

**DESARROLLO DE UNA HERRAMIENTA SOFTWARE UTILIZANDO REDES NEURONALES
ARTIFICIALES PARA ESTIMAR PROPIEDADES DEL COMPORTAMIENTO MECÁNICO DE ARENAS
LIMPIAS PERTENECIENTES A UNA CUENCA COLOMBIANA**

LAURA VIVIANA GALVIS CARREÑO
CESAR AUGUSTO OCHOA CORONADO



UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2009

**DESARROLLO DE UNA HERRAMIENTA SOFTWARE UTILIZANDO REDES NEURONALES
ARTIFICIALES PARA ESTIMAR PROPIEDADES DEL COMPORTAMIENTO MECÁNICO DE ARENAS
LIMPIAS PERTENECIENTES A UNA CUENCA COLOMBIANA**

**LAURA VIVIANA GALVIS CARREÑO
CESAR AUGUSTO OCHOA CORONADO**

**TRABAJO DE GRADO PARA OPTAR
AL TÍTULO DE INGENIERO DE SISTEMAS**

DIRECTOR:

**Henry Arguello Fuentes
Magíster en Potencia Eléctrica
Profesor, Escuela de Ingeniería de Sistemas, UIS**

CODIRECTOR:

**Jenny Mabel Carvajal
Ingeniera de Petróleos
Responsable del Laboratorio Mecánica de Rocas, ICP**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2009

A Dios, por darme la sabiduría necesaria para recorrer este camino.

A mi mami Ludy, por ser una mujer ejemplo, acompañarme, confiar siempre en mis habilidades y por su amor y cariño brindado.

A mi papi Heriberto, por su esfuerzo que me permitió salir adelante y por estar siempre dispuesto a ayudarme y aconsejarme en todas las circunstancias a las que me he enfrentado en la vida.

A mi hermana Diana, de quien siempre espero lo mejor por su apoyo y amistad incondicional.

A mi novio y amigo Andrés, por acompañarme en este camino y ser una fuente de inspiración.

A toda mi familia por creer en mí y compartir conmigo grandes momentos de mi vida.

A mis amigos y compañeros de carrera por los maravillosos momentos compartidos y por su valiosa amistad.

LAURA VIVIANA GALVIS CARREÑO

A mi madre María Martha por su amor y comprensión, a ella quien siempre me ha brindado su apoyo y confianza, permitiéndome estar más cerca de mis metas profesionales.

A mi abuelo Ismael, mi tío Oscar, todas mis tías y mi hermanita, que siempre han creído en mí y me han apoyando en todo momento.

A mi novia Diana por su dulce y tierno amor, su comprensión y consejos que me hace ser un mejor hombre cada día.

A mis suegros y cuñados que me han acogido y querido como parte de su familia, convirtiéndose en parte de la mía.

CESAR AUGUSTO OCHOA CORONADO

AGRADECIMIENTOS

Los Autores de este proyecto dan su agradecimiento:

Al **GRUPO DE INVESTIGACIÓN ESTABILIDAD DE POZO**, que nos brindó la oportunidad de desempeñarnos en un ambiente investigativo de alta calidad, relacionarnos con estudiantes y profesionales de diferentes áreas y por la experiencia en el campo laboral que nos fortalecieron como personas íntegras.

A **HENRY ARGUELLO FUENTES**, Director de este Proyecto, quien nos acompañó desde el momento en que nació la idea, nos transmitió el conocimiento en el área de desarrollo y por su gran colaboración y oportunos consejos que permitieron desarrollar esta investigación.

A **JENNY MABEL CARVAJAL JIMÉNEZ**, Co-Director, quien nos brindó la oportunidad de trabajar en el laboratorio de Mecánica de Rocas, por su valiosa asesoría en temas totalmente desconocidos y por confiar en nosotros y en nuestras capacidades.

A **NÉSTOR FERNANDO SAAVEDRA TRUJILLO**, Ingeniero de Petróleos y Director del Instituto Colombiano del Petróleo, por haber sido un ejemplo de profesionalismo a seguir.

A todos nuestros compañeros de carrera, del grupo GIIIB y del grupo Estabilidad de Pozo quienes de alguna u otra forma ayudaron a hacer realidad este trabajo.

A la **UNIVERSIDAD INDUSTRIAL DE SANTANDER** por brindarnos los medios necesarios para llegar a ser profesionales.

Al **INSTITUTO COLOMBIANO DEL PETRÓLEO (ICP)**, por facilitar los recursos e instalaciones para el desarrollo de esta Investigación.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
<u>1 MARCO DE REFERENCIA</u>	<u>3</u>
1.1. MARCO TEÓRICO	3
1.1.1. PRUEBAS DE LABORATORIO DE MECÁNICA DE ROCAS	3
1.1.1.1. Propiedades mecánicas a partir de pruebas de laboratorio	4
1.1.1.2. Pruebas de resistencia a la tensión o tracción	6
1.1.1.3. Pruebas de resistencia a la compresión	7
1.1.2. REDES NEURONALES.....	12
1.1.2.1. Redes neuronales y reconocimiento de patrones	13
1.1.2.2. Sistema de reconocimiento de patrones	14
1.2. ESTADO DEL ARTE	17
<u>2 DESARROLLO DEL PROYECTO.....</u>	<u>19</u>
2.1 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	19
2.2 MÓDULO WEB	20
2.2.1 JAVA WEB START	20
2.2.2 DESARROLLO DEL MÓDULO	21
2.2.2.1 Descripción del módulo Web	21
2.2.2.2 Tipos de Usuarios	23
2.2.2.3. Tipos de Permisos	24
2.2.2.4. Diagrama de Casos de Uso	25
2.3. RECOPIACIÓN DE DATOS DE ENTRENAMIENTO	28
2.3.1. TIPOS DE DATOS A UTILIZAR.....	28
2.3.2. FORMA EN LA QUE SE ENCUENTRAN LOS DATOS	28
2.3.3. DESARROLLO DE HERRAMIENTA PARA INTERPRETACIÓN Y ALMACENAMIENTO	29
2.3.3.1. Descripción de la herramienta	29

2.3.3.2. Almacenamiento de la información en la base de datos	38
2.3.3.3. Archivos de configuración local.....	39
2.3.3.4. Análisis de requisitos	40
2.3.3.5. Diagrama de casos de uso.....	40
2.3.3.6. Tabulación de datos	43
2.4. CONFIGURACIÓN DE ARQUITECTURA DE RED UTILIZANDO ALGORITMOS GENÉTICOS	43
2.4.1. DESCRIPCIÓN DEL PROBLEMA DE LA BÚSQUEDA DE LA ARQUITECTURA DE RED.....	43
2.4.2. PLANTEAMIENTO DE LA SOLUCIÓN COMO UN PROBLEMA DE BÚSQUEDA.....	45
2.4.3. FASES DE EVOLUCIÓN DE LA SOLUCIÓN	46
2.4.3.1. Fuerza Bruta	46
2.4.3.2. Fuerza Bruta con condiciones (Heurístico)	47
2.4.3.3. Algoritmo Genético.....	50
2.4.3.3.1. Implementación del primer prototipo	53
2.4.3.3.2. Segundo prototipo	58
2.4.4. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL ALGORITMO GENÉTICO 60	
2.4.5. DESCRIPCIÓN DE LA HERRAMIENTA Y SU ENTORNO GRÁFICO.....	61
2.4.6. MEJORAS	67
2.4.6.1. Entrada de Datos.....	67
2.4.6.2. Manejo de Errores.....	68
2.4.6.3. Conversión de Arquitectura de Red Neuronal a XML	69
2.5. IMPLEMENTACIÓN DE LA ARQUITECTURA DE RED NEURONAL EN JAVA.....	70
2.5.1. ¿POR QUÉ USAR LA RED FUERA DE MATLAB?.....	70
2.5.2. JCORTEX.....	70
2.5.3. EDICIÓN DE JCORTEX	71
2.5.4. ARCHIVO XML	71
2.5.5. FORMA DE CARGAR ARCHIVO XML A LA BASE DE DATOS	71

2.5.6. DESARROLLO DE LA HERRAMIENTA	72
2.5.6.1. Descripción de la herramienta	72
2.5.6.2. Diagrama de Casos de Uso	75
<u>3. RESULTADOS.....</u>	<u>77</u>
3.1. RESULTADOS INTÉRPRETE	77
3.2. RESULTADOS ESTIMADOR.....	79
3.2.1. PRUEBA DE LABORATORIO 1. TENSIÓN INDIRECTA Ó BRAZILIAN TEST 81	
3.2.2. PRUEBA DE LABORATORIO 2. COMPRESIÓN UNIAXIAL	88
3.2.3. PRUEBA DE LABORATORIO 3. COMPRESIÓN TRIAXIAL	94
<u>CONCLUSIONES Y RECOMENDACIONES</u>	<u>103</u>
<u>BIBLIOGRAFÍA</u>	<u>105</u>
<u>ANEXOS</u>	<u>107</u>
<u>ANEXO 1 BASE DE DATOS IMPLEMENTADA</u>	<u>107</u>
<u>ANEXO 2 ARTÍCULO SELECCIONADO PARA PONENCIA TIPO POSTER EN XIII SIMPOSIO DE TRATAMIENTO DE SEÑALES, IMÁGENES Y VISIÓN ARTIFICIAL</u>	<u>113</u>

LISTADO DE FIGURAS

<i>Figura 1 Esfuerzos aplicados al espécimen.</i>	4
<i>Figura 2 Tipos de pruebas en rocas.</i>	5
<i>Figura 3 Prueba brasilera.</i>	7
<i>Figura 4 Prueba de compresión uniaxial.</i>	8
<i>Figura 5 Curva esfuerzo Vs. Deformación en una prueba uniaxial.</i>	10
<i>Figura 6. Dibujo esquemático del laboratorio de mecánica de rocas.</i>	10
<i>Figura 7. Bosquejo del principio de la celda triaxial.</i>	11
<i>Figura 8. Prueba de compresión triaxial aplicada hasta lograr falla.</i>	11
<i>Figura 9. Etapas de un sistema de Reconocimiento de patrones.</i>	15
<i>Figura 10 Ventana de bienvenida al módulo web y características de sesión.</i>	22
<i>Figura 11 Información del proyecto y links de descarga de la aplicación.</i>	22
<i>Figura 12 Gestión de usuarios en la sesión del Administrador y datos petrofísicos.</i>	23
<i>Figura 13 Diagrama de Casos de uso del módulo Web.</i>	26
<i>Figura 14 Splash de inicio del intérprete junto con progreso de carga de datos petrofísicos.</i>	31
<i>Figura 15 Pestaña inicial, selección de tipo de prueba y búsqueda de archivo a procesar.</i>	31
<i>Figura 16 Pestaña inicial, opción leer nuevamente en caso de que la prueba haya sido interrumpida.</i>	32
<i>Figura 17. Pestaña de resultados.</i>	33
<i>Figura 18 Error al guardar resultados pues se requieren los datos petrofísicos.</i>	33
<i>Figura 19 Validación de usuario para guardar resultados en la base de datos.</i>	34
<i>Figura 20 Barra de progreso que muestra que se está subiendo la información.</i>	35
<i>Figura 21 Actualización de datos.</i>	35
<i>Figura 22 Pestaña que solicita la información necesaria para generar el reporte.</i>	36
<i>Figura 23 Ejemplo de reporte generado.</i>	37
<i>Figura 24 Diagrama de casos de uso del módulo que interpreta pruebas de laboratorio.</i>	41

<i>Figura 25 Partes de una Red Neuronal.</i>	44
<i>Figura 26 Mínimos locales de la superficie de error.</i>	47
<i>Figura 27 Error Vs. Número de neuronas.</i>	48
<i>Figura 28 Estructura de un Algoritmo Genético Simple.</i>	53
<i>Figura 29 Ejemplo de Cruce entre 2 genes.</i>	54
<i>Figura 30 Gráfica de error de red configurada para obtener el seno utilizando 100 iteraciones.</i>	57
<i>Figura 31 Gráfica de error de red configurada para obtener el seno utilizando 500 iteraciones.</i>	57
<i>Figura 32 Gráfica de error de red configurada para obtener el seno utilizando 1000 iteraciones.</i>	58
<i>Figura 33 Ventana principal configurador de redes neuronales artificiales.</i>	62
<i>Figura 34 Carga de datos, utilizando detención temprana en el entrenamiento.</i>	62
<i>Figura 35 Parámetros que se ingresan al algoritmo genético.</i>	64
<i>Figura 36 Interfaz en la que el configurador muestra los resultados.</i>	66
<i>Figura 37 Análisis estadístico de la búsqueda realizada por el algoritmo genético.</i>	67
<i>Figura 38 Interfaz modificada que facilita el ingreso de datos.</i>	68
<i>Figura 39 Interfaz modificada que permite guardar configuración como XML.</i>	69
<i>Figura 40 Carga de configuraciones a la base de datos a través del módulo Web.</i>	72
<i>Figura 41 Splash del estimador mientras se cargan datos del servidor.</i>	73
<i>Figura 42 Interfaz del Estimador. Pestaña de selección de red.</i>	73
<i>Figura 43 Interfaz del Estimador. Pestaña de Estimación manual.</i>	74
<i>Figura 44 Interfaz del estimador. Pestaña de estimación desde archivo.</i>	75
<i>Figura 45 Diagrama de Casos de uso del Estimador.</i>	75
<i>Figura 46 Interfaz de carga de datos para la prueba Brazilian Test.</i>	79
<i>Figura 47 Carga de parámetros para la Prueba Brazilian.</i>	81
<i>Figura 48 Configuraciones halladas Prueba Brazilian.</i>	82
<i>Figura 49 Arquitectura de Red Neuronal de la Prueba Brazilian.</i>	83
<i>Figura 50 Gráfica de error de Prueba Brazilian.</i>	83

<i>Figura 51 Estadísticas del configurador para Prueba Brazilian.</i>	<i>84</i>
<i>Figura 52 Arquitectura de Red Neuronal de la Prueba Uniaxial.</i>	<i>89</i>
<i>Figura 53 Gráfica de error de Prueba Uniaxial.</i>	<i>89</i>
<i>Figura 54 Estadísticas del configurador para Prueba Uniaxial.</i>	<i>90</i>
<i>Figura 55 Configuraciones halladas Prueba Triaxial.</i>	<i>95</i>
<i>Figura 56 Arquitectura de Red Neuronal de la Prueba Triaxial.</i>	<i>95</i>
<i>Figura 57 Gráfica de error de Prueba Triaxial.</i>	<i>96</i>
<i>Figura 58 Estadísticas del configurador para Prueba Triaxial.</i>	<i>97</i>
<i>Figura 59 Base de datos implementada.</i>	<i>112</i>

LISTADO DE TABLAS

<i>Tabla 1 Tipos de Usuarios.</i>	<i>24</i>
<i>Tabla 2 Tipos de Permisos.</i>	<i>25</i>
<i>Tabla 3 Descripción de los casos de uso del administrador.</i>	<i>27</i>
<i>Tabla 4 Descripción de los casos de uso de un usuario normal.</i>	<i>27</i>
<i>Tabla 5 Descripción de casos de uso de un usuario normal sobre el intérprete.</i>	<i>42</i>
<i>Tabla 6 Descripción de casos de uso de un usuario con permisos de escritura sobre el intérprete.</i>	<i>43</i>
<i>Tabla 7 Funciones del algortimo genético.</i>	<i>55</i>
<i>Tabla 8 Descripción de casos de uso del Estimador.</i>	<i>76</i>
<i>Tabla 9 Tablas de resultados del intérprete.</i>	<i>78</i>
<i>Tabla 10 Tabla de datos reales y resultados en MATLAB de Prueba Brazilian.</i>	<i>86</i>
<i>Tabla 11 Tabla de resultados de la ejecución de la RNA en MATLAB y en JAVA para la prueba Brazilian.</i>	<i>88</i>
<i>Tabla 12 Tabla de datos reales y resultados en MATLAB de Prueba Uniaxial.</i>	<i>92</i>
<i>Tabla 13 Tabla de resultados de la ejecución de la RNA en MATLAB y en JAVA para la prueba Uniaxial.</i>	<i>94</i>
<i>Tabla 14 Tabla de datos reales y resultados en MATLAB de Prueba Triaxial.</i>	<i>100</i>
<i>Tabla 15 Tabla de resultados de la ejecución de la RNA en MATLAB y en JAVA para la prueba Triaxial.</i>	<i>102</i>

LISTADO DE ANEXOS

<i>Anexo 1 BASE DE DATOS IMPLEMENTADA</i>	<i>107</i>
<i>Anexo 2 ARTÍCULO SELECCIONADO PARA PONENCIA TIPO POSTER EN XIII SIMPOSIO DE TRATAMIENTO DE SEÑALES, IMÁGENES Y VISIÓN ARTIFICIAL</i>	<i>113</i>

LISTADO DE ECUACIONES

<i>Ecuación 1 Resistencia a la tensión.</i>	<i>7</i>
<i>Ecuación 2 Resistencia uniaxial de un cilindro de roca.</i>	<i>8</i>
<i>Ecuación 3 Resistencia a la compresión uniaxial.</i>	<i>8</i>
<i>Ecuación 4 m_j Cantidad de bits necesarios para representar una solución.</i>	<i>53</i>
<i>Ecuación 5 Función de aptitud utilizada.</i>	<i>54</i>
<i>Ecuación 6 Función maximizada inicialmente.</i>	<i>56</i>

RESUMEN

TITULO: DESARROLLO DE UNA HERRAMIENTA SOFTWARE UTILIZANDO REDES NEURONALES ARTIFICIALES PARA ESTIMAR PROPIEDADES DEL COMPORTAMIENTO MECÁNICO DE ARENAS LIMPIAS PERTENECIENTES A UNA CUENCA COLOMBIANA.*

AUTORES: LAURA VIVIANA GALVIS CARREÑO **
CESAR AUGUSTO OCHOA CORONADO **

PALABRAS CLAVE: Red neuronal, Algoritmo genético, JAVA, Configuración, Areniscas, Resistencias, http tunneling.

DESCRIPCIÓN:

En la industria del petróleo, es importante el estudio de las rocas que se extraen de determinada zona para obtener la información necesaria en la toma de decisiones a la hora de realizar trabajos de exploración y explotación. Para encontrar los valores de resistencia a la tensión, UCS (Resistencia a la compresión sin confinamiento) y CCS (Resistencia a la compresión con confinamiento), que indican el comportamiento mecánico de la roca, es necesario realizar una prueba en un laboratorio. Sin llevar a cabo un ensayo sobre una muestra de roca, no se pueden obtener aproximaciones de los valores anteriormente mencionados. En este trabajo se presenta el desarrollo de una herramienta computacional que soluciona el problema de la estimación de los valores de resistencia mediante una arquitectura de red neuronal que utiliza para su entrenamiento datos de petrofísica como datos de entrada y, como datos de salida, valores de resistencia obtenidos de pruebas ya realizadas sobre rocas en laboratorio.

Además, también se presenta un algoritmo genético diseñado para buscar la configuración de la red a utilizar, que realiza una búsqueda y optimización de configuraciones, teniendo como objetivo disminuir el valor de error presente en la red neuronal.

* Proyecto de Grado.

** Facultad de Ingenierías Físico - Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Henry Arguello Fuentes. Codirector: Jenny Mabel Carvajal.

ABSTRACT

TITLE: DEVELOPMENT OF A SOFTWARE TOOL USING ARTIFICIAL NEURAL NETWORKS TO ESTIMATE PROPERTIES OF THE MECHANICAL BEHAVIOUR OF CLEAN SANDSTONES BELONGING TO A COLOMBIAN BASIN.*

AUTHORS: LAURA VIVIANA GALVIS CARREÑO **
CESAR AUGUSTO OCHOA CORONADO **

KEY WORDS: Neural network, Genetic algorithm, JAVA, Configuration, Sandstones, Strength, http tunneling.

DESCRIPTION:

In the petroleum industry, is really important the study of the extracted rocks from certain areas in order to obtain information needed to make decisions about the process of exploration and exploitation. To find the values of tensile strength, UCS (Unconfined Compressive Strength) y CCS (Confined Compressive Strength), which indicates the mechanical behavior of the rock, it is necessary to perform a test in the laboratory. It is not possible to obtain an approximation of the aforementioned data, without performing the test to a sample of rock. In this dissertation is shown the development of a computational tool that solves the problem of estimating the values of strength, using the architecture of an artificial neural network, which uses for its own training, petrophysical information as inputs data and, as outputs, generates values of strength obtained from tests that were already performed in a laboratory.

Moreover, it is presented a genetic algorithm design to find the precise configuration of the network to use; this algorithm executes search and optimization of networks configurations, having as its objective the minimization of the error in the neural network.

* Final Project.

** Physics and Mechanics Engineering College, Systems and Informatics School.
Henry Arguello Fuentes. Jenny Mabel Carvajal.

INTRODUCCIÓN

La industria del petróleo es un área interdisciplinaria en la que diversos perfiles profesionales aportan su experiencia para lograr un mayor fortalecimiento en su actividad principal, la búsqueda, perforación y extracción de hidrocarburos; el ingeniero de sistemas como eje de apoyo en cualquier área de conocimiento participa de ese proceso aportando su conocimiento y habilidad en el manejo de información en las diferentes formas en las que ésta se presenta.

En esta industria, se realizan diferentes tipos de pruebas de laboratorio sobre muestras de rocas que al fallarsen (fracturar) permiten obtener valores como resistencia a la tensión, UCS (*Resistencia a la compresión sin confinamiento*) y CCS (*Resistencia a la compresión con confinamiento*) que indican el comportamiento mecánico de éstas. Después de analizar e interpretar estos resultados se obtiene información vital para la toma de decisiones en trabajos de exploración y explotación.

Sin llevar a cabo un ensayo sobre una muestra de roca, no se pueden obtener aproximaciones de los valores anteriormente mencionados. Para solucionar este problema se planteó entrenar redes neuronales capaces de estimar con cierto grado de acierto los valores de éstas propiedades. El entrenamiento de estas redes se realizó utilizando datos de petrofísica como datos de entrada y como datos de salida, resultados de pruebas ya realizadas en el laboratorio.

Esta información permite a los ingenieros encargados del laboratorio, diseñar ensayos no destructivos con cierto grado de confianza. En estos, se podría evitar el fracturamiento de la muestra, dando lugar a una reducción de costos.

Adicionalmente permite simular y aumentar el volumen de datos de un pozo, brindando más información a los profesionales de la industria del petróleo que trabajan en modelamiento mecánico para estabilidad de pozo y fracturamiento

hidráulico, esto representa modelos más robustos, es decir, más cercanos a la realidad.

Este trabajo se desarrolló con el respaldo del Grupo de Investigación Estabilidad de Pozo (GIEP) de la Escuela de Ingeniería de Petróleos en convenio con el Instituto Colombiano del Petróleo (ICP) y el Grupo de Investigación en Ingeniería Biomédica (GIIB) de la Escuela de Ingeniería de Sistemas e Informática.

1 MARCO DE REFERENCIA

1.1. MARCO TEÓRICO

1.1.1. PRUEBAS DE LABORATORIO DE MECÁNICA DE ROCAS

Cuando un espécimen (muestra o ejemplar) de roca está sometido a esfuerzos grandes, ocurrirá algún tipo de falla¹. El modo de falla dependerá del estado de esfuerzos (orientación y magnitud de esfuerzos relacionados a las superficies orientadas en las tres direcciones ortogonales x,y,z), del tipo de material que constituya la roca así como de su geometría.

Por otra parte también la historia de esfuerzos sobre la roca puede ser importante, si se tiene en cuenta que la roca se debilita con el tiempo. Es por ésta razón, que las rocas fallan usualmente por debajo del nivel de esfuerzos al cual un espécimen fresco del mismo material lo hace².

Las técnicas sugeridas para determinar la deformabilidad, resistencia y las constantes elásticas son dadas por la ISRM (Sociedad Internacional de Mecánica de Rocas). Se han establecido normas en la industria las cuales han sido un marco de referencia para la estandarización de pruebas de laboratorio y procedimientos operacionales en la industria petrolera, manejo y preparación de pruebas, manipulación de corazones, entre otros, en éste caso el Laboratorio de mecánica de rocas se rige por las normas ASTM D2664-04 y D5407-95.

Por ejemplo, una muestra de prueba típica según los parámetros estándar de la ISRM establece que los especímenes dirigidos para pruebas estándar de mecánica de rocas deben ser rectos, cilindros circulares y con una relación de longitud a diámetro entre 2 y 3.

¹ X. Chen, C:P Tan and C.M Haberfield "A Comprehensive Practical Approach For Wellbore Instability Management" SPE 48898. 1998.

² Torres, M.E., Gonzalez, A.J., Last, N.C. "In-Situ Stress State Eastern Cordillera Colombia" SPE 81074 Colombia National University -Schlumberger DCS Bogotá, Colombia National University, BP Exploration 2003.

De acuerdo a esto, en Mecánica de rocas, existen una prueba de resistencia a la tensión (Brazil Test) y dos pruebas de compresión (Uniaxial y Triaxial). Las pruebas de compresión se efectúan sometiendo un núcleo de roca a un esfuerzo único axial (prueba Uniaxial), o a un esfuerzo axial y un esfuerzo radial constante o confinamiento (prueba Triaxial).

En la figura 1 se observa una muestra con una relación longitud – diámetro de 2:1. Un esfuerzo en la dirección axial y otro esfuerzo, de diferente magnitud, es aplicado en la circunferencia del cilindro. Si el esfuerzo radial es cero se tiene una prueba de esfuerzo uniaxial; cuando la prueba es realizada con una presión de confinamiento diferente a cero, se obtiene una prueba triaxial³.

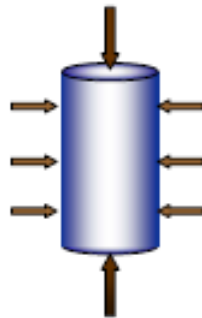


Figura 1 Esfuerzos aplicados al espécimen.

Fuente: GÓMEZ VANEGAS César Augusto y SANTAMARÍA BOLIVAR Rafael. *ANÁLISIS DE ESTABILIDAD DE POZO UTILIZANDO EL SOFTWARE PBORE. Colombia Tesis 2004.*

1.1.1.1. Propiedades mecánicas a partir de pruebas de laboratorio

Un número variado de pruebas de laboratorio se han desarrollado para determinar la resistencia de las formaciones de rocas. El tipo de prueba seleccionada para medir una característica particular de la roca deberá simular lo más cerca posible las condiciones encontradas en el yacimiento.

³ GÓMEZ VANEGAS César Augusto y SANTAMARÍA BOLIVAR Rafael. *ANÁLISIS DE ESTABILIDAD DE POZO UTILIZANDO EL SOFTWARE PBORE. Colombia Tesis 2004.*

Ninguna técnica actual es capaz de medir in-situ la resistencia de una roca. Por lo tanto pruebas de laboratorio estándar basadas en corazones pueden proporcionar esos datos pero no reflejar exactamente las propiedades in situ a gran escala, porque estas son influenciadas por diaclasas (fractura en las rocas), fallas, inhomogeneidades, planos de debilidad y otros factores.

Las pruebas de laboratorio usualmente consisten en experimentos simples apropiados para la naturaleza de la roca en el cual importantes cantidades, a menudo esfuerzos y deformaciones, son determinados.

Los datos de corazones (muestras de rocas) a partir de pruebas de campo y registros son aplicados para evaluar el comportamiento mecánico de la roca in-situ. Es así como los corazones nos proporcionan la posibilidad de las mediciones directas de los parámetros de resistencia y las propiedades elásticas estáticas.

Aunque algunas propiedades del material pueden ser obtenidas a partir de consideraciones básicas de la naturaleza atómica de los materiales, se confían en los experimentos para determinar las propiedades que controlan la deformación y resistencia de la roca. Para estas pruebas de laboratorio (Ver figura 2) es necesario hacer una serie de suposiciones con respecto a la condición ideal de la roca; esto es, que sean homogéneas, continuas, isotrópicas (propiedades no direccionales) y elásticas⁴.

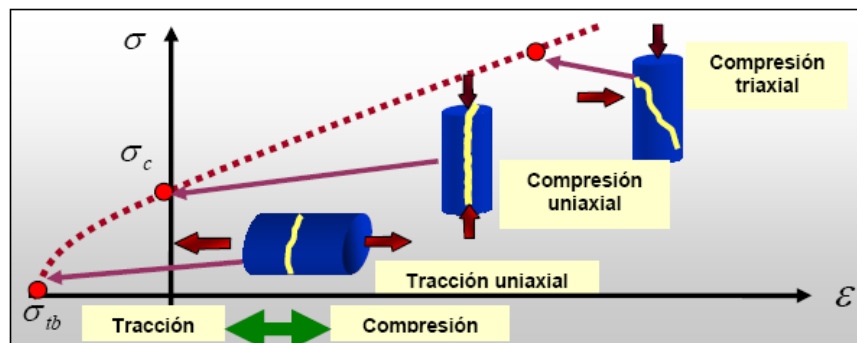


Figura 2 Tipos de pruebas en rocas.

⁴ JAEGER J.C., and COOK N. G. W. Fundamentals of Rock Mechanics, 1969.

Fuente: CABRERA J. R., and DE GENNARO S. *Aplicaciones Prácticas de la Geomecánica en la Industria Petrolera y Gasífera, ICP, Diciembre 2003.*

1.1.1.2. Pruebas de resistencia a la tensión o tracción

Es muy importante conocer la resistencia a la tensión de una roca puesto que la falla a menudo ocurre en este modo. En general, las rocas son mucho más débiles en tensión que en compresión.

A pesar de la importancia de la resistencia a la tensión de la roca, en la práctica y su conexión con las teorías de falla, mediciones directas de la resistencia a la tensión son difíciles y comúnmente no son realizadas, esto debido a los problemas asociados al agarre de la muestra.

La resistencia a la tensión de la roca es más variable y más influenciada por la forma del espécimen que cualquier otra propiedad mecánica de la roca. Pruebas indirectas para determinar la resistencia a la tensión de la roca son más comunes que la prueba de tensión uniaxial, pero sufren del defecto de que ellas involucran esfuerzos no homogéneos.

Prueba brasilera

También llamada prueba de tracción indirecta, Brazil test o de resistencia a la tensión, este ensayo mide el esfuerzo a la tracción de la roca y consiste en la compresión en el punto de soporte de una muestra cilíndrica. Con esta prueba se halla la resistencia a la tensión inducida (T_o). Asumiendo que la roca es homogénea e isotrópica y que la elasticidad lineal se mantiene, la fractura primaria empezará en tensión desde el centro del espécimen⁵ (Ver figura 3).

⁵ CABRERA J. R., and DE GENNARO S. *Aplicaciones Prácticas de la Geomecánica en la Industria Petrolera y Gasífera, ICP, Diciembre 2003.*

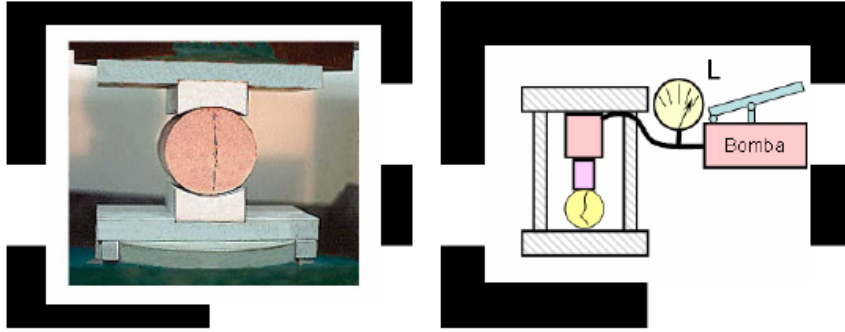


Figura 3 Prueba brasilera.

Fuente: CABRERA J. R., and DE GENNARO S. *Aplicaciones Prácticas de la Geomecánica en la Industria Petrolera y Gasífera, ICP, Diciembre 2003.*

Para realizar la prueba se utiliza una muestra de roca cilíndrica, donde su longitud (L) es igual o más corta a su diámetro (D) y se lleva a cabo aplicando una carga lineal a compresión a través del diámetro. La resistencia a la tensión está dada por la relación de la carga pico (F_c) y el producto del diámetro por la longitud:

$$t_0 = 0.636 \frac{F_c}{DL} \quad \text{Ecuación 1 Resistencia a la tensión.}$$

Donde D y L deben darse en mm.

1.1.1.3. Pruebas de resistencia a la compresión

La respuesta mecánica de las rocas se debe conocer al afrontar un problema en el cual se involucran parámetros de deformación bajo carga. Los ensayos de compresión se desarrollan para dar parámetros de resistencia y deformabilidad que de forma teórica se ajusten a un modelo de falla. En el presente estudio se consideran dos tipos de pruebas para calcular la resistencia a la compresión⁶: uniaxial y triaxial.

Prueba de compresión uniaxial

La prueba consiste en someter a carga un cilindro recto de muestra de roca con una presión de confinamiento igual a cero hasta que esta falle (figura 4). La

⁶ FJAER E., and RASMUS R. *Petroleum Related Rock Mechanics*, 1996.

prueba es una de las más convenientes y útiles para la determinación de las propiedades de la roca. Actualmente se usan cilindros relativamente largos con una relación entre la longitud, L, y el diámetro, D, el cual varía entre 2 a 3. Obert et. al. (1946) encontró que la resistencia uniaxial de los cilindros de roca varía como ⁷:

$$C_0 = C_1 [0.778 + 0.222 D/L] \quad \text{Ecuación 2 Resistencia uniaxial de un cilindro de roca.}$$

Donde C_1 es el valor de la resistencia cuando $D/L=1$. Esto indica que la resistencia decrece con el incremento de la longitud en relación al diámetro. El valor del esfuerzo en el momento de la falla es definido como la UCS (Uniaxial Compressive Strength - resistencia a la compresión uniaxial):

$$C_0 = \frac{F_{max}}{A} \quad \text{Ecuación 3 Resistencia a la compresión uniaxial.}$$

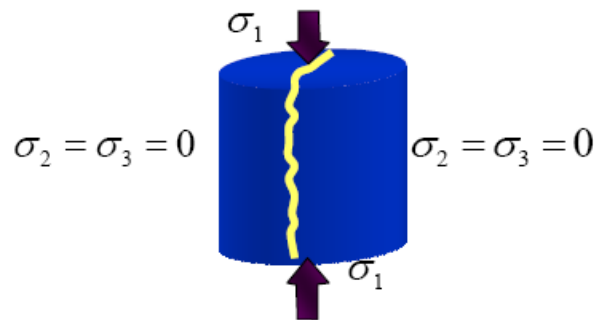


Figura 4 Prueba de compresión uniaxial.

Fuente: CABRERA J. R., and DE GENNARO S. *Aplicaciones Prácticas de la Geomecánica en la Industria Petrolera y Gasífera*, ICP, Diciembre 2003.

Cuando una muestra es comprimida entre las placas del marco de prueba ésta tenderá a expandirse lateralmente (efecto Poisson). Por otro lado, las restricciones friccionales en las placas de contacto tienden a prevenir la expansión; por lo tanto no hay un estado de compresión uniaxial cerca al final de los planos. Otro factor que influye en el valor de la UCS es la rata de carga en la prueba. La UCS de la

⁷ CORZO RUEDA Reinel y RINCÓN PABÓN Claudio Omar. MEDICIÓN Y EVALUACIÓN DE LA MAGNITUD Y DIRECCIÓN DE LOS ESFUERZOS IN-SITU EN CAMPO. Colombia Tesis 2004.

roca generalmente se incrementa al incrementar la rata de carga en el momento de la prueba. Por lo tanto el ISMR en pruebas de laboratorio sugiere que la rata de carga se debe mantener dentro del rango: $0.5 \text{ Mpa/s} < \dot{\epsilon} < 1.0 \text{ Mpa/s}$.

Monitoreando los esfuerzos axiales más las deformaciones axiales y radiales se puede medir:

- La resistencia a la compresión no confinada (UCS – *Unconfined Compressive Strength*).
- Módulo de Young (E).
- Relación de Poisson (ν).
- Curva esfuerzo – deformación.

A continuación se muestra un diagrama de esfuerzos contra deformación en una Prueba de Compresión Uniaxial (Ver figura 5). En la práctica, la región dúctil puede ser muy pequeña. Varios conceptos importantes son definidos en la figura⁸.

- Región elástica: si el esfuerzo es retirado el espécimen retornará al estado original.
- Punto de cedencia: es el punto más allá en el cual permanentes cambios ocurrirán. La muestra no retornará a su estado original bajo alivios de esfuerzos.
- UCS (Resistencia Compresiva Uniaxial): Es el esfuerzo pico.
- Región dúctil: Región en la cual la muestra sufre deformación permanente sin perder la habilidad de soportar carga.
- Región frágil: Región en la cual la habilidad de la muestra de soportar esfuerzos decrece rápidamente mientras la deformación es aumentada.

⁸ CORONEL Iván Leonardo y TARAZONA Darwin Mateus. EVALUACIÓN DE LOS MECANISMOS DE FALLA QUE CONDUCE A LA INESTABILIDAD DE POZO. Colombia Tesis 2004.

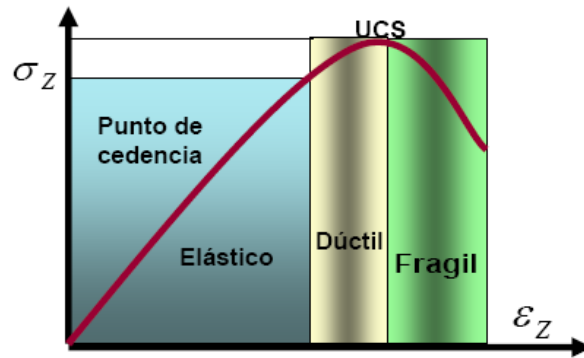


Figura 5 Curva esfuerzo Vs. Deformación en una prueba uniaxial.

Fuente: CORONEL Iván Leonardo y TARAZONA Darwin Mateus. *EVALUACIÓN DE LOS MECANISMOS DE FALLA QUE CONDUCEN A LA INESTABILIDAD DE POZO. Colombia Tesis 2004.*

Prueba de compresión triaxial

La prueba triaxial es usualmente realizada incrementando la carga axial y de confinamiento simultáneamente hasta un nivel de esfuerzo definido (Ver Figuras 6 y 7). Entonces, la presión de confinamiento es mantenida constante mientras la carga axial se va incrementando hasta que la falla ocurra (Ver figura 8)³. La carga axial normalmente es aplicada originando una rata de deformación axial constante.

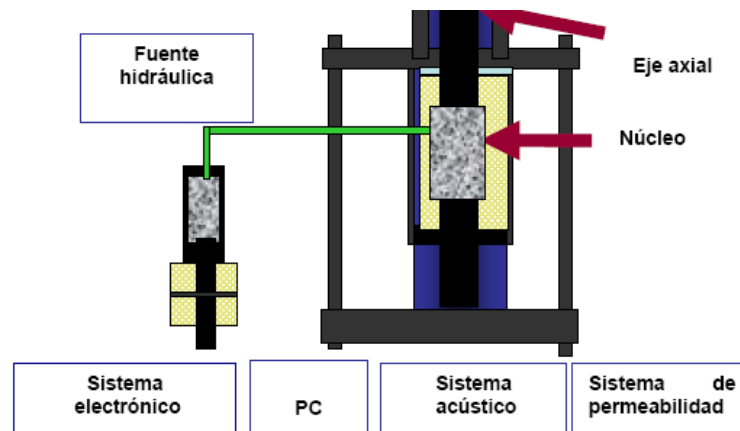


Figura 6. Dibujo esquemático del laboratorio de mecánica de rocas.

Fuente: FJAER E., and RASMUS R. *Petroleum Related Rock Mechanics*, 1996.

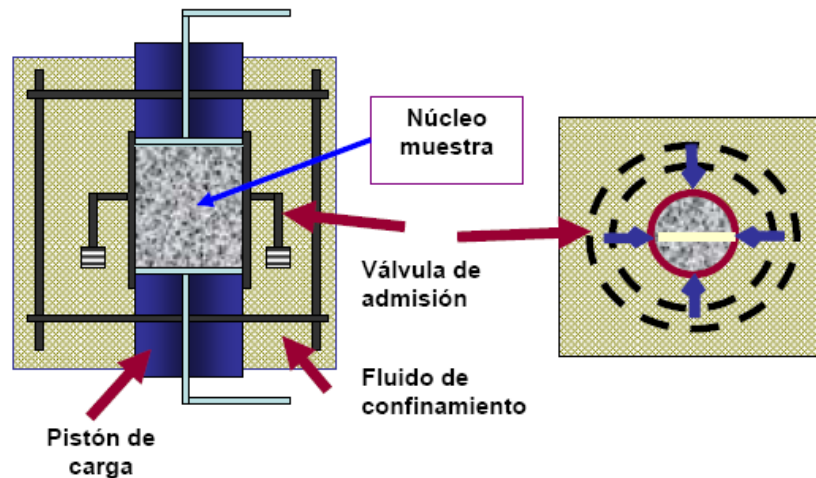


Figura 7. Bosquejo del principio de la celda triaxial.

Fuente: FJAER E., and RASMUS R. *Petroleum Related Rock Mechanics*, 1996.

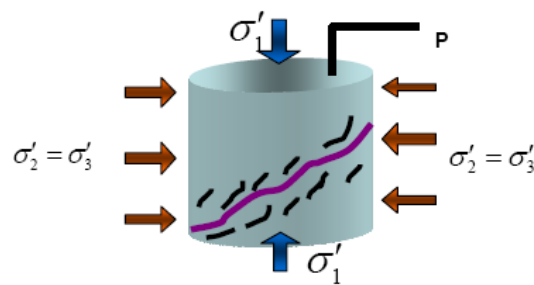


Figura 8. Prueba de compresión triaxial aplicada hasta lograr falla.

Fuente: CABRERA J. R., and DE GENNARO S. *Aplicaciones Prácticas de la Geomecánica en la Industria Petrolera y Gasífera*, ICP, Diciembre 2003.

Este ensayo ha probado ser la prueba más útil en el estudio de las propiedades mecánicas de la roca en un amplio rango de valores para esfuerzos y para diferentes temperaturas, aunque se pueden realizar sin temperatura, además puede hacerse tanto en especímenes secos con presión de poro igual a cero (especímenes no drenados), como en especímenes saturados con una presión de poro (especímenes drenados). Resumiendo se puede decir que una prueba triaxial es la Influencia del esfuerzo efectivo (esfuerzo axial menos la presión de confinamiento) versus curvas de deformación axial.

En general la falla en una prueba triaxial resulta de una fractura de cizalladura simple, plana áspera, inclinada en un ángulo cercano a la dirección del esfuerzo máximo principal, el cual es usualmente paralelo al eje del espécimen. Para caracterizar adecuadamente la muestra la prueba debe ser realizada usando varias presiones de confinamiento. Y, con ella se hallan valores de:

- La resistencia a la compresión confinada (CCS – Confined Compressive Strength).
- Módulo de Young (E).
- Relación de Poisson (ν).
- Curva esfuerzo – deformación.

1.1.2. REDES NEURONALES

Las redes neuronales artificiales nacen principalmente de la necesidad de resolver problemas de la misma manera como lo haría el cerebro humano, debido a esto, se diseñan sistemas que simulen el comportamiento del hombre, es decir que de cierta manera logren adaptarse ante cualquier situación que se les presente, así como normalmente lo hace el humano sin siquiera darse cuenta por ejemplo cuando es capaz de distinguir a personas que lo rodean solo escuchando sus voces.

El reconocimiento de patrones ha sido ampliamente trabajado a través de los años por la estadística, así, se le ha llamado a este estudio, el reconocimiento estadístico de patrones y tal ha sido el esfuerzo por esta disciplina que en el mercado existen actualmente numerosos sistemas de reconocimiento que emplean ésta técnica.

Un patrón es representado por un vector numérico de dimensión n , siendo un punto en un espacio n -dimensional (de características). Y funciona de dos modos diferentes, entrenamiento y reconocimiento, en modo de entrenamiento, se diseña el extractor de características para representar los patrones de entrada y se entrena al clasificador con un conjunto de datos de entrenamiento de forma que el

número de patrones mal identificados se minimice. En el modo de reconocimiento, el clasificador ya entrenado toma como entrada el vector de características de un patrón desconocido y lo asigna a una de las clases o categorías⁹.

Sin embargo, existe otra forma ampliamente conocida, dicha alternativa son las redes neuronales artificiales, que han obtenido un mayor éxito en aplicaciones reales tales como el reconocimiento de patrones, optimizaciones, etc.

1.1.2.1. Redes neuronales y reconocimiento de patrones

El termino de reconocimiento de patrones abarca gran cantidad de problemas y maneja principalmente una vasta área de procesamiento de información, estos problemas son resueltos por los humanos de una manera tan casual que se hace difícil el solo hecho de comprender la complejidad de esta tarea para un computador.

PATRON

Siguiendo la definición de Watanabe¹⁰, un patrón es una entidad a la que se le puede dar un nombre y que está representada por un conjunto de propiedades medidas y las relaciones entre ellas (vector de características). Un patrón entonces puede ser por ejemplo una imagen con de una cara en la que el vector de características estaría formado por valores numéricos calculados a partir de la misma¹⁰.

Gran variedad de disciplinas científicas como la medicina, la biología y la inteligencia artificial tienen entre sus actividades importantes el reconocimiento automático, descripción, clasificación y agrupamiento de patrones, por lo que es un campo de aplicación de muchas expectativas.

⁹ **Dr. Luis Alonso Romero, Dr. Teodoro Calonge Cano.** Capitulo 1. Redes Neuronales y Reconocimiento de patrones. –España. <http://lisisu02.usal.es/~airene/capit1.pfd>

¹⁰ **S. Watanabe.** Pattern Recognition: Human and Mechanical. –Wiley, New York 1985.

1.1.2.2. Sistema de reconocimiento de patrones

Un sistema de reconocimiento de patrones tiene de cualquier manera uno de los siguientes objetivos¹⁰:

1. Identificar el patrón como miembro de una clase ya definida, es decir una clasificación supervisada.
2. Asignar el patrón a una clase no definida, es decir una clasificación no supervisada, agrupamiento o clustering.

En general, se siguen tres fases en el diseño de un sistema de reconocimiento de patrones que son (Ver figura 9):

1. Adquisición y preproceso de datos.
2. Extracción de características.
3. Toma de decisiones, agrupamiento ó clasificación.

La adquisición trata la forma en que se adquieren los datos y las fuentes de los mismos, por ejemplo una máquina de laboratorio, una cámara en el caso de una imagen, etc. Y, el preproceso son todos los procedimientos que permiten organizar y limpiar de cierta manera los datos, desde un análisis de sensibilidad hasta una sencilla tabulación.

En la extracción de características se enumeran las propiedades del vector de características del sistema, esto, mediante un análisis profundo de los patrones para determinar exactamente cuáles son las mediciones cruciales en la identificación de las diferentes categorías bajo investigación. Finalmente, en la etapa de clasificación se determina el tipo de aprendizaje o entrenamiento¹¹.

La figura 9 no debe verse como un modelo de pasos a seguir en la construcción de un sistema de reconocimiento de patrones. Además es importante recalcar que todos los sistemas de reconocimiento no deben incorporar todas las fases y

¹¹ **Leiva Lucas, Acosta Nelson, Vázquez Martín.** Herramienta para diseño de arquitecturas a medida basadas en redes neuronales para reconocimiento de patrones visuales. -Argentina

mucho menos estar claramente separadas, es tan solo un modelo que puede ser visto de forma funcional.

Dependiendo de la tarea para la cual se crea una red neuronal, se define el mecanismo de aprendizaje que es el método sistemático para encontrar un valor adecuado de los pesos que inicialmente son asignados y van evolucionando a unos valores que lleven a encontrar un estado estable de la red, esto puede ser de forma supervisada ó no supervisada.

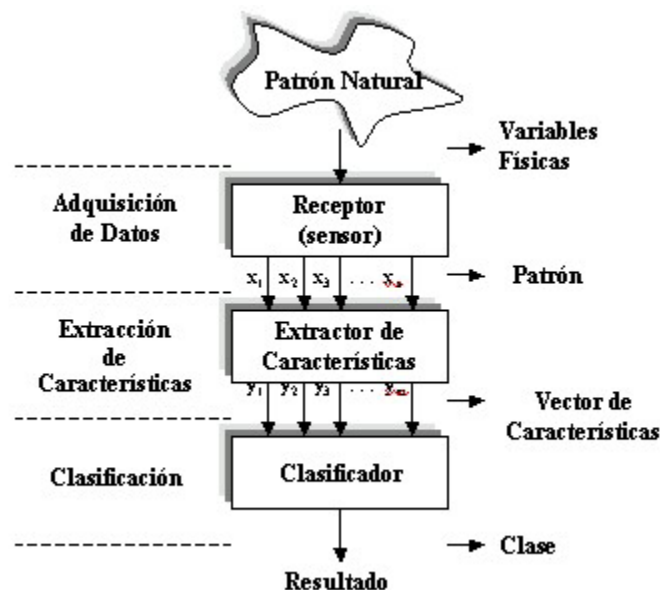


Figura 9. Etapas de un sistema de Reconocimiento de patrones.

Fuente: Leiva Lucas, Acosta Nelson, Vázquez Martín. -Herramienta para diseño de arquitecturas a medida basadas en redes neuronales para reconocimiento de patrones visuales. –Argentina.

Aprendizaje supervisado: se presenta cuando a la red se le proporcionan tanto entradas como salidas correctas con la que se ajustan los pesos tratando de minimizar el error de la salida calculada, este tipo de aprendizaje es el utilizado en problemas que implican reconocimiento de patrones y dado que el problema que

se trata en este proyecto es de este tipo, el aprendizaje supervisado será la manera como éste se maneje.

Aprendizaje no supervisado: por otro lado, este tipo de aprendizaje se da en aquellos casos en los que a la red se le proporcionan únicamente estímulos siendo ella misma la encargada de ajustar sus interconexiones y generar una salida de acuerdo a estos estímulos, es utilizado en problemas de clustering o agrupamiento.

El clustering o agrupamiento a diferencia del reconocimiento de patrones, no presenta etiquetas o salidas prefijadas de antemano para las diversas clases, el sistema extrae las características diferenciadoras por sí solo y por lo tanto fija las fronteras o límites entre las nubes de puntos que se encuentren. No necesariamente estas fronteras se establecen en el espacio de entradas, sino que pueden establecerse en un espacio de características subyacente. Una red diseñada para este tipo de problemas es el mapa autoorganizado de Kohonen.

En cualquiera de los casos, la red aprende mediante los ejemplos, pero la importancia radica en la capacidad de generalización, que es la calidad de una salida ante una entrada que no ha sido utilizada en el entrenamiento, así, entonces se diferencian dos fases de funcionamiento, entrenamiento y reconocimiento; en el entrenamiento se fijan los pesos y en el reconocimiento se procesan entradas correspondientes a la aplicación real.

Es importante entender una característica de las redes neuronales artificiales que las hace especiales y es que debido a que estas crean sus propias reglas (lo que hace que no dependan de un programador que les indique que hacer), aprenden con la experiencia y se castigan por así decirlo cuando cometen errores, no dan precisión en sus resultados, así lo que ellas arrojan son soluciones aproximadas al problema.

La capacidad que tienen las redes neuronales en identificar patrones de comportamientos, generalmente no lineales está dada por la existencia de una

“función de transferencia” que es una función matemática que puede presentar características lineales como no lineales.

1.2. ESTADO DEL ARTE

En la Universidad Hindú Banaras se realizó una investigación que mediante redes neuronales artificiales logró estimar propiedades geomecánicas, presentando como propuesta técnica un algoritmo en C¹².

Las rocas tenidas en cuenta para el estudio se obtuvieron de una compleja y tectónicamente activa secuencia de rocas en la región del Himalaya. Estas rocas pertenecen a la Fundación de Nathpa-Jhakri, un proyecto hidroeléctrico en el distrito de Kinnaur, Himachal Pradesh, India.

Las variables utilizadas como entradas a la red neuronal fueron 11 y se mencionan a continuación:

- Tipo de roca (*Quartzite, Chlorite, Quartz mica y Biotite*)
- Porcentaje de cuarzo
- Porcentaje de mica
- Porcentaje de feldespato
- Porcentaje de Clorita
- Porcentaje de clay
- Tamaño de grano
- Área de ponderación
- Relación de aspecto
- Factor de forma
- Orientación

Las variables de salida de la red neuronal fueron 3 a mencionar:

¹² V.K. Singh, D. Singh, T.N. Singh. 'Prediction of strength properties of some schistose rocks from petrographic properties using artificial neural networks'. Department of Mining Engineering, Institute of Technology, Banaras Hindu University, Varanasi, 221005 India. Department of Electrical Engineering, Institute of Technology, Banaras Hindu University, Varanasi, 221005 India. 28 November 2000

- Resistencia a la Compresión Uniaxial (UCS)
- Resistencia a la Tensión
- Carga axial

En cuanto a la arquitectura de la red neuronal, se utilizó una red perceptrón multicapa, que se argumenta adecuada por su capacidad de resolver problemas basados en patrones, tendiendo hacia la predicción.

En cuanto al diseño de la red, se tomó un set de 140 datos experimentales como ejemplos, de los cuales, 112 se tomaron para realizar el entrenamiento y 28 para pruebas.

La arquitectura finalmente utilizada consistió en 11 nodos de entrada y 3 nodos de salida, en cuanto a las neuronas de la capa oculta, se establecieron realizando pruebas de entrenamiento con diferentes números de neuronas y se encontró que diferentes configuraciones eran posibles, pero en definitiva se decidió una configuración con 10 neuronas en la capa oculta.

Para la validación de los resultados de predicción dados por el modelo de red neuronal artificial, se llevaron a cabo predicciones estadísticas utilizando el mismo conjunto de datos que se tomó para el entrenamiento y pruebas de la red neuronal, resultando éstos más precisos que los encontrados mediante modelos de estadística convencional, encontrando además que la red neuronal es capaz de generalizar mucho más que el modelo estadístico utilizado.

2 DESARROLLO DEL PROYECTO

2.1 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Para estimar propiedades mecánicas a partir de propiedades petrofísicas en arenas limpias, primero se debían cumplir algunos requisitos, por lo que se dividió el problema en tres partes:

- *Adquisición de la información necesaria para realizar los entrenamientos de las redes neuronales:* Esta parte se abordó desarrollando un software de interpretación de los registros obtenidos en el sistema MTS815, agilizando el proceso de interpretación y asistiendo en la elaboración de los informes de resultados en el laboratorio de mecánica de rocas del ICP. Esta información es almacenada en una base de datos centralizada para su posterior consulta. Este intérprete de datos fue desarrollado en el lenguaje de programación JAVA.
- *Búsqueda de las configuraciones más apropiadas para resolver este problema:* Se participó en el proceso de desarrollo del software de configuración automática de redes neuronales *Wötan Genetics*¹³, programado en MATLAB, el cual fue adaptado y utilizado en este proyecto para encontrar las configuraciones de red neuronal que se utilizan para la estimación de propiedades mecánicas. Los entrenamientos fueron realizados con la información almacenada en la base de datos por el intérprete de datos mencionado en el punto anterior.
- *Uso de estas redes en un sistema con un mínimo de restricciones de uso:* Se exportaron las configuraciones de red neuronal encontradas con el software *Wötan Genetics*, al framework Jcortex mediante un archivo de extensión XML, este framework se importó y utilizó en la herramienta que finalmente permite utilizar las redes neuronales para obtener estimaciones

¹³ MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.

de propiedades mecánicas. El software anteriormente mencionado se programó utilizando el lenguaje de programación JAVA.

Finalmente la información del proyecto fue publicada en un sistema web, y además brinda una interfaz de acceso a la base de datos para cumplir algunas tareas administrativas y control de acceso a la aplicación.

Los detalles de cada uno de los sistemas serán descritos en los siguientes puntos de este libro.

2.2 MÓDULO WEB

2.2.1 JAVA WEB START

Es un software ó implementación desarrollada por *Sun Microsystems* que hace referencia a la especificación JNLP (*Java Networking Launching Protocol*), que permite ejecutar aplicaciones Java ubicadas en un servidor web, comprobando previamente si el cliente tiene la última versión de dicha aplicación, de no ser así, se descargará la última versión para luego ejecutarla en modo local.

Este software permite:

- Ejecutar las aplicaciones de forma intuitiva y sencilla.
- Garantiza que se está ejecutando la última versión de la aplicación.
- Elimina complejos procedimientos de instalación ó actualización.

El proceso que se sigue con ésta herramienta es el siguiente:

Cada vez que se realice alguna actualización al software, esta última versión se sube al servidor en el cual se almacena la aplicación.

Luego, cuando el usuario necesita hacer uso por primera vez de la aplicación, éste accede a una página web, en donde obtiene el archivo de extensión JNLP; este archivo se comportará como el ejecutable de la aplicación, al ejecutarlo,

automáticamente el software *Java Web Start* se encarga de acceder al servidor y descargar e instalar la aplicación en la máquina del usuario. Las siguientes ejecuciones serán instantáneas pues solo se hará uso de los recursos disponibles localmente.

Posteriormente, cada vez que se inicie la aplicación (se ejecute el archivo JNLP), el software de *Java Web Start* comprueba si en el servidor donde se encuentra la aplicación hay una nueva versión disponible y, de ser así la descarga y la ejecuta automáticamente.

Para usuarios que posean conexiones lentas la instalación y actualización utilizando la tecnología *Java Web Start* puede presentar inconvenientes, por esto se provee un método de instalación local que podrá ser descargado del módulo web o transportado en cualquier medio de almacenamiento.

2.2.2 DESARROLLO DEL MÓDULO

2.2.2.1 Descripción del módulo Web

El módulo web ha sido desarrollado con el objetivo de brindar un espacio para administrar cada una de las herramientas implementadas y los usuarios que tienen acceso a cada una de ellas (Ver figura 10).

Se tiene un inicio de sesión para que accedan por este a los servicios autorizados a los diferentes tipos de usuarios, cada sesión cuenta con las características de cualquier sesión tales como el manejo del perfil de usuario y cambio de contraseña (Ver figura 10).

Después de iniciada la sesión se encuentran además de toda la información concerniente al proyecto, los links de descarga de cada uno de los módulos para que los usuarios autorizados puedan acceder desde allí (Ver figura 11), en el caso del administrador, se encuentran las diferentes opciones que le permiten gestionar

los usuarios con acceso al módulo y los permisos que cada uno de estos pueden tener (Ver figura 12).

Otra función que se puede realizar en el módulo sólo por el administrador es el de crear datos petrofísicos que después de ser creados, serán descargados por el intérprete y al guardar información de alguna prueba se guardará la respectiva información petrofísica completa en la base de datos (Ver figura 12).



Figura 10 Ventana de bienvenida al módulo web y características de sesión.

Fuente: Autores del proyecto.



Figura 11 Información del proyecto y links de descarga de la aplicación.

Fuente: Autores del proyecto.



Figura 12 Gestión de usuarios en la sesión del Administrador y datos petrofísicos.

Fuente: Autores del proyecto.

2.2.2.2 Tipos de Usuarios

En cuanto al tipo de usuarios, se han definido los siguientes usuarios que son tenidos en cuenta tanto para el módulo web como para los demás módulos.

Usuarios	Descripción
Administrador	<p>Profesional del laboratorio Mecánica de Rocas del ICP capacitado en el uso de la herramienta interesado en la interpretación de pruebas de laboratorio y estimación de valores de resistencia con acceso a los diferentes módulos que conforman la herramienta final. Encargado a través del módulo web de la creación, modificación y eliminación de cuentas de usuarios que tienen acceso a la descarga y utilización de la herramienta.</p> <p>Encargado de otorgar permisos de escritura, descarga y carga de redes a los usuarios creados.</p> <p>Usuario con todos los permisos disponibles, es decir de Lectura/Escritura, de descarga tanto del intérprete</p>

	como del configurador y de cargar las redes obtenidas mediante el configurador.
Usuario con permiso de Lectura	Profesional del laboratorio Mecánica de Rocas del ICP, funcionario ó estudiante que requiera por fines investigativos realizar interpretaciones, generar reportes de pruebas, acceder al historial de pruebas ó estimar propiedades geomecánicas a partir de datos petrofísicos. Sólo tiene acceso a la descarga de la herramienta si el administrador lo ha autorizado creando un usuario a través del módulo web.
Usuario con permisos de Lectura/Escritura	Profesional del laboratorio Mecánica de Rocas del ICP, funcionario ó estudiante que requiera por fines investigativos realizar las mismas acciones que un usuario con permisos de sólo lectura, además de poder guardar en la base de datos ubicada en el servidor, los resultados obtenidos de la interpretación, la información de los informes generados, los valores de resistencia estimados y las configuraciones de red neuronal encontradas a través del algoritmo genético.

Tabla 1 Tipos de Usuarios.

Fuente: Autores del proyecto.

2.2.2.3. Tipos de Permisos

A los usuarios se les pueden asignar diferentes permisos de acuerdo a lo que crea conveniente el administrador. La siguiente tabla presenta estos permisos:

Permisos	Descripción
Sólo lectura	Cuando un usuario es creado por el administrador, con

	<p>su cuenta tiene acceso a la descarga de la aplicación y, por lo tanto a las funcionalidades que sólo necesitan lectura de la base de datos, tales como interpretar una prueba, generar un reporte, buscar en el historial una prueba y estimar propiedades geomecánicas.</p> <p>Es el permiso que tiene un usuario por defecto al ser creado por el administrador.</p>
Lectura/Escritura	<p>La escritura es un permiso que otorga el administrador a un usuario que puede escribir en la base de datos, es decir que puede guardar interpretaciones realizadas, tanto resultados como información de informes.</p>
Cargar configuraciones	<p>El permiso de cargar configuraciones permite al usuario tanto la descarga de los datos existentes hasta el momento en la base de datos como la subida del archivo XML creado después de generar una nueva configuración en el Configurador de MATLAB respectivamente.</p>

Tabla 2 Tipos de Permisos.

Fuente: Autores del proyecto.

2.2.2.4. Diagrama de Casos de Uso

Al módulo Web pueden ingresar todos los tipos de usuarios definidos anteriormente, las siguientes dos tablas describen los casos de uso de los actores que en él participan, es decir, el administrador y un usuario normal.

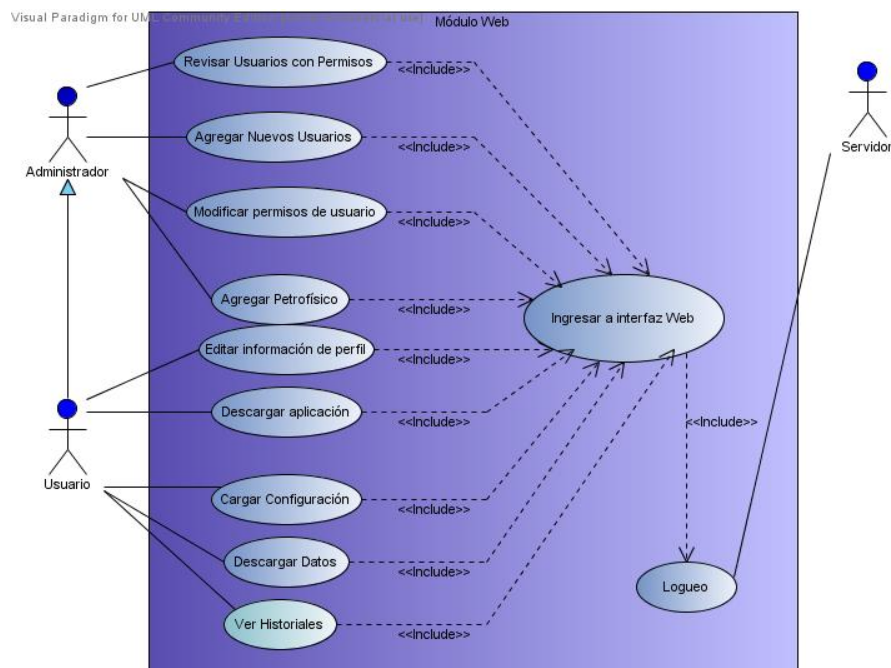


Figura 13 Diagrama de Casos de uso del módulo Web.

Fuente: Autores del proyecto (Realizado en Visual Paradigm for UML Edición community).

Caso de Uso del administrador	Descripción
Agregar nuevos usuarios	El módulo web permite al administrador el registro de cuentas de usuarios para lo cual se debe proporcionar al sistema un correo electrónico y los permisos de usuario que el administrador considere necesarios.
Revisar usuarios con permisos	El administrador puede a través del módulo web hacer una revisión de los usuarios existentes y los permisos que tienen cada uno de ellos.
Modificar permisos de usuario	Si el administrador cree conveniente modificar los permisos de usuario, ya sea porque considera necesario otorgar permisos de escritura que no se tenían anteriormente ó que no es necesario que

	cierto usuario tenga todos los permisos, mediante el módulo se pueden hacer estas modificaciones, agregando o quitando según sea el caso, de esta manera no es necesario eliminar usuarios y volverlos a crearlos con nuevos permisos.
Agregar Petrofísico	Si se hace necesario almacenar un dato petrofísico más de los que se encuentran actualmente, el administrador puede hacerlo desde el módulo.

Tabla 3 Descripción de los casos de uso del administrador.

***Fuente:** Autores del proyecto.*

Caso de Uso de usuarios normales	Descripción
Editar información de perfil	El módulo web permite a un usuario con una cuenta creada por el administrador editar información de perfil haciendo de ésta una cuenta personalizable.
Descargar aplicación	Si el administrador ha creado la cuenta de usuario, éste puede acceder al módulo web, teniendo acceso a los links de descarga de la aplicación.
Ver historiales	Si el usuario desea ver información sobre pruebas interpretadas que han sido guardadas en la base de datos, puede acceder al historial de pruebas y proporcionar el criterio de búsqueda, así tendrá acceso a los resultados de esa prueba.
Cargar Configuración y Descargar Datos	Si el usuario tiene los permisos para cargar configuraciones, puede hacerlo desde el módulo por medio de un formulario amigable e igualmente para el usuario que tiene el permiso de descargar datos.

Tabla 4 Descripción de los casos de uso de un usuario normal.

***Fuente:** Autores del proyecto.*

2.3. RECOPIACIÓN DE DATOS DE ENTRENAMIENTO

2.3.1. TIPOS DE DATOS A UTILIZAR

Los datos a utilizar para la estimación han sido definidos desde el inicio del proyecto con la colaboración de la persona encargada del laboratorio de Mecánica de Rocas del ICP.

Como datos de entrada se tienen propiedades petrofísicas de las rocas y como datos de salida valores de resistencia a la tensión, resistencia a la compresión sin confinamiento y resistencia a la compresión con confinamiento respectivamente según el tipo de prueba sobre la que se quiera realizar la estimación.

2.3.2. FORMA EN LA QUE SE ENCUENTRAN LOS DATOS

En el laboratorio no se cuenta con una base de datos digital en la que se almacene la información completa sobre pruebas realizadas, se cuenta con un libro de registros en el que se lleva cierta información como la fecha de prueba, el nombre de la muestra, la prueba realizada y algunas observaciones, pero para el proyecto es necesario tener los resultados numéricos o en su defecto el archivo que presenta la MTS 815.02 para realizar la interpretación y obtener esos resultados. Es por esto que teniendo como guía el nombre de las muestras a las que se les han realizado pruebas y que han sido registradas en el laboratorio, se realizó la búsqueda de los archivos .dat, interpretaciones y datos petrofísicos en el backup que se lleva desde el año 2003, estos se encuentran almacenados en aproximadamente 100 cd's y dvd's por lo que tomó aproximadamente mes y medio su tabulación.

Al tabular los datos se observó que la gran mayoría de pruebas se encontraban sin ser interpretadas y teniendo en cuenta que cada interpretación toma varias horas, se pensó en desarrollar una herramienta software que automatizara y

agilizara este procedimiento y al mismo tiempo se encargara de ir almacenando ordenadamente la información en una base de datos (Ver anexo A).

2.3.3. DESARROLLO DE HERRAMIENTA PARA INTERPRETACIÓN Y ALMACENAMIENTO

El primer módulo que interpreta los archivos que arroja la MTS 815.02 se desarrolló como medio para obtener resultados de pruebas de forma automática ya que éste es un trabajo realizado actualmente por los ingenieros a cargo del laboratorio de Mecánica de Rocas de forma manual y, teniendo en cuenta la forma en la que se encontraban los datos necesarios para la estimación de valores de resistencia mediante redes neuronales, se hacía necesario una aplicación que encontrara esos resultados de forma rápida evitando ese trabajo manual y ahorrando tiempo.

El intérprete de datos, cumple una labor muy importante para el futuro del proyecto ya que almacena constantemente la información generada en el laboratorio de mecánica de rocas, esto permitirá que después de un tiempo el volumen de los datos haya aumentado así que podrán ser utilizados para configurar nuevas redes neuronales, mejorando los modelos que se hayan generado en el proyecto. El intérprete no es un simple formulario para el almacenamiento de la información en la base de datos; el intérprete presta el servicio de interpretación de generación de informes y como efecto colateral del uso de estos servicios se almacena la información en la base de datos, estos servicios estimulan al usuario a hacer uso de la herramienta.

2.3.3.1. Descripción de la herramienta

En el laboratorio de Mecánica de Rocas del ICP se realizan diferentes tipos de pruebas sobre muestras de rocas, en este proyecto se trabaja sobre tres pruebas específicamente, la prueba de tensión indirecta o Brazilian test, la prueba de compresión uniaxial y la prueba de compresión triaxial.

Después de haber realizado a una muestra cualquiera de las tres pruebas anteriores, la celda triaxial modelo 815 que es el equipo utilizado, presenta al usuario un archivo con extensión dat en el que se muestran dependiendo de la prueba las mediciones por ejemplo de tiempo, desplazamiento axial y fuerza axial tomadas durante todo el tiempo de desarrollo de la prueba. Sobre este archivo la persona encargada del laboratorio realiza el respectivo análisis e interpretación y obtiene resultados, es en esta parte en la que entra en funcionamiento la herramienta, pues permite realizar esta interpretación de forma automática, el procedimiento es el siguiente:

Al abrir el intérprete, mientras es mostrado un splash (imagen inicial que presenta la herramienta), se cargan de la base de datos localizada en el servidor los tipos de datos petrofísicos existentes con el fin de crear de forma dinámica estos campos dentro de la herramienta (junto con el splash es mostrada una barra que muestra el progreso de este proceso) (Ver Figura 14), luego se muestra la pestaña inicial en donde se selecciona el tipo de prueba a analizar entre las tres existentes, tensión indirecta, compresión uniaxial y compresión triaxial (Ver Figura 15), en seguida se selecciona el archivo .dat que contiene la información obtenida durante la prueba y la ventana presenta al usuario datos que sirven para verificar que se hayan tomado los valores acertados (esto debido a que algunas pruebas son interrumpidas y la información queda registrada en el mismo archivo) tales como el dato inicial de tiempo y el intervalo en segundos entre un registro y el siguiente, de ser modificado alguno de éstos dos valores, se presenta un botón con el que se lee nuevamente la información del archivo teniendo en cuenta los cambios realizados.

Igualmente algunas pruebas dentro del archivo presentan diferentes valores de longitud y diámetro por ejemplo para el caso de una prueba de tensión indirecta; con el fin de que se tomen los valores reales ya que hay operaciones matemáticas que los utilizan, se muestran los pares encontrados y así el usuario puede

seleccionar de la lista desplegable los utilizados realmente sobre la muestra (Ver Figura 16).



Figura 14 Splash de inicio del intérprete junto con progreso de carga de datos petrofísicos.

Fuente: Adaptado del Laboratorio Mecánica de rocas.

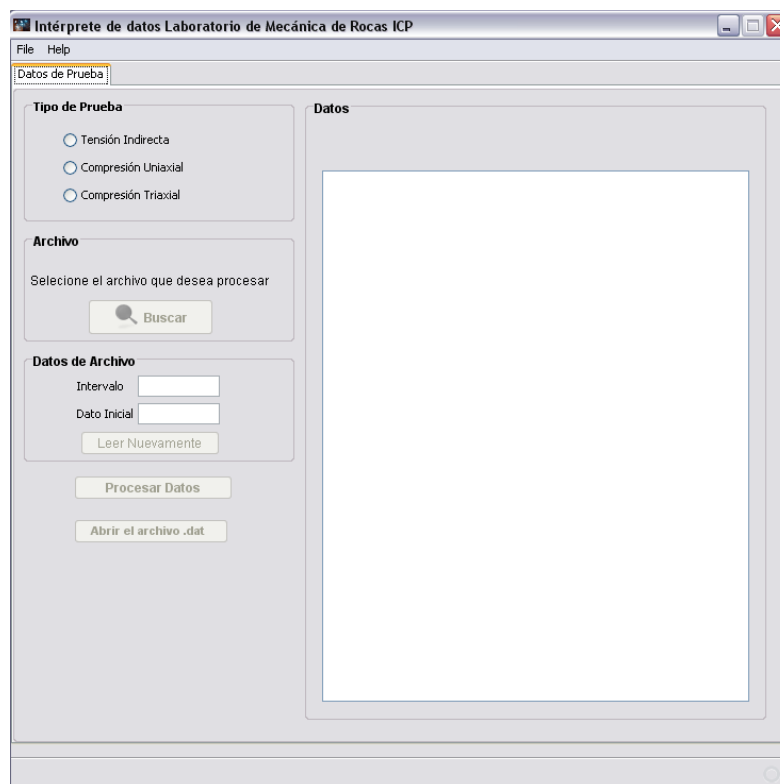


Figura 15 Pestaña inicial, selección de tipo de prueba y búsqueda de archivo a procesar.

Fuente: Interfaz diseñada por los autores del proyecto.

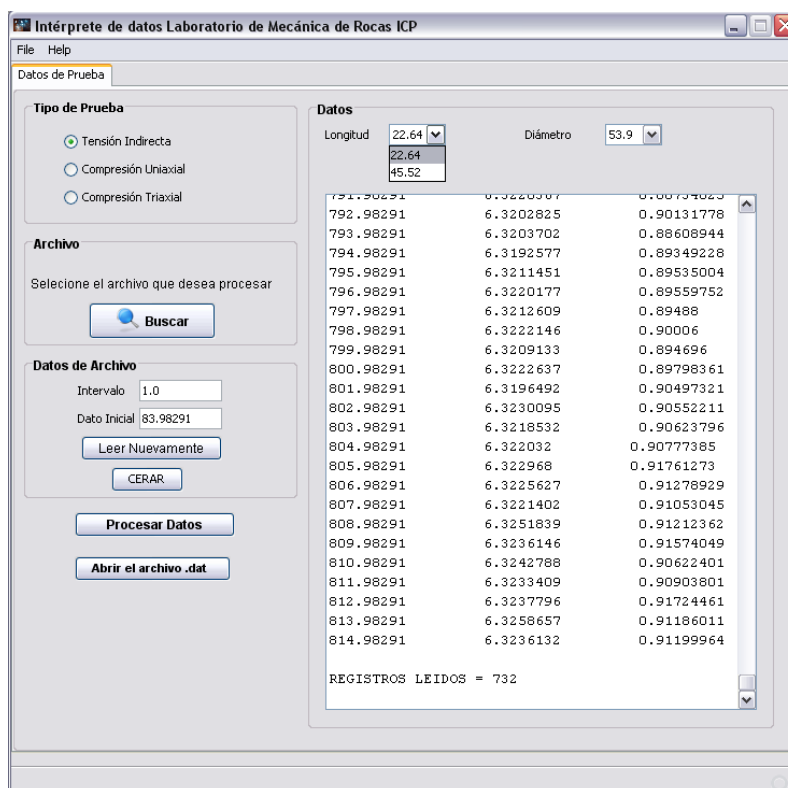


Figura 16 Pestaña inicial, opción leer nuevamente en caso de que la prueba haya sido interrumpida.

Fuente: Interfaz diseñada por los autores del proyecto.

Después de que se ha comprobado la información mencionada anteriormente se pueden procesar los datos y se obtienen los resultados (Ver Figura 17).

Hasta este momento es posible guardar información en la base de datos pues ya se tienen resultados, al tratar de hacerlo desde esta última pestaña, se pedirá al usuario que complete la información de datos petrofísicos pues el objetivo principal de éste módulo es organizar la información necesaria para el entrenamiento de las redes neuronales y, la información debe estar completa, es decir deben estar asociadas por muestra, las propiedades petrofísicas con el respectivo valor de resistencia (Ver figura 18).

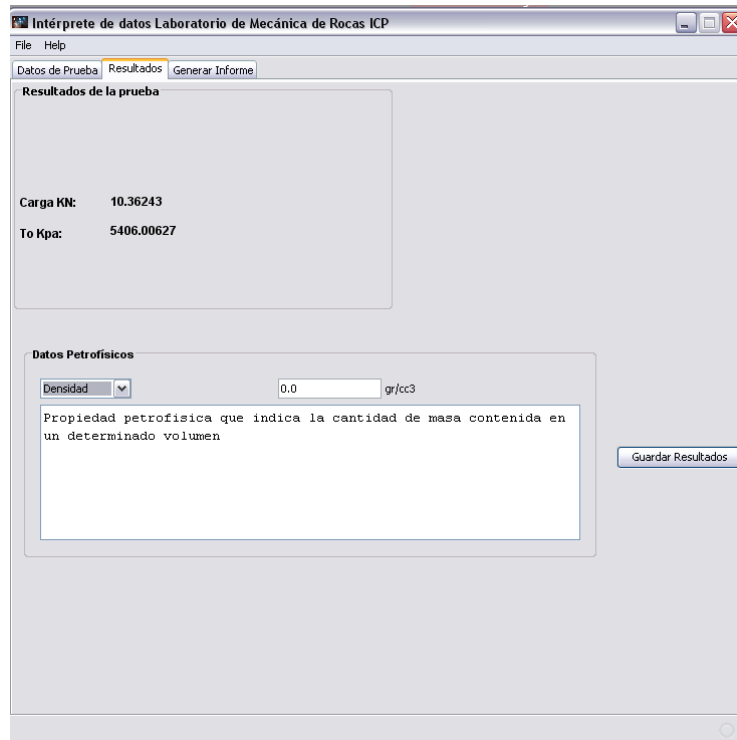


Figura 17. Pestaña de resultados.

Fuente: Interfaz diseñada por los autores del proyecto.

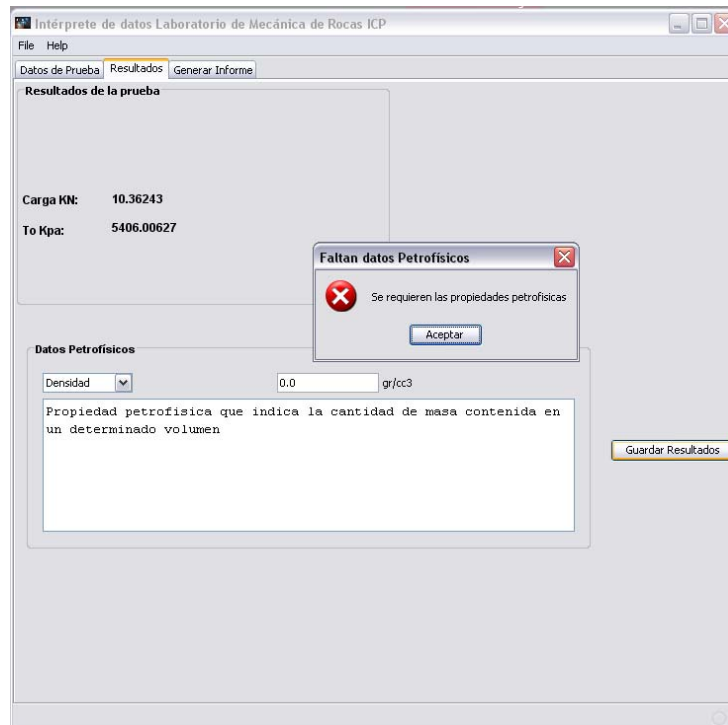


Figura 18 Error al guardar resultados pues se requieren los datos petrofísicos.

Fuente: Interfaz diseñada por los autores del proyecto.

Es en este momento en el que se hace uso de la información sobre tipos de datos petrofísicos que se mencionó anteriormente, la forma como se lleva a cabo es la siguiente, al iniciar la aplicación se consulta a la base de datos los tipos de datos petrofísicos existentes y esta información es guardada en un archivo con extensión conf que a su vez se localiza en un directorio llamado <User>/InterpreteLMR sobre la carpeta de usuario del sistema operativo en el cual se esté ejecutando la aplicación, de ésta manera, la aplicación dinámicamente puede crear estos campos que se deben guardar de forma obligatoria.

Después de tener la información completa, se realiza la consulta a la base de datos para verificar que el usuario tenga permiso de escritura (Ver figura 19).

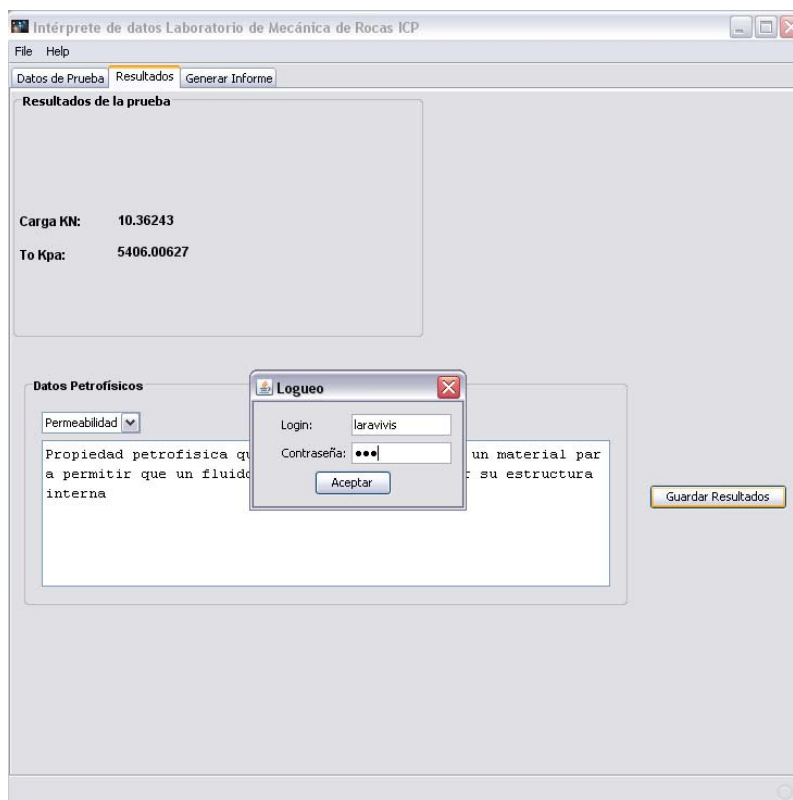


Figura 19 Validación de usuario para guardar resultados en la base de datos.

Fuente: Interfaz diseñada por los autores del proyecto.

Si se comprueba el permiso, se guarda la información y se muestra el progreso del proceso en funcionamiento en la parte inferior de la pestaña, esta tarea es

realizada por un hilo, permitiendo al usuario continuar usando la aplicación, es decir esta no se bloquea por estar realizando esta acción. (Ver figura 20)

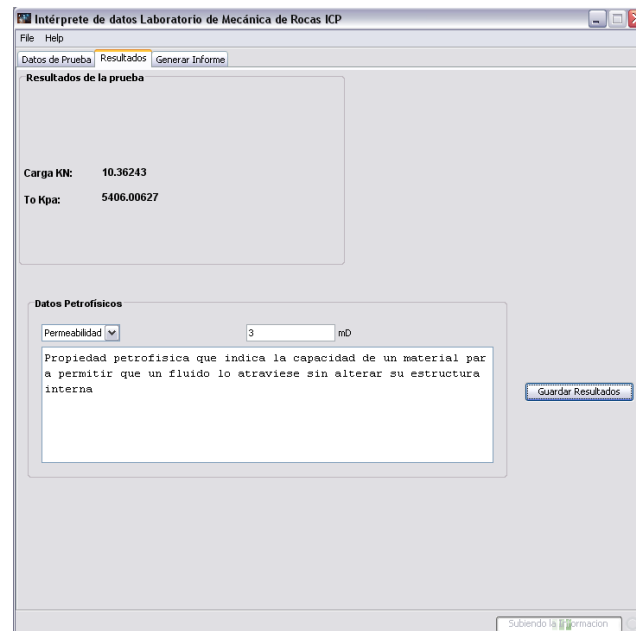


Figura 20 Barra de progreso que muestra que se está subiendo la información.

Fuente: Interfaz diseñada por los autores del proyecto.

Si se encuentra que el registro ya aparece antes de sobrescribirlo, se da la opción de actualizarlo.

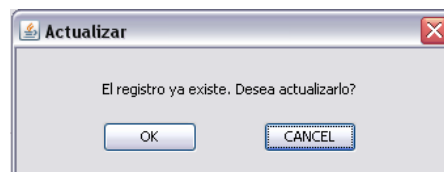


Figura 21 Actualización de datos.

Fuente: Interfaz diseñada por los autores del proyecto.

Si se desea generar un informe sobre la prueba en formato pdf con el estándar utilizado actualmente en el laboratorio, es posible hacerlo oprimiendo el botón generar informe que aparece en esta pestaña de resultados, abriendo otra que solicita la información a incluir en el reporte.

En esta tercera pestaña se presentan los campos que lleva el reporte para que el usuario ingrese la información sobre la prueba interpretada, se realizan validaciones para los campos numéricos ó de solo texto y correos electrónicos. Son necesarios el campo número de reporte pues es con este campo con el que se nombra el archivo pdf, la fecha de ejecución del ensayo, el ID de la muestra y el panel de emisión de reportes(Ver figura 22).


Figura 22 Pestaña que solicita la información necesaria para generar el reporte.

Fuente: Interfaz diseñada por los autores del proyecto.

Es posible en esta parte guardar la información en la base de datos realizando previamente la validación de usuario. De la misma manera como en la pestaña de resultados se solicitan los datos petrofísicos si se encuentran vacíos. En la base de datos se almacenan los datos petrofísicos, la información de la prueba y los resultados, adicionalmente se almacena la información del reporte en la base de datos ubicada en el servidor.

Si no se quiere guardar la información simplemente se puede generar el reporte, con lo que se muestra al usuario el archivo en el visor de pdf que se tenga por defecto (Ver figura 23), e igualmente como sucede con el archivo que contiene los tipos de datos petrofísicos, se guarda el informe en pdf en el directorio InterpreteLMR sobre la carpeta de usuario del sistema operativo sobre el cual se

esté ejecutando la aplicación para que el usuario pueda acceder a éste si no se guardó antes de cerrar la aplicación.

	INSTITUTO COLOMBIANO DEL PETRÓLEO	Versión: 03	ICP- 151-F-009
	REPORTE DE RESULTADOS PARA ENSAYOS DE TENSION INDIRECTA	Fecha aprobación: Febrero 29/2008	Hoja 1 de 3


I. INFORMACIÓN GENERAL:			
Numero de Reporte	3452		
Responsable de Laboratorio	Jenny Carvajal		
Personal Técnico	Leonardo Galvis Sergio Arias		
Fecha de Ejecución del Ensayo	Día: 1	Mes: 2	Año: 2008

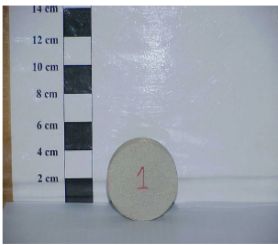
II. INFORMACIÓN DEL CLIENTE			
Cliente	Interno ICP	x	Distrito ECP
No de Solicitud	4563		
Nombre del Proyecto	Apoyo Tecnológico E&P Perfor. Pozo ECP 2004/50		
Nombre del Contacto	Jesús Augusto Rodríguez		
Dirección	Ed 1 Piso 2		
Teléfono	7267		
Correo Electrónico	jesus.rodriquez@ecopetrol.com.co		

III. CRONOGRAMA EJECUTADO			
Recepción de muestras	Día: 29	Mes: 1	Año: 2008
Inicio de actividades	Día: 31	Mes: 1	Año: 2001
Culminación de actividades	Día: 1	Mes: 2	Año: 2008

IV. INFORMACIÓN EXPERIMENTAL	
Equipo utilizado	MTS 815
Normas Utilizadas	

Los resultados obtenidos en este estudio aplican solo a la(s) muestra(s) analizada(s), el Laboratorio de Mecánica de Rocas no se hace responsable por la interpretación, extrapolación, aplicación o uso que se haga a la información contenida en este reporte. Sin excepción, este informe no se debe reproducir o fotocopiar sin autorización escrita del Laboratorio que lo emite. Para cualquier queja o sugerencia se puede dirigir a la siguiente dirección: Ecopetrol - Instituto Colombiano del Petróleo - Laboratorio Mecánica de Rocas. km 7 vía Piedecuesta Tel 6547122 6547055 FAX 6547444

	INSTITUTO COLOMBIANO DEL PETRÓLEO	Versión: 03	ICP- 151-F-009
	REPORTE DE RESULTADOS PARA ENSAYOS DE TENSION INDIRECTA	Fecha aprobación: Febrero 29/2008	Hoja 2 de 3

V. RESULTADOS																																
FOTOGRAFÍA DE LA MUESTRA																																
																																
<table border="1"> <tr> <th colspan="3">DATOS DE LA MUESTRA</th> </tr> <tr> <td>ID de la Muestra</td> <td colspan="2">2354</td> </tr> <tr> <td>Nombre del Pozo</td> <td colspan="2">Tambor B21 Mesa Santos</td> </tr> <tr> <td>Profundidad (m)</td> <td colspan="2">37750</td> </tr> <tr> <td>Diámetro (mm)</td> <td colspan="2">38.3</td> </tr> <tr> <td>Longitud (mm)</td> <td colspan="2">21.5</td> </tr> <tr> <td>Densidad (gr/cc3)</td> <td colspan="2">1.9</td> </tr> <tr> <td>Porosidad (%)</td> <td colspan="2">2.0</td> </tr> <tr> <td>Permeabilidad (mD)</td> <td colspan="2">3.0</td> </tr> <tr> <td>Temperatura (°C)</td> <td colspan="2">29</td> </tr> </table>			DATOS DE LA MUESTRA			ID de la Muestra	2354		Nombre del Pozo	Tambor B21 Mesa Santos		Profundidad (m)	37750		Diámetro (mm)	38.3		Longitud (mm)	21.5		Densidad (gr/cc3)	1.9		Porosidad (%)	2.0		Permeabilidad (mD)	3.0		Temperatura (°C)	29	
DATOS DE LA MUESTRA																																
ID de la Muestra	2354																															
Nombre del Pozo	Tambor B21 Mesa Santos																															
Profundidad (m)	37750																															
Diámetro (mm)	38.3																															
Longitud (mm)	21.5																															
Densidad (gr/cc3)	1.9																															
Porosidad (%)	2.0																															
Permeabilidad (mD)	3.0																															
Temperatura (°C)	29																															
<table border="1"> <tr> <th colspan="3">RESULTADOS</th> </tr> <tr> <td>Carga Axial en la Falla (kN)</td> <td>Resistencia a la Tensión (psi)</td> <td>Tasa de Carga (mm/s)</td> </tr> <tr> <td>10.36243</td> <td>5496.00627</td> <td>65</td> </tr> </table>			RESULTADOS			Carga Axial en la Falla (kN)	Resistencia a la Tensión (psi)	Tasa de Carga (mm/s)	10.36243	5496.00627	65																					
RESULTADOS																																
Carga Axial en la Falla (kN)	Resistencia a la Tensión (psi)	Tasa de Carga (mm/s)																														
10.36243	5496.00627	65																														

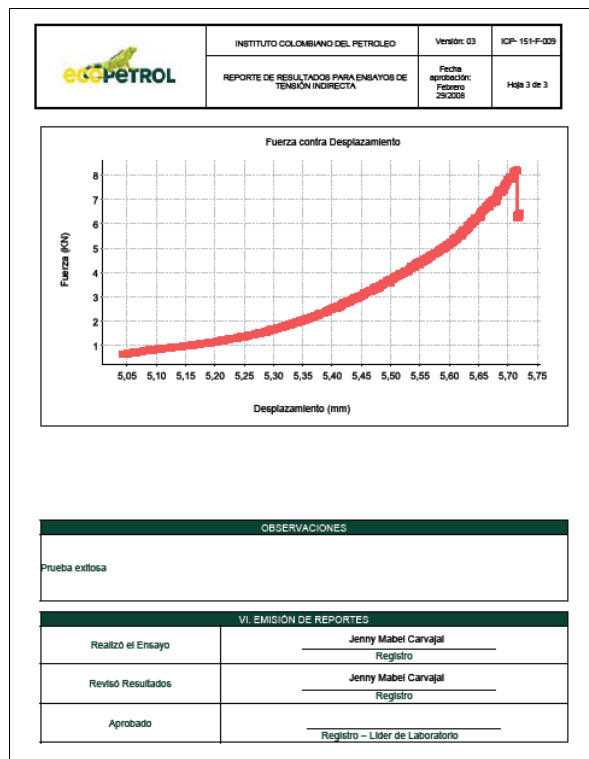


Figura 23 Ejemplo de reporte generado.

Fuente: PDF generado por la herramienta basado en el estándar del Laboratorio.

2.3.3.2. Almacenamiento de la información en la base de datos

La base de datos fue ubicada en el servidor web FAST DOMAIN para facilitar su acceso desde cualquier ubicación. La conexión JDBC por el puerto 3306 que comúnmente se utiliza para este tipo de comunicación no pudo ser realizada, ya que los administradores del servidor tienen bloqueado el puerto por razones de seguridad. Se intento ubicar la base de datos en un nuevo servidor pero se encontró que esta medida de seguridad es habitual y no se pudo realizar este tipo de conexión. Por lo anterior debió utilizarse un método llamado *http tunneling* para poder realizar las transacciones con la base de datos desde la aplicación de escritorio.

El *http tunneling* es una técnica en la que se utiliza el protocolo http para encapsular una comunicación con un servidor, una petición http se utiliza para codificar una transacción con cualquier servicio del servidor.

Para utilizar esta técnica se requiere de un intermediario en el servidor, que reciba el mensaje codificado, lo decodifique, realice la transacción directamente en el servidor y retorne una respuesta. En este caso el intermediario es un código PHP que recibe las consultas SQL con las que la aplicación de escritorio puede obtener información de la base de datos, utilizando peticiones post, como las utilizadas en las páginas web, el código PHP realiza la consulta en la base de datos y retorna el resultado.

La comunicación entre la aplicación de escritorio y el código PHP se encripta con el método CBC (*Cipher Block Chaining*), evitando que las consultas y respuesta viajen en texto plano por la red. Adicionalmente se envía una llave o clave, asegurando que el código PHP no acepte peticiones no autorizadas.

Se programó una librería que puede ser insertada en cualquier momento en un proyecto JAVA, realizando todo el proceso sin preocuparse por los detalles pues

solo se debe instanciar un objeto de la clase ConexionBD, con la que se pueden realizar las consultas SQL de una forma muy similar a como se hace al utilizar JDBC.

2.3.3.3. Archivos de configuración local

Al iniciar las aplicaciones JAVA se inicializan algunos de sus objetos con información obtenida de la base de datos, esto puede demorar el inicio de la aplicación si la velocidad de la conexión es muy baja, además evitaría el correcto inicio de la aplicación si en el momento de su ejecución no se tiene acceso a internet. Para evitar estos inconvenientes y teniendo en cuenta que los datos de inicio obtenidos de la base de datos no cambian constantemente, se implementó el código necesario para que esta lectura solo se haga cuando sea completamente necesario, es decir cuando se haya realizado una actualización.

La base de datos contiene una tabla en la que se guardan las fechas de actualización del sistema, una para cada aplicación; estas son actualizadas por el modulo web al ser utilizadas las interfaces que alteran la información de inicio de alguna de las aplicaciones.

Al inicio de la aplicación se verifica la fecha de actualización asociada a esta y se compara con la última fecha de actualización de la aplicación que es almacenada en un archivo de texto ubicado en la carpeta de la aplicación, si esta fecha es anterior a la de la última actualización del sistema o el archivo no existe, se realizan las consultas a la base de datos, luego se escribe esta información en un formato que puede ser leído posteriormente para inicializar los objetos con la misma información y otro archivo que contiene la fecha actual del servidor, que será la fecha de la última actualización de la aplicación; si la fecha obtenida de la base de datos es posterior a la ultima fecha de actualización de la aplicación, se inicializan los objetos leyendo el archivo de configuración local, evitando la

comunicación con el servidor, el caso es el mismo de no obtener respuesta del servidor, esto sucede si no hay conexión.

2.3.3.4. Análisis de requisitos

Con el fin de hacer una buena planeación y dejar clara la funcionalidad sugerida por los autores del proyecto teniendo en cuenta la necesidad de recolectar y organizar la información sobre pruebas de laboratorio, durante el desarrollo del proyecto se tuvo en cuenta la siguiente lista de requisitos de éste módulo:

- Interpretación de pruebas de laboratorio tales como Brazilian Test, compresión uniaxial y compresión triaxial.
- Importación de archivos con extensión dat generados por la MTS-815 del laboratorio, en el cual se registran segundo a segundo los diferentes valores de tiempo, fuerza, deformación, entre otras.
- Generación de reportes de las pruebas interpretadas en formato PDF, incluyendo las gráficas de deformación y resultados obtenidos de la interpretación.
- Registro en la base de datos de los valores ó resultados obtenidos de las interpretaciones, junto con la información representativa de la prueba y de la muestra.
- Asociación de cada uno de los reportes generados con un registro en la base de datos para su posterior descarga.
- Consulta de información sobre pruebas interpretadas, reportes generados y resultados obtenidos.

2.3.3.5. Diagrama de casos de uso

Al intérprete de datos pueden acceder los usuarios que tengan cuenta de usuario y el administrador es tomado simplemente como un usuario más. En la figura 24 se presenta el diagrama UML de casos de uso que muestra de forma gráfica la

interacción de un usuario sobre la herramienta. Los casos de uso se describen en las tablas 5 y 6.

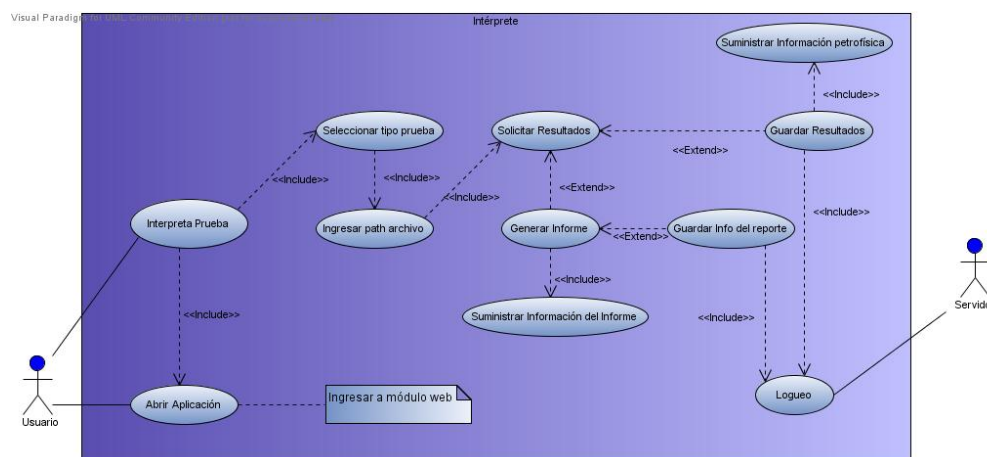


Figura 24 Diagrama de casos de uso del módulo que interpreta pruebas de laboratorio.

Fuente: Autores del proyecto (Realizado en Visual Paradigm for UML Edición community).

Caso de Uso	Descripción
Interpretar prueba	Para poder interpretar una prueba, es necesario descargar la aplicación, si ha sido descargada con anterioridad, al ejecutarla ésta se actualizará si hay una nueva versión, se podrán interpretar cualquier número de pruebas de los tres tipos disponibles, Brazilian test, Compresión Uniaxial y Compresión Triaxial; para hacerlo, el usuario debe proporcionar la ubicación del archivo .dat que contiene la información de la prueba y verificar que los valores que encuentra la aplicación tales como dato inicial, intervalo y propiedades de la muestra son los correctos, de no ser así seleccionar los correctos y mediante el botón leer nuevamente, se cargarán de nuevo, luego si se pueden generar resultados.

Generar Reporte de Prueba	<p>Después que se han obtenido resultados de alguna interpretación, es posible generar un reporte sobre la misma en el formato estándar que el laboratorio Mecánica de Rocas maneja.</p> <p>Se debe proporcionar la información necesaria para llenarlo si así se requiere, con información general de la prueba y del cliente, el cronograma ejecutado, la información experimental, los datos de la muestra, observaciones e información de la emisión de reportes.</p> <p>Además es posible agregar fotos de la muestra, proporcionando la ubicación de las mismas y de ésta manera el reporte estará completo.</p> <p>El reporte se genera en formato PDF y una copia es ubicada en el directorio InterpreteLMR sobre la carpeta de usuario del sistema operativo sobre el cual se esté ejecutando la aplicación.</p>
----------------------------------	---

Tabla 5 Descripción de casos de uso de un usuario normal sobre el intérprete.

Fuente: Autores del proyecto.

Un usuario normal pero con permisos de escritura tiene las mismas funciones ó actividades mencionadas anteriormente además de las acciones que implican escritura sobre la base de datos y que se muestran a continuación:

Caso de Uso	Descripción
Guardar Resultados	Después que se han generado resultados, es posible guardarlos en la base de datos, para esto es necesario proporcionar los valores petrofísicos porosidad y permeabilidad y autenticarse con una cuenta de usuario con permisos de escritura.
Guardar Información del	También es posible guardar los resultados junto con la información del reporte generado, se debe proporcionar

Reporte	de igual forma los valores petrofísicos de la muestra, porosidad y permeabilidad y autenticarse con una cuenta con permisos de escritura.
----------------	---

Tabla 6 Descripción de casos de uso de un usuario con permisos de escritura sobre el intérprete.

Fuente: Autores del proyecto.

2.3.3.6. Tabulación de datos

Los datos tabulados y organizados de los tres tipos de pruebas se pueden ver en las tablas 10,12 y 14.

2.4. CONFIGURACIÓN DE ARQUITECTURA DE RED UTILIZANDO ALGORITMOS GENÉTICOS

2.4.1. DESCRIPCIÓN DEL PROBLEMA DE LA BÚSQUEDA DE LA ARQUITECTURA DE RED

Las partes más esenciales e importantes al momento de resolver un problema utilizando redes neuronales artificiales son, su configuración y entrenamiento. La teoría no establece un método a seguir para poder ordenar de manera adecuada, las diferentes partes de la red, que respondan a un problema en particular ya que el entrenamiento depende directamente de la configuración, hace que ésta se resalte como la parte más importante y determinante en el proceso de obtención de la red neuronal más adecuada para la solución de un problema específico.

La forma en que se ordenan las diferentes partes de una red se basa en la experiencia de los encargados de esta tarea y algunos tips que brinda la bibliografía. Los ensayos de prueba y error son los abanderados a la hora de encontrar la mejor configuración. Cuando se habla de configuración de una red neuronal, se está teniendo en cuenta la organización de sus partes. Las partes de una red neuronal son:

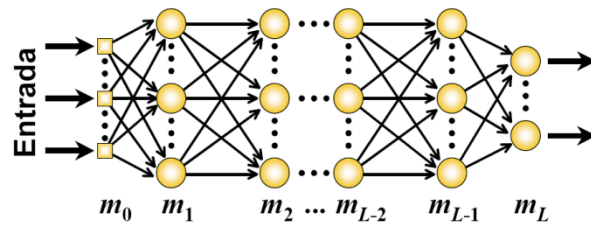


Figura 25 Partes de una Red Neuronal.

Fuente: Universidad Nacional de Cuyo, Argentina.

- Número de capas
- Número de neuronas por capa
- Funciones de activación por capa

La teoría no establece un método claro a la hora de organizar estas partes, para configurarlas dependiendo del tipo de problema que se quiere resolver, proporciona algunos puntos a tener en cuenta, pero la mayor parte del proceso depende del conocimiento previo de las personas encargadas de organizar estas partes y de pruebas de ensayo y error. Estas pruebas pueden llegar a consumir un tiempo considerable ya que las diferentes configuraciones que se pueden hacer son muchas.

El diseño de una arquitectura óptima para una red neuronal puede verse como un problema de búsqueda en un espacio de arquitecturas donde cada punto de la superficie representa una arquitectura distinta. Dado un criterio de rendimiento acerca de las arquitecturas, como el error de entrenamiento más bajo o la complejidad de red más baja, etc., el nivel de desempeño de todas las arquitecturas formará una superficie discreta en el espacio. Obtener el diseño de arquitectura óptimo es equivalente a encontrar el punto más alto sobre esta superficie. Hay muchas características sobre dicha superficie por ejemplo¹⁴:

- La superficie es infinitamente grande ya que el número de nodos y conexiones posibles es ilimitado.

¹⁴ **Xin** (1999) Evolving Artificial Neural Networks. School of Computer Science. The University of Birmingham. B15 2TT.

- La superficie no es diferenciable porque los cambios en el número de nodos o conexiones son discretos y pueden tener un efecto discontinuo sobre el rendimiento de las redes neuronales.
- La superficie es compleja y ruidosa puesto que el mapeo desde una arquitectura hacia su rendimiento optimo, es indirecto y dependiente del método de evaluación utilizado.
- La superficie es engañosa ya que arquitecturas similares pueden tener un rendimiento bastante diferente.
- La superficie es multimodal porque diferentes arquitecturas pueden tener rendimientos similares.

2.4.2. PLANTEAMIENTO DE LA SOLUCIÓN COMO UN PROBLEMA DE BÚSQUEDA

Se planteó la posibilidad de crear un algoritmo que realizara los entrenamientos de las diferentes configuraciones, las comparara, y regresara el grupo que obtuviera el error más bajo. Aprovechando que es un proceso en el cual se cambian los parámetros y se realiza una y otra vez, se puede llevar a la computadora y dejar que ella se encargue del trabajo rutinario y después el investigador interprete la información que regresa el algoritmo, ocupándose cada uno de la tarea en la que mejor se desempeña. De manera que el trabajo que realiza el algoritmo es una búsqueda sobre la superficie infinita de configuraciones de red neuronal, para encontrar un grupo que se ajuste a los datos de entrenamiento y prueba.

Este algoritmo se planteó utilizando MATLAB, aprovechando que las funciones de creación y entrenamiento de redes neuronales ya se encuentran implementadas en esta herramienta, de esta manera todo el esfuerzo se enfocó en el desarrollo del algoritmo de búsqueda, dejando la parte de los entrenamientos al toolbox de MATLAB.

Se partió de cero en la construcción del algoritmo de búsqueda, por lo que se construyó en varias etapas, partiendo de una idea básica y a medida que se comprendía más el problema se fueron incorporando nuevas características, que dieron lugar a tres prototipos, los cuales se describirán en los puntos siguientes.

2.4.3. FASES DE EVOLUCIÓN DE LA SOLUCIÓN

2.4.3.1. Fuerza Bruta

Inicialmente se planteó un algoritmo que recorriera y probara una a una las combinaciones de red neuronal, buscando una que retornara errores bajos.

El algoritmo estaba compuesto por tres funciones, la primera se encargaba de recibir todos los datos disponibles para el entrenamiento de la red, dividirlos en tres grupos y enviar las diferentes configuraciones y datos a la segunda función, luego ésta, al recibir las configuraciones las convertía en un objeto red de MATLAB, el cual era entrenado y retornado. Y, por último la tercera función recibía una red junto con los datos de prueba y luego regresaba el error que tenía la red con respecto a estos datos.

La primera función además se encargaba de ir guardando la red que presentara el menor error y al final la guardaba en un archivo para que pudiera ser utilizada posteriormente en MATLAB.

Los tres grupos en que la primera función dividía todos los datos disponibles eran los siguientes, un 30% que se usaba para probar el resultado, el restante 70% que se dividía en un 80% utilizado para el entrenamiento de la red y 20% usado como un parámetro de verificación en el entrenamiento, para vigilar que la red no perdiera su capacidad de generalización.

Este algoritmo empleaba la fuerza bruta para realizar esta búsqueda. Fuerza bruta es el nombre que se les da a los algoritmos que realizan búsquedas iterando todas las posibilidades sin tener en cuenta ninguna optimización.

El principal problema que se encontró al momento de implementar esta solución, fue el tiempo de ejecución, pues la cantidad de pruebas era inmensa. Adicionalmente se encontró que por cada configuración de red se debían realizar varios entrenamientos para poder descartarla, esto debido a que los pesos de las conexiones de las neuronas se inicializaban aleatoriamente, y se presentaba que así se tuviese una buena configuración, se podían obtener malos resultados. La única forma de reconocer por lo tanto el potencial de una configuración de red, era entrenarla en varias ocasiones, aumentando aun más el tiempo de ejecución, pues el proceso que más consumía tiempo de procesamiento era precisamente el del entrenamiento.

Además, el carácter ordenado o secuencial de la búsqueda aumentaba las probabilidades de que el algoritmo quedara atrapado en un mínimo local de la superficie de error y por esto detuviera la búsqueda sin haber llegado a la mejor solución. (Ver figura 26)

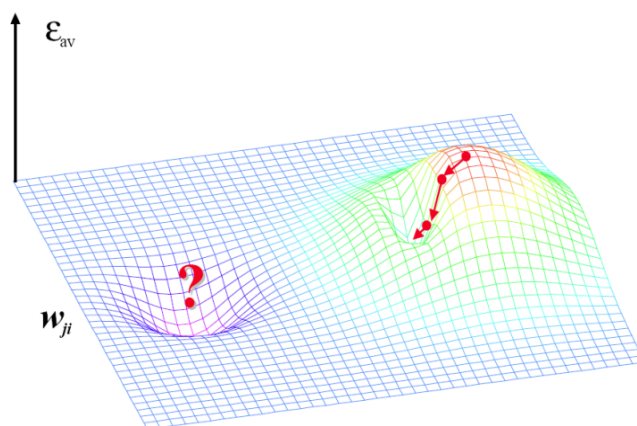


Figura 26 Mínimos locales de la superficie de error.

Fuente: Universidad Nacional de Cuyo, Argentina.

2.4.3.2. Fuerza Bruta con condiciones (Heurístico)

Se intentó incluir en la búsqueda el conocimiento que se poseía, para dotar al algoritmo de una inteligencia artificial básica; planteando la hipótesis en la que se

consideraba que al realizar esta búsqueda teniendo en cuenta algunos puntos se podrían evitar iteraciones reduciendo el tiempo de ejecución. Los puntos que se tuvieron en cuenta y el funcionamiento del algoritmo es el siguiente.

- Se considera que a mayor número de neuronas habrá un error más bajo.

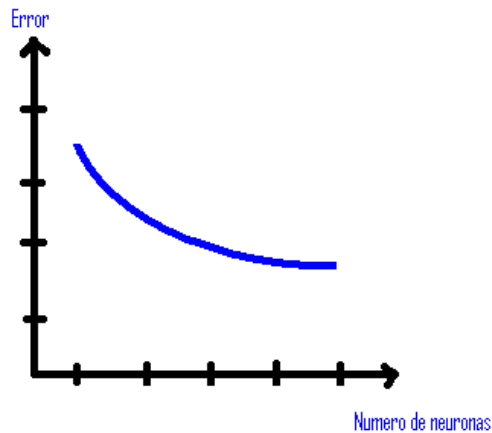


Figura 27 Error Vs. Número de neuronas.

Fuente: Autores del proyecto.

- Al aumentar una capa en una red neuronal se incrementa su complejidad en un alto grado.
- El error de una red se evalúa con datos diferentes a los usados en el entrenamiento de la red.
- Un entrenamiento en una red neuronal se inicia con pesos aleatorios, así que se deben realizar varios intentos antes de determinar si una configuración es mala o buena.

Funcionamiento:

En el proceso iterativo se intenta observar el comportamiento del problema evaluado en diferentes configuraciones de red neuronal.

Inicialmente se tienen las dos capas fundamentales, la de entrada y la de salida, sus tamaños son determinados por el número de datos de entrada y de salida respectivamente. Se inicia el proceso incluyendo una capa oculta con una

neurona, se realizan varios entrenamientos (el número de entrenamientos es decidido por el usuario) para determinar el comportamiento de esta configuración en el problema, el error se evalúa obteniendo el error absoluto entre los resultados arrojados por la red (simulados) y los resultados existentes de algunas pruebas (teóricos) que no se usaron en el entrenamiento.

El proceso continúa incrementando el número de neuronas y almacenando el comportamiento promedio de cada configuración hasta llegar al tope de neuronas que haya establecido el usuario. Se espera que el error valla disminuyendo en la medida en que se aumentan las neuronas, si esto no sucede y por el contrario al aumentar las neuronas el error se incrementa el proceso se detiene, ya que se considera innecesario continuar.

Después de esto los datos almacenados son analizados y el algoritmo verifica que se haya cumplido en algún punto el nivel de tolerancia al error que el usuario desee y que previamente ha ingresado.

Si el error es aceptable se evalúa en que rango se dan los mejores resultados y se decide qué número de neuronas se dejen en la respuesta final. Teniendo en cuenta que la mejor solución es la que tenga menor cantidad de neuronas, pues su complejidad es menor y se puede manejar más fácilmente. De esta manera si se aumentaron 10 neuronas pero el error disminuye muy poco se decidirá que la mejor solución es la anterior.

Si después de esta prueba con una capa no se obtienen resultados satisfactorios, se procede a aumentar el nivel de complejidad incrementando otra capa. Al aumentar una capa el número de combinaciones se incrementa y de igual manera lo hace el tiempo computacional necesario para llevar a cabo la prueba.

Después de incrementar una capa, el ciclo se repite para cada combinación posible, este proceso tiene en cuenta también el criterio que indica que al aumentar el número de neuronas disminuye el error, si este no es el caso pasa a la siguiente configuración sin completar la que estaba procesando, esto se hace

para evitar realizar pruebas innecesarias y reducir el tiempo de ejecución del programa. Pasadas estas pruebas se realiza la evaluación de la información almacenada para determinar si ya se ha encontrado una configuración que reduzca el error a los niveles que el usuario desea. Si nuevamente no se ha encontrado una respuesta adecuada se incrementa en uno el número de capas, y por consiguiente la complejidad, este incremento se realiza hasta encontrar una solución o hasta llegar al tope de capas establecido por el usuario.

Aunque el método era adecuado y se obtuvieron algunos resultados satisfactorios, se encontró que continuaba quedando atrapado en mínimos locales pues el proceso de búsqueda se seguía realizando de una manera ordenada.

2.4.3.3. Algoritmo Genético

En la presente investigación se ha intentado establecer una metodología para poder hallar una configuración adecuada. Se ha venido desarrollando una rutina software que se encargue de esta tarea y poco a poco se ha fortalecido incluyéndole nuevos conceptos. En esta última fase se incluyó un algoritmo genético (AGs) en la herramienta, pues en los problemas donde hay un gran espacio de posibles soluciones, es en donde más se destaca esta técnica.

¿Cuáles son las ventajas de los Algoritmos Genéticos?

- El primer y más importante punto es que los algoritmos genéticos son intrínsecamente paralelos. Ya que los AGs tienen descendencia múltiple, pueden explorar el espacio de soluciones en múltiples direcciones a la vez.
- Otra ventaja notable de los algoritmos genéticos es que se desenvuelven bien en problemas con un paisaje adaptativo complejo, aquéllos en los que la función de aptitud es discontinua, ruidosa, cambia con el tiempo o tiene muchos óptimos locales. La mayoría de los problemas prácticos tienen un espacio de soluciones enorme, imposible de explorar exhaustivamente; el reto se convierte entonces en cómo evitar los óptimos locales (soluciones que son mejores que todas las que

son similares a ella), pero que no son mejores que otras soluciones distintas situadas en algún otro lugar del espacio de soluciones. Muchos algoritmos de búsqueda pueden quedar atrapados en los óptimos locales, si llegan a lo alto de una colina del paisaje adaptativo, descubrirán que no existen soluciones mejores en las cercanías y concluirán que han alcanzado la mejor de todas, aunque existan picos más altos en algún otro lugar del mapa.

Los algoritmos evolutivos, por otro lado, han demostrado su efectividad al escapar de los óptimos locales y descubrir el óptimo global incluso en paisajes adaptativos muy escabrosos y complejos.

- Los algoritmos genéticos se destacan por su habilidad para manipular muchos parámetros simultáneamente. Muchos problemas de la vida real no pueden definirse en términos de un único valor que hay que minimizar o maximizar, sino que deben expresarse en términos de múltiples objetivos, a menudo involucrando contrapartidas (uno sólo puede mejorar a expensas de otro).
- Una de las cualidades de los algoritmos genéticos que, a primera vista puede parecer un aspecto negativo, resulta ser una de sus ventajas. Los AGs no tienen una información previa sobre el problema que deben resolver. Los diseñadores humanos utilizan información específica conocida a priori para guiar cada paso y realizar cambios esperando tener mejoras, en cambio, los algoritmos genéticos realizan cambios aleatorios en sus soluciones candidatas y luego utilizan la función de aptitud para determinar si esos cambios producen una mejora.
- La virtud de esta técnica es que permite a los algoritmos genéticos comenzar con una mente abierta, por así decirlo. Como sus decisiones están basadas en la aleatoriedad, todos los caminos de búsqueda posibles están abiertos teóricamente a un AG; en contraste, cualquier estrategia de resolución de problemas que dependa de un conocimiento previo, debe inevitablemente comenzar descartando

muchos caminos a priori, perdiendo así cualquier solución novedosa que pueda existir¹⁵.

¿Cuáles son las limitaciones de los Algoritmos Genéticos?

- La primera y más importante consideración al crear un algoritmo genético es definir una representación del problema. El lenguaje utilizado para especificar soluciones candidatas debe ser robusto; es decir, debe ser capaz de tolerar cambios aleatorios que no produzcan constantemente errores fatales o resultados sin sentido.
- Un problema con el que los algoritmos genéticos tienen dificultades son los problemas con las funciones de aptitud “engañosas” ¹⁶, en las que la situación de los puntos mejorados ofrecen información engañosa sobre dónde se encuentra probablemente el óptimo global.
- Un problema muy conocido que puede surgir con un AG se conoce como convergencia prematura. Si un individuo que es más apto que la mayoría de sus competidores emerge muy pronto en el curso de la ejecución, se puede reproducir tan abundantemente que merme la diversidad de la población demasiado pronto, provocando que el algoritmo converja hacia el óptimo local que representa ese individuo, en lugar de rastrear el paisaje adaptativo lo bastante a fondo para encontrar el óptimo global²¹. Esto es un problema especialmente común en las poblaciones pequeñas, donde incluso una variación aleatoria en el ritmo de reproducción puede provocar que un genotipo se haga dominante sobre los otros.
- Finalmente, varios investigadores ¹⁷ aconsejan no utilizar algoritmos genéticos en problemas que se pueden resolver de manera analítica. No es que los algoritmos genéticos no puedan encontrar soluciones buenas para estos

¹⁵ **John R. Koza** (Stanford University) Genetic Programming IV: Routine Human-Competitive Machine Intelligence, 2003.

¹⁶ **Malanie Mitchell** (1996) An Introduction to Genetic Algorithm.

¹⁷ **North-Holland** (1992) Tree Automata and Languages.

problemas; simplemente es que los métodos analíticos tradicionales consumen mucho menos tiempo y potencia computacional que los AGs.

2.4.3.3.1. Implementación del primer prototipo

Se implementó un primer prototipo con el fin de realizar una aplicación práctica y obtener conclusiones propias. Este primer prototipo se desarrolló en *MATLAB*, herramienta que brinda un fácil manejo de redes neuronales.

El objetivo de este algoritmo es encontrar los pesos de una red neuronal para resolver un problema específico. El número de capas y neuronas son pasados como parámetros, no son hallados por el algoritmo. Se implementó la estructura de un algoritmo genético simple que se puede observar en la figura 28. En la que se usa una representación binaria.

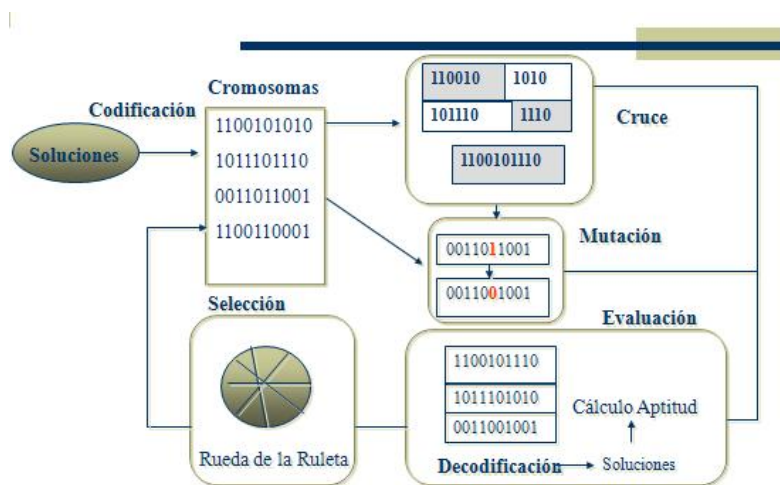


Figura 28 Estructura de un Algoritmo Genético Simple.

Fuente: Universidad Nacional de Cuyo, Argentina.

Para encontrar la cantidad de bits que se necesitan para representar una solución se uso la siguiente formula.

$$2^{m_j-1} \leq (b_j - a_j) * 10^n \leq 2^{m_j} - 1 \quad \text{Ecuación 4} \quad m_j \text{ Cantidad de bits necesarios para representar una solución.}$$

Donde a_j y b_j son respectivamente los extremos inferior y superior del rango y n es la precisión de la variable que se quiere representar.

La Selección *elitista*, fue el método de selección que se utilizó en este algoritmo, conservando intacto el gen que tenga la mejor respuesta al problema en cada iteración.

El cruce se realiza buscando un punto aleatorio en el cual los genes que se cruzan, intercambian su código genético.

$$\left\{ \begin{array}{l} V_1' = [001110101 | 110011000000010101001000] \\ V_3' = [000001010 | 1001010011011101111110] \\ V_1'' = [001110101001010011011101111110] \\ V_3'' = [000001010110011000000010101001000] \\ V_2'' = [001110101110011000000010101001000] = V_2' \end{array} \right.$$

Figura 29 Ejemplo de Cruce entre 2 genes.

Fuente: Universidad Nacional de Cuyo, Argentina.

Los bits de cada gen hijo tienen una baja probabilidad de mutación, si se cumple esta probabilidad se hace efectiva invirtiendo el valor de este bit.

La función de aptitud utilizada para este algoritmo fue:

$$E = \frac{1}{2} \sum (t_k - o_k)^2 \quad \text{Ecuación 5 Función de aptitud utilizada.}$$

Donde t_k son los resultados deseados y o_k son los regresados por la red neuronal.

El algoritmo se dividió en funciones que realizan una tarea específica, para facilitar la futura evolución del mismo, estas especialidades son:

FUNCIONES	DESCRIPCIÓN
ConfigRedGenetico	Esta función recibe los datos de entrada y de salida de una red neuronal y adicionalmente un

	vector en el que se define el número de capas y neuronas de la red.
<i>cal_num_bit</i>	Calcula el número de bits necesarios para representar un variable.
<i>binarioTodecimal</i>	Convierte un número binario en decimal.
<i>binarioTorepresentacion</i>	Convierte un número binario en el número que está representando.
<i>genético</i>	Es la función encargada de organizar el flujo de los datos, la cantidad de iteraciones, la llamada a las funciones, entre otros.
<i>cromo_aleatorio</i>	Recibe un tamaño y genera un cromosoma con bits aleatorios.
<i>cruzar_pob</i>	Recibe una población de cromosomas y selecciona aleatoriamente cuales se van a cruzar.
<i>cruzar_cromosoma</i>	Recibe dos cromosomas y los cruza escogiendo un punto aleatorio.
<i>mutar</i>	Recibe un gen, recorre cada uno de sus bits y si un número aleatorio cumple cierta probabilidad, intercambia su valor.
<i>aptitud</i>	Recibe un gen y lo evalúa, regresando su error.
<i>order_pobla</i>	Ordena la población dependiendo de los resultados obtenidos después de haber pasado por la función de amplitud.
<i>Represent_pobla</i>	Recibe una población de genes y regresa los valores que están representando.

Tabla 7 Funciones del algoritmo genético.

Fuente: Autores del proyecto.

Se verificó el correcto funcionamiento del algoritmo genético, utilizándolo para resolver un problema sencillo en el cual se buscaban 2 variables que maximizaran la siguiente función:

$$f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + \sin(20\pi x_2)$$

Ecuación 6 Función maximizada inicialmente.

Se obtuvieron los resultados esperados comprobando que el algoritmo funcionaba correctamente.

PRUEBA DE CONFIGURACIÓN DE PESOS EN UNA RED NEURONAL

Se intentó configurar una pequeña red de una capa oculta de 3 neuronas que recibiera como dato de entrada una variable y diera como resultado el seno de esa entrada.

El algoritmo estaba configurado para detenerse después de un número de iteraciones preestablecidas.

El error fue definido como el promedio del valor absoluto de las diferencias entre el valor real y el valor obtenido con la red neuronal.

Con 100 iteraciones el error fue de 70%.

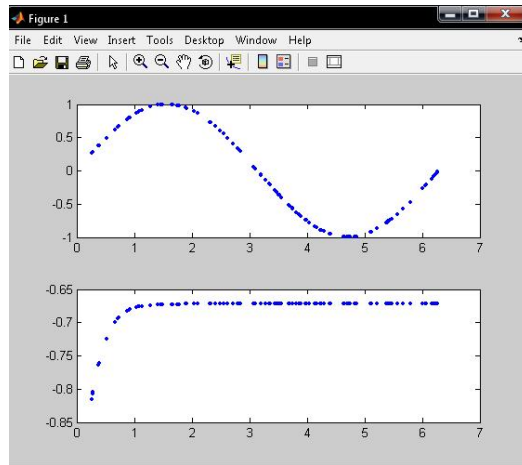


Figura 30 Gráfica de error de red configurada para obtener el seno utilizando 100 iteraciones.

Fuente: Autores del proyecto.

Con 500 iteraciones el error bajó a 30%.

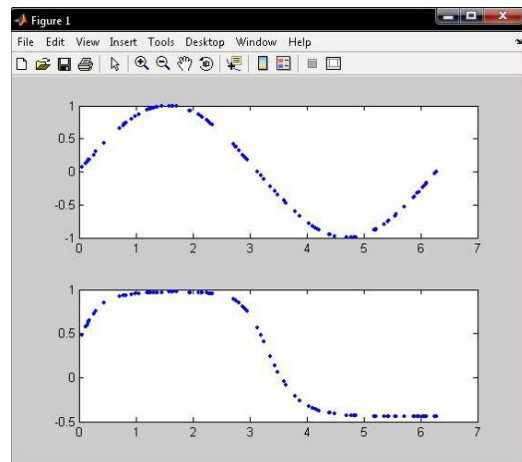


Figura 31 Gráfica de error de red configurada para obtener el seno utilizando 500 iteraciones.

Fuente: Autores del proyecto.

Con 1000 iteraciones el error bajó a 16%.

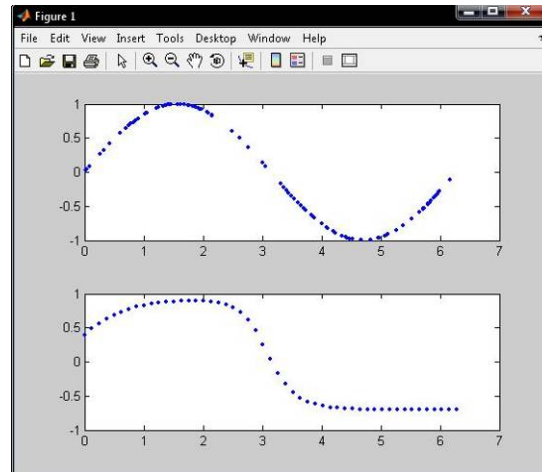


Figura 32 Gráfica de error de red configurada para obtener el seno utilizando 1000 iteraciones.

Fuente: Autores del proyecto.

Se modificó el algoritmo para que se detuviera en el momento en el que el error se redujera a un 1%, y después de 4 horas de procesamiento no se detuvo.

En el transcurso de estas pruebas se pudo observar que el espacio de búsqueda es infinito, pues se trata de números reales en el rango $[-3,3]$ con múltiples combinaciones posibles.

Los inconvenientes presentados a la hora de tratar de configurar una red tan simple hicieron pensar que tratar de buscar la configuración de los pesos a través de un algoritmo genético, no es el camino más apropiado. El enfoque del algoritmo debía ir encaminado en la búsqueda de la configuración de capas y neuronas por capa. Esto dió lugar a un siguiente prototipo.

2.4.3.3.2. Segundo prototipo

Se programó un algoritmo genético utilizando MATLAB, el algoritmo busca el número de capas ocultas y el número de neuronas en cada una de estas capas, en una red neuronal para que ésta se ajuste apropiadamente a los valores de la solución de un problema y pueda generalizarlo de manera adecuada.

El algoritmo recibe como parámetros:

Entradas: Estructura de datos que contiene todos los datos de entrada que se utilizarán para el entrenamiento de las redes neuronales probadas por el algoritmo.

Salidas: Estructura de datos que contiene las salidas correspondientes a cada entrada.

Población: Cantidad de individuos que conforman la población del algoritmo genético.

Probabilidad de cruce: Es un valor entre 0 y 1 que indica la probabilidad de que un individuo de la población se cruce y genere un nuevo individuo.

Probabilidad de Mutación: Es un parámetro que indica la probabilidad de que un individuo sufra una mutación. Una mutación en este algoritmo es un cambio aleatorio en un valor del gen que representa una solución, esto se utiliza para generar soluciones diferentes, en el caso en el que el algoritmo esté convergiendo a un mínimo local, la mutación podría generar una solución fuera del rango al que se dirige éste; si la mutación genera una solución buena le indicaría al algoritmo un nuevo camino de búsqueda.

Máximo número de capas: Este parámetro indica cual será el número máximo de capas ocultas que podrá tener una red neuronal.

Máximo número de neuronas por capa: Este parámetro indica cual será el número máximo de neuronas que podrá tener una capa oculta.

Máximo número de generaciones: Este será el número de veces que itera el algoritmo genético.

Tolerancia al error: Es un porcentaje que indica hasta qué valor se valida una respuesta. Por ejemplo si se ingresa 20, la tolerancia será de 20% y si un resultado se encuentra por debajo de este límite será tomado como un acierto.

Número de entrenamientos: Señala el número de veces que será entrenada una configuración de red. El entrenamiento de las redes neuronales inicia con pesos aleatorios, por esto no puede descartarse una configuración de red con un solo entrenamiento que haya tenido malos resultados; se debe entrenar varias veces, para encontrar una configuración de pesos que se adecúe mejor al problema.

El algoritmo genético dependiendo de los parámetros con los que sea ejecutado, requiere de una gran cantidad de tiempo de procesamiento, el tiempo necesario para una ejecución con 100 generaciones, tamaño de población 100 y 15 entrenamientos por configuración, fué de 6 horas y 32 minutos.

El algoritmo regresa una estructura de datos que contiene la última generación del algoritmo genético; esta estructura contiene cada uno de los individuos, sus respectivos errores, y la red neuronal ya entrenada para cada caso.

2.4.4. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL ALGORITMO GENÉTICO

Desde el inicio del proyecto se planteó la posibilidad de desarrollar un algoritmo que fuera capaz de configurar automáticamente las redes neuronales necesarias para cumplir con los objetivos del proyecto. Este fue un trabajo extenso que se llevo a cabo en varias fases o prototipos, el planteamiento inicial y los prototipos descritos en puntos anteriores, fueron desarrollados en conjunto con los estudiantes de sistemas EDGAR ALBERTO MENDEZ Y JUAN SEBASTIAN MARIÑO quienes desarrollaron el proyecto de grado titulado SISTEMA AUTOMÁTICO DE ENTRENAMIENTO DE REDES NEURONALES ARTIFICIALES BASADO EN EL AJUSTE GENÉTICO DE PARÁMETROS Y VARIACIÓN DE ARQUITECTURA ¹³. La herramienta que finalmente fue desarrollada en esa tesis fue retomada y luego de algunas modificaciones fue

utilizada para realizar la búsqueda de las configuraciones que se utilizaron para cumplir con los objetivos del proyecto.

2.4.5. DESCRIPCIÓN DE LA HERRAMIENTA Y SU ENTORNO GRÁFICO

EL SISTEMA AUTOMÁTICO DE ENTRENAMIENTO DE REDES NEURONALES ARTIFICIALES BASADO EN EL AJUSTE GENÉTICO DE PARÁMETROS Y VARIACIÓN DE ARQUITECTURA fue desarrollado en MATLAB 7.5.0 R2007b. Se utilizó este programa ya que incluye funciones especializadas en el manejo de las redes ya mencionadas, permitiendo que los desarrolladores se enfocaran en la construcción del algoritmo de búsqueda, confiando plenamente en los resultados de los entrenamientos realizados por las funciones de MATLAB.

La ventana principal, permite escoger el tipo de error con el que se quiere trabajar:

- Absoluto: Diferencia entre el valor simulado y el valor real.
- Relativo: Razón del error absoluto y el valor real.

Adicionalmente se escoge el tipo de detección que se utilizará en los entrenamientos:

- Detención temprana: En la detención temprana se agrega un grupo de datos adicional en el entrenamiento, estos datos se utilizan para detener el entrenamiento en el momento en el que las respuestas de la red se alejen demasiado de las respuestas de los datos de verificación.
- Especificando épocas:

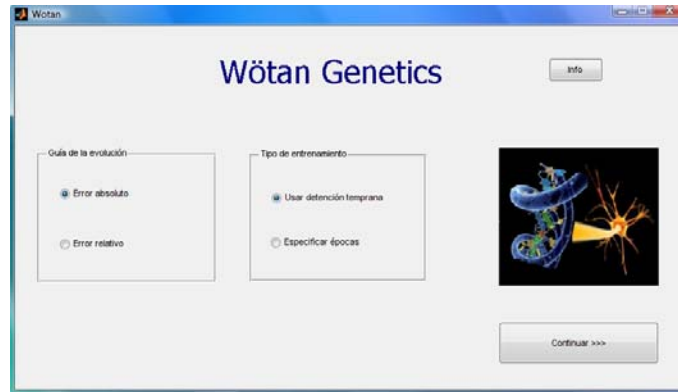


Figura 33 Ventana principal configurador de redes neuronales artificiales.

Fuente: MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.

Si se escoge detención temprana y se pulsa el botón siguiente, se abre una interfaz en la que se solicitan 6 grupos de datos:



Figura 34 Carga de datos, utilizando detención temprana en el entrenamiento.

Fuente: MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.

- Entrenamiento (Entrada y Salida)
- Verificación (Entrada y Salida)
- Prueba (Entrada y Salida)

La segunda opción de entrenamiento, es en la cual se determina previamente el número de épocas o número de veces en las que le serán mostrados los datos de entrenamiento a la red.

Al escoger este tipo de entrenamiento se abre una interfaz similar a la descrita anteriormente, teniendo como diferencia la solicitud de cuatro grupos de datos, entrenamiento y prueba cada una con sus respectivas entradas y salidas. En esta interfaz adicionalmente se solicita el número de épocas para los entrenamientos.

Luego de llenar los datos se pulsa el botón continuar, a continuación se puede observar una interfaz en la que se solicitan los parámetros del algoritmo genético (esta interfaz es la misma para los dos tipos de entrenamiento) los parámetros solicitados son:

Carga de parámetros

Parámetros

Tamaño de la población	20
Probabilidad de mutación	0.1
Probabilidad de cruce	0.8
Máximo de capas	4
Máximo de neuronas	10
Número de entrenamientos	30
Algoritmo de entrenamiento	Trainlm
Generaciones	20
Tolerancia error	0.01

<<< Atrás Ejecutar >>>

Figura 35 Parámetros que se ingresan al algoritmo genético.

Fuente: MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.* 2008.

- Tamaño de la población.
- Probabilidad de mutación.
- Probabilidad de cruce.
- Máximo número de capas.
- Máximo número de neuronas por capa.
- Número de entrenamiento para cada red.
- Algoritmo de entrenamiento:

Estos son los diferentes tipos de algoritmos de entrenamiento del toolbox de MATLAB.

1. *Trainlm.* es una función de entrenamiento de red que actualiza los valores de los pesos y bias, de acuerdo con la optimización de Levenberg-Marquardt.
2. *Traingdx.* es una función de entrenamiento de red que actualiza los valores de los pesos y bias, de acuerdo con el gradiente de impulso descendiente y una tasa de aprendizaje adaptativo.
3. *Trainbfg.* es una función de entrenamiento de red que actualiza los valores de los pesos y bias, de acuerdo con el método de BFGS quasi-Newton.
4. *Trainrp.* es una función de entrenamiento de red que actualiza los valores de los pesos y bias, de acuerdo con la resistencia del algoritmo de Backpropagation.
5. *Traingda.* es una función de entrenamiento de red que actualiza los valores de los pesos y bias, de acuerdo con el gradiente descendente con tasa de aprendizaje adaptativa.

- Número de generaciones.
- Tolerancia al error.

Estos datos ya vienen almacenados por defecto, pero si el usuario ha trabajado con algoritmos genéticos y conoce el significado de cada uno de los parámetros, puede alterarlos a voluntad.

Para iniciar el proceso de búsqueda se pulsa el botón Ejecutar.

La duración del proceso de búsqueda de la configuración más apropiada para el conjunto de datos varía dependiendo de los parámetros ingresados al algoritmo genético.

Al finalizar el proceso de búsqueda la herramienta presenta una interfaz con la lista de redes de la última población (Ver figura 36). En esta ventana se pueden seleccionar cada una de las redes y ver su gráfica de error, además si se desea se puede guardar la red, con extensión mat, para utilizarla en MATLAB.

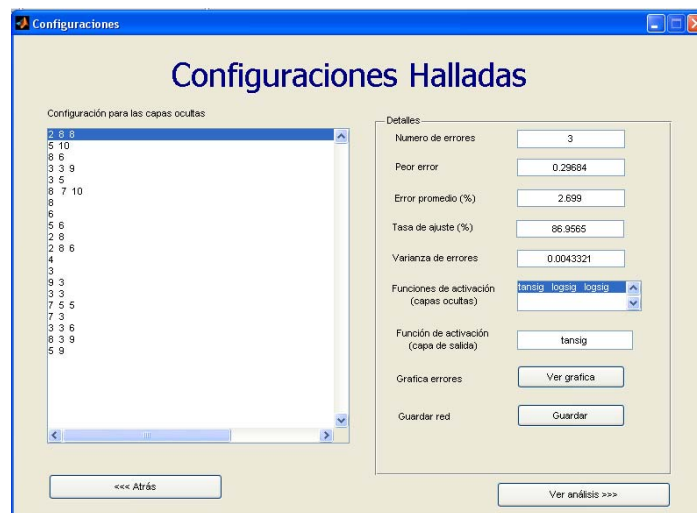


Figura 36 Interfaz en la que el configurador muestra los resultados.

Fuente: MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.* 2008.

Al pulsar el botón Ver análisis, la herramienta presenta un análisis estadístico de la búsqueda realizada por el algoritmo genético, en la figura 37 se pueden ver algunas de las gráficas.

- Error vs número de capas ocultas.
- Redes probadas vs número de capas.
- Errores vs número de neuronas por red.
- Redes probadas vs número de neuronas por configuración.
- Variación de los errores en las generaciones.

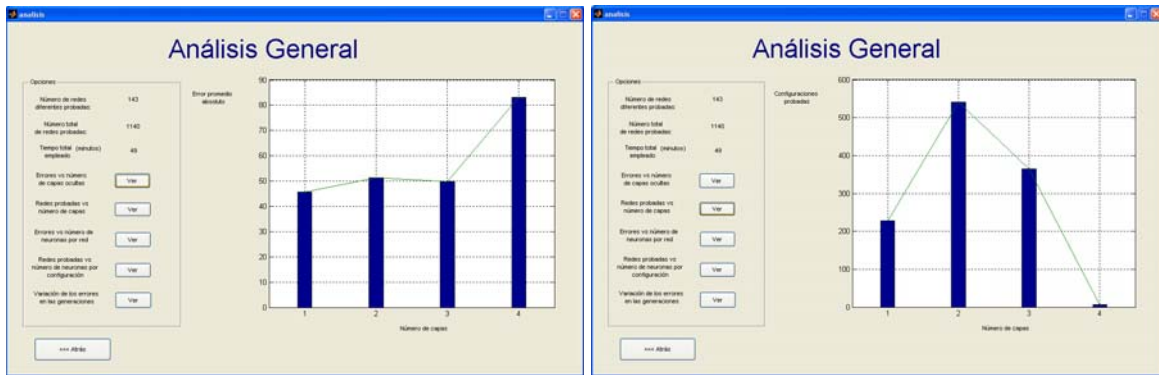


Figura 37 Análisis estadístico de la búsqueda realizada por el algoritmo genético.

Fuente: MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.*

2.4.6. MEJORAS

Para la búsqueda de configuración y entrenamiento de las redes neuronales que se utilizaron en el proyecto se utilizó el SISTEMA AUTOMÁTICO DE ENTRENAMIENTO DE REDES NEURONALES ARTIFICIALES BASADO EN EL AJUSTE GENÉTICO DE PARÁMETROS Y VARIACIÓN DE ARQUITECTURA, el funcionamiento de esta herramienta fue descrito en la sección anterior. Se hicieron algunos cambios en la interfaz y se agregaron otras funcionalidades para mejorar la herramienta, logrando con esto mejores prestaciones ajustadas a las necesidades de este proyecto.

2.4.6.1. Entrada de Datos

Se realizaron algunos cambios en la interfaz de entrada de datos para agilizar y facilitar el ingreso de los mismos en el sistema.

Los datos se debían normalizar antes de ingresar en el sistema, esto se convierte en un proceso tedioso si se están realizando múltiples pruebas y reordenamiento de datos. Además las entradas y las salidas debían ingresarse en archivos separados.

Para facilitar el ingreso de datos y evitar los inconvenientes antes mencionados se modificó la interfaz. Esta nueva interfaz solicita el tipo de normalización que se quiere utilizar en las entradas y salidas respectivamente:

- Ninguna: No realiza ninguna acción sobre los datos.
- Sigmoidal: Este tipo de normalización ajusta los datos en un rango entre cero y uno $[0,1]$.
- Tangente Hiperbólica: Este tipo de normalización ajusta los datos en un rango entre menos uno y uno $[-1,1]$.

La interfaz también solicita el número de entradas y de salidas. Después de llenar estos datos, se pueden cargar los archivos de datos en los que se encuentran agrupadas las entradas y salidas en sus valores originales (Ver figura 38).



Figura 38 Interfaz modificada que facilita el ingreso de datos.

Fuente: Adaptado de MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.*

2.4.6.2. Manejo de Errores

En cuanto al manejo de errores como criterio de verificación se realizaron algunos cambios ya que el error calculado (comparación entre resultados reales y simulados) se hallaba tomando los valores normalizados que por estar en un

rango pequeño ya fuese entre $[-1,1]$ o $[0,1]$ tomaba valores pequeños en comparación con los que se obtienen cuando se calcula tomando los valores desnormalizados.

2.4.6.3. Conversión de Arquitectura de Red Neuronal a XML

En el proyecto se tenía la necesidad de utilizar las configuraciones de red seleccionadas en el configurador automático fuera de MATLAB, esto para evitar la limitante que conllevaría el estar ligado a esta herramienta, ya que así las redes solo podrían ser utilizadas por los usuarios que tuvieran una instalación de esta herramienta en su computadora.

Para solucionar esto, se creó una función en MATLAB que recibe un objeto red neuronal y luego de procesarlo lo guarda en un archivo de extensión XML y que contiene en este mismo lenguaje la descripción de las partes de la red neuronal. Esta función se adaptó a la interfaz gráfica, para facilitar su manejo sobre el configurador, con esto, solo se debe escoger una ubicación y un nombre para que el programa guarde este archivo en el lugar que el usuario desee.

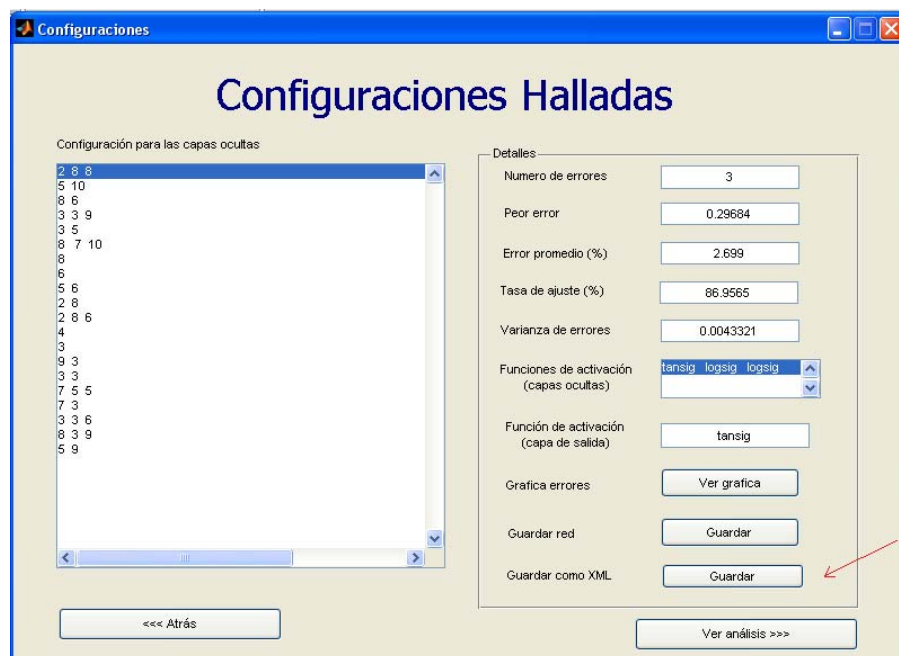


Figura 39 Interfaz modificada que permite guardar configuración como XML.

Fuente: Adaptado de MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.*

2.5. IMPLEMENTACIÓN DE LA ARQUITECTURA DE RED NEURONAL EN JAVA

2.5.1. ¿POR QUÉ USAR LA RED FUERA DE MATLAB?

La herramienta que se desarrolló para realizar estimaciones de propiedades mecánicas a partir de propiedades petrofísicas utilizando redes neuronales, se desarrolló para el laboratorio de mecánica de rocas del Instituto Colombiano del Petróleo (ICP). La herramienta será utilizada por los diferentes profesionales encargados del laboratorio, que se pueden encontrar en diversas ubicaciones y plataformas computacionales, por esto se diseña y construye la herramienta para que pueda ser utilizada con gran facilidad y con la menor cantidad de restricciones. Por lo anterior se decidió que JAVA era la mejor opción para la implementación de este software ya que podía ser ejecutado en cualquier sistema operativo, sin embargo para hacer realidad esto se debía encontrar la forma de llevar y utilizar las configuraciones de red neuronal configuradas y entrenadas en MATLAB a esta aplicación JAVA, para esto se utilizó el framework JCORTEx.

2.5.2. JCORTEx

FRAMEWORK PARA REDES NEURONALES EN JAVA

JCortex es una solución completa que permite a los desarrolladores de software crear, entrenar y utilizar redes neuronales artificiales en proyectos Java. Está dividido en dos partes: Framework JCortex, librería para la implementación de redes neuronales y JCortexBuilder, un ambiente gráfico en el que se pueden crear y entrenar redes neuronales.

Su autor es Miguel Lara Encabo, y fue desarrollado como proyecto de grado en la Universidad Pontificia Comillas ICAI España en el año 2006, con Juan Carlos Esquivel Díaz como director de proyecto. Está licenciado bajo la licencia GNU LESSER GENERAL PUBLIC LICENSE (LGPL), que permite su modificación, distribución y uso libre.

2.5.3. EDICIÓN DE JCORTX

Se realizó un estudio del código y del funcionamiento del framework, ya que no se realizarían entrenamientos de redes neuronales sobre la herramienta JAVA, pues ya se tenían estas configuraciones en MATLAB, se editó entonces el código dejando solo la parte necesaria para cargar las configuraciones y poder realizar la parte de ejecución de las redes.

2.5.4. ARCHIVO XML

El framework trae incorporados métodos que permiten exportar e importar las configuraciones de red neuronal que se crean, esta utilidad se aprovechó para realizar la exportación de las configuraciones de red neuronal de MATLAB a JAVA. Se construyó una función en MATLAB que escribe la configuración de una red neuronal en el formato que el Jcortex es capaz de interpretar y de esta manera se pueden utilizar redes neuronales entrenadas en MATLAB sobre JAVA.

2.5.5. FORMA DE CARGAR ARCHIVO XML A LA BASE DE DATOS

El archivo XML que contiene la configuración de red neuronal que se utilizará para determinada estimación, debe ser almacenado en el servidor del proyecto junto con algunos datos que se almacenan en la base de datos, esto se realiza a través del módulo web del proyecto, los usuarios que tengan el permiso correspondiente tendrán el privilegio de acceder al formulario web que les permite realizar esta tarea. Se requiere de un permiso especial para realizar este procedimiento para

garantizar que las configuraciones de red que usa el sistema sean provistas o suministradas sólo por el personal que ha sido previamente escogido para realizar esta labor.



Figura 40 Carga de configuraciones a la base de datos a través del módulo Web.

Izquierda: Configuraciones existentes en la BD con la opción de cargar configuración.

Derecha: Formulario para cargar configuración.

Fuente: Autores del proyecto.

2.5.6. DESARROLLO DE LA HERRAMIENTA

2.5.6.1. Descripción de la herramienta

En el momento que la aplicación es ejecutada se realiza una consulta al servidor del proyecto, solicitando la información de las redes neuronales que se encuentran cargadas en el sistema, durante este proceso se muestra el splash de la aplicación (Ver figura 41). Con la información obtenida de la base de datos, son inicializados los objetos del programa, como por ejemplo los mínimos y máximos utilizados en la normalización de la información de entrenamiento de una red, para que se puedan trabajar debidamente las entradas y salidas en cada red. Al terminarse este proceso se lanza el GUI de la herramienta.



Figura 41 Splash del estimador mientras se cargan datos del servidor.

Fuente: Adaptado del Laboratorio Mecánica de rocas.

La interfaz cuenta con tres pestañas, en la primera se puede escoger el tipo de prueba con la que se va a trabajar, al escoger el tipo de prueba inmediatamente se llena la lista desplegable con la red o redes disponibles para este tipo de prueba. Al escoger alguna de las redes, se muestra su información.

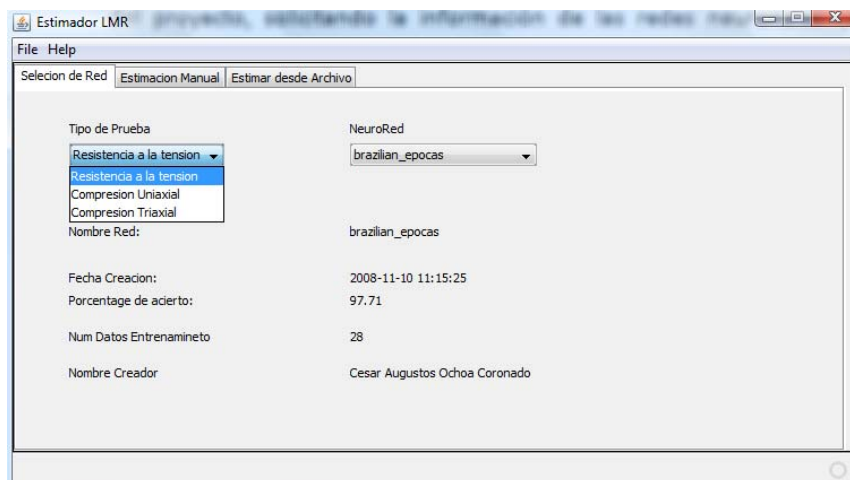


Figura 42 Interfaz del Estimador. Pestaña de selección de red.

Fuente: Interfaz diseñada por los autores del proyecto.

En la segunda y tercera pestaña se hace uso de la red neuronal seleccionada. En la segunda pestaña se hace un ingreso manual de los datos (Ver figura 43), en la tercera se carga un archivo de datos para que sean procesados en bloque (Ver figura 44). Al seleccionarse alguna de estas dos pestañas el sistema hace solicitud del archivo XML de configuración al servidor, a través del protocolo FTP es

descargado este archivo, con el que se inicializa el objeto red neuronal para posteriormente ser utilizado como se describió.

El sistema almacena una copia del archivo XML en la carpeta de la aplicación la cual está localizada en <USER>/Estimador_LMR, la aplicación revisa si el archivo ya existe, si es así no se conecta al servidor, esto se hace para agilizar el proceso, evitando conexiones innecesarias.

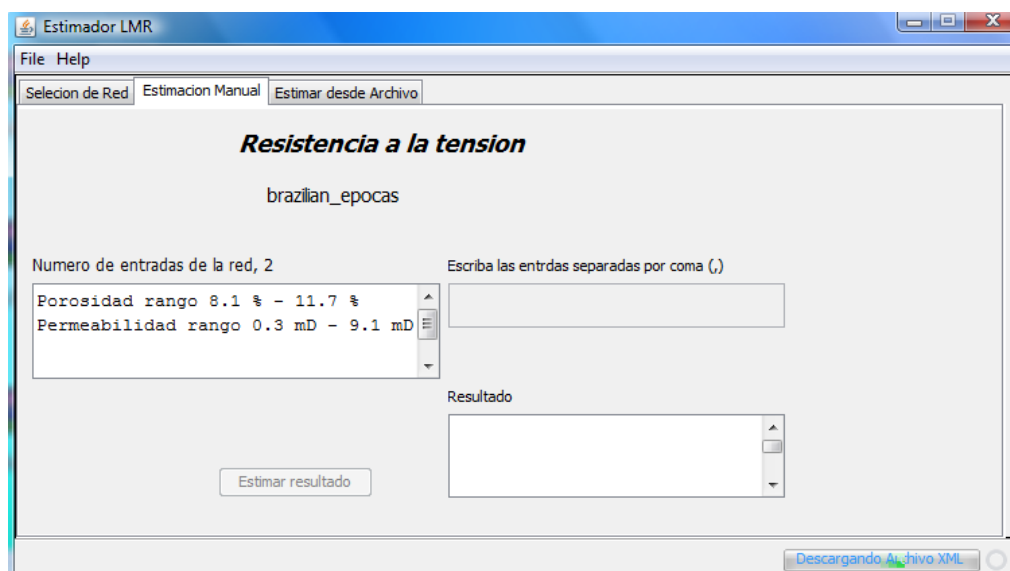


Figura 43 Interfaz del Estimador. Pestaña de Estimación manual.

Fuente: Interfaz diseñada por los autores del proyecto.

Las pestañas en la que se hace uso de la red neuronal muestran los rangos que se utilizaron en el entrenamiento de la red y por lo tanto los rangos en los que son válidas las entradas a ésta.

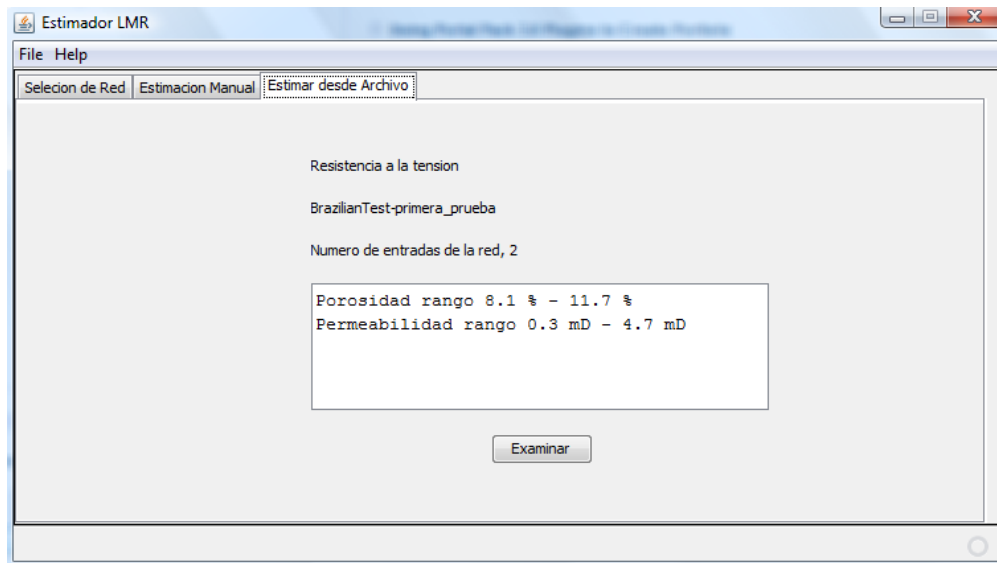


Figura 44 Interfaz del estimador. Pestaña de estimación desde archivo.

Fuente: Interfaz diseñada por los autores del proyecto.

2.5.6.2. Diagrama de Casos de Uso

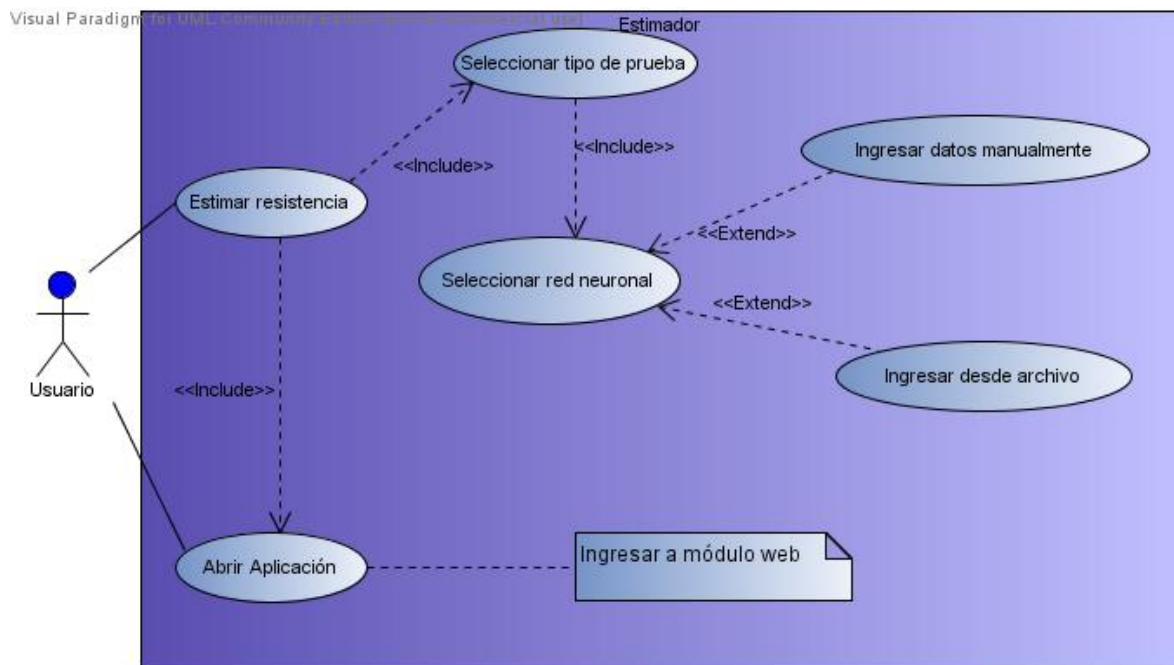


Figura 45 Diagrama de Casos de uso del Estimador.

Fuente: Autores del proyecto (Realizado en Visual Paradigm for UML Edición community).

Casos de uso	Descripción
Estimar resistencia	Para poder estimar un valor de resistencia, el usuario debe conocer el tipo de prueba de laboratorio sobre la que quiere encontrar el valor de la resistencia.
Seleccionar red neuronal	Después de escoger la prueba, se muestran las redes disponibles para que el usuario seleccione la que crea conveniente utilizar en la estimación de la resistencia.
Ingresar datos manualmente	La estimación se puede hacer manualmente ingresando los datos de entrada separados por coma.
Ingresar datos desde archivo	La estimación también se puede realizar adjuntando un archivo que contiene las columnas de entrada, para el caso en el que se quieren estimar varias resistencias.

Tabla 8 Descripción de casos de uso del Estimador.

Fuente: Autores del proyecto.

3. RESULTADOS

3.1. RESULTADOS INTÉRPRETE

Los resultados obtenidos por el intérprete fueron satisfactorios, en las tablas se observa una abstracción de resultados que muestra el correcto funcionamiento del mismo.

El pequeño error presente en estos resultados se debe a que en JAVA se utiliza un mayor número de decimales comparado con los utilizados en las interpretaciones originalmente realizadas en Excel y que la selección del punto de falla o resistencia se realizaba manualmente con la incertidumbre que este proceso implica.

RESULTADOS LABORATORIO	RESULTADOS INTÉRPRETE	Error
TO (Kpa)	TO (Kpa)	Relativo (%)
5597,75000	5671,03027	1,30910
7384,88000	7379,52295	0,07254
4841,22345	4841,22219	0,00003
6181,23000	6181,23042	0,00001
5325,76	5325,76011	0,00000
6072,25	6060,49488	0,19359
6695,51	6695,51366	0,00005
6198,82	6209,45834	0,17162
2736,35212	2736,35640	0,00016
2774,24094	2774,23745	0,00013
3053,08139	3053,08056	0,00003

RESULTADOS LABORATORIO	RESULTADOS INTÉRPRETE	Error
UCS PSI	UCS PSI	Relativo (%)
15744,3	15744,3134	8,51102E-05
12939,3	12939,33125	0,000241512
9549,98	9549,98069	7,22515E-06
17163,7	17163,6995	2,91312E-06
13963	13963,02185	0,000156485
10457,1	10457,13685	0,000352392
10515	10514,97053	0,000280266
11977,9	11977,92863	0,000239024
14284,4	14284,44271	0,000298998
12750,5	12750,49081	7,20756E-05
13989,64219	13989,64219	2,21447E-08

RESULTADOS LABORATORIO	RESULTADOS INTÉRPRETE	Error
CCS PSI	CCS PSI	Relativo (%)
107407,6	107407,7919	0,000178665
42524,58115	42524,65805	0,000180833
68485,5299	68485,65374	0,000180823
59700	60885,02705	1,984969933

Tabla 9 Tablas de resultados del intérprete.

Fuente: Autores del proyecto.

3.2. RESULTADOS ESTIMADOR

Se presentan a continuación las redes neuronales obtenidas por cada una de las pruebas de laboratorio, las tablas con los resultados estimados mediante las arquitecturas seleccionadas, comparación con resultados reales y los errores presentes en cada uno de estos datos.

Se normalizaron las entradas ya que éstas se encuentran en diferentes unidades y se quiso que la red recibiera entradas sobre una misma escala evitando así el manejo de estas diferencias en el entrenamiento. Las salidas fueron normalizadas también ya que estas en la red neuronal están limitadas al rango de la función de activación. Se utilizó tanto para las entradas como para las salidas el rango de normalización $[-1,1]$ denominado tansig. (Ver figura 46)



Figura 46 Interfaz de carga de datos para la prueba Brazilian Test.

Fuente: Adaptado de MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.

Se utilizó un entrenamiento por épocas ya que al hacerlo mediante detención temprana se encontró que el algoritmo no permitía pasar de un número muy pequeño de épocas ya que la red en sus primeras etapas no detectaba una

generalización con respecto al grupo de validación por lo que detenía el proceso de entrenamiento rápidamente sin permitir que el algoritmo *backpropagation* realizara un correcto ajuste de los pesos. Obteniéndose resultados muy malos ya que eran redes que no completaban el proceso de entrenamiento.

El número de épocas que se utilizó para las 3 pruebas fue de 300, este número se escogió porque a medida que se realizaban pruebas, se determinó que con este valor se lograba un buen entrenamiento sin llegar a sobre entrenar.

En cuanto a los parámetros del algoritmo genético, los valores escogidos fueron los siguientes (Ver figura 47):

Población: 20. En las pruebas realizadas en conjunto con el desarrollo de la tesis *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura*¹³, se encontró que con éste valor se presentaba una buena diversidad en la población y no se sobrecargaba el algoritmo con muchos individuos.

Los valores de probabilidad de mutación y de cruce, máximo número de capas, máximo número de neuronas por capa, número de entrenamientos y algoritmo de entrenamiento fueron los suministrados por defecto en la herramienta, los cuales fueron determinados en las múltiples pruebas realizadas en la tesis *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura*¹³.

Número de generaciones: 40 es un valor que permite al algoritmo realizar múltiples pruebas manejando tiempos de ejecución aceptables que se definieron como no superiores a cuatro horas.

Tolerancia de error: 1.5. Debido a que la tolerancia de error máxima aceptada por el laboratorio de mecánica de rocas es de 20%, se escogieron tolerancias entre 10 y 15% ya que si se presentaban valores fuera del rango, éstos aún podían estar por debajo de la tolerancia máxima.

Figura 47 Carga de parámetros para la Prueba Brazilian.

Fuente: MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. *Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.*2008.

3.2.1. PRUEBA DE LABORATORIO 1. Tensión INDIRECTA Ó BRAZILIAN TEST

La búsqueda y selección de la arquitectura de red con mejores resultados se llevo a cabo sobre el configurador con un set de 38 datos, teniendo dos entradas, porosidad y permeabilidad (ambas propiedades petrofísicas de las muestras de rocas) y una salida, la resistencia a la tensión (Ver tabla 9).

Del set total de datos, se utilizó un 84% para entrenar la red y el restante 16% para pruebas de generalización, la distribución en cada uno de los dos grupos se realizó en varias sesiones de ejecución del configurador observando el ajuste logrado, considerando que los datos que obtenían un error no despreciable en el grupo de prueba, debían incluirse en el entrenamiento por presentar cierta representatividad sobre el set total, además de tener en cuenta que los datos extremos no debían ser incluidos en el grupo de prueba ya que al hacerse esto

quedarían fuera del rango con el que fue entrenada la red y se estaría probando con una extrapolación.

Las configuraciones halladas por el configurador al terminar la ejecución del mismo se presentan en la figura 48 y como se puede observar la red seleccionada está compuesta por 6 capas cada una con 6 neuronas, con ésta se obtiene un error relativo promedio de 2,8906% con un valor extremo en 20.193%, los errores se pueden ver en la figura 50.

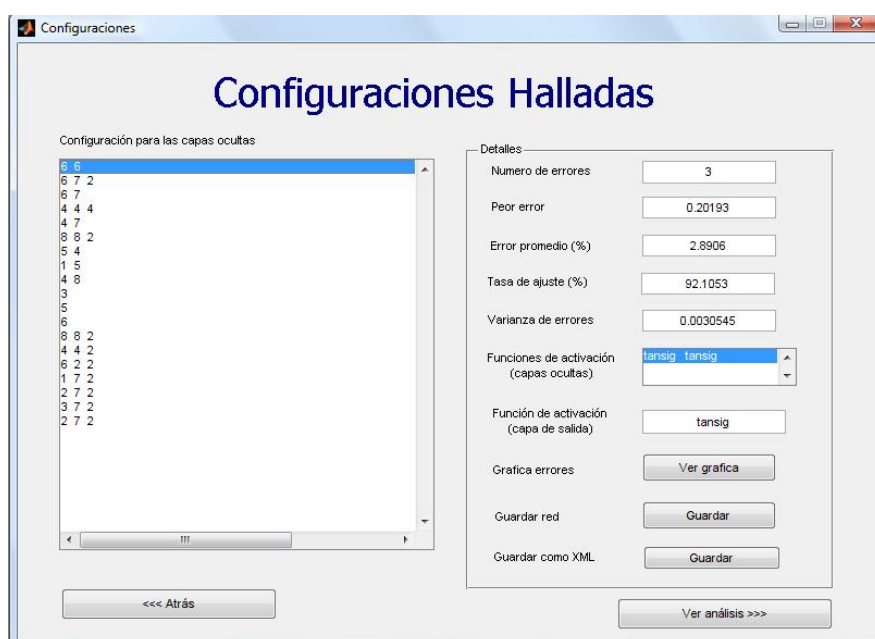


Figura 48 Configuraciones halladas Prueba Brazilian.

Fuente: Adaptado de MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.

En la figura 49 se puede observar la arquitectura de red seleccionada.

Figura 49 Arquitectura de Red Neuronal de la Prueba Brazilian.

Fuente: Adaptado de framework JCortex.

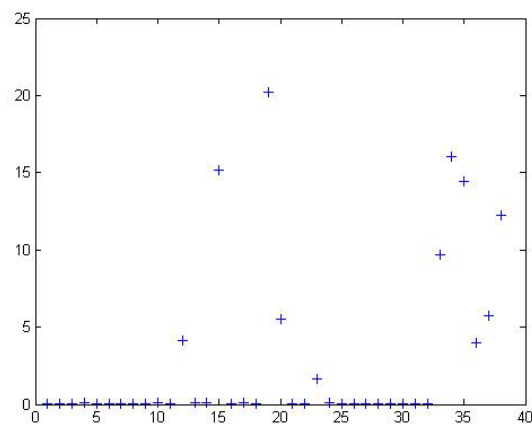


Figura 50 Gráfica de error de Prueba Brazilian.

Fuente: Autores del proyecto.

Las figura 51 muestra gráficas estadísticas que el configurador presenta como ayuda visual, la primera presenta el error promedio absoluto alcanzado utilizando redes con 1,2,3 y 4 capas y la segunda muestra el número de configuraciones probadas con el mismo número de capas.

También se puede observar el número de redes probadas que en este caso fue de 420 evitando probar redes más de una vez ya que el número total de redes fue de 4520 y el tiempo empleado por el configurador que fue de 3,1 horas.

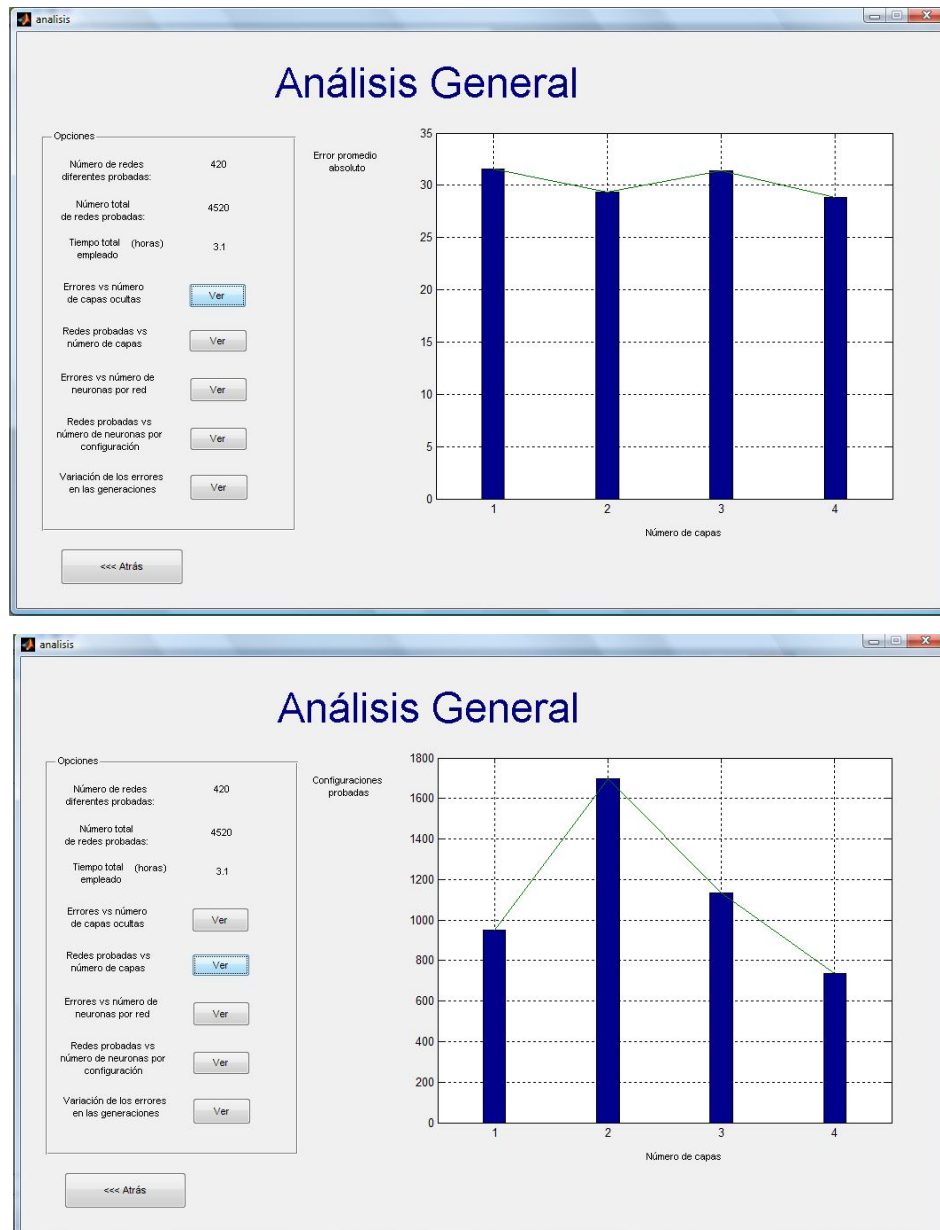


Figura 51 Estadísticas del configurador para Prueba Brazilian.

Fuente: Autores del proyecto.

La siguiente tabla muestra los datos de entrada, los resultados de TO (Resistencia a la tensión) obtenidos por el laboratorio, es decir reales, los datos estimados por la arquitectura de red seleccionada, ejecutada en MATLAB y los errores entre éstos.

Entradas		Valor Real	Valor TO	Error
Porosidad (%)	Permeabilidad (md)	TO Laboratorio (Kpa)	Estimado por Red Neuronal MATLAB (Kpa)	redMATLAB / Valor Real (%)
11,1	4,7	4276,60	4277,0305	0,01007
8,8	1,1	7487,00	7486,0438	0,01277
10,7	2,4	9277,90	9280,5401	0,02846
11,1	3,5	6587,30	6581,9410	0,08135
9,3	1,3	7496,70	7496,3638	0,00448
10,5	2,9	8208,00	8207,6150	0,00469
10,9	2,6	7488,40	7489,8043	0,01875
9,6	1,1	9004,40	9006,0060	0,01784
10,4	3,3	7776,70	7778,3345	0,02102
11,1	3	7103,40	7109,1033	0,08029
10,4	1,1	7589,20	7587,0991	0,02768
8,3	6,78E-01	5406,00	5180,8609	4,16462
8,6	7,23E-01	4173,30	4176,6411	0,08006
10,1	1,1	5597,80	5602,3316	0,08095
8,1	7,02E-01	7384,90	6263,6679	15,18277
9,4	4,40E+00	4841,20	4841,3019	0,00210
9,8	3,00E-01	6181,20	6175,0736	0,09911
9,8	6,99E-01	4476,30	4479,2977	0,06697
8,1	7,24E-01	5325,80	6401,2065	20,19239
8,4	9,78E-01	6072,30	6408,3716	5,53450
10,4	2,3	6198,80	6201,4783	0,04321
10,8	3,2	5898,30	5895,1024	0,05421
8,5	1,3	9192,50	9038,4498	1,67583
8	1,1	7800,10	7792,6201	0,09589
9,6	5,6	5259,70	5259,6030	0,00184
9,8	5,3	3815,00	3815,0119	0,00031

11,7	4,3	7701,00	7699,5764	0,01849
8,6	7,80E-01	6174,80	6174,7570	0,00070
8,7	8,55E-01	6166,90	6164,0512	0,04620
10,7	9,1	4363,40	4364,0056	0,01388
11,2	4,1	8321,60	8322,5662	0,01161
10,8	2,7	12442,00	12438,9598	0,02444
10,4	1,8	6428,90	7051,1552	9,67903
9,9	2,6	4544,30	3815,0129	16,04839
11,1	3,9	6569,10	7517,3857	14,43555
11	3,4	8593,40	8937,9361	4,00931
8,8	1,20E+00	8444,50	8928,0142	5,72579
8,7	6,88E-01	5918,60	6643,5913	12,24937

Tabla 10 Tabla de datos reales y resultados en MATLAB de Prueba Brazilian.

Fuente: Autores del proyecto.

Se puede observar el correcto ajuste que presentó la red neuronal comparado con los resultados del laboratorio.

La red mostró una buena generalización sobre los datos de prueba que se observan resaltados al final de la tabla 10.

La tabla 11 muestra los valores estimados por la arquitectura de red seleccionada, ejecutada en MATLAB y en JAVA y los respectivos errores entre estos.

Valor TO Estimado por Red Neuronal MATLAB (Kpa)	Valor TO Estimado en JAVA (Kpa)	Error redMaTLAB/ redJAVA (%)
4277,00	4276,7046	0,00690671
7486,00	7481,719	0,057186749
9280,50	9279,209	0,013910888

6581,90	6580,8745	0,015580607
7496,40	7489,7397	0,088846646
8207,60	8205,351	0,027401433
7489,80	7487,9956	0,024091431
9006,00	8997,568	0,093626471
7778,30	7775,1064	0,041057815
7109,10	7108,1157	0,013845634
7587,10	7585,5757	0,02009068
5180,90	5179,2573	0,031706846
4176,60	4174,7197	0,045019873
5602,30	5600,7944	0,026874676
6263,70	6261,223	0,039545317
4841,30	4841,289	0,000227212
6175,10	6172,681	0,039173455
4479,30	4478,1606	0,02543701
6401,20	6398,6772	0,03941136
6408,40	6405,7334	0,041611011
6201,50	6199,6284	0,030179795
5895,10	5894,014	0,018422079
9038,40	9035,276	0,03456364
7792,60	7789,567	0,038921541
5259,60	5262,8315	0,061440033
3815,00	3815,012	0,000314548
7699,60	7694,7188	0,063395501
6174,80	6169,026	0,093509102
6164,10	6160,1113	0,064708554
4364,00	4365,8027	0,041308433
8322,60	8319,99	0,031360392
12439,00	12438,958	0,000337648
7051,20	7049,3584	0,02611754

3815,00	3815,013	0,00034076
7517,40	7515,7676	0,021714955
8937,90	8937,053	0,009476499
8928,00	8922,247	0,064437724
6643,60	6640,2285	0,050748088

Tabla 11 *Tabla de resultados de la ejecución de la RNA en MATLAB y en JAVA para la prueba Brazilian.*

Fuente: Autores del proyecto.

Se puede comprobar que la configuración de red neuronal fue exportada correctamente a la herramienta desarrollada en JAVA, ya que su funcionamiento es idéntico, presentándose una pequeña diferencia en la precisión de las cifras decimal es debido a la utilización de diferentes tipos de datos, double en MATLAB y float en JAVA.

3.2.2. PRUEBA DE LABORATORIO 2. COMPRESIÓN UNIAXIAL

La búsqueda y selección de la arquitectura de red con mejores resultados se llevo a cabo sobre el configurador con un set de 38 datos igual que en la primera prueba, teniendo las mismas dos entradas, porosidad y permeabilidad (ambas propiedades petrofísicas de las muestras de roca) y una salida, la resistencia a la compresión (Ver tabla 12).

Del set total de datos, se utilizó igual que en la prueba Brazilian un 84% para entrenar la red y el restante 16% para pruebas de generalización y se utilizó el mismo criterio de selección descrito en la prueba anterior.

En la figura 52 se muestra la arquitectura seleccionada para esta prueba, compuesta por dos capas ocultas de 8 y 5 neuronas respectivamente, se obtuvo un error relativo promedio de 2,2058% con un valor extremo en 18.919% (Ver Figura 53).

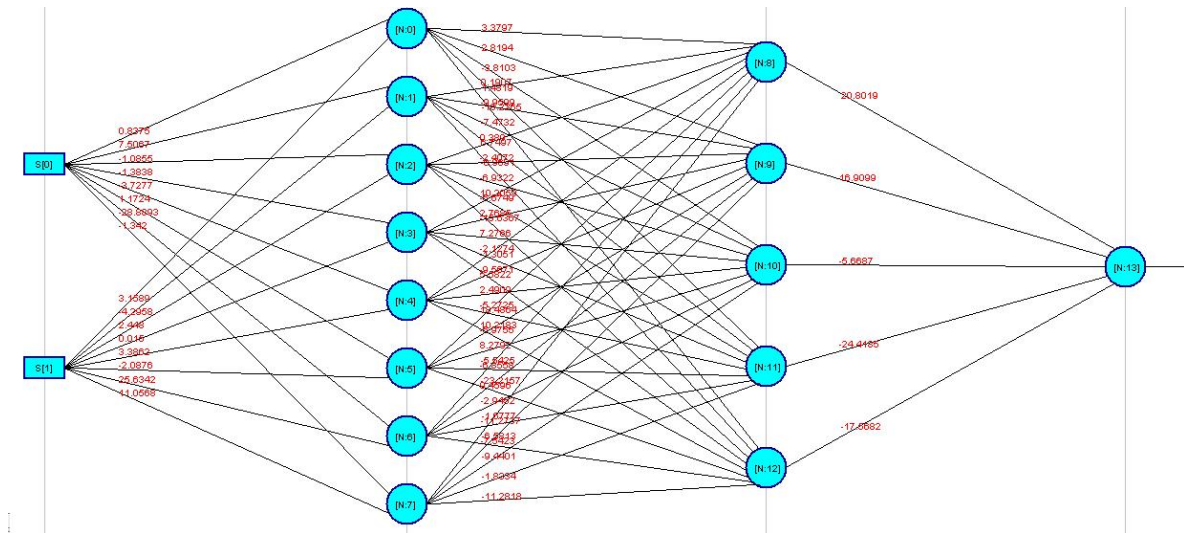


Figura 52 Arquitectura de Red Neuronal de la Prueba Uniaxial.

Fuente: Adaptado de framework JCortex.

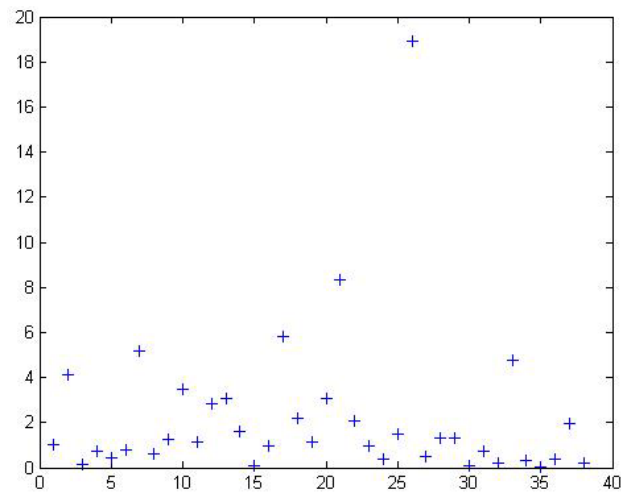


Figura 53 Gráfica de error de Prueba Uniaxial.

Fuente: Autores del proyecto.

Las figura 54 muestra las gráficas estadísticas obtenidas, la primera presenta el error promedio absoluto alcanzado utilizando redes con 1,2,3 y 4 capas y la segunda muestra el número de configuraciones probadas con el mismo número de capas.

También se puede observar el número de redes probadas que en este caso fue de 416 evitando probar redes más de una vez ya que el número total de redes fue de 4520 y el tiempo empleado por el configurador que fue de 3,9 horas.

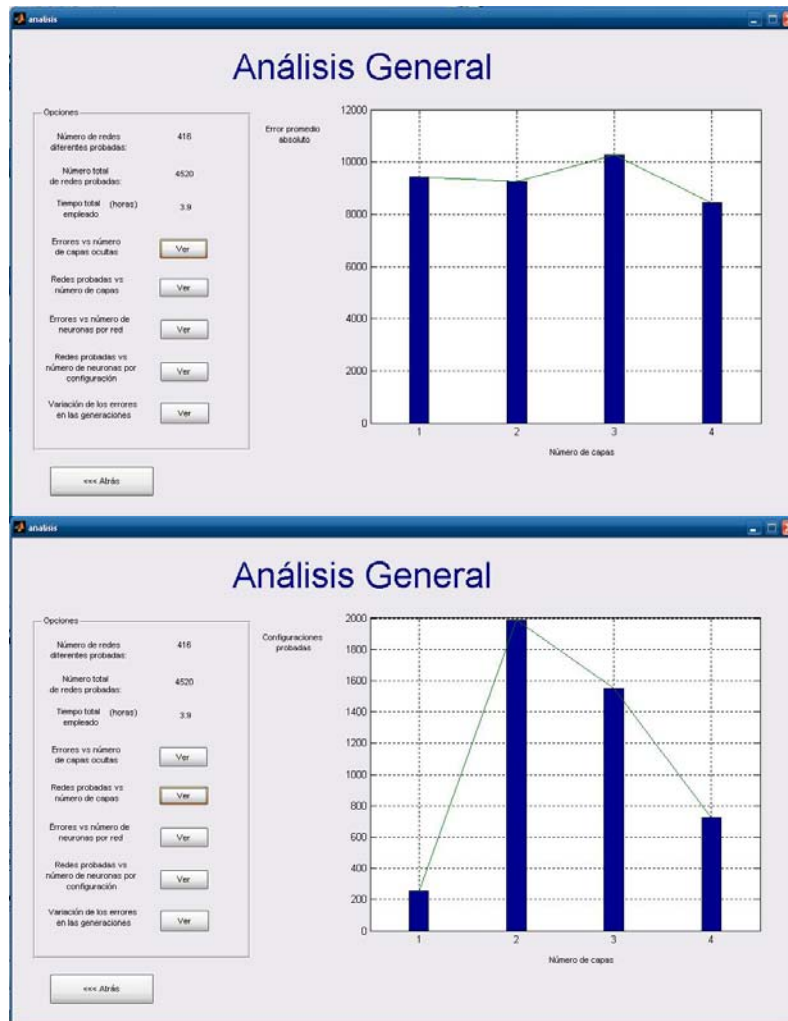


Figura 54 Estadísticas del configurador para Prueba Uniaxial.

Fuente: Autores del proyecto.

La siguiente tabla muestra los datos de entrada, los resultados de UCS (Resistencia a la compresión sin confinamiento) obtenidos por el laboratorio, es decir reales, los datos estimados por la arquitectura de red seleccionada, ejecutada en MATLAB y los errores entre éstos.

Entradas		Valor Real	Valor UCS	Error
Porosidad (%)	Permeabilidad (md)	UCS Laboratorio (PSI)	Estimado por Red Neuronal MATLAB (PSI)	redMATLAB/ Valor Real (%)
5,9	2,6	36207	35829,0502	1,043858386
5,6	0,245	4381,8	4563,3117	4,14240113
2,8	0,029	24858	24899,9385	0,168712321
5,7	2,4	14770	14873,8977	0,703437308
5	1,4	20865	20952,4737	0,41923666
5,3	6,80E-01	31411	31656,6943	0,782191935
9,1	0,228	1282,7	1349,2311	5,186801257
3,8	3,50E-02	25673	25518,9297	0,600125816
3	3,42E-02	31351	31736,3315	1,229088519
3,1	2,50E-02	8610,9	8909,7882	3,471044773
2,5	0,0234	7353,8	7269,9970	1,139587619
3,2	0,0084	7338,2	7129,1475	2,848825416
10,4	2,3	15744	15257,7699	3,088351728
11,1	4,7	9550	9394,7851	1,625286488
8,8	1,1	17164	17148,1846	0,092142815
10,7	2,4	13963	14098,0826	0,967432476
11,1	3,5	10457	11063,8626	5,803409766
11,1	3,9	10515	10286,8456	2,169799462
11,7	4,3	11978	11843,2060	1,12534628
10,9	2,6	14284	13845,8329	3,067538113
11	3,4	12751	11691,1923	8,31156554
11,1	3	3473,2	3544,8111	2,061819697
9,8	0,699	13990	13851,9395	0,986851228
8	1,1	20765	20846,9921	0,394857298
12	1,3	17940	17670,9676	1,499623198
3,3	0,0309	8436,1	10032,1192	18,91892281

3,6	0,0081	20154	20256,7356	0,509752945
5,4	0,0181	16581	16800,7727	1,325448897
8,6	0,187	3694	3741,6875	1,290943704
8,2	2,1	4792,3	4797,7792	0,114332527
10,8	3,2	12939	13037,0538	0,757815583
6,6	6,5	31160	31232,5426	0,23280672
3,3	2,70E-02	26951	25665,8353	4,768523375
4,4	2,10E-01	24003	24085,2234	0,342554671
10,5	6,665	17817	17810,8880	0,034304478
8	1,1	20765	20846,9921	0,394857298
12	4,1	17940	17583,7315	1,985889165
5,4	1,5	25791	25848,6263	0,22343585

Tabla 12 Tabla de datos reales y resultados en MATLAB de Prueba Uniaxial.

Fuente: Autores del proyecto.

Se puede observar igual que en la prueba Brazilian el correcto ajuste que presentó la red neuronal comparado con los resultados del laboratorio.

La red mostró una buena generalización sobre los datos de prueba que se observan resaltados al final de la tabla 12.

La tabla 13 muestra los valores estimados por la arquitectura de red seleccionada, ejecutada en MATLAB y en JAVA y los respectivos errores entre estos.

Valor UCS Estimado por Red Neuronal MATLAB (PSI)	Valor UCS Estimado en JAVA (PSI)	Error redMaTLAB/ redJAVA (%)
35829,0502	35828,707	0,000957866
4563,3117	4563,7017	0,008545708
24899,9385	24896,02	0,015737022

14873,8977	14872,108	0,012032423
20952,4737	20945,236	0,034543554
31656,6943	31635,01	0,068498336
1349,2311	1349,4026	0,012710964
25518,9297	25541,049	0,086678011
31736,3315	31691,91	0,139970625
8909,7882	8892,139	0,198087699
7269,9970	7269,7026	0,004049598
7129,1475	7125,191	0,055497426
15257,7699	15297,024	0,257272828
9394,7851	9399,344	0,048525427
17148,1846	17205,725	0,335548013
14098,0826	14137,581	0,280168619
11063,8626	11081,894	0,162976001
10286,8456	10306,184	0,187991676
11843,2060	11876,903	0,284525806
13845,8329	13881,567	0,258085913
11691,1923	11708,128	0,14485881
3544,8111	3551,464	0,187679345
13851,9395	13891,8545	0,288154498
20846,9921	20888,555	0,199371122
17670,9676	17697,188	0,148381245
10032,1192	10078,318	0,460508409
20256,7356	20274,666	0,0885157
16800,7727	16835,934	0,209283936
3741,6875	3748,542	0,183193804
4797,7792	4798,482	0,014649326
13037,0538	13052,908	0,121609084
31232,5426	31205,84	0,085495998
25665,8353	25724,15	0,227207625

24085,2234	24117,648	0,134624461
17810,8880	17771,352	0,221976418
20846,9921	20888,555	0,199371122
17583,7315	17635,996	0,297232224
25848,6263	25879,031	0,117625825

Tabla 13 Tabla de resultados de la ejecución de la RNA en MATLAB y en JAVA para la prueba Uniaxial.

Fuente: Autores del proyecto.

Al igual que en la prueba anterior se verifica que se hizo una correcta exportación de la red neuronal a la herramienta JAVA.

3.2.3. PRUEBA DE LABORATORIO 3. COMPRESIÓN TRIAXIAL

Para esta tercera prueba se agrega otra entrada además de la porosidad y la permeabilidad que es la presión de confinamiento y como salida se tiene la resistencia a la compresión con confinamiento. La búsqueda y selección de la arquitectura de red con mejores resultados se llevo a cabo sobre el configurador con un set de 67 datos (Ver Tabla 14).

Del set total de datos, se utilizó un 85% para entrenar la red y el restante 15% para pruebas de generalización y se utilizó el mismo criterio de selección descrito en las pruebas anteriores.

Las configuraciones halladas se pueden observar en la figura 55 y se muestran para la primera arquitectura detalles como el peor error, la tasa de ajuste entre otros.

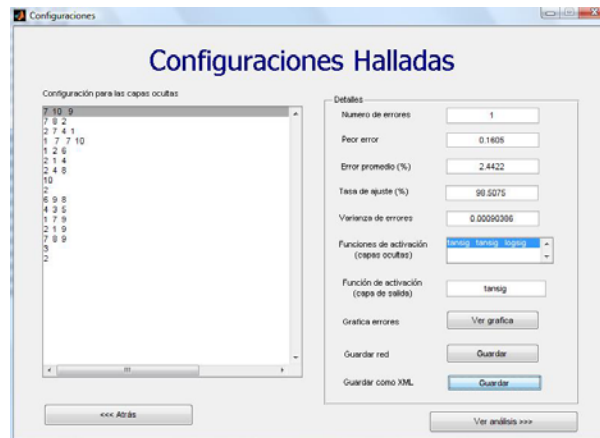


Figura 55 Configuraciones halladas Prueba Triaxial.

Fuente: Adaptado de MENDEZ EDGAR ALBERTO, MARIÑO JUAN SEBASTIAN. Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura.2008.

En la figura 56 se muestra la arquitectura de red seleccionada para esta prueba, compuesta por tres capas ocultas de 7, 10 y 9 neuronas respectivamente, se obtuvo un error relativo promedio de 2.4422% con un valor extremo en 16.05% (Ver Figura 57).

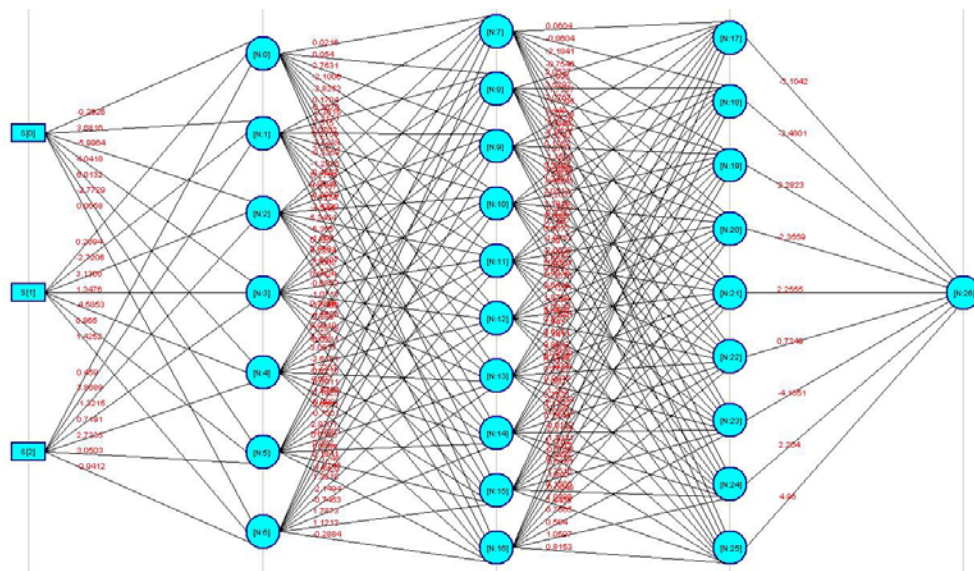


Figura 56 Arquitectura de Red Neuronal de la Prueba Triaxial.

Fuente: Adaptado de framework JCortex.

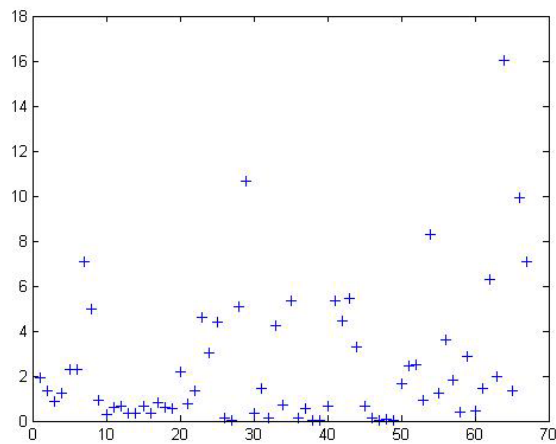
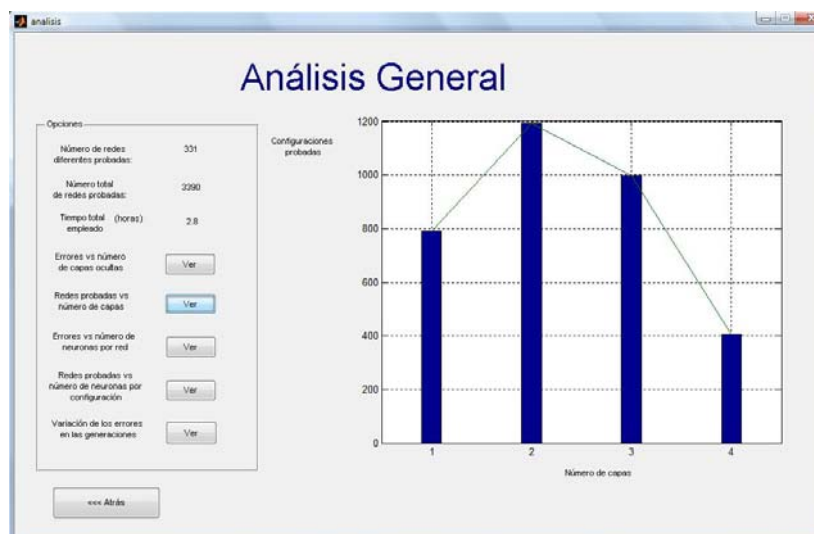


Figura 57 Gráfica de error de Prueba Triaxial.

Fuente: Autores del proyecto.

La figura 58 muestra las gráficas que presentan el error promedio absoluto alcanzado utilizando redes con 1,2,3 y 4 capas y el número de configuraciones probadas igualmente con 1,2,3 y 4 capas.

También se puede observar el número de redes probadas que en este caso fue de 331 evitando probar redes más de una vez ya que el número total de redes fue de 3390 y el tiempo empleado por el configurador que fue de 2,8 horas.



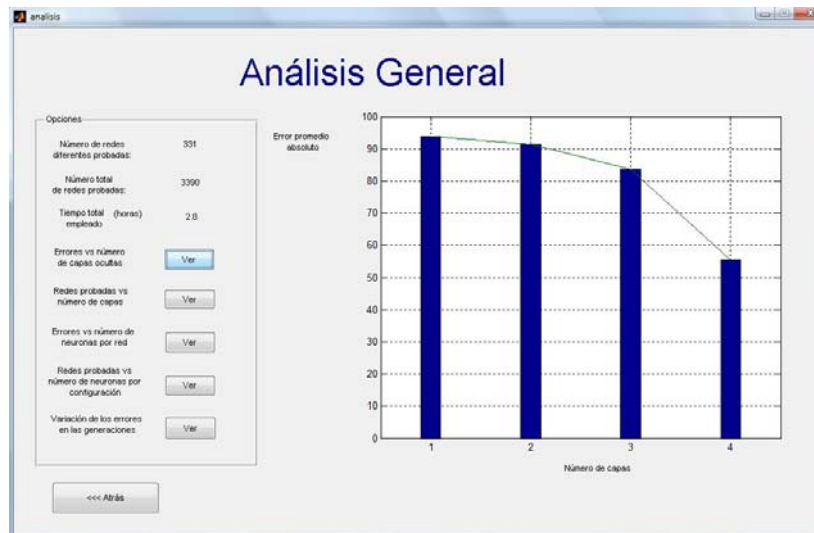


Figura 58 Estadísticas del configurador para Prueba Triaxial.

Fuente: Autores del proyecto.

La tabla 14 muestra los datos de entrada, los resultados de CCS (Resistencia a la compresión con confinamiento) obtenidos por el laboratorio, es decir reales, los datos estimados por la herramienta, resultados obtenidos al estimar con la misma arquitectura de red sobre MATLAB y finalmente los errores tanto con los valores reales como con MATLAB.

Entradas			Valor Real	Valor CCS Estimado por Red Neuronal MATLAB (PSI)	Error redMATLAB / Valor Real (%)
Porosidad (%)	Permeabilidad (md)	Presión de Confinamiento (PSI)	CCS Laboratorio (PSI)		
6,5	7,4	16000	74141	75584	1,946291526
5,9	2,4	16000	59700	58913	1,318257956
2,6	2,21E-02	10000	101660	100790	0,855793823
6,5	7,4	0	31204	31594	1,249839764
2,3	3,30E-03	10000	23006	22474	2,312440233
2,3	3,30E-03	10000	23003	22474	2,299700039

2,3	3,30E-03	20000	10019	10727	7,06657351
1,2	0,0017	10000	23026	24165	4,946582125
8,3	0,6782	2000	42525	42916	0,919459142
10,1	1,1	8000	63165	63338	0,273885855
8,8	1,2	2000	41821	42083	0,62647952
8,8	1,2	4000	55765	56137	0,667085089
8,8	1,2	6000	66005	66218	0,322702826
8,8	1,2	8000	71844	72097	0,352151885
9,8	0,3	4000	45198	45504	0,677021107
9,8	0,3	6000	54139	54316	0,32693622
8,6	7,73E-01	4000	55896	56339	0,792543295
8,1	7,23E-01	8000	71830	72267	0,608380899
9,3	2,9	20	15831	15748	0,52428779
10	4,5	2000	39714	38845	2,188145238
9,8	4,6	4000	54381	53956	0,781522958
10,7	9,1	6000	9734	9867	1,362249458
5,1	1,40E-02	1500	37958	36206	4,615627799
3,3	1,23E-02	3000	53675	55297	3,021891011
3,8	3,50E-02	10	25673	26796	4,374245316
3,8	3,50E-02	5000	65678	65777	0,150735406
3,8	3,50E-02	16350	129160	129170	0,007742335
3	3,42E-02	10	31351	32941	5,071608561
3	3,42E-02	5000	52375	57959	10,66157518
3	3,42E-02	10000	62680	62893	0,339821315
3	3,42E-02	15138	69575	70573	1,434423284
3	3,42E-02	16350	138180	138010	0,123027935
4,8	2,67E-02	5	28417	27217	4,222824366
6,5	7,4	10000	107410	106660	0,698259008
5,5	2,1	5	30157	31767	5,338727327

5,5	2,1	5000	85233	85114	0,139617284
5,5	2,1	10000	116500	117170	0,575107296
5,5	2,1	13534	130300	130240	0,046047583
5,5	2,1	14620	130560	130580	0,015318627
6,1	7,34E-01	2000	65563	65979	0,634504217
6,1	7,34E-01	5000	81733	86085	5,324654668
6,1	7,34E-01	7000	91788	95860	4,436309757
3,6	8,10E-03	20	18550	19558	5,433962264
3,6	8,10E-03	1500	33342	34439	3,290144562
3,6	8,10E-03	4000	50476	50808	0,657738331
3,6	8,10E-03	6000	65761	65832	0,107966728
3,6	8,10E-03	8000	72428	72390	0,052465897
4,6	3,00E-03	20	15899	15914	0,094345556
4,6	3,00E-03	1500	27758	27762	0,01441026
4,6	3,00E-03	3000	32753	33301	1,673129179
4,6	3,00E-03	5000	30591	31335	2,432087869
5,4	1,81E-02	20	22311	21755	2,492044283
5,4	1,81E-02	1500	32040	31741	0,933208489
5,4	1,81E-02	4000	43114	46688	8,289650694
5,4	1,81E-02	8000	54853	55539	1,250615281
9,2	3	20	15776	15205	3,619421907
10,3	6	4000	51915	50970	1,820283155
10,8	7,6	20	11444	11486	0,367004544
4,8	2,67E-02	5000	72040	69969	2,874791782
6,5	6,4	16000	69500	69820	0,460431655
3,8	3,50E-02	10000	104460	102970	1,426383305
8,1	7,02E-01	6000	68486	64162	6,31369915
9,8	0,3	8000	61337	62561	1,995532876
5,4	1,81E-02	6000	52143	60512	16,05009301

3,8	3,50E-02	15138	128470	130200	1,346617887
9,8	0,3	2000	33692	37034	9,919268669
2,3	3,30E-03	20000	10020	10727	7,055888224

Tabla 14 Tabla de datos reales y resultados en MATLAB de Prueba Triaxial.

Fuente: Autores del proyecto.

Valor CCS Estimado en MATLAB (PSI)	Valor CCS Estimado en JAVA (PSI)	Error redMaTLAB/ redJAVA (%)
75584	75584	0
58913	58913	0
100790	100790	0
31594	31594	0
22474	22474	0
22474	22474	0
10727	10727	0
24165	24165	0
42916	42916	0
63338	63338	0
42083	42083	0
56137	56137	0
66218	66218	0
72097	72097	0
45504	45504	0
54316	54316	0
56339	56339	0
72267	72267	0
15748	15748	0
38845	38845	0

53956	53956	0
9867	9867	0
36206	36206	0
55297	55297	0
26796	26796	0
65777	65777	0
129170	129170	0
32941	32941	0
57959	57959	0
62893	62893	0
70573	70573	0
138010	138010	0
27217	27217	0
106660	106660	0
31767	31767	0
85114	85114	0
117170	117170	0
130240	130240	0
130580	130580	0
65979	65979	0
86085	86085	0
95860	95860	0
19558	19558	0
34439	34439	0
50808	50808	0
65832	65832	0
72390	72390	0
15914	15914	0
27762	27762	0

33301	33301	0
31335	31335	0
21755	21755	0
31741	31741	0
46688	46688	0
55539	55539	0
15205	15205	0
50970	50970	0
11486	11486	0
69969	69969	0
69820	69820	0
102970	102970	0
64162	64162	0
62561	62561	0
60512	60512	0
130200	130200	0
37034	37034	0
10727	10727	0

Tabla 15 *Tabla de resultados de la ejecución de la RNA en MATLAB y en JAVA para la prueba Triaxial.*

Fuente: Autores del proyecto.

Se observa en la tabla 15, la correcta exportación de la configuración de red neuronal al sistema JAVA. En los valores de CCS del total de datos recopilado en el laboratorio de mecánica de rocas, no se manejan valores decimales, por lo que la diferencia que se presenta en los otros tipos de pruebas, debida a los distintos tipos de datos utilizados, no se presenta.

CONCLUSIONES Y RECOMENDACIONES

- ✓ Se desarrolló una herramienta capaz de interpretar de forma rápida pruebas de resistencia a la tensión, compresión Uniaxial y compresión Triaxial, asiste el proceso de generación de reportes de resultados que se elaboran en el laboratorio de Mecánica de Rocas del Instituto Colombiano del Petróleo y adicionalmente permite almacenar de forma ordenada información de estas pruebas para su posterior consulta.
- ✓ Se participó en el desarrollo del sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste de parámetros y variación de arquitectura Wötán Genetics, desarrollado como proyecto de grado de la Escuela de Ingeniería de Sistemas e Informática, posteriormente se adaptaron y mejoraron algunas de sus partes para ajustarla a los requerimientos del proyecto. Con esta herramienta se obtuvieron las arquitecturas de red capaces de estimar propiedades mecánicas a partir de propiedades petrofísicas de arenas limpias.
- ✓ Se desarrolló una herramienta en JAVA que estima valores de propiedades mecánicas a partir de datos petrofísicos utilizando las configuraciones de red encontradas con el configurador desarrollado en MATLAB que fueron seleccionadas por presentar los resultados más acertados entre todas las configuraciones probadas.
- ✓ Utilizando el software *Wötán Genetics* se probaron en promedio 331.6 diferentes configuraciones de red neuronal por tipo de prueba, buscando las que más se ajustaran al conjunto de datos y las escogidas presentaron errores promedio de:

Brazilian Test: 2.8906% con un valor extremo en 20.193%.

Compresión Uniaxial: 2.2058% con un valor extremo en 18.919%.

Compresión Triaxial: 2.4422% con un valor extremo en 16.05%.

La tolerancia de error aceptada por el Laboratorio de Mecánica de Rocas es de 20%, valor obtenido a través de la experiencia en la manipulación de éste tipo de datos, por lo que los resultados cumplen con las expectativas.

- ✓ Se lograron buenas generalizaciones teniendo en cuenta que el set de datos para las tres pruebas no era grande y, a medida que el número de datos aumente gracias a la forma como se desarrolló la herramienta, se van a obtener mejores resultados.
- ✓ Se logró exportar correctamente la arquitectura de red neuronal de MATLAB a JAVA facilitando la utilización de los modelos de red neuronal pues el único requisito para su uso es la máquina virtual de java versión 1.6.
- ✓ El intérprete como medio recopilador de datos es de vital importancia pues su correcto uso en el laboratorio garantizará un almacenamiento completo y ordenado de las pruebas practicadas que permitirá generar modelos de red más exactos y confiables.
- ✓ El sistema está diseñado para soportar la generación y uso de cualquier modelo de red neuronal por lo que la inclusión de nuevos tipos de prueba requeriría adaptar solo el intérprete de datos para que almacene los datos correspondientes a ese nuevo tipo de prueba.

BIBLIOGRAFÍA

- [1] **X. Chen, C:P Tan and C.M Haberfield** “A Comprehensive Practical Approach For Wellbore Instability Management” SPE 48898. 1998.
- [2] **Torres, M.E., Gonzalez, A.J., Last, N.C.** “In-Situ Stress State Eastern Cordillera Colombia” SPE 81074 Colombia National University -Schlumberger DCS Bogotá, Colombia National University, BP Exploration 2003.
- [3] **GÓMEZ VANEGAS César Augusto y SANTAMARÍA BOLIVAR Rafael.** ANÁLISIS DE ESTABILIDAD DE POZO UTILIZANDO EL SOFTWARE PBORE. Colombia Tesis 2004.
- [4] **JAEGER J.C., and COOK N. G. W.** Fundamentals of Rock Mechanics, 1969.
- [5] **CABRERA J. R., and DE GENNARO S.** Aplicaciones Prácticas de la Geomecánica en la Industria Petrolera y Gasífera, ICP, Diciembre 2003.
- [6] **FJAER E., and RASMUS R.** Petroleum Related Rock Mechanics, 1996.
- [7] **CORZO RUEDA Reinel y RINCÓN PABÓN Claudio Omar.** MEDICIÓN Y EVALUACIÓN DE LA MAGNITUD Y DIRECCIÓN DE LOS ESFUERZOS IN-SITU EN CAMPO. Colombia Tesis 2004.
- [8] **CORONEL Iván Leonardo y TARAZONA Darwin Mateus.** EVALUACIÓN DE LOS MECANISMOS DE FALLA QUE CONDUCEN A LA INESTABILIDAD DE POZO. Colombia Tesis 2004.
- [9] **Dr. Luis Alonso Romero, Dr. Teodoro Calonge Cano.** Capitulo 1. Redes Neuronales y Reconocimiento de patrones. –España.
<http://lisisu02.usal.es/~airene/capit1.pfd>
- [10] **S. Watanabe.** Pattern Recognition: Human and Mechanical. –Wiley, New York 1985.
- [11] **Leiva Lucas, Acosta Nelson, Vázquez Martín.** Herramienta para diseño de arquitecturas a medida basadas en redes neuronales para reconocimiento de patrones visuales. -Argentina
- [12] **V.K. Singh, D. Singh, T.N. Singh.** ‘Prediction of strength properties of some schistose rocks from petrographic properties using artificial neural networks’. Department of Mining Engineering, Institute of Technology, Banaras Hindu

University, Varanasi, 221005 India. Department of Electrical Engineering, Institute of Technology, Banaras Hindu University, Varanasi, 221005 India. 28 November 2000

[13] **MENDEZ Edgar Alberto, MARIÑO Juan Sebastian.** Sistema automático de entrenamiento de redes neuronales artificiales basado en el ajuste genético de parámetros y variación de arquitectura. 2008.

[14] **Xin** (1999) Evolving Artificial Neural Networks. School of Computer Science. The University of Birmingham. B15 2TT.

[15] **John R. Koza** (Stanford University) Genetic Programming IV: Routine Human-Competitive Machine Intelligence, 2003.

[16] **Malanie Mitchell** (1996) An Introduction to Genetic Algorithm.

[17] **North-Holland** (1992) Tree Automata and Languages.

ANEXOS

Anexo 1 BASE DE DATOS IMPLEMENTADA

El objetivo de la base de datos es almacenar la información que se va generando a medida que se van interpretando pruebas en el laboratorio utilizando el software desarrollado para este fin. Esta información es consultada a través del modulo web y es utilizada para realizar los entrenamientos de las redes neuronales en el software de configuración de redes neuronales *Wötan Genetics*. Otro de los objetivos de la base de datos es almacenar la información de las redes neuronales seleccionadas en el configurador, para que el software de estimación pueda descargar y utilizar correctamente esta información. Por último, en la base de datos se almacena la información de los usuarios que tienen acceso a las interfaces del módulo web que permiten realizar acciones en la base de datos.

La base de datos está compuesta por las siguientes tablas:

tipouser: Contiene los tipos de usuarios que pueden ser creados por el administrador del módulo web.

- idTipoUser
- NombreTipo
- DescripcionTipo

usuario: Contiene los usuarios creados que tienen acceso a la descarga y otras funciones de acuerdo a los permisos otorgados por el administrador.

- Idusuario
- idTipoUser
- Nombres
- Otros campos de información personal del usuario.

tipodatos: En esta tabla se almacenan todos los tipos de datos con los que se trabaja en el sistema sus atributos son:

- idTipoDatos: identificador único del tipo de dato.
- NombreDato: Nombre que se le asigna a este tipo de dato.
- DescripDato: Esta es una descripción que se hace del tipo de dato, para poder identificarlo fácilmente.
- Unidades: Unidades en las que está este tipo de dato.
- Petrofisico: Este es un campo de tipo enum, que indica si un dato es petrofísico o no.

archivodat: Existe un registro en esta tabla por cada uno de los archivos de extensión dat que son interpretados, en la tabla se almacena la información general de archivo.

- idArchivoDat: identificador único del registro.
- NombreArch: Nombre que se le dio al archivo en el sistema MTS815.
- FechaArch: Fecha en la que se generó el archivo.
- HoraArch: Hora de generación de archivo.
- AgeArch: Nombre específico del espécimen (muestra de roca).
- AdditionalInfoArch: Información adicional que se pudo haber agregado en el archivo.

specimencharacteristics: En esta tabla se almacena la información que se obtiene del encabezado del archivo de extensión dat que se genera al realizar una prueba en el sistema MTS815 del laboratorio. Cada espécimen (muestra de roca) posee varios datos.

- idSpecimenCharacteristics: Identificador único del registro.
- idTipoDatos: Identificador del tipo de dato que se va a almacenar en el registro
- idArchivoDat: Identificador del archivo dat al que pertenece este registro.
- DatoSpeChar: Valor del dato que se almacena.

tipoprueba: En esta tabla se almacenan los diferentes tipos de pruebas que se trabajan en el sistema.

- idTipoPrueba: Identificador único del registro.
- NombreTipoPrueba: Nombre de la prueba.
- DescripTipoPrueba: Breve descripción de la prueba.

prueba: En esta tabla se almacena la información correspondiente a una interpretación.

- idPrueba: Identificador único
- idusuario: identificador del usuario que realizo la interpretación.
- idArchivoDat: Identificador del archivo dat que se interpretó.
- idTipoPrueba: Identificador del tipo de prueba.
- IdMuestra: Identificador de la muestra en el laboratorio.
- NombrePozo: Nombre del pozo del que se tomó la muestra.
- Profundidad: Profundidad a la que se tomó la muestra.

datospetrofisicos: En esta tabla se almacena los datos petrofísicos relacionados con una prueba.

- idDatosPetrofisicos: Identificador único
- idPrueba: Identificador de la prueba a la que pertenece el petrofísico.
- idTipoDatos: Identificador del tipo de dato.
- DatoPetrofisico: Valor del dato.

datosinterpretacion: En esta tabla se almacena la información generada después de una interpretación de un archivo de extensión dat.

- idDatosInterpretacion: Identificador único.
- idPrueba: Identificador de la prueba a la que pertenece el dato.
- idTipoDatos: Identificador del tipo de dato.
- DatoInterp: Valor del dato.

reporte: En esta tabla se almacena la información del reporte de resultados generado en el laboratorio.

- idReporte: Identificador único.
- idPrueba: Identificador de la prueba a la que pertenece la información del reporte.
- Numreporte: Numero asignado al reporte en el laboratorio.
- Otros campos con la información general del reporte.

redneuronal: En esta tabla se almacena la información de una red neuronal cargada en el sistema.

- Id_red: Identificador único.
- Idusuario: Identificador del usuario que cargo la red en el sistema.
- idTipoPrueba: Identificador del tipo de prueba a la que pertenece la red.
- NombreRed; Nombre de la red.
- Ubicacion_XML: Ubicación en el servidor del archivo de extensión XML que contiene la configuración de la red.
- FechaCrea: Fecha en la que se creó la red.
- PorcentageAcierto: Porcentaje de acierto de la red.
- NumDEntre: Numero de datos que se utilizaron en el entrenamiento de esta red.

funcionactivacion: Contiene los registros de los diferentes tipos de función de activación que puede tenerse en la conexión entre capas de una neurona.

- idFuncionActivacion: Identificador único.
- NombreFA; nombre de la función d activación.

capa: En esta tabla se almacena información básica de las capas de una red neuronal.

- id_Capa: Identificador único de la capa.
- Id_red: Identificador de la red a la que pertenece la capa.

- idFuncionActivacion: Identificador de la función de activación de la capa.
- Pos_Capa: Numero de la posición de la capa en la red neuronal.
- Num_Neuronas: Numero de neuronas de la capa.

entra_salid: Esta tabla almacena la información respectiva de cada entrada y salida de una red neuronal, esta información es vital para la correcta des-normalización de las salidas de la red neuronal.

- idEntrada: Identificador único.
- Id_red: Identificador de la red a la que pertenece el registro.
- idTipoDatos : Identificador del tipo de dato.
- Tipo: Dato tipo enum, que indica si el dato es de entrada o salida.
- Posición: Posición de la entrada o salida en la red neuronal.
- Vmin: Valor mínimo utilizado en la normalización de la información.
- Vmax: Valor maximo de utilizado en la normalización de la información.
- TNomalizacion: Tipo de normalización utilizada en los datos de esta entrada o salida.

UltimaActualizacionSistema: Esta tabla contiene dos registros, que almacenan la fecha y hora de la última actualización del sistema, el primer registro para el estimador y el segundo para el intérprete. Esta fecha es actualizada por el modulo web en el momento que se utilizan su interfaz para alterar algún dato critico del sistema. Esta fecha le indica al estimador o al intérprete si deben cargar la información del servidor o pueden leer su archivo de configuración local.

- idActualizacionSis: Identificador único.
- Fecha: Fecha y hora de la actualización.

Anexo 2 ARTÍCULO SELECCIONADO PARA PONENCIA TIPO POSTER EN XIII SIMPOSIO DE TRATAMIENTO DE SEÑALES, IMÁGENES Y VISIÓN ARTIFICIAL

Desarrollo de una herramienta software utilizando redes neuronales artificiales para estimar el comportamiento mecánico de arenas limpias pertenecientes a una cuenca Colombiana

Abstract—En la industria del petróleo, es importante el estudio de las rocas que se extraen de determinada zona para obtener la información necesaria en la toma de decisiones a la hora de realizar trabajos de exploración y explotación.

Para encontrar los valores de resistencia a la tensión, UCS (Unconfined Compressive Strength) y CCS (Confined Compressive Strength), que indican el comportamiento mecánico de la roca, es necesario realizar una prueba en un laboratorio que puede durar varias horas o hasta días dependiendo del tipo de prueba. Sin llevar a cabo un ensayo sobre una muestra de roca, no se pueden obtener aproximaciones de los valores anteriormente mencionados.

En este trabajo se presenta el desarrollo de una herramienta computacional que soluciona el problema de la estimación de los valores de resistencia mediante una arquitectura de red neuronal que utiliza para su entrenamiento datos de petrofísica como datos de entrada y, como datos de salida, valores de resistencia obtenidos de pruebas ya realizadas sobre rocas en laboratorio.

Además, también se presenta un algoritmo genético diseñado para buscar la configuración de la red a utilizar, que realiza una búsqueda y optimización de configuraciones, teniendo como objetivo disminuir el valor de error presente en la red neuronal.

Palabras Clave— *Redes neuronales artificiales, Petrofísica, Arenas limpias.*

I. INTRODUCCIÓN

Actualmente no es posible obtener aproximaciones (resistencia a la tensión, UCS, CCS) para lograr caracterizar el comportamiento mecánico de una roca antes de llevar a cabo un ensayo sobre la muestra.

Mediante el conocimiento de la teoría elástica lineal y con base en los datos y resultados de los diferentes ensayos junto con un tratamiento especial a los mismos para obtener valores más precisos y representativos se

puede crear una herramienta que simule el comportamiento mecánico de las rocas, así, la solución que se propone es la creación de un software que simule una prueba de laboratorio y arroje datos de UCS, CCS, módulos de Young y relación de Poisson.

La información obtenida mediante la herramienta desarrollada les permite a los ingenieros encargados del laboratorio diseñar ensayos no destructivos con cierto grado de confianza. En estos casos, se puede evitar el fracturamiento de la muestra, dando lugar a una reducción de costos. Adicionalmente permite simular y aumentar el volumen de datos de un pozo, brindando más información a los profesionales de la industria del petróleo que trabajan en modelamiento mecánico para estabilidad de pozo y fracturamiento hidráulico, esto representándose en modelos más robustos, es decir, más cercanos a la realidad.

II. MARCO TEÓRICO

Cuando una muestra de roca sólida es sometida a grandes esfuerzos lo más seguro es que una falla ocurrirá; esa falla dependerá del estado de esfuerzos, del tipo del material y de la geometría de la muestra. También, la historia de los esfuerzos del espécimen puede ser de gran importancia. Una falla debido a fatiga puede ocurrir muy por debajo del nivel de esfuerzos al cual un espécimen fresco falla. Las técnicas sugeridas para determinar la deformabilidad, resistencia y las constantes elásticas son dadas por la ISRM (Sociedad Internacional de Mecánica de Rocas) [1].

Los datos de corazonas a partir de pruebas de campo y registros son aplicados para evaluar el comportamiento mecánico de la roca in-situ. Es así como los corazonos nos proporcionan la posibilidad de las mediciones directas de los parámetros de resistencia y las propiedades elásticas estáticas.

La humedad puede tener un significativo efecto sobre la deformación de la muestra. Las condiciones de humedad in situ podrían por consiguiente ser preservadas y cambiadas (perdida en humedad) pudiendo ser registradas antes del experimento.

Un número variado de pruebas de laboratorio se han desarrollado para determinar la resistencia de las formaciones de rocas. El tipo de prueba seleccionada para medir una característica particular de la roca deberá simular lo más cerca posible las condiciones encontradas en el yacimiento.

Ninguna técnica actual es capaz de medir in-situ la resistencia de una roca. Por lo tanto pruebas de laboratorio estándar basadas en corazonos pueden proporcionar esos datos pero no reflejar exactamente las propiedades in situ a gran escala, porque estas son influenciadas por diaclasas, fallas, inhomogeneidades, planos de debilidad y otros factores.

Las pruebas de laboratorio usualmente consisten en experimentos simples apropiados para la naturaleza de la roca, en el cual importantes cantidades, a menudo esfuerzos y deformaciones, son determinados.

Estas cantidades son luego expresadas en una forma matemática idealizada, sobre la base de que estas pueden ser extendidas para cubrir situaciones más complicadas que aquellas de los experimentos [1].

Así, gracias al estudio de la petrofísica, que es el estudio de las propiedades físicas y químicas que describen la incidencia y el comportamiento de las rocas, los sólidos y los fluidos, es posible caracterizar por ejemplo un depósito de petróleo o de gas, para lo que se realizan mediciones tales como resistividad, neutrones y densidad, a partir de las cuales se pueden cuantificar la permeabilidad, las saturaciones y la porosidad efectiva [2].

III. PROPIEDADES PETROFÍSICAS COMO PARÁMETROS DE ENTRADA

En el proyecto para la estimación de resultados de pruebas de laboratorio, se hace necesaria la definición

de las entradas y las salidas de manera que entre éstas se encuentre la relación que permita generalizar el proceso de obtención de resultados y para esto se han definido las siguientes entradas y salidas:

A. Parámetros de entrada

Como la petrofísica puede describir el comportamiento de las rocas, las variables de entrada que se han definido son propiedades de éste tipo, y se describen a continuación:

Porosidad (%)

Es el volumen poroso por unidad de volumen de una formación. La porosidad de una formación puede variar considerablemente. Los carbonatos densos (piedra caliza) por ejemplo y las evaporitas pueden tener una porosidad de cero para todos los fines prácticos. Por otra parte, las areniscas pueden tener del 10% al 15% de porosidad, mientras que las areniscas no consolidadas pueden tener una porosidad del 30% o más. Finalmente, los esquistos y las arcillas pueden tener una alta porosidad del 40%, pero los poros individuales generalmente son tan pequeños que la roca es impermeable para que el fluido pueda fluir.

Permeabilidad (md)

Es la medición de la capacidad de una roca para transmitir fluidos. Se dice que un material es permeable si deja pasar a través de él una cantidad apreciable de fluido en un tiempo dado, e impermeable si la cantidad de fluido es despreciable.

La velocidad con la que el fluido atraviesa el material depende de tres factores básicos:

- La porosidad del material.
- La densidad del fluido considerado, afectada por su temperatura.
- La presión a que está sometido el fluido.

Para ser permeable un material debe ser poroso, es decir, debe contener espacios vacíos o poros que le permitan absorber fluido. A su vez tales deben estar interconectados para que el fluido disponga de caminos a través del material, por lo tanto, hay una relación general entre la porosidad y la permeabilidad.

B. Datos de salida

Las características geomecánicas a obtener mediante la simulación de las pruebas son las siguientes respectivamente por pruebas.

Prueba Brazilian Test

- Resistencia Tensil
- Carga

Prueba de Compresión Uniaxial

- UCS

Prueba de Compresión Triaxial

- CCS

IV. IMPLEMENTACIÓN

Para la creación del algoritmo que crea una red neuronal y la entrena, es necesario tener presente que depende de la forma como esté configurada la red, como se entrene la red y como se dispongan los datos que la alimentan, debido a este último punto, es importante el manejo de datos con los que se cuentan, porque de éstos tres factores depende la calidad del algoritmo creado y el éxito o no de los resultados obtenidos.

A. Arquitectura de la red

Se utilizó una red perceptrón multicapa por su gran capacidad de resolver problemas basados en patrones.

En cuanto al diseño de la red, se tomó un set de datos de 32 pruebas de las cuales se seleccionaron 22 pruebas para entrenamiento, 10 pruebas del total de pruebas para la validación que es aquella que se realiza durante el entrenamiento y 5 de los 10 de validación para realizar pruebas después de haber realizado el entrenamiento.

B. Configuración de la red

Para hallar una configuración de red que no sólo logre generalizar sino que teniendo en cuenta que la cantidad de datos es pequeña, arroje buenos resultados, entendiendo como buenos resultados, valores con un porcentaje de error de 15 % comparado con los resultados reales, se creó una aplicación que mediante algoritmos genéticos logre analizar gran cantidad de configuraciones y muestre aquellas que presentan mejores resultados.

V. PRUEBAS Y RESULTADOS CON EL SET DE DATOS DISPONIBLE

Teniendo en cuenta que el volumen de datos con el que se cuenta para el entrenamiento, validación y pruebas de la red neuronal, es bastante bajo, el análisis de éstos

ocupa la principal preocupación con respecto al proyecto debido a que si los datos existentes no logran entrenar una red que arroje resultados buenos, prácticamente, el proyecto pierde validez.

Debido a lo anterior se ha venido trabajando con los datos existentes de pruebas Brazilian Test, para los que se han realizado distintas pruebas, logrando disminuir en gran medida los errores inicialmente encontrados.

Las pruebas realizadas con el set de datos fueron las siguientes:

Se tabularon los datos y en cuanto al entrenamiento, se creó una pequeña función que buscaba buenas configuraciones de red (la que menor error presente) entre varias que se prueban cambiando número de capas, número de neuronas por capas y funciones de transferencia (modelos matemáticos que siguen las neuronas para enviar los datos que reciben de las neuronas de la capa anterior a la siguiente neurona), al poner a correr esta función que duraba alrededor de 20 minutos, se encontraron diferentes configuraciones, algunas se muestran en la tabla 1.

Configuración	1	2	3	4
Número de capas ocultas	2	2	2	2
Neuronas primera capa	9	7	7	8
Neuronas segunda capa	3	3	1	1
Funciones de transferencia	Tansig g	Tansig	Tansig	Tansig

Tabla I. Configuraciones a probar inicialmente.

Los resultados obtenidos con las cuatro configuraciones anteriores no fueron muy buenos, pues se presentaron errores del 40, 30, 40 y 42%, lo que llevó a buscar otra forma de manejar los datos.

Se pensó entonces en modificar el número de datos utilizados en la prueba, teniendo como base la distribución tomada inicialmente, en la que se tenían un 70% del total de datos para entrenamiento y un 30% para pruebas, así, se inicio tomando 22 datos para entrenamiento y 10 para pruebas y se empezaron a aumentar los datos de entrenamiento y por lo tanto a disminuir los de pruebas, se varió el número de pruebas primero de 22 a 24 y luego de de 24 a 26 y se realizaron 20 corridas por cambio teniendo una misma configuración de red neuronal, los resultados no fueron alentadores, pues se presentaron errores del 42, 44 y 38%, así que se descartaron las nuevas distribuciones de datos.

Luego, se deseaba observar el comportamiento de la red variando el orden de los datos con los que se entrena. En este momento se consideró que la mejor opción era tratar de llevar los datos con los que se probaría la red, a una zona central, para que los datos del entrenamiento los representen de alguna manera. Esto se hizo implementando una secuencia de código que se encargó de este ordenamiento y selección.

Primero se probó ordenando los datos de porosidad de mayor a menor (Fig.1), este ordenamiento afecta a todo el conjunto de datos y la correspondencia entre ellos no se pierde. Los datos con los que se prueba la red son escogidos de la parte central del nuevo orden que se forma. Esto se hace pues se considera que el conjunto central puede ser representado o mantener un patrón con los demás que se encuentran en los extremos.

En el nuevo orden se observa la línea roja representando los datos de porosidad, la verde los de permeabilidad y los azules sus respectivas respuestas.

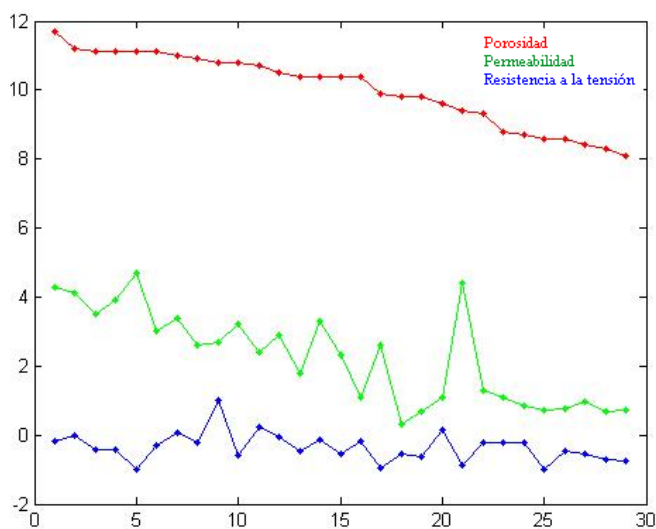


Fig. 1. Datos ordenados por porosidad en forma descendiente.

Después de entrenar la red con este nuevo orden, se observa una variación en el promedio de error, ya que se redujo al 28.5083%, el promedio resulto afectado por un dato que resulto con un 100% de error. El resultado que obtuvo un error del 100%, debe ser movido al conjunto de entrenamiento, ya que no está siendo representado de ninguna forma por este conjunto. Este cambio aportaría más información al entrenamiento pues sus características parecen ser diferentes al conjunto en el cual se encuentra.

Se realizó una nueva prueba, con los datos de permeabilidad ordenados en orden descendiente (Fig.2), se observan unos resultados con un error más bajo, el promedio de error disminuyó al 31,7557% y quitando el peor de los resultados se puede hablar de un 26,3850% de error.

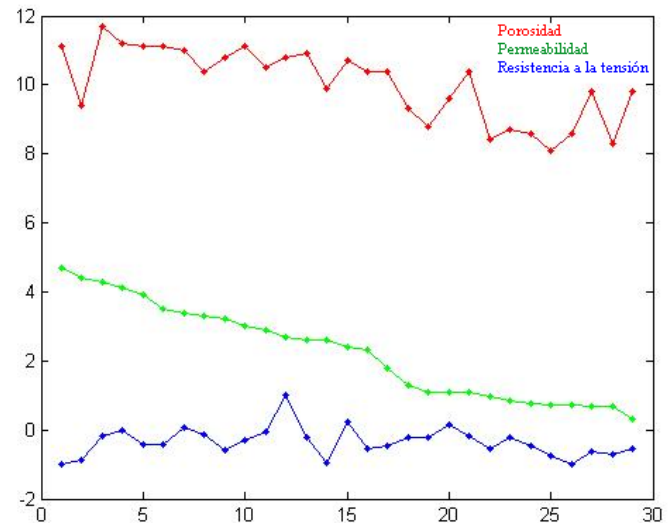


Fig. 2. Datos ordenados por permeabilidad en forma descendiente.

Por último se ordenaron los datos de salida (Fig. 3). En esta grafica se pueden observar los datos de salida en un descenso constante, mientras que los datos de entrada presentan una gran variación.

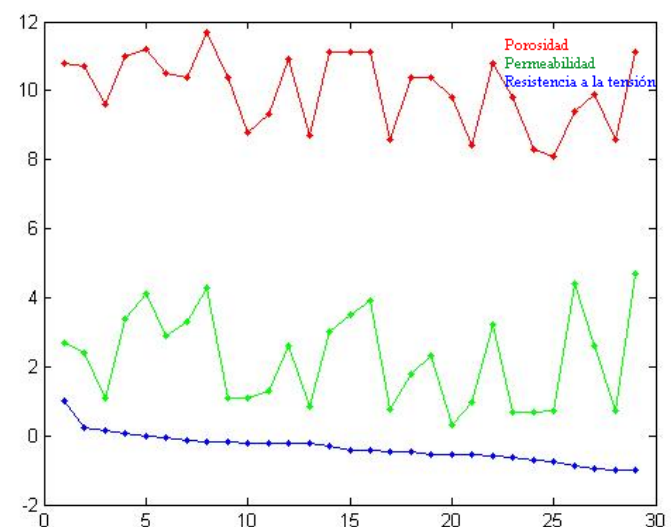


Fig. 3. Datos de salida ordenados de forma descendiente.

Los resultados son muy similares a los anteriores.

Se concluye entonces que la variación en el orden de los datos y la escogencia de información de la zona central para realizar las pruebas, presenta un leve mejoramiento, con respecto a las pruebas realizadas anteriormente.

Debido a esto, finalmente se busca formar un set de datos de validación separando dato por dato como único del set de validación y realizando por lo menos diez corridas en busca de obtener al menos un error entre el 20 y el 30%, así, se va formando un nuevo grupo de validación, el procedimiento se realizó de ésta manera hasta lograr un set de siete datos de validación.

Ya con esta nueva distribución de datos, se creó un configurador, mediante el uso de algoritmos genéticos y los errores se vieron reducidos hasta en un 15%.

VI. CONCLUSIONES

Es fundamental contar con un volumen de datos considerable y representativo pues no importa que tan buena sea la configuración, si no se cuenta con el set de datos adecuado, el porcentaje de error no será el esperado.

Además, es importante realizar sobre los datos un pre-procesado que permita organizar y seleccionar el conjunto que mejor represente la población en el entrenamiento y de esta manera obtener finalmente una generalización adecuada.

Finalmente se obtuvieron resultados dentro del rango posible de error por lo que se satisfacen los objetivos y expectativas del proyecto.

VII. REFERENCIAS

[1] CORZO RUEDA Reinel y RINCÓN PABÓN Claudio Omar. MEDICIÓN Y EVALUACIÓN DE LA MAGNITUD Y DIRECCIÓN DE LOS ESFUERZOS IN-SITU EN CAMPO. Colombia Tesis 2004.

[2] FAMIGLIETTI Nathaly, INTERPRETACIÓN Y PROCESAMIENTO DE DATOS SÍSMICOS, Schlumberger.

[3] SINGH V.K., SINGH D., SINGH T.N. 'PREDICTION OF STRENGTH PROPERTIES OF SOME SCHISTOSE ROCKS FROM PETROGRAPHIC PROPERTIES USING ARTIFICIAL NEURAL NETWORKS'. Department of Mining Engineering, Institute of Technology, Banaras Hindu University, Varanasi, 221005 India. Department of Electrical Engineering,

Institute of Technology, Banaras Hindu University, Varanasi, 221005 India. 28 November 2000.

[4] ROMERO Luis Alonso, CANO Teodoro Calonge. Capítulo 1. REDES NEURONALES Y RECONOCIMIENTO DE PATRONES.- España.

[5] S. Watanabe. PATTERN RECOGNITION: HUMAN AND MECHANICAL. - Wiley, New York 1985.

[6] FREEMAN James y SKAPURA David (1991), REDES NEURONALES. ALGORITMOS, APLICACIONES Y TÉCNICAS DE PROGRAMACIÓN. Addison - Wesley Iberoamericana S.A. U.S.A.

BIOGRAFIAS

Laura Viviana Galvis Carreño.

Nacida el 16 de Noviembre de 1985 en Bucaramanga, Santander. Graduada del Instituto Santa María Goretti como Bachiller académico en el año 2002. Actualmente cursa décimo semestre de Ingeniería de Sistemas en la Universidad Industrial de Santander y sus intereses son la inteligencia artificial, ingeniería del Software y agentes inteligentes.



César Augusto Ochoa Coronado.

Nacido el 22 de Diciembre de 1982 en Bucaramanga, Santander. Graduado del Instituto Técnico Superior Dámaso Zapata como Bachiller técnico electricista en el año 2000. Actualmente cursa décimo semestre de Ingeniería de Sistemas en la Universidad Industrial de Santander. Sus áreas de interés son: la inteligencia artificial, la minería de datos y la ingeniería del Software.



Henry Arguello Fuentes. Nacido en

Julio de 1976 en Simacota, Santander Colombia. Graduado como ingeniero electricista en el 2000 en la Universidad Industrial de Santander. Graduado como magíster en potencia eléctrica en el 2003 en la Universidad Industrial de Santander. Actualmente se desempeña como profesor auxiliar en dedicación de tiempo completo en la escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander. Sus áreas de



interés son el procesamiento de señales digitales, la inteligencia artificial y las telecomunicaciones.