Primer Parcial de Electiva de Ingeniería de Software

Juan Camilo Rodríguez Caicedo

Arturo Barona

Cali, Valle del Cauca, Colombia

Resumen –

# Introducción

## Contexto general

En el campo de la inteligencia artificial y el aprendizaje automático, las redes neuronales se han consolidado como herramientas poderosas para resolver una amplia gama de problemas. Este informe aborda dos aplicaciones prácticas de las redes neuronales monocapa, cada una con objetivos y desafíos específicos que ilustran la versatilidad y las limitaciones de estas redes en diferentes contextos.

El primer problema planteado se centra en la **clasificación del alfabeto en mayúscula.** El objetivo principal es entrenar una red neuronal de una capa para reconocer letras mayúsculas del alfabeto, representadas en matrices de 5x5 llenas de bits binarios (1 o 0). Cada letra tendrá una representación única en esta matriz, y la red neuronal deberá generar un código binario de 5 bits que corresponda a cada letra. Este problema implica varias tareas clave: definir la representación de cada letra, desarrollar y entrenar la red para la clasificación, evaluar su rendimiento utilizando un conjunto de prueba, y analizar el impacto del ruido en las entradas sobre el desempeño de la red. Este ejercicio permite explorar cómo una red neuronal monocapa maneja la clasificación de patrones simples y cómo la robustez del modelo puede verse afectada por perturbaciones en los datos.

El segundo problema se enfoca en la **separación de pistas de sonido**. Aquí, el objetivo es entrenar una red neuronal monocapa para separar tres fuentes de audio mezcladas en una única señal. Las señales originales incluyen dos voces humanas hablando simultáneamente y una tercera fuente proveniente de un instrumento musical o pista de audio. La tarea consiste en recuperar las señales individuales a partir de la mezcla lineal. Las actividades asociadas incluyen la definición de la estructura de entrada y salida de la red, el entrenamiento del modelo para aproximar las tres señales originales, la evaluación del rendimiento mediante el error cuadrático medio, y la visualización de los resultados a través de gráficos y la reproducción de los audios. Este problema aborda desafíos en el procesamiento de señales y demuestra cómo las redes neuronales pueden aplicarse a la separación de fuentes en el dominio del audio.

A través de estos problemas, el informe busca proporcionar una comprensión detallada del diseño, entrenamiento y evaluación de redes neuronales monocapa en contextos específicos, además de ilustrar las capacidades y limitaciones de estos modelos en la clasificación de patrones visuales y la separación de señales auditivas.

# Introducción al problema 1

## *Contexto del Problema 1*

La historia del reconocimiento de caracteres mediante redes neuronales comenzó con la creación del Perceptrón en 1958 por Frank Rosenblatt, inspirado en los trabajos de McCulloch y Pitts. Este modelo podía realizar tareas sencillas de clasificación binaria, utilizando pesos para determinar la importancia de cada entrada. En 1965, se introdujo el Perceptrón multicapa, que permitió expandir la capacidad de procesamiento al incluir capas de entrada, ocultas y de salida [1]. Con el pasar de los años, las redes neuronales se utilizan para la clasificación de patrones, incluyendo el reconocimiento de caracteres e imágenes, debido a su capacidad para aprender y generalizar a partir de datos. Estas redes son capaces de identificar características y patrones complejos dentro de los datos de entrada, lo que las hace útiles para tareas como el reconocimiento óptico de caracteres (OCR) [2] y la clasificación de imágenes. En el caso del reconocimiento de caracteres, una red neuronal se entrena con ejemplos de letras o números para aprender las características distintivas de cada uno. Una vez entrenada, la red puede reconocer caracteres en nuevas imágenes.

A pesar de los avances significativos en el campo del reconocimiento de caracteres, siguen existiendo desafíos, como la variabilidad en la presentación de los caracteres y la influencia del ruido en los datos de entrada. Las redes neuronales, y en particular las de una capa, han mostrado potencial para abordar estos problemas, pero aún se requiere una mayor investigación para mejorar su precisión y robustez. Este trabajo se centra en el desarrollo y evaluación de una red neuronal de una capa para la clasificación de letras mayúsculas del alfabeto, explorando su capacidad para manejar variaciones y ruidos en las entradas. Así, buscamos contribuir a la mejora de técnicas de reconocimiento de caracteres, que son fundamentales en aplicaciones modernas como la digitalización de documentos y el procesamiento automatizado de texto.

## Justificación del Problema 1

El reconocimiento preciso y eficiente de caracteres es una tarea crucial en numerosas aplicaciones. Las redes neuronales ofrecen una solución prometedora al aprender patrones complejos y adaptarse a diferentes representaciones de letras. Este estudio justifica la necesidad de explorar y mejorar estos modelos, dado su potencial para optimizar procesos que dependen del reconocimiento de caracteres, impactando positivamente en la accesibilidad y procesamiento de la información.

## Objetivo del Problema 1

Entrenar una red neuronal de una capa para reconocer letras mayúsculas del alfabeto.

# Metodología del problema 1

## Descripción del problema

El problema consiste en clasificar las letras mayúsculas del alfabeto utilizando una red neuronal de una capa. Las letras se representan mediante matrices de 5x5 que contienen bits (1 o 0), donde cada patrón único define una letra específica. El objetivo es entrenar la red neuronal para que reconozca y clasifique cada una de estas letras, asignándoles un código binario de 5 bits.

## Diseño del modelo

Se implementó un perceptrón multicapa con una sola capa oculta para este problema. El modelo toma las matrices 5x5 aplanadas como entradas y tiene una salida de 5 bits. Los pesos se inicializan aleatoriamente con valores pequeños y se ajustan durante el entrenamiento utilizando un algoritmo de retropropagación. El modelo fue diseñado con los siguientes parámetros:

* Número de entradas: 25 (5x5 matriz aplanada).
* Número de salidas: 5 bits para representar las letras en código binario.
* Tasa de aprendizaje: 0.1, para el ajuste de los pesos.

## Función de activación

Se utilizó la función de activación sigmoide para transformar la entrada ponderada de la red neuronal. Esta función se seleccionó debido a su capacidad para manejar problemas de clasificación binaria, permitiendo la salida en un rango entre 0 y 1, lo que es útil para interpretar los resultados como probabilidades.

## Entrenamiento del modelo

El entrenamiento se realizó utilizando el algoritmo de retropropagación con una tasa de aprendizaje de 0.1. Se entrenó la red durante 100 épocas, donde en cada época se realizaron los siguientes pasos:

* **Propagación hacia adelante:** Se calculó la salida de la red neuronal para cada entrada de la matriz 5x5 aplanada.
* **Cálculo del error:** Se determinó la diferencia entre la salida esperada y la salida actual de la red.
* **Propagación hacia atrás:** Se ajustaron los pesos y sesgos utilizando la derivada de la función de activación sigmoide y el error calculado.
* **Actualización de pesos y sesgo:** Los pesos y el sesgo se actualizaron basándose en el gradiente y la tasa de aprendizaje.

## Evaluación

La evaluación del modelo se realizó utilizando un conjunto de prueba independiente para medir su capacidad de generalización. Se calcularon métricas como la precisión y la exactitud para evaluar el desempeño del modelo en la clasificación correcta de las letras. Además, se introdujo ruido en las entradas para analizar cómo afecta el rendimiento de la red. Se observó el error cuadrático medio durante el entrenamiento y se generó una gráfica para visualizar la evolución del error a lo largo de las épocas.

# Desarrollo del problema 1

## Definición de las matrices de letras

Para representar las letras mayúsculas del alfabeto en un formato binario adecuado para la red neuronal, se utilizan matrices de 5x3 bits. Cada letra se mapea a una matriz de 5 filas por 3 columnas, donde cada elemento de la matriz es un bit (0 o 1). Estas matrices permiten que la red neuronal monocapa clasifique las letras basándose en patrones específicos en estas representaciones binarias. A continuación, se muestra la representación en matriz la letra T como ejemplo:

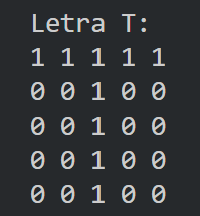


Ilustración 1. Representación en matriz de la letra T.

## Código del Perceptrón

### Inicialización

* El perceptrón comienza con pesos iniciales que generalmente se establecen en cero. Estos pesos se actualizan durante el proceso de entrenamiento.
* Se incluye un término adicional conocido como sesgo, que también se inicializa en cero. Este término ayuda a ajustar el umbral de activación del perceptrón.

### Predicción

* Para hacer una predicción, el perceptrón calcula una combinación lineal de las entradas utilizando los pesos actuales y el sesgo.
* La combinación lineal se pasa a través de una función de activación. En el caso del perceptrón, la función de activación es una función escalón que produce una salida de 1 si la combinación lineal es mayor o igual a cero, y 0 en caso contrario.

### Entrenamiento

* Durante el entrenamiento, el perceptrón ajusta sus pesos y el sesgo en función del error de predicción.
* Para cada ejemplo en el conjunto de datos de entrenamiento, se calcula la diferencia entre la etiqueta real y la predicción realizada por el perceptrón.
* Los pesos y el sesgo se actualizan utilizando esta diferencia, con el objetivo de reducir el error en futuras predicciones. El proceso de ajuste se realiza mediante un parámetro llamado tasa de aprendizaje, que determina la magnitud de las actualizaciones.

### Evaluación

* Después del entrenamiento, el perceptrón se evalúa utilizando un conjunto de datos de prueba para verificar su capacidad de clasificación. La precisión de las predicciones se mide para evaluar el desempeño del modelo.

Este proceso permite que el perceptrón aprenda a clasificar patrones binarios, como las representaciones de letras en formato binario, ajustando sus parámetros para mejorar la exactitud de sus predicciones.

## Entrenamiento

El proceso de entrenamiento del perceptrón se realiza mediante la presentación de un conjunto de datos de entrenamiento que incluye las representaciones binarias de las letras y sus correspondientes etiquetas. El entrenamiento se lleva a cabo mediante el algoritmo de ajuste de pesos del perceptrón, que busca minimizar el error de predicción.

* **Datos de Entrenamiento:** Se utilizan matrices binaras de 5x3 correspondientes a cada letra, codificadas en vectores de 15 bits.
* **Ajuste de Pesos:** El perceptrón ajusta sus pesos en función del error entre la predicción y la etiqueta real. Se utiliza una tasa de aprendizaje de 0.01 para actualizar los pesos durante el entrenamiento.
* **Evaluación:** Después del entrenamiento, se evalúa el rendimiento del perceptrón en un conjunto de prueba para verificar su capacidad de generalización. Se mide la precisión de la clasificación y se ajustan los parámetros según sea necesario para mejorar el rendimiento.

Este proceso garantiza que el perceptrón pueda identificar y clasificar correctamente las letras en función de sus representaciones binarias.

# Resultados del problema 1

## Rendimiento del modelo

Durante el entrenamiento de la red neuronal, el Error Cuadrático Medio (MSE) evolucionó de esta manera:

* Época 0: MSE = 0.2517
* Época 100: MSE = 0.0824
* Época 200: MSE = 0.0630
* Época 300: MSE = 0.0525
* Época 400: MSE = 0.0450
* Época 500: MSE = 0.0392
* Época 600: MSE = 0.0346
* Época 700: MSE = 0.0308
* Época 800: MSE = 0.0278
* Época 900: MSE = 0.0252

Se observa una disminución constante del MSE a lo largo de las épocas, lo que indica una mejora continua en el rendimiento del modelo durante el entrenamiento.

## Gráficas de errores

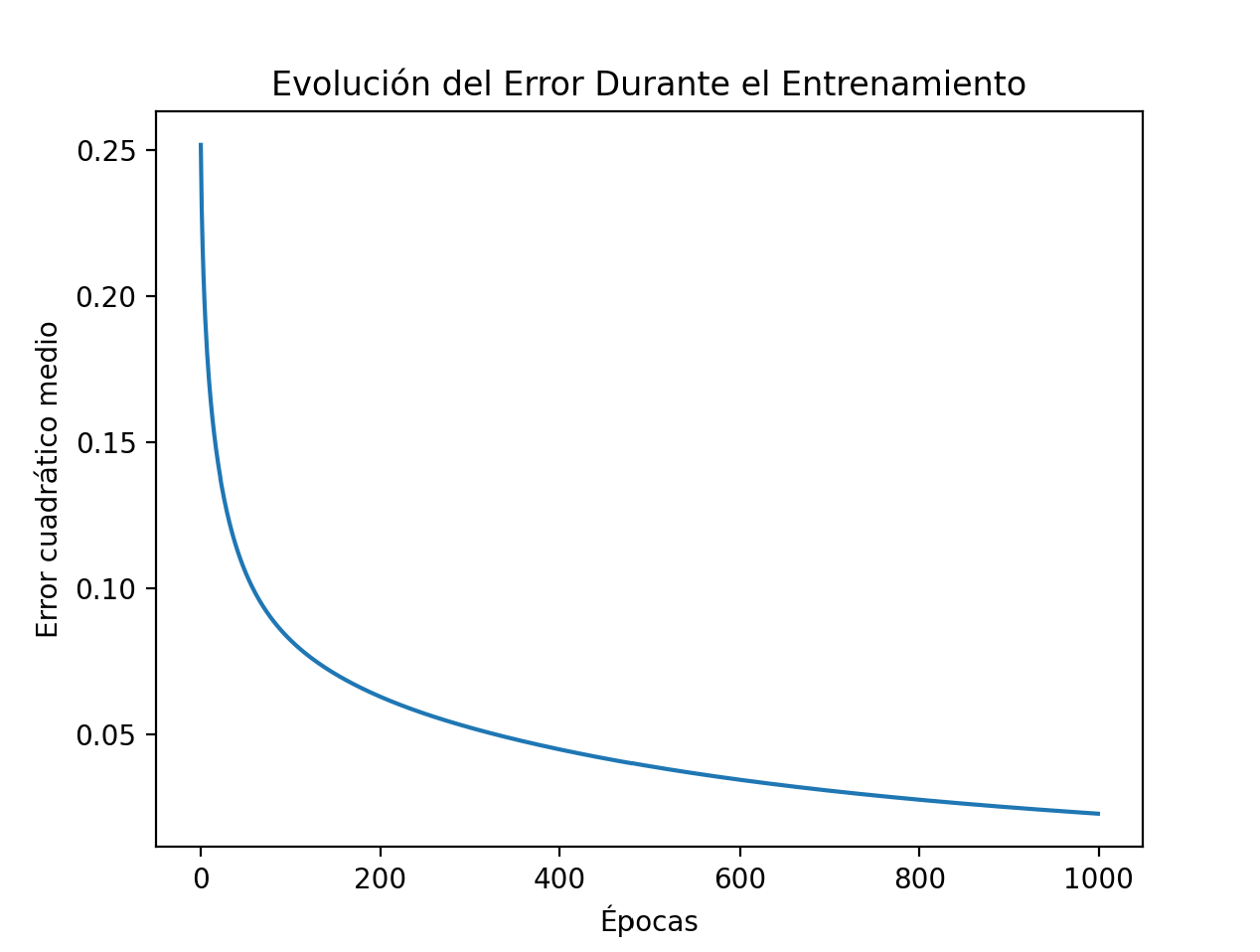


Ilustración 2. MSE en función de las épocas arrojada por el Software.

La gráfica muestra la reducción del MSE a lo largo de las épocas. El MSE disminuye progresivamente, lo que indica que el modelo está aprendiendo y ajustando sus parámetros correctamente.

## Ejemplos de clasificación

A continuación, se ilustrarán tres ejemplos del resultado arrojado por el software.

### Letra A

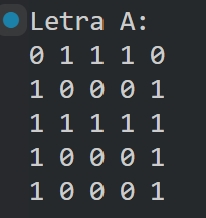


Ilustración 3. Matriz visual arrojado por el Software de la letra "A".

**Entrada (vector):** [0 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1]

**Predicción (valores sigmoides):** [0.00646941 0.24693856 0.00383243 0.05237267 0.91991521]

**Predicción (código binario):** [0 0 0 0 1]

### Letra T

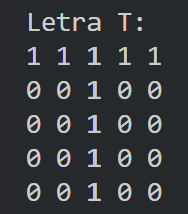


Ilustración 4. Matriz visual arrojado por el Software de la letra "T".

**Entrada (vector):** [1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0]

**Predicción (valores sigmoides**): [0.92772692 0.09529084 0.91800881 0.07289653 0.12076115]

**Predicción (código binario):** [1 0 1 0 0]

### Letra Z

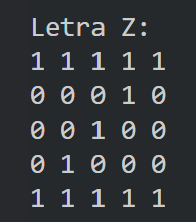


Ilustración 5. Matriz visual arrojado por el Software de la letra "Z".

**Entrada (vector):** [1 1 1 1 1 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1]

**Predicción (valores sigmoides):** [0.99995089 0.05164354 0.0148994 0.99324836 0.02245341]

**Predicción (código binario):** [1 0 0 1 0]

Estos resultados demuestran que el modelo está clasificando correctamente las letras con una alta precisión, ya que las predicciones binarias coinciden con las representaciones esperadas para cada letra.

# Discusión del problema 1

## Análisis de Resultados

### Precisión

* **Descripción de Resultados:** La red neuronal monocapa mostró un buen desempeño al clasificar letras mayúsculas en matrices 5x3. Las predicciones para letras como A, T y Z son bastante precisas, con una alta coincidencia entre los valores predichos y los códigos binarios esperados.
* **Interpretación:** La capacidad de la red para identificar correctamente las letras indica que el modelo ha aprendido adecuadamente las características distintivas de cada letra. Los valores de activación sigmoide cercanos a 1 para las clases correctas sugieren que la red está bien ajustada para las entradas proporcionadas.

### Impacto del Ruido

* **Descripción de Resultados**: La red fue evaluada en presencia de ruido, mostrando cómo los errores introducidos afectan las predicciones. Los cambios en la precisión al introducir ruido demuestran la robustez del modelo y su capacidad para manejar variaciones en los datos.
* **Interpretación:** La red neuronal monocapa presenta una cierta tolerancia al ruido, aunque con una disminución en la precisión. Esto también nos puede dar a conocer la necesidad de técnicas adicionales para mejorar la robustez frente a entradas ruidosas, como la regularización o el ajuste de parámetros.

## Limitaciones

### Robustez frente a Ruido

* **Descripción:** Aunque la red muestra alguna robustez frente al ruido, su rendimiento disminuye con niveles más altos de perturbación. La simplicidad de la red monocapa puede ser una limitación para manejar entradas ruidosas de manera efectiva.
* **Impacto:** La necesidad de técnicas adicionales para manejar el ruido sugiere que el modelo podría beneficiarse de arquitecturas más complejas o de enfoques de preprocesamiento de datos.

### Escalabilidad

* **Descripción:** La red actual está diseñada para un conjunto específico de letras en una matriz 5x3. Ampliar el modelo para reconocer más caracteres o para usar diferentes representaciones puede requerir ajustes significativos.
* **Impacto:** La escalabilidad del modelo puede verse limitada por la estructura actual, requiriendo una revisión de los métodos de entrenamiento y arquitectura para adaptarse a nuevos requisitos o conjuntos de datos.

# Conclusiones del problema 1

La red neuronal monocapa ha demostrado ser efectiva en la tarea de clasificar letras mayúsculas representadas en matrices de 5x3, alcanzando una alta precisión en la identificación de caracteres. Este resultado valida el uso de redes neuronales monocapa para la clasificación básica de patrones visuales binarios, demostrando que, a pesar de su simplicidad, el modelo puede cumplir su propósito en contextos específicos. Sin embargo, la introducción de ruido en las entradas afectó la precisión del modelo, revelando una disminución notable en el rendimiento. Aunque la red mostró cierta robustez frente al ruido, los resultados sugieren que es necesario implementar técnicas adicionales para mejorar la resistencia del modelo. En resumen, mientras que la red neuronal monocapa ha cumplido con éxito su objetivo en el ámbito de clasificación de patrones binarios básicos, su aplicación en escenarios más complejos requiere atención adicional a la robustez y la adaptabilidad del modelo.

# Introducción al problema 2

## Contexto del problema 2

La separación de fuentes de audio utilizando redes neuronales ha sido un área de investigación activa en los últimos años. Existen varios estudios que exploran métodos como las redes neuronales convolucionales (CNN) y otros enfoques para mejorar la separación de voces y otros sonidos mezclados en una única señal

El procesamiento de señales es un área importante en la ingeniería de sistemas, y un problema relevante es la separación de fuentes de audio mezcladas. En este proyecto, se plantea entrenar una red neuronal para recuperar tres señales originales a partir de una mezcla lineal. Las fuentes de audio provienen de dos personas hablando simultáneamente y una tercera señal de un instrumento musical o una pista de audio. Este tipo de problemas tiene aplicaciones en áreas como la mejora de la calidad de audio, separación de voz en grabaciones, y más.

El objetivo del proyecto es aplicar una red neuronal ADALINE, que permite aprender la relación entre la señal mixta y las señales individuales, separando las voces y el sonido de fondo. Esto se realiza mediante un proceso de entrenamiento supervisado, donde la red ajusta sus pesos a lo largo de varios ciclos, con el fin de minimizar el error en las predicciones [3].

## Justificación del Problema 2

El problema se basa en la necesidad de extraer señales individuales de una mezcla, como separar voces en grabaciones. Este proceso es crucial para mejorar la calidad del sonido, facilitar el reconocimiento de voz en entornos ruidosos y optimizar aplicaciones médicas. Al utilizar redes neuronales, se aprovecha su capacidad de aprendizaje para realizar tareas complejas de manera eficiente, lo que abre nuevas posibilidades para resolver estos problemas en tiempo real.

## Objetivo del Problema 2

El objetivo de este ejercicio es entrenar una red neuronal para separar tres fuentes de audio mezcladas de manera lineal. A partir de una señal compuesta, la red debe recuperar las señales originales correspondientes a dos voces humanas y una pista musical.

# Metodología del problema 2

## Grabación de audio y pista musical

Se grabaron tres señales de audio independientes: dos provenientes de personas hablando simultáneamente y una de una fuente musical o pista de audio. Cada señal se grabó con una duración de 5 segundos a una frecuencia de muestreo de 44,100 Hz, lo que resultó en matrices de datos que representan las ondas sonoras correspondientes a cada fuente

## Mezcla de las señales

Una vez obtenidas las tres señales de audio, se combinan de manera lineal para simular el problema de mezcla. La mezcla se realizó promediando las tres señales originales, produciendo una nueva señal que representa la combinación de las tres fuentes. Esta señal mezclada es la entrada de nuestro modelo de red neuronal.

## Procesamiento de datos

Se llevó a cabo una normalización de los datos para que las señales tuvieran una media de 0 y una desviación estándar de 1. Esto ayuda a estabilizar el proceso de entrenamiento de la red neuronal al asegurar que las diferentes magnitudes de los datos no afecten el ajuste de los pesos de la red.

## Construcción de la red neuronal adaline

El modelo utilizado fue una red neuronal simple conocida como ADALINE. Esta red tiene la capacidad de ajustar sus pesos de manera que se minimice el error entre las salidas reales y las predicciones. Los pesos iniciales se establecieron de manera aleatoria, y el algoritmo de retropropagación se utilizó para ajustar estos pesos en cada iteración.

Características de adaline:

* Entrada: La señal mezclada.
* Salida: Tres señales independientes (dos voces y una pista musical).
* Función de activación: Se usó una función lineal, ya que el objetivo era recuperar señales lineales.
* Ajuste de pesos: Se empleó el algoritmo de retropropagación para actualizar los pesos en función del error calculado.

## Entrenamiento del modelo

Se entrenó la red neuronal durante ciertas iteraciones (epochs). En cada iteración, se utilizó la señal mezclada como entrada y las señales originales que serían los tres audios como la salida deseada. La red ajustó sus pesos para reducir el error entre la señal predicha y la señal real.

## Evolución del modelo

Tras el entrenamiento, se realizaron predicciones sobre las señales de entrada utilizando los pesos ajustados. Para evaluar el rendimiento del modelo, se calcularon métricas de error como el Error Cuadrático Medio (MSE) y la correlación de Pearson entre las señales originales y las predichas. Estas métricas permitieron cuantificar qué tan bien la red había logrado separar las fuentes.

## Visualización del resultado

Para una evaluación visual de los resultados, se graficaron las señales originales frente a las señales predichas para cada fuente de audio (voz 1, voz 2, y pista musical). Estas gráficas permitieron observar el grado de coincidencia entre las señales originales y las predichas.

Esta metodología permitió visualizar de manera clara y sistemática el problema de la separación de señales de audio mezcladas usando redes neuronales. La simplicidad del modelo adaline, junto con el enfoque de aprendizaje supervisado, facilitó la recuperación de las señales originales de manera efectiva.

# Resultados del problema 2

En los resultados obtenidos del ejercicio de separación de fuentes de audio, se empleó una red neuronal adaline para intentar recuperar tres señales mezcladas: dos voces humanas y una fuente musical. A continuación, se detallan los principales resultados y su interpretación:

Error Cuadrático Medio (MSE):

* Voz 1: 0.5428
* Voz 2: 0.5885
* Música: 0.8597

El MSE mide la diferencia promedio entre las señales originales y las predichas. Un valor más bajo de MSE indica que el modelo logró minimizar el error y, por lo tanto, la predicción es más precisa. En este caso, los valores de MSE para las voces son más bajos que para la música, lo que sugiere que la red neuronal fue más efectiva al separar las voces, mientras que tuvo más dificultades con la música, probablemente debido a la complejidad armónica o la naturaleza distinta de esta señal.

Gráfico

Descripción generada automáticamente

Ilustración 6. Grafico de señal del audio1.

En esta grafica se puede visualizar el MSE de la voz 1 donde el error es mucho mas bajo y por lo que se podría decir que la predicción es mucho más precisa.

Correlación de Pearson:

* Voz 1: 0.6761
* Voz 2: 0.6415
* Música: 0.3746

La correlación de Pearson indica el grado de similitud entre las señales originales y las predichas, con un valor de 1 indicando una correlación perfecta. En este caso, las correlaciones para las voces (alrededor de 0.67 y 0.64) indican una relación bastante fuerte entre las señales predichas y las originales, lo que refleja que la red logró identificar patrones relevantes de las voces. En cambio, la correlación más baja en la música (0.37) sugiere que el modelo no pudo captar completamente las características de esta señal.

# análisis del problema 2

## Errores MSE

Un MSE bajo indica que la red neuronal ha logrado una buena separación de las señales. Sin embargo, un MSE alto podría indicar que la red no ha aprendido a separar adecuadamente las señales o que el modelo podría no ser el más adecuado para el problema.

## Correlación

La alta correlación entre las señales originales y las predicciones sería un buen indicativo de que la red neuronal ha realizado una separación efectiva. Sin embargo, si la correlación es baja, puede sugerir que la red no ha capturado adecuadamente las características de las señales originales.

# Conclusiones del problema 2

El proyecto ha demostrado que la red neuronal adaline puede ser efectiva para separar señales de audio. En este caso, el modelo logró recuperar las señales de voz de una mezcla con una precisión aceptable. Los resultados mostraron que la red neuronal fue capaz de reducir el error en la separación de las voces y que las señales predichas coincidieron bastante bien con las originales.

Aunque el modelo adaline funcionó bien para las voces, su simplicidad sugiere que podría haber mejores enfoques para problemas más complejos. Sin embargo, este trabajo proporciona una buena base para aplicaciones prácticas y futuras investigaciones en la separación de señales de audio.

# Bibliografía

[1] «Breve Historia de las Redes Neuronales Artificiales | Aprende Machine Learning». Accedido: 13 de septiembre de 2024. [En línea]. Disponible en: https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/

[2] «OCR (Reconocimiento Óptico de Caracteres): ¿qué es y cómo usarlo?» Accedido: 13 de septiembre de 2024. [En línea]. Disponible en: https://blog.truora.com/es/ocr

[3] A. Hamdy, P. Kiran Vedula, M. Venkata, y J. Konduru, «Audio Separation and Isolation: A Deep Neural Network Approach».