



Estimación del módulo de resiliencia usando redes neuronales

Paul Garnica Anguas
Luis Enrique Gómez Carreño
Natalia Pérez García

Publicación Técnica No. 528
Sanfandila, Qro, 2018

SECRETARÍA DE COMUNICACIONES Y TRANSPORTES
INSTITUTO MEXICANO DEL TRANSPORTE

**Estimación del módulo de resiliencia usando redes
neuronales**

Publicación Técnica No. XXX
Sanfandila, Qro, 2018

Esta investigación fue realizada en la Coordinación de Infraestructura del Instituto Mexicano del Transporte, por el Dr. Paul Garnica Anguas, el Ing. Luis Enrique Gómez Carreño (estudiante de maestría de la Universidad Autónoma de Querétaro) y la Dra. Natalia Pérez García.

Esta investigación es el producto final del proyecto de investigación interna II-21/16. “Estimación del módulo de resiliencia por medio de redes neuronales”.

Contenido

| | |
|---|------|
| Índice de figuras | v |
| Índice de tablas | vii |
| Sinopsis | ix |
| Abstract | xi |
| Resumen ejecutivo | xiii |
| Introducción | 1 |
| Objetivo general | 1 |
| Objetivos específicos | 2 |
| 1. Las redes neuronales artificiales | 3 |
| 1.1. Definiciones de red neuronal artificial | 4 |
| 1.2. Elementos básicos de las redes neuronales artificiales | 4 |
| 1.2.1. Componentes | 4 |
| 1.2.2.1. Función de entrada | 4 |
| 1.2.2.2. Función de activación | 4 |
| 1.2.2.3. Función de salida. | 5 |
| 1.3. Mecanismos de aprendizaje | 5 |
| 1.3.1. Aprendizaje supervisado | 5 |
| 1.3.1.1. Algoritmo de gradiente descendente | 6 |
| 1.3.1.2. Algoritmo Gauss-Newton | 7 |
| 1.3.1.3. Algoritmo Levenberg-Marquardt | 10 |
| 1.3.1.4. Algoritmo Levenberg-Marquardt modificado | 12 |
| 1.3.2. Aprendizaje no supervisado | 15 |
| 1.4. Validación de la red neuronal | 15 |
| 1.5. Aplicaciones de las redes neuronales en ingeniería de pavimentos | 15 |
| 1.6. Softwares comerciales | 16 |
| 2. El módulo de resiliencia | 17 |
| 2.1. Definición | 17 |
| 2.2. Factores que afectan el módulo de resiliencia de suelos finos. | 18 |
| 2.2.1. Efecto del contenido de agua | 18 |
| 2.2.2. Efecto del peso volumétrico | 19 |
| 2.2.3. Efecto del estado de esfuerzos | 20 |
| 2.2.4. Efecto del índice plástico | 21 |
| 2.2.5. Efecto del tipo de compactación | 22 |
| 3. Evaluación de las variables de entrada de las redes neuronales | 25 |
| 3.1. Recopilación de las bases de datos | 25 |
| 3.1.1. Base de datos de Mao (1995) | 25 |
| 3.1.2. Base de datos de Maher et al. (2000) | 26 |

| | |
|--|----|
| 3.1.3. Base de datos de Malla y Joshi (2006) | 27 |
| 3.1.4. Base de datos de Richardson et al. (2009) | 28 |
| 3.1.5. Base de datos de Titi et al. (2011) | 29 |
| 3.1.6. Base de datos desarrollada en el Instituto Mexicano del Transporte | 30 |
| 3.2. Evaluación de las variables de entrada para las redes neuronales artificiales | 32 |
| 3.2.1. Suelos tipo limo y arcillas | 32 |
| 3.2.2. Suelos arenosos | 35 |
| 4. Redes neuronales artificiales para estimar el módulo de resiliencia | 39 |
| 4.1. Normalización de las bases de datos | 39 |
| 4.2. Estructura de las redes neuronales | 39 |
| 4.3. Desarrollo de las redes neuronales artificiales | 40 |
| 4.4. Entrenamiento de las redes neuronales artificiales | 41 |
| 4.4.1. Entrenamiento utilizando el algoritmo Levenberg-Marquardt | 41 |
| 4.4.2. Entrenamiento utilizando el algoritmo Levenberg-Marquardt modificado | 43 |
| 4.5. Validación de los modelos de redes neuronales | 45 |
| 4.6. Valores de Mr estimados versus datos de laboratorio | 46 |
| 4.7. Aplicación para estimar el módulo de resiliencia | 51 |
| 5 Conclusiones | 55 |
| Bibliografía y Referencias | 57 |
| Anexo 1. Programa | 63 |

Índice de figuras

| | Página |
|--|--------|
| Figura 1.1. Ejemplo de una red neuronal totalmente conectada (fuente: Matich, 2001) | 4 |
| Figura 1.2. Funciones de activación. (A) Escalón unitario, (B) lineal, (C) Logística Sigmoidal, (D) Sigmoidal tangente hiperbólica (Das, 2013) | 5 |
| Figura 2.1. Deformación de un espécimen bajo carga repetida (Huang, 2004) | 18 |
| Figura 2.2. Efecto del contenido de agua en el módulo de resiliencia de un suelo tipo CH (González, 2014) | 19 |
| Figura 2.3. Efecto del contenido de agua en el módulo de resiliencia de un suelo tipo SM (González, 2014) | 19 |
| Figura 2.4. Variación del módulo de resiliencia con el incremento en la energía de compactación | 20 |
| Figura 2.5. Módulo de resiliencia para un suelo tipo ML (González et al., 2014) | 21 |
| Figura 2.6. Efecto del índice de plasticidad en el Mr | 21 |
| Figura 2.7. Módulo de resiliencia obtenidos con muestras compactadas 2% abajo del contenido de agua óptimo | 22 |
| Figura 2.8. Módulos de resiliencia obtenidos con muestras compactadas con contenido de agua óptimo | 23 |
| Figura 2.9. Módulos de resiliencia obtenidos con muestras compactadas 2% arriba del contenido de agua óptimo. | 23 |
| Figura 4.1. Número de iteraciones del modelo de red neuronal con arquitectura 6-6-1 (arcillas y limos) | 46 |
| Figura 4.2. Número de iteraciones del modelo de red neuronal con arquitectura 4-7-1 (arenas) | 46 |
| Figura 4.3. Entrenamiento del modelo de red neuronal con arquitectura 6-6-1 para suelos tipo CH, CL, MH y ML | 49 |
| Figura 4.4. Validación del modelo de red neuronal con arquitectura 6-6-1 para suelos tipo CH, CL, MH y MH | 50 |
| Figura 4.5. Entrenamiento del modelo de red neuronal con arquitectura 4-7-1 para suelos tipo SM, SC y SP. | 50 |
| Figura 4.6. Validación del modelo de red neuronal con arquitectura 4-7-1 para suelos tipo SM, SC y SP. | 51 |
| Figura 4.7. Interfaz inicial del programa para la obtención del Mr en suelos finos compactados | 51 |
| Figura 4.8. Interfaz para la obtención del módulo de resiliencia en suelos tipo CH, CL, MH y ML | 52 |

Figura 4.9. Interfaz para la obtención del módulo de resiliencia en suelos tipo SM, SC y SP 52

Índice de tablas

| | Página |
|--|--------|
| Tabla 3.1. Propiedades índice, de compactación y clasificación de suelos. | 25 |
| Tabla 3.2. Módulos de resiliencia en la condición óptima de compactación | 26 |
| Tabla 3.3. Resumen de las propiedades de los suelos. | 26 |
| Tabla 3.4. Módulos de resiliencia en la condición óptima de compactación. | 27 |
| Tabla 3.5. Propiedades índice, de compactación y clasificación de suelos. | 27 |
| Tabla 3.6. Módulos de resiliencia en la condición óptima de compactación. | 28 |
| Tabla 3.7. Propiedades índice y clasificación de los suelos. | 28 |
| Tabla 3.8. Módulos de resiliencia en la condición óptima de compactación. | 29 |
| Tabla 3.9. Propiedades índice y clasificación de suelos. | 29 |
| Tabla 3.10. Módulos de resiliencia en condición óptima. | 30 |
| Tabla 3.11. Propiedades índice y clasificación de los suelos estudiados. | 31 |
| Tabla 3.12. Módulos de resiliencia en condición óptima de compactación. | 32 |
| Tabla 3.13. Nomenclatura de las variables utilizadas. | 33 |
| Tabla 3.14. Análisis de regresión lineal utilizando cuatro variables de entrada. | 33 |
| Tabla 3.15. Análisis de regresión lineal utilizando cinco variables de entrada. | 33 |
| Tabla 3.16. Análisis de regresión lineal utilizando seis variables de entrada. | 34 |
| Tabla 3.17. Rango de las variables y módulo de resiliencia. | 35 |
| Tabla 3.18. Tamaño de las arenas de acuerdo con la norma ASTM D2487-06. | 35 |
| Tabla 3.19. Análisis de regresión lineal utilizando cuatro variables de entrada. | 36 |
| Tabla 3.20. Análisis de regresión lineal utilizando cinco variables de entrada. | 36 |
| Tabla 3.21. Análisis de regresión lineal utilizando seis variables de entrada. | 37 |

| | | |
|-------------|---|----|
| Tabla 3.22. | Variables de entrada y rangos para las arenas. | 37 |
| Tabla 4.1. | Tipos de arquitectura de las redes neuronales. | 40 |
| Tabla 4.2. | Criterio del parámetro R^2 de bondad de ajuste. | 47 |
| Tabla 4.3. | Coeficientes de determinación (R^2) para el entrenamiento de las redes neuronales artificiales para suelos tipo arcilla y limo. | 48 |
| Tabla 4.4. | Coeficientes de determinación (R^2) para la validación de las redes neuronales artificiales para suelos tipo arcilla y limo. | 48 |
| Tabla 4.5. | Coeficientes de determinación (R^2) para el entrenamiento de las redes neuronales artificiales para suelos tipo arena. | 48 |
| Tabla 4.6. | Coeficientes de determinación (R^2) para la validación de las redes neuronales artificiales para suelos tipo arena. | 49 |

Sinopsis

El módulo de resiliencia es un parámetro de suma importancia en los métodos de diseño de pavimentos actuales, ya que es la base para determinar esfuerzos, desplazamientos y deformaciones en los mismos. Este parámetro ha sido estudiado ampliamente y un gran número de estas investigaciones se ha enfocado en proponer metodologías para predecirlo sin que sea necesario llevar a cabo pruebas de laboratorio. En este trabajo se propone utilizar las redes neuronales artificiales para predecir este parámetro, tras conocer las propiedades índice del suelo y el estado de esfuerzos. En el documento se presenta la programación de la red neuronal; y como producto final, se pone a disposición del usuario una aplicación para estimar el módulo de resiliencia.

Abstract

The resilient modulus is an important parameter in the new pavement design methods, because it is the basis for determining stresses, displacements and deformations in the structure. This parameter has been extensively researched and a large number of the studies have focused on predicting this parameter without performing laboratory testing. This research indicates that artificial neural networks provide a reliable tool to predict the resilient modulus of fine grained soils based only on index properties and stress states. In this document, it is shown the programming of the neural network, and as a final product, an application is given to the user to estimate the resilient modulus.

Resumen ejecutivo

El diseño de pavimentos requiere conocer varias propiedades de los materiales; una de ellas es el módulo de resiliencia, el cual es evaluado a través de una prueba triaxial de laboratorio que requiere equipo con diversos sensores; además, el tiempo de prueba puede ir desde dos horas -o más- dependiendo del tipo de material ensayado (suelos finos o materiales granulares). Debido a esto, varios investigadores se han dado a la tarea de encontrar formas alternativas para estimar dicho parámetro. Algunos proponen correlaciones con propiedades índice; algunos otros han encontrado que las redes neuronales artificiales son una forma adecuada de estimar el módulo de resiliencia al tener como base propiedades índice y estado de esfuerzos.

En este documento se investigó la pertinencia de estimar el módulo de resiliencia de suelos tipo arcillas, limos y arenas, a través de redes neuronales.

En el primer capítulo del trabajo, se discuten los conceptos sobre las redes neuronales y algunas de sus aplicaciones en la ingeniería de pavimentos. En el capítulo dos, aparecen las gráficas en las que se observa cómo el contenido de agua, la densidad, el estado de esfuerzos, el índice de plasticidad y el método de compactación influyen en la determinación del módulo de resiliencia. En el capítulo tres, se muestran los análisis de regresión llevados a cabo para determinar las variables de entrada de las redes neuronales. De las variables estudiadas se pueden mencionar las siguientes:

- Compacidad,
- Índice de grupo,
- Presión de confinamiento,
- Esfuerzo desviador,
- Índice de plasticidad,
- Contenido de agua óptimo,
- Esfuerzo octaédrico, etc.

En este mismo capítulo, se muestran las variables de entrada elegidas para las redes neuronales. En el capítulo cuatro, se comenta sobre el desarrollo de las redes neuronales, su entrenamiento y validación. También se desarrolla una aplicación para estimar el módulo de resiliencia. Esta aplicación utiliza las redes

neuronales artificiales programadas en este trabajo para estimar el módulo de resiliencia. La publicación lleva adjunta esta aplicación, para que pueda ser utilizada por todos aquellos involucrados en el diseño de pavimentos.

Finalmente, se exponen las conclusiones derivadas del trabajo. También se presenta un anexo con el programa de las redes neuronales.

Introducción

Las nuevas metodologías para el diseño de un pavimento requieren conocer las propiedades de los materiales que conformarán la estructura. De estas propiedades, el módulo de resiliencia de los materiales de las capas inferiores a la carpeta asfáltica o concreto hidráulico es de especial importancia; ya que es la base para determinar los esfuerzos, desplazamientos y deformaciones en el pavimento. Posteriormente, estas respuestas son usadas como insumos en los criterios de falla para predecir el número de repeticiones de carga que soporta el pavimento antes de fallar (Huang, 2004).

Desde 1986, cuando es introducido el parámetro módulo de resiliencia, se han realizado diversas investigaciones en torno a éste. Un gran número de ellas ha sido enfocado a la obtención de modelos para predecir este parámetro, sin llevar a cabo pruebas de laboratorio. También han propuesto otras alternativas de predicción como la inteligencia computacional, en donde no es necesario conocer el modelo físico que rige el fenómeno en cuestión; solo basta tener una base de datos del comportamiento del fenómeno o parámetro estudiado. Posteriormente, los algoritmos computacionales se encargan de aprender asociaciones entre los patrones y con ello obtienen una predicción confiable del fenómeno analizado.

En esta investigación, se utilizó el modelo de redes neuronales artificiales de aprendizaje supervisado para predecir el módulo de resiliencia. En algunas investigaciones -como las reportadas por Castro (1998), Honarmand-Ebrahimi (2006), Dione et al., (2013), Tarawneh y Nazzal (2014), etc.- vemos que esta técnica es útil para predecir este parámetro. A diferencia de las investigaciones anteriormente mencionadas, en este documento se propuso utilizar los algoritmos de entrenamiento de Quasi-Newton, para la creación de los modelos de redes neuronales; ya que estos no han sido probados para la predicción del módulo de resiliencia.

Objetivo general

Desarrollar un modelo de red neuronal para estimar el módulo de resiliencia de suelos finos compactados, en función de propiedades físicas y estado de esfuerzos; además, poner a disposición una aplicación –que tenga implícita la red neuronal- para estimar el módulo de resiliencia.

Objetivos específicos:

- Realizar un estudio exhaustivo sobre el módulo de resiliencia y los factores que lo afectan.
- Llevar a cabo la recopilación de las bases de datos que contengan las propiedades físicas, estado de esfuerzos y valores de módulos de resiliencia de suelos finos en la condición óptima de compactación (respecto a la prueba Proctor estándar).
- Crear la base de datos para el entrenamiento y validación de la red neuronal.
- Evaluar las variables de entrada del modelo, utilizando un análisis de regresión lineal múltiple.
- Programar (en lenguaje Visual Basic) el algoritmo de Levenberg-Marquardt modificado.
- Validar las redes neuronales artificiales con suelos no utilizados en el entrenamiento.
- Crear una aplicación para estimar el módulo de resiliencia de suelos finos compactados en la condición óptima de compactación.

1. Las redes neuronales artificiales

1.1. Definiciones de red neuronal artificial

Existen varias definiciones de red neuronal, desde definiciones cortas o las que intentan explicarla detalladamente. De acuerdo con Matich (2001), algunas de estas definiciones son:

1. Una nueva forma de computación inspirada en modelos biológicos,
2. Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles,
3. Un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas,
4. Son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.

Otra definición es, por ejemplo, la indicada por Rabuñal y Puertas (2006) (citados por Alemán, 2017):

“Una red neuronal artificial (RNA) es un sistema de procesamiento de información que está basado en la forma en que funciona el sistema nervioso de los animales. Constituyen, por tanto, modelos electrónicos o computacionales basados en la estructura neuronal del cerebro.”

Salas (2004) (citado por Alemán, 2017) señala que las RNA son un método para resolver problemas de forma individual o combinada con otros métodos, para aquellas tareas de clasificación, identificación, diagnóstico, optimización o predicción en las que el balance datos/conocimiento se inclina hacia los datos y donde, adicionalmente, puede haber la necesidad de aprendizaje en tiempo de ejecución y de cierta tolerancia a fallos. En estos casos las RNA se adaptan dinámicamente reajustando constantemente los pesos de sus interconexiones.

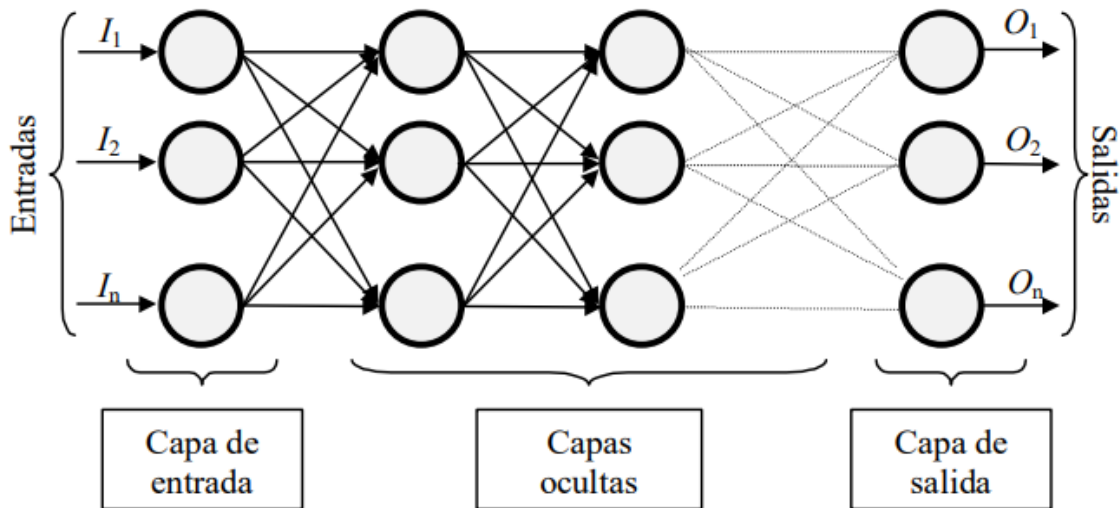
De acuerdo con Lefebvre (2005), una red neuronal es un sistema adaptable que puede aprender relaciones a través de presentación repetida de datos, y es capaz de generalizar a nuevos datos que no ha visto.

Shahin et al. (2002) dicen que las redes neuronales artificiales son una forma de inteligencia artificial; la que, por medio de su arquitectura, trata de simular el comportamiento propio del cerebro humano para la resolución de ciertos problemas ingenieriles.

1.2. Elementos básicos de las redes neuronales artificiales

1.2.1. Componentes

La Figura 1.1 indica los elementos de la red neuronal. Se observa que la red está constituida por neuronas interconectadas y arregladas en tres capas (esto último puede variar). Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida” (Matich, 2001).



Fuente: Matich, 2001

Figura 1.1. Ejemplo de una red neuronal totalmente conectada

1.2.1.1. Función de entrada

La neurona trata muchos valores de entrada como si fueran uno solo, esto se conoce con el nombre de entrada global. Algunas de las funciones de entrada más comúnmente utilizadas son: (a) sumatoria de las entradas pesadas, (b) producto de las entradas pesadas y (c) máximo de las entradas pesadas.

1.2.1.2. Función de activación

La función de activación calcula el estado de actividad de una neurona; transforma la entrada global en un valor (estado) de activación, cuyo rango normalmente va de 0 o 1 o de -1 a 1. Entonces, una neurona puede estar totalmente inactiva (0 o -1) o activa (1).

Las funciones de activación más utilizadas en ingeniería geotécnica son la función escalón de Heaviside, lineal, sigmoideal y la tangente hiperbólica. La Figura 1.2 muestra las características de las funciones anteriormente mencionadas (Castro, 1998; Das, 2013).

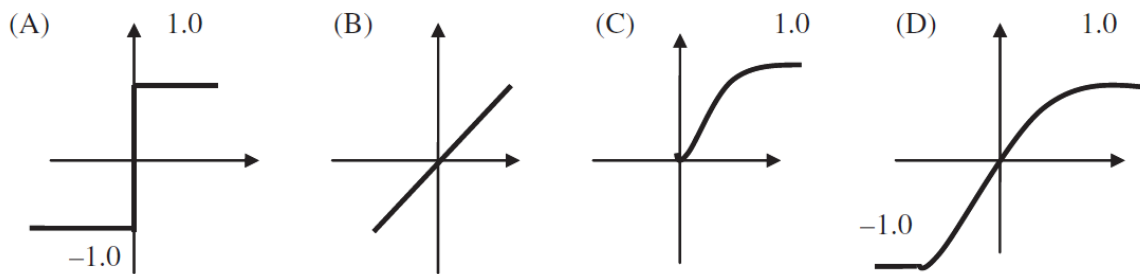


Figura 1.2. Funciones de activación: (A) Escalón Unitario, (B) Lineal, (C) Logística sigmoideal, (D) Sigmoideal tangente hiperbólica (Das, 2013).

1.2.1.3. Función de salida

El último componente que necesita una neurona es la *función de salida*. El valor resultante de esta función es la salida de la neurona i ; por lo tanto, la función de salida determina qué valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona subsiguiente. Normalmente, no cualquier valor es permitido como una entrada para una neurona; por lo tanto, los valores de salida están comprendidos en el rango $[0, 1]$ o $[-1, 1]$ (Matich, 2001).

1.3. Mecanismos de aprendizaje

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada; el proceso termina cuando los valores de los pesos permanecen estables (Matich, 2001).

1.3.1. Aprendizaje supervisado

Existen dos métodos de aprendizaje: el supervisado y el no supervisado. El primero se caracteriza porque realiza el proceso de aprendizaje mediante un entrenamiento controlado por un agente externo (supervisor) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y, en caso de que esta no coincida con la deseada, procederá a modificar los pesos de las conexiones, para conseguir que la salida obtenida se aproxime a la deseada (Matich, 2001).

El objetivo principal en el aprendizaje supervisado es minimizar la suma de los cuadrados de los residuales entre la salida predicha por la red y la deseada. Lo anterior es conocido como la función de rendimiento de la red y se expresa como (Das, 2013):

$$E = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^M (e_{p,m})^2 \quad (1.1)$$

$$e_{p,m} = d_{p,m} - o_{p,m} \quad (1.2)$$

Donde:

$e_{p,m}$ = Error en el entrenamiento de la salida m por una entrada P

$d_{p,m}$ = Vector de salida deseada

$o_{p,m}$ = Vector de salida actual o de la red

P = Vector de entrada

A continuación, se presentan los diversos algoritmos de Quasi-Newton existentes en la literatura, para lograr minimizar la suma de los cuadrados de los residuales entre la salida predicha y la deseada.

1.3.1.1. Algoritmo de gradiente descendente

Este algoritmo usa las derivadas de primer orden de la función de rendimiento para encontrar el error mínimo en el espacio de soluciones. Normalmente, el gradiente g está definido como la derivada de primer orden de la función de error expresada de acuerdo con la ecuación 1.3 (Caglar y Arman, 2007; Yu y Wilamowski, 2011).

$$g = \nabla E = \frac{\partial E}{\partial w} = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_3}, \dots, \frac{\partial E}{\partial w_k} \right) \quad (1.3)$$

En cada paso de entrenamiento, el conjunto inicial de vectores de peso son ajustados hacia la dirección de la disminución máxima del error (E) y ordenados

por una tasa de aprendizaje λ (α). Matemáticamente, un peso es actualizado a su valor nuevo como sigue (Das, 2013).

$$W_{\text{nuevo}} = W_{\text{anterior}} - \alpha \nabla E \quad (1.4)$$

El algoritmo utilizado con frecuencia en el aprendizaje supervisado es el de retropropagación de gradiente descendente; también es conocido como un algoritmo ineficiente, debido a que su convergencia es lenta. Existen dos razones por la que su convergencia es lenta, la primera razón es que los pasos de entrenamiento no se adecuan al tamaño de la superficie de error y provoca que la solución oscile o no converja; mientras que la segunda razón es que la curvatura de la superficie del error puede no ser la misma en todas las direcciones (Yu y Wilamowski, 2011).

La convergencia lenta de este método de gradiente descendente puede ser mejorada utilizando el algoritmo de Gauss-Newton, el cual usa las segundas derivadas de la función de error para evaluar la curvatura de la superficie de error. Este método puede encontrar pasos adecuados para cada dirección de la superficie de error y, por lo tanto, la convergencia es rápida; especialmente si la función de error tiene una superficie cuadrática (Yu y Wilamowski, 2011).

1.3.1.2. Algoritmo de Gauss-Newton

El algoritmo de Gauss-Newton asume que los componentes del gradiente (g_1, g_2, \dots, g_n) son funciones de pesos y también son linealmente independientes:

$$g_1 = F_1 (w_1, w_2, \dots, w_n)$$

$$g_2 = F_2 (w_1, w_2, \dots, w_n) \quad (1.5)$$

$$g_n = F_N (w_1, w_2, \dots, w_n)$$

Donde

F_1, F_2, \dots, F_n = Relaciones no lineales entre los pesos y los componentes del gradiente

El sistema linealmente independiente puede ser expresado a través de series de Taylor, y con ello tomar la aproximación de primer orden.

$$g_1 \approx g_{1,0} + \frac{\partial g_1}{\partial w_1} \Delta w_1 + \frac{\partial g_1}{\partial w_2} \Delta w_2 + \dots + \frac{\partial g_1}{\partial w_N} \Delta w_n$$

$$g_2 \approx g_{2,0} + \frac{\partial g_1}{\partial w_1} \Delta w_1 + \frac{\partial g_1}{\partial w_2} \Delta w_2 + \dots + \frac{\partial g_1}{\partial w_N} \Delta w_N \quad (1.6)$$

$$g_n \approx g_{n,0} + \frac{\partial g_N}{\partial w_1} \Delta w_1 + \frac{\partial g_N}{\partial w_2} \Delta w_2 + \dots + \frac{\partial g_N}{\partial w_N} \Delta w_N$$

Por definición, el vector gradiente puede ser expresado también como:

$$\frac{\partial g_1}{\partial w_j} = \frac{\partial \left(\frac{\partial E}{\partial w_j} \right)}{\partial w_j} = \frac{\partial^2 E}{\partial w_j \partial w_j} \quad (1.7)$$

Si se sustituye $\frac{\partial g_1}{\partial w_j}$ en el sistema de ecuaciones 1.6, se obtiene el siguiente sistema.

$$\begin{aligned} g_1 &\approx g_{1,0} + \frac{\partial^2 E}{\partial w_1 \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_1 \partial w_N} \Delta w_N \\ g_2 &\approx g_{2,0} + \frac{\partial^2 E}{\partial w_2 \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_2 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_2 \partial w_N} \Delta w_N \\ g_n &\approx g_{n,0} + \frac{\partial^2 E}{\partial w_n \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_n \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_n \partial w_N} \Delta w_N \end{aligned} \quad (1.8)$$

Para conseguir que el error sea mínimo, el vector gradiente debe igualarse con 0; por lo tanto.

$$\begin{aligned} -\frac{\partial E}{\partial w_1} = -g_{1,0} &\approx \frac{\partial^2 E}{\partial w_1 \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_1 \partial w_N} \Delta w_N \\ -\frac{\partial E}{\partial w_2} = -g_{2,0} &\approx \frac{\partial^2 E}{\partial w_2 \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_2 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_2 \partial w_N} \Delta w_N \\ -\frac{\partial E}{\partial w_N} = -g_{N,0} &\approx \frac{\partial^2 E}{\partial w_N \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_N \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_N \partial w_N} \Delta w_N \end{aligned} \quad (1.9)$$

El sistema anterior puede ser expresado en sistema matricial de la siguiente forma:

$$-\begin{Bmatrix} g_1 \\ g_2 \\ \dots \\ g_N \end{Bmatrix} = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1 \partial w_1} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N \partial w_N} \end{bmatrix} \begin{Bmatrix} \Delta w_1 \\ \Delta w_2 \\ \dots \\ \Delta w_N \end{Bmatrix} \quad (1.10)$$

Donde la matriz cuadrada de las derivadas de segundo orden también es conocida como “Matriz de Hessian”. Los pesos serán actualizados de acuerdo con el gradiente, de la siguiente forma:

$$-g = H \Delta w \quad (1.11)$$

$$\Delta w = -H^{-1}g \quad (1.12)$$

$$w_{k+1} = w_k + \Delta w_k = w_k - H^{-1}_k g_k \quad (1.13)$$

La matriz de Hessian resulta ser difícil de obtener, ya que sus términos contienen derivadas de segundo orden. Para simplificar los cálculos, el algoritmo de Gauss-Newton propone introducir la matriz del Jacobiano (Yu y Wilamowski, 2011).

$$J = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \dots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \dots & \frac{\partial e_{1,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{1,M}}{\partial w_1} & \frac{\partial e_{1,M}}{\partial w_2} & \dots & \frac{\partial e_{1,M}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{p,1}}{\partial w_1} & \frac{\partial e_{p,1}}{\partial w_2} & \dots & \frac{\partial e_{p,1}}{\partial w_N} \\ \frac{\partial e_{p,2}}{\partial w_1} & \frac{\partial e_{p,2}}{\partial w_2} & \dots & \frac{\partial e_{p,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{p,M}}{\partial w_1} & \frac{\partial e_{p,M}}{\partial w_2} & \dots & \frac{\partial e_{p,M}}{\partial w_N} \end{bmatrix} \quad (1.14)$$

Integrando la función de rendimiento, se obtiene que el vector gradiente puede ser expresado como:

$$g_i = \frac{\partial E}{\partial w_i} = \frac{\partial \left(\frac{1}{2} \sum_{p=1}^P \sum_{m=1}^M (e_{p,m})^2 \right)}{\partial w_i} = \sum_{p=1}^P \sum_{m=1}^M \frac{\partial e_{p,m}}{\partial w_i} e_{p,m} \quad (1.15)$$

$$g = Je \quad (1.16)$$

La finalidad del algoritmo es simplificar la matriz de Hessian, es por ello que se toma el enésimo término de esta matriz.

$$H_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} = \frac{\partial^2 \left(\frac{1}{2} \sum_{p=1}^P \sum_{m=1}^M (e_{p,m})^2 \right)}{\partial w_i \partial w_j} = \frac{\partial e_{p,m}}{\partial w_i} \frac{\partial e_{p,m}}{\partial w_j} \quad (1.17)$$

$$H = J^T J \quad (1.18)$$

Finalmente, el algoritmo de Gauss-Newton actualiza los pesos de la siguiente manera:

$$w_{n+1} = w_n + \Delta w_n = w_n - H_n^{-1} g_n \quad (1.19)$$

$$w_{n+1} = w_n - (J_n^T J_n)^{-1} J_n e_n \quad (1.20)$$

Si la matriz $J^T J$ no es invertible, la red tendrá un error en la convergencia de la solución.

1.3.1.3. Algoritmo de Levenberg-Marquardt

Para descartar las derivadas de segundo orden de la función de rendimiento, el método de Levenberg-Marquardt hace uso de la aproximación de Gauss-Newton que acepta la matriz jacobiana (Nawi et al., 2013). En el caso de sistemas lineales que tienen una superficie de rendimiento cuadrática, el algoritmo de Levenberg-Marquardt puede encontrar el mínimo en un solo paso, pero en los sistemas no lineales, como redes neuronales donde la superficie del error tiende a ser convexa, se requiere de varios pasos para la convergencia de la solución (Khan et al., 2013). El algoritmo de Levenberg-Marquardt asegura que la matriz de Hessian $J^T J$ sea invertible. Introduciendo una constante llamada coeficiente de aprendizaje (positivo), se asegura que la diagonal principal de la matriz de Hessian sea

invertible. Cuando el coeficiente μ es muy grande este algoritmo se convierte en el gradiente descendente, es por ello que este coeficiente debe ser un escalar de pequeña dimensión para que se pueda controlar el proceso de aprendizaje (Yu y Wilamowski, 2011; Suratgar et al., 2007; Farajzadeh et al., 2014).

$$H = J^T J + \mu I \quad (1.21)$$

Donde:

μ = Coeficiente de combinación (positivo)

I = Matriz identidad

Por lo tanto, los pesos de la red neuronal son actualizados con la siguiente regla:

$$w_{n+1} = w_n - (J^T J + \mu I)^{-1} J_n^T e_n \quad (1.22)$$

Donde:

w_{n+1} = Vector de pesos actuales

w_n = Vector de pesos

J_n = Matriz del jacobiano

e_n = Vector de error = $e(w_{n+1} - w_n)$

Este algoritmo minimiza la función de error al tratar de mantener el paso entre el vector de pesos anterior y el actualizado. La red puede tomar diversas funciones de rendimiento como la suma de los errores al cuadrado, el error cuadrático medio, la raíz del error cuadrático medio, entre otras (Gopinath y Dwarakish, 2015).

El número de capas ocultas determinará la complejidad de la función de salida de la red neuronal $f(w_k)$. El número de neuronas ocultas, así como las funciones de transferencia por utilizar en cada una de las neuronas. Previamente, se determinó la salida para cualquier arquitectura de una red neuronal

Para implementar el algoritmo de Levenberg-Marquardt en el entrenamiento de redes neuronales de aprendizaje supervisado, se debe calcular el jacobiano y posteriormente, el proceso de aprendizaje. El jacobiano lo conforman las derivadas de primer orden del error saliente de la red con respecto a los pesos que la integran; a continuación, se presenta la derivación de la matriz jacobiana (Yu y Wilamowski, 2010).

La matriz del jacobiano tendrá como último término:

$$\frac{\partial e_n}{\partial w_n} = \frac{\partial f(w_n)}{\partial w_n} \quad (1.23)$$

Con lo anterior se obtiene la matriz jacobiana para alcanzar la minimización de la función de salida de la red.

Implementación del algoritmo de Levenberg-Marquardt

Conociendo cómo actualizar los pesos dentro del algoritmo de Levenberg-Marquardt, se siguen los siguientes pasos para su optimización (De la Fuente, 1997).

1. Definir el vector de pesos iniciales de la red neuronal (w_n) y $\mu = 0.01$ (valor apropiado).

2. Calcular $p_n = (J^T J + \mu I)^{-1} J^T e_n$

3. Calcular $r_n = e_n^T e_n$

4. Si $r_n(w_n + p_n) < r_n(w_n)$ entonces

 Si $p_n < \text{Tolerancia}$. Problema resuelto

 Si $p_n > \text{Tolerancia}$. Realizar:

$$\mu = \frac{\mu}{10}$$

$$n = n + 1$$

$$w_{n+1} = w_n - p_n$$

 Volver a calcular p_n

Si no

$$\mu = 10 \mu$$

 Volver a calcular p_n , sin calcular J_n

5. Fin del ciclo

6. Fin del programa

1.3.1.4. Algoritmo de Levenberg-Marquardt modificado

Suratgar et al., (2007) propusieron una modificación al algoritmo de Levenberg-Marquardt. La principal ventaja de este método es que reduce la cantidad de oscilación en el proceso de aprendizaje.

La función gradiente indica la dirección donde se encuentra el mínimo de la función de error. El algoritmo de Gauss-Newton para la obtención del gradiente del error utiliza las segundas derivadas de la función. Este algoritmo podemos expresarlo de la siguiente forma:

$$-\begin{Bmatrix} g_1 \\ g_2 \\ \dots \\ g_n \end{Bmatrix} = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1 \partial w_1} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N \partial w_N} \end{bmatrix} \begin{Bmatrix} \Delta w_1 \\ \Delta w_2 \\ \dots \\ \Delta w_n \end{Bmatrix} \quad (1.24)$$

Donde los pesos quedarán actualizados de acuerdo con:

$$-g = H\Delta w \quad (1.25)$$

$$\Delta w = -H^{-1}g \quad (1.26)$$

$$w_{n+1} = w_n + \Delta w_n = w_n - H^{-1}_n g_n \quad (1.27)$$

La matriz anterior contiene las derivadas de segundo orden de la función de error por minimizar, pero esto genera una complejidad en el cálculo; por lo que se introduce una matriz que contenga solo las derivadas de primer orden o matriz del Jacobiano (Yu y Wilamowski, 2010).

Sustituyendo las ecuaciones 1.23 y 1.25, los pesos quedarán actualizados de la siguiente forma:

$$w_{n+1} = w_n - (J^T J)^{-1} J^T_n e_n \quad (1.28)$$

Como se había mencionado anteriormente, la matriz Hessian $J^T J$ puede no ser invertible; por ello es necesario hacer la siguiente modificación.

$$G = H + \mu I \quad (1.29)$$

Con lo anterior se supone que los eigenvalores y eigenvectores de la matriz Hessian son $(\lambda_1, \lambda_2, \dots, \lambda_n)$ y (z_1, z_2, \dots, z_n) , por lo tanto:

$$Gz_n = (H + \mu I) z_n \quad (1.30)$$

$$Gz_n = Hz_n + \mu z_n \quad (1.31)$$

$$Gz_n = \lambda_n z_n + \mu z_n \quad (1.32)$$

Los eigenvectores de G son los mismos como los eigenvectores de H , y los eigenvalores de G son $(\lambda_n + \mu)$. La matriz G es definida positiva aumentando el valor de la constante de aprendizaje μ hasta $(\lambda_n + \mu) > 0$ para todo i ; por lo tanto la matriz será invertible (Suratgar et al., 2007). Entonces, dado que se asegura que la matriz Hessian sea invertible, la actualización de pesos queda de la forma:

$$w_{n+1} = w_n - (J^T J + \mu I)^{-1} J^T e_n \quad (1.33)$$

A diferencia del algoritmo de Levenberg-Marquardt clásico, este algoritmo modificado se distingue porque la constante de aprendizaje no es parámetro constante, sino que es igual a:

$$\mu = 0.01 e^T e \quad (1.34)$$

Implementación del algoritmo de Levenberg-Marquardt modificado

Conociendo cómo actualizar los pesos dentro del algoritmo de Levenberg-Marquardt modificado, se siguen los siguientes pasos para su optimización en redes neuronales artificiales (Suratgar et al., 2007).

1. Definir el vector de pesos iniciales de la red neuronal (w_n)
2. Calcular el error en la predicción inicial e_n
3. Calcular $p_n = (J^T J + \mu I)^{-1} J^T e_n$
4. Calcular $r_n = e_n^T e_n$
5. Calcular $\mu = 0.01 e_n^T e_n$
6. Hacer las iteraciones necesarias hasta que $r_{n+1} < r_n$

$$n = n+1$$

$$p_n = (J^T J + \mu I)^{-1} J^T e_n$$

$$w_{n+1} = w_n - p_n$$

$$\mu = 0.01 e_n^T e_n$$

7. Fin del programa

1.3.2. Aprendizaje no supervisado

Las redes con aprendizaje sin supervisión no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si su salida es o no correcta. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que puedan ser establecidas entre los datos que se presenten en su entrada (Matich, 2001).

1.4. Validación de la red neuronal

Después del proceso de entrenamiento, los pesos de las conexiones en la red neuronal quedan fijos. Como paso siguiente, se debe comprobar si la red neuronal puede resolver nuevos problemas del tipo general, para los que ha sido entrenada. Por lo tanto, con el propósito de validar la red neuronal, se requiere de otro conjunto de datos, denominado conjunto de validación (Matich, 2001).

1.5. Aplicaciones de las redes neuronales en ingeniería de pavimentos

Las redes neuronales pueden ser utilizadas en un gran número y variedad de aplicaciones. A continuación, se enumeran algunas de las aplicaciones publicadas en ingeniería de pavimentos:

- Estimar módulos elásticos en pavimentos (Beltran y Romo, 2011; Vila, 2017),
- Predicción del escalonamiento en pavimentos rígidos (Navarro, 2017),
- Predicción de índice de agrietamiento y condición del pavimento (Yang et al., 2003),
- Análisis de pavimento de concreto con técnicas no destructivas (Birka, 2008).
- Predicción de módulo de resiliencia (Park et al., 2009; Ibrahim et al., 2017; Tarawneh y Nazzal, 2014, Kim et al. 2014; etc).
- Predicción de módulo dinámico de mezclas asfálticas (Leiva et al., 2013),
- Predicción del módulo elástico de concreto con agregado de concreto reciclado (Duan et al., 2013), etc.

La mayoría de las aplicaciones de las redes neuronales consiste en realizar un reconocimiento de patrones, como puede ser: buscar un patrón en una serie de

ejemplos, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado. La ventaja de las redes neuronales reside en el procesado paralelo, adaptativo y no lineal. El dominio de aplicación de las redes neuronales también puede ser clasificado de la siguiente forma: asociación y clasificación, regeneración de patrones, regresión y generalización, y optimización.

1.6. Softwares comerciales

Cabe mencionar que la mayor parte de los investigadores han demostrado que las redes predicen de forma adecuada propiedades de materiales o comportamiento de pavimentos, pero la pregunta es ¿cómo implementar las redes neuronales? En este sentido, la investigación mostrada en este documento no utilizó un programa de redes neuronales comercial, sino que la red neuronal se programó y ésta fue utilizada para formular una aplicación (app) y así poder estimar el módulo de resiliencia de suelos finos (materiales que pasan la malla No. 4) teniendo como base propiedades índice de los materiales y estado de esfuerzos.

2 El módulo de resiliencia

2.1. Definición

Los materiales que forman parte de la estructura del pavimento son considerados como no elásticos, ya que después de cada aplicación de carga ocurre una deformación permanente en el material. Sin embargo; si la carga es pequeña, comparada con la resistencia del material y es repetida con alta frecuencia, la deformación es cercana a ser recuperable y puede ser considerada como elástica (Huang, 2004).

El módulo elástico basado en la deformación recuperable o elástica bajo la acción de cargas repetidas es llamado módulo de resiliencia. En los años cincuenta fue introducido este parámetro y quedó definido como la relación de la magnitud del esfuerzo desviador repetido en compresión triaxial con respecto a la deformación axial recuperable o resiliente, y se expresa como sigue (Garnica et al., 2000):

$$M_r = \frac{\sigma_d}{\epsilon_r} \quad (2.1)$$

Donde:

M_r = Módulo de resiliencia

σ_d = Esfuerzo desviador

ϵ_r = Deformación recuperable

La Figura 2.1 muestra la deformación de un espécimen (dentro de cualquier capa del pavimento) bajo la acción de carga repetida. En la etapa inicial de la aplicación de carga se presenta una deformación permanente tal y como se mencionó anteriormente. A medida que el número de repeticiones aumenta, la deformación plástica disminuye, ya que el material llega a comportarse como elástico (Garnica et al., 2000; Huang, 2004).

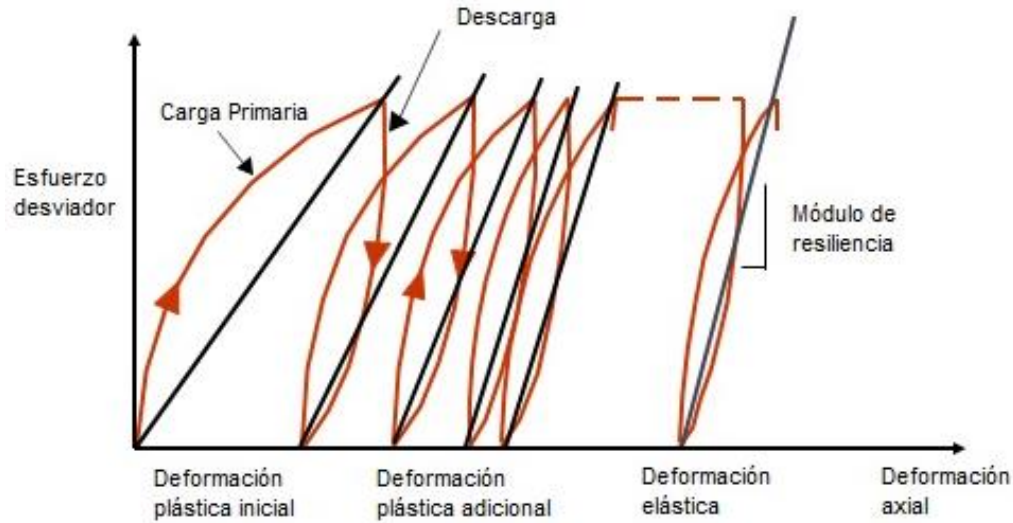


Figura 2.1. Deformación de un espécimen bajo carga repetida

(Huang, 2004).

2.2. Factores que afectan el módulo de resiliencia en suelos finos

El módulo de resiliencia de suelos finos compactados depende de varios factores. A continuación, se resume y demuestra con resultados de laboratorio cómo es el efecto de algunos de ellos.

2.2.1. Efecto del contenido de agua

González (2014) llevó a cabo pruebas de módulo de resiliencia en varios suelos compactados. En su investigación, realizó pruebas a tres diferentes contenidos de agua. En el óptimo de compactación, dos por ciento por debajo del óptimo y dos por ciento por arriba del óptimo. La Figura 2.2 muestra los valores de módulo de resiliencia para una arcilla CH, a los contenidos de agua indicados.

Como se puede observar, los valores más altos de módulo de resiliencia aparecen cuando el material tiene un contenido de agua más bajo respecto del óptimo. Cuando las muestras tienen contenido de agua óptimo o mayor, los valores de módulo de resiliencia tienden a valores similares.

Es importante señalar que suelos como las arenas no presentan el mismo comportamiento. En la Figura 2.3 aparecen los módulos de resiliencia de una arena clasificada como SM. En este caso, también se ensayó a tres contenidos de agua, óptimo y $\pm 2\%$ con respecto al óptimo; para este suelo, los módulos de resiliencia parecen ser independientes del contenido de agua ya que se obtienen valores similares en los tres casos.

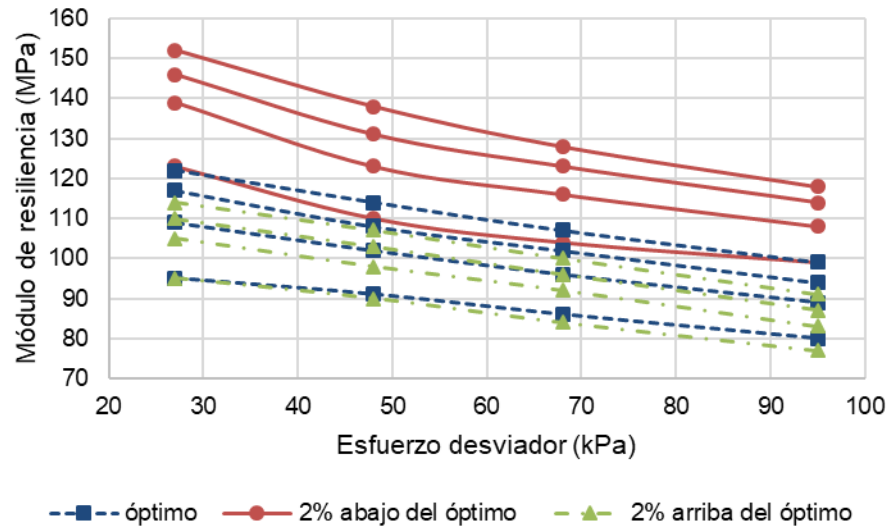


Figura 2.2. Efecto del contenido de agua en el módulo de resiliencia para un suelo tipo CH (González, 2014).

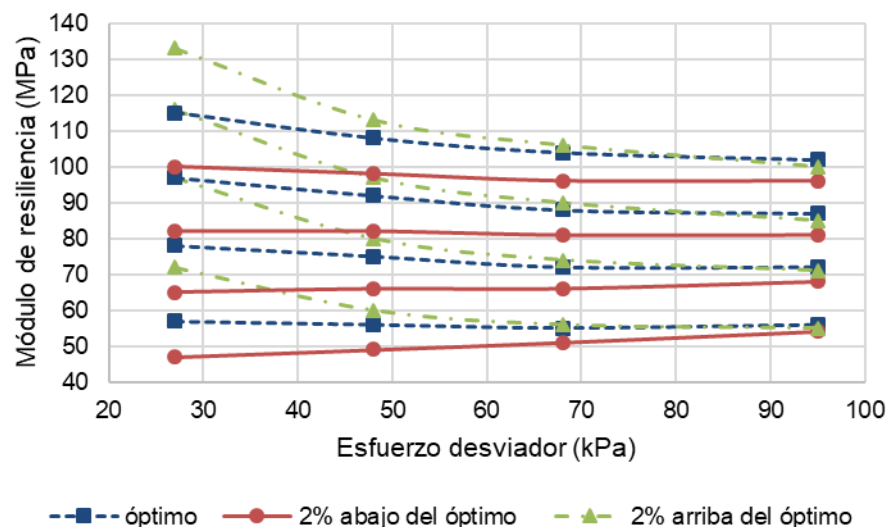


Figura 2.3. Efecto del contenido de agua en el módulo de resiliencia para un suelo tipo SM (González, 2014).

2.2.2. Efecto de la energía de compactación

La energía de compactación también es una de las variables que afecta el módulo de resiliencia. La Figura 2.4 señala los valores de módulo de resiliencia de muestras que fueron compactadas en la condición óptima tanto para la energía de

compactación Proctor estándar y modificada. El suelo era una arcilla clasificada como CH.

Como se observa de la Figura 2.4; a mayor energía de compactación, mayor módulo de resiliencia. Esta variable involucra directamente el peso volumétrico, ya que éste se incrementa con la energía de compactación.

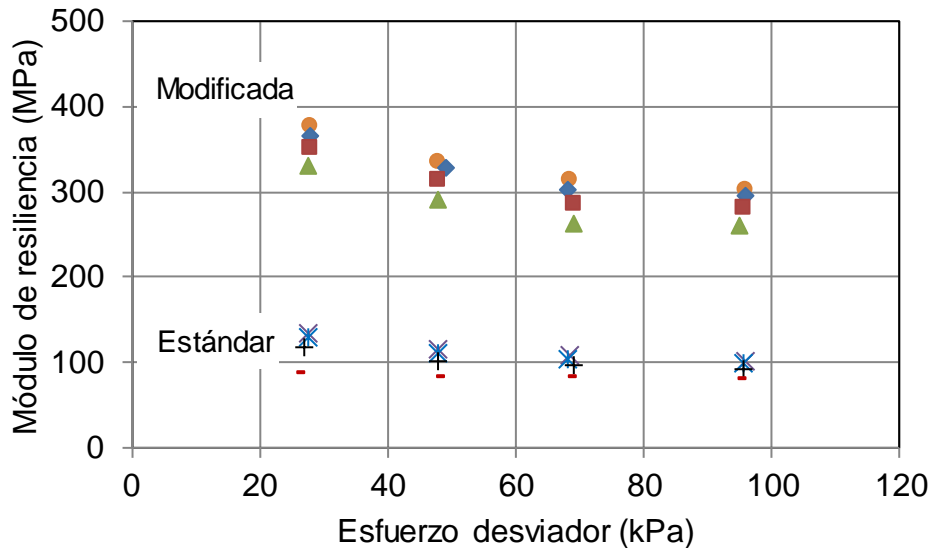


Figura 2.4. Variación del módulo de resiliencia con el incremento en energía de compactación.

2.2.3. Efecto del estado de esfuerzos

Otra variable que afecta el M_r es el estado de esfuerzos (esfuerzo de confinamiento y esfuerzo desviador). Gonzalez et al. (2014), en su investigación, muestran que para todos los suelos estudiados, el módulo de resiliencia se reduce a medida que el esfuerzo desviador se incrementa; y por otro lado, este se incrementa con la presión confinante. La Figura 2.5 muestra resultados típicos de módulos de resiliencia de un limo de baja plasticidad. En la gráfica se ve claramente el efecto tanto del esfuerzo desviador como de la presión confinante.

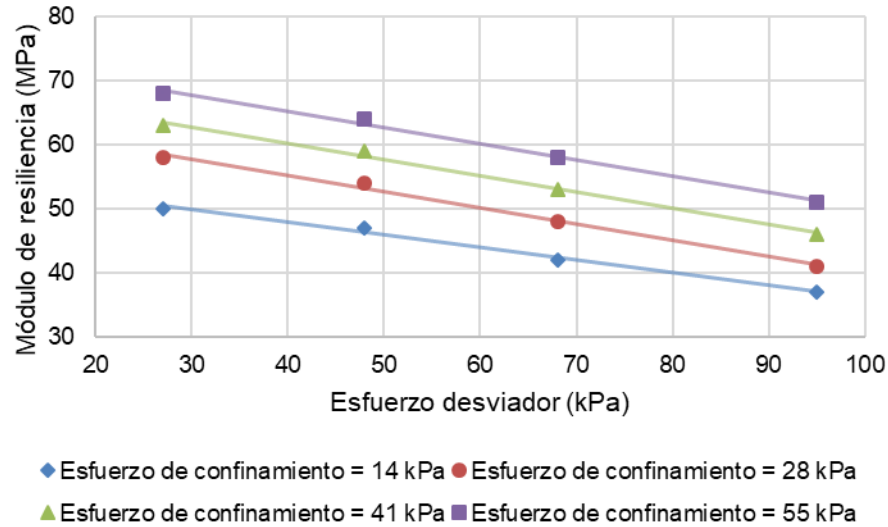


Figura 2.5. Módulo de resiliencia para un suelo tipo ML (González et al., 2014).

2.2.4. Efecto del índice plástico

Para observar el efecto del índice de plasticidad, se graficaron los resultados de módulo de resiliencia de cuatro tipos de suelo con diferentes índices de plasticidad. En la Figura 2.6 no se ve una tendencia clara con el incremento del índice de plasticidad.

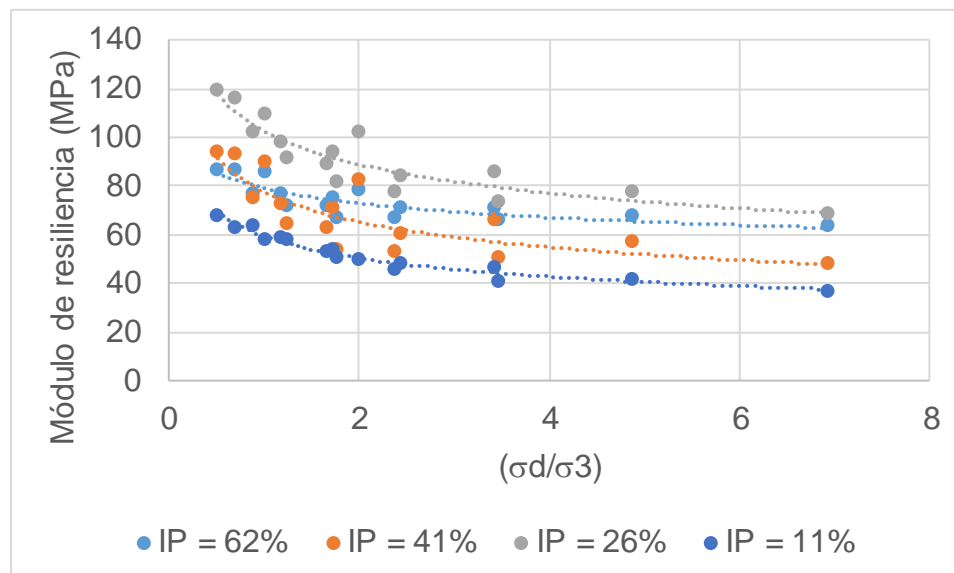


Figura 2.6. Efecto del índice de plasticidad en el Mr

2.2.5. Efecto del tipo de compactación

Pérez et al. (2012) llevaron a cabo una investigación sobre compactación con equipo giratorio. Como parte de ese estudio, los autores evaluaron el efecto del tipo de compactación en el módulo de resiliencia. Para ello, compactaron probetas en equipo giratorio y prepararon otras con compactación dinámica. Las probetas fueron compactadas con tres contenidos de agua: en el óptimo, 2% abajo del óptimo y 2% arriba del óptimo. Los resultados indicaron que el tipo de compactación sí tiene efecto en los módulos de resiliencia cuando las muestras son compactadas dos por ciento abajo del óptimo. Los módulos de resiliencia fueron independientes del tipo de compactación para probetas compactadas con contenido de agua óptimo o dos por ciento por encima de éste (Figuras 2.7, 2.8 y 2.9).

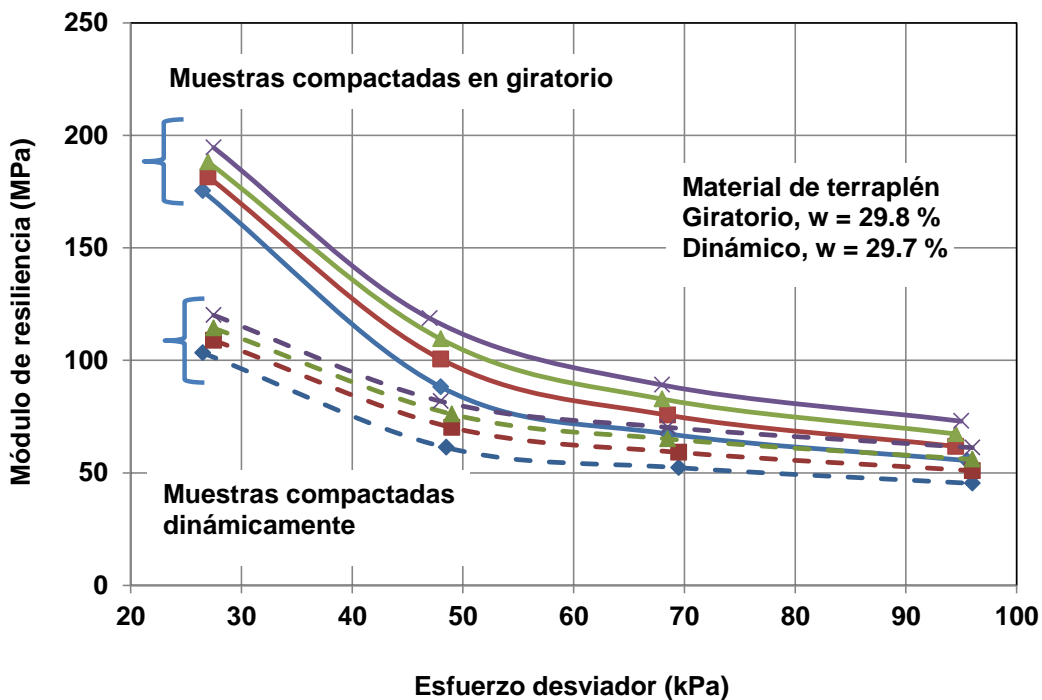


Figura 2.7. Módulos de resiliencia obtenidos con muestras compactadas 2% abajo del contenido de agua óptimo.

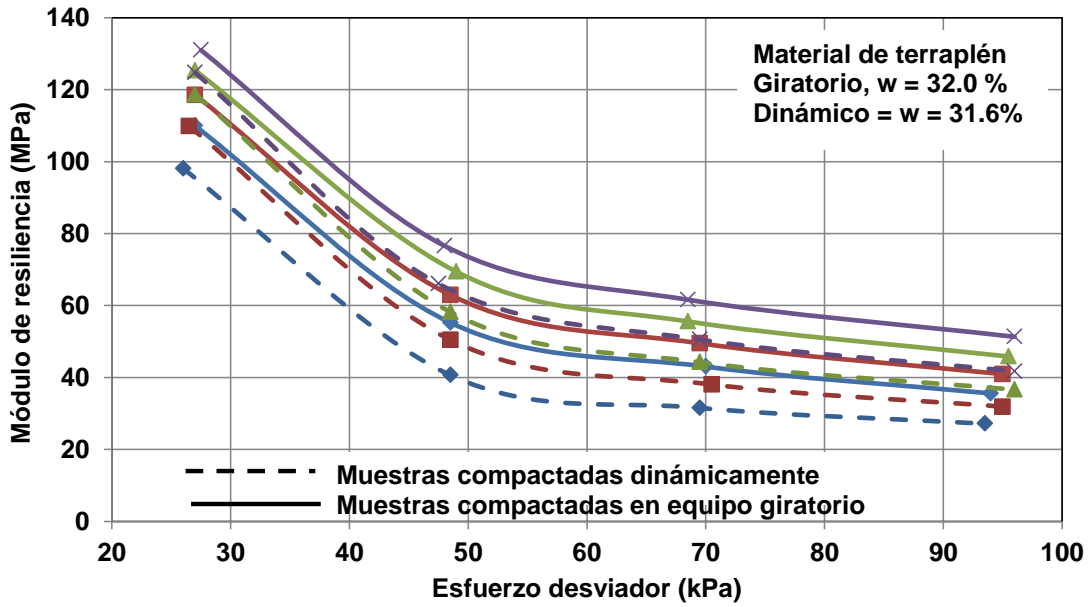


Figura 2.8. Módulos de resiliencia obtenidos con muestras compactadas con contenido de agua óptimo.

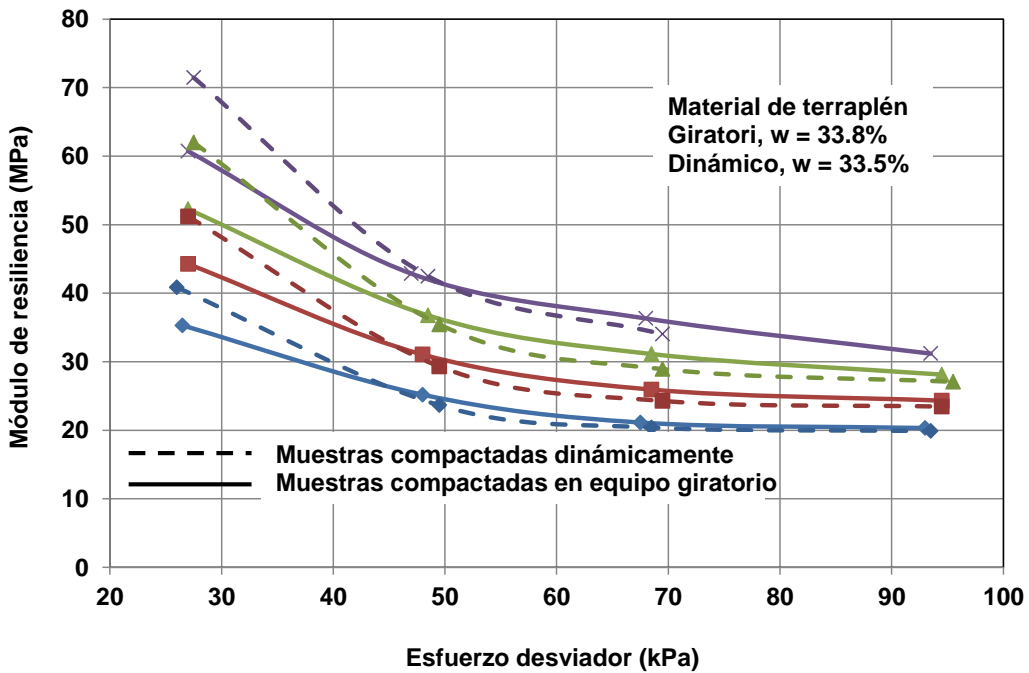


Figura 2.9. Módulos de resiliencia obtenidos con muestras compactadas 2% arriba del contenido de agua óptimo.

3 Evaluación de las variables de entrada de las redes neuronales

3.1. Recopilación de las bases de datos

Las bases de datos utilizadas en las redes neuronales fueron recopiladas de siete investigaciones. Las investigaciones de Mao (1995), Richardson et al., (2009), Titi et al., (2011) y González (2014) fueron utilizados para integrar la base de datos para suelos tipo CH, CL, MH, ML. Mientras que las investigaciones de Maher et al., (2000), Malla y Joshi (2006), Richardson et al., (2009), Titi et al., (2011), González (2014) y Curiel (2015) fueron utilizadas para la base de datos de suelos tipo SM, SC y SP; en los siguientes incisos se presentan los aspectos relevantes de las investigaciones de las cuales se extrajeron los datos. De igual forma se resumen en tablas los datos de las propiedades índice y valores de módulo de resiliencia utilizados en esta investigación.

3.1.1. Base de datos Mao (1995)

Mao (1995) desarrolló un modelo matemático para la predicción del módulo de resiliencia para suelos tipo CL. Los suelos utilizados fueron muestreados en diversas zonas de la región de Ohio en los Estados Unidos de América. Después de realizar el muestreo, determinaron las propiedades índice y de compactación (Tabla 3.1).

Tabla 3.1. Propiedades índice, de compactación y clasificación de suelos.

| Banco | SUCS | LL(%) | LP (%) | IP (%) | P ₂₀₀ (%) | S _s | W _{opt} (%) | γ _d (kN/m ³) |
|-------|------|-------|--------|--------|----------------------|----------------|----------------------|-------------------------------------|
| NB4 | CL | 40.4 | 19 | 21.4 | 70.5 | 2.74 | 15.7 | 17.5 |
| NB6 | CL | 31.1 | 17.7 | 13.4 | 71.3 | 2.72 | 15.5 | 18 |
| SB3 | CL | 30 | 16.4 | 13.6 | 72.5 | 2.7 | 15.1 | 18.2 |
| SB4 | CL | 31.7 | 17.9 | 13.8 | 73.2 | 2.72 | 14.9 | 18.1 |
| MN2 | CL | 27.4 | 16.7 | 10.7 | 65.9 | 2.69 | 13.9 | 18.5 |
| MN4 | CL | 37.6 | 19.6 | 18 | 75.8 | 2.76 | 14.5 | 18.3 |
| GN2 | CL | 24.3 | 15.2 | 9.1 | 53.4 | 2.68 | 12.9 | 18.7 |
| GS5 | CL | 35.6 | 20.8 | 14.8 | 95.1 | 2.75 | 17.4 | 17.5 |
| BN3 | CL | 33.4 | 18.2 | 15.2 | 73.9 | 2.73 | 15.2 | 18.2 |
| BS2 | CL | 24.9 | 16.3 | 8.6 | 65.2 | 2.67 | 13.2 | 19.1 |

Para la obtención de los módulos de resiliencia, se utilizó el protocolo SHRP P46. Los especímenes de prueba fueron de 10 cm de diámetro y 20 cm de altura; stos

fueron sometidos a 15 secuencias de carga; en cada secuencia se aplicaron 100 repeticiones de carga; excepto en la de condicionamiento, en donde se aplican 1000 ciclos de carga. La carga fue aplicada durante 0.1 de segundo, seguido por un periodo de descarga de 0.9 segundo utilizando una señal tipo haversine.

Mao (1995) llevó a cabo una sola prueba de módulo de resiliencia por cada tipo de suelo; los resultados aparecen en la Tabla 3.2.

Tabla 3.2. Módulos de resiliencia en condición óptima de compactación.

| Banco | Módulo de resiliencia (MPa) | | | | | | | | | | | | | | |
|-------|-----------------------------|----|----|----|----|---------------------|----|----|----|----|---------------------|----|----|----|----|
| | $\sigma_3 = 14$ kPa | | | | | $\sigma_3 = 28$ kPa | | | | | $\sigma_3 = 41$ kPa | | | | |
| | $\sigma_d = 14$ kPa | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 |
| NB4 | 64 | 49 | 44 | 40 | 36 | 61 | 50 | 45 | 41 | 36 | 58 | 51 | 44 | 39 | 35 |
| NB6 | 56 | 40 | 35 | 30 | 25 | 52 | 40 | 35 | 30 | 26 | 47 | 37 | 30 | 26 | 24 |
| SB3 | 52 | 31 | 25 | 22 | 20 | 49 | 31 | 25 | 22 | 20 | 45 | 30 | 24 | 21 | 20 |
| SB4 | 56 | 37 | 31 | 28 | 26 | 54 | 39 | 31 | 28 | 25 | 55 | 44 | 37 | 30 | 26 |
| MN2 | 42 | 33 | 28 | 23 | 19 | 42 | 33 | 29 | 23 | 20 | 43 | 35 | 27 | 24 | 20 |
| MN4 | 82 | 56 | 47 | 47 | 44 | 80 | 57 | 48 | 47 | 44 | 62 | 54 | 48 | 45 | 43 |
| GN2 | 45 | 35 | 32 | 30 | 28 | 45 | 36 | 32 | 30 | 28 | 41 | 34 | 28 | 27 | 26 |
| GS5 | 43 | 31 | 22 | 19 | 17 | 44 | 33 | 24 | 20 | 18 | 42 | 33 | 26 | 21 | 19 |
| BN3 | 49 | 39 | 31 | 26 | 22 | 51 | 41 | 32 | 26 | 22 | 50 | 44 | 33 | 26 | 22 |
| BS2 | 30 | 21 | 16 | 15 | 15 | 31 | 22 | 17 | 16 | 15 | 29 | 23 | 17 | 15 | 15 |

3.1.2. Base de datos de Maher et al., (2000)

Maher et al., 2000 propusieron un modelo para la predicción del módulo de resiliencia para suelos tipo SM, SC, SP. Los suelos utilizados en el desarrollo del modelo fueron muestreados en diversas zonas de New Jersey en los Estados Unidos de América. Los autores indicaron que habían utilizado normas ASTM para determinar las propiedades índice y de compactación de los suelos, las cuales se resumen en la Tabla 3.3.

Tabla 3.3. Resumen de propiedades de los suelos.

| Banco | SUCS | LL (%) | LP (%) | IP (%) | P ₂₀₀ (%) | P ₄₀ (%) | S _s | w _{opt} (%) | γ_d (kN/m ³) |
|--------|------|--------|--------|--------|----------------------|---------------------|----------------|----------------------|---------------------------------|
| Rt-46 | SC | 15 | N.P | N.P | 30.1 | 66.49 | 2.67 | 8.5 | 20.17 |
| Rt-80a | SC | 0 | N.P | N.P | 33.3 | 64.9 | 2.6 | 9 | 18.89 |
| Rt 295 | SM | 0 | N.P | N.P | 9.9 | 83.03 | 2.66 | 9 | 18.09 |
| Rt 80b | SC | 20.5 | 19 | 1.5 | 36.6 | 63.70 | 2.66 | 8.25 | 20.06 |
| Rt-206 | SM | 21 | 17 | 4 | 43 | 62.71 | 2.69 | 8.5 | 20.09 |

En esta investigación se usó el protocolo AASHTO T-307 para la obtención del módulo de resiliencia. Conforme al protocolo, se prepararon especímenes en

moldes cilíndricos de 7 cm de diámetro y 14 cm de altura. Posteriormente, cada espécimen fue sometido a 15 secuencias de carga; en cada una de ellas, varió tanto el confinamiento como el esfuerzo desviador. En cada secuencia fueron aplicadas 100 repeticiones de carga, usando una señal tipo haversine. La carga fue aplicada durante 0.1 de segundo seguido por un periodo de descarga de 0.9 de segundo.

Maher et al., (2000) reportaron valores de M_r que eran promedio de tres réplicas (Tabla 3.4), lo cual incrementó la confiabilidad de sus resultados.

Tabla 3.4. Módulos de resiliencia en condición óptima de compactación.

| Banco | Módulo de resiliencia (MPa) | | | | | | | | | | | | | | |
|--------|-----------------------------|----|----|----|---------------------|----|----|----|---------------------|----|-----|-----|----|----|----|
| | $\sigma_3 = 14$ kPa | | | | $\sigma_3 = 28$ kPa | | | | $\sigma_3 = 41$ kPa | | | | | | |
| | $\sigma_d = 14$ kPa | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 |
| Rt-46 | 75 | 58 | 50 | 51 | 54 | 80 | 76 | 68 | 66 | 65 | 120 | 104 | 95 | 90 | 82 |
| Rt-80a | 77 | 67 | 62 | 56 | 53 | 93 | 77 | 69 | 66 | 64 | 126 | 103 | 88 | 85 | 78 |
| Rt 295 | 63 | 56 | 50 | 50 | 52 | 94 | 71 | 64 | 62 | 66 | 114 | 95 | 91 | 91 | 86 |
| Rt 80b | 68 | 59 | 51 | 50 | 52 | 83 | 75 | 70 | 70 | 68 | 104 | 97 | 93 | 90 | 90 |
| Rt-206 | 73 | 59 | 49 | 50 | 52 | 87 | 73 | 65 | 63 | 63 | 120 | 97 | 92 | 80 | 76 |

3.1.3. Base de datos Malla y Joshi (2006)

Malla y Joshi (2006) publicaron un catálogo de módulos de resiliencia para suelos tipo CH, CL, MH, SM, SC. Los suelos utilizados en su estudio fueron muestreados en diversas zonas de Nueva Inglaterra en los Estados Unidos de América.

Posterior al muestreo, se determinaron las propiedades índice y de compactación de acuerdo con la normativa AASHTO (Tabla 3.5). La norma AASHTO T307 se utilizó para determinar los módulos de resiliencia (Tabla 3.6).

Tabla 3.5. Propiedades índice, de compactación y clasificación de los suelos.

| Banco | SUCS | LL (%) | LP (%) | IP (%) | P ₂₀₀ (%) | Pasa No. 40 (%) | W _{opt} (%) | γ_d (kN/m ³) |
|-------|-------|--------|--------|--------|----------------------|-----------------|----------------------|---------------------------------|
| CT-1 | SP-SM | 22 | NP | NP | 5 | 35 | 11.2 | 19.07 |
| CT-4 | SC-SM | 28 | 22 | 6 | 26.9 | 46 | 11.1 | 19.39 |
| CT-7 | SP | 18 | NP | NP | 3.1 | 79.3 | 14.7 | 16.11 |
| CT-8 | SP | 18 | NP | NP | 4 | 69.4 | 12.5 | 16.37 |
| CT-14 | SP | 17 | NP | NP | 2.6 | 75.5 | 13.8 | 16.23 |

Tabla 3.6. Módulos de resiliencia en condición óptima.

| Banco | Módulo de resiliencia (MPa) | | | | | | | | | | | | | | |
|-------|-----------------------------|----|----|----|----|-----------------------------|----|----|----|----|-----------------------------|----|----|----|----|
| | $\sigma_3 = 14 \text{ kPa}$ | | | | | $\sigma_3 = 28 \text{ kPa}$ | | | | | $\sigma_3 = 41 \text{ kPa}$ | | | | |
| | $\sigma_d = 14 \text{ kPa}$ | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 |
| CT-1 | 45 | 47 | 51 | - | - | 66 | 67 | 71 | 76 | 78 | 90 | 95 | 96 | 97 | 98 |
| CT-4 | 57 | 51 | 48 | 48 | 50 | 79 | 70 | 64 | 62 | 62 | 92 | 86 | 77 | 74 | 72 |
| CT-7 | 35 | 36 | 40 | 41 | 37 | 51 | 51 | 54 | 58 | 60 | 64 | 66 | 67 | 71 | 73 |
| CT-8 | 36 | 37 | 40 | 39 | 29 | 54 | 54 | 56 | 59 | 60 | 67 | 70 | 71 | 72 | 74 |
| CT-14 | 32 | 33 | 36 | 38 | 38 | 49 | 49 | 52 | 55 | 57 | 63 | 66 | 67 | 69 | 71 |

3.1.4 Base de datos de Richardson et al., (2009)

Richardson et al., (2009) presentaron un catálogo de módulos de resiliencia para suelos tipo CH, CL, MH, SM, SC. Los suelos fueron muestreados en diversas zonas de Missouri en los Estados Unidos de América. Las propiedades de los suelos aparecen a continuación:

Tabla 3.7. Propiedades índice y clasificación de los suelos.

| Banco | SUCS | LL (%) | LP (%) | IP (%) | P ₂₀₀ (%) | P ₄₀ (%) | S _s | W _{opt} (%) | γ_d (kN/m ³) |
|----------|------|--------|--------|--------|----------------------|---------------------|----------------|----------------------|---------------------------------|
| 6MKWM011 | CH | 50 | 21 | 29 | 59 | 74 | 2.7 | 20 | 15.85 |
| 6MKWM012 | SC | 30 | 17 | 13 | 24 | 48 | 2.66 | 11 | 18.83 |
| 6MKWM013 | SC | 23 | 13 | 10 | 37 | 75 | 2.66 | 11 | 18.99 |
| 6MKWM015 | CL | 32 | 18 | 14 | 82 | 100 | 2.68 | 15 | 17.57 |
| 6MKWM016 | CH | 63 | 21 | 42 | 92 | 98 | 2.7 | 21 | 15.06 |
| 6MKWM017 | CH | 65 | 23 | 42 | 67 | 83 | 2.7 | 20 | 15.22 |
| 6MKWM019 | CL | 48 | 17 | 31 | 74 | 93 | 2.68 | 16 | 16.95 |
| 6MKWM020 | CL | 35 | 22 | 13 | 97 | 99 | 2.68 | 20 | 16.01 |
| 6MKWM021 | CL | 37 | 23 | 14 | 82 | 96 | 2.68 | 18 | 15.85 |
| 6MKWM022 | MH | 61 | 33 | 28 | 76 | 80 | 2.68 | 32 | 12.56 |
| 6MKWM023 | CL | 46 | 24 | 22 | 90 | 97 | 2.68 | 20 | 15.70 |
| 6MKWM024 | CL | 49 | 24 | 25 | 91 | 95 | 2.68 | 21 | 15.07 |
| 6MKWM025 | SC | 44 | 19 | 25 | 46 | 56 | 2.66 | 13 | 17.58 |
| 6MKWM026 | CH | 57 | 18 | 39 | 75 | 94 | 2.7 | 22 | 15.54 |
| 6MKWM027 | SM | 27 | 20 | 7 | 25 | 33 | 2.66 | 9 | 19.78 |
| 6MKWM028 | CH | 52 | 22 | 30 | 96 | 99 | 2.7 | 22 | 15.23 |
| 6MKWM029 | CL | 34 | 20 | 14 | 75 | 88 | 2.68 | 15 | 17.11 |
| 6MKWM030 | SC | 38 | 23 | 15 | 36 | 44 | 2.66 | 12 | 18.52 |
| 6MKWM032 | CH | 60 | 20 | 40 | 97 | 99 | 2.7 | 24 | 14.75 |
| 6MKWM033 | CL | 43 | 17 | 26 | 90 | 98 | 2.68 | 18 | 16.16 |
| 6MKWM034 | CL | 48 | 19 | 29 | 98 | 99 | 2.68 | 19 | 15.85 |
| 6MKWM035 | CL | 36 | 21 | 15 | 93 | 96 | 2.68 | 18 | 15.69 |
| 6MKWM036 | CH | 55 | 24 | 31 | 96 | 98 | 2.7 | 23 | 14.75 |
| 6MKWM037 | CL | 48 | 19 | 29 | 88 | 98 | 2.7 | 19 | 16.48 |

El módulo de resiliencia fue obtenido con protocolo AASHTO T-307; los especímenes compactados fueron de 10 cm de diámetro y 20 cm de altura.

Richardson et al., (2009) indicaron que habían utilizado tres réplicas para cada tipo suelo. En la Tabla 3.8 se presentan los valores promedio de módulo de resiliencia obtenidos.

Tabla 3.8. Módulos de resiliencia en condición óptima.

| Banco | Módulo de resiliencia (MPa) | | | | | | | | | | | | | | |
|----------|-----------------------------|------|------|------|------|---------------------|------|------|------|------|---------------------|------|------|------|------|
| | $\sigma_3 = 14$ kPa | | | | | $\sigma_3 = 28$ kPa | | | | | $\sigma_3 = 41$ kPa | | | | |
| | $\sigma_d = 14$ kPa | 27.6 | 41.4 | 55.2 | 68.9 | 13.8 | 27.6 | 41.4 | 55.2 | 68.9 | 13.8 | 27.6 | 41.4 | 55.2 | 68.9 |
| 6MKWM011 | 120 | 119 | 116 | 108 | 102 | 147 | 139 | 130 | 123 | 113 | 161 | 150 | 138 | 126 | 115 |
| 6MKWM012 | 153 | 156 | 150 | 146 | 140 | 186 | 183 | 175 | 168 | 161 | 203 | 196 | 191 | 179 | 172 |
| 6MKWM013 | 163 | 146 | 136 | 130 | 123 | 184 | 167 | 160 | 147 | 140 | 209 | 185 | 167 | 152 | 144 |
| 6MKWM015 | 53 | 44 | 41 | 41 | 40 | 63 | 53 | 49 | 47 | 47 | 78 | 70 | 60 | 57 | 38 |
| 6MKWM016 | 86 | 78 | 70 | 64 | 57 | 95 | 86 | 77 | 69 | 61 | 101 | 91 | 81 | 71 | 63 |
| 6MKWM017 | 125 | 78 | 69 | 62 | 57 | 129 | 91 | 79 | 70 | 63 | 134 | 100 | 82 | 73 | 63 |
| 6MKWM019 | 90 | 76 | 68 | 63 | 60 | 118 | 91 | 80 | 72 | 66 | 106 | 87 | 79 | 74 | 68 |
| 6MKWM020 | 53 | 29 | 23 | 21 | 21 | 55 | 34 | 26 | 24 | 23 | 53 | 41 | 31 | 26 | 24 |
| 6MKWM021 | 103 | 70 | 59 | 54 | 51 | 98 | 78 | 69 | 62 | 58 | 97 | 85 | 75 | 68 | 63 |
| 6MKWM022 | 110 | 99 | 92 | 85 | 80 | 128 | 108 | 98 | 90 | 84 | 120 | 112 | 97 | 92 | 87 |
| 6MKWM023 | 165 | 135 | 108 | 96 | 88 | 417 | 148 | 120 | 107 | 99 | 173 | 143 | 124 | 110 | 101 |
| 6MKWM024 | 91 | 90 | 84 | 80 | 75 | 109 | 108 | 98 | 87 | 81 | 125 | 121 | 101 | 92 | 83 |
| 6MKWM025 | 88 | 76 | 75 | 77 | 80 | 103 | 83 | 81 | 80 | 81 | 115 | 97 | 83 | 80 | 83 |
| 6MKWM026 | 60 | 42 | 33 | 27 | 23 | 62 | 44 | 35 | 28 | 24 | 65 | 49 | 36 | 29 | 24 |
| 6MKWM027 | 70 | 52 | 48 | 51 | 55 | 82 | 59 | 55 | 57 | 59 | 104 | 83 | 67 | 59 | 64 |
| 6MKWM028 | 143 | 89 | 75 | 68 | 61 | 120 | 94 | 84 | 73 | 64 | 140 | 106 | 88 | 76 | 64 |
| 6MKWM029 | 63 | 46 | 37 | 33 | 32 | 72 | 54 | 44 | 39 | 37 | 80 | 66 | 52 | 43 | 39 |
| 6MKWM030 | 130 | 121 | 109 | 101 | 97 | 136 | 129 | 118 | 110 | 103 | 145 | 136 | 122 | 114 | 109 |
| 6MKWM032 | 97 | 91 | 84 | 78 | 74 | 110 | 101 | 92 | 85 | 79 | 119 | 109 | 98 | 89 | 82 |
| 6MKWM033 | 125 | 115 | 109 | 105 | 101 | 143 | 131 | 126 | 120 | 114 | 164 | 148 | 136 | 131 | 122 |
| 6MKWM034 | 112 | 105 | 98 | 92 | 89 | 131 | 118 | 109 | 102 | 97 | 135 | 127 | 115 | 107 | 101 |
| 6MKWM035 | 159 | 144 | 135 | 129 | 124 | 171 | 160 | 150 | 143 | 136 | 192 | 170 | 157 | 146 | 140 |
| 6MKWM036 | 62 | 89 | 76 | 68 | 62 | 68 | 105 | 86 | 74 | 66 | 129 | 104 | 87 | 75 | 67 |
| 6MKWM037 | 65 | 47 | 38 | 33 | 31 | 77 | 56 | 44 | 38 | 34 | 77 | 61 | 47 | 39 | 35 |

3.1.5 Base de datos de Titi et al., (2011)

En las Tablas 3.9 y 3.10 aparece el resumen de las propiedades y módulo de resiliencia de suelos de Wisconsin (Titi et al., 2011).

Tabla 3.9. Propiedades índice y clasificación de suelos.

| Banco | SUCS | LL(%) | LP (%) | IP (%) | P ₂₀₀ (%) | Ss | w _{opt} (%) | γ_d (kN/m ³) |
|--------------|------|-------|--------|--------|----------------------|-------|----------------------|---------------------------------|
| Fondulac | MH | 55.5 | 33.5 | 22 | 91 | 2.81 | 21.25 | 16 |
| Deer creek A | CL | 44.4 | 25.65 | 18.75 | 83.05 | 2.535 | 16.5 | 16.85 |
| Deer creek B | CL | 42.85 | 24.95 | 17.9 | 80.4 | 2.5 | 16.5 | 17.1 |
| Superior-1 | MH | 63.4 | 36.9 | 26.5 | 84.65 | 2.64 | 24.5 | 14.8 |
| Racine-1 | CL | 35.45 | 22.75 | 12.7 | 85.7 | 2.56 | 15.95 | 17.45 |
| Highland-1 | CL | 37 | 22.5 | 14.5 | 83.25 | 2.74 | 15.75 | 16.7 |
| Highland-2 | CL | 37 | 24 | 13 | 81.95 | 2.77 | 14.5 | 17.35 |
| Highland-3 | CH | 58.15 | 25.9 | 32.25 | 87.45 | 2.525 | 23 | 15.5 |
| Antigo | CL | 29.5 | 18.5 | 11 | 91 | 2.63 | 14.5 | 17.5 |
| Dodgeville | CL | 37 | 25 | 12 | 97 | 2.55 | 18.8 | 16.1 |

Tabla 3.10. Módulos de resiliencia en condición óptima.

| Banco | Módulo de resiliencia (MPa) | | | | | | | | | | | | | | |
|--------------|-----------------------------|-----|-----|-----|-----|-----------------------------|-----|-----|-----|-----|-----------------------------|-----|-----|-----|-----|
| | $\sigma_3 = 14 \text{ kPa}$ | | | | | $\sigma_3 = 28 \text{ kPa}$ | | | | | $\sigma_3 = 41 \text{ kPa}$ | | | | |
| | $\sigma_d = 14 \text{ kPa}$ | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 | 14 | 28 | 41 | 55 | 69 |
| Fondulac | 90 | 88 | 86 | 85 | 83 | 98 | 94 | 92 | 90 | 88 | 102 | 100 | 96 | 93 | 91 |
| Deer creek A | 83 | 80 | 77 | 75 | 73 | 93 | 89 | 85 | 81 | 78 | 99 | 95 | 90 | 85 | 80 |
| Deer creek B | 111 | 110 | 109 | 109 | 108 | 123 | 123 | 122 | 123 | 120 | 129 | 133 | 130 | 127 | 124 |
| Superior-1 | 99 | 98 | 97 | 96 | 95 | 104 | 104 | 103 | 101 | 99 | 108 | 108 | 107 | 105 | 101 |
| Racine-1 | 79 | 76 | 72 | 70 | 66 | 88 | 83 | 79 | 75 | 71 | 96 | 92 | 85 | 79 | 74 |
| Highland-1 | 103 | 102 | 102 | 103 | 104 | 112 | 115 | 114 | 115 | 113 | 124 | 126 | 124 | 121 | 120 |
| Highland-2 | 86 | 85 | 84 | 82 | 81 | 99 | 96 | 93 | 91 | 89 | 108 | 106 | 102 | 96 | 94 |
| Highland-3 | 125 | 127 | 129 | 131 | 131 | 132 | 137 | 139 | 141 | 143 | 137 | 143 | 144 | 147 | 148 |
| Antigo | 80 | 74 | 71 | 69 | 67 | 88 | 86 | 82 | 78 | 75 | 95 | 97 | 94 | 87 | 82 |
| Dodgeville | 49 | 42 | 37 | 34 | 32 | 58 | 51 | 46 | 41 | 39 | 66 | 62 | 57 | 50 | 46 |

3.1.6 Base de datos desarrollada en el Instituto Mexicano del Transporte

El Instituto Mexicano del Transporte ha estado desarrollando una base de datos de módulo de resiliencia de suelo finos (pasan la malla No. 4) desde el 2013. Esta base de datos cuenta con módulos de resiliencia de suelos clasificados como CH, MH, CL, ML, SC y SM.

Para desarrollar dicha base de datos, se llevaron a cabo muestreos de suelos en el estado de Querétaro, Guadalajara y Toluca. Después del muestreo, determinaron las propiedades índice y de compactación de los suelos utilizando procedimientos ASTM. Los resultados son resumidos en la Tabla 3.11.

Para llevar a cabo la prueba de módulo de resiliencia, primeramente se fabricaron los especímenes de prueba controlando las condiciones de compactación en el óptimo de compactación (W_{opt} y γ_{dmax} con respecto a la prueba Proctor estándar). Para cada suelo, la prueba de módulo de resiliencia fue evaluada en al menos tres réplicas, utilizando el protocolo de prueba NCHRP 1-28A; los resultados mostrados en la Tabla 3.12 son valores promedio.

Tabla 3.11. Propiedades índice y clasificación de los suelos estudiados.

| Banco | SUCS | LL (%) | LP (%) | IP (%) | P ₂₀₀ (%) | S _s | W _{opt} (%) | γ _d (kN/m ³) |
|-------------------------|------|--------|--------|--------|----------------------|----------------|----------------------|-------------------------------------|
| La Venta | CH | 69 | 23 | 46 | 87 | 2.56 | 33.5 | 12.49 |
| Santa Rosa | CH | 56 | 19.5 | 36.5 | 87.6 | 2.56 | 27.6 | 13.99 |
| La Cañada | CH | 55 | 30 | 25 | 81.4 | 2.59 | 31.25 | 13.07 |
| Pedro Escobedo | CH | 70 | 28 | 42 | 79.4 | 2.65 | 34.2 | 12.7 |
| Jalisco (arcilla negra) | CH | 94 | 32 | 62 | 95.6 | 2.74 | 36.1 | 12.49 |
| Jalisco (arcilla roja) | CH | 54 | 30 | 24 | 95.5 | 2.72 | 29.25 | 14.22 |
| San Pedro | CH | 81 | 36 | 45 | 95.1 | 2.67 | 39.5 | 11.57 |
| Arcilla IMT | CH | 66 | 25 | 41 | 85.7 | 2.61 | 30 | 13.3 |
| Mezcla 1 | SC | 49.5 | 21.5 | 28 | 29.2 | 2.61 | 28 | 13.7 |
| Aeropuerto | MH | 54 | 34 | 20 | 94.4 | 2.64 | 36.5 | 12.19 |
| La Galera | MH | 59 | 33 | 26 | 89.6 | 2.68 | 33.8 | 12.92 |
| T. Autobuses | ML | 46.5 | 36 | 10.5 | 92.4 | 2.66 | 36.6 | 12.21 |
| Misha | ML | 44 | 33 | 11 | 87 | 2.56 | 32 | 13.03 |
| Comercial | ML | NP | NP | NP | 57.2 | 2.38 | 28.6 | 12.69 |
| San José | SM | NP | NP | NP | 13.3 | 2.43 | 20.7 | 14.74 |
| Los Cues | SM | NP | NP | NP | 37 | 2.52 | 23.4 | 13.98 |
| Mezcla 2 | SC | 30.5 | 17.5 | 13 | 26.4 | 2.71 | 21 | 15.39 |
| Pueblo Nuevo | CL | 37 | 17 | 20 | 61.1 | 2.62 | 22.5 | 14.94 |

Nota: Mezcla 1: 50% Arena – 50% Arcilla Pedro Escobedo; Mezcla 2: 50% Arena – 50% limo Banco Misha

Tabla 3.11. Propiedades índice y clasificación de los suelos estudiados (continuación).

| Banco | SUCS | LL (%) | LP (%) | IP (%) | P ₂₀₀ (%) | Pasa No. 40 (%) | W _{opt} (%) | γ _d (kN/m ³) |
|----------|------|--------|--------|--------|----------------------|-----------------|----------------------|-------------------------------------|
| Conin | SM | NP | NP | NP | 32.86 | 67.18 | 24 | 13 |
| Conin 2 | SM | NP | NP | NP | 32.21 | 62.15 | 25 | 13.7 |
| HVS | SM | 51 | 29.67 | 21.33 | 43.66 | 56.88 | 25 | 14.5 |
| Mezcla 1 | SC | 34.45 | 17.5 | 16.95 | 43.24 | 65.18 | 23 | 15 |
| Mezcla 2 | SC | 35 | 22 | 13 | 33.86 | 61.38 | 21.5 | 15.6 |

Tabla 3.12. Módulos de resiliencia en condición óptima de compactación.

| Banco | Módulo de resiliencia (MPa) | | | | | | | | | | | | | | | |
|-----------------|-----------------------------|-----|-----|-----|---------------------|-----|-----|-----|---------------------|-----|-----|-----|---------------------|-----|-----|-----|
| | $\sigma_3 = 14$ kPa | | | | $\sigma_3 = 28$ kPa | | | | $\sigma_3 = 41$ kPa | | | | $\sigma_3 = 55$ kPa | | | |
| | $\sigma_d = 27$ kPa | 48 | 68 | 95 | 27 | 48 | 68 | 95 | 27 | 48 | 68 | 95 | 27 | 48 | 68 | 95 |
| La Venta | 83 | 78 | 76 | 73 | 94 | 86 | 82 | 78 | 97 | 89 | 85 | 81 | 99 | 92 | 87 | 83 |
| Santa Rosa | 90 | 85 | 83 | 80 | 105 | 98 | 94 | 89 | 112 | 105 | 100 | 94 | 117 | 109 | 104 | 98 |
| La Cañada | 95 | 91 | 86 | 80 | 109 | 102 | 96 | 89 | 117 | 108 | 102 | 94 | 122 | 114 | 107 | 99 |
| Pedro Escobedo | 104 | 93 | 90 | 87 | 128 | 107 | 102 | 96 | 136 | 115 | 107 | 101 | 141 | 119 | 111 | 104 |
| Jalisco (Negra) | 79 | 71 | 68 | 64 | 86 | 75 | 71 | 66 | 87 | 77 | 72 | 67 | 87 | 77 | 72 | 67 |
| Jalisco (Roja) | 72 | 63 | 53 | 43 | 92 | 80 | 66 | 53 | 104 | 89 | 75 | 59 | 109 | 95 | 80 | 64 |
| San Pedro | 109 | 102 | 100 | 99 | 129 | 118 | 114 | 111 | 139 | 127 | 121 | 116 | 146 | 132 | 124 | 119 |
| Arcilla IMT | 83 | 66 | 57 | 48 | 90 | 71 | 61 | 51 | 93 | 73 | 63 | 53 | 94 | 75 | 65 | 54 |
| Mezcla 1 | 104 | 101 | 100 | 101 | 122 | 118 | 115 | 113 | 135 | 128 | 123 | 120 | 143 | 134 | 128 | 124 |
| Aeropuerto | 62 | 58 | 53 | 47 | 71 | 65 | 59 | 51 | 76 | 70 | 63 | 55 | 80 | 73 | 67 | 59 |
| La Galera | 102 | 86 | 78 | 69 | 110 | 94 | 84 | 74 | 116 | 98 | 89 | 78 | 120 | 102 | 92 | 82 |
| T. Autobuses | 76 | 76 | 75 | 73 | 91 | 89 | 86 | 81 | 100 | 96 | 93 | 87 | 106 | 102 | 97 | 90 |
| Misha | 50 | 47 | 42 | 37 | 58 | 54 | 48 | 41 | 63 | 59 | 53 | 46 | 68 | 64 | 58 | 51 |
| Comercial | 95 | 65 | 56 | 50 | 119 | 87 | 77 | 69 | 137 | 106 | 95 | 86 | 154 | 124 | 111 | 102 |
| San José | 51 | 50 | 46 | 46 | 75 | 73 | 69 | 66 | 98 | 92 | 89 | 84 | 119 | 112 | 107 | 101 |
| Los Cues | 57 | 56 | 55 | 56 | 78 | 75 | 72 | 72 | 97 | 92 | 88 | 87 | 115 | 108 | 104 | 102 |
| Mezcla 2 | 63 | 55 | 47 | 40 | 76 | 66 | 58 | 50 | 87 | 76 | 67 | 60 | 96 | 84 | 75 | 68 |
| Pueblo Nuevo | 69 | 61 | 55 | 48 | 81 | 71 | 64 | 55 | 89 | 78 | 69 | 59 | 96 | 84 | 75 | 65 |
| Conin | 56 | 57 | 58 | 61 | 77 | 77 | 78 | 78 | 99 | 96 | 95 | 95 | 121 | 116 | 114 | 113 |
| Conin 2 | 49 | 53 | 54 | 55 | 73 | 76 | 77 | 76 | 95 | 96 | 97 | 96 | 116 | 116 | 116 | 114 |
| HVS | 60 | 58 | 56 | 51 | 81 | 75 | 69 | 62 | 94 | 85 | 78 | 70 | 103 | 94 | 87 | 79 |
| Mezcla 1 | 112 | 76 | 59 | 45 | 124 | 86 | 68 | 53 | 133 | 93 | 75 | 60 | 146 | 100 | 81 | 66 |
| Mezcla 2 | 39 | 35 | 32 | 31 | 53 | 48 | 44 | 40 | 67 | 59 | 54 | 50 | 80 | 71 | 65 | 60 |

3.2 Evaluación de las variables de entrada para las redes neuronales artificiales

3.2.1. Suelos tipo limo y arcillas

Como ya se observó en el capítulo dos; el módulo de resiliencia depende del peso volumétrico, del contenido de agua óptimo, del estado de esfuerzos, etc.

Para determinar las variables de entrada de las redes neuronales, se llevó a cabo un análisis de regresión combinando diferentes variables y para obtener el valor del coeficiente de determinación. La Tabla 3.13 muestra la nomenclatura de las variables; las Tablas 3.14, 3.15 y 3.16 muestran los análisis de regresión y sus correspondientes coeficientes de correlación.

Tabla 3.13. Nomenclatura de las variables utilizadas

| Símbolo | Parámetro | Observaciones |
|------------------|---|---|
| LL (%) | Límite líquido | --- |
| LP (%) | Límite plástico | --- |
| IP (%) | Índice de plasticidad | IP = LL-LP |
| IG | Índice de grupo | $IG = (P_{200}-35)[0.2+0.005(LL-40)]+0.01(P_{200}-15)(IP-10)$ |
| C | Compacidad | $C = (\gamma_d)/(Ss\gamma_w)$ |
| γ_w | Peso volumétrico del agua | --- |
| P200 (%) | Porcentaje de suelo que pasa la malla No. 200. | --- |
| γ_d | Peso volumétrico seco máximo (kN/m ³) | --- |
| W _{opt} | Contenido de agua óptimo | --- |
| τ (kPa) | Esfuerzo octaédrico | --- |
| σ_d (kPa) | Esfuerzo desviador | $\sigma_d = \sigma_1 - \sigma_3$ |
| σ_3 (kPa) | Presión de confinamiento | --- |
| θ (kPa) | Esfuerzo volumétrico | $\theta = \sigma_1 + \sigma_2 + \sigma_3$ |
| wIP | Porcentaje que pasa la malla No. 200 multiplicado por el índice de plasticidad (uno de los dos parámetros es decimal) | --- |
| Fina | Porcentaje de arena fina | --- |
| Media | Porcentaje de arena media | --- |
| Gruesa | Porcentaje de arena gruesa | --- |
| P ₁₀ | Porcentaje de suelo que pasa la mallas No. 10 | --- |
| P ₄₀ | Porcentaje de suelo que pasa la malla No. 40 | --- |

Tabla 3.14. Análisis de regresión lineal utilizando 4 variables de entrada.

| Variables de entrada | | | | Salida | R ² |
|----------------------|------------------|------------------|-----------------------|--------|----------------|
| IP | P ₂₀₀ | σ_d | σ_3 | Mr | 0.1685 |
| IP | P ₂₀₀ | W _{opt} | σ_d / σ_3 | Mr | 0.139 |
| γ_d | P ₂₀₀ | W _{opt} | σ_d / σ_3 | Mr | 0.2592 |
| γ_d | P ₂₀₀ | IP | σ_d / σ_3 | Mr | 0.1502 |
| γ_d | W _{opt} | IP | σ_d / σ_3 | Mr | 0.263 |
| IP | P ₂₀₀ | θ | τ | Mr | 0.2 |

Tabla 3.15. Análisis de regresión lineal utilizando 5 variables de entrada.

| Variables de entrada | | | | | Salida | R ² |
|----------------------|------------------|------------------|------------|-----------------------|--------|----------------|
| IP | P ₂₀₀ | W _{opt} | γ_d | σ_d / σ_3 | Mr | 0.2674 |
| IP | P ₂₀₀ | W _{opt} | σ_d | σ_3 | Mr | 0.174 |
| LP | P ₂₀₀ | W _{opt} | σ_d | σ_3 | Mr | 0.161 |
| wIP | θ | W _{opt} | C | τ | Mr | 0.296 |
| W _{opt} | θ | γ_d | C | τ | Mr | 0.309 |
| P ₂₀₀ | θ | IG | C | τ | Mr | 0.2345 |
| IP | P ₂₀₀ | IG | C | θ | Mr | 0.19 |
| IP | P ₂₀₀ | IG | C | τ | Mr | 0.311 |

Tabla 3.16. Análisis de regresión lineal utilizando 6 variables de entrada.

| Variables de entrada | | | | | | Salida | R ² |
|----------------------|--------------------------------|-----------------|--------------------------------|----------------|--------------------------------|--------|----------------|
| IP | P ₂₀₀ | w _{op} | γ _d | C | σ _d /σ ₃ | Mr | 0.2815 |
| IP | P ₂₀₀ | w _{op} | γ _d | C | τ | Mr | 0.271 |
| IP | P ₂₀₀ | w _{op} | σ _d /σ ₃ | C | τ | Mr | 0.2829 |
| IP | P ₂₀₀ | γ _d | σ _d /σ ₃ | C | τ | Mr | 0.188 |
| IP | P ₂₀₀ | θ | σ _d /σ ₃ | C | τ | Mr | 0.1933 |
| IP | P ₂₀₀ | θ | w _{op} | C | τ | Mr | 0.2968 |
| σ ₃ | wIP | θ | w _{op} | C | τ | Mr | 0.2432 |
| IP | P ₂₀₀ | θ | γ _d | C | τ | Mr | 0.195 |
| IP | P ₂₀₀ | θ | IG | C | τ | Mr | 0.206 |
| IP | γ _d | θ | IG | C | τ | Mr | 0.196 |
| IP | w _{op} | θ | IG | C | τ | Mr | 0.2966 |
| P ₂₀₀ | w _{op} | θ | IG | C | τ | Mr | 0.296 |
| wIP | w _{op} | θ | IG | C | τ | Mr | 0.2997 |
| IP | σ _d /σ ₃ | θ | IG | C | τ | Mr | 0.1937 |
| IP | P ₂₀₀ | C | IG | γ _d | θ | Mr | 0.1736 |
| IP | P ₂₀₀ | C | IG | γ _d | τ | Mr | 0.301 |

De acuerdo con los análisis anteriores, se puede comprobar que la interacción del módulo de resiliencia con las propiedades físicas y estado de esfuerzos del suelo es un fenómeno no lineal; ya que los coeficientes de determinación son muy bajos. A pesar de esto, el análisis proporciona una idea de cuáles son las variables más significativas.

Tomando en cuenta los coeficientes de correlación, las variables de entrada de la red neuronal para arcillas y limos fueron:

- Índice de plasticidad (IP)
- Contenido de humedad óptimo (w_{opt})
- Porcentaje que pasa la malla No. 200 (P₂₀₀)
- Compacidad (C)
- Esfuerzo de confinamiento (σ₃)
- Esfuerzo desviador (σ_d)

Es importante mencionar que la red neuronal se entrenó con propiedades que varían dentro de ciertos rangos (Tabla 3.17). Por lo tanto, es recomendable que el usuario de la herramienta que se propone en este documento la utilice con suelos cuyas propiedades se encuentran dentro de los rangos indicados.

Tabla 3.17. Rango de las variables y módulo de resiliencia.

| Variables | Rangos | |
|----------------------|----------|----------|
| | Inferior | Superior |
| IP (%) | 9 | 62 |
| P ₂₀₀ (%) | 54 | 98 |
| w _{opt} (%) | 13 | 39 |
| C (%) | 43 | 71 |
| σ_3 (kPa) | 14 | 55 |
| σ_d (kPa) | 13 | 95 |
| Mr (MPa) | 15 | 192 |

3.2.2. Suelos arenosos

Para las arenas, se realizó el mismo análisis; pero en este caso se incluyeron variables como el porcentaje de arena fina, media y gruesa; tomando en cuenta que la norma ASTM D2487-06 define los rangos de estas arenas de acuerdo con lo indicado en la Tabla 3.18.

Tabla 3.18. Tamaño de las arenas de acuerdo con la norma ASTM D2487-06.

| Arena | Tamaño |
|--------|--|
| Gruesa | Pasa la malla No. 4 (4.75 mm) y es retenida en la malla No. 10 (2 mm) |
| Media | Pasa la malla No. 10 (2 mm) y es retenida en la malla No. 40 (0.425 mm) |
| Fina | Pasa la malla No. 40 (0.425 mm) y es retenida en la malla No. 200 (0.074 mm) |

En las Tablas 3.19, 3.20 y 3.21 se resumen los coeficientes de correlación obtenidos del análisis de regresión.

Tabla 3.19. Análisis de regresión lineal para arenas utilizando cuatro variables de entrada.

| Variables de entrada | | | | Salida | R ² |
|----------------------|---------------------|------------|------------------|--------|----------------|
| P ₂₀₀ | σ_3 | σ_d | W _{opt} | Mr | 0.1782 |
| P ₁₀ | σ_3 | σ_d | W _{opt} | Mr | 0.2453 |
| P ₄₀ | σ_3 | σ_d | W _{opt} | Mr | 0.2272 |
| P ₂₀₀ | σ_d/σ_3 | C | W _{opt} | Mr | 0.1424 |
| P ₄₀ | σ_d/σ_3 | C | W _{opt} | Mr | 0.192 |
| P ₁₀ | σ_d/σ_3 | C | W _{opt} | Mr | 0.2024 |
| P ₁₀ | σ_d/σ_3 | γ_d | W _{opt} | Mr | 0.1991 |
| P ₄₀ | σ_d/σ_3 | γ_d | W _{opt} | Mr | 0.1878 |
| P ₂₀₀ | σ_d/σ_3 | γ_d | W _{opt} | Mr | 0.1275 |
| Gruesa | σ_3 | σ_d | W _{opt} | Mr | 0.1821 |
| Media | σ_3 | σ_d | W _{opt} | Mr | 0.1692 |
| fina | σ_3 | σ_d | W _{opt} | Mr | 0.2571 |

Tabla 3.20. Análisis de regresión lineal utilizando cinco variables de entrada.

| Variables de entrada | | | | | Salida | R ² |
|----------------------|------------------|------------|------------|-----------------|--------|----------------|
| W _{opt} | P ₂₀₀ | σ_3 | σ_d | C | Mr | 0.1993 |
| W _{opt} | P ₂₀₀ | σ_3 | σ_d | γ_d | Mr | 0.1824 |
| W _{opt} | P ₂₀₀ | σ_3 | σ_d | P ₁₀ | Mr | 0.2569 |
| γ_d | P ₂₀₀ | σ_3 | σ_d | P ₁₀ | Mr | 0.2558 |
| W _{opt} | P ₂₀₀ | σ_3 | σ_d | P ₄₀ | Mr | 0.2571 |
| W _{opt} | P ₂₀₀ | σ_3 | σ_d | P ₄₀ | Mr | 0.2454 |
| W _{opt} | C | σ_3 | σ_d | P ₄₀ | Mr | 0.234 |
| W _{opt} | C | σ_3 | σ_d | P ₁₀ | Mr | 0.2497 |
| W _{opt} | C | σ_3 | σ_d | Gruesa | Mr | 0.1955 |
| W _{opt} | C | σ_3 | σ_d | Media | Mr | 0.1867 |
| W _{opt} | C | σ_3 | σ_d | Fina | Mr | 0.2641 |
| W _{opt} | γ_d | σ_3 | σ_d | Gruesa | Mr | 0.1866 |
| W _{opt} | γ_d | σ_3 | σ_d | Media | Mr | 0.174 |
| W _{opt} | γ_d | σ_3 | σ_d | Fina | Mr | 0.2572 |

Tabla 3.21. Análisis de regresión lineal utilizando seis variables de entrada.

| Variables de entrada | | | | | | Salida | R ² |
|----------------------|------------------|--------------------------------|----------------|-----------------|-----------------|--------|----------------|
| W _{opt} | P ₂₀₀ | σ ₃ | σ _d | C | P ₁₀ | Mr | 0.2622 |
| W _{opt} | P ₄₀ | σ ₃ | σ _d | C | P ₁₀ | Mr | 0.2498 |
| W _{opt} | P ₂₀₀ | σ ₃ | σ _d | C | P ₄₀ | Mr | 0.2643 |
| W _{opt} | P ₂₀₀ | σ ₃ | σ _d | P ₁₀ | P ₄₀ | Mr | 0.2619 |
| W _{opt} | P ₂₀₀ | σ _d /σ ₃ | C | P ₁₀ | P ₄₀ | Mr | 0.2236 |
| W _{opt} | P ₂₀₀ | σ _d /σ ₃ | Y _d | P ₁₀ | P ₄₀ | Mr | 0.2189 |

Para este caso, las variables de entrada de la red neuronal y sus rangos aparecen en la Tabla 3.22.

Tabla 3.22. Variables de entrada y rangos para arenas

| Variables | Rangos | |
|------------------------------|----------|----------|
| | Inferior | Superior |
| W _{opt} (%) | 8 | 25 |
| Porcentaje de arena fina (%) | 8 | 76 |
| σ ₃ (kPa) | 14 | 55 |
| σ _d (kPa) | 12 | 97 |
| Mr (MPa) | 29 | 144 |

4 Redes neuronales artificiales para estimar el módulo de resiliencia

4.1. Normalización de las bases de datos

Después de obtener las bases de datos, se realiza la normalización conocida también como escalamiento. Las razones para escalar los datos son: (a) dar a cada entrada la misma importancia y (b) para evitar la saturación prematura de funciones de activación sigmoideal. Los datos de salida o de destino se escalan si las funciones de activación de salida tienen un rango limitado y los objetivos no escalados no coinciden con este rango.

Las funciones de activación tienen la finalidad de crear salidas ante ciertas entradas de la red. En este trabajo se utilizó la función tangente hiperbólica en las neuronas ocultas; y para la neurona de salida, se utilizó la función lineal. Estas funciones se eligieron porque son un buen aproximador universal.

En este trabajo se utilizó la normalización de $[-1,1]$ por las funciones de activación elegidas. La base se normaliza de acuerdo con:

$$p_{[-1,1]} = \left(\frac{2(p-p_{\min})}{p_{\max}-p_{\min}} \right) - 1 \quad (4.1)$$

Donde:

$p_{[-1,1]}$ = Entrada o salida normalizada en el rango $[-1,1]$

p_{\max} = Entrada o salida máxima

p_{\min} = Entrada o salida mínima

4.2. Estructura de las redes neuronales

En la literatura recomiendan, como primer intento, construir una red neuronal pequeña en la que se tenga solo una capa de neuronas ocultas con un número pequeño de neuronas que puede ser determinada de la siguiente forma:

$$N_o = \frac{N_i + N_f}{2} \quad (4.2)$$

Donde:

N_o = Número de neuronas ocultas

N_i = Número de neuronas iniciales = Variables de entrada para los dos casos

N_f = Número de neuronas finales = M_r

La red es entrenada con una arquitectura sencilla. Hasta determinado número de corridas se entrena y se prueba una siguiente arquitectura incrementándole una neurona a la vez a la capa oculta y así sucesivamente hasta que parezca grande el número de estas (sin incrementar a un número excesivo de neuronas ya que esto produciría un sobreaprendizaje). Es hasta entonces cuando se le adiciona una capa de neuronas ocultas comenzado ambas de nuevo con un número pequeño de neuronas como se indica en la Tabla 4.1.

Tabla 4.1. Tipos de arquitecturas de las redes neuronales.

| Tipo de suelo | Arcillas y limos | Arenas |
|---|------------------|--------|
| Tipos de Arquitecturas utilizadas en las redes neuronales | 6-3-1 | 4-3-1 |
| | 6-4-1 | 4-4-1 |
| | 6-5-1 | 4-5-1 |
| | 6-6-1 | 4-6-1 |
| | 6-7-1 | 4-7-1 |
| | 6-8-1 | 4-8-1 |

Nota: La nomenclatura 6-3-1 significa 6 neuronas de entrada, 3 neuronas ocultas y 1 salida.

4.3. Desarrollo de las redes neuronales artificiales

En la creación de las redes neuronales en suelos tipo CH, CL, MH y ML se utilizaron 698 ejemplos de módulo de resiliencia. Estos quedaron repartidos en 596 ejemplos; 39 suelos para la creación y entrenamiento. Para suelos del tipo SM, SC y SP se utilizaron 334 ejemplos; 23 suelos para el entrenamiento de la red.

Existen diferentes tipos de arquitecturas; por lo tanto, existen diversas matrices de pesos. Para la primera capa de pesos, el tamaño de dicha matriz está dado por la cantidad de neuronas que contenga y las entradas a la red. En la segunda capa de neuronas la matriz de pesos debe tener un rango delimitado por el número de neuronas en cada capa, mientras que la matriz de los pesos propios de cada neurona será dimensionada de acuerdo con su número de neuronas.

La interacción entre las entradas y la primera capa oculta es de la forma:

$$W_{n \times 5} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ \dots & \dots & \dots & \dots & \dots \\ W_{n1} & W_{n2} & W_{n3} & W_{n4} & W_{n5} \end{bmatrix} \quad b_{n \times 1} = \begin{bmatrix} W_{11} \\ W_{21} \\ W_{31} \\ \dots \\ W_{n1} \end{bmatrix} \quad (4.3)$$

La interacción entre la capa oculta y la salida es:

$$W_{1 \times n} = [w_{11} \ w_{12} \ w_{13} \ \dots \ w_{1n}] \quad b_{1 \times 1} = [w_{11}] \quad (4.4)$$

Una vez creadas las distintas redes neuronales de acuerdo con las arquitecturas especificadas anteriormente, se procede a obtener la salida de la red utilizando los 596 ejemplos para suelos CH, CL, MH y ML y 334 ejemplos para suelos SM, SC y SP. La salida de la red se obtuvo de la siguiente manera:

$$s[N,1] = f(Wp+b) \quad (4.5)$$

Donde:

N = Número de ejemplos en la base de datos

Después de cada ejemplo de entrenamiento presentado a la red, las diferencias entre las salidas de la red y las esperadas son calculadas; es entonces cuando se obtiene la función de rendimiento de la red, la cual es el error cuadrático medio. Esta función es fundamental para los diversos algoritmos de aprendizaje de las redes.

4.4. Entrenamiento de las redes neuronales artificiales

Una vez que la red neuronal es desarrollada, se somete a un proceso de aprendizaje que determina los parámetros adecuados o pesos que participan en la red con la finalidad de obtener una salida deseada. Para enseñar a una red neuronal, es necesario entrenarla con un conjunto de datos; el cual consta de señales de entrada que son asignadas con salidas deseadas. Este entrenamiento es un proceso iterativo y en cada iteración los pesos de los nodos se modifican usando nuevos datos del conjunto para el entrenamiento.

A continuación, se describen los distintos tipos de entrenamiento utilizados.

4.4.1 Entrenamiento utilizando el algoritmo de Levenberg-Marquardt

En este caso, primero se calculó la matriz del jacobiano $J(n,m)$ de cada uno de los pesos de la red. La matriz jacobiana es la matriz formada por las derivadas

parciales de primer orden de la función de error que genera las salidas de la red respecto a las salidas deseadas.

$$J(n,m) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \frac{\partial e_1}{\partial w_m} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \frac{\partial e_2}{\partial w_m} \\ \frac{\partial e_3}{\partial w_1} & \frac{\partial e_3}{\partial w_2} & \dots & \frac{\partial e_3}{\partial w_m} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_n}{\partial w_1} & \frac{\partial e_n}{\partial w_2} & \dots & \frac{\partial e_n}{\partial w_m} \end{bmatrix} \quad (4.6)$$

Donde:

$E(n,1)$ = Vector del error en la salida de la red = $S(n,1)-Y(n,1)$

w_m = Peso de la m -ésima neurona

n = Número de ejemplos en la base de datos (596 ejemplos para suelos CH, CL, MH y ML y 334 ejemplos para suelos SM, SC y SP)

Una vez generada la matriz del jacobiano, se propuso un valor inicial μ . Este valor es conocido como un parámetro de aprendizaje de la red. En el algoritmo de Levenberg-Marquardt clásico el valor de μ es un valor constante. El parámetro de aprendizaje por utilizar es:

$$\mu = 0.01 \quad (4.7)$$

Con la constante de aprendizaje inicial y la matriz del jacobiano, se obtiene el cambio que tendrán los pesos $\Delta w(m,1)$ en esta iteración.

$$\Delta w(m,1) = (J(m,n) J(n,m) + \mu I(m,m))^{-1} J(m,n) e(n,1) \quad (4.8)$$

Donde:

$J(n,m)$ = Matriz del jacobiano de las derivadas de primer orden

$J(m,n)$ = Matriz transpuesta del jacobiano de las derivadas de primer orden

μ = Parámetro de aprendizaje

$I(m,m)$ = Matriz identidad

$e(n,1)$ = Vector del error en la salida de la red

n = Número de ejemplos en la base de datos (596 ejemplos y 334 ejemplos)

m = Número de pesos en la red (dependiendo de la arquitectura)

Después se procede a la actualización de los pesos

$$w(m,1)_i = w(m,1)_{i-1} + \Delta w(m,1) \quad (4.9)$$

Donde:

$w(m,1)_i$ = Vector de pesos actuales de la red

$w(m,1)_{(i-1)}$ = Vector de pesos anteriores a la actualización
 m = Número de pesos en la red

Con los nuevos pesos actualizados se procede a encontrar la nueva salida de la red

$$S(n,1)_i = f(Wp+b) \quad (4.10)$$

Donde:

$S(n,1)_i$ = Salida de la red con los pesos actualizados

$F(Wp+b)$ = Función que genera la salida de la red

Con la nueva salida de la red, se procede a calcular la nueva suma de errores cuadráticos $SSE(1)_i$.

$$SSE(1)_i = (S(n,1)_i - Y(n,1))^2 \quad (4.11)$$

Donde:

$S(n,1)_i$ = Vector de la nueva salida de la red

$Y(n,1)$ = Vector de salida deseada u objetivo

n = Número de ejemplos en la base de datos

Si $SSE(1)_{i+1} < SSE(1)_i$ entonces

$$\mu = \mu/10$$

Se calcula $J(n,m)$,

Se recalcula $\Delta w(m,1) = (J(m,n) J(n,m) + \mu I(m,m))^{-1} J(m,n) e(n,1)$

Si no

$$\mu = \mu*10$$

Se recalcula $\Delta w(m,1) = (J(m,n) J(n,m) + \mu I(m,m))^{-1} J(m,n) e(n,1)$

Fin.

4.4.2 Entrenamiento utilizando el algoritmo de Levenberg-Marquardt modificado

Para este caso, primero se calculó la matriz del jacobiano $J(n,m)$ de cada uno de los pesos de la red. La matriz jacobiana es la matriz formada por las derivadas parciales de primer orden de la función de error que genera las salidas de la red respecto a las salidas deseadas.

$$J(n,m) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \frac{\partial e_1}{\partial w_m} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \frac{\partial e_2}{\partial w_m} \\ \frac{\partial e_3}{\partial w_1} & \frac{\partial e_3}{\partial w_2} & \frac{\partial e_3}{\partial w_m} \\ \dots & \dots & \dots \\ \frac{\partial e_n}{\partial w_1} & \frac{\partial e_n}{\partial w_2} & \frac{\partial e_n}{\partial w_m} \end{bmatrix} \quad (4.12)$$

Donde:

$E(n,1)$ = Vector del error en la salida de la red = $S(n,1)-Y(n,1)$

w_m = Peso de la m -ésima neurona

n = Número de ejemplos en la base de datos (596 ejemplos para suelos CH, CL, MH y ML y 334 ejemplos para suelos SM, SC y SP)

Una vez que fue generada la matriz del jacobiano, se propuso un valor inicial μ . Este valor es conocido como un parámetro de aprendizaje de la red. μ en el algoritmo de Levenberg-Marquardt clásico es un valor constante; en esta investigación fue modificado -según las investigaciones hechas por (Suratgar et al., 2007)-, con la finalidad de obtener una mejor predicción. El parámetro de aprendizaje utilizado es:

$$\mu = 0.01 e(1,n) e(n,1) \quad (4.13)$$

Donde:

$e(1,n)$ = Vector transpuesto del error en la salida de la red

$e(n,1)$ = Vector del error en la salida de la red

Con la constante de aprendizaje inicial y la matriz del jacobiano se obtiene el cambio que tendrán los pesos $\Delta w(m,1)$ en esta iteración.

$$\Delta w(m,1) = (J(m,n) J(n,m) + \mu I(m,m))^{-1} J(m,n) e(n,1) \quad (4.14)$$

Donde:

$J(n,m)$ = Matriz del jacobiano de las derivadas de primer orden

$J(m,n)$ = Matriz transpuesta del jacobiano de las derivadas de primer orden

μ = Parámetro de aprendizaje

$I(m,m)$ = Matriz identidad

$e(n,1)$ = Vector del error en la salida de la red

n = Número de ejemplos en la base de datos a utilizar de acuerdo al tipo de suelo

m = Número de pesos en la red

Enseguida se procede a la actualización de los pesos.

$$w(m,1)_i = w(m,1)_{i-1} + \Delta w(m,1) \quad (4.15)$$

Donde:

$w(m,1)_i$ = Vector de pesos actuales de la red

$w(m,1)_{i-1}$ = Vector de pesos anteriores a la actualización

m = Número de pesos en la red

Con los nuevos pesos actualizados se procede a encontrar la nueva salida de la red.

$$S(n,1)_i = f(Wp+b) \quad (4.16)$$

Donde:

$S(n,1)_i$ = Salida de la red con los pesos actualizados

$f(Wp+b)$ = Función que genera la salida de la red

Con la nueva salida de la red se calcula la nueva suma de errores cuadráticos $SSE(1)_i$.

$$SSE(1)_i = (S(n,1)_i - Y(n,1))^2 \quad (4.17)$$

Donde:

$S(n,1)_i$ = Vector de la nueva salida de la red

$Y(n,1)$ = Vector de salida deseada u objetivo

n = Número de ejemplos en los dos tipos base de datos

Con el nuevo error de salida se recalcula el valor de μ y los pesos nuevamente. Las iteraciones terminan hasta el momento en que se alcance la tolerancia deseada.

4.5. Validación de los modelos de redes neuronales

Una vez terminado el proceso de entrenamiento con el algoritmo deseado, se obtuvieron los pesos de las neuronas e interconexiones entre ellas. Con estos se puede estimar el módulo de resiliencia y compararlo con una base de datos nueva que contenga suelos no utilizados en el entrenamiento. La comparación se realiza para evaluar la capacidad de la red de estimar valores desconocidos.

Para la validación, se utilizaron siete suelos entre limos y arcillas y seis suelos tipo arena.

4.6 Valores de M_r estimados versus datos de laboratorio

El proceso de entrenamiento de las redes neuronales termina hasta un cierto número de iteraciones en el que se obtiene el valor mínimo de la función de rendimiento (error cuadrático medio) utilizada. La finalidad de realizar este proceso iterativo es que la red neuronal llegue a un mínimo global de la superficie del error evitando que la red tenga un sobreaprendizaje ante la predicción de situaciones diferentes a las presentadas en el entrenamiento.

Las Figuras 4.1 y 4.2 presentan el proceso de entrenamiento de las redes, con la mejor predicción.

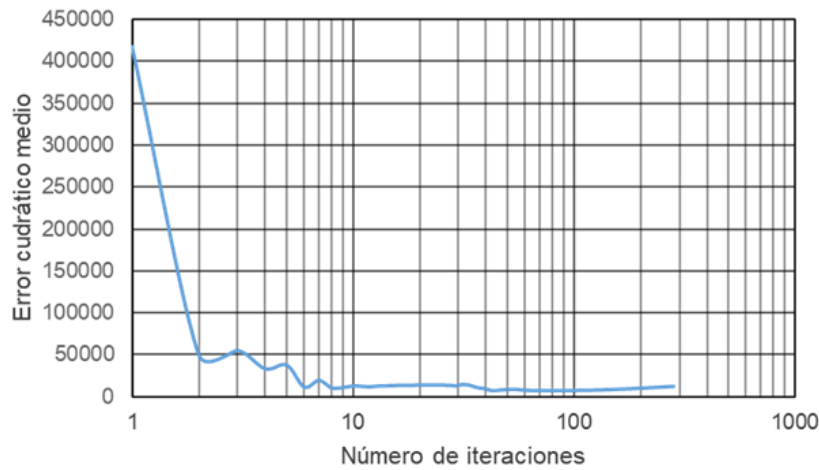


Figura 4.1. Número de iteraciones del modelo de red neuronal con arquitectura 6-6-1 (arcillas y limos).

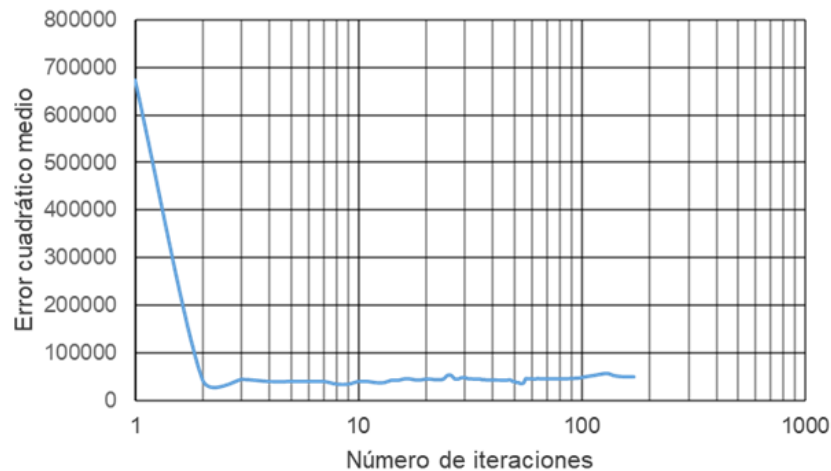


Figura 4.2. Número de iteraciones del modelo de red neuronal con arquitectura 4-7-1 (arenas).

En las Figuras 4.1 y 4.2 se observa que los algoritmos de Quasi-Newton tienen una rápida convergencia para la predicción del módulo de resiliencia ya que en aproximadamente 70 y 90 iteraciones se logra un valor muy bajo de la función de rendimiento para suelos tipo arcilla y limos y arenas, respectivamente.

Después del entrenamiento, se llevó a cabo una comparativa de los valores estimados con respecto a los de laboratorio. Se probaron 3 tipos de algoritmo de Quasi-Newton: Gauss-Newton, Levenberg-Marquardt y Levenberg-Marquardt modificado. Para medir la capacidad de predicción del modelo de redes neuronales se utilizó el valor de R^2 . De acuerdo con Bayomy et al., (2012), los valores de la Tabla 4.2 indican cómo se califica un modelo de acuerdo con el valor de R^2 . Los valores de esta tabla también fueron utilizados para calificar la red neuronal.

Tabla 4.2. Criterio del parámetro R^2 de bondad de ajuste.

| Criterio | R^2 |
|-----------|------------|
| Excelente | > 0.9 |
| Bueno | 0.7 - 0.89 |
| Regular | 0.4 - 0.69 |
| Mala | 0.2 - 0.39 |
| Muy mala | < 0.19 |

Las Tablas 4.3 a 4.6 muestran un resumen de los resultados de los modelos de redes neuronales desarrollados. Los resultados muestran que los valores del coeficiente de determinación R^2 para la validación del modelo de redes en suelo tipo arcilla y limo se encuentran entre 0.3159 y 0.8143. Solo un modelo tiene un valor de R^2 mayor a 0.8; nueve modelos tienen valores de R^2 mayores de 0.5 pero menores de 0.8 y dos modelos con valores por debajo del 0.5. Para suelos arenosos los resultados de validación de los modelos de redes neuronales muestran que el coeficiente de determinación varía en un rango entre -0.997 y 0.884. Dos modelos tienen valores de R^2 mayores a 0.8; cinco modelos tienen valores mayores a 0.5 pero menores a 0.8 y cinco modelos con valores por debajo del 0.5.

Tabla 4.3. Coeficientes de determinación (R^2) para el entrenamiento de las redes neuronales artificiales para suelos tipo arcilla y limos.

| Arquitectura | Gauss-Newton | Levenberg-Marquardt | Levenberg-Marquardt modificado |
|--------------|----------------|---------------------|--------------------------------|
| 6-3-1 | Sin resultado* | 0.6771 | 0.6996 |
| 6-4-1 | Sin resultado* | 0.7075 | 0.7449 |
| 6-5-1 | Sin resultado* | 0.8685 | 0.8713 |
| 6-6-1 | Sin resultado* | 0.9192 | 0.9256 |
| 6-7-1 | Sin resultado* | 0.937 | 0.9397 |
| 6-8-1 | Sin resultado* | 0.7634 | 0.6634 |

Tabla 4.4. Coeficientes de determinación (R^2) para la validación de las redes neuronales artificiales para suelos tipo arcilla y limo.

| Arquitectura | Gauss-Newton | Levenberg-Marquardt | Levenberg-Marquardt modificado |
|--------------|----------------|---------------------|--------------------------------|
| 6-3-1 | Sin resultado* | 0.691 | 0.7894 |
| 6-4-1 | Sin resultado* | 0.7287 | 0.4832 |
| 6-5-1 | Sin resultado* | 0.6023 | 0.6034 |
| 6-6-1 | Sin resultado* | 0.6301 | 0.8143 |
| 6-7-1 | Sin resultado* | 0.7611 | 0.3159 |
| 6-8-1 | Sin resultado* | 0.7809 | 0.6706 |

Tabla 4.5. Coeficientes de determinación (R^2) para el entrenamiento de las redes neuronales artificiales para suelos tipo arena.

| Arquitectura | Gauss-Newton | Levenberg-Marquardt | Levenberg-Marquardt modificado |
|--------------|----------------|---------------------|--------------------------------|
| 4-3-1 | Sin resultado* | 0.0343 | 0.545 |
| 4-4-1 | Sin resultado* | 0.5646 | 0.8886 |
| 4-5-1 | Sin resultado* | 0.7431 | 0.8631 |
| 4-6-1 | Sin resultado* | 0.9221 | 0.9311 |
| 4-7-1 | Sin resultado* | 0.9365 | 0.9146 |
| 4-8-1 | Sin resultado* | 0.0701 | 0.4467 |

Tabla 4.6. Coeficientes de determinación (R^2) para la validación de las redes neuronales artificiales para suelos tipo arena.

| Arquitectura | Gauss-Newton | Levenberg-Marquardt | Levenberg-Marquardt modificado |
|--------------|----------------|---------------------|--------------------------------|
| 4-3-1 | Sin resultado* | -0.997 | 0.5578 |
| 4-4-1 | Sin resultado* | 0.5196 | 0.7971 |
| 4-5-1 | Sin resultado* | 0.776 | 0.0476 |
| 4-6-1 | Sin resultado* | 0.4109 | 0.4848 |
| 4-7-1 | Sin resultado* | 0.8596 | 0.884 |
| 4-8-1 | Sin resultado* | -0.007 | 0.6519 |

De acuerdo con el criterio de R^2 de la Tabla 4.2 y utilizando los resultados de la validación, se observa que la arquitectura 6-6-1 (modelo de Levenberg-Marquardt modificado) y la arquitectura 6-3-1 (mismo modelo), proporcionan una predicción de M_r en suelos tipo arcilla y limo que puede ser calificada como buena. En el caso de las arenas, la arquitectura 4-4-1 y 4-7-1 son un modelo bueno para predecir el M_r . En las Figuras 4.3 a 4.6 se grafica el entrenamiento y validación de los modelos de redes neuronales que utilizaron los algoritmos de Levenberg-Marquardt modificado, ya que mostraron la mejor predicción del módulo de resiliencia. El modelo para suelos tipo CH, CL, MH y ML consiste de seis variables de entrada, una capa oculta con seis neuronas intermedias y una variable de salida. Mientras para suelos tipo SM, SC, SP el modelo consiste de cuatro variables de entrada, una capa oculta con siete neuronas intermedias y una variable de salida.

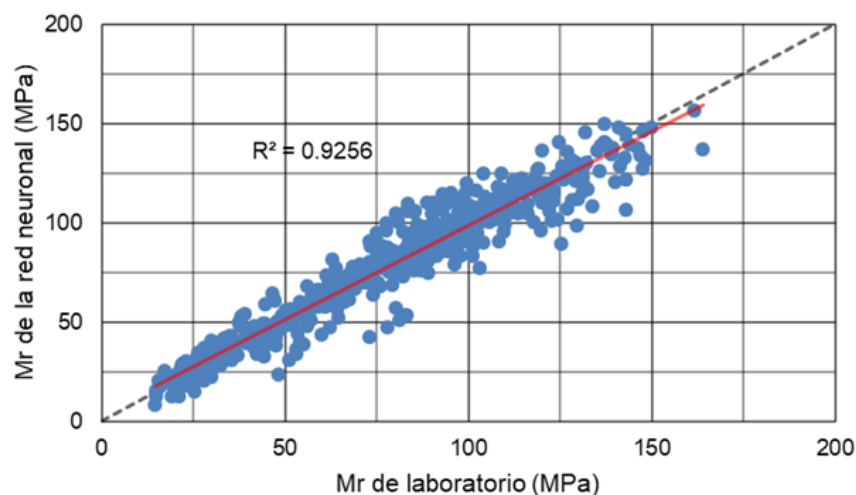


Figura 4.3. Entrenamiento del modelo de red neuronal con arquitectura 6-6-1 para suelos tipo CH, CL, MH y ML.

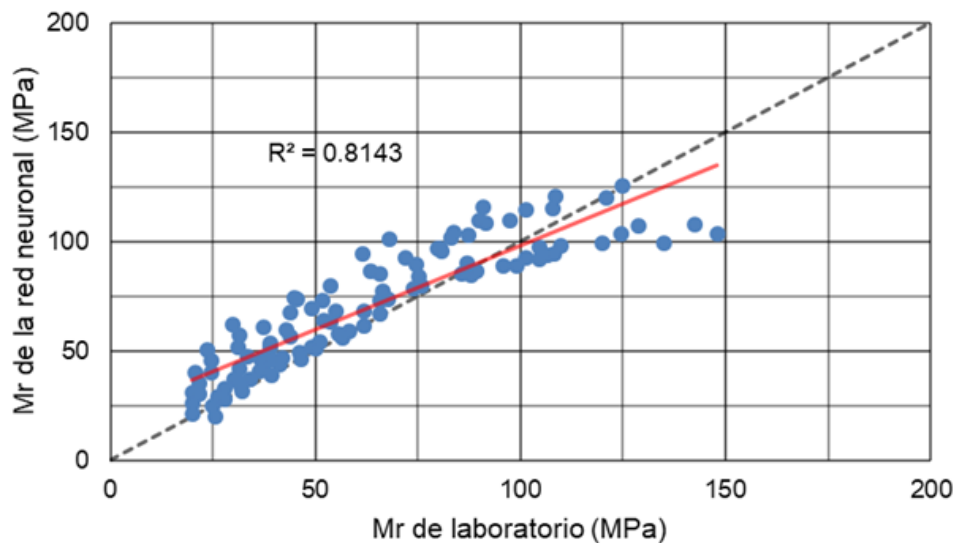


Figura 4.4. Validación del modelo de red neuronal con arquitectura 6-6-1 para suelos tipo CH, CL, MH y ML.

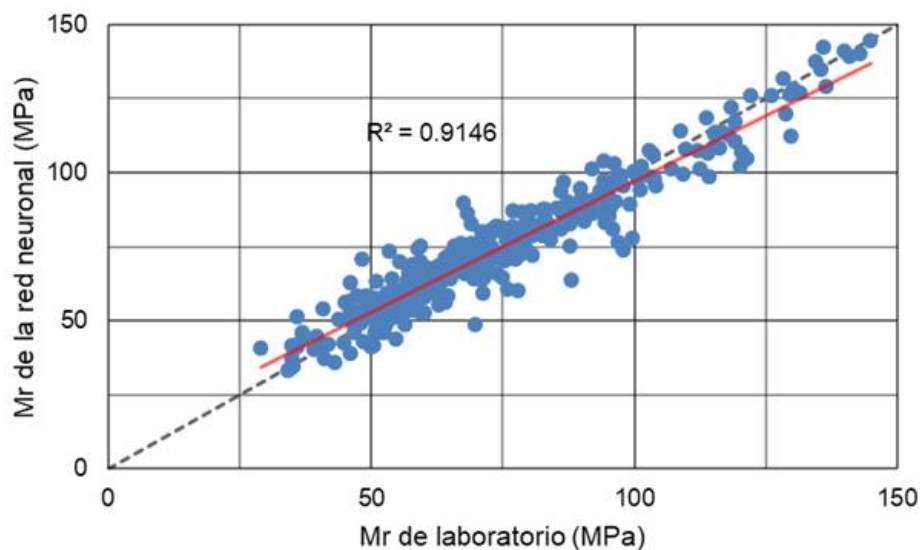


Figura 4.5. Entrenamiento del modelo de red neuronal con arquitectura 4-7-1 para suelos tipo SM, SC y SP.

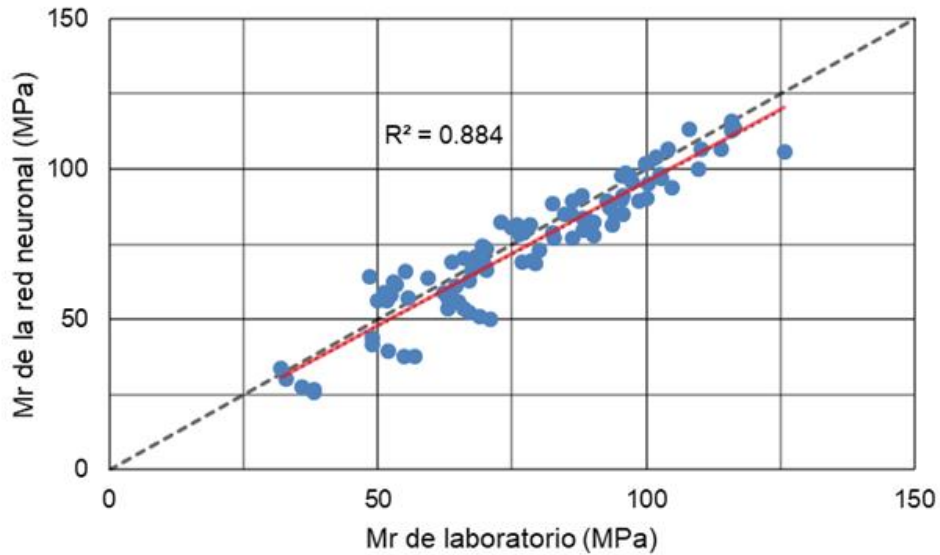


Figura 4.6. Validación del modelo de red neuronal con arquitectura 4-7-1 para suelos tipo SM, SC y SP.

4.7. Aplicación para estimar el módulo de resiliencia

Una vez que se obtuvieron los modelos de redes neuronales, se procedió a la realización de un programa ejecutable utilizando el lenguaje computacional Visual Basic y realizado a través de Visual Studio 2015. La Figura 4.7 muestra la ventana principal del programa.



Figura 4.7. Interfaz inicial del programa para la predicción del Mr en suelos finos compactados.

El programa tiene embebidos los modelos desarrollados presentados anteriormente. Al seleccionar cualquiera de los dos modelos aparecerá una interfaz como muestran las Figuras 4.8 y 4.9.

Form2

Mr - PREDICTOR

Índice de plasticidad (%) ?

Porcentaje que pasa la malla No. 200 (%) ?

Contenido de agua óptimo (%) ?

Compacidad (%) ?

Esfuerzo de confinamiento (kPa) ?

Esfuerzo desviador (kPa) ?

Mr

Figura 4.8. Interfaz para la obtención del módulo de resiliencia en suelos tipo CH, CL, MH y ML.

Form3

Mr - PREDICTOR

Contenido de agua óptimo (%) ?

Porcentaje de arena fina (%) ?

Esfuerzo de confinamiento (kPa) ?

Esfuerzo desviador (kPa) ?

Mr

Figura 4.9. Interfaz para la obtención del módulo de resiliencia en suelos tipo SM, SC y SP.

La aplicación anterior solo requiere del conocimiento de propiedades básicas de los suelos y el estado de esfuerzos para poder estimar el módulo de resiliencia. La aplicación se publica en conjunto con este documento de tal forma que los interesados puedan utilizarla y tener una idea de los módulos de resiliencia que puedan ser empleados cuando lleven a cabo diseños preliminares de pavimentos.

Conclusiones

Con base en los resultados del presente estudio, se puede concluir lo siguiente:

- Las propiedades IP, P_{200} , w_{opt} , C, σ_3 y σ_d son variables significativas para la predicción del módulo de resiliencia en suelos finos tipo CH, CL, MH y ML.
- Las propiedades % de arena fina, w_{opt} , σ_3 y σ_d son variables significativas para la predicción del módulo de resiliencia en suelos finos tipo SM, SC y SP.
- El uso de redes neuronales artificiales de aprendizaje supervisado para la modelación del fenómeno de módulo de resiliencia mostró ser eficiente con respecto al análisis estadístico de regresión multivariable. El análisis de regresión lineal múltiple probó que el fenómeno de módulo de resiliencia muestra algunas deficiencias en la modelación de este.
- El algoritmo de Levenberg-Marquardt modificado muestra mayor eficiencia para predecir el módulo de resiliencia. El Algoritmo de Levenberg-Marquardt modificado presenta una mejor optimización en la superficie del error, ya que los valores de la constante de aprendizaje son variables y tienden a cambiar respecto a la nueva predicción del modelo; mientras que en el algoritmo clásico de Levenberg-Marquardt los valores de este parámetro de aprendizaje son constantes.
- Para los suelos tipo CH, CL, MH, y ML el modelo con arquitectura 6-6-1 alcanzó una mejor optimización, pues mostró un coeficiente de determinación (R^2) de 0.93 para la predicción de suelos dentro de la base de datos utilizada; mientras que para la validación con suelos diferentes proporcionó un coeficiente de determinación (R^2) de 0.8143. De acuerdo con la prueba de bondad de ajuste el modelo es bueno ante la predicción de nuevos ejemplos.
- Para suelos tipo SM, SC y SP, el modelo neuronal con arquitectura 4-7-1 obtuvo una mejor optimización al mostrar un coeficiente de determinación (R^2) de 0.9146 para la predicción de suelos de entrenamiento; mientras que para la predicción del Mr de nuevos suelos se obtuvo un coeficiente (R^2) de 0.884; el cual es bueno.
- Se puede concluir que las redes neuronales son una técnica que ayuda a la predicción de fenómenos no lineales; pues a diferencia de las técnicas de regresión, las redes neuronales artificiales plantean un sistema de

ecuaciones no lineales, y a mayor número de ejemplos se puede obtener una mejor aproximación. Dentro de las desventajas de los modelos de redes neuronales se puede observar la extrapolación y la predicción cercana a los límites de las variables de entrada.

- El programa ejecutable presentado, se creó con la finalidad que el Mr se obtenga rápidamente. Este programa no pretende sustituir la prueba triaxial sino que puede ser usado como una herramienta preliminar en los diseños de pavimentos. En futuras investigaciones se pretende aumentar la base de datos de suelos finos así como de granulares para enriquecer este proyecto y con ello tener un rango de suelos más amplio respecto al que se tiene actualmente.

Recomendaciones

En la creación de las redes neuronales artificiales, se recomienda tener una base de datos con una correcta distribución; ya que si esto no se cumple, las redes neuronales pueden llegar a predecir correctamente suelos de entrenamiento, pero no podrá predecir situaciones futuras.

Para redes neuronales de aprendizaje supervisado en las que se requiera optimizar parámetros de entrada, es necesaria una correcta selección de las funciones de activación; la función tangente hiperbólica es una de las más utilizadas en la literatura de redes neuronales.

Si se requiere trabajar con el algoritmo de Levenberg-Marquardt modificado para cualquier investigación, se recomienda programar el algoritmo; ya que en los programas computacionales que emplean redes neuronales no utilizan este algoritmo de entrenamiento.

Para investigaciones futuras, se recomienda que la base de datos sea de una sola fuente, ya que durante la selección de las bases de datos se encontraron muchas variables involucradas en su generación; lo cual complicó el estudio.

Bibliografía y Referencias

Abo-Hashema, M. (2009). Artificial Neural Network Approach for Overlay Design of Flexible Pavements. The International Arab Journal of Information Technology. Vol. 6, No. 2. April, 2009.

Alemán, M. D. (2017). Técnicas de inteligencia artificial aplicadas a problemas de Ingeniería civil. Universidad & Ciencia. Vol 6. No. especial UNICA. Pp. 164-175. Noviembre (2017).

Beltran, G. y Romo, M. (2011). Estimación de módulos elásticos en pavimentos usando redes neuronales artificiales. Pan-Am CGS. Geotechnical Conference. Toronto, Canadá.

Birkan, M. (2008). Analysis of jointed plain concrete pavement systems with nondestructive test results using artificial neural networks. Iowa State University.

Barksdale, R. G. (1971). "Compressive Stress Pulse Times in Flexible Pavements for Use in Dynamic Testing" Highway Research Record 345, Highway Research Board: 32–44.

Brown, S. F. (1973). "Determination of Young's Modulus for Bituminous Materials in Pavement Design", Highway Research Record 431, Highway Research Board: 38-49.

Caglar, N., Arman, H. (2007), "The applicability of Neural Networks in the determination of soil profiles", Bull Eng Geol Environ, pp: 295–301.

Castro, G.S. (1998), "Aplicación de las redes neuronales para la obtención del módulo de resiliencia utilizando información producida mediante deflectómetros de impacto". Tesis de maestría, Universidad Autónoma de Chihuahua, Chihuahua, Chihuahua.

Coleri, E., Guler, M., Gungor, A.G., Harvey, J.T. (2010), "Prediction of subgrade resilient modulus using genetic algorithm and curve shifting methodology: Alternative to nonlinear constitutive models", Transportation Research Record (2170): 64-73.

Curiel, N.P. (2015), "Catálogo de módulos de resiliencia para suelos finos y materiales granulares". Informe técnico de proyecto de residencia, Instituto Tecnológico de Tepic, Tepic, Nayarit.

Das, S.K. (2013), "Artificial Neural Networks in Geotechnical Engineering: Modeling and Application Issues", *Metaheuristics in Water, Geotechnical and Transport Engineering*, DOI 10.1016/B978-0-12-398296-4.00010-6.

De la Fuente, J.L. (1997), "Técnicas de cálculo para sistemas de ecuaciones, programación lineal y programación entera", Universidad Politécnica de Madrid. España.

Demuth, H., Beale, M., Hagan, M. (2008). *Neural network toolbox 6. User's guide*.

Dione, A., Fall, M., Bâ, M. (2013), "Estimation of Resilient Modulus of Unbound Granular Materials from Senegal (West Africa)" *Geomaterials*, 3(4):172-178. Doi: 10.4236/gm.2013.34022.

Duan, Z. H., Kou, S.C. y Poon, C.S. (2013). Using artificial neural networks for predicting the elastic modulus of recycled aggregate concrete. *Construction and Building Materials*. 44 (2013). Pp. 524-532.

Farajzadeh, J., Fard, A.F., Lotfi, S. (2014), "Modeling of monthly rainfall and runoff of Urmia lake basin using feed-forward neural network and time series analysis model", *Water resources and industry* (7-8): 38-48.

Garnica, P., Pérez, N y Gómez, J.A. (2000), "Módulos de resiliencia en suelos finos y materiales granulares", *Publicación Técnica No. 142*. Sanfandila, Querétaro.

González, J.L. (2014), "Catálogo de módulos de resiliencia para suelos finos (fase 1. Materiales de Querétaro y Guadalajara)". Tesis de licenciatura, Universidad Autónoma de Querétaro, Querétaro, Querétaro.

Gopinath, D.I., Dwarakish, G.S. (2015), "Wave prediction using neural networks at New Mangalore Port along west coast of India", *International conference on water resources, coastal and ocean engineering*, DOI: 10.1016/j.aqpro.2015.02.020. (4):143-150.

Honarmand-Ebrahimi, A. (2006), "Regression and neural network modeling of resilient modulus based on routine soil properties and stress states", Tesis de doctorado, University of Oklahoma, Oklahoma, United States of America.

Ibrahim, S. F., Kadhim, A. J. y Athman, H. A. (2017). Prediction of resilient modulus using artificial neural network and extreme learning machine from index properties for Iraqi subgrade soil. *Imperial Journal of Interdisciplinary Research*. Vol. 3. Issue 2. 2017.

Huang, Y. H. (2004), "Pavement Analysis and Design", 2da edición, Pearson Hall, Estados Unidos de América: 279-330.

Karray, F., Silva, C. (2004), "Soft Computing and Intelligent Systems Design", Pearson Addison Wesley. England.

Khan, N., Gaurav, D., Kandl, T. (2013), "Performance evaluation of Levenberg-Marquardt technique in error reduction for diabetes condition classification", International conference on Computational Science (18): 2629-2637.

Kim, S-H., Yang, J., Jeong, J-H. (2014). Prediction of Subgrade Resilient Modulus using artificial neural network. KSCE Journal of Civil Engineering (0000) 00(0): 1-8.

Khoury, C.N., Khoury, N.N., Miller, G.A. (2011), "Effect of Cyclic Suction History (Hydraulic Hysteresis) on Resilient Modulus of Unsaturated Fine-Grained Soil", Transportation Research Record: Journal of the Transportation Research Board, 2232(1): 68-75.

Leiva, F., Aguilar, J. P., y Loría, L. G. (2013). Desarrollo de un modelo mejorado para la predicción de modulo dinámico E^* para mezclas asfálticas. XVII Congreso Ibero-Latinoamericano del asfalto (CILA). Antigua Guatemala, Guatemala.

Maher, A., Gucunski, N., Bennert, T. (2000). "Resilient modulus properties of New Jersey subgrade soils" New Jersey Department of Transportation, Division of Research and Technology, U.S Department of Transportation.

Malla, R. B., Joshi, S. (2006). "Establish subgrade support values for typical soils in New England", New England Transportation Consortium, University of Connecticut, Department of Civil and Environmental Engineering.

Malla, R.B., Joshi, S. (2007), "Resilient Modulus Prediction Models based on Analysis of LTPP Data for Subgrade Soils and Experimental Verification", Journal of Transportation Engineering Vol 133, No. 9.

Mao, B. (1995), "Predicting resilient modulus of highway subgrade soils in Ohio", Tesis de maestría, Ohio University, Ohio, United States of America.

Matich, D. (2001). Redes neuronales: Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional. Facultad Regional Rosario. Departamento de Ingeniería Química.

National Cooperative Highway Research Program, (2008) "Guide for mechanistic-empirical design of new and rehabilitated pavement structures", submitted by ARA, Inc., Division), National Research Council, Transportation Research Board, Final Rep. NCHRP 1-37A (submitted by ARA, Inc., ERES Consultants Division.

Navarro, C. J. I. (2017). Desarrollo de un modelo de predicción del escalonamiento en pavimentos rígidos. Memoria de Ingeniería Civil. Universidad Técnica Federico Snta María. Julio 2017.

Nawi, N.M., Khan, A., Rehman, M.Z. (2013), "A new Levenberg Marquardt based back propagation algorithm trained with cuckoo search", The 4th International conference on electrical engineering and informatics (11):18-23.

Osorio, J.R. (2015), "Módulos de resiliencia para materiales granulares". Tesis de maestría, Instituto Politécnico Nacional, México, México.

Park, H. I., Kweon, G.C., y Lee, R. (2009). Prediction of resilient modulus of granular subgrade soils and subbase materials using artificial neural network. Road Material sand Pavement Design. Pp. 647-665.

Pérez, G. N., Garnica, A. P., Flores, M. B., Mendoza, P. I., y Reyes, R. M. (2012). Comportamiento de suelos finos compactados con altos niveles de esfuerzo cortante. Publicación técnica No. 350. Instituto Mexicano del Transporte.

Ponce, P. (2010), "Inteligencia Artificial con Aplicaciones a la Ingeniería", 1ra Edición, Alfaomega, Grupo Editor, S.A de C.V., México, ISBN: 978-607-7854-83-8.

Puppala, A.J. (2008), "Estimating Stiffness of Subgrade and Unbound Materials for Pavement Design", NCHRP Synthesis 382, Transportation Research Board, National Research Council, Washington, DC. 139 pp.

Rani, Ch.S., Vaddi, P.K., Togati, N.V. (2013), "Predicting settlement of shallow foundations using Neural Networks", International Journal of Innovative Technology and Exploring Engineering (IJITEE), 3: 123–130.

Richardson, D., Petry, T., Ge, L., Han, Y., Lusher, S. M. (2009). "Resilient moduli of Typical Missouri Soils and Unbound Granular Base Materials", Missouri University of Science and Technology and Missouri Department of Transportation.

Shahin, M.A., Maier, H.R., Jaksa, M.B. (2002), "Predicting settlement of shallow foundations using neural network", J. Geotech. Geoenviron. Eng. ASCE. 128 (9): 785-793.

Shahin, M.A., Jaksa, M.B., Maier, H.R. (2008), "State of the Art of Artificial Neural Networks in Geotechnical Engineering", The Electronic Journal of Geotechnical Engineering Bouquest08: 1-26.

Suratgar, A., Tavakoli, M., Hoseinabadi, A. (2007), "Modified Levenberg-Marquardt method for neural networks training", International Journal of Computer, Control, Quantum and Information Engineering (1): 1719-1721.

Taghavifar, H., Mardani, A., Karim-Maslak, H., Kalbkhani, H. (2013), "Artificial neural network estimation of wheel rolling resistance in clay loam soil", Applied soft computing, Doi: 10.1016/j.asoc.2013.03.017. (13): 3544-3551.

Tarawneh, B., Nazzal, M. (2014), "Optimization of resilient modulus prediction from FWD results using artificial neural network", *Periodica Polytechnica*, 58(2):143-154. Doi 10.3311/PPci.2201.

Titi, H. H., Elias, M. B., Helwany, S. (2011). "Determination of Typical Resilient Modulus Values for Selected Soils in Wisconsin" Wisconsin Highway Research Program Project ID 0092-03-11, University of Wisconsin-Milwaukee, Department of Civil Engineering and Mechanics.

Tutumluer, E. y Meier, R. (1996). Attempt at resilient modulus modeling using artificial neural networks. *Transportation Research Record* 1540. Pp. 1-6.

Yang, J., Lu, J. J., G., M. (2003). Application of neural network models for forecasting of pavement crack index and pavement condition rating. University of South Florida. March, 2003.

Yasin, Y., Ahmad, F., Ghaffari-Moghaddam, M., Khajeh, M. (2014), "Application of a hybrid artificial neural network-genetic algorithm approach to optimize the lead ions removal from aqueous solutions using intercalated tartrate -Mg-Al layered double hydroxides", *Environmental Nanotechnology, Monitoring & Management*, Doi: 10.1016/j.enmm.2014.03.001. (1): 2-7.

Yu, H., Wilamowski, B. M. (2011). Levenberg-marquardt training. *Industrial Electronics Handbook*, 5, 12-1.

Roberts, C.A., y –attah-Okine, N. (1998). A comparative analysis of two Artificial Neural Networks using Pavement Performance Prediction. *Computer-Aided and Infrastructure Engineering* 13 (1998). 339-348.

Vila, Z. R. (2017). Análisis del comportamiento lineal de la estructura del pavimento flexible mediante redes neuronales en la carretera Panamericana Norte. Tesis de licenciatura. Universidad Nacional del Centro de Perú.

Anexo 1. Programa

Sub Programa()

'Programa creado por Ing. Luis Enrique Gómez Carreño

'Tesis de maestría en Geotecnia

'Universidad Autónoma de Querétaro – IMT

'Red neuronal artificial tipo 6-6-1 utilizando el algoritmo de Levenberg-Marquardt

'6 = número de variables de entrada

'364 = número de datos a analizar

Dim We(49, 1) As Double 'Pesos y biases de la red

Dim f(596, 1) As Double 'Salida de la red neuronal

Dim w(596, 1) As Double 'Error

Dim y(596, 1) As Double 'Salida deseada

Dim J(596, 49) As Double 'Jacobiano

Dim JT(49, 596) As Double 'Jacobiano transpuesto

Dim JTJ(49, 49) As Double

Dim JTe(49, 1) As Double

Dim WW(49, 1) As Double

Dim Mul(49, 49) As Double

Dim MJ(49, 49) As Double

Dim ID(49, 49) As Double

'Pesos y biases de la red neuronal

For i = 1 To 49

We(i, 1) = Cells(2 + i, 11)

Next i

'Salida de la red y jacobianos

ssa = 0

For i = 1 To 596

p = Cells(3 + i, 2)

q = Cells(3 + i, 3)

r = Cells(3 + i, 4)

s = Cells(3 + i, 5)

t = Cells(3 + i, 6)

u = Cells(3 + i, 7)

y(i, 1) = Cells(3 + i, 8) 'Salida deseada

$$A = p * We(1, 1) + q * We(2, 1) + r * We(3, 1) + s * We(4, 1) + t * We(5, 1) + u * We(6, 1) + We(7, 1)$$

$$b = p * We(8, 1) + q * We(9, 1) + r * We(10, 1) + s * We(11, 1) + t * We(12, 1) + u * We(13, 1) + We(14, 1)$$

$$c = p * We(15, 1) + q * We(16, 1) + r * We(17, 1) + s * We(18, 1) + t * We(19, 1) + u * We(20, 1) + We(21, 1)$$

$$h = p * We(22, 1) + q * We(23, 1) + r * We(24, 1) + s * We(25, 1) + t * We(26, 1) + u * We(27, 1) + We(28, 1)$$

$$l = p * We(29, 1) + q * We(30, 1) + r * We(31, 1) + s * We(32, 1) + t * We(33, 1) + u * We(34, 1) + We(35, 1)$$

$$n = p * We(36, 1) + q * We(37, 1) + r * We(38, 1) + s * We(39, 1) + t * We(40, 1) + u * We(41, 1) + We(42, 1)$$

$$d = (\text{Exp}(A) - \text{Exp}(-1 * A)) / (\text{Exp}(A) + \text{Exp}(-1 * A))$$

$$e = (\text{Exp}(b) - \text{Exp}(-1 * b)) / (\text{Exp}(b) + \text{Exp}(-1 * b))$$

$$g = (\text{Exp}(c) - \text{Exp}(-1 * c)) / (\text{Exp}(c) + \text{Exp}(-1 * c))$$

$$o = (\text{Exp}(h) - \text{Exp}(-1 * h)) / (\text{Exp}(h) + \text{Exp}(-1 * h))$$

$$m = (\text{Exp}(l) - \text{Exp}(-1 * l)) / (\text{Exp}(l) + \text{Exp}(-1 * l))$$

$$ni = (\text{Exp}(n) - \text{Exp}(-1 * n)) / (\text{Exp}(n) + \text{Exp}(-1 * n))$$

$$d1 = (\text{Exp}(A) + \text{Exp}(-1 * A))$$

$$d2 = (\text{Exp}(A) - \text{Exp}(-1 * A))$$

$$e1 = (\text{Exp}(b) + \text{Exp}(-1 * b))$$

$$e2 = (\text{Exp}(b) - \text{Exp}(-1 * b))$$

$$g1 = (\text{Exp}(c) + \text{Exp}(-1 * c))$$

$$g2 = (\text{Exp}(c) - \text{Exp}(-1 * c))$$

$$o1 = (\text{Exp}(h) + \text{Exp}(-1 * h))$$

$$o2 = (\text{Exp}(h) - \text{Exp}(-1 * h))$$

$$m1 = (\text{Exp}(l) + \text{Exp}(-1 * l))$$

$$m2 = (\text{Exp}(l) - \text{Exp}(-1 * l))$$

$$ni1 = (\text{Exp}(n) + \text{Exp}(-1 * n))$$

$$ni2 = (\text{Exp}(n) - \text{Exp}(-1 * n))$$

'Salida de la red neuronal

$$f(i, 1) = d * We(43, 1) + e * We(44, 1) + g * We(45, 1) + o * We(46, 1) + m * We(47, 1) + ni * We(48, 1) + We(49, 1)$$

'Error en la salida

$$w(i, 1) = f(i, 1) - y(i, 1)$$

$$ssa = ssa + (w(i, 1) * w(i, 1))$$

Next i

$$'Cells(26, 10) = ss(1)$$

$$mu = 0.01$$

For nu = 1 To 280

For i = 1 To 22

Cells(27 + i, 20 + nu) = We(i, 1)

Next i

For i = 1 To 596

p = Cells(3 + i, 2)

q = Cells(3 + i, 3)

r = Cells(3 + i, 4)

s = Cells(3 + i, 5)

t = Cells(3 + i, 6)

u = Cells(3 + i, 7)

y(i, 1) = Cells(3 + i, 8) 'Salida deseada

A = p * We(1, 1) + q * We(2, 1) + r * We(3, 1) + s * We(4, 1) + t * We(5, 1) + u * We(6, 1) + We(7, 1)

b = p * We(8, 1) + q * We(9, 1) + r * We(10, 1) + s * We(11, 1) + t * We(12, 1) + u * We(13, 1) + We(14, 1)

c = p * We(15, 1) + q * We(16, 1) + r * We(17, 1) + s * We(18, 1) + t * We(19, 1) + u * We(20, 1) + We(21, 1)

h = p * We(22, 1) + q * We(23, 1) + r * We(24, 1) + s * We(25, 1) + t * We(26, 1) + u * We(27, 1) + We(28, 1)

l = p * We(29, 1) + q * We(30, 1) + r * We(31, 1) + s * We(32, 1) + t * We(33, 1) + u * We(34, 1) + We(35, 1)

n = p * We(36, 1) + q * We(37, 1) + r * We(38, 1) + s * We(39, 1) + t * We(40, 1) + u * We(41, 1) + We(42, 1)

d = (Exp(A) - Exp(-1 * A)) / (Exp(A) + Exp(-1 * A))

e = (Exp(b) - Exp(-1 * b)) / (Exp(b) + Exp(-1 * b))

g = (Exp(c) - Exp(-1 * c)) / (Exp(c) + Exp(-1 * c))

o = (Exp(h) - Exp(-1 * h)) / (Exp(h) + Exp(-1 * h))

m = (Exp(l) - Exp(-1 * l)) / (Exp(l) + Exp(-1 * l))

ni = (Exp(n) - Exp(-1 * n)) / (Exp(n) + Exp(-1 * n))

d1 = (Exp(A) + Exp(-1 * A))

d2 = (Exp(A) - Exp(-1 * A))

e1 = (Exp(b) + Exp(-1 * b))

e2 = (Exp(b) - Exp(-1 * b))

g1 = (Exp(c) + Exp(-1 * c))

g2 = (Exp(c) - Exp(-1 * c))

$$\begin{aligned} o1 &= (\text{Exp}(h) + \text{Exp}(-1 * h)) \\ o2 &= (\text{Exp}(h) - \text{Exp}(-1 * h)) \\ m1 &= (\text{Exp}(l) + \text{Exp}(-1 * l)) \\ m2 &= (\text{Exp}(l) - \text{Exp}(-1 * l)) \\ ni1 &= (\text{Exp}(n) + \text{Exp}(-1 * n)) \\ ni2 &= (\text{Exp}(n) - \text{Exp}(-1 * n)) \end{aligned}$$

'Obtener la matriz del jacobiano

$$\begin{aligned} J(i, 1) &= ((\text{We}(43, 1)) / (d1 * d1)) * ((d1 * d1 * p) - (p * d2 * d2)) \\ J(i, 2) &= ((\text{We}(43, 1)) / (d1 * d1)) * ((d1 * d1 * q) - (q * d2 * d2)) \\ J(i, 3) &= ((\text{We}(43, 1)) / (d1 * d1)) * ((d1 * d1 * r) - (r * d2 * d2)) \\ J(i, 4) &= ((\text{We}(43, 1)) / (d1 * d1)) * ((d1 * d1 * s) - (s * d2 * d2)) \\ J(i, 5) &= ((\text{We}(43, 1)) / (d1 * d1)) * ((d1 * d1 * t) - (t * d2 * d2)) \\ J(i, 6) &= ((\text{We}(43, 1)) / (d1 * d1)) * ((d1 * d1 * u) - (u * d2 * d2)) \\ J(i, 7) &= ((\text{We}(43, 1)) / (d1 * d1)) * ((d1 * d1 * 1) - (1 * d2 * d2)) \end{aligned}$$

$$\begin{aligned} J(i, 8) &= ((\text{We}(44, 1)) / (e1 * e1)) * ((e1 * e1 * p) - (p * e2 * e2)) \\ J(i, 9) &= ((\text{We}(44, 1)) / (e1 * e1)) * ((e1 * e1 * q) - (q * e2 * e2)) \\ J(i, 10) &= ((\text{We}(44, 1)) / (e1 * e1)) * ((e1 * e1 * r) - (r * e2 * e2)) \\ J(i, 11) &= ((\text{We}(44, 1)) / (e1 * e1)) * ((e1 * e1 * s) - (s * e2 * e2)) \\ J(i, 12) &= ((\text{We}(44, 1)) / (e1 * e1)) * ((e1 * e1 * t) - (t * e2 * e2)) \\ J(i, 13) &= ((\text{We}(44, 1)) / (e1 * e1)) * ((e1 * e1 * u) - (u * e2 * e2)) \\ J(i, 14) &= ((\text{We}(44, 1)) / (e1 * e1)) * ((e1 * e1 * 1) - (1 * e2 * e2)) \end{aligned}$$

$$\begin{aligned} J(i, 15) &= ((\text{We}(45, 1)) / (g1 * g1)) * ((g1 * g1 * p) - (p * g2 * g2)) \\ J(i, 16) &= ((\text{We}(45, 1)) / (g1 * g1)) * ((g1 * g1 * q) - (q * g2 * g2)) \\ J(i, 17) &= ((\text{We}(45, 1)) / (g1 * g1)) * ((g1 * g1 * r) - (r * g2 * g2)) \\ J(i, 18) &= ((\text{We}(45, 1)) / (g1 * g1)) * ((g1 * g1 * s) - (s * g2 * g2)) \\ J(i, 19) &= ((\text{We}(45, 1)) / (g1 * g1)) * ((g1 * g1 * t) - (t * g2 * g2)) \\ J(i, 20) &= ((\text{We}(45, 1)) / (g1 * g1)) * ((g1 * g1 * u) - (u * g2 * g2)) \\ J(i, 21) &= ((\text{We}(45, 1)) / (g1 * g1)) * ((g1 * g1 * 1) - (1 * g2 * g2)) \end{aligned}$$

$$\begin{aligned} J(i, 22) &= ((\text{We}(46, 1)) / (o1 * o1)) * ((o1 * o1 * p) - (p * o2 * o2)) \\ J(i, 23) &= ((\text{We}(46, 1)) / (o1 * o1)) * ((o1 * o1 * q) - (q * o2 * o2)) \\ J(i, 24) &= ((\text{We}(46, 1)) / (o1 * o1)) * ((o1 * o1 * r) - (r * o2 * o2)) \\ J(i, 25) &= ((\text{We}(46, 1)) / (o1 * o1)) * ((o1 * o1 * s) - (s * o2 * o2)) \\ J(i, 26) &= ((\text{We}(46, 1)) / (o1 * o1)) * ((o1 * o1 * t) - (t * o2 * o2)) \\ J(i, 27) &= ((\text{We}(46, 1)) / (o1 * o1)) * ((o1 * o1 * u) - (u * o2 * o2)) \\ J(i, 28) &= ((\text{We}(46, 1)) / (o1 * o1)) * ((o1 * o1 * 1) - (1 * o2 * o2)) \end{aligned}$$

$$\begin{aligned} J(i, 29) &= ((\text{We}(47, 1)) / (m1 * m1)) * ((m1 * m1 * p) - (p * m2 * m2)) \\ J(i, 30) &= ((\text{We}(47, 1)) / (m1 * m1)) * ((m1 * m1 * q) - (q * m2 * m2)) \\ J(i, 31) &= ((\text{We}(47, 1)) / (m1 * m1)) * ((m1 * m1 * r) - (r * m2 * m2)) \\ J(i, 32) &= ((\text{We}(47, 1)) / (m1 * m1)) * ((m1 * m1 * s) - (s * m2 * m2)) \\ J(i, 33) &= ((\text{We}(47, 1)) / (m1 * m1)) * ((m1 * m1 * t) - (t * m2 * m2)) \end{aligned}$$

$$J(i, 34) = ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * u) - (u * m2 * m2))$$

$$J(i, 35) = ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * 1) - (1 * m2 * m2))$$

$$J(i, 36) = ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * p) - (p * ni2 * ni2))$$

$$J(i, 37) = ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * q) - (q * ni2 * ni2))$$

$$J(i, 38) = ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * r) - (r * ni2 * ni2))$$

$$J(i, 39) = ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * s) - (s * ni2 * ni2))$$

$$J(i, 40) = ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * t) - (t * ni2 * ni2))$$

$$J(i, 41) = ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * u) - (u * ni2 * ni2))$$

$$J(i, 42) = ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * 1) - (1 * ni2 * ni2))$$

$$J(i, 43) = d2 / d1$$

$$J(i, 44) = e2 / e1$$

$$J(i, 45) = g2 / g1$$

$$J(i, 46) = o2 / o1$$

$$J(i, 47) = m2 / m1$$

$$J(i, 48) = ni2 / ni1$$

$$J(i, 49) = 1$$

'Matriz transpuesta del Jacobiano

$$JT(1, i) = ((We(43, 1)) / (d1 * d1)) * ((d1 * d1 * p) - (p * d2 * d2))$$

$$JT(2, i) = ((We(43, 1)) / (d1 * d1)) * ((d1 * d1 * q) - (q * d2 * d2))$$

$$JT(3, i) = ((We(43, 1)) / (d1 * d1)) * ((d1 * d1 * r) - (r * d2 * d2))$$

$$JT(4, i) = ((We(43, 1)) / (d1 * d1)) * ((d1 * d1 * s) - (s * d2 * d2))$$

$$JT(5, i) = ((We(43, 1)) / (d1 * d1)) * ((d1 * d1 * t) - (t * d2 * d2))$$

$$JT(6, i) = ((We(43, 1)) / (d1 * d1)) * ((d1 * d1 * u) - (u * d2 * d2))$$

$$JT(7, i) = ((We(43, 1)) / (d1 * d1)) * ((d1 * d1 * 1) - (1 * d2 * d2))$$

$$JT(8, i) = ((We(44, 1)) / (e1 * e1)) * ((e1 * e1 * p) - (p * e2 * e2))$$

$$JT(9, i) = ((We(44, 1)) / (e1 * e1)) * ((e1 * e1 * q) - (q * e2 * e2))$$

$$JT(10, i) = ((We(44, 1)) / (e1 * e1)) * ((e1 * e1 * r) - (r * e2 * e2))$$

$$JT(11, i) = ((We(44, 1)) / (e1 * e1)) * ((e1 * e1 * s) - (s * e2 * e2))$$

$$JT(12, i) = ((We(44, 1)) / (e1 * e1)) * ((e1 * e1 * t) - (t * e2 * e2))$$

$$JT(13, i) = ((We(44, 1)) / (e1 * e1)) * ((e1 * e1 * u) - (u * e2 * e2))$$

$$JT(14, i) = ((We(44, 1)) / (e1 * e1)) * ((e1 * e1 * 1) - (1 * e2 * e2))$$

$$JT(15, i) = ((We(45, 1)) / (g1 * g1)) * ((g1 * g1 * p) - (p * g2 * g2))$$

$$JT(16, i) = ((We(45, 1)) / (g1 * g1)) * ((g1 * g1 * q) - (q * g2 * g2))$$

$$JT(17, i) = ((We(45, 1)) / (g1 * g1)) * ((g1 * g1 * r) - (r * g2 * g2))$$

$$JT(18, i) = ((We(45, 1)) / (g1 * g1)) * ((g1 * g1 * s) - (s * g2 * g2))$$

$$JT(19, i) = ((We(45, 1)) / (g1 * g1)) * ((g1 * g1 * t) - (t * g2 * g2))$$

$$JT(20, i) = ((We(45, 1)) / (g1 * g1)) * ((g1 * g1 * u) - (u * g2 * g2))$$

$$JT(21, i) = ((We(45, 1)) / (g1 * g1)) * ((g1 * g1 * 1) - (1 * g2 * g2))$$

$$JT(22, i) = ((We(46, 1)) / (o1 * o1)) * ((o1 * o1 * p) - (p * o2 * o2))$$

$$JT(23, i) = ((We(46, 1)) / (o1 * o1)) * ((o1 * o1 * q) - (q * o2 * o2))$$

$$\begin{aligned} JT(24, i) &= ((We(46, 1)) / (o1 * o1)) * ((o1 * o1 * r) - (r * o2 * o2)) \\ JT(25, i) &= ((We(46, 1)) / (o1 * o1)) * ((o1 * o1 * s) - (s * o2 * o2)) \\ JT(26, i) &= ((We(46, 1)) / (o1 * o1)) * ((o1 * o1 * t) - (t * o2 * o2)) \\ JT(27, i) &= ((We(46, 1)) / (o1 * o1)) * ((o1 * o1 * u) - (u * o2 * o2)) \\ JT(28, i) &= ((We(46, 1)) / (o1 * o1)) * ((o1 * o1 * 1) - (1 * o2 * o2)) \end{aligned}$$

$$\begin{aligned} JT(29, i) &= ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * p) - (p * m2 * m2)) \\ JT(30, i) &= ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * q) - (q * m2 * m2)) \\ JT(31, i) &= ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * r) - (r * m2 * m2)) \\ JT(32, i) &= ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * s) - (s * m2 * m2)) \\ JT(33, i) &= ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * t) - (t * m2 * m2)) \\ JT(34, i) &= ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * u) - (u * m2 * m2)) \\ JT(35, i) &= ((We(47, 1)) / (m1 * m1)) * ((m1 * m1 * 1) - (1 * m2 * m2)) \end{aligned}$$

$$\begin{aligned} JT(36, i) &= ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * p) - (p * ni2 * ni2)) \\ JT(37, i) &= ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * q) - (q * ni2 * ni2)) \\ JT(38, i) &= ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * r) - (r * ni2 * ni2)) \\ JT(39, i) &= ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * s) - (s * ni2 * ni2)) \\ JT(40, i) &= ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * t) - (t * ni2 * ni2)) \\ JT(41, i) &= ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * u) - (u * ni2 * ni2)) \\ JT(42, i) &= ((We(48, 1)) / (ni1 * ni1)) * ((ni1 * ni1 * 1) - (1 * ni2 * ni2)) \end{aligned}$$

$$\begin{aligned} JT(43, i) &= d2 / d1 \\ JT(44, i) &= e2 / e1 \\ JT(45, i) &= g2 / g1 \\ JT(46, i) &= o2 / o1 \\ JT(47, i) &= m2 / m1 \\ JT(48, i) &= ni2 / ni1 \\ JT(49, i) &= 1 \end{aligned}$$

Next i

'Cells(26, 11) = ss
'Valor de mu
'Cells(29, 11) = Mu

'Multiplicación de JT(22,318)*J(318,22)

```
For k = 1 To 49
For n = 1 To 49
multiplicacion = 0
For m = 1 To 596
multiplicacion = multiplicacion + JT(k, m) * J(m, n)
Next m
JTJ(k, n) = multiplicacion
'Cells(27 + k, 10 + n) = JTJ(k, n)
```

```

Next n
Next k

'Matriz identidad multiplicada por Mu
For x = 1 To 49
For Z = 1 To 49
  If x = Z Then
    Mul(x, Z) = mu
  Else
    Mul(x, Z) = 0
  End If
  'Cells(27 + x, 10 + Z) = Mul(x, Z)
Next Z

Next x

```

```

'Multiplicación de JT(x,z)*w(x,z)

For k = 1 To 49
For n = 1 To 1
multiplicacion = 0
For m = 1 To 596
multiplicacion = multiplicacion + JT(k, m) * w(m, n)
Next m
JTe(k, n) = multiplicacion
'Cells(27 + k, 10 + n) = JTe(k, n)
Next n
Next k
For x = 1 To 49
For Z = 1 To 49
MJ(x, Z) = JTJ(x, Z) + Mul(x, Z)
'Cells(27 + x, 10 + Z) = MJ(x, Z)
Next Z
Next x
'Calculo de la inversa
'Inversa de la matriz MJ(x,z)

```

```

'Agrega 1 a la diagonal de la Matriz
For x = 1 To 49
For Z = 1 To 49
  If x = Z Then
    ID(x, Z) = 1
  Else

```

```
        ID(x, Z) = 0
    End If
    'Cells(27 + x, 10 + Z) = Mul(x, Z)
Next Z

Next x

'For x = 1 To 22
'For Z = 1 To 22
    'Cells(27 + x, 10 + Z) = ID(x, Z)
'Next
'Next

For i = 1 To 49
    If MJ(i, i) = 0 Then
        For t = i + 1 To 49
            If MJ(t, i) <> 0 Then
                For k = 1 To 49
                    MJ(i, k) = MJ(i, k) + MJ(t, k)
                    ID(i, k) = ID(i, k) + ID(t, k)
                Next
                Exit For
            End If
        Next
    End If
    c = MJ(i, i)
    For k = 1 To 49
        MJ(i, k) = MJ(i, k) / c
        ID(i, k) = ID(i, k) / c
    Next

    For t = i + 1 To 49
        If MJ(t, i) <> 0 Then
            c = MJ(t, i)
            For k = 1 To 49
                MJ(t, k) = MJ(t, k) - c * MJ(i, k)
                ID(t, k) = ID(t, k) - c * ID(i, k)
            Next
        End If
    Next
Next
Next
```



```

For i = 49 To 1 Step -1
  For t = i - 1 To 1 Step -1
    If MJ(t, i) <> 0 Then
      c = MJ(t, i)
      For k = 49 To 1 Step -1
        MJ(t, k) = MJ(t, k) - c * MJ(i, k)
        ID(t, k) = ID(t, k) - c * ID(i, k)
      Next
    End If
  Next
Next

```

```

'Imprime la Matriz Identidad en la Posición indicada
'For x = 1 To 22
'For Z = 1 To 22
  'Cells(27 + x, 10 + Z) = ID(x, Z)
'Next
'Next

```

'Multiplicación de ID(x,z)*JTe(x,z)

```

For k = 1 To 49
  For n = 1 To 1
    multiplicacion = 0
    For m = 1 To 49
      multiplicacion = multiplicacion + ID(k, m) * JTe(m, n)
    Next m
    WW(k, n) = multiplicacion
  'Cells(27 + k, 10 + n) = WW(k, n)
  Next n
Next k
For i = 1 To 49
  'Cells(27 + i, 11) = WW(i, 1)
  We(i, 1) = We(i, 1) - WW(i, 1)
  Cells(2 + i, 11 + nu) = We(i, 1)
Next i

```

```

ssp = 0
For i = 1 To 596

```

```

p = Cells(3 + i, 2)
q = Cells(3 + i, 3)

```

r = Cells(3 + i, 4)
s = Cells(3 + i, 5)
t = Cells(3 + i, 6)
u = Cells(3 + i, 7)

y(i, 1) = Cells(3 + i, 8) 'Salida deseada

A = p * We(1, 1) + q * We(2, 1) + r * We(3, 1) + s * We(4, 1) + t * We(5, 1) + u * We(6, 1) + We(7, 1)

b = p * We(8, 1) + q * We(9, 1) + r * We(10, 1) + s * We(11, 1) + t * We(12, 1) + u * We(13, 1) + We(14, 1)

c = p * We(15, 1) + q * We(16, 1) + r * We(17, 1) + s * We(18, 1) + t * We(19, 1) + u * We(20, 1) + We(21, 1)

h = p * We(22, 1) + q * We(23, 1) + r * We(24, 1) + s * We(25, 1) + t * We(26, 1) + u * We(27, 1) + We(28, 1)

l = p * We(29, 1) + q * We(30, 1) + r * We(31, 1) + s * We(32, 1) + t * We(33, 1) + u * We(34, 1) + We(35, 1)

n = p * We(36, 1) + q * We(37, 1) + r * We(38, 1) + s * We(39, 1) + t * We(40, 1) + u * We(41, 1) + We(42, 1)

d = (Exp(A) - Exp(-1 * A)) / (Exp(A) + Exp(-1 * A))

e = (Exp(b) - Exp(-1 * b)) / (Exp(b) + Exp(-1 * b))

g = (Exp(c) - Exp(-1 * c)) / (Exp(c) + Exp(-1 * c))

o = (Exp(h) - Exp(-1 * h)) / (Exp(h) + Exp(-1 * h))

m = (Exp(l) - Exp(-1 * l)) / (Exp(l) + Exp(-1 * l))

ni = (Exp(n) - Exp(-1 * n)) / (Exp(n) + Exp(-1 * n))

d1 = (Exp(A) + Exp(-1 * A))

d2 = (Exp(A) - Exp(-1 * A))

e1 = (Exp(b) + Exp(-1 * b))

e2 = (Exp(b) - Exp(-1 * b))

g1 = (Exp(c) + Exp(-1 * c))

g2 = (Exp(c) - Exp(-1 * c))

o1 = (Exp(h) + Exp(-1 * h))

o2 = (Exp(h) - Exp(-1 * h))

m1 = (Exp(l) + Exp(-1 * l))

m2 = (Exp(l) - Exp(-1 * l))

ni1 = (Exp(n) + Exp(-1 * n))

ni2 = (Exp(n) - Exp(-1 * n))

'Salida de la red neuronal

f(i, 1) = d * We(43, 1) + e * We(44, 1) + g * We(45, 1) + o * We(46, 1) + m * We(47, 1) + ni * We(48, 1) + We(49, 1)

'Error en la salida

w(i, 1) = f(i, 1) - y(i, 1)

```
ssp = ssp + (w(i, 1) * w(i, 1))
Next i
```

```
Cells(53, 10 + nu) = ssp
```

```
If (ssp < ssa) Then
mu = mu / 10
```

```
For i = 1 To 49
'Cells(27 + i, 11) = WW(i, 1)
We(i, 1) = We(i, 1)
Cells(2 + i, 11 + nu) = We(i, 1)
Next i
```

```
ssa = 0
```

```
For i = 1 To 596
```

```
p = Cells(3 + i, 2)
q = Cells(3 + i, 3)
r = Cells(3 + i, 4)
s = Cells(3 + i, 5)
t = Cells(3 + i, 6)
u = Cells(3 + i, 7)
```

```
y(i, 1) = Cells(3 + i, 8) 'Salida deseada
```

```
A = p * We(1, 1) + q * We(2, 1) + r * We(3, 1) + s * We(4, 1) + t * We(5, 1) + u *
We(6, 1) + We(7, 1)
```

```
b = p * We(8, 1) + q * We(9, 1) + r * We(10, 1) + s * We(11, 1) + t * We(12, 1) + u *
We(13, 1) + We(14, 1)
```

```
c = p * We(15, 1) + q * We(16, 1) + r * We(17, 1) + s * We(18, 1) + t * We(19, 1) + u *
We(20, 1) + We(21, 1)
```

```
h = p * We(22, 1) + q * We(23, 1) + r * We(24, 1) + s * We(25, 1) + t * We(26, 1) +
u * We(27, 1) + We(28, 1)
```

```
l = p * We(29, 1) + q * We(30, 1) + r * We(31, 1) + s * We(32, 1) + t * We(33, 1) + u *
We(34, 1) + We(35, 1)
```

```
n = p * We(36, 1) + q * We(37, 1) + r * We(38, 1) + s * We(39, 1) + t * We(40, 1) +
u * We(41, 1) + We(42, 1)
```

```
d = (Exp(A) - Exp(-1 * A)) / (Exp(A) + Exp(-1 * A))
```

```
e = (Exp(b) - Exp(-1 * b)) / (Exp(b) + Exp(-1 * b))
```

```
g = (Exp(c) - Exp(-1 * c)) / (Exp(c) + Exp(-1 * c))
```

```
o = (Exp(h) - Exp(-1 * h)) / (Exp(h) + Exp(-1 * h))
```

```
m = (Exp(l) - Exp(-1 * l)) / (Exp(l) + Exp(-1 * l))
```

```
ni = (Exp(n) - Exp(-1 * n)) / (Exp(n) + Exp(-1 * n))
```

```
d1 = (Exp(A) + Exp(-1 * A))
d2 = (Exp(A) - Exp(-1 * A))
e1 = (Exp(b) + Exp(-1 * b))
e2 = (Exp(b) - Exp(-1 * b))
g1 = (Exp(c) + Exp(-1 * c))
g2 = (Exp(c) - Exp(-1 * c))
o1 = (Exp(h) + Exp(-1 * h))
o2 = (Exp(h) - Exp(-1 * h))
m1 = (Exp(l) + Exp(-1 * l))
m2 = (Exp(l) - Exp(-1 * l))
ni1 = (Exp(n) + Exp(-1 * n))
ni2 = (Exp(n) - Exp(-1 * n))
```

'Salida de la red neuronal

```
f(i, 1) = d * We(43, 1) + e * We(44, 1) + g * We(45, 1) + o * We(46, 1) + m * We(47, 1) + ni * We(48, 1) + We(49, 1)
```

'Error en la salida

```
w(i, 1) = f(i, 1) - y(i, 1)
```

```
ssa = ssa + (w(i, 1) * w(i, 1))
```

```
Next i
```

Else

```
mu = mu * 10
```

```
For i = 1 To 49
```

```
'Cells(27 + i, 11) = WW(i, 1)
```

```
We(i, 1) = We(i, 1) + WW(i, 1)
```

```
Cells(2 + i, 11 + nu) = We(i, 1)
```

```
Next i
```

```
ssa = 0
```

```
For i = 1 To 596
```

```
p = Cells(3 + i, 2)
```

```
q = Cells(3 + i, 3)
```

```
r = Cells(3 + i, 4)
```

```
s = Cells(3 + i, 5)
```

```
t = Cells(3 + i, 6)
```

```
u = Cells(3 + i, 7)
```

```
y(i, 1) = Cells(3 + i, 8) 'Salida deseada
```

$$A = p * We(1, 1) + q * We(2, 1) + r * We(3, 1) + s * We(4, 1) + t * We(5, 1) + u * We(6, 1) + We(7, 1)$$

$$b = p * We(8, 1) + q * We(9, 1) + r * We(10, 1) + s * We(11, 1) + t * We(12, 1) + u * We(13, 1) + We(14, 1)$$

$$c = p * We(15, 1) + q * We(16, 1) + r * We(17, 1) + s * We(18, 1) + t * We(19, 1) + u * We(20, 1) + We(21, 1)$$

$$h = p * We(22, 1) + q * We(23, 1) + r * We(24, 1) + s * We(25, 1) + t * We(26, 1) + u * We(27, 1) + We(28, 1)$$

$$l = p * We(29, 1) + q * We(30, 1) + r * We(31, 1) + s * We(32, 1) + t * We(33, 1) + u * We(34, 1) + We(35, 1)$$

$$n = p * We(36, 1) + q * We(37, 1) + r * We(38, 1) + s * We(39, 1) + t * We(40, 1) + u * We(41, 1) + We(42, 1)$$

$$d = (\text{Exp}(A) - \text{Exp}(-1 * A)) / (\text{Exp}(A) + \text{Exp}(-1 * A))$$

$$e = (\text{Exp}(b) - \text{Exp}(-1 * b)) / (\text{Exp}(b) + \text{Exp}(-1 * b))$$

$$g = (\text{Exp}(c) - \text{Exp}(-1 * c)) / (\text{Exp}(c) + \text{Exp}(-1 * c))$$

$$o = (\text{Exp}(h) - \text{Exp}(-1 * h)) / (\text{Exp}(h) + \text{Exp}(-1 * h))$$

$$m = (\text{Exp}(l) - \text{Exp}(-1 * l)) / (\text{Exp}(l) + \text{Exp}(-1 * l))$$

$$ni = (\text{Exp}(n) - \text{Exp}(-1 * n)) / (\text{Exp}(n) + \text{Exp}(-1 * n))$$

$$d1 = (\text{Exp}(A) + \text{Exp}(-1 * A))$$

$$d2 = (\text{Exp}(A) - \text{Exp}(-1 * A))$$

$$e1 = (\text{Exp}(b) + \text{Exp}(-1 * b))$$

$$e2 = (\text{Exp}(b) - \text{Exp}(-1 * b))$$

$$g1 = (\text{Exp}(c) + \text{Exp}(-1 * c))$$

$$g2 = (\text{Exp}(c) - \text{Exp}(-1 * c))$$

$$o1 = (\text{Exp}(h) + \text{Exp}(-1 * h))$$

$$o2 = (\text{Exp}(h) - \text{Exp}(-1 * h))$$

$$m1 = (\text{Exp}(l) + \text{Exp}(-1 * l))$$

$$m2 = (\text{Exp}(l) - \text{Exp}(-1 * l))$$

$$ni1 = (\text{Exp}(n) + \text{Exp}(-1 * n))$$

$$ni2 = (\text{Exp}(n) - \text{Exp}(-1 * n))$$

'Salida de la red neuronal

$$f(i, 1) = d * We(43, 1) + e * We(44, 1) + g * We(45, 1) + o * We(46, 1) + m * We(47, 1) + ni * We(48, 1) + We(49, 1)$$

'Error en la salida

$$w(i, 1) = f(i, 1) - y(i, 1)$$

$$ssa = ssa + (w(i, 1) * w(i, 1))$$

Next i

End If

Next nu

End Sub



Km 12+000 Carretera Estatal 431 “El Colorado-Galindo”
Parque Tecnológico San Fandila
Mpio. Pedro Escobedo, Querétaro, México
CP 76703
Tel +52 (442) 216 9777 ext. 2610
Fax +52 (442) 216 9671

publicaciones@imt.mx

<http://www.imt.mx/>

Esta publicación fue desarrollada en el marco de un sistema de gestión de calidad
certificada bajo la norma ISO 9001:2015