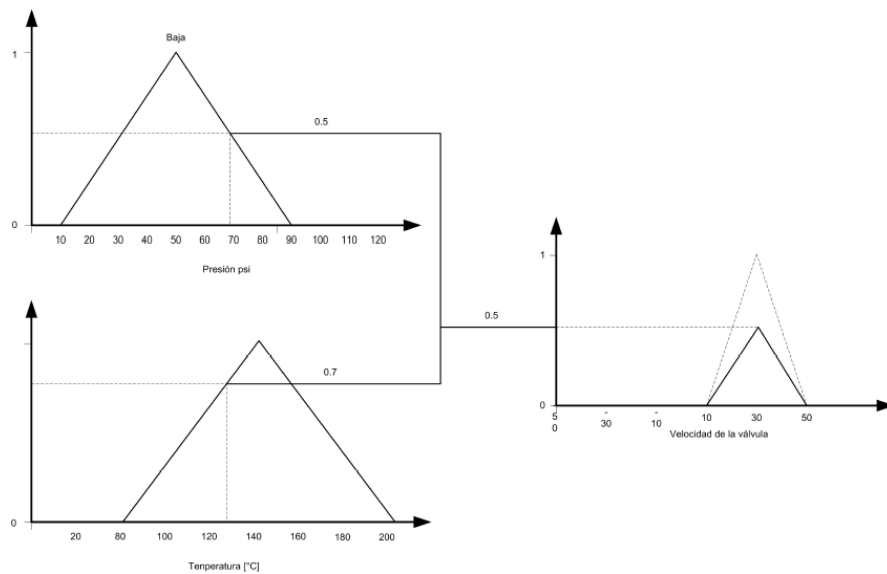
Métodos de correlación

Los mecanismos de implicación *Min/Max* y adición fuzzy utilizados para las proposiciones condicionales fuzzy, involucran una reducción del valor de verdad de la región fuzzy consecuente basada en el valor de verdad de las reglas, antes

que la región fuzzy de la variable solución que se está trabajando, sea actualizada. Existen dos métodos principales para restringir la altura del conjunto fuzzy consecuente: Correlación mínima y Correlación producto.

- Correlación mínima: Es el método más común de correlacionar el consecuente con el valor de verdad de la premisa, truncando (cortado) la región fuzzy a la altura del valor verdad de la premisa. El método se denomina de correlación mínima, puesto que el conjunto fuzzy es minimizado, trucándolo a la altura del valor de verdad mínimo del predicado. Este método de correlaciones fue el que se utilizó anteriormente para desarrollar el ejemplo.

Figura 160. Método de correlación producto.



Usualmente el mecanismo de correlación mínima crea una meseta, dado que la cima de la región fuzzy es cortada a la altura del valor de verdad del predicado.

Lo anterior introduce una cierta cantidad de pérdida de la información. Si el conjunto fuzzy es multimodal, o bien irregular, la topología de la superficie por encima del nivel del valor de verdad, el predicado es descartado. El método de correlación mínima, sin embargo es a menudo preferido sobre el

de correlación producto (el cual preserva la forma de la región fuzzy) puesto que reduce intuitivamente el valor de verdad del consecuente al nivel del mínimo valor del predicado, involucrando menos complejidad y procesamiento aritmético más rápido (una consideración muy importante a nivel de microprocesadores y microcontroladores), y casi siempre genera una superficie agregada de salida que es más fácil de defuzificar usando las técnicas convencionales de centro de momentos (centroide) o centro de máximos.

- Correlación producto: Con este método la región fuzzy intermedia (consecuente) es escalada en vez de truncada, o sea, la función de pertenencia se escala a la altura del valor predicado. Lo anterior tiene el efecto de escoger la región fuzzy mientras retiene aun la forma original del conjunto fuzzy. En la figura 146, se muestra cómo opera el método aplicando a la regla (Min-Max).

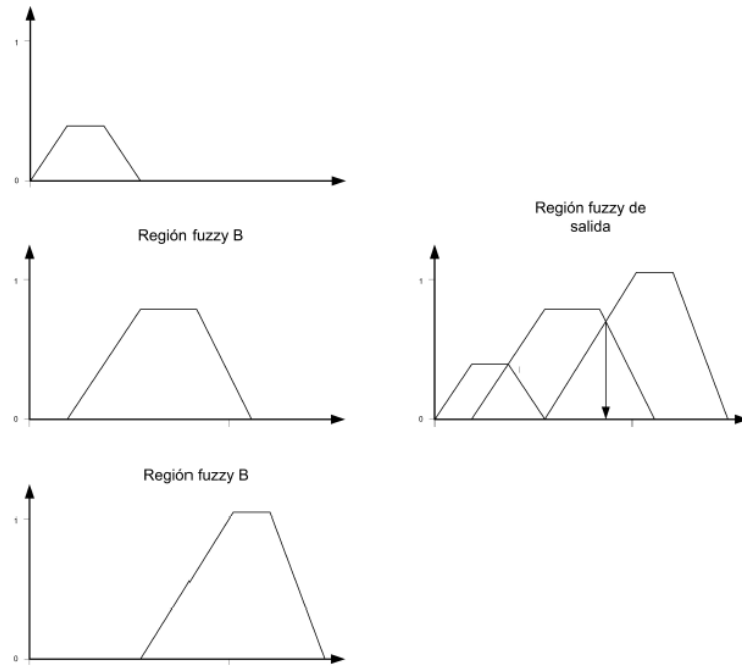
Métodos de descomposición y defuzificación

Usando las reglas generales de una inferencia fuzzy, la evaluación de una proposición produce un conjunto fuzzy con cada variable solución del modelo. Por ejemplo, las siguientes proposiciones, cuando sean evaluadas, correlacionaran los consecuentes conjuntos fuzzy A, B y C para producir el conjunto fuzzy representativo de la variable solución D.

Si w es Y entonces D es A

Si x es X entonces D es B

Figura 161. Proceso de agregación y defuzificación.



Si y es Z entonces D es C

Para encontrar el valor actual para la correspondiente escalar d , se tendrá que encontrar un valor que represente mejor la información contenida en el conjunto fuzzy D . Este proceso, ilustrado en la figura 147, se denomina defuzificación y produce el valor esperado de la variable de salida para una ejecución (estado) particular del modelo fuzzy.

La defuzificación es la fase final del razonamiento fuzzy. Como se ilustra en la figura 147, la evaluación de las proposiciones del modelo es manejada a través de un proceso de agregación que produce las regiones fuzzy finales para cada variable solución. Estas regiones son entonces descompuestas usando uno de los métodos de defuzificación.

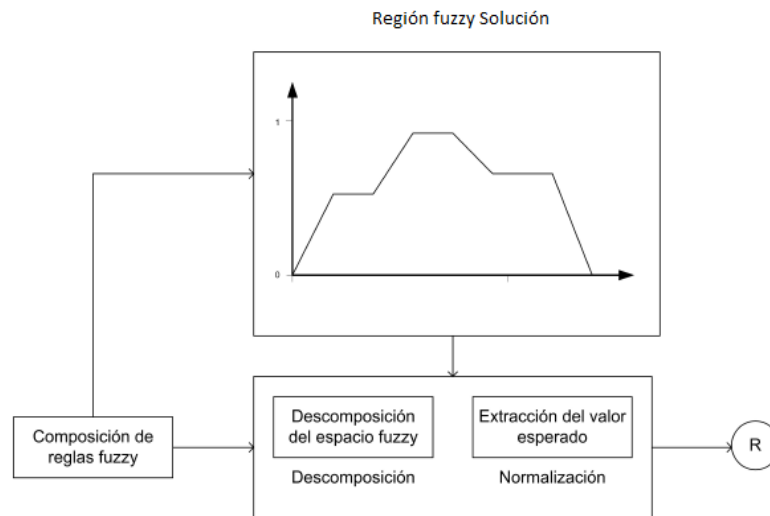
Para los modelos fuzzy, existen varios métodos para determinar el valor esperado de la región fuzzy solución. Estos son los “métodos de descomposición”, denominados también “métodos de defuzificación”, y describen las formas en que

se puede extraer (encontrar) un valor esperado para un estado del espacio fuzzy final.

El entendimiento actual de las reglas de descomposición, son de origen más heurístico que algorítmico, y proviene de los “primeros principios”. Esta es probablemente la mejor manera de poder determinar un valor de la región fuzzy solución, puesto que no parece posible poder representar un espacio multidimensional y complejo, mediante un simple número.

Como se muestra en la figura 149, la defuzificación permite dejar caer una línea recta hacia el dominio. En el punto donde esta línea corta el eje del dominio, se lee el valor esperado del conjunto fuzzy solución, para la variable de salida.

Figura 162. Proceso de descomposición.

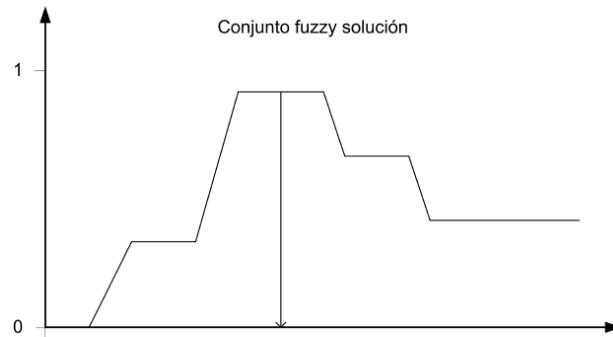


Centro de momento (centroide)

La técnica del centroide, del centro de momento o del centro de gravedad, encuentra el punto de balance de la región fuzzy solución, calculando el peso medio de la región fuzzy. Aritméticamente, para la región fuzzy solución A, esta formulado como:

$$R = \frac{\sum_{i=0}^n d_i \mu_A(d_i)}{\sum_{i=0}^n \mu_A d_i}$$

Figura 163. Defuzificación de un conjunto fuzzy solución.



Donde d es el i -ésimo valor del dominio $\mu(d)$ es el valor de pertenencia para ese punto del dominio. Se puede decir entonces que la defuzificación por el centro de momento o centroide, encuentra un punto que representa el centro de gravedad del conjunto fuzzy.

La técnica del centro de gravedad es la más usada dado que tiene unas propiedades muy importantes:

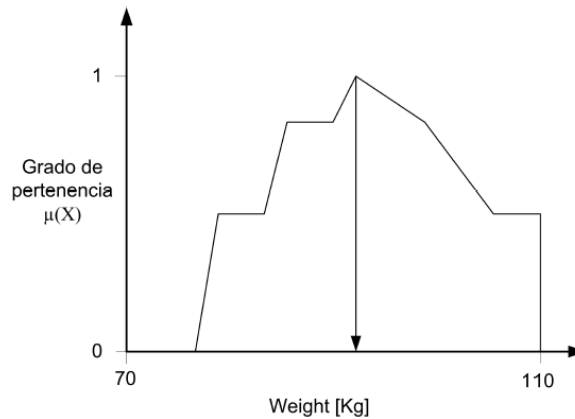
- Los valores defuzificados tienden a moverse suavemente alrededor de la región fuzzy de salida, esto es, cambios en la topología de un conjunto fuzzy de un modelo construido al siguiente, usualmente generan pequeños cambios en el valor esperado.
- Este método es relativamente fácil de calcular.
- Puede aplicarse a conjuntos fuzzy tanto geométricos como de tipo singleton o discretos.

Altura máxima

Existen tres categorías, estrechamente relacionadas con la técnica de la altura máxima: El promedio de máximos, el centro de máximos y la altura máxima simple. A diferencia de la técnica del centroide, la descomposición por altura máxima tiene algunos atributos que son generalmente aplicables a una muy reducida clase de problemas, estos son:

1. El valor esperado es sensitivo a una simple regla que domine el conjunto de reglas fuzzy.
 2. El valor esperado tiende a saltar de un modelo construido al siguiente tanto como la forma de la región cambie.
- Altura máxima simple: Este método (ver figura 150), encuentra el punto del dominio que tiene el máximo valor de verdad. Si este punto es ambiguo (esto es, el máximo es una meseta) entonces se usa otro método asociado para resolver el conflicto: promedio de máximos.
 -

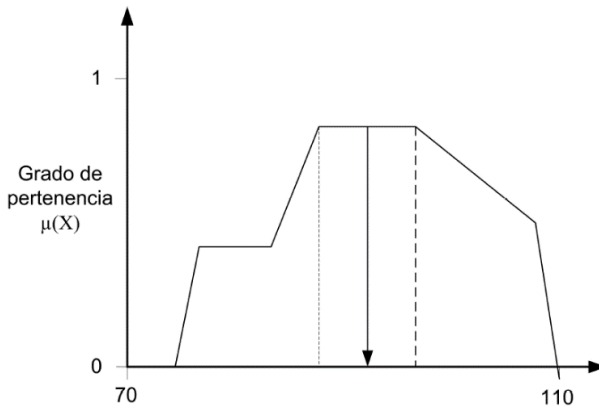
Figura 164. Defuzificación con el método de altura máxima simple.



- Promedio de máximos: Este método (ver figura 151), encuentra la media del valor máximo de la región fuzzy. Si el máximo es un simple punto,

retorna dicho valor; de otra manera el promedio de la meseta es calculado y retornado.

Figura 165. Defuzificación por el método de promedio de máximos.



- Centro de máximos: En una región fuzzy multimeseta, este método encuentra la meseta más alta y la siguiente meseta más alta. El punto medio entre los centros de estas dos mesetas es seleccionado.

Existen otros métodos para descomponer o defuzificar un conjunto fuzzy hacia un valor esperado. Sin embargo, los métodos presentados anteriormente son las formas de defuzificación usadas más ampliamente. En particular, el método de centro de momentos es el preferido en la gran mayoría de los modelos, puesto que este parece asimilar toda la información contenida en el conjunto fuzzy de salida. A menos que se tenga razones para creer que el modelo requiere de métodos de defuzificación más avanzados o especializados.

Figura 166. Defuzificación por método de centros máximos.

