Introducción

En el contexto del creciente interés y aplicación de técnicas de aprendizaje automático, el desarrollo de aplicaciones basadas en aprendizaje supervisado se ha vuelto fundamental para resolver una amplia variedad de problemas. En este trabajo, se aborda el desafío de desarrollar una aplicación centrada en el aprendizaje supervisado, diseñada específicamente para resolver problemas sencillos mediante el método de corrección de error.

La aplicación propuesta tiene como objetivo principal proporcionar una herramienta efectiva y accesible para la resolución de problemas simples, donde el conjunto de patrones de entrada y salida se carga previamente en un archivo. La aplicación debe ser capaz de procesar cualquier cantidad de entradas y salidas, lo que la hace versátil y adaptable a una amplia gama de situaciones.

Uno de los aspectos clave de la aplicación es la implementación de la gráfica del error de la iteración durante el proceso de simulación. Esta característica permite monitorear y analizar el progreso del entrenamiento de la red neuronal, lo que facilita la comprensión de su rendimiento y la identificación de posibles mejoras.

Objetivos

- Desarrollar una aplicación basada en aprendizaje supervisado para resolver problemas sencillos mediante el método de corrección de error.

- Permitir la carga previa de conjuntos de patrones de entrada y salida desde archivos para su procesamiento.

- Implementar una red neuronal capaz de procesar cualquier cantidad de entradas y salidas, lo que garantiza la flexibilidad y la adaptabilidad de la aplicación.

- Incluir la funcionalidad de generar y visualizar la gráfica del error de la iteración durante el proceso de simulación, facilitando el análisis y la optimización del entrenamiento de la red.

Análisis del problema

La problemática consiste en desarrollar una aplicación que utilice técnicas de aprendizaje supervisado basadas en corrección de error para resolver problemas simples. La aplicación debe ser capaz de cargar conjuntos de datos previamente almacenados en archivos y utilizarlos para entrenar y simular una red neuronal artificial. La red neuronal debe ser capaz de procesar cualquier cantidad de entradas y salidas, y durante el proceso de simulación, la aplicación debe mostrar una gráfica que represente el error de la iteración para facilitar el análisis y seguimiento del proceso de entrenamiento.

Requerimientos Funcionales:

- **Carga de Datos desde Archivos:** La aplicación debe permitir cargar conjuntos de datos de entrenamiento desde archivos previamente almacenados en el sistema. Estos archivos contendrán patrones de entrada y salida etiquetados que se utilizarán para entrenar la red neuronal.

- **Establecer los parámetros de entrenamiento:** La aplicación debe permitir establecer los parámetros de entrenamiento. Los parámetros clave a configurar en el proceso de entrenamiento son:

    - ***Número de Iteraciones:*** Representa la cantidad de veces que el conjunto de entrenamiento completo se procesa durante el entrenamiento. Un número suficiente de iteraciones garantiza que la red neuronal tenga la oportunidad de converger hacia una solución óptima. Sin embargo, un número excesivo de iteraciones puede llevar al sobreajuste del modelo a los datos de entrenamiento.

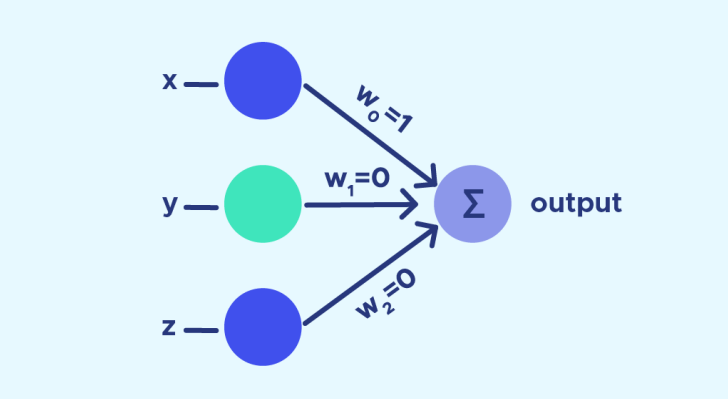
    - ***Tasa de Aprendizaje:*** Controla la magnitud de los ajustes realizados en los pesos de la red durante cada paso de entrenamiento. Una tasa de aprendizaje más alta puede acelerar la convergencia del modelo, pero también puede provocar oscilaciones o divergencias. Por otro lado, una tasa de aprendizaje más baja puede garantizar una convergencia más estable, pero el entrenamiento puede ser más lento.

    - ***Tolerancia al Error:*** Especifica la precisión deseada o la cantidad aceptable de error que se puede tolerar durante el entrenamiento. Durante el proceso de entrenamiento, la red intentará minimizar este error, ajustando los pesos y los sesgos de las neuronas para reducir la discrepancia entre las predicciones y las salidas reales. Una vez que el error alcanza o se acerca al valor de tolerancia especificado, el entrenamiento puede detenerse, ya que es poco probable que la red mejore significativamente más allá de este punto.

- **Diseño de la red neuronal:** La aplicación hará uso de una red neuronal de aprendizaje supervisado monocapa basada en corrección de error. Esta es un tipo de red neuronal artificial que consta de una sola capa de neuronas, donde cada neurona está conectada a todas las entradas y produce una única salida. Este tipo de red es adecuada para problemas simples donde la relación entre las entradas y las salidas puede modelarse de manera lineal o con funciones de activación simples.

  En una red neuronal de una sola capa, las entradas se multiplican por los pesos de las conexiones y se suman junto con un sesgo. La suma ponderada se pasa luego a través de una función de activación (en este caso una función de tipo limitador duro), que determina la salida de la neurona. La salida de todas las neuronas se combina linealmente para producir la salida final de la red.

  Durante el entrenamiento, la red ajusta sus pesos y sesgos para minimizar la diferencia entre las salidas predichas y las salidas reales utilizando un algoritmo de corrección de error, como el descenso de gradiente. Este proceso implica calcular el gradiente de la función de pérdida con respecto a los pesos y ajustarlos en la dirección que minimice la pérdida.



- **Entrenamiento de la Red Neuronal:** Una vez que los datos de entrenamiento se cargan en la aplicación, se debe permitir entrenar una red neuronal utilizando estos datos. El proceso de entrenamiento se basará en técnicas de aprendizaje supervisado, donde la red aprenderá a partir de ejemplos etiquetados, minimizando el error entre las salidas predichas y las salidas reales.

- **Simulación de la Red Neuronal:** Después de entrenar la red neuronal, la aplicación debe permitir simular su funcionamiento utilizando nuevos conjuntos de datos. Durante la simulación, la red deberá procesar las entradas proporcionadas y generar salidas predichas. Estas predicciones se compararán con las salidas reales para evaluar el rendimiento de la red.

- **Gráfica del Error de la Iteración:** Durante el proceso de entrenamiento y simulación, la aplicación debe mostrar una gráfica que represente el error de la iteración. Esta gráfica proporcionará una visualización del progreso del entrenamiento y permitirá identificar cualquier tendencia o patrón en el error a lo largo del tiempo.

Requerimientos No Funcionales:

- **Interfaz de Usuario Intuitiva:** La aplicación debe contar con una interfaz de usuario fácil de usar que permita a los usuarios cargar datos, configurar el entrenamiento de la red y visualizar los resultados de manera clara y comprensible.

- **Eficiencia Computacional:** El proceso de entrenamiento y simulación de la red neuronal debe ser eficiente en términos de uso de recursos computacionales, para garantizar un rendimiento óptimo incluso con grandes conjuntos de datos.

- **Escalabilidad:** La aplicación debe ser capaz de manejar conjuntos de datos de cualquier tamaño, así como ajustar la arquitectura de la red neuronal para adaptarse a diferentes problemas y requisitos de rendimiento.

El desarrollo de esta aplicación requerirá la implementación de algoritmos y técnicas de aprendizaje automático, así como una sólida comprensión de los principios detrás de las redes neuronales artificiales y el aprendizaje supervisado. Además, se necesitarán habilidades de programación y diseño de software para crear una aplicación robusta y funcional que cumpla con los requisitos especificados.

Manual de usuario

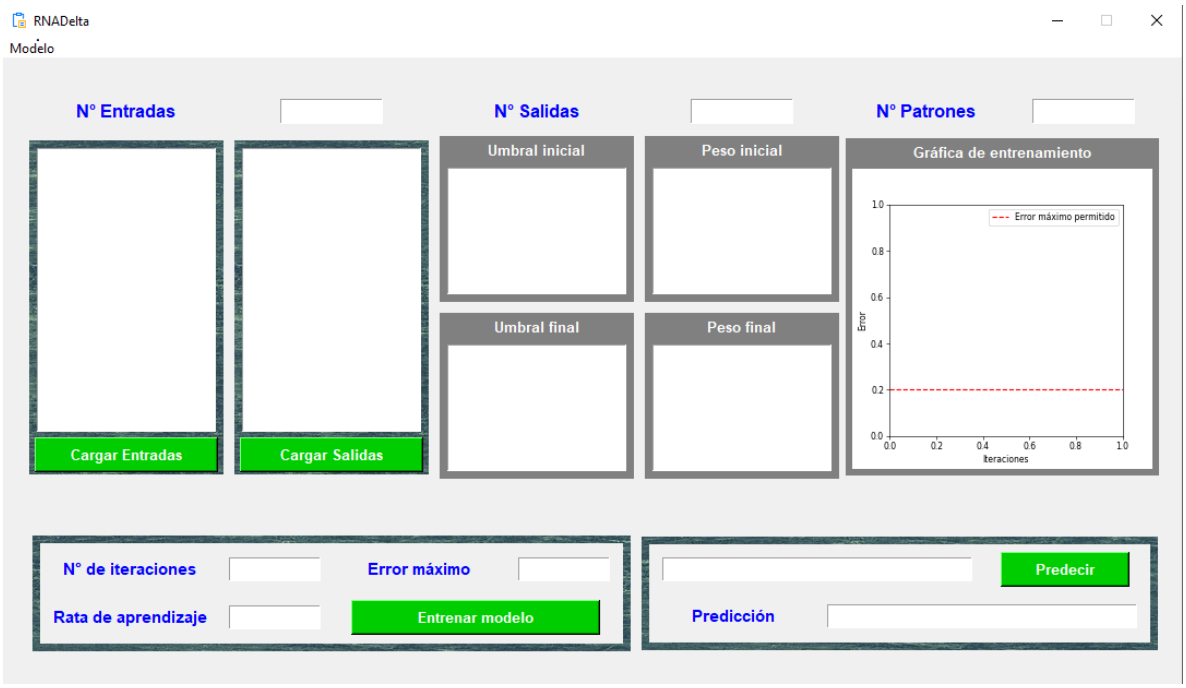
Esta aplicación está diseñada para proporcionar una herramienta intuitiva y poderosa para entrenar y simular redes neuronales artificiales de una sola capa, utilizando el método de corrección de error.

El aprendizaje supervisado es una técnica fundamental en el campo del aprendizaje automático, donde se enseña a un modelo a realizar predicciones basadas en ejemplos de entrada y salida conocidos. En el caso de las redes neuronales, este proceso implica ajustar los pesos de las conexiones entre las neuronas para minimizar la diferencia entre las predicciones del modelo y las salidas reales.

Esta aplicación ofrece una interfaz fácil de usar que permite cargar conjuntos de datos de entrenamiento y ajustar los parámetros de entrenamiento. Además, proporciona herramientas de visualización para analizar el progreso del entrenamiento y entender mejor el comportamiento de la red neuronal durante la simulación.

A lo largo de este manual, se proporcionarán instrucciones detalladas sobre cómo utilizar cada función de la aplicación, así como recomendaciones y mejores prácticas para obtener los mejores resultados en sus tareas de entrenamiento y simulación de redes neuronales.

Ventana de la aplicación

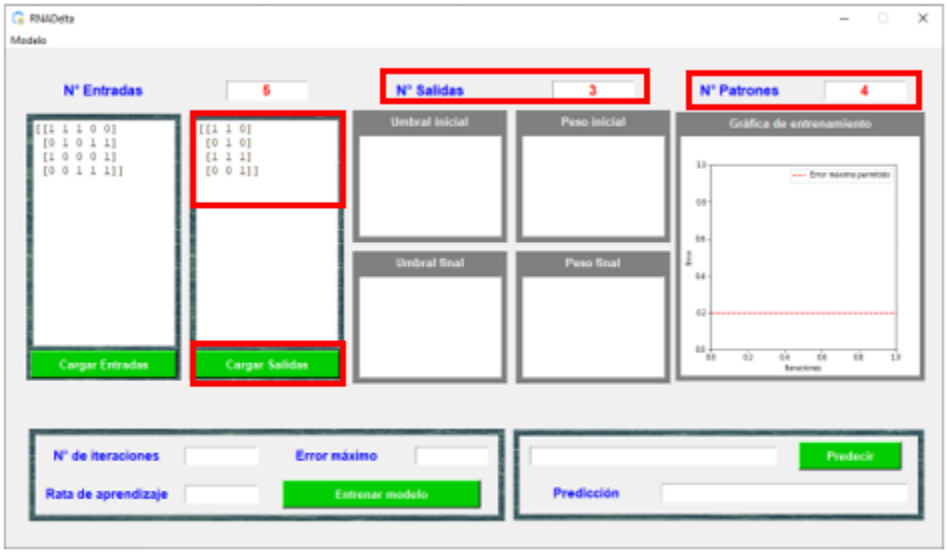


Cargar entradas

El botón cargar entradas nos permite cargar las entradas que tengamos disponibles en el banco de datos para el posterior entrenamiento de la red neuronal. Una vez cargadas las entradas estas serán desplegadas en el área superior al botón. También se desplegará el número de entradas del conjunto de datos.

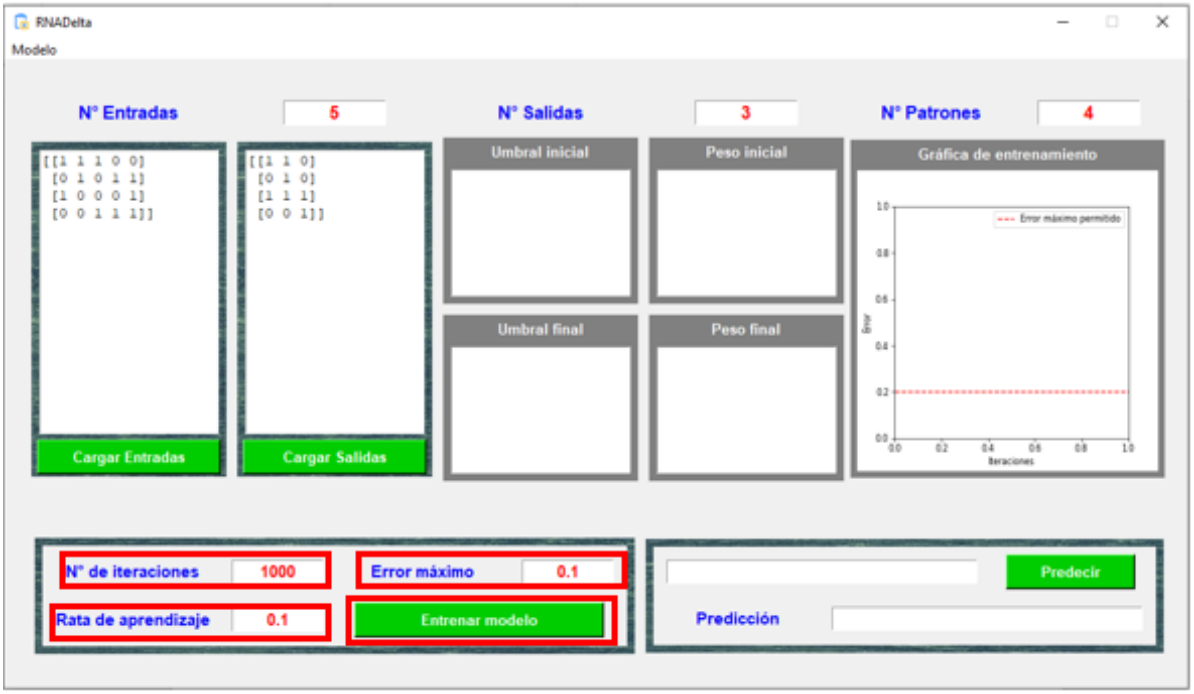
Cargar salidas

El botón cargar salidas nos permite cargar las salidas que tengamos disponibles en el banco de datos para el posterior entrenamiento de la red neuronal. Una vez cargadas las salidas estas serán desplegadas en el área superior al botón. También se desplegará el número de salidas y el número de patrones del conjunto de datos.



Entrenar modelo

Esta opción nos permite realizar el entrenamiento del modelo por medio de los parámetros exigidos por la interfaz. La aplicación tomará los parámetros ingresados por el usuario y entrenará el modelo basado en esos parámetros usando el banco de datos cargado previamente.



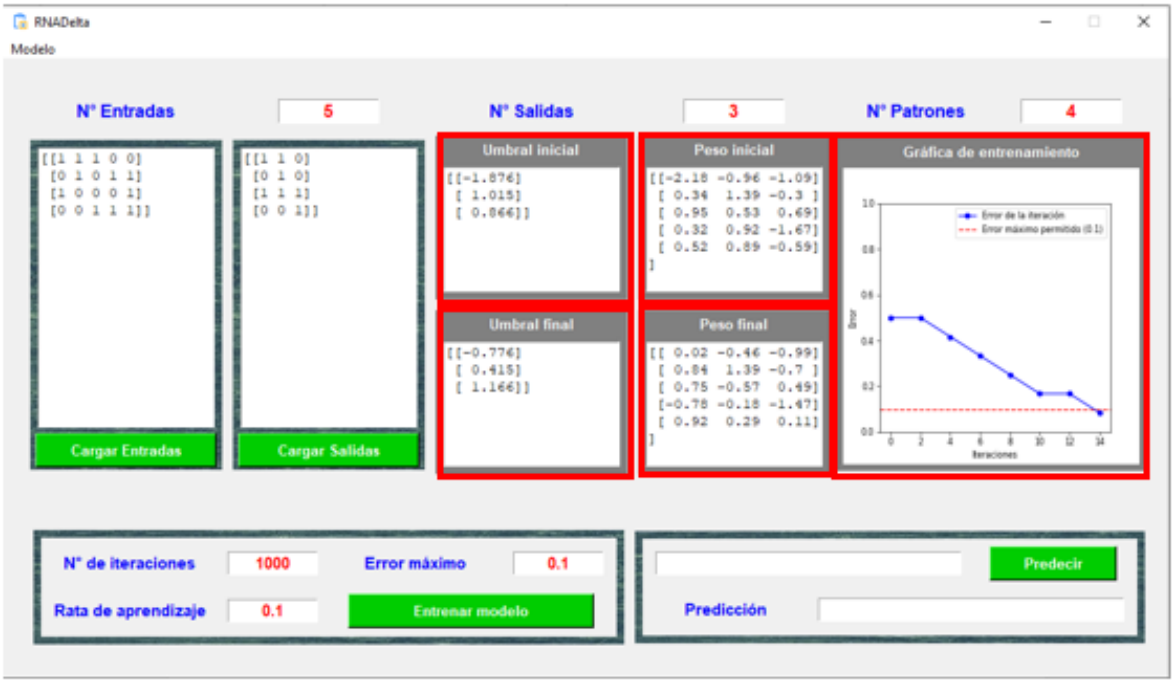
- **N° de iteraciones:** El número de iteraciones representa la cantidad de veces que se procesa todo el conjunto de datos de entrenamiento durante el entrenamiento de la red. Cada iteración consiste en un paso hacia adelante y un paso hacia atrás a través de la red, seguido de la actualización de los pesos basada en el error cometido. Un número adecuado de iteraciones garantiza que la red tenga la oportunidad de converger hacia una solución óptima sin sobreajustarse a los datos de entrenamiento.

- **Error máximo:** El error máximo, también conocido como tolerancia al error, especifica la precisión deseada o la cantidad aceptable de error que se puede tolerar durante el entrenamiento. Durante el proceso de entrenamiento, la red intentará minimizar este error, ajustando los pesos y los sesgos de las neuronas para reducir la discrepancia entre las salidas predichas y las salidas reales. Una vez que el error alcanza o se acerca al valor máximo definido, el entrenamiento puede detenerse, ya que es poco probable que la red mejore significativamente más allá de este punto.

- **Rata de aprendizaje:** La tasa de aprendizaje es un parámetro que controla la magnitud de los ajustes realizados en los pesos de la red durante cada paso de entrenamiento. Una tasa de aprendizaje más alta puede permitir que la red converja más rápidamente, pero también puede provocar oscilaciones o divergencias en el proceso de optimización. Por otro lado, una tasa de aprendizaje más baja puede garantizar una convergencia más suave y estable, pero el entrenamiento puede ser más lento. En la práctica, encontrar el valor óptimo de la tasa de aprendizaje es crucial para garantizar un entrenamiento eficiente y efectivo de la red neuronal.

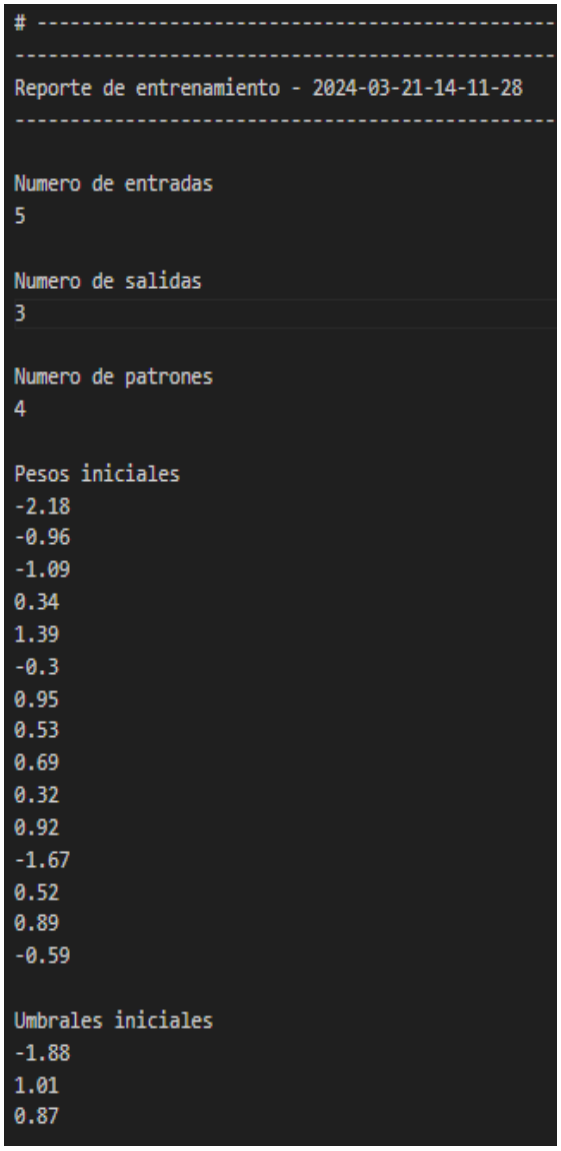
Gráfica de entrenamiento

Una vez realizado el entrenamiento del modelo, se mostrará en pantalla los umbrales y pesos iniciales (escogidos aleatoriamente), como también los umbrales y pesos finales (cuando el modelo ya ha alcanzado el error mínimo o el máximo de iteraciones). También se desplegará la gráfica correspondiente al entrenamiento del modelo. Esta gráfica proporcionará una visualización del progreso del entrenamiento y permitirá identificar cualquier tendencia o patrón en el error a lo largo del tiempo.

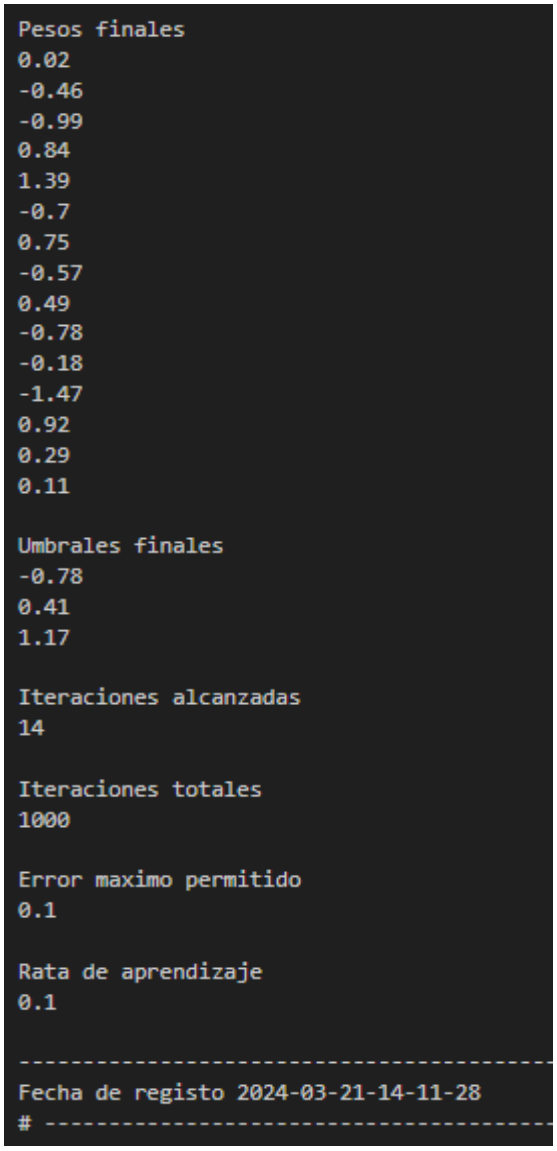


Reporte

También se generará un informe breve con los datos del entrenamiento del modelo de la red neuronal, donde se especificarán: el número de entradas, número de salidas, numero de patrones, umbrales iniciales, pesos iniciales, umbrales finales, pesos finales, iteraciones alcanzadas, iteraciones totales, error máximo permitido y la rata de aprendizaje.



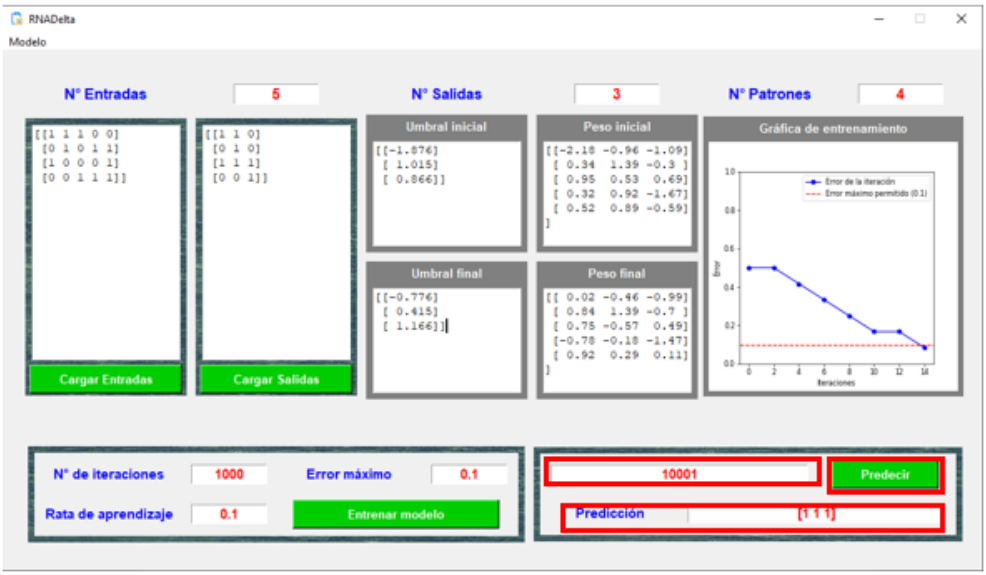
```
# ------------------------------------------------
------------------------------------------------
Reporte de entrenamiento - 2024-03-21-14-11-28
------------------------------------------------

Numero de entradas
5

Numero de salidas
3

Numero de patrones
4

Pesos iniciales
-2.18
-0.96
-1.09
0.34
1.39
-0.3
0.95
0.53
0.69
0.32
0.92
-1.67
0.52
0.89
-0.59

Umbrales iniciales
-1.88
1.01
0.87
```



```
Pesos finales
0.02
-0.46
-0.99
0.84
1.39
-0.7
0.75
-0.57
0.49
-0.78
-0.18
-1.47
0.92
0.29
0.11

Umbrales finales
-0.78
0.41
1.17

Iteraciones alcanzadas
14

Iteraciones totales
1000

Error maximo permitido
0.1

Rata de aprendizaje
0.1

----------------------------------------
Fecha de registo 2024-03-21-14-11-28
# ----------------------------------------
```

Predecir

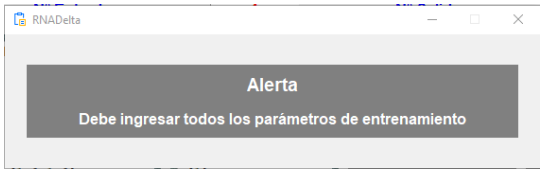
Esta opción nos permite realizar una predicción una vez que el modelo este entrenado, solo se requiere ingresar el patrón al que se requiere realizar la predicción. La predicción resultante se desplegará en el cuadro de texto inferior.

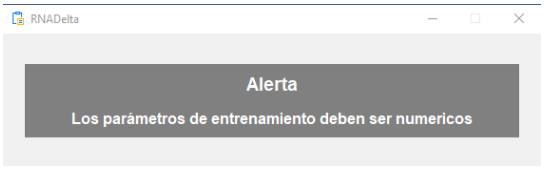


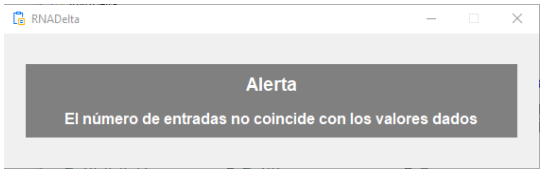
Validaciones

El programa realiza las siguientes validaciones:

- Verifica que se hayan ingresado las entradas y salidas antes de realizar el entrenamiento del modelo.
- Verifica que se hayan ingresado los parámetros de entrenamiento antes de realizar al entrenamiento del modelo.
- Verifica que los parámetros ingresados sean numéricos.
- Verifica que el patrón a predecir corresponda al número de entradas del modelo.

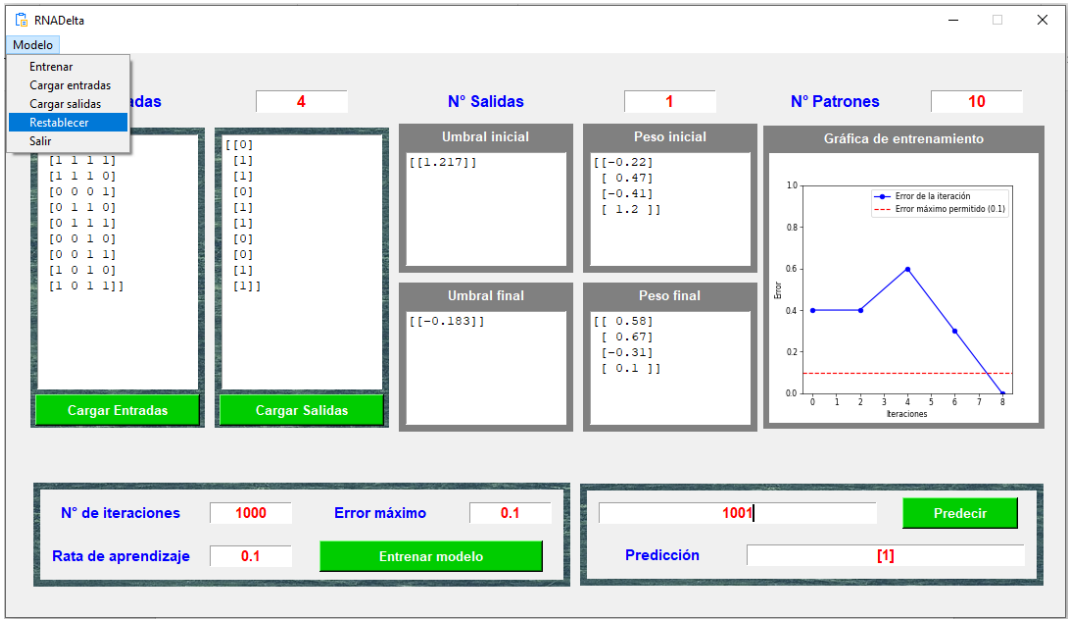






Restablecer

Esta opción nos permite restablecer todos los valores de la aplicación para realizar una nueva configuración del modelo de red neuronal. Todos los parámetros y datos del modelo serán eliminados.



Conclusión

En el transcurso del desarrollo de este informe, se presentó una descripción detallada de la aplicación propuesta, incluyendo su diseño, implementación y funcionalidades clave. Se proporcionó una explicación breve de los conceptos de aprendizaje supervisado, así como una guía paso a paso sobre cómo utilizar la aplicación para resolver problemas específicos.

Además, se presentó ejemplos de aplicación y casos de uso práctico, demostrando la eficacia y la versatilidad de la herramienta en una variedad de situaciones del mundo real. Finalmente, pudimos identificar el desarrollo de áreas futuras para mejorar y ampliar la aplicación.

Este trabajo representa un esfuerzo significativo para desarrollar una aplicación de aprendizaje supervisado centrada en la resolución de problemas sencillos, con un enfoque en la flexibilidad, la eficacia y la facilidad de uso.

Referencias bibliográficas

- Las Redes Neuronales Artificiales y su importancia como herramienta en la toma de decisiones. Villanueva Espinoza, María del Rosario. https://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/basic/Villanueva_EM/enPDF/Cap5.PDF

- Procesamiento Digital de Imágenes. Pablo Roncagliolo B. (2007). http://www2.elo.utfsm.cl/~elo328/PDI21_RedesNeuronales.pdf

- Fundamentos de las Redes Neuronales. Luis Moreno, Santiago Garrido, Dorin Copaci. (2009). https://ocw.uc3m.es/pluginfile.php/143/mod_page/content/24/Fundamentos%20de%20las%20Redes%20Neuronales.%20Parte%20II%20L7.pdf