

## Redes neuronales artificiales y análisis de datos de tráfico en el corredor vial Zipaquirá-Palenque

Edisson Caicedo Rojas

[ercaicedo@unisangil.edu.co](mailto:ercaicedo@unisangil.edu.co)

<https://orcid.org/0000-0001-9702-0144>

Carlos Uriel Cortes

[ccortes@unisangil.edu.co](mailto:ccortes@unisangil.edu.co)

Chiquinquirá - Colombia

### RESUMEN

El presente trabajo muestra una síntesis de las redes neuronales artificiales (RNA), haciendo énfasis en una arquitectura de red elaborada en el software de Matlab® para la predicción de tráfico en el corredor vial Zipaquirá-Palenque. Es de aclarar que se tiene en cuenta parámetros de la ley 787 de 2002 (Tasas, tarifas y peajes en la infraestructura de transporte a cargo de la Nación Colombiana) y requisitos teóricos para la construcción de las RNA. El trabajo muestra los resultados del entrenamiento basado en datos recopilados en páginas gubernamentales y la implementación del modelo para estimar el flujo vehicular en dicho corredor, así como la precisión de las aproximaciones obtenidas.

*Palabras claves:* análisis de datos; matlab; redes neuronales; tráfico.

Correspondencia: [ercaicedo@unisangil.edu.co](mailto:ercaicedo@unisangil.edu.co)

Artículo recibido 29 noviembre 2022 Aceptado para publicación: 29 diciembre 2022

Conflictos de Interés: Ninguna que declarar

Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo

Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) 

Cómo citar: Caicedo Rojas, E., & Uriel Cortes, C. (2022). Redes neuronales artificiales y análisis de datos de tráfico en el corredor vial Zipaquirá-Palenque. *Ciencia Latina Revista Científica Multidisciplinar*, 6(6), 10739-10757. [https://doi.org/10.37811/cl\\_rcm.v6i6.4162](https://doi.org/10.37811/cl_rcm.v6i6.4162)

## Artificial Neural Networks and traffic data analysis in the Zipaquirá-Palenque road corridor

### ABSTRACT

Shows a synthesis of artificial neural networks (ANN), with emphasis on a network architecture developed in Matlab® software for traffic prediction in the Zipaquirá-Palenque road corridor. The parameters of Law 787 of 2002 (Fees, tariffs and tolls in the transportation infrastructure in charge of the Colombian Nation) and theoretical requirements for the construction of ANNs were taken into account. The work shows the results of the training based on data collected in governmental pages and the implementation of the model to estimate the vehicular flow in said corridor, as well as the accuracy of the approximations obtained.

**Keywords:** *data analysis; matlab; neural networks; traffic.*

## INTRODUCCIÓN

La evolución de las diversas industrias ha impulsado cambios que prometen facilitar las actividades diarias de la sociedad. En este sentido, se han formulado e implementando algoritmos, modelos matemáticos y arquitecturas basadas en máquinas de aprendizaje automático que son capaces de analizar e identificar patrones aproximados a los obtenidos en procesos reales, intentando satisfacer las nuevas necesidades tecnológicas que surgen con la disponibilidad de datos de dichos procesos reales. Uno de estos sistemas son las Redes Neuronales Artificiales, que con su avance en los últimos años se han convertido en una potente herramienta para el procesamiento y análisis de datos.

En los últimos años, las redes neuronales han tomado gran importancia en medio de este auge tecnológico gracias a la cantidad exponencial de información que se procesa a diario. Además, la integración con otros sistemas que hacen uso de este método de inteligencia artificial potencializa sus funciones de análisis, conectividad y desarrollo[1].

Basados en lo expuesto previamente sobre las ventajas que ofrecen las RNA, su potencial para el análisis y procesamiento de datos y la versatilidad en su desarrollo, el presente trabajo abordará el problema de la aproximación del tráfico vehicular en el corredor vial Zipaquirá-Palenque, usando datos del flujo real para el entrenamiento y prueba de un modelo de RNA.

## METODOLOGÍA

### a. Datos

Una red neuronal artificial necesita de una gran cantidad de datos que le permita identificar patrones para su correspondiente entrenamiento, en el presente proyecto los datos utilizados se extrajeron de la página web de la agencia nacional de infraestructura (ANI), cuenta con información de todos los corredores viales del país, de esta base fue seleccionando el corredor vial Zipaquirá-Palenque con datos disponibles para un periodo de tiempo que va desde el año 2003 hasta el mes de Junio de 2017, debido a que “El 30 de junio de 2017, revirtió a la Nación los peajes del proyecto de concesión vial Zipaquirá-Palenque.”[2] y se dejaron de publicar estos datos. Cabe resaltar que los datos también se dividieron en dos, donde los datos de 2003 a 2015 se utilizaron para el entrenamiento y los años restantes para la realización de pruebas con datos plenamente desconocidos para la RNA. El corredor vial comprende 5 peajes que son: Casablanca, Saboya, Los Curos, Curití, Oiba.

Los datos fueron organizados en vectores de entrada por año, mes y peaje, y vectores de salida que contienen el número de carros por peaje (ver Tabla 1), para el reconocimiento de datos, se tuvo en cuenta que Matlab® no permite caracteres para el entrenamiento de la red, por lo que la variable de peajes (cualitativa) fue cambiada por una variable cuantitativa donde 1111 corresponde a Los Curos, 2222 a Curití, 3333 a Oiba, 4444 a Saboya y 5555 a Casablanca, dando como resultado que toda la información ingresada fuera numérica.

**Tabla 1. Estructura de organización de datos.**

Datos de Entrada			Datos de Salida
Año	Mes	Peaje	Numero de carros
2003	1	1111	108753
2003	1	2222	98992
2003	1	3333	90666
2003	1	4444	48605
2003	1	5555	147443

825 filas

#### b. Entrenamiento RNA

**Tabla 2. Parámetros de arquitectura de la RNA.**

Tipo de RNA	Feedforwardnet
Capas Ocultas	2
Neuronas por capa variables de entrada	9-5 respectivamente
Variabes de Salida	3
Función de activación	1
Función de entrenamiento	Tangente hiperbólica capas ocultas y lineal capa de salida
Criterio de desempeño	Levenberg-Marquardt (trainlm)
Iteraciones	Error cuadrático Medio (MSE)
Parada temprana	400
	221

Para la arquitectura de la red neuronal artificial se utilizó los parámetros enunciados en la Tabla 2. En el software de Matlab® el cual tiene por defecto definido algunos tipos de red, la utilizada en este caso fue el diseño de red feedforward “(red neuronal de avance) que consiste en una serie de capas. La primera capa tiene una conexión desde la entrada de red. Cada capa posterior tiene una conexión desde la capa anterior. La capa final produce la salida de la red.”[3]. El entrenamiento backpropagation que es un tipo de entrenamiento basado en una retroalimentación constante con el fin de dar un resultado o una respuesta cercana frente a una variable objetivo, este a su vez tiene varios tipos de entrenamientos que permiten optimizar el aprendizaje predictivo de una red neuronal artificial.

Para el desarrollo de la red neuronal primero se trabajó con trainlm que “es una función de entrenamiento en red que actualiza los valores de peso y sesgo según la optimización de Levenberg-Marquardt. Minimiza una combinación de errores al cuadrado y pesos, y luego determina la combinación correcta para producir una red que generalice bien. Llamado regularización bayesiana.”[3] Otra de las trabajadas es trainscg “es una función de entrenamiento en red que actualiza los valores de peso y sesgo de acuerdo con el método de gradiente conjugado escalado.”[3], dando como resultado que el método de Levenberg- Marquardt da mejores resultados para la predicción de los datos.

La función tangente hiperbólica transforma los valores introducidos a una escala (-1,1), donde los valores altos tienden de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a -1. [4] Esta se basa en la fórmula:

$$\frac{2}{1 + e^{-2x}} - 1$$

Adicionalmente para la capa de salida se utilizó una función lineal, también conocida como identidad, permite que lo de la entrada sea igual a la salida por lo que si tengo una red neuronal de varias capas y aplicó función lineal se dice que es una regresión lineal. Por lo tanto, esta función de activación lineal se usa si a la salida se requiere una regresión lineal y de esta manera a la red neuronal que se le aplica la función va a generar un valor único. Por ejemplo, se usa cuando se solicita predecir el valor de un número de ventas.[5]

**La fórmula que representa la función lineal corresponde a:**

$$f(x) = W * x$$

$$\forall x \in R$$

Por otro lado, se tiene que el criterio de desempeño utilizado fue el del error cuadrático medio, donde se suman todos los errores y se dividen entre el número total de puntos.  $M$  es el número total de puntos[6]. Esto se expresa con la siguiente formula:

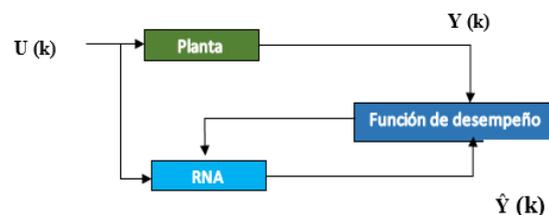
$$MSE = \frac{1}{M} \sum_{i=1}^M (real_i - estimado_i)^2$$

## REDES NEURONALES ARTIFICIALES (RNA) Y APLICACIÓN AL ANÁLISIS DE DATOS

Una red neuronal artificial (RNA) es una simulación abstracta del sistema nervioso biológico, con el objetivo de resolver, y automatizar problemas por medio de paradigmas de programación y matemáticas en tiempos cortos y de manera eficaz. Las RNA están compuestas por nodos interconectados que resultan en una salida aproximada de datos reales mediante la interacción, transmisión y proceso de información en ciclos computacionales de muchas iteraciones, para que la red tenga éxito, esta se compone por:

- La arquitectura de la red: se debe tener en cuenta la relación a ser aprendida, cantidad de capas y número de neuronas por cada capa, función de activación a ser usada, función de desempeño y el algoritmo de entrenamiento.
- Proceso de entrenamiento: se ingresan a la red datos reales con los que la red se entrenara y posteriormente comprobara los resultados de esta con los datos reales, si se presenta un error muy grande, la RNA ajusta los pesos y vuelve a realizar el entrenamiento y la comparación hasta que arroje el menor error (Ver Figura 1).

Figura 1. Proceso de entrenamiento de RNA.



Las salidas de la red se calculan de manera iterativa de acuerdo con los valores de entrenamiento, con el objeto de minimizar el error cometido entre la predicción obtenida por la red neuronal y la salida deseada mediante un proceso de optimización matemática ajustando una función de desempeño (tolerancia o error). La salida de la red está determinada por la ecuación:

$$\hat{y}_k = \varphi\left(\sum_{k=1}^n w_k x_k\right)$$

**Donde:**

$\hat{y}_k$  = Valor de Salida por la RNA

$\varphi$  = Es el margen de predicción, indica que tan alejado está el promedio de predicciones respecto al promedio de observaciones en el conjunto de datos

$w_k$  = Pesos

$x_k$  = entradas

$$\sum_{k=1}^n \text{sumatoria de 1 hasta } n \text{ iteraciones}$$

[1] En su libro *Redes Neuronales, conceptos básicos y aplicaciones* expresa que: el término de redes neuronales apareció en torno a una abstracción de la composición y funcionamiento del sistema biológico, cuya propuesta fue realizada en 1943 por McCulloch y Pitts en un contexto computacional. Aquel planteamiento correspondió a un modelo binario donde las neuronas eran imprescindibles de un umbral prefijado que sería útil para el diseño de estructuras posteriores. Más tarde Marvin Minsky construyó SNARC, el primer simulador de cadena neuronal, el cual ajustaba los pesos sinápticos de la red neuronal automáticamente, aunque no llegó a procesar datos. En 1956 en la conferencia internacional de inteligencia artificial se presentó un modelo de RNA que simulaba centenas de neuronas interconectadas en un sistema que verificaba como la red respondía a estímulos ambientales y posteriormente en los años 80 se dio surgimiento al algoritmo backpropagation siendo el primer algoritmo sin restricciones. Antes de entrar en detalle sobre el tratamiento que se realiza a los datos una red neuronal, es necesario contemplar un flujo de trabajo típico para diseñar redes neuronales, propuesto por Mathworks en su página oficial cuyo desarrollo de la red suele implicar los siguientes pasos:

1. Acceder a los datos y prepararlos.
2. Crear la red neuronal.
3. Configurar las entradas y salidas de la red.
4. Ajustar los parámetros de la red (las ponderaciones y tendencias) para optimizar el rendimiento.
5. Entrenar la red.

6. Validar los resultados de la red.
7. Integrar la red en un sistema de producción.

En cuanto al manejo de datos dentro de una RNA, estos se abordan designándoles variables específicas indicando de que tipo son, ya sea de entrada o salida, asociados a unas neuronas que constituyen la capa de entrada, cuyos pesos pueden priorizarse o permitirle a la red decidir internamente, las conexiones entre neuronas modelan el comportamiento de la red mediante una ponderación de pesos que se ajustan según los criterios indicados en un inicio hasta que la red adopte la capacidad de llevar a cabo el propósito por el que se creó de manera correcta.

Desde la perspectiva adquirida por proyectos realizados con redes neuronales artificiales, se ha identificado que típicamente en las redes dividen los datos en tres grupos, donde alrededor de un 80% de ellos están orientados a la etapa de entrenamiento, la cual en algunos casos termina con la etapa de validación que le permite evitar un sobre entrenamiento. Esto, está definido básicamente por el método de optimización matemática que use la función de entrenamiento de la RNA. Finalmente, en la etapa de prueba, la capacidad predictiva de la red se evalúa por el asertividad en los resultados, esto es, comparación entre los datos reales y los obtenidos por la red, a través de una función llamada función de desempeño.

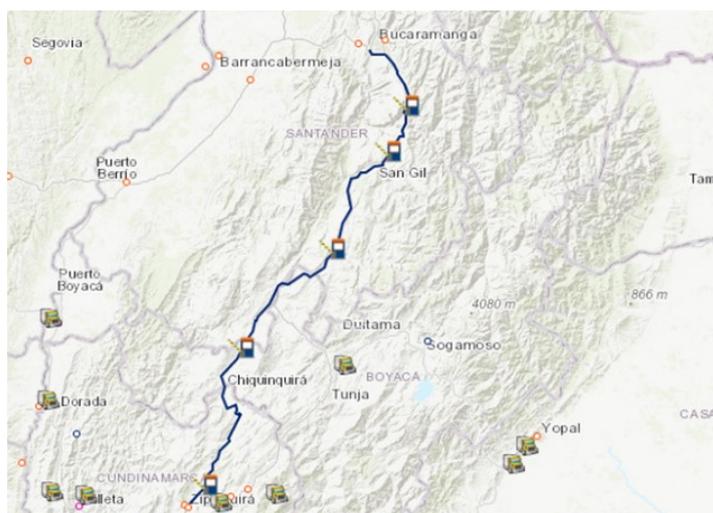
### **CORREDOR VIAL ZIPAQUIRA-PALENQUE**

Las vías en un país son de vital importancia teniendo en cuenta que representan el soporte para el transporte de personas y carga, aportando desde su función a satisfacer necesidades básicas en cuanto a educación, trabajo, alimentación, salud, entre otras. Gracias a la configuración de la malla vial del país las vías o corredores viales del país permiten ser clasificados y evaluados en cuanto a la capacidad del flujo vehicular que soportan, eso sí, dicha capacidad se da en función a la productividad y comercio de las zonas conectadas entre sí, dependiendo de factores como el recaudo en sus peajes y mejoras dependerán de los ingresos recogidos.

Colombia es un país que cuenta con numerosos recursos naturales permitiendo la realización de actividades que favorezcan la productividad y comercio entre diversas zonas, sin embargo, existen diversos factores que retrasan dicho avance en el territorio; uno de ellos, el mal estado de las vías y la falta de inversión en ellas.

Por otra parte, los peajes son los derechos que se deben pagar para poder transitar por una vía y que sus precios son regulados mediante la ley 787 de 2002 que establece que, “la nación puede fijar la tarifa y que debe destinar los recursos obtenidos por este medio exclusivamente a la inversión en el transporte terrestre. Así mismo señala que se deben fijar tasas de peaje diferenciales en proporción a las distancias recorridas, las características de los vehículos y sus respectivos costos de operación” (Escobar, 2018). El corredor vial Zipaquirá – Palenque (Figura 2), que comunica a Bogotá, la capital del país, con Bucaramanga, la capital del departamento de Santander, cuenta con 5 peajes interconectando diversas ciudades a lo largo de 370 Km, entre ellas la ciudad de Chiquinquirá, lugar de estudio de los desarrolladores de este modelo de Red Neuronal Artificial permitiendo los autores del proyecto obtener una visión más clara basada en el contexto donde habitan.

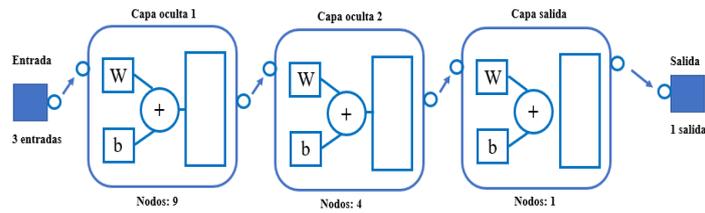
**Figura 2. Corredor Vial**



## RESULTADOS

Primeramente, se elige una arquitectura compuesta por una capa de entrada, una capa de salida que trabaja bajo una función lineal y 2 capas ocultas con 9 y 4 neuronas respectivamente que utilizan una función sigmoide (Ver Figura 2), con la cual se obtuvieron los siguientes resultados:

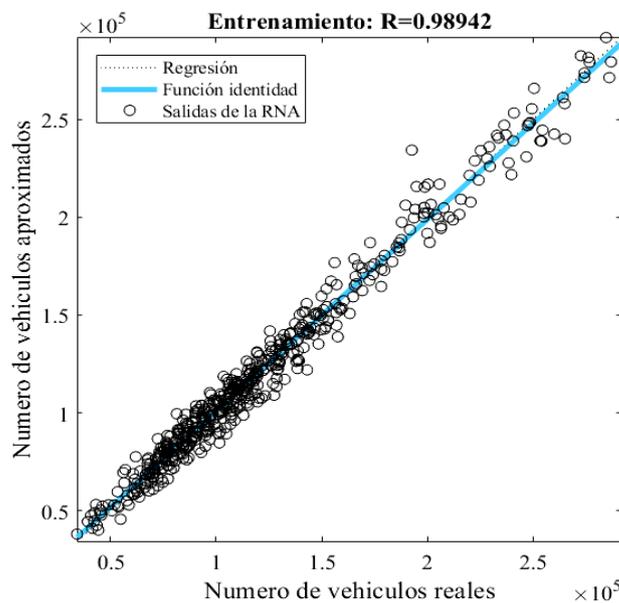
Figura 3. Arquitectura de la RNA



Las siguientes 3 figuras representan los datos de entrenamiento, validación y prueba. La línea discontinua de cada gráfico representa el resultado ideal (objetivos). La línea sólida representa la línea de regresión lineal mejor ajustada entre los resultados y los objetivos. El valor R es una indicación de la relación entre los resultados y los objetivos. Si  $R = 1$ , indica que existe una relación lineal exacta entre los resultados y los objetivos. Si R se aproxima a cero, entonces no existe una relación lineal entre las salidas y los objetivos [3].

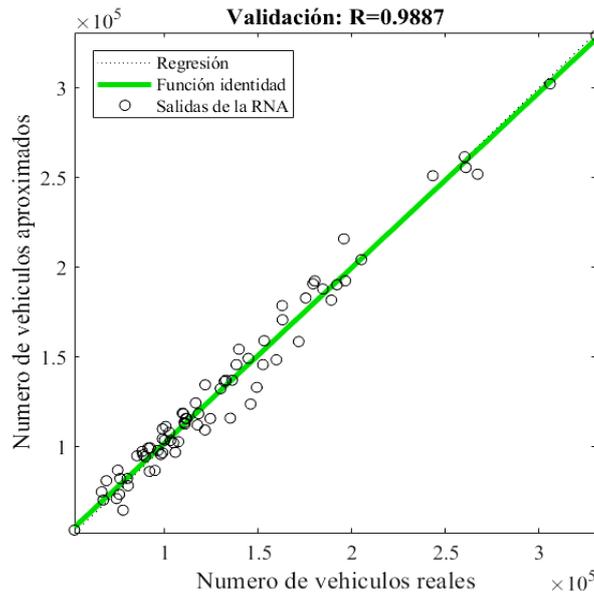
La Figura 4 muestra los resultados del entrenamiento de la RNA., Podemos observar que en los parametros de la regresión nuestro coeficiente de rendimiento R se encuentra cercano a 1, lo cual denota un entrenamiento adecuado de la RNA.

Figura 4. Entrenamiento de la RNA



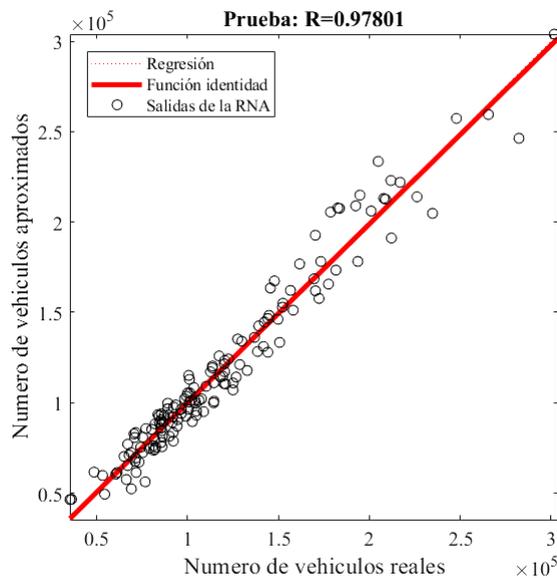
En la Figura 5, que muestra el proceso de validación, observamos que la red tiene una baja dispersión en su regresión lo cual indica que no se tiene un sobre entrenamiento de la red y se verifican la calidad de las estimaciones con un factor R cercano a la unidad.

Figura 5. Validación de la RNA



En la Figura 6 se muestra el nivel de exactitud en las predicciones, los datos confluyen de manera muy cercana respecto a la regresión ideal y no se observan valores atípicos relevantes con respecto a la tendencia en las salidas de la RNA. Es decir, que las etapas de entrenamiento y validación se desarrollaron eficientemente, y su impacto es notorio en el equilibrio conseguido del modelo. Además, se observa que como la mayor disponibilidad de datos se encontraba en un flujo vehicular menor a 150.000, los resultados en esas cifras denotan una mejor aproximación que los datos con cifras superiores a 150.000 vehículos.

Figura 6. Prueba de la RNA.



Los resultados visibles en las gráficas de las etapas de entrenamiento, validación y pruebas corresponden a un proceso matemático iterativo donde se intenta reducir la pérdida entre las salidas de la red y los valores reales.

Para esto, la red realiza ajustes en cada paso por los datos, modificando los pesos y los días hasta obtener una regresión esperada, es decir una adecuada relación lineal entre los datos y las aproximaciones obtenidas de la red. De manera que a continuación se expresan las ecuaciones que describen estos procesos:

$$m = \frac{N \sum_{i=1}^N y_i \hat{y}_i - \sum_{i=1}^N y_i \sum_{i=1}^N \hat{y}_i}{N \sum_{i=1}^N y_i^2 - (\sum_{i=1}^N y_i)^2}$$

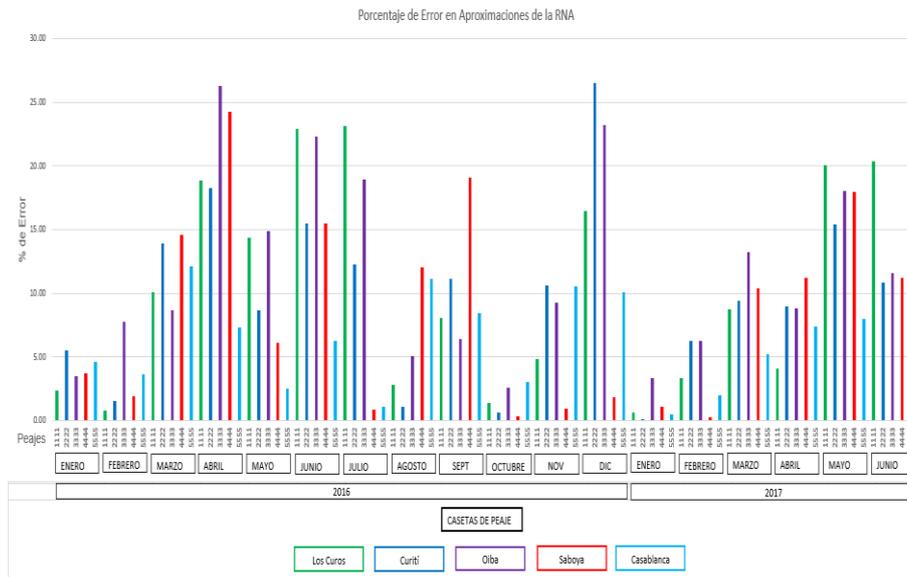
$$b = \frac{\sum_{i=1}^N \hat{y}_i - m \sum_{i=1}^N y_i}{N}$$

$$R = \frac{N \sum_{i=1}^N y_i \hat{y}_i - \sum_{i=1}^N y_i \sum_{i=1}^N \hat{y}_i}{\sqrt{[N \sum_{i=1}^N y_i^2 - (\sum_{i=1}^N y_i)^2][N \sum_{i=1}^N \hat{y}_i^2 - (\sum_{i=1}^N \hat{y}_i)^2]}}$$

Por otra parte, con el objetivo de comprobar el funcionamiento de la RNA, los datos correspondientes a los años 2016 y 2017 fueron usados como datos de entrada para obtener los de salida haciendo uso de la RNA elaborada, sin que estos hubiesen sido usados en el entrenamiento de la RNA; una vez completado el proceso, se realizó una comparación con el fin de determinar la diferencia entre los datos reales y las aproximaciones obtenidas.

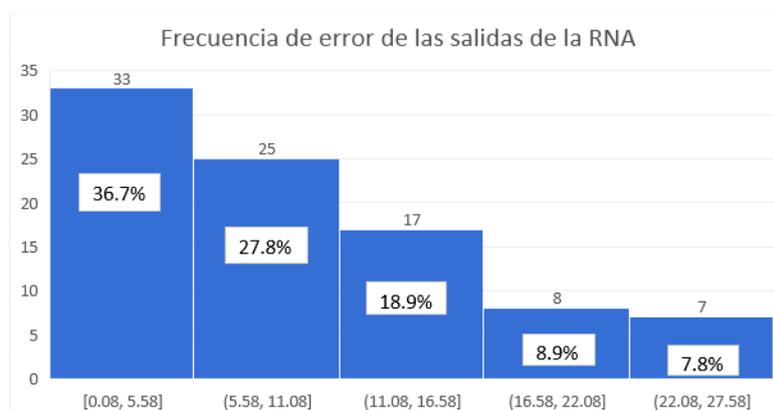
La figura 7 ilustra el porcentaje de error de las salidas de la red en cada peaje del corredor vial Zipaquirá-Palenque mes a mes para los años 2016 y 2017. Se puede observar que los errores más pequeños en las predicciones se encuentran en el peaje de Casablanca, seguido por Saboya, una aparente igualdad entre los Curos y Curití, y en última instancia Oiba donde se concentra el mayor porcentaje de error.

Figura 7. Grafica de porcentajes de diferencia en aproximaciones obtenidas por la RNA



Por otro lado, la Figura 8 representa un histograma con los diferentes intervalos de error agrupando un conteo de las salidas de la red que se encuentran entre esos porcentajes de error, por ejemplo, que de un porcentaje de error entre 0.08 a 5.58 se ubican 33 salidas de la red. De tal manera que es correcto afirmar que el 83.4% de las salidas dadas por la red ante datos de entrada desconocidos están localizados entre el 0.08% y el 16.58% de error. En otras palabras, que, de un total de 90 salidas, 75 salidas están por debajo del 16.6%.

Figura 8. Frecuencia de error en las salidas de la RNA en los diferentes intervalos



La Figura 9 muestra un gráfico de dispersión creado mediante el editor Chart Plotly, que ofrece la posibilidad de crear gráficos interactivos, donde las burbujas reemplazan los puntos de datos. Cada dato representado este dado por una coordenada compuesta por

un valor absoluto en el eje x donde se describe la diferencia entre el tráfico real y el obtenido por la red y en cuanto al eje y se ubica la escala de tráfico aproximado. Los indicadores ubicados en la parte derecha de la ilustración corresponden a los peajes manteniendo la estructura con la que se organizaron los datos. Es decir, las convenciones se refieren a las casetas de peaje en el siguiente orden:

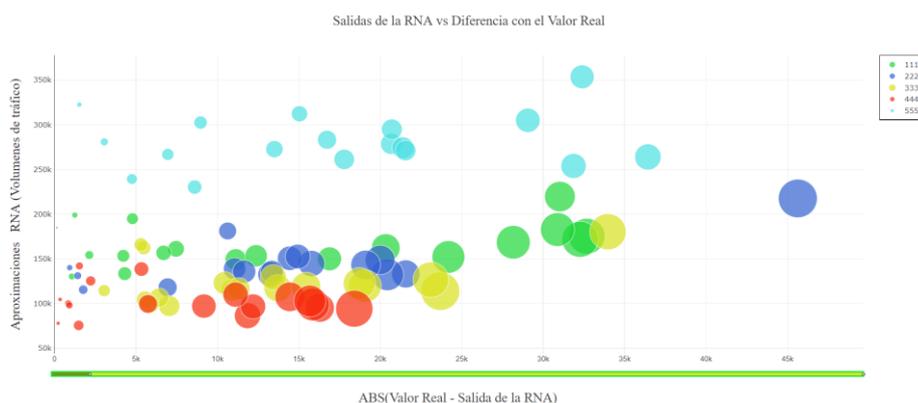
- Los Curos
- Curití
- Oiba
- Saboya
- Casablanca

Además, cada burbuja es directamente proporcional al error en cada predicción. Es por esa razón que al ubicarse sobre una de las burbujas de menor tamaño se puede encontrar hasta un error mínimo de 0.08% donde el desfase de vehículos en la aproximación es de tan solo 146 de una cifra muy cercana a los 200000.

Dado que la gráfica es interactiva, en ella se puede visualizar más detalles de los datos con tan sólo posicionar el clic sobre alguna de las burbujas, la información desplegada corresponde al volumen de tráfico aproximado, el año, el mes, el peaje y el porcentaje de error.

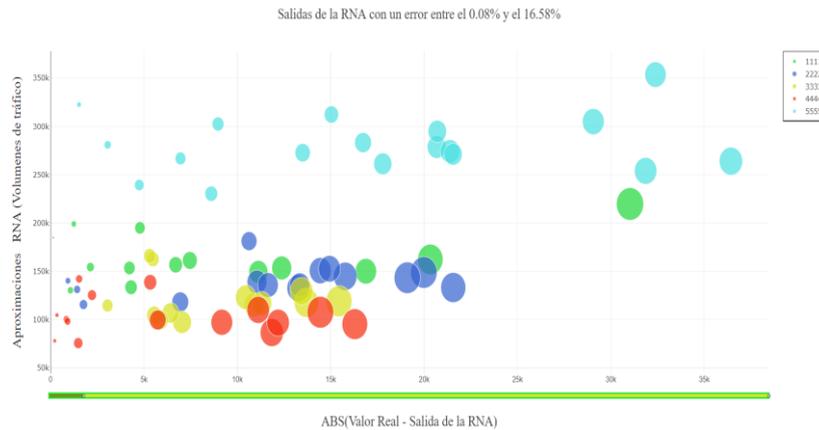
Como parte del análisis se puede corroborar que el peaje con indicador 5555 correspondiente a Casablanca contiene las predicciones con menor porcentaje de error. También, que el mayor volumen de tráfico se concentra en este peaje y el menor en Saboya; la mayor diferencia de vehículos es de 45635 y solo es una instancia, la gran mayoría parece ubicarse por debajo de los 20000 vehículos.

**Figura 9. Aproximación del flujo vehicular ante datos desconocidos en la fase de prueba vs la diferencia con el valor real**



Adicionalmente, en la Figura 10 se aplicó un filtro para visualizar únicamente las aproximaciones con un error entre el 0.08% y el 16.58% correspondiente al 83.4% de las salidas. De lo cual, el 77.77% de las salidas por el modelo se concentra en diferencias de 1000, 300, 10000 vehículos sin superar los 21000 constatando lo que se veía un poco confuso en el histograma de error inicial ofrecido por Matlab como resultado de la RNA.

**Figura 10. Salidas de la RNA con un error entre 0.08% y 16.58%**

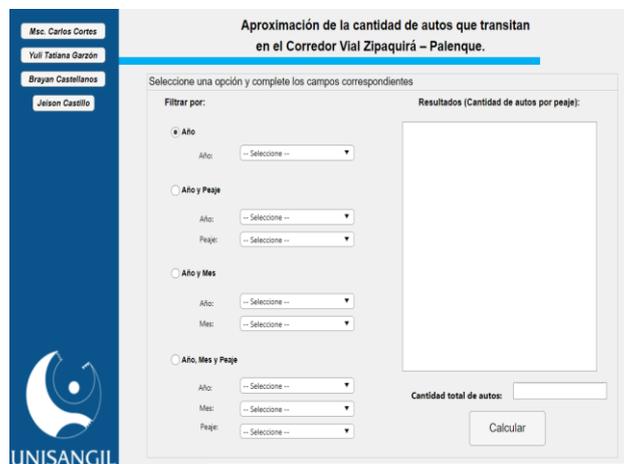


En ejercicio de hacer uso de la red neuronal elaborada, se diseñó una interfaz (Figura 11) que permita mostrar los datos de predicción a partir de la selección del peaje y un intervalo de tiempo específico.

**El usuario tendrá la posibilidad de observar los datos en las siguientes clasificaciones:**

- Los datos de todos los peajes por año.
- Datos del año en un peaje determinado.
- Datos de todos los peajes en un año y mes específico.
- Datos de año mes y peaje.

**Figura 11. Interfaz diseñada en Matlab para la implementación y uso de la Red Neuronal.**

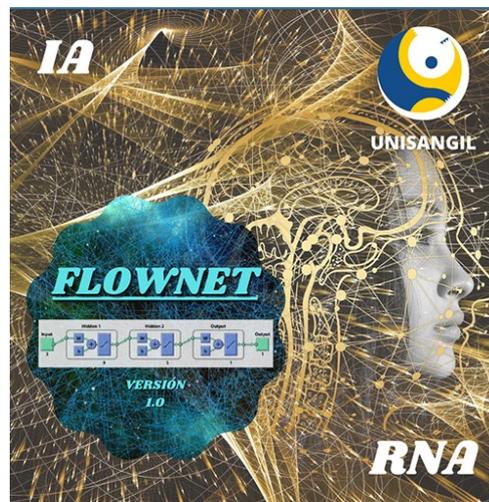


Con la interfaz ya funcional se optó por consolidarla en una aplicación de escritorio que se denominó FlowNet versión 1.0, el proceso de instalación se llevó a cabo en una máquina virtual con Windows 7. La Figura 12 muestra el icono de ejecución, y la Figura 13 representa la imagen de carga que aparece luego de correr la app. También es necesario destacar que para lanzar el programa se requiere tener instalado el compilador de Matlab en su versión 9.8 y, además, un equipo con mínimo 4 gb en RAM, y alrededor de 10 gb de espacio disponible en disco.

Figura 12. Icono en el escritorio luego de instalar Flownet 1.0



Figura 13. Imagen de carga de la app



Por otra parte, a fin de extrapolar el marco de trabajo realizado en Matlab a Python, se modeló una red neuronal con una capa de entrada, dos capas ocultas y una capa de salida, cuyo número de neuronas por capa corresponde a 16, 8, 8 y 1, respectivamente. Adicionalmente, después de entrenarla, validarla y evaluarla, se construyó una app utilizando el framework Flask la cual fue conectada con el modelo desarrollado en Python mediante la librería Tensorflow 2.1.0 a un formulario HTML para que cualquier usuario pueda interactuar con el modelo. La Figura 14 ilustra la página de inicio.

Figura 14. Menú de inicio de la Flask app

## FLASK APP RUNNING

La siguiente app realiza aproximaciones de flujo vehicular en el corredor vial Zipaquirá-Paleaque en Colombia mediante una Red Neuronal Artificial, se recomienda que el usuario ingrese años desde el 2016 en adelante y meses numericos como por ejemplo 1 en el caso de Enero y finalmente los peajes quedan a su elección de acuerdo a la lista de abajo

- Caseta de peaje los Cueros  
identificador: 1111
- Caseta de peaje Curiti  
identificador: 2222
- Caseta de peaje Olba  
identificador: 3333
- Caseta de peaje Saboya  
identificador: 4444
- Caseta de peaje Casablanca  
identificador: 5555

A continuación ingrese un año, un mes y un peaje, recuerde que las entradas son netamente numéricas como se indica anteriormente

Año

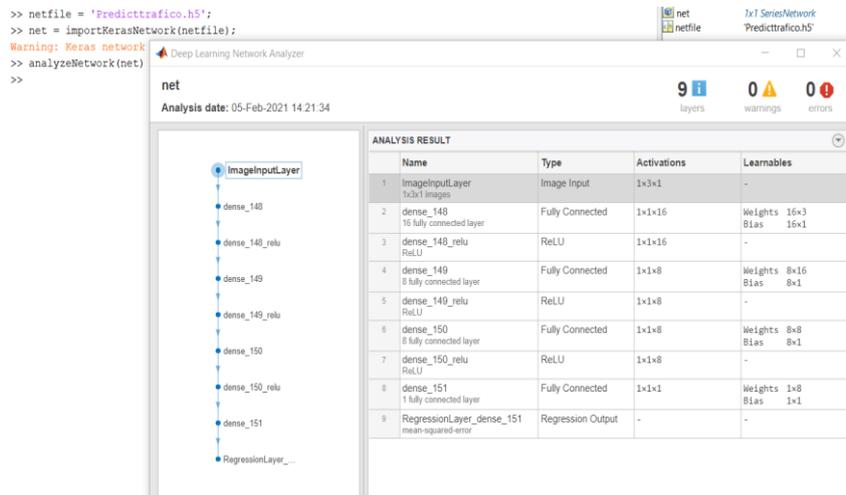
Mes

Peaje

Analyze

Finalmente, en pro de determinar la interoperabilidad a través del intercambio de modelos con otros frameworks se hizo uso del Deep Learning Toolbox de Matlab para importar el modelo desarrollado en Python, el proceso descrito fue exitoso y se ilustra en la Figura 15.

Figura 15. Importando el modelo de Python a Matlab



## CONCLUSIONES

El cambio en la función de entrenamiento, pasando del gradiente conjugado escalado (trainscg) a Levenberg-Marquardt (trainlm) otorgó un mejor desempeño en las aproximaciones de la RNA, evidenciado en un 83.4% de las salidas con un error por debajo

del 16.58%; lo anterior tiene sentido, pues según la documentación de Matlab este suele ser el algoritmo de aprendizaje más rápido de su kit de herramientas (Toolbox)

Mediante la gráfica de desempeño se logró evidenciar que en la iteración 121 el rendimiento de la validación alcanzó un mínimo. Posteriormente, el entrenamiento continuó durante 100 iteraciones más, debido al valor establecido en el parámetro `max_fail=100` que significa el número máximo de comprobaciones en las cuales el error de la validación puede aumentar antes de que se detenga el entrenamiento, por ende, se alcanzaron 221 iteraciones.

Agregando a lo anterior, gracias a la función de desempeño se determina que no hubo sobreajuste (overfitting), puesto que las curvas de validación y prueba muestran un comportamiento simétrico y se mantienen a lo largo de las iteraciones con pequeñas variaciones hasta la parada del entrenamiento.

En las regresiones resultantes de las fases de entrenamiento, validación y pruebas respectivamente, se pudo concluir que la precisión y la confianza en las aproximaciones tienen lugar cuando la línea discontinua que representa el resultado perfecto y la línea sólida que representa la regresión coinciden,

A través de comparaciones entre diferentes modelos se evidencia un R casi igual a 1 indicando una relación lineal casi perfecta entre las salidas y los valores reales, pero los otros modelos con un R menor conseguían predicciones más certeras ante datos desconocidos porque la función identidad convergía con la trayectoria discontinua dada por los valores objetivo.

Dado que el modelo respondió de buena forma a los datos de entrenamiento, se sugiere dar continuidad al trabajo llevándolo a una instancia donde se intente predecir otra variable como el recaudo total, información relevante en busca de regular el cobro y tarifas de peajes en relación al estado de las vías, asegurando transparencia y beneficiando los sectores de transporte, agropecuarios, turismo, entre otros.

#### REFERENCIAS BIBLIOGRÁFICAS

Matich Damián, "Redes Neuronales: Conceptos Básicos y Aplicaciones.," *Historia Santiago.*, p. 55, 2001, [Online]. Available: <ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf>

Agencia Nacional de Infraestructura, "PROYECTOS DE INFRAESTRUCTURA -

- CARRETERAS,” 2016. <https://www.ani.gov.co/proyectos-de-infraestructura-carreteras>
- Mathworks, “Introducción a MATLAB,” 2020. <https://es.mathworks.com/help/matlab/getting-started-with-matlab.html>
- D. Calvo, “Función de activación – Redes neuronales,” 2018. [https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/#:~:text=La función de activación se,o \(-1%2C1\)](https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/#:~:text=La función de activación se,o (-1%2C1))
- E. Feire and S. Silva, “Redes Neuronales,” 2019. <https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb#:~:text=Función Lineal&text=Por lo tanto%2C esta función,de un número de ventas>
- J. Martinez Heras, “Error Cuadrático Medio para Regresión,” 2020. <https://www.iartificial.net/error-cuadratico-medio-para-regresion/>