Diseño de un clasificador para la identificación de los sonidos emitidos por aves mediante técnicas de aprendizaje no supervisado y aprendizaje profundo

Herrera Jaramillo, Estefania Alejandra

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniera en Electrónica y Telecomunicaciones

Ing. Carrera Erazo, Enrique Vinicio

15 de agosto del 2023

Copyleaks

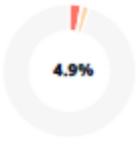
Plagiarism report

TesisEstefaniaHerrera.docx

Scan details

Scan time:	Total Pages:	Total Words:
August 15th, 2023 at 13:8 UTC	44	10880

Plagiarism Detection



Types of plagiarism		Words
Identical	2.9%	311
Minor Changes	0.3%	33
Paraphrased	1.8%	193
Omitted Words	0%	0

AI Content Detection

Text coverage
- AI text
- Human text

Plagiarism Results: (67)

Inteligencia artificial y redes neuronales artificiales... 0.7%

https://leotronics.eu/es/nuestro-blog/inteligencia-artificial-y-...

Robots de oruga TrackReitar Resumen TrackReitar TrackReitar Plataforma universal TrackReit...

CONTROL DE COMPRAS DE SUMINISTROS MEDIANTE... 0.6%

https://1library.co/document/q5mlmlgy-control-compras-su...

...

11Tesis_DOCII_Nov_2017 Aldonso_Becerra_Sanchez.... 0.5%

http://ricaxcan.uaz.edu.mx/jspui/bitstream/20.500.11845/23...

Universidad Autónoma de Zacatecas "Francisco García Salinas" Unidad Académica de Ingeniería Eléctrica Doctorado en Ciencias de la Ingenie...

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Certificación

Certifico que el trabajo de titulación: **Diseño de un clasificador para la identificación de los sonidos emitidos por aves mediante técnicas de aprendizaje no supervisado y aprendizaje profundo** fue realizado por la señorita **Herrera Jaramillo, Estefania Alejandra**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 14 de agosto del 2023

ENRIQUE VINICIO CARRERA ERAZO

Ing. Carrera Erazo, Enrique Vinicio

C.C.: 1708792104

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA
Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Responsabilidad de Autoría

Yo **Herrera Jaramillo, Estefanía Alejandra**, con cédula de ciudadanía 140077124-0, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño de un clasificador para la identificación de los sonidos emitidos por aves mediante técnicas de aprendizaje no supervisado y aprendizaje profundo** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 14 de agosto del 2023

Herrera Jaramillo Estefanía Alejandra

C.C.: 1400771240

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingenieria en Electrónica y Telecomunicaciones

Autorización de Publicación

Yo **Herrera Jaramillo, Estefania Alejandra**, con cédula de ciudadanía 140077124-0, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño de un clasificador para la identificación de los sonidos emitidos por aves mediante técnicas de aprendizaje no supervisado y aprendizaje profundo** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 14 de agosto del 2023

Herrera Jaramillo Estefania Alejandra

C.C.: 1400771240

Dedicatoria

Este gran logro lo dedico a mis padres y familia en general que siempre estuvo apoyándome y aconsejándome en los momentos duros de esta etapa; a Dios por llenarme de salud, de fuerza y sabiduría que me ayudó a enfrentar las grandes batallas y llegar a cumplir mis metas. No importa el tiempo que te demores en llegar a la cima, solo se trata de seguir avanzando y subir.

Estefania Alejandra Herrera Jaramillo

Agradecimiento

Doy gracias a Dios por llenarme de salud y sabiduría y permitirme cada día seguir con fuerza y constancia cada etapa de mi vida, por guiar mi camino y llenarme de valor para seguir adelante. Agradezco de todo corazón a mis padres y hermanos que son el motor de mi vida y son testigos de todo el esfuerzo por alcanzar mis metas, todo su amor y sus enseñanzas me ayudaron a formarme personal y profesionalmente, ser fuerte y superarme cada día. A toda mi familia por sus consejos, su apoyo brindado y por no dejarme vencer en cada paso durante esta etapa universitaria.

Agradezco a mi novio que estuvo a mi lado todo este camino lleno de obstáculos, y me ayudó a superarlos. Gracias por creer en mí cuando a veces ni yo lo hacía, su confianza y su apoyo incondicionales mi luz en esos momentos de oscuridad.

Quiero agradecer especialmente al Ing. Carrera Vinicio por toda su paciencia y su ayuda durante el desarrollo del proyecto y el conocimiento transmitido para culminar con éxito este trabajo de investigación.

Estefania Alejandra Herrera Jaramillo

Índice de Contenidos

Índice de Tablas

Índice de Figuras

Resumen

Durante estos últimos años el hombre ha ido evolucionando y expandiendo más su territorio, ocupando gran parte de los espacios naturales donde existen variedades de especies silvestres, destruyendo así su habitad. Estas especies han disminuido su número al ser utilizadas por el hombre para su subsistencia, llevando algunas a su extinción. Para evitar este suceso se han creado centros de apoyo para proteger la fauna silvestre del mundo, ya que cada especie contribuye a mantener el equilibrio en la biodiversidad del planeta; entre estas especies se encuentran las aves, por lo que es importante mantenerlas en continuo monitoreo. Una de las formas para preservar las especies de aves en peligro de extinción es clasificarlas para identificar las necesidades de cada una. El proceso de observación de las aves puede tomar de días a semanas o hasta meses para identificar los distintos tipos que puedan encontrarse en una determinada región. Con este proyecto se busca facilitar la clasificación de las aves, automatizando el proceso por medio de inteligencia artificial aplicando modelos de aprendizaje no supervisado y aprendizaje profundo, desarrollando así un clasificador que permite identificar las diferentes especies de aves por medio de su canto. Este clasificador presentó resultados de su rendimiento mediante la exactitud y el error cuadrático medio, para la extracción de características por medio de autoencoders más el clasificador softmax llegó a obtener un porcentaje del 99.2% y un error del 0.0025954, y para el modelo utilizando redes recurrentes LSTM se llegó a obtener un porcentaje del 85.67% utilizando una mayor cantidad de audios.

Palabras clave: inteligencia artificial, redes neuronales, aprendizaje de máquina, aprendizaje profundo.

Abstract

During these past years, humans have been evolving and expanding their territory, occupying a huge portion of the natural spaces where various wild species exist, thus destroying their habitat. These species have seen a decrease in their numbers as they have been used by humans for sustenance, leading some to extinction. To prevent this occurrence, support centers have been established to protect the world's wildlife, as each species contributes to maintaining the balance in the planet's biodiversity; among these species are birds, making it important to keep them under continuous monitoring. One of the ways to preserve endangered bird species is by classifying them to identify the specific needs of each one. The process of bird observation can take days, weeks, or even months to identify the diverse types that can be found in a particular region. This project aims to streamline the classification of birds by automating the process through artificial intelligence using unsupervised learning models and deep learning, thus developing a classifier that can identify different bird species based on their calls. This classifier demonstrated its performance with metrics such as accuracy and mean squared error. For feature extraction using autoencoders plus the softmax classifier, it achieved a percentage of 99.2% and an error of 0.0025954. For the model utilizing recurrent LSTM networks, a percentage of 85.67% was achieved, especially with a larger quantity of audio samples.

Keywords: artificial intelligence, neural networks, machine learning, deep learning.

Capítulo I

Introducción

Antecedentes

Existen diferentes especies de fauna silvestre a nivel mundial, las cuales se encuentran amenazadas por el aumento excesivo de la población humana presentando varios factores que los han llevado a estar en una situación de extinción. Todas las especies de animales forman una parte importante de la diversidad biológica del planeta que nutren y mantienen en equilibrio el ecosistema (Capacete, 2019); por ello, las aves también son especies importantes que necesitan ser protegidas ya que a diario invaden y destruyen su habitad natural reduciendo su población.

Algunas especies de aves silvestres se encuentran en peligro de extinción, por esta razón existen organizaciones que se encargan de recopilar información y realizar un monitoreo diario para la conservación de esta especie. Durante estos años se han registrado un total de 10622 especies de aves a nivel mundial que se pueden observar en la plataforma web eBird; los países que presentan una mayor diversidad de aves son Colombia, Perú, Ecuador, Brasil e Indonesia que sobrepasan las 1500 especies de aves (eBird, 2022).

Ecuador se encuentra entre los cinco países con mayor número de aves presentando un total de 1660 especies (eBird, 2022), considerando las provincias que más observaciones se ha tenido de las diferentes especies de aves se encuentra liderando Napo, Sucumbíos y Morona Santiago (Cajas & Villalva, 2019). Cada institución dedicada a realizar el monitoreo de las aves emplea distintos métodos que les permiten obtener resultados óptimos, entre los centros de monitoreo que existen en el Ecuador se encuentra el Centro de Investigación, Posgrado y Conservación Amazónica (CIPCA) que realiza una técnica para el registro de las aves de forma visual y por medio de las grabaciones de los cantos de cada una de las especies (Shiguango & Bañol, 2020).

Al pasar los años se han implementado nuevas técnicas para el monitoreo de las diferentes especies de aves, en general, cada espécimen emite sonidos o llamados representativos de la especie para comunicarse entre sí, por lo que los nuevos métodos de monitoreo integran patrones de sonidos para identificar con mayor efectividad y emplear un método menos invasivo para las especies que se necesiten realizar el estudio. Por otro lado, el reconocimiento de audio presenta cada vez una mayor aplicación de los sistemas de aprendizaje automático, actualmente se utilizan técnicas como el aprendizaje no supervisado y el aprendizaje profundo que permiten realizar la identificación del sonido mediante una extracción de características, en la cual el sistema realiza de forma automática todo el procesamiento del sonido. Los modelos de aprendizaje no supervisado y aprendizaje profundo tienen una mayor accesibilidad en la identificación de los sonidos de las diferentes especies de aves por la complejidad que se presenta para estudiarlas visualmente.

Objetivos

General

Diseñar un sistema que permita la clasificación de audios mediante técnicas de aprendizaje no supervisado y aprendizaje profundo para la identificación de los sonidos emitidos por las aves.

Específicos

- Investigar las técnicas de aprendizaje automático y aprendizaje profundo para tareas de procesamiento y clasificación de audio.
- Determinar las mejores bases de datos existentes con sonidos de las diferentes especies de aves.
- Implementar el modelo de autoencoder para automatizar el proceso de extracción de características y posterior clasificación.

- Comparar el modelo anterior con esquemas basados en aprendizaje profundo para establecer las tendencias en cuanto a exactitud de clasificación.
- Evaluar el sistema de reconocimiento de los sonidos de las aves mediante una interfaz iterativa realizada en el software Matlab.

Metodología

El desarrollo del sistema para el procesamiento de audio empieza mediante una investigación del estado del arte de las técnicas de aprendizaje no supervisado como son los autoencoders y el aprendizaje profundo. Adicionalmente, se realizará una recopilación de información sobre las mejores bases de datos en la cual se encuentre los sonidos emitidos de diferentes especies de aves.

Se procede entonces a realizar la implementación del sistema para la identificación de los sonidos emitidos por las aves con la ayuda del software Matlab, en la cual se realiza el procesamiento de la señal de audio ingresada de la base de datos que proporciona la plataforma seleccionada; mediante el modelo de aprendizaje no supervisado se automatiza el proceso de extracción de las características utilizando autoencoders y se añade una capa softmax para la clasificación (Angulo, 2020). Posteriormente, mediante el desarrollo por modelos de aprendizaje profundo, las señales de audio obtenidas de la base de datos ingresarán en el algoritmo de redes neuronales recurrentes (RNN), los cuales cuentan con capas que realizan la función de extracción de características y clasificación de los datos (González J. , 2021). Al extraer los resultados finales, se aplicarán métricas de evaluación para medir el rendimiento de los modelos utilizados, las cuales pueden ser: matriz de confusión, exactitud, sensibilidad, precisión o puntuación F1 (Altamirano, 2021).

Finalmente se implementará el sistema de procesamiento y reconocimiento de audio mediante el diseño de una interfaz iterativa en Matlab que realiza la identificación de los sonidos emitido por las aves utilizando técnicas de aprendizaje no supervisado y aprendizaje profundo para su comparación.

Organización

El documento relata todo el proceso a desarrollar para realizar el trabajo de investigación propuesto, en el cual toda la información se encuentra organizado en cinco capítulos que describe cada una de las etapas que se siguió para llegar a la solución y poder tener una lectura más comprensiva.

En este primer capítulo se especifica una breve introducción que acoge los antecedentes del proyecto, la justificación, la importancia, la metodología y los objetivos planteados para el trabajo de investigación.

El segundo capítulo detalla el marco teórico, donde se presentan los conceptos básicos de las técnicas principales que se utilizan para el desarrollo del trabajo de investigación, profundizando con mayor detalle estos temas para una mejor comprensión de las bases en las que se sustenta este proyecto.

En el tercer capítulo se describe la metodología utilizada para el sustento y desarrollo del proyecto, indicando paso a paso el proceso que se debe seguir para cumplir todos los objetivos planteados.

El cuarto capítulo presenta los resultados obtenidos de las múltiples evaluaciones realizadas con los modelos y la comparativa para determinar el modelo más eficiente. Además, se proporcionan ideas (basadas en experiencia) para un desarrollo más efectivo de este proyecto o de proyectos que se deriven de este.

En el quinto capítulo se detallan las conclusiones que se obtuvieron en este trabajo de investigación y propuestas de trabajos futuros para mejoras que se pueda realizar a este sistema de clasificación.

Capítulo II

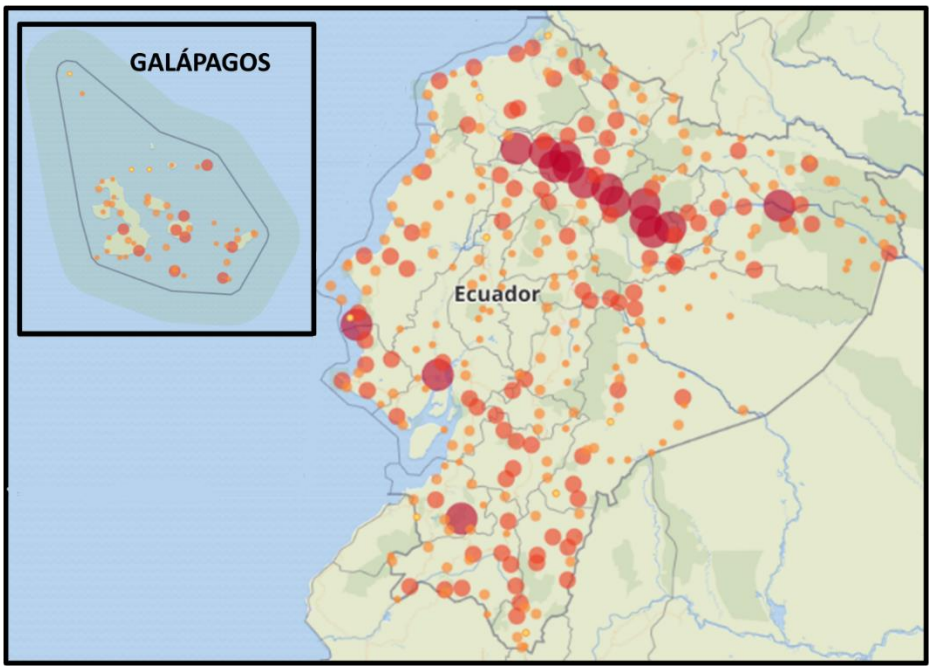
Marco Teórico

Aves en peligro de extinción del Ecuador

El Ecuador es un país con una alta variedad de especies de aves silvestres como se pude visualizar en la **¡Error! No se encuentra el origen de la referencia.**, donde se encuentran alrededor de 1660 especies de aves distribuidas en todas las regiones, y se mantienen en continuas observaciones por el alto índice de aves que se encuentran en peligro de extinción. Cada una de las especies se las puede reconocer auditivamente mediante los cantos y llamados, cuyos sonidos se diferencian por la información a transmitir, estas formas de relacionarse permiten identificarlas con mayor rapidez ya que es compleja su visibilidad por el habitad en el que viven (González, Padrón, Barbero, Custodio, & Merchán, 2019).

Figura 1

Aves Silvestres en todo el Ecuador



Muchas de estas especies de aves están vulnerables ante el crecimiento excesivo de la población humana, que las caza por su colorido plumaje entre otras características. Esta actividad ha llevado a la extinción de 6 especies en el país, y ha puesto a un gran número en diferentes categorías por su severidad, estas son: 10 en peligro crítico (CR), 16 en peligro (EN), 63 como vulnerable (VU), 85 en margen de riesgo (MR) y 25 en datos insuficientes (DD) (Chanco & Narváez, 2018). Para reducir estos valores y conservar la fauna silvestre del país se ha implementado en ciertos lugares sistemas de monitoreo diario utilizando los sonidos emitidos por las diferentes especies de aves que pueden ser cantos o llamados y permiten mantenerlas en constante vigilancia, dependiendo el comportamiento que presenten las aves en el momento emiten varios sonidos característicos cada uno con un mensaje diferente.

Las aves para relacionarse entre sí utilizan sus cantos, los cuales presentan una mayor complejidad para el monitoreo ya que su estructura cuenta con notas o una serie de segmentos diferentes en todo el sonido; en cambio los sonidos que realizan para comunicar una información lo hacen mediante los llamados que están formados por repeticiones que ayudan a identificar con más detalle el tipo de ave que lo está emitiendo (González, Padrón, Barbero, Custodio, & Merchán, 2019).

Estas muestras de audio se las puede encontrar en el sitio web Aves del Ecuador, el cual contiene bases de datos de los cantos y llamados de las diferentes especies de aves silvestres del Ecuador (bioWeb Ecuador, 2018).

Pre-procesamiento de Audio

Normalización

Las señales que ingresan deben presentar datos que pueden ser reconocidos por el sistema, estas señales presentan amplitudes de diferentes escalas, y algunas pueden ser negativas. Por ello se realiza un ajuste de los valores a un mismo rango, ya que el algoritmo utilizado de aprendizaje no

supervisado no acepta ese tipo de datos. Para alinear las medidas de tendencia de cada una de las señales que van ingresando para su respectivo procesamiento, se utiliza la normalización basada en la unidad. Esto quiere decir que todos los datos se escalan a un rango entre 0 y 1 de amplitud, este método evita distorsiones en los resultados, excluyendo valores de la base de datos erróneos. La normalización se realiza mediante la ecuación (1), en la cual, se cambia los valores máximos y mínimos de cada señal.

$$\widehat{r_l} = \frac{S - S_{min}}{S_{max} - S_{min}} \qquad (1)$$

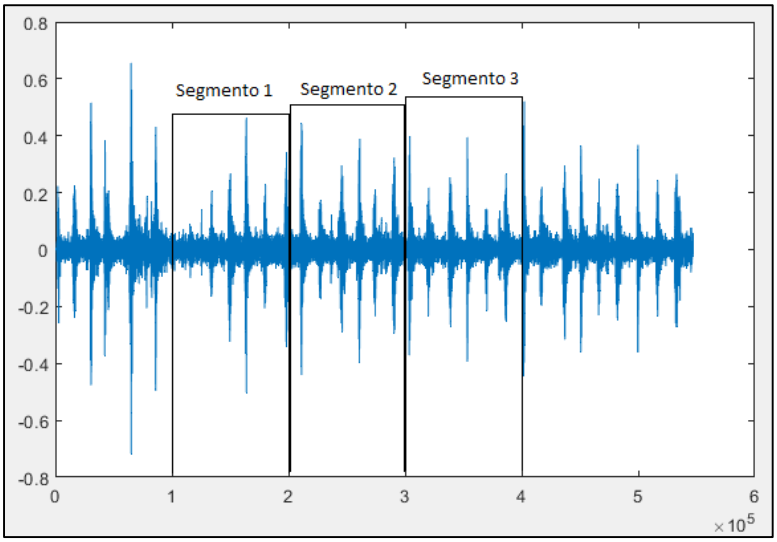
En donde $\widehat{r_l}$ es la señal normalizada, S la señal original, S_{min} los valores mínimos y S_{max} los valores máximos que presenta la señal.

Segmentación

Las señales de audio presentan longitudes diferentes que retrasan el procesamiento por la gran cantidad de datos que ingresan, por ello, se realiza un proceso de segmentación que trata de dividir las señales en fragmentos más pequeños de igual longitud para la mejora computacional y la facilidad de aprovechar los segmentos que contengan información válida para el proceso. Esta segmentación representa el número de muestras en base a la frecuencia de muestreo del audio de entrada (Martínez S. , 2020). La longitud está definida por el tiempo de duración del segmento, y este tiempo depende de la frecuencia de muestreo. En la **¡Error! No se encuentra el origen de la referencia.** se muestra una representación de la segmentación de la señal de audio.

Figura 2

Segmentación de la señal de audio



Se han considerado dos formas de realizar la segmentación de una señal de audio, las cuales son:

Con solapamiento: cada fragmento se sobrepone uno del otro en un pequeño porcentaje, repitiéndose ciertos datos en los segmentos posteriores.

Sin solapamiento: los segmentos serán continuos, es decir, no existirá separación entre dos segmentos adyacentes.

Para facilitar la selección de los segmentos con más información útil, al realizar la segmentación, se considera escoger tiempos bastante pequeños. El modelo con solapamiento fue utilizado para el desarrollo de este trabajo de investigación, porque disminuye la pérdida de información duran el procesamiento de la señal de audio.

Ventana de Hamming

Las señales segmentadas pueden tener transiciones bruscas que generan distorsiones al obtener la FFT, por tal motivo es necesario realizar el proceso de ventanado para mejorar la dispersión y

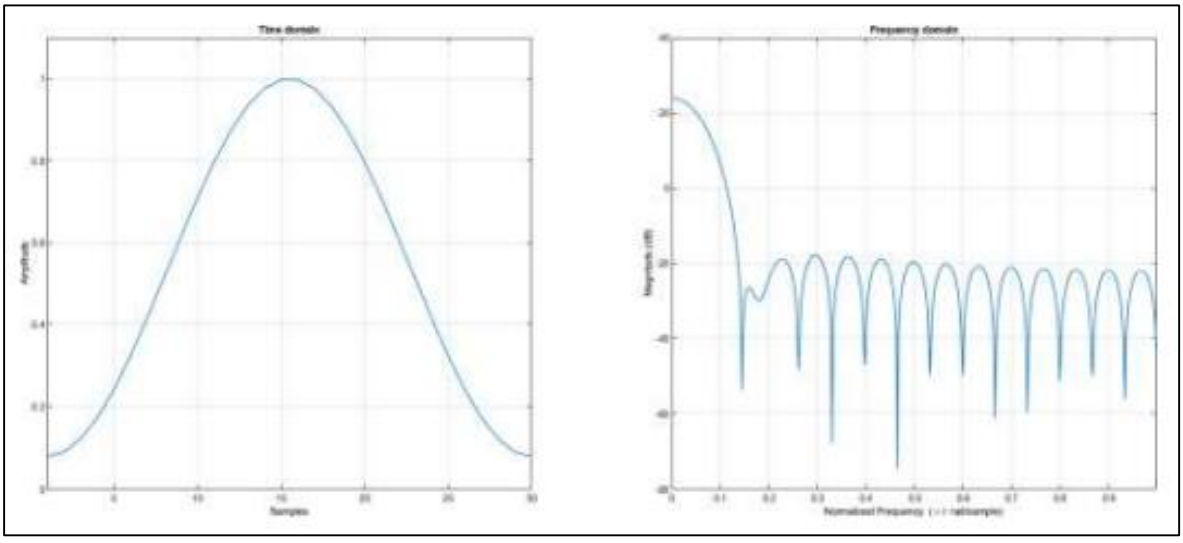
resolución espectral de las señales (Martínez S. , 2020). El resultado de este proceso es una señal sin discontinuidades mediante el producto con la función matemática que representa la ventana suavizante. La función de ventana tiene efecto en el cambio del espectro de frecuencia de la señal y depende de sus características se puede escoger entre varios tipos de ventanas.

En este proyecto se utiliza la ventana de Hamming, propuesta por Richard W. Hamming, para obtener una buena resolución de la frecuencia y suavizar los datos antes de realizar la FFT, así poder adecuar de una forma aceptable las mediciones ruidosas (Rodriguez, 2020).

La representación gráfica de la ventana de Hamming se observa en la **¡Error! No se encuentra el origen de la referencia.**, en la cual se muestra a la izquierda la respuesta en el dominio del tiempo y a su derecha en el dominio de la frecuencia.

Figura 3

Ventana Hamming en el dominio del tiempo y de la frecuencia, respectivamente



Nota. Recuperado de: (Rodriguez, 2020)

La ecuación de la ventana de Hamming en el dominio del tiempo es:

$$W_H(n) = 0.54 - 0.46\cos\frac{2\pi n}{M-1} \quad (2)$$

La ecuación de la ventana de Hamming en el dominio de la frecuencia es:

$$W_H(e^{jw}) = \sum_{n=0}^{N-1} W_H(n)e^{-jwn} \quad (3)$$

Coeficientes LPCs

Los coeficientes de predicción lineal (LPC) se utiliza con mayor frecuencia en el tratamiento de las señales de audio, reduciendo al mínimo el error de predicción. Se utiliza en la aplicación de filtros y codificación de voz (MathWorks, 2023). Los LPCs se los consideran filtros FIR que modela el tracto vocal y su ecuación es:

$$s(n) = \sum_{k=1}^{p} s(n-k)a_k + Gu(n) \quad (4)$$

Donde $s(n)$ es la señal original, a_k es el conjunto de coeficientes de predicción y p es el orden del filtro, cuyo valor debe ser menor e igual a la longitud del conjunto de entrada. Para obtener el error de predicción $e(n)$ se aplica la ecuación (5):

$$e(n) = s(n) - s'(n) = Gu(n) \quad (5)$$

Donde $s'(n)$ es la señal predicha y $Gu(n)$ es la ganancia (García, Gallego, Domínguez, Correa, & Rodríguez, 2017).

Potencia Promedio

En una señal de audio existen segmentos que no presentan una fuerte intensidad en el sonido, por lo tanto, se busca identificar aquellos donde la fuerza del sonido aumenta ya que cuentan con mayor información que puede analizarse. Esta intensidad se la puede medir como el flujo medio de energía (potencia) y su ecuación es:

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt \qquad (6)$$

Donde $x(t)$ es cada uno de los segmentos de la señal de audio (Rosario, 2017).

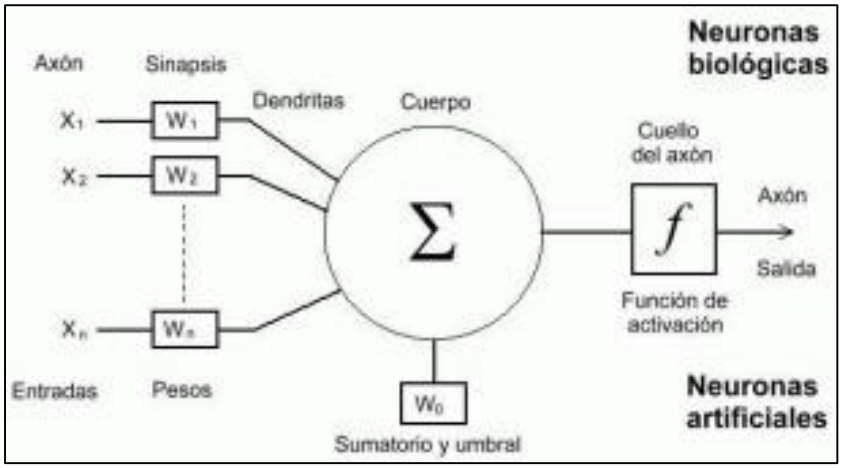
Redes Neuronales

El aprendizaje automático es una de las ramas de la Inteligencia Artificial en las que se basan las redes neuronales, las cuales están inspiradas en el funcionamiento del cerebro humano. Están estructuradas de componentes básicos o nodos (neuronas), los cuales se encuentran interconectados de tal manera que permiten adaptarse a cualquier cambio, para procesar información y producir una salida óptima, acorde a las necesidades del ámbito para el cual fueron diseñadas.

La red neuronal es un tema que se ha tratado desde el año 1943 basando su diseño y funcionamiento de las redes neuronales biológicas relacionando cada una de sus partes como lo muestra la **¡Error! No se encuentra el origen de la referencia.**. Se han presentado casos como el de W. McCulloch y W. Pitts que propusieron el funcionamiento del modelo basado de una neurona natural, su desarrollo se detuvo en vista de la necesidad de una alta potencia computacional que no era suficiente en esos tiempos en que la tecnología empezaba con su evolución, siendo una de las razones que imposibilitaron el entrenamiento profundo de la red.

Figura 4

Red Neuronal Artificial/Biológica

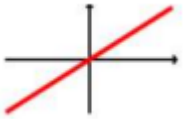
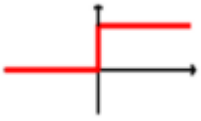
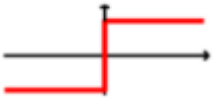


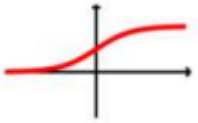
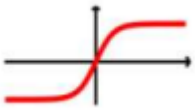
Nota. Recuperado de: (Oña, 2020)

Las expresiones matemáticas de las funciones de activación utilizadas en el área de las redes neuronales artificiales se muestran en la Tabla 1.

Tabla 1

Funciones de Activación de las Redes Neuronales Artificiales

Función de Activación	*Ecuación*	*Ejemplo*	*Gráfico*
Linear	$\emptyset(z) = z$	Adaline, linear regression	
Unit Step (Heaviside Function)	$\emptyset(z) = \begin{cases} 0 & z < 0 \\ 0.5 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Sign (signum)	$\emptyset(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	

Función de Activación	*Ecuación*	*Ejemplo*	*Gráfico*
Piece-wise Linear	$\emptyset(z) = \begin{cases} 0 & z \leq -1/2 \\ z + 1/2 & -1/2 \leq z \leq 1/2 \\ 1 & z \geq 1/2 \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\emptyset(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multilayer NN	
Hyperbolic Tangent (tanh)	$\emptyset(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multilayer NN, RNNs	
ReLU	$\emptyset(z) = \begin{cases} 0 & z < 0 \\ z & z > 0 \end{cases}$	Multilayer NN, CNNs	

Nota. Recuperado de: (Oña, 2020)

Aprendizaje No Supervisado

Es un tipo de algoritmos de Machine Learning (Aprendizaje de Máquinas) que puede clasificar una serie de datos sin la necesidad de presenciar un supervisor que ayude en el aprendizaje del sistema, los datos no presentan un etiquetamiento previo por lo cual el sistema se encarga de reconocer patrones específicos que caractericen a cada uno de los datos de entrada y que se pueda etiquetar sin problema nuevos datos que se siga recibiendo (González J. , 2019).

El uso de estas técnicas es muy favorable para diversas aplicaciones, ya que se puede presentar un conjunto amplio de datos en la cual varía sus características en cualquier instante de tiempo siendo esencial este procesamiento al no presentar una supervisión en cada momento, obteniendo resultados óptimos y no presenta un elevado coste (Martínez R. , 2017).

Las técnicas de aprendizaje más comunes de este modelo de machine learning se puede observar en la Tabla 2.

Tabla 2

Algoritmos de Aprendizaje no Supervisado

Aprendizaje no Supervisado	*Concepto*
K-Means, K-Medoids	Esta técnica agrupa cada una de las observaciones a un clúster diferente minimizando la distancia entre los datos.
Difuso de C-Means	Este algoritmo presenta un elemento que corresponde de manera difusa a cada grupo generado.
Jerárquico	Esta técnica hace referencia a realizar el agrupamiento dependiendo su nivel en orden en forma de un árbol de clústeres.
Mezclas Gaussianas	Se ajusta a la probabilidad de que ciertos datos que fueron asignados a un clúster realmente pertenezcan a ese grupo.
Redes Neuronales	Son redes interconectadas entre sí para transmitir señales, similar a una neurona biológica del sistema nervioso.
Modelos Ocultos de Markov	Es un modelo estocástico que trata de recuperar una secuencia de estados partiendo de datos observables que generaron este conjunto.

Nota. Recuperado de: (MathWorks, 2022)

El modelo de aprendizaje no supervisado que es capaz de aprender mediante repeticiones y que utiliza el algoritmo de redes neuronales es el autoencoder, en el cual está enfocado este trabajo de investigación para realizar la extracción de características, a este proceso se integra una capa de clasificación simple, softmax, que utiliza una metodología de aprendizaje supervisado.

Autoencoder

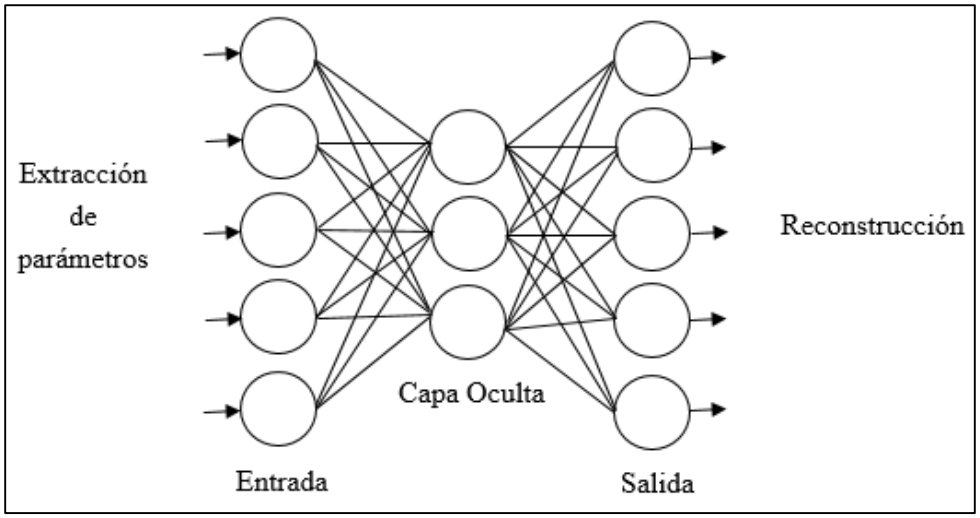
Los autoencoders son redes neuronales cuya estructura permite replicar los datos de entrada en la salida de la red mediante una organización autónoma e inteligente, este algoritmo trata de reducir significativamente la distorsión que se pueda presentar en los datos de salida del sistema, por tal motivo

se ha considerado para tratar señales ruidosas, como puede afirmar Marvin Coto Jiménez en su trabajo acerca de la síntesis de voz (Coto, 2017), esta característica ayuda a eliminar el ruido introducido en las señales de audio y tomar únicamente los datos que contengan información válida para el procesamiento. La arquitectura del autoencoder está conformada por varias neuronas que se distribuyen en conjuntos de acuerdo a una función que cumplen en común dependiendo del tipo de información que ingrese a la neurona, cada conjunto de neuronas se las representa como capas conectadas mediante enlaces por donde se moviliza la información (Caicedo & López, 2017), el autoencoder presenta tres tipos de capas como se visualiza en la Figura 5, y estas son:

- Capa de entrada: esta capa se encuentra formada por un conjunto de neuronas que únicamente cumple la función de recepción de las señales de entrada, transmitiendo los datos a la capa posterior para procesarlos (Zumba & Zumba, 2022).
- Capa oculta: las neuronas de esta capa realizan un proceso de codificación que comprime los datos, por tal razón, su número es inferior al número de neuronas de la capa de entrada, permitiendo así capturar los conjuntos de características que presenten mayor importancia y sean útiles para el aprendizaje de la red (Zumba & Zumba, 2022).
- Capa de salida: las neuronas de esta capa cumplen la función de decodificación de los datos comprimidos por la capa anterior, encargándose de reconstruir las señales de entrada en la salida obteniendo la respuesta del sistema, por ello, al replicar la señal su dimensión será la misma que la capa de entrada (Zumba & Zumba, 2022).

Figura 5

Estructura de un Autoencoder



Nota. Recuperado de: (Coto, 2017)

Los autoencoders presentan varios tipos de arquitecturas que cumplen funciones de reducción de dimensionalidad y extracción de características, cuyos métodos varían dependiendo de la rama de la inteligencia artificial en la que fue introducido su modelo, la arquitectura que se muestra en la Figura 5 representa un autoencoder elemental, cuya expresión matemática se describe en la ecuación (7).

$$x = g(f(y)) \qquad (7)$$

En donde $f(y)$ contiene la codificación de la señal de entrada, la cual realiza el proceso de extracción de características en la capa oculta, esta función vectorial se escribe en la ecuación (8) (Salgado, 2022).

$$f(y) = \sigma_f(W_1 y + b) \qquad (8)$$

Donde y es un vector de los datos que ingresan a la red neuronal, σ_f es una función que se encuentra en la capa oculta como activación, W_1 es la matriz de pesos que se encuentra entre las capas de entrada y oculta, y b es una función vectorial que representa las bias de entrada (Salgado, 2022).

La salida de la arquitectura del autoencoder muestra la reconstrucción de la señal de entrada, es decir, la decodificación de los datos internos que se obtiene de la capa oculta cuya expresión matemática se describe en la ecuación (9).

$$x = g\big(f(y)\big) = \sigma_g(W_2 f(y) + c) \qquad (9)$$

Donde σ_g es una función que se encuentra en la capa de salida como activación, W_2 es la matriz de pesos que se encuentra entre las capas oculta y de salida, y c es una función vectorial que representa las vías de salida (Salgado, 2022).

Aprendizaje Profundo

El aprendizaje profundo es una rama del aprendizaje de máquina que cuenta con varios modelos basados en redes neuronales. Desde el año 2006 ya se empezó a revelar estudios más concretos en la utilización de aprendizaje profundo, se realizaron varias publicaciones acerca de estos modelos, entre las cuales observaron un método descrito por G. Hinton, quién ganó el premio Turing junto a otros informáticos en el año 2018 por el trabajo dedicado en Aprendizaje profundo, el cual demuestra efectividad en el entrenamiento a una red neuronal con aprendizaje profundo (González J. , 2017).

Este avance dio apertura al desarrollo de nuevas técnicas mejoradas, que presentan arquitecturas capaces de entrenar modelos con un alto número de datos de entrada y que puedan tener comportamientos complejos, ante estos casos la implementación de la red neuronal profunda mantiene un bajo coste computacional (Atienza, 2019). Hoy en día los algoritmos de aprendizaje profundo han

presentado buenos resultados que van cada vez introduciéndolos en varias aplicaciones para optimizar su funcionamiento.

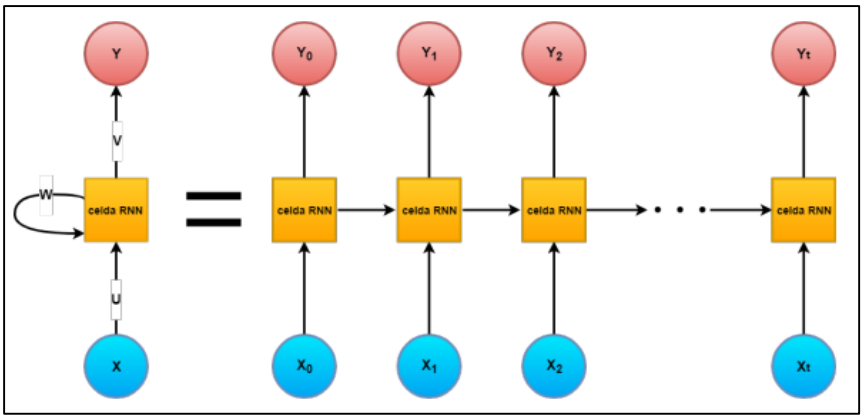
El aprendizaje profundo presenta otros algoritmos para el entrenamiento de una red neuronal que son: redes neuronales convolucionales (CNN), redes neuronales recurrentes (RNN) y redes neuronales dinámicas (DNN) (González J. , 2017). Para este proyecto de investigación se enfocó el estudio en las redes neuronales recurrentes.

Redes Neuronales Recurrentes (RNN)

Son redes retroalimentadas cuyos datos para su procesamiento son secuenciales, estos bucles son capaces de almacenar la información que se obtiene en cada capa en una memoria, alimentando el paso actual con los datos adquiridos de los procesos anteriores, su estructura se lo puede observar en la Figura 6.

Figura 6

Estructura básica de una Red Neuronal Recurrente (RNN) a lo largo del tiempo



Nota. Recuperado de: (González I. , 2019)

La expresión matemática de la salida en cada uno de los instantes de tiempo se describe en la ecuación (10) y del estado temporal en la ecuación (11).

$$Y_t = f_y(V_{t_{state}}) \quad (10)$$

$$t_{state} = f_{t_{state}}(U_{X_t} + W_{t_{state}-1}) \quad (11)$$

Donde U, V, W son matrices que se encuentran conectando la entrada, el estado temporal y la salida de la red como se observa en la Figura 6; f_y y $f_{t_{state}}$ son funciones de activación de la salida y el estado temporal (González I. , 2019).

Las redes neuronales recurrentes se las puede utilizar para el reconocimiento de la voz y el análisis de la escritura, pero su entrenamiento puede resultar complejo si se utiliza una estructura básica ya que presenta un problema de desvanecimiento del gradiente (Rivas, 2020). Esta red al utilizar la información de partes anteriores y procesar un mayor valor de elementos puede olvidar cierta información adquirida en los procesos de las primeras capas.

Se ha implementado una celda de memoria que puede extraer características utilizando datos secuenciales que presenten mayores longitudes, esta celda solucionó el problema de memoria de corto plazo, optando el nombre de Memorias de Corto y Largo Plazo (LSTM).

a) Redes de Memoria de Corto y Largo Plazo (LSTM)

Las redes LSTM ayudan a reducir el problema del desvanecimiento de la gradiente que presenta las redes neuronales recurrentes básicas, logrando recordar información de datos anteriores durante un tiempo bastante largo; en este tipo de redes las neuronas que forman la capa oculta son reemplazadas por memorias, esta celda de memoria cuenta con tres puertas o gates para procesar la información y la celda aprenda a identificar los datos con información relevante y desechar los que no lo tienen (Martínez R. , 2019), estas puertas que controlan el estado de la celda y la protegen son (Oña, 2020):

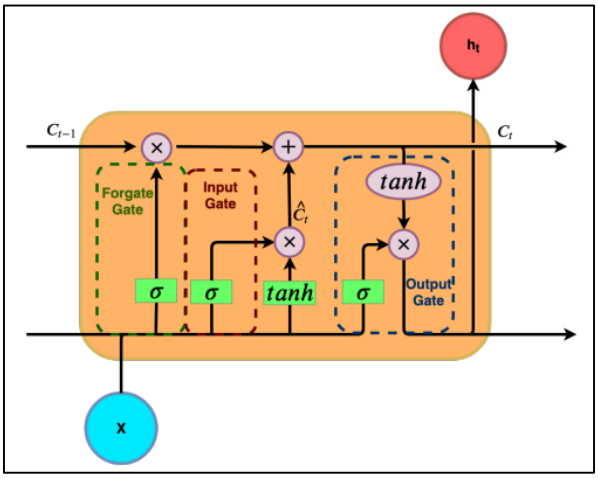
- Input gate: la puerta de entrada se encarga de decidir los datos que ingresan a la celda.

- Forget gate: la puerta de olvido se encarga de validar si los datos no cuentan con información importante para el aprendizaje y proceder a quitarlos de la celda.
- Output gate: en la puerta de salida se obtiene un vector de valores filtrados con datos relevantes para el procesamiento.

Estos gates para dejar pasar la información necesaria cuentan con dos funciones de activación llamadas Sigmoidal y Tanh (tangente hiperbólica); el resultado de las funciones de activación en las capas para permitir o desechar los datos tendrá un valor de 0 y 1 (Rivas, 2020), si la información del vector de datos que ingresa es válida se marca un valor de 1 y la información pasa, si se detecta que ciertos datos del vector no contienen la información necesaria para el aprendizaje el valor resultante es de 0 desechando esos datos evitando que pase a la siguiente capa. La arquitectura de la red LSTM en la cual se observa las tres puertas y las funciones de activación para su funcionalidad se muestra en la Figura 7.

Figura 7

Arquitectura de la red LSTM



Nota. Recuperado de: (González I. , 2019)

Métricas de Rendimiento

En este trabajo de investigación al utilizar algoritmos de Machine y Aprendizaje profundo para la clasificación de un volumen alto de datos se necesita calcular el rendimiento del sistema, así determinar el método que presenta mayor efectividad en el aprendizaje de la red. En este caso se utiliza la matriz de confusión, la cual representa las predicciones resultantes de la clasificación y se puede evaluar la precisión de los modelos implementados. En la Tabla 3 se observa la matriz de confusión, cuyas filas representa la clase predicha y las columnas la clase real (Borja, Monleón, & Rodellar, 2020).

Tabla 3

Matriz de Confusión

		Predicción	
		Positivo	*Negativo*
Clase Real	*Positivo*	Verdadero Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadero Negativo (VN)

Nota. Recuperado de: (Altamirano, 2021)

En la matriz de confusión los valores que son verdaderos positivo y negativo representan los datos que fueron correctamente clasificados (Borja, Monleón, & Rodellar, 2020), mientras que el falso positivo muestra una predicción verdadera pero el valor real es falso, y el falso negativo el valor de la predicción es falso y el valor real es verdadero.

Mediante estos valores que proporciona la Tabla 3 se puede obtener las siguientes métricas (Altamirano, 2021):

- **Exactitud**

Indica la clasificación de predicciones correctas sobre el total de predicciones. Su fórmula matemática es la siguiente:

$$Accuracy(\%) = \frac{VP + VN}{VP + VN + FP + FN} \quad (12)$$

- **Sensibilidad**

Indica el valor de las predicciones positivas que se encuentran bien clasificadas. Su fórmula matemática es la siguiente:

$$Recall(\%) = \frac{VP}{VP + FN} \quad (13)$$

- **Especificidad**

Indica el valor de las predicciones negativas que se encuentran bien clasificadas. Su fórmula matemática es la siguiente:

$$Specificity(\%) = \frac{VN}{VN + FP} \quad (14)$$

- **Precisión**

Indica la relación de los valores que son verdaderos positivos sobre el total de predicciones positivas. Su fórmula matemática es la siguiente:

$$Precision(\%) = \frac{VP}{VP + FP} \quad (15)$$

Capítulo III

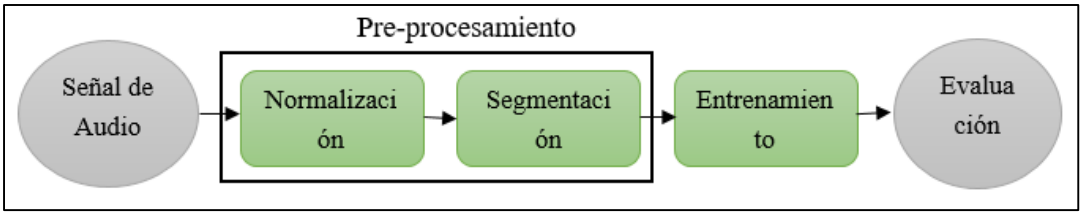
Materiales y Métodos

Este sistema fue programado en el software Matemático MATLAB versión R2020a, el algoritmo permite manejar la base de datos donde se encuentran registrados los sonidos de las diferentes aves del Ecuador para su procesamiento mediante la extracción de características detectadas en grabaciones de audio. La resolución del sistema de identificación se la realizó por medio de dos procedimientos, el primero utilizando autoencoders y el segundo utilizando redes neuronales recurrentes LSTM.

Cada sistema cuenta con un conjunto de etapas que realiza un procesamiento de la señal de audio ingresada, y así obtener datos que pueda interpretar el método utilizado de Machine Learning y Aprendizaje Profundo para su aprendizaje e identificación de los sonidos de las diferentes aves. El procesamiento de la señal varía conforme al tipo de datos que debe ingresarse al sistema, por lo cual, esta etapa difiere su metodología en el algoritmo que se esté utilizando. En la Figura 8 se observa las etapas por las que interactuó la señal de entrada para obtener el resultado.

Figura 8

Diagrama de Bloques del modelo de Clasificación



En la primera etapa se evidencia el preprocesamiento de la señal de audio, el cual consta de una normalización a la señal de entrada para evitar complejidades en el análisis de la misma, una segmentación, obtención de la energía y enventanamiento utilizando el método de Hamming.

En la segunda etapa se aplica los algoritmos del autoencoder y redes neuronales LSTM para el proceso de extracción de características y posterior obtención del modelo de clasificación de los sonidos de las aves del Ecuador.

La base de datos “Aves del Ecuador” contiene un total de 1691 variedades de sonidos de aves, tanto de los machos como de las hembras; para el pre-procesamiento se toman los audios que representan los llamados de las diferentes aves, estos cuentan con mayor cantidad de información, por lo tanto, el sistema puede tomarlos para su aprendizaje y posterior clasificación.

En la Figura 9 se puede observar los componentes más importantes de la base de datos que se encuentran separados por columnas, así en la segunda columna llamada “sound” se visualizan las muestras de cada uno de los audios, y en la tercera columna llamada “fs” se visualiza la frecuencia de muestreo de cada una de las señales de audio con la que se va a trabajar.

Figura 9

Base de Datos de los sonidos de las Aves del Ecuador

Fields	sound	fs
1	*997632x1 do...*	44100
2	*488448x1 do...*	44100
3	*1499904x1 d...*	44100
4	*574848x1 do...*	44100
5	*4606848x1 d...*	44100
6	*975744x1 do...*	44100
7	*1412352x1 d...*	44100
8	*639360x1 do...*	44100
9	*1247616x1 d...*	44100
10	*603648x1 do...*	44100
11	*1790208x1 d...*	44100
12	*3012480x1 d...*	44100
13	*464256x1 do...*	44100
14	*1037985x2 d...*	44100

Pre-Procesamiento

La señal de audio que contiene el canto de cada especie de ave presenta una frecuencia de muestreo de 44100 Hz, el número de muestras varía con respecto a la duración de cada audio original.

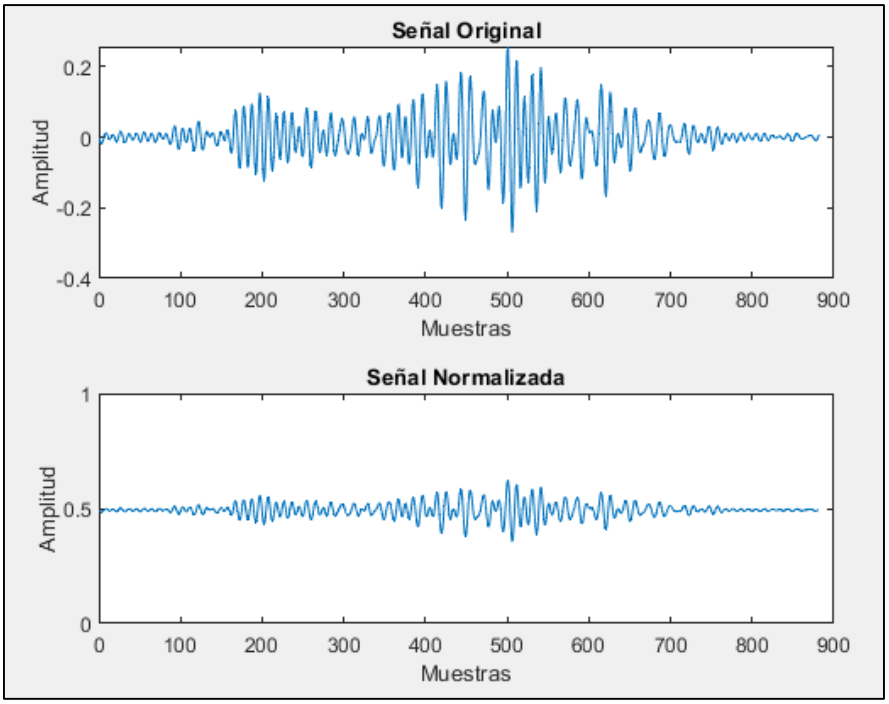
Normalización

Las señales de audio presentan diversas amplitudes con respecto a la intensidad de los sonidos que están capturados. Estas amplitudes presentan picos altos y bajos, lo que puede generar valores negativos, que al ser utilizados en el algoritmo produce un bloqueo del sistema.

Para que los datos sean reconocidos e ingresen al algoritmo del aprendizaje, tanto para las redes LSTM como para los autoencoders, se realiza una normalización de la señal de audio que ingresa y se la lleva a una escala entre 0 y 1 de amplitud aplicando la ecuación (1). La señal normalizada se la representa en la Figura 10, en la cual se observa también la escala de la señal antes de su normalización.

Figura 10

Amplitud de la señal normalizada a una escala entre 0 y 1



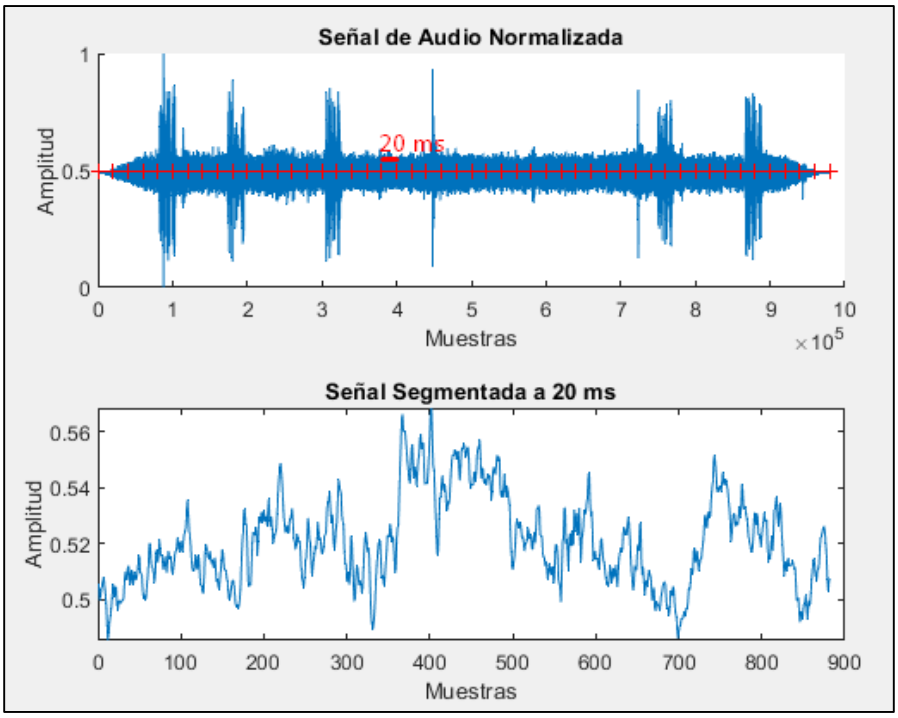
Segmentación

En este proyecto se utiliza el comando Buffer para la segmentación de la señal de audio normalizada en la cual se ingresa tanto la señal como la longitud para el segmento. Cada una de las señales de audio extrae una cantidad de muestras variables, por lo cual, para el procedimiento se realiza una división en fragmentos de 20 milisegundos como se observa en la Figura 11.

Cada segmento presenta el mismo tamaño, aprovechando de mejor manera los segmentos que contengan información válida, lo que permite obtener mejores resultados en el aprendizaje del sistema. Con este proceso se obtiene vectores que cuentan con 882 muestras.

Figura 11

Señal Segmentada en 20 ms

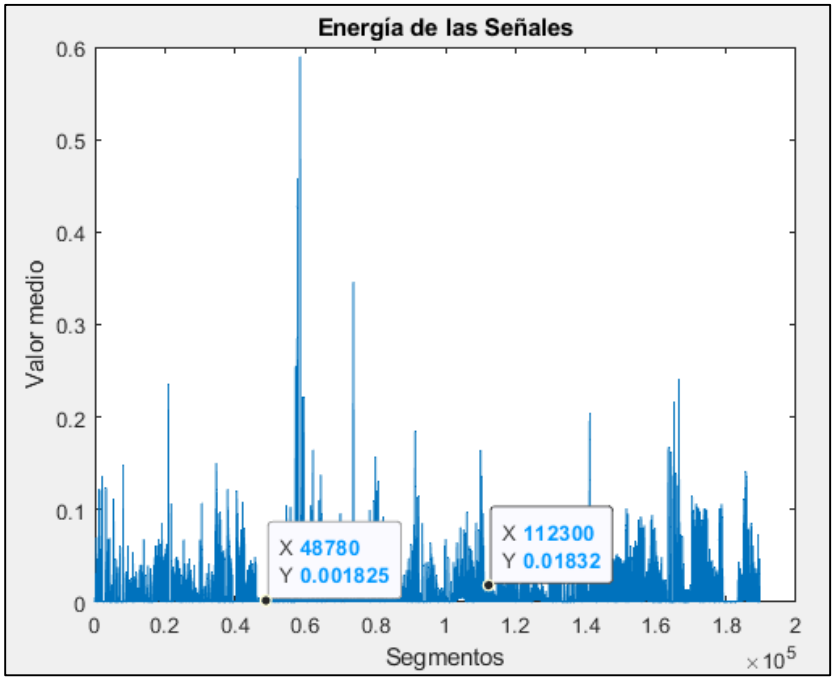


En esta etapa se genera un vector de etiquetas que enumera cada uno de los segmentos, permitiendo identificar el audio al que pertenecen.

Para acelerar y optimizar el proceso, se calcula el valor medio de cada segmento y se seleccionan los que contengan mayor información o aquellos segmentos con un valor alto de energía. Para realizar esta selección se estima un valor mínimo, es decir, se toman los segmentos con una media de energía mayor a este valor. El valor estimado para realizar este procedimiento es 0.008 de la energía de la señal.

Figura 12

Energía de cada uno de los segmentos de la señal de audio



En la Figura 12 se muestra la energía obtenida de las señales segmentadas, con ventanas de 20 ms cada uno, de los cuales se observa que algunos poseen un valor medio muy bajo. En estos segmentos no existe mucha información que el algoritmo pueda tomar para el aprendizaje, debido a esto son descartadas y se concentra el procesamiento en los segmentos con datos útiles.

Luego de este proceso se extrae de todos los segmentos los LPCs, así se trata de reducir el error de predicción al mínimo. Para el orden del filtro se tomará un valor menor o igual a la longitud del conjunto de segmentos.

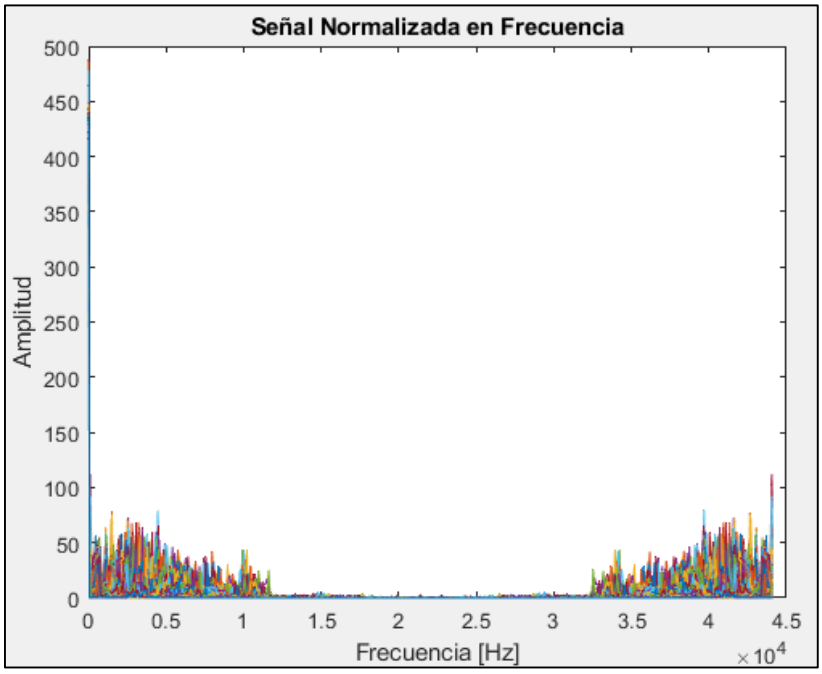
Entrenamiento

Autoencoders y capa de clasificación softmax

Las señales de audio presentan una gran cantidad de ruido de entorno, mediante la transformada rápida de Fourier se identifica con mayor eficiencia la frecuencia de la señal que ingresa al algoritmo. Para este proceso se utiliza el comando “fft” en Matlab en la señal de entrada ya procesada para la separación del ruido, así se pueden visualizar los picos de mayor magnitud al ingresar una gran cantidad de datos. Esto podemos observarlo en la Figura 13.

Figura 13

Señal Normalizada con transformada de Fourier

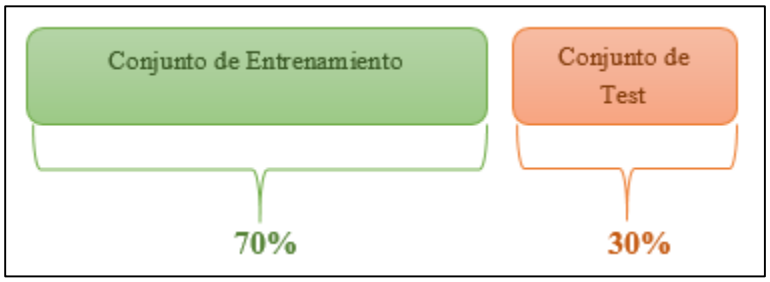


En esta etapa se realiza la extracción de características utilizando el algoritmo del autoencoder seleccionando, de la matriz de segmentos, los que se utilicen para el entrenamiento y para las pruebas,

como se muestra en la Figura 14. Se utiliza el comando cvpartition para seleccionar de forma aleatoria un 70 % de los segmentos que ingresan para el entrenamiento del autoencoder, con un total de 13314 segmentos y, cada uno con 882 muestras. El 30% restante se utilizan para las pruebas, obteniendo un total de 5706 segmentos; este proceso es ejecutado también para la selección de las etiquetas que identifican el audio al cual pertenece cada uno de los segmentos.

Figura 14

Total de los segmentos distribuidos para el entrenamiento y test



Para la división de los segmentos se escogió el modelo de cvpartition especificando una división HoldOut, ya que nos divide automáticamente los segmentos en dos conjuntos que se puede tomar para entrenamiento y prueba, cuyo porcentaje para cada conjunto se escoge con libertad, este modelo presenta un porcentaje mayor en el rendimiento del sistema y mejora la eficiencia computacional acelerando el procesamiento.

Los autoencoders tienen pesos iniciales de forma aleatoria que pueden afectar los resultados. Cada vez que se ejecute el algoritmo se obtendrán respuestas diferentes, por tal motivo se deben establecer dichos pesos de forma explícita para evitar este comportamiento.

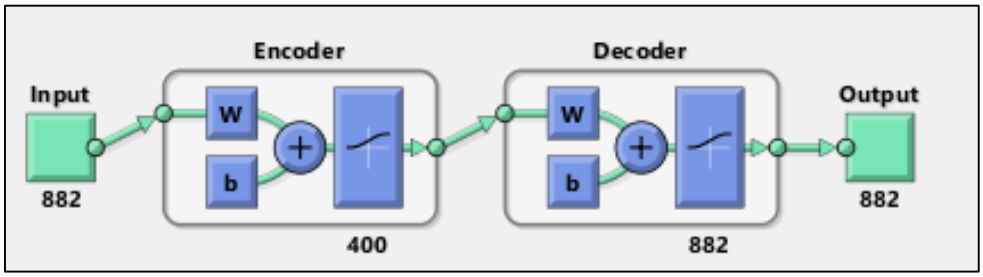
En el entrenamiento del autoencoder se establece un valor que especifica el número de neuronas en la capa oculta. El proceso que se realiza desde la capa de entrada a la capa oculta es la compresión de datos, por lo tanto, el número de neuronas debe ser menor al tamaño de los datos de entrada, capturando así datos importantes para el aprendizaje.

Para un mejor reconocimiento de los datos, que representan las señales de sonido emitidos por las aves, se trabajó con dos capas ocultas. En el algoritmo, el número de neuronas en la primera capa oculta no debe ser un valor excesivamente pequeño comparado con el número de datos de entrada, esto puede ocasionar que el sistema comprima abruptamente los datos aumentando el tiempo de procesamiento y obteniendo una gran pérdida de información.

El número de neuronas en esta capa es de 400, como se visualiza en la Figura 15; este dato es variable dependiendo del tamaño de entrada.

Figura 15

Diagrama del Autoencoder con la capa oculta de 400 neuronas



En la segunda capa se reduce aún más la cantidad de neuronas para que el autoencoder aprenda representaciones más pequeñas de los datos, debido a esto cuenta con 90 neuronas. Se escogieron estos valores mencionados para el tamaño de la primera y segunda capa oculta al observar que en los resultados el porcentaje del desempeño del algoritmo supera el 80%.

Para que el aprendizaje de los autoencoders sea más eficiente, las capas ocultas cuentan con parámetros regularizadores, cuyos valores estándar utilizables se los detalla en la Tabla 4.

Tabla 4

Parámetros Regularizadores de los Autoencoders

Parámetros	Valor Estándar	Valor Utilizado
`L2WeightRegularization`	0.001 (valor positivo)	0.0005
`SparsityRegularization`	1 (valor positivo)	1
`SparsityProportion`	0.05 (0 a 1)	0.1

Nota. Recuperado de: (MathWorks, 2023)

La configuración de los parámetros con valores cercanos a los propuestos en el estándar ayuda a minimizar el error de validación. Las cantidades que se observan en la columna de "Valor Utilizado", que se encuentran configurados en la primera y segunda capa oculta, muestran mejores resultados en la extracción de características.

El coeficiente de regulación de peso L_2 (`L2WeightRegularization`) controla los pesos de la red y, de preferencia, son valores positivos muy pequeños. Así, el número de pesos aumenta mejorando el entrenamiento.

El parámetro `SparsityRegularization` es el coeficiente que toma un valor escalar positivo y se encarga de controlar el impacto que ocasiona el regularizador de dispersión. Este parámetro asigna una restricción para la escasez de los datos en la salida de la capa oculta. La escasez de los datos hace referencia a obtener una perfección en los datos de salida.

El valor del parámetro `SparsityProportion` especifica la reacción de cada neurona en la capa oculta, ante un conjunto pequeño de ejemplos de entrenamiento. El rango de valores que puede tomar va de 0 a 1 (si la proporción es baja, la neurona aprende mediante la activación de una pequeña cantidad de ejemplos).

La salida de la segunda capa oculta se conecta a una última capa, Softmax, con una dimensión de 200. Esta capa utiliza las etiquetas de los datos de entrenamiento para realizar la clasificación de las

características, con lo que se concluye el cálculo de los resultados para los conjuntos de entrenamiento y prueba.

Redes Recurrentes LSTM

En las redes neuronales recurrentes LSTM, después de pasar la señal de entrada por las etapas de pre-procesamiento y procesamiento, se realiza una transformación de los datos en serie para que ingresen al algoritmo LSTM como se muestra en la Figura 16. En este caso se genera un vector, en la cual, cada segmento se posiciona uno a continuación de otro. Al trabajar con secuencias de datos, el tamaño de las ventanas de segmentación aumenta su valor (la señal se divide en ventanas de 500 ms). Mediante las pruebas realizadas al aminorar este valor, el porcentaje en el rendimiento del sistema disminuye.

Figura 16

Segmentos de la señal de audio en serie

	1
1	22050x1 do...
2	22050x1 do...
3	22050x1 do...
4	22050x1 do...
5	22050x1 do...
6	22050x1 do...
7	22050x1 do...
8	22050x1 do...
9	22050x1 do...
10	22050x1 do...
11	22050x1 do...

El algoritmo LSTM realizará la partición de los segmentos que se van a utilizar para el entrenamiento y para las pruebas, con ayuda del comando cvpartition; luego ingresarán en la red LSTM para su entrenamiento.

a) Configuración de la Red LSTM

En el algoritmo de la red LSTM se configuran las capas y los parámetros de optimización de la red; antes de ello se especifica el tamaño de la secuencia de entrada. Estas configuraciones las podemos apreciar en la Figura 17.

Figura 17

Capas de la red LSTM

```
layers = [ ...
    sequenceInputLayer(numSTrain)
    bilstmLayer(numHiddenUnits,'OutputMode','last')
    bilstmLayer(numHiddenUnits,'OutputMode','last')
    fullyConnectedLayer(numfiles)
    softmaxLayer
    classificationLayer];
```

La arquitectura de las capas está compuesta por:

- Capa de la secuencia de entrada: se especifica el parámetro numSTrain que representa el tamaño del conjunto de entrenamiento que ingresa al algoritmo.
- Capa oculta bidireccional (bilstmLayer): permite observar la secuencia que pasa a la siguiente capa, y especifica la cantidad de información que recordará (en este caso su tamaño es de 200, que son el número de nodos ocultos); se utiliza dos capas LSTM bidireccionales para ampliar aún más la cantidad de información que recordará la red LSTM. El parámetro numHiddenUnits varía su valor dependiendo de la cantidad de información que ingresa, y cuanta información necesite recordar para ser transmitida a las siguientes capas. Existen dos modos de salida: modo sequence que toma la secuencia de salida completa y, el modo last que toma el último dato de la clasificación, el cual fue seleccionado por presentar un porcentaje mayor en la validación de la precisión del aprendizaje de la red LSTM.

- Capa totalmente conectada (fullyConnectedLayer): especifica la cantidad de audios de los sonidos de las aves que ingresan al algoritmo por medio del parámetro numfiles.
- Capa softmax (softmaxLayer): se utiliza para clasificar los audios.
- Capa de clasificación (classificationLayer): se utiliza para visualizar los resultados finales de la clasificación.

El conjunto de parámetros de optimización de la red se llama “options”, y se especifican varios parámetros para el entrenamiento, como se muestra en la Figura 18.

Figura 18

Opciones de optimización de la red LSTM

```
options = trainingOptions('adam', ...
    'ExecutionEnvironment','cpu', ...
    'GradientThreshold',1, ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'SequenceLength','longest', ...
    'Shuffle','never', ...
    'Verbose',0, ...
    'ValidationData', {STest, ETest}, ...
    'Plots','training-progress');
```

Entre los parámetros se utiliza el solucionador de estimación de momento adaptativo (Adam), que limita el descenso de la gradiente al asignar un valor único para la tasa de aprendizaje. Otros parámetros utilizados son: el valor del umbral de la gradiente que permite realizar un ajuste en los parámetros de la red y su valor es de 2, el número máximo de épocas en que se ejecutará el algoritmo de entrenamiento se lo especifica en el parámetro (maxEpochs) con un valor de 400, el mínimo tamaño del conjunto de datos que la red toma para procesar a la vez (miniBatchSize) es de 250.

Se utiliza también un parámetro de validación de los datos de prueba (ValidationData), y una gráfica por medio del parámetro training-progress para observar el progreso de la red de entrenamiento por cada época que transcurra, en la cual también se puede notar la exactitud y las pérdidas del sistema.

Capítulo IV

Resultados

En esta sección se muestra el porcentaje del rendimiento: del sistema para los métodos propuestos en el capítulo III, de la red al realizar la extracción de características por medio de autoencoders y posterior clasificación utilizando la capa softmax, y del aprendizaje profundo por medio de redes LSTM.

Autoencoders más un clasificador simple

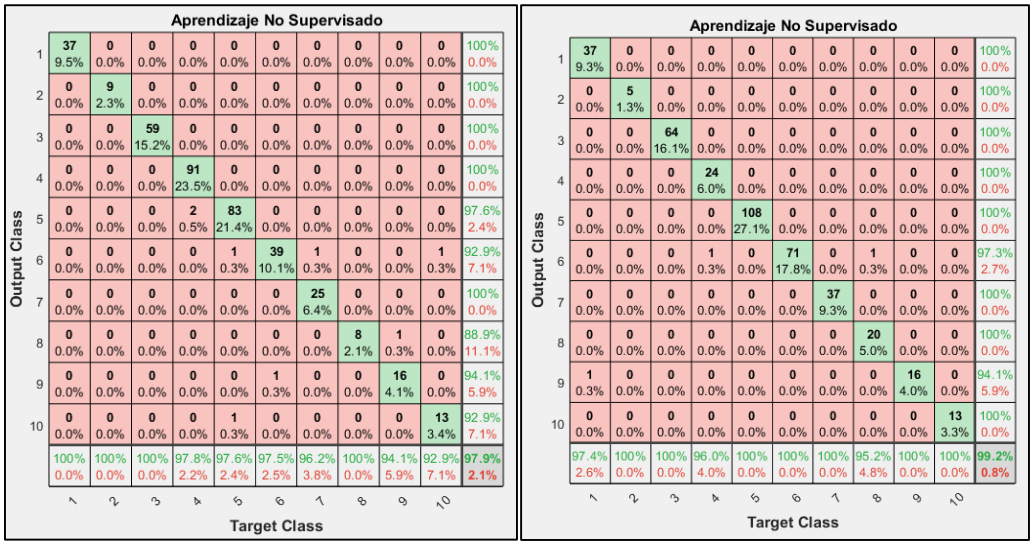
El desempeño del algoritmo de la red, al utilizar autoencoders, se evalúa por medio de la matriz de confusión. Esta permite valorar que tan bueno es el aprendizaje con dicho modelo, obteniendo los datos de exactitud y error medio, para los conjuntos de entrenamiento y de prueba.

En este apartado se mostrará los resultados bajo varios escenarios diferentes, en los cuales se varia el tamaño del enventanamiento y la cantidad de audios que ingresen al sistema. Con estas variaciones y el procesamiento mencionado en el capítulo anterior se ingresan los segmentos en el autoencoder, cuyos parámetros regularizadores que se utilizarán para todas las pruebas se encuentran especificados en la Tabla 4.

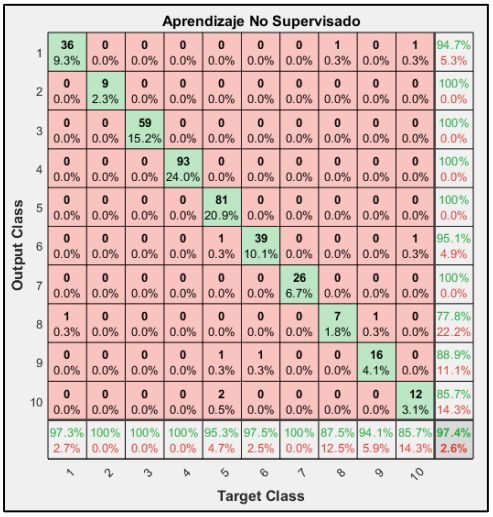
En la Figura 19 se observa el rendimiento del sistema al variar el número de capas ocultas a utilizar en el autoencoder.

Figura 19

Exactitud del sistema con a) una capa oculta, b) dos capas ocultas y c) tres capas ocultas.



a) b)



c)

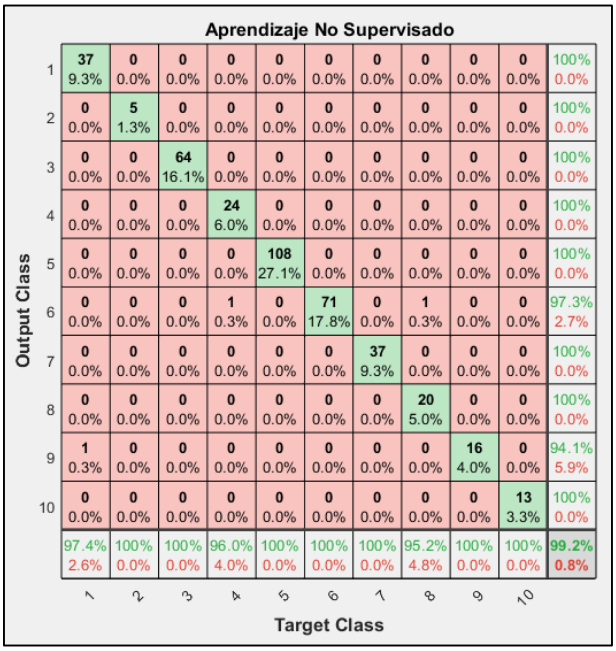
Al utilizar una capa oculta en el autoencoder se obtuvo una exactitud del 97.9 %, para aumentar la eficiencia del sistema se implementó otra capa oculta cuyo resultado incrementó a un 99.2% mejorando su exactitud. Con este dato se realizó otra prueba aumentando el número de capas ocultas a tres y se evidencia en la Figura 19 que su resultado difiere en un 1.8% menos que al utilizar dos capas ocultas. Por esta razón se trabaja para las posteriores pruebas con dos capas ocultas en el autoencoder.

Se evalúa el rendimiento para un tamaño de ventana de 20 ms, que da un número de características para cada segmento de 882 muestras y el conjunto que se elige para las pruebas es de 398 segmentos. Se presentan dos escenarios con dos entradas de: 10 audios y 50 audios.

a) Rendimiento del sistema con 10 audios

Figura 20

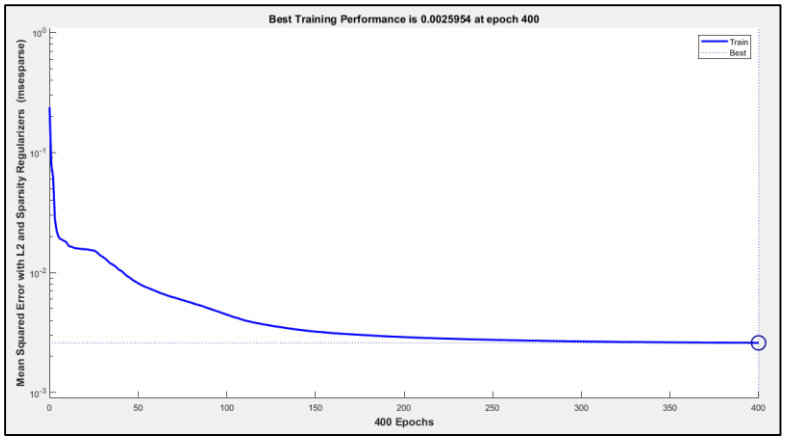
Matriz de Confusión con 10 audios



En la Figura 20 se puede observar que la probabilidad de que la red consiga aprender es de un 99.2% de exactitud, el sistema presenta un alto porcentaje de acertamiento en las predicciones, es decir las señales de audio se encuentran bien clasificadas reduciendo al mínimo el error medio como se muestra en la Figura 21 tomando un valor de 0.0025954.

Figura 21

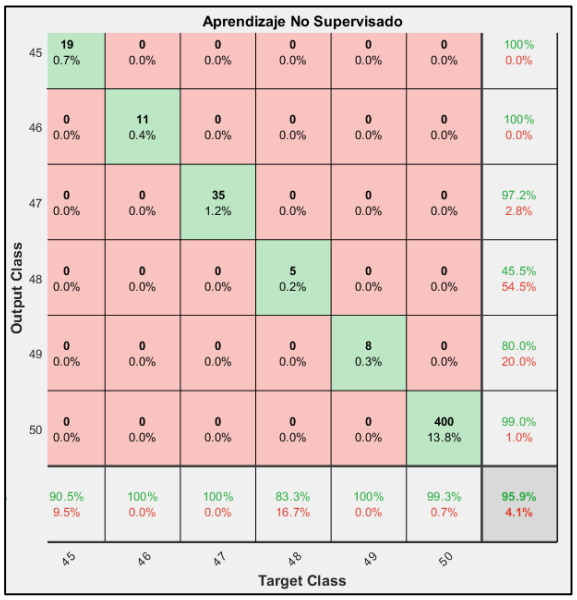
Comportamiento del error cuadrático medio con 10 audios



b) Rendimiento del sistema con 50 audios

Figura 22

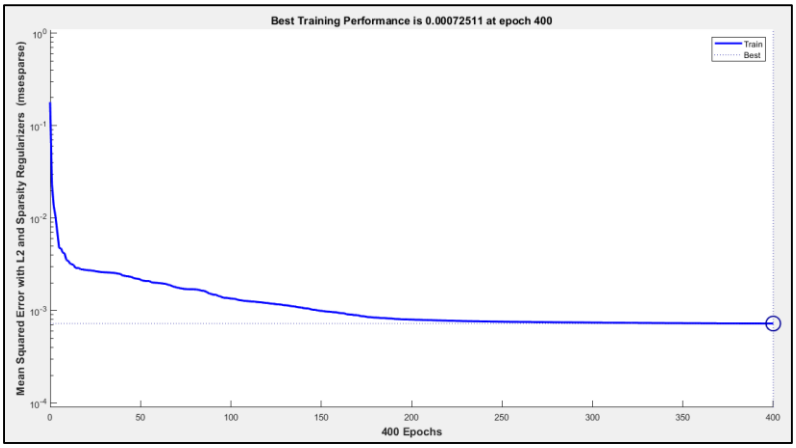
Matriz de Confusión con 50 audios



En la Figura 22 se puede observar que al ingresar una mayor cantidad de datos la probabilidad de que el sistema consiga aprender disminuye a un 95.9%, obteniendo un comportamiento del error medio como se muestra en la Figura 23 con un valor de 0.00072511.

Figura 23

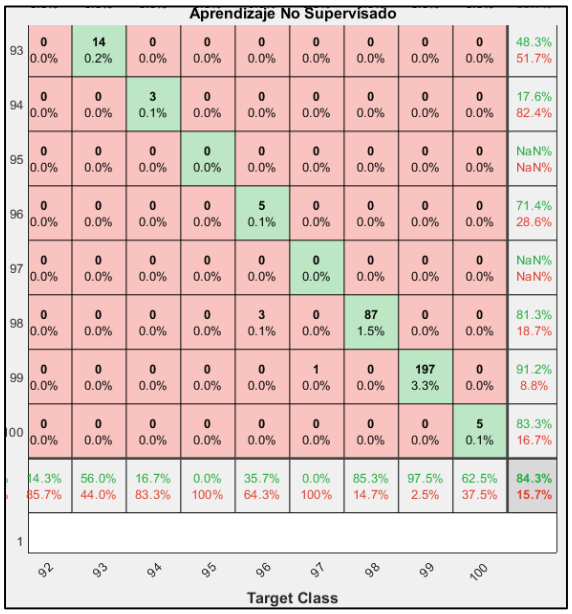
Comportamiento del error cuadrático medio con 50 audios



c) Rendimiento del sistema con 100 audios

Figura 24

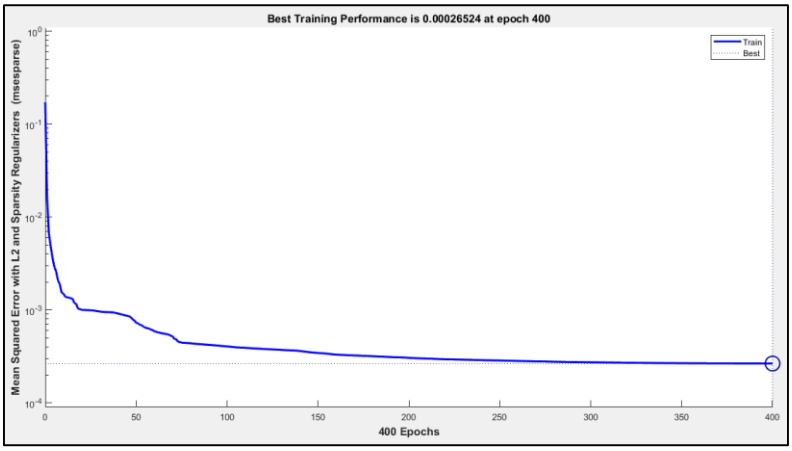
Matriz de Confusión con 100 audios



En la Figura 24 se observa que al aumentar la cantidad de audios a 100, la exactitud del sistema disminuye hasta un 84.3%, el sistema baja su rendimiento al procesar una mayor cantidad datos. La curva del error medio (Figura 25) durante el proceso del entrenamiento llega a 0.00026524.

Figura 25

Comportamiento del error cuadrático medio con 100 audios

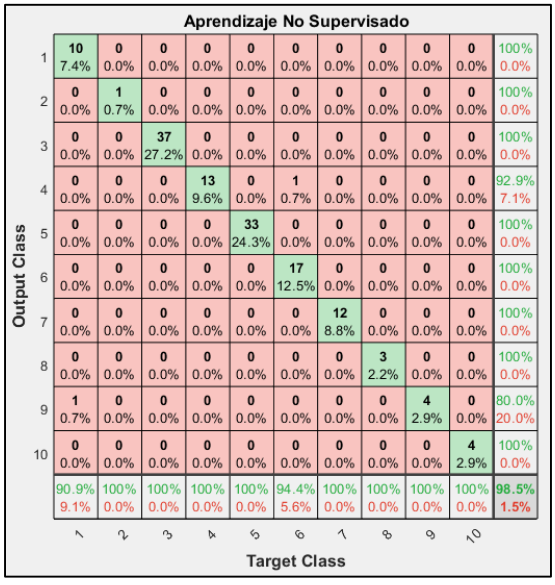


La segunda variación es con ventanas de 60 ms, el número de características en cada segmento es de 2646 muestras. Para las pruebas se toman un conjunto de 318 segmentos y se evalúa para los escenarios de 10 audios y 50 audios.

a) Rendimiento del sistema con 10 audios

Figura 26

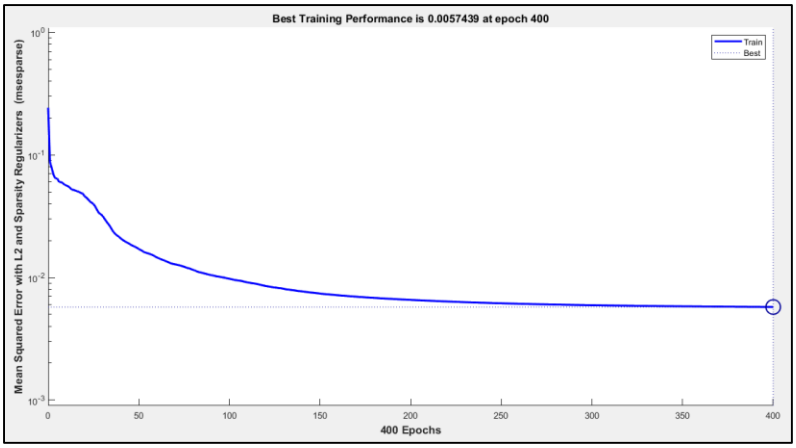
Matriz de Confusión con 10 audios



En la Figura 26 se puede observar que la probabilidad de que la red consiga aprender es de un 98.5%, el sistema presenta un porcentaje menor al incrementar el tamaño de las ventanas manteniendo la misma cantidad de audios, en la Figura 27 se observa el comportamiento del error medio en este escenario con un valor de 0.0057439.

Figura 27

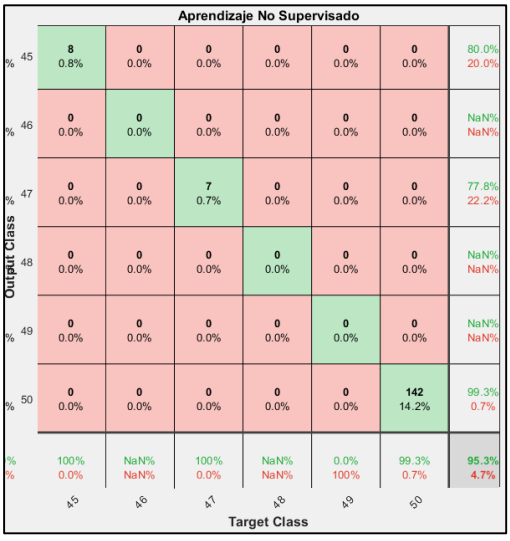
Comportamiento del error cuadrático medio con 10 audios



b) Rendimiento del sistema con 50 audios

Figura 28

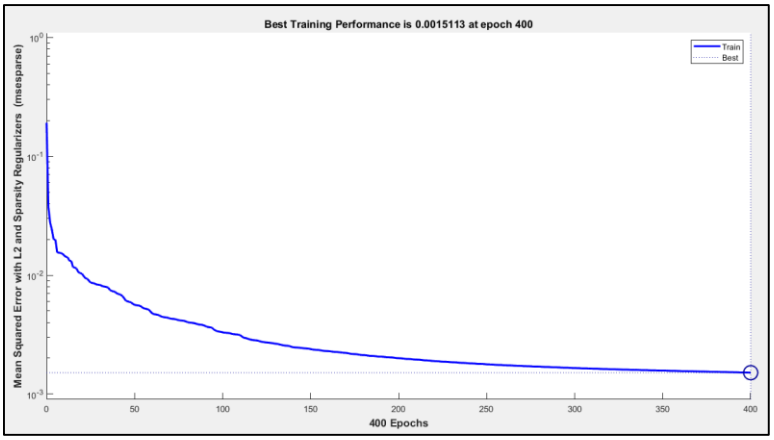
Matriz de Confusión con 50 audios



En la Figura 28 se observa que la exactitud del sistema al aumentar más audios para su aprendizaje es de 95.3%, lo cual es un 3.2% menor que al usar 10 audios manteniendo aún una alta probabilidad de que la red consiga aprender, el comportamiento de error medio se muestra en la Figura 29 con un valor del 0.0015113.

Figura 29

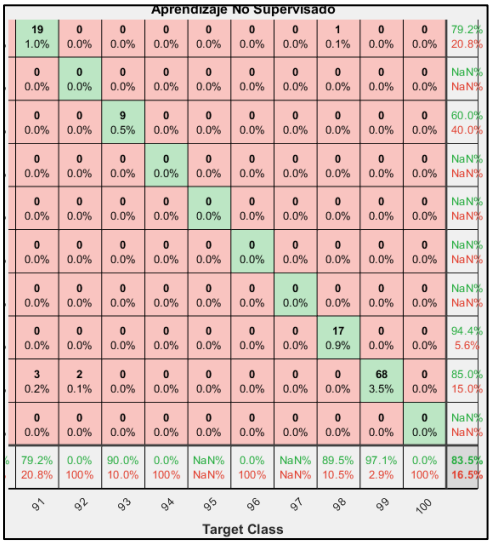
Comportamiento del error cuadrático medio con 50 audios



c) Rendimiento del sistema con 100 audios

Figura 30

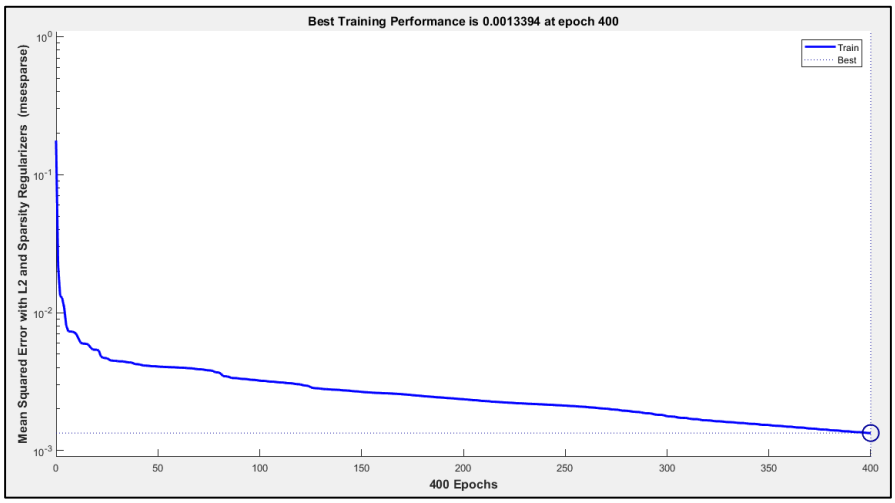
Matriz de Confusión con 100 audios



Con ventanas de 60 ms y al ingresar 100 audios, el sistema obtiene una exactitud del 83.5% como se muestra en la figura. Este resultado, con un 0.8%, está por debajo de la exactitud obtenida con ventanas de 20 ms, siendo mínima la diferencia. En ambos escenarios el sistema reduce su rendimiento mientras se ingrese una mayor cantidad de datos. El resultado de la curva del error medio durante el entrenamiento se puede observar en la Figura 31, cuyo resultado baja hasta un 0.0013394.

Figura 31

Comportamiento del error cuadrático medio con 100 audios



El resumen de los resultados en cada escenario observado se muestra en la Tabla 5.

Tabla 5

Resultados al realizar pruebas utilizando Autoencoders en los diferentes escenarios

#Ventanas	Escenario	Exactitud (%)	Error Medio
20 ms	10 audios	99.2%	0.0025954
	50 audios	95.9%	0.00072511
	100 audios	84.3%	0.00026524
60 ms	10 audios	98.5%	0.0057439
	50 audios	95.3%	0.0015113
	100 audios	83.5%	0.0013394

En este modelo se puede evidenciar, con los resultados expuestos en la Tabla 5, que al segmentar las señales de audio a 20 ms la respuesta del sistema presenta un mejor resultado en la exactitud del aprendizaje de la red. Por lo tanto, al tener un tamaño de ventana pequeño, se reconoce de mejor manera los segmentos que cuentan con información útil para el procesamiento, así únicamente ingresan estos segmentos al algoritmo del autoencoder y logre aprender.

Los porcentajes obtenidos con valor NaN en la matriz de confusión se dan al presentarse valores de cero en las proporciones indicando que no se han presentado datos falsos, al dividirse para cero el sistema lo cataloga como NaN.

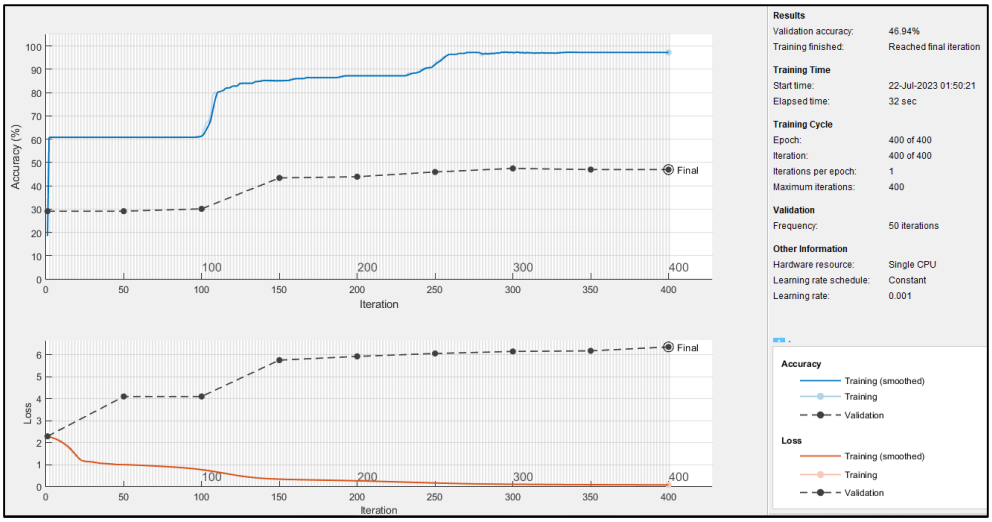
Redes Recurrentes de corto y Largo Plazo (LSTM)

Este modelo presenta una gráfica del control del rendimiento del sistema, la cual permite observar la exactitud y las pérdidas que pueda tener el sistema. Las pruebas se realizan bajo dos escenarios diferentes: variaciones en la cantidad de audios a clasificar y la utilización de coeficientes LPC para su procesamiento.

Se realiza la primera variación al utilizar 10 audios, un tamaño de ventana de 500 ms y el tratamiento de la señal con coeficientes LPC.

Figura 32

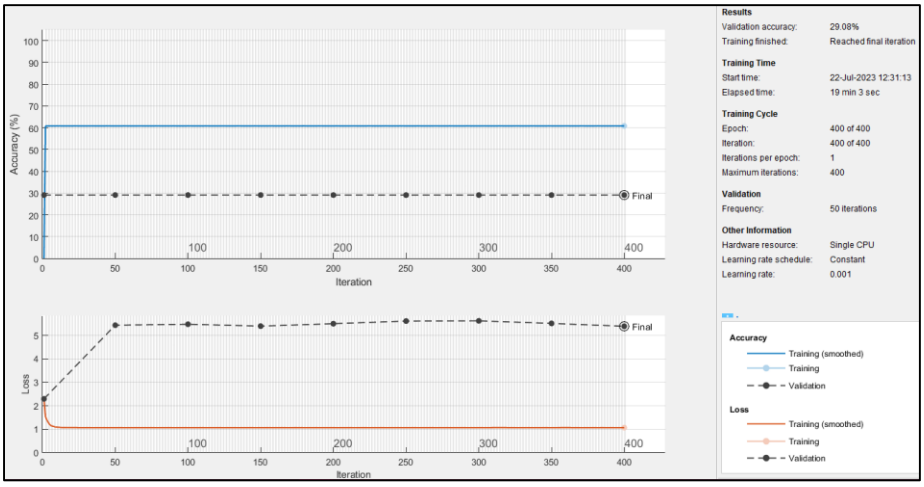
Rendimiento del sistema con 10 audios y utilizando coeficientes LPC



Se vuelve a realizar las pruebas con 10 audios sin utilizar coeficientes LPCs para procesar los datos de entrada al algoritmo LSTM.

Figura 33

Rendimiento del sistema con 10 audios sin utilizar coeficientes LPC



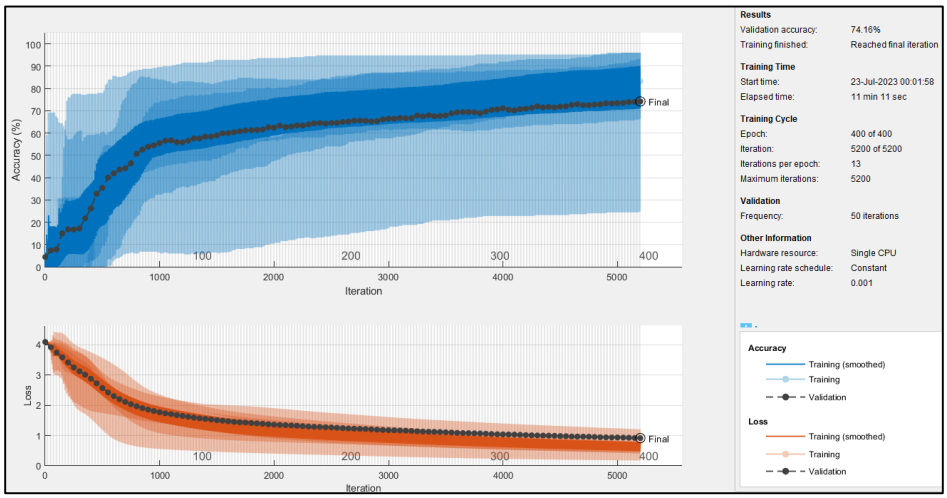
Se puede observar los resultados del rendimiento en la Figura 32 y Figura 33, el valor de la exactitud es de 46.94% y 29.08% respectivamente, la probabilidad del rendimiento del sistema con LPCs

aumenta en un 17.86% ya que estos coeficientes ayudan a minimizar el error en la predicción. Con este dato se utiliza para las siguientes pruebas los coeficientes LPC.

En la Figura 34 se muestra el resultado del aprendizaje de la red con 60 audios de entrada y un tamaño de ventana de 500 ms.

Figura 34

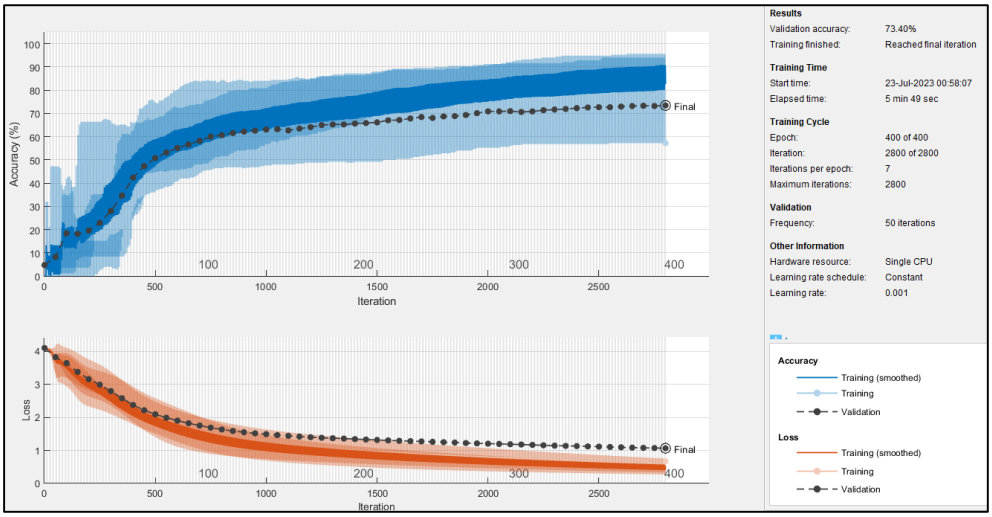
Rendimiento del sistema con 60 audios y ventanas de 500 ms



Al aumentar los audios el porcentaje de acierto subió a 74.16%, el sistema es capaz de entrenar un conjunto grande de datos y mantener una exactitud considerable. Al observar un aumento en la exactitud del sistema se realiza una variación en el tamaño de las ventanas a 900 ms y se mantiene el número de audios de entrada. El resultado de esta prueba se visualiza en la Figura 35.

Figura 35

Rendimiento del sistema con 60 audios y ventanas de 900 ms

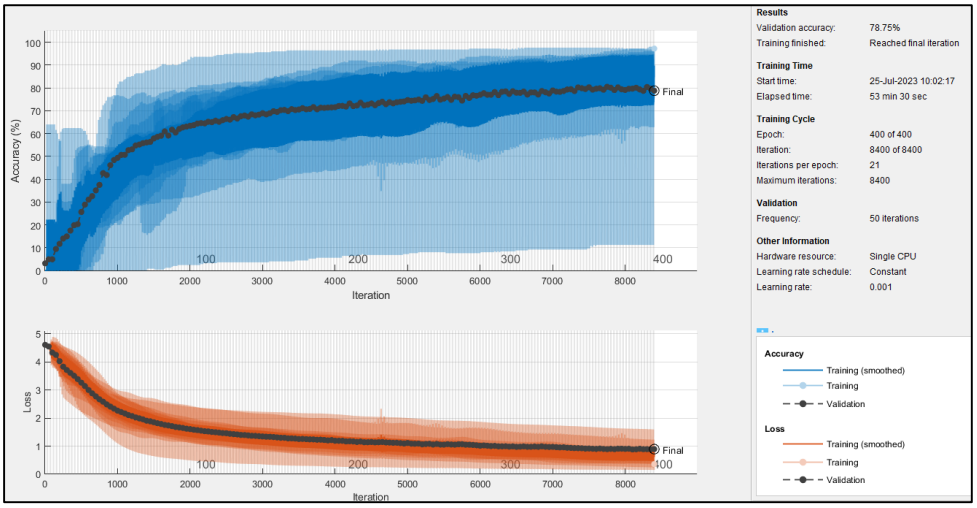


Al utilizar ventanas de 900 ms la exactitud del sistema descendió a 73.4%, la diferencia es mínima ya que disminuye un 0.76%. Al aumentar el tamaño de la ventana la cantidad de pérdidas es mayor, es decir que el aprendizaje de la red se verá afectado obteniendo predicciones erróneas.

Al observar este resultado se toma el valor del tamaño de las ventanas de 500 ms para el entrenamiento de la red mediante redes recurrentes LSTM al obtener resultados favorables en el aprendizaje de la red. En la Figura 36 se visualiza el rendimiento del sistema al aumentar aún más la cantidad de audios, para esta prueba se ingresa 100 audios.

Figura 36

Rendimiento del sistema con 100 audios y ventanas de 500 ms

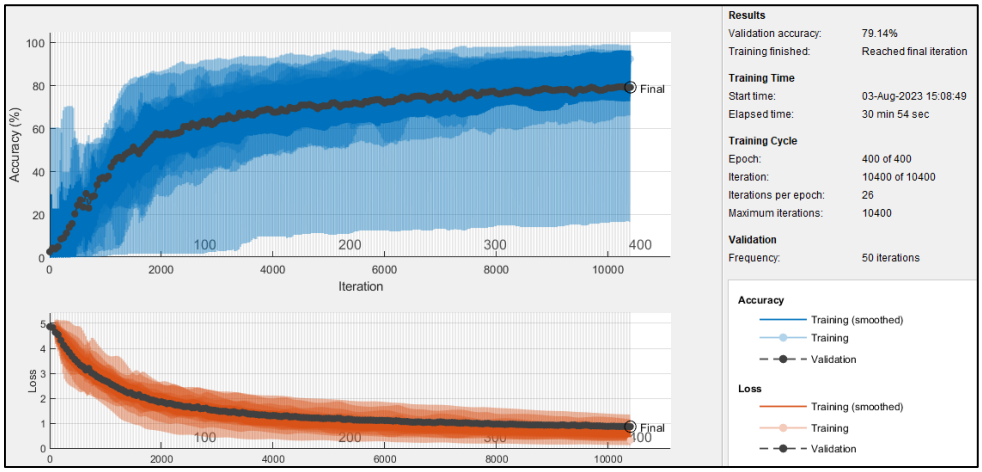


La exactitud, al ingresar 100 audios, es de 78.75%, se tiene un incremento del 4.59% al resultado presentado con 60 audios. Se evidencia que existe una relación directamente proporcional entre la cantidad de audios que ingresen al sistema y la exactitud en su rendimiento para conseguir un buen aprendizaje de la red.

Al observar un incremento en la exactitud del sistema aumentando la cantidad de audios, se realiza otra prueba con 130 audios.

Figura 37

Rendimiento del sistema con 130 audios y ventanas de 500 ms



En la Figura 37 se puede observar que se presentó un mínimo incremento en la exactitud, cuyo valor llegó hasta el 79.14%. Al aumentar la cantidad de audios el rendimiento del sistema sube.

El resumen de los resultados en cada uno de los escenarios planteados anteriormente se muestra en la Tabla 6.

Tabla 6

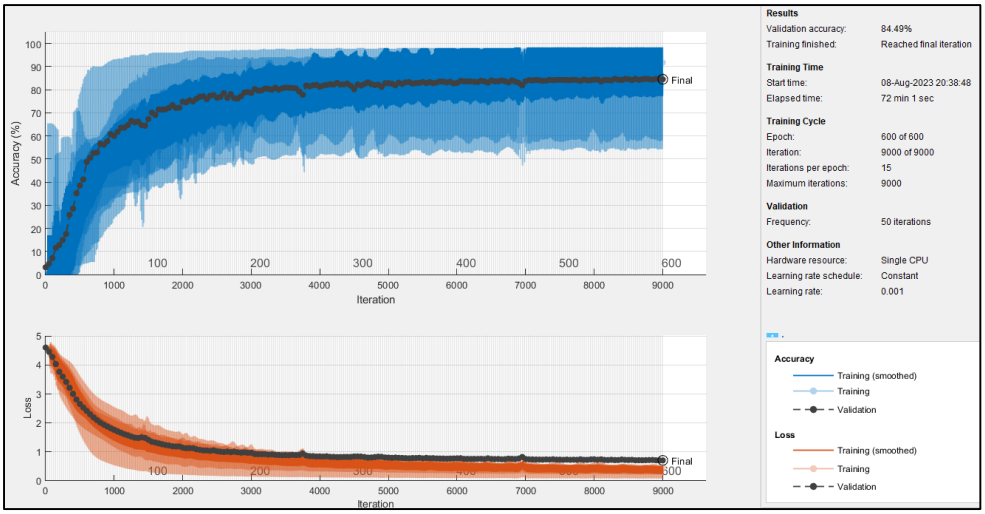
Resultados de las pruebas realizadas con diferentes variaciones

Escenario	Interacciones	Tiempo	Exactitud (%)
10 audios con LPCs	400	32 sec	46.94 %
10 audios sin LPCs	400	19 min 3 sec	29.08%
60 audios y ventanas de 500 ms	5200	11 min 11 sec	74.16%
60 audios y ventanas de 900 ms	2800	5 min 49 sec	73.40%
100 audios y ventanas de 500 ms	8400	34 min 23 sec	78.75%
130 audios y ventanas de 500 ms	10400	30 min 54 sec	79.14%

Con las pruebas realizadas se puede evidenciar que aun el rendimiento se encuentra bajo, llevando a modificar el orden de los coeficientes LPC aumentando su valor para permitir el ingreso de una cantidad mayor de segmentos que contengan información válida que se pudo haber descartado para el aprendizaje de la red. Con estas variaciones se realizaron pruebas ingresando 100 y 130 audios.

Figura 38

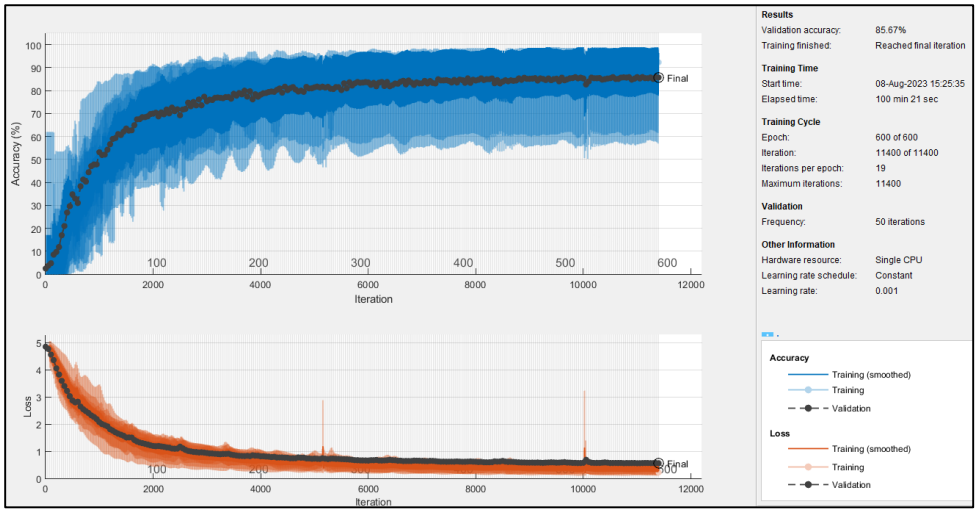
Rendimiento del sistema con 100 audios y un orden de 64 coeficientes LPC



En la Figura 38 se puede observar que el rendimiento aumentó hasta un 84.49%, al tener una mayor cantidad de datos a procesar el tiempo será mayor en comparación con los resultados que se visualizan en la Tabla 6, pero su exactitud incrementa en un 5.74%. Se ingresa 30 audios más para visualizar la variación del rendimiento que presenta el sistema.

Figura 39

Rendimiento del sistema con 130 audios y un orden de 64 coeficientes LPC



Con 130 audios la exactitud subió hasta un 85.67% como se muestra en la Figura 39, mejorando el aprendizaje del sistema. Este porcentaje supera el resultado que se ha obtenido al utilizar autoencoders con un ingreso de 100 audios. Se presenta a continuación en la Tabla 7 un resumen de los últimos resultados realizando las modificaciones propuestas.

Tabla 7

Resultados de las pruebas realizadas al modificar el orden de los coeficientes LPCs

Escenario	Interacciones	Tiempo	Exactitud (%)
100 audios	9000	72 min 1 sec	84.49 %
130 audios	11400	100 min 21 sec	85.67%

Capítulo V

Conclusiones

Con la implementación del modelo propuesto de aprendizaje no supervisado con autoencoders, se logra obtener resultados aceptables. Al trabajar con señales de audio, este modelo cuenta con características que reduce la distorsión permitiendo llegar a una exactitud del 99.2%. Al utilizar un autoencoder con dos capas ocultas llevó al fortalecimiento en el entrenamiento ayudando a incrementar la exactitud, obteniendo un error medio del 0.0025954. Este modelo tiende a disminuir su rendimiento al ingresar más audios para su análisis. Por tal motivo los autoencoders presentan altas probabilidades en el aprendizaje de la red cuando trabajan con una cantidad limitada de datos.

Se comprendió el funcionamiento de las redes recurrentes LSTM, para una mejor predicción de la red se realiza el procesamiento previo de los datos utilizando coeficientes LPCs, con ello se obtuvo una reducción en el tiempo de procesamiento de 19 min con 3 sec a tan solo 32 sec trabajando con 10 audios, al trabajar con 130 audios y utilizando los LPCs el tiempo subió a 100 min con 21 sec, ya que se ingresó más información al proceso. La exactitud del modelo mejoró con la utilización de estos coeficientes hasta llegar a un porcentaje del 85.67%.

Al aumentar la misma cantidad de audios a procesar en los modelos implementados, se tiene que con 100 audios ambos modelos presentan un rendimiento alrededor del 84%, las redes recurrentes LSTM llegan al nivel del rendimiento de los autoencoders. El comportamiento difiere al aumentar la cantidad de audios, por la parte de los autoencoders disminuye su rendimiento, pero las redes recurrentes LSTM incrementa el valor de la exactitud del sistema.

Finalmente se logró implementar un clasificador para la identificación de los sonidos emitidos por las aves. Con este sistema se puede ayudar en los centros de protección de aves silvestres al poder reconocer ciertas especies que se encuentren en la zona, ya que, al ingresar una cantidad específica de audios, la probabilidad de aprendizaje será alta.

Trabajos Futuros

En este trabajo de investigación se utiliza, para la clasificación de los audios de las aves, los autoencoders y las redes recurrentes LSTM, se propone extender hacia la aplicación de otras técnicas tanto de machine learning como de Aprendizaje profundo, y observar la variación en las métricas de rendimiento.

Sería interesante aplicar este sistema para identificar el sonido de otras especies de fauna silvestre, de instrumentos musicales, de automóviles o en la identificación del idioma de las personas que presentan audios más largos y complejos.

Este proyecto se puede aplicar también en el área de la ingeniería automotriz, en la identificación del sonido emitido por el motor del automóvil, para detectar algún problema de forma inmediata y conocer las posibles soluciones.

Bibliografía

Altamirano, S. (2021). *Sistema de reconocimiento de microterremotos en tiempo real del volcán Cotopaxi aplicando aprendizaje supervisado.* Sangolquí.

Angulo, W. (2020). *Análisis nutricional de los alimentos soportado en Aprendizaje No Supervisado.* Popayán.

Atienza, A. (2019). *Detección e identificación automática de actrices y actores mediante el uso de algoritmos de Deep Learning.* Valladolid-España.

bioWeb Ecuador. (2018). Obtenido de Aves del Ecuador: https://bioweb.bio/faunaweb/avesweb/Vocalizaciones/

Borja, R., Monleón, A., & Rodellar, J. (2020). *Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning.* Barcelona-España.

Caicedo, E., & López, J. (2017). *Una aproximación práctica a las redes neuronales artificiales.* Cali-Colombia.

Cajas, C., & Villalva, L. (2019). *Sistema Web de Monitoreo de Aves de la Provincia de Cotopaxi (SIMA).* Latacunga-Ecuador.

Capacete, F. (2019). *Protección antes del peligro de extinción.* Baleares.

Chanco, W., & Narváez, M. (2018). *DISEÑO EDITORIAL ILUSTRADO DE LAS AVES EN PELIGRO DE EXTINCIÓN DE LA ZONA 3.* Latacunga-Ecuador.

Coto, M. (2017). *Síntesis de voz basada en modelos ocultos de Markov y algoritmos de aprendizaje profundo.* México.

eBird. (27 de Julio de 2022). Obtenido de eBird: https://ebird.org/home

García, A., Gallego, E., Domínguez, I., Correa, A., & Rodríguez, J. (2017). *Codificación de voz mediante coeficientes de prediccion lineal (LPC) sobre Microblaze.* Cuba.

González, I. (2019). *Análisis y comparación de extracción de características en señales de audio.* Madrid-España.

González, J. (2017). *Aprendizaje profundo para el procesamiento del lenguaje natural.* Valencia.

González, J. (2019). *Procesamiento de Audio con Técnicas de Inteligencia Artificial.* Madrid.

González, J. (2021). *Modelos de Aprendizaje Profundo con Auto-atención para detección de eventos de audio.* Madrid.

González, J., Padrón, J., Barbero, I., Custodio, L., & Merchán, F. (2019). Reconocimiento de canto de aves basado en el análisis de componentes principales del espectrograma. *RIC*, 6.

Lara, M. (2021). *Sistema de Reconocimiento Automático de Micro sismos Volcánicos basado en Redes Neuronales Convolucionales.* Quito-Ecuador.

Martínez, R. (2017). *Análisis de las máquinas "Sparse Autoencoders" como extractores de características.* Cartagena.

Martínez, R. (2019). *Simulating music from the latent space of a Variational Autoencoder.* Madrid-España.

Martínez, S. (2020). *Sistema en matlab para la segmentación y clasificación automática de sonidos de ballenas.* Quito-Ecuador.

MathWorks. (2022). Obtenido de Machine Learning: https://la.mathworks.com/discovery/machine-learning.html

MathWorks. (2023). Obtenido de https://la.mathworks.com/help/signal/ref/lpc.html

MathWorks. (2023). Obtenido de https://la.mathworks.com/help/deeplearning/ref/trainautoencoder.html

Oña, M. (2020). *Desarrollo de una aplicación para generar ritmos de batería a través de técnicas relacionadas con aprendizaje automático.* Quito-Ecuador.

Rivas, M. (2020). *Análisis de Líneas de Costa con Redes Neuronales LSTM.* Cataluña-España.

Rodriguez, J. (2020). *Diseño y Simulación de un Filtro Digital para Señales EEG con el Paradigma de Imaginación Motora en FPGA.* Lima-Perú.

Rosario, E. d. (6 de Febrero de 2017). *Señales y Sistemas*. Obtenido de Señales de Energía y Potencia: http://blog.espol.edu.ec/telg1001/senales-de-energia-y-potencia/

Salgado, L. (2022). *Aprendizaje en Máquina Aplicado a la Detección de Fallas Mecánicas.* Ciudad de México-México.

Shiguango, W., & Bañol, C. (2020). Evaluación rápida de la avifauna en el Centro de Investigación, Posgrado y Conservación. *Cienc Tecn UTEQ*, 10.

Zumba, E., & Zumba, F. (2022). *DESARROLLO DE UN SISTEMA PROTOTIPO DE GENERACIÓN DE ROSTROS A PARTIR DE SEÑALES DE VOZ UTILIZANDO REDES GENERATIVAS ADVERSARIAS.* Cuenca-Ecuador.