Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Maestría en Ciencias
en Ciencias de la Computación

Clasificación de ladridos de perros domésticos usando aprendizaje profundo

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de

Maestro en Ciencias

Presenta:

José Ramón Gómez Armenta

Ensenada, Baja California, México

2019

Tesis defendida por

José Ramón Gómez Armenta

y aprobada por el siguiente Comité

Dr. José Alberto Fernández Zepeda

Codirector de tesis

Dr. Humberto Pérez Espinosa

Codirector de tesis

Dr. Hugo Homero Hidalgo Silva

Dr. Raúl Rangel Rojo

Dr. Ubaldo Ruiz López

Coordinador del Posgrado en Ciencias de la Computación

Dra. Rufina Hernández Martínez

Directora de Estudios de Posgrado

Resumen de la tesis que presenta José Ramón Gómez Armenta como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Clasificación de ladridos de perros domésticos usando aprendizaje profundo

Resumen aprobado por:

Dr. José Alberto Fernández Zepeda	Dr. Humberto Pérez Espinosa
Codirector de tesis	Codirector de tesis

Esta investigación se sitúa dentro del campo de análisis inteligente de audio. En este campo se busca desarrollar métodos computacionales que puedan ser útiles para resolver diferentes problemas de interés; en particular, en este trabajo se estudia la clasificación de ladridos del perro doméstico. El método propuesto se desarrolló como una herramienta basada en el aprendizaje profundo que facilita la identificación del ladrido del perro y reconoce al individuo, especie, edad, sexo y contexto asociado a cada ladrido, con la finalidad de estimar información que aporte un perfil más amplio del animal. Esto puede ser de interés para las personas que trabajan y conviven con estos animales.Se construyeron cinco modelos distintos para cada una de las tareas de clasificación y la metodología se divide en 3 etapas: En la etapa de preprocesamiento se utilizó la información de tres bases de datos. Se aplicaron técnicas para la preparación de los datos con el formato para cada uno de estos modelos. En la etapa de caracterización. Se evaluaron distintas técnicas para seleccionar la técnica de mejor desempeño, con las que se extrajeron los atributos más importantes y se prepararon los datos para la siguiente etapa. En la tercera etapa de clasificación. Se entrenó una red neuronal profunda para cada modelo, se evaluaron distintas arquitecturas de redes neuronales para seleccionar las más adecuada y se afinaron los hiper-parámetros para cada modelo. En esta investigación se concluyen cuáles fueron las características más relevantes que se extrajeron de las bases de datos y la arquitectura de red neuronal profunda más adecuada para ese tipo de características, además de el método para encontrar la mejor combinación de hiper-parámetros.

Palabras clave: aprendizaje Profundo, red neuronales, hiper-parametros

Abstract of the thesis presented by José Ramón Gómez Armenta as a partial requirement to obtain the Master of Science degree in Computer Science.

Classification of domestic dog vocalizations using deep learning

Abstract approved by:

Dr. José Alberto Fernández Zepeda
Thesis Co-Director

Dr. Humberto Pérez Espinosa
Thesis Co-Director

The research done in this investigation corresponds to the field of intelligent audio analysis, applying a deep learning approach to distinct properties of sounds that may be useful to solve different tasks of classification for domestic dog vocalizations. The proposed method was developed as a tool that helps the identification of the dog's bark and recognizes the identity, breed, age, sex and context associated with each bark, in order to estimate information that provides a broader profile of the animal, this may be of interest to people who work and interact with these animals. Five different models are constructed for each of the classification tasks, and the methodology is divided into three stages: In the preprocessing stage, a combination of three databases are used, all the similar tags for each task are joined together, then processing techniques were applied to prepare the data in the appropriate format for each of these models. In the stage of characterization, different techniques are evaluated, where the best performance technique is selected, the most important attributes are extracted and the data is prepared for the next stage. In the stage of classification, each classification model is trained selecting the best hyper-parameters search for each model. Once the model is tuned and trained, the final prediction results are achieved. This investigation concludes with the most relevant characteristics extracted from the databases and the most appropriate deep neural network architecture for that type of characteristics, in addition to the method to find the best combination of hyper-parameters. This method is an effective proposal with outstanding performance and is in the state of the art, regarding the problem of barking classification, providing results that surpass previous investigations and classifying additional tasks that were not included before.

Keywords: deep learning, neural networks, hyper-parameters

Dedicatoria

Dedico a este trabajo a mi perro Chopin, y a los demás perros que no tienen hogar, con la esperanza que algún día todos puedan estar bien cuidados y ser queridos.

Agradecimientos

Le agradezco a todas las personas involucradas en mi desarrollo durante esta etapa de mi vida, como lo han sido mis profesores y compañeros de posgrado por las enseñanzas, los retos, la motivación y la ayuda durante el camino a convertirme en una mejor persona tanto en el ámbito profesional como en el personal con valores de perseverancia, solidaridad, respeto y compromiso.

Le agradezco a mis co-directores y al comité de la tesis por compartir su conocimiento, por el apoyo, la disciplina y la iniciativa de aprender e innovar, que fueron necesarios para este trabajo.

A mis padres que han estado ahí durante toda mi formación académica, ayudándome a crecer y brindando su apoyo moral e incondicional.

Finalmente le agradezco al CONACYT y a CICESE por el apoyo financiero durante este periodo de estudios y trabajo.

Gracias a todos, sin su apoyo no seria posible mi formación y mucho menos los logros con los que se concluye este trabajo.

Tabla de contenido

Página

Tabla de contenido (continuación)

Lista de figuras

Figura Página

Lista de figuras (continuación)

Figura Página

Lista de tablas

Tabla Página

Capítulo 1. Introducción

Este estudio se sitúa dentro del campo de análisis inteligente de audio y busca desarrollar métodos computacionales y técnicas que logren identificar propiedades de los sonidos que puedan ser útiles para resolver diferentes problemas y tareas relacionadas con la clasificación de ladridos del perro doméstico. En el campo de análisis de audio se han desarrollado una gran variedad de técnicas de caracterización para reconocimiento del habla humana. Con estas técnicas se busca encontrar propiedades de la señal acústica que sean útiles para clasificar correctamente las tareas involucradas en el contexto del habla humana. Estas técnicas podrían adaptarse para clasificar otro tipo de vocalizaciones.

A pesar de ésto, aún son limitadas las investigaciones que hacen uso de estas técnicas para el caso de vocalizaciones emitidas por animales y, en específico, las del perro. Se puede ampliar la investigación en este tema proponiendo el uso de diferentes métodos con el potencial de generar mayor conocimiento, lo que impactaría en la aplicación de esta tecnología.

Este trabajo se enfoca en el perro doméstico ya que forma parte de las mascotas más comunes que se pueden encontrar. Al perro se le trata como a un miembro más de la familia, ya que brinda buena compañía, cariño y seguridad. En general, las personas que conviven frecuentemente con perros suelen preocuparse por la salud y el bienestar de estos animales.

También existe interés de los etólogos, que se dedican a estudiar el comportamiento de los animales, en crear perfiles de conducta para poder monitorear y predecir los comportamientos de los perros. El perro, por su gran olfato e instinto, sirve al humano en labores de importancia en los que son adiestrados. Algunos ejemplos son los perros lazarillos, perros detectores y perros de salvamento, entre otros. Esta investigación podría servir a los entrenadores y profesionales que trabajan con estos animales para facilitarles el trabajo de cuidarlos, además de ayudar a comprender mejor su comportamiento y conductas.

Los ladridos y otras vocalizaciones del perro doméstico tienen características acústicas importantes, tales como frecuencia, amplitud, tono y ritmo, entre otras Pongrácz *et al.* (2005). Estas características pueden relacionarse con una emoción, reacción fi-

siológica, actitud o estado particular del perro. En el campo del análisis de audio inteligente se utilizan diversos métodos para el procesado de la señal. Estos métodos se basan en el aprendizaje automático y otros tipos de algoritmos para analizar las características de la señal acústica digitalizada.

Para el estudio del comportamiento animal es importante tener métodos capaces de identificar las propiedades de la señal de audio que se obtiene a partir de grabaciones. Un punto importante en el análisis de la información ya procesada es deducir cuáles de estas propiedades son de mayor utilidad para poder caracterizar diferentes tareas.

Para clasificar las tareas que se proponen en este trabajo de investigación se utiliza un conjunto de bases de datos con grabaciones de ladridos. Éstas contienen información acerca del individuo, raza, sexo y edad del perro, además de distintos contextos en los que se grabó el ladrido del perro.

Se debe tener en cuenta que la tecnología presentada aún está en desarrollo, pero se obtuvo un buen desempeño al clasificar las tareas especificadas a partir de la información disponible en las bases de datos; sin embargo, ya que al construir un modelo de clasificación siempre existe la pequeña posibilidad de error, se hace un esfuerzo para minimizar éste.

También se debe considerar que la diversidad de datos es limitada, por lo que los resultados de este trabajo no incluyen todas las tareas posibles que existen en el mundo real. No se cuenta con muestras de datos para todas las razas, edades y contextos. Cuando se intenta clasificar para algún caso no conocido por el entrenamiento del modelo, habrá una confusión en el modelo y la predicción será alguna clase con la que tenga el mayor parecido.

A pesar de que en pocos trabajos de investigación ya se ha utilizado algunas técnicas de reconocimiento de audio para este problema, se busca seguir explorando cómo mejorar la exactitud de la clasificación en diferentes tareas. Algunos ejemplos de las tareas son encontrar el contexto en el que se emitió el ladrido, identificar al individuo que corresponde al ladrido, el sexo del perro, el tipo de raza correspondiente al ladrido y la categoría de edad en la que entra el ladrido del perro; por lo que se aprovecha la información de los datos que se tiene a disposición.

El propósito principal de este estudio es analizar la viabilidad de utilizar redes neuronales profundas y comparar la mejora que se puede lograr contra otros algoritmos previamente utilizados en el área de aprendizaje automático.

Se usan las redes neuronales artificiales ya que en la actualidad han demostrado tener un desempeño sobresaliente, tanto en tareas de reconocimiento de imágenes como en clasificación de audio. Para lograr este objetivo, se desarrolla una estrategia con la que se buscan los parámetros de afinación óptimos para distintas redes neuronales implementadas para resolver las tareas de clasificación que se proponen.

También se hace una aportación importante para la parte de caracterización, donde se evalúa cuáles son las características acústicas más adecuadas para enfrentar este problema.

Adicionalmente, se pretende mejorar la flexibilidad del modelo de clasificación al incluir mayor diversidad de datos. Para lograrlo, se obtuvieron distintas bases de datos con tareas muy similares y se buscó un método para unificarlas. En resumen, este trabajo busca ampliar la investigación en el tema al evaluar diversas técnicas de caracterización y de clasificación con diferentes arquitecturas de redes neuronales profundas, para identificar un método adecuado para la clasificación de ladridos.

1.1. Justificación de la investigación

El ladrido del perro proporciona información sobre el estado del perro, sobre lo que sucede en su ambiente y sus características físicas. Con esta información se pueden desarrollar aplicaciones que hagan uso de la interpretación automática de ladridos. Lo cual puede servir para mejorar el cuidado del perro en el hogar, uso recreativo para la familia o dueño del perro, aplicaciones de seguridad y salud de los perros.

Los dueños y personas que conviven frecuentemente con un perro se preocupan por su aspecto, estado físico, salud y ánimo que presenta. Ésto los lleva a que inviertan en su bienestar, el que puede incluir distintos aspectos como los son la higiene, ejercicio, compra de juguetes y diversos accesorios.

Existe una gran cantidad de dispositivos inteligentes que se han incorporado a

la vida cotidiana y hay un gran oportunidad de mercado para las mascotas como el perro. Los productos que existen de este tipo no han sido muy populares, debido a que son poco confiables en la práctica. Los esfuerzos han sido insuficientes comparados a los que se han logrado en dispositivos que hacen uso del reconocimiento del habla humana. Es complejo crear un modelo sólido para todos los perros en general; por lo que se busca desarrollar modelos que sean eficientes en la práctica.

En la actualidad se ha popularizado el tema de la inteligencia artificial, tanto en proyectos emprendedores, como en la industria. Se han presentando una gran cantidad de retos tecnológicos en los que el desempeño del aprendizaje automático y las redes neuronales son sorprendentes. En algunas áreas, la inteligencia artificial ofrece soluciones de vanguardia, como en el caso del reconocimiento visual y el reconocimiento del lenguaje natural.

El trabajo desarrollado en este proyecto se basa en algunos trabajos del estado del arte que utilizan técnicas de aprendizaje profundo. También se apoya de técnicas de aprendizaje automático y procesamiento de señales. Se busca crear un método basado en aprendizaje profundo, para ésto se cuenta con la cantidad de datos y poder de cómputo suficiente. El trabajo realizado involucra el análisis de datos, caracterización, construcción de una arquitectura para la red neuronal, afinación de parámetros, clasificación, y el reajuste del modelo con base en los resultados. El proceso anterior involucra desarrollar una metodología específica para este problema, lo cual es el principal enfoque de trabajo en esta investigación.

El conocimiento generado a partir de este estudio aporta al desarrollo de esta tecnología y puede ayudar a los etólogos a perfilar al animal y comprender mejor su conducta. Con ayuda de estos métodos computacionales se podrían crear nuevos métodos que generen nuevo conocimiento en el campo de la etología y faciliten el estudio de las diferencias individuales entre perros. Se podría mejorar el bienestar de los perros y seleccionarlos para ciertas labores en un ámbito más práctico. Esto también ayuda al momento de trabajar con una gran cantidad de perros, ya que una clasificación automática podría crear los perfiles de distintos individuos de una manera más eficiente.

Existen otros campos de estudio que se interesan en la información que se pue-

de obtener de los ladridos. Algunos ejemplos son los siguientes: en la guía auditiva para personas ciegas a través de las propiedades que se obtienen de los ladridos; en el estudio de padecimientos patológicos y problemas psicológicos dentro de razas de perros; para el adiestramiento de perros que tienen problemas de conducta; en el desarrollo de técnicas de interpretación de vocalizaciones con otros animales y en el desarrollo de aplicaciones como lo son la alarma perruna que se trabajado por parte de CICESE-UT[3].

1.2. Antecedentes

El problema de reconocimiento de ladridos no es un problema reciente ya que existen algunos investigadores que previamente lo estudiaron con técnicas del área de aprendizaje automático y otros algoritmos.

Sin embargo, en los trabajos previos que se mencionan más adelante sólo se experimenta con la clasificación de una tarea especifica para la base de datos que se utilizó. En este trabajo de investigación se utilizan todas las bases de datos y tareas clasificadas como recursos para el desarrollo de los experimentos propuestos. Por lo que en esta sección se mencionan los trabajos más destacados a los que se hará referencia continuamente.

Existen trabajos en el área del comportamiento animal en cuyos experimentos se analiza si los ladridos se pueden interpretar. En estos experimentos se busca saber si entre los contextos específicos en los que se produce el ladrido se puede percibir información que los humanos puedan asociar a un contexto y ser capaces de clasificarlos.

En el estudio de Pongrácz *et al.* (2005) se realizó un experimento que consistió en clasificar muestras de ladridos de perro mediante oído y asociarlos a estados emocionales del perro como: agresión, miedo, desesperación, alegría y juego. Los resultados fueron favorables ya que la mayoría de las respuestas de los participantes coincidieron con los estados asociados a las muestras de los ladridos que se escucharon.

A partir del trabajo de Yin *et al* (2002), se sabe que los ladridos tienen características acústicas que se pueden aprovechar, algunas de las más básicas son: frecuencia,

amplitud, tonalidad y ritmo. Estas características pueden variar bajo un contexto específico, de manera que se asocia al estado que presenta el perro. También se descubrió que ciertas razas de perro ladran en baja frecuencia ante la presencia de un extraño en el hogar, y en alta frecuencia cuando se les deja aislados. Por lo que no es extraño que los humanos puedan reconocer algunas de estas características y relacionarlas con estados emocionales del perro. Lo anterior puede deberse a que los humanos cuando están bajo la influencia de cierta emoción también emiten variaciones acústicas similares al contexto que se presenta.

Molnár *et al.* (2009) estudiaron las propiedades acústicas del ladrido y encontraron ciertos patrones relevantes. La señal acústica tiene componentes regulares e irregulares, como lo es la relación armónico-ruido (HNR), con la que se ha conseguido ordenar expresiones vocales de mayor a menor ruido. Se demostró que las personas pueden identificar mejor los ladridos de perro cuando se escuchan con una relación armónico-ruido baja.

En los trabajos anteriores se buscaron encontrar propiedades de la señal de audio que sirvan para reconocer ladridos y una caracterización que tenga los mejores atributos para representar la información relacionada con el perro y su estado.

A continuación se introducen los trabajos previos que analizan los ladridos aplicando técnicas de aprendizaje automático. Estos trabajos se utilizaron como referencia para comparar los resultados de los experimentos realizados. También se usaron como recursos bibliográficos que sirven de base para la investigación.

El trabajo realizado por Molnár *et al.* (2008) utiliza aprendizaje automático para diferenciar los ladridos registrados por una sola raza de perro, el Mudi húngaro. Este perro de pastoreo tiene estatura media, alrededor de 42–46 cm, con peso promedio entre 12–15 kg, y al cual se grabó en siete contextos: ladridos ante un extraño, durante entrenamiento de pelea, mientras el dueño se prepara para llevarlo a una caminata, cuándo se le deja solo atado a un árbol, al pedir una pelota, cuándo se le muestra un plato de comida y mientras se juega con él.

En su base de datos se registraron 6615 ladridos en contextos conocidos correspondientes a individuos conocidos. Posteriormente, su sistema selecciona el conjunto de características (un pequeño conjunto de variables acústicas) con la que se obtiene la

mejor clasificación. Esta base de datos es la más utilizada dentro de la literatura para este campo de investigación, por lo cual también se utiliza en esta tesis. También sirve de referencia para comparar los resultados de los experimentos que se realizaron.

Su algoritmo clasifica los ladridos de prueba en los contextos correctos con una exactitud del 43 % sobre el conjunto de prueba. Los datos en este conjunto de prueba corresponden a los mismos perros utilizando un tercio de la base de datos. Donde el conjunto restante es el de entrenamiento. En este trabajo se utilizó una caracterización diferente y un clasificador de Naive-Bayes con configuración de hiper-parámetros optimizada, pero similar. Su algoritmo identificó a los perros individuales con una exactitud del 52 %.

Su sistema fue más exitoso en la clasificación de contextos de ladridos ante la presencia de un extraño y en el entrenamiento de pelea. En el reconocimiento de los perros individuales, los ladridos correspondientes a las clases ’Extraño’ y ’Pelea’ fueron las más difíciles de clasificar y el éxito de la categorización fue mayor para los ladridos emitidos en otros contextos.

A partir de este trabajo, Pérez *et al* (2016). utiliza la misma base de datos de perros Mudi y clasifica los perros por contexto del ladrido consiguiendo un 74 % de exactitud con el algoritmo de máquina de soporte vectorial. En su trabajo propuso evaluar un conjunto de características acústicas que se conocen como los descriptores acústicos de bajo nivel, los cuales se agrupan por categorías que son: espectrograma de Mel ’Melspec’, coeficiente cepstrales de frecuencia de Mel ’MFCCs’, energía de la señal (energy), bandas espectrales, desvanecimiento espectral, rodaje espectral (spec roll-off), flujo espectral, centroide espectral, posición mínima y máxima espectral, probabilidad de vocalización, tasa de cruces por cero, frecuencia fundamental y envolvente. Todas las características anteriores se extraen utilizando el software ’OpenSmile’. Dicho software ya cuenta con algoritmos optimizados para obtener estos atributos.

En sus resultados, al comparar diversos grupos de descriptores acústicos de bajo nivel, sobresalen en primer lugar los ‘MFCCs’, en segundo ‘Melspec’ y en tercero ‘energía de la señal’. Por cuestiones prácticas, sólo se discuten estos tres por el momento. En muchos otros trabajos relacionados al análisis de audio, estos atributos se trabajan por separado, obteniendo buenos resultados. Los descriptores acústicos se han utili-

zado previamente en modelos para-lingüísticos de información del habla humana y se demostró que pueden aplicarse satisfactoriamente en el análisis de ladridos.

En el trabajo de Larrañaga *et al.* (2015) se analizaron los ladridos de perro Mudi y se investigó sobre su rol de comunicación intraespecífico. Este estudio compara la cuatro métodos de aprendizaje automático diferentes que son: Naive Bayes, árboles de clasificación, k-vecinos cercanos y regresión logística. En este estudio se clasificó a los perros Mudi por sexo, edad, contexto e individuo. Los resultados de exactitud en la predicción se consiguieron con una validación de pliegues cruzada. Se alcanzó un 85.13 % de exactitud al determinar el sexo, 80.25 % al predecir la edad, 55.5 % en siete contextos y 67.63 % al reconocer 8 individuos. Donde el mejor clasificador fue el de los k-vecinos más cercanos.

El punto de partida del presente trabajo de investigación se basa en el artículo de Pérez *et al* (2016). En dicho trabajo, ellos utilizaron una red neuronal convolucional como técnica para afinar sus parámetros. Dicha técnica se compara con otros algoritmos de aprendizaje automático, junto con la caracterización mediante descriptores acústicos de bajo nivel. Ellos introdujeron una base de datos para la clasificación, la cual consta de 37 perros, grabada por la empresa mexicana ‘Mescalina’ en el 2015. Esta base de datos contiene ladridos de diferentes razas de perros (Chihuahua, Schnauzer, Labrador, Poodle y otros). Además, contiene cuatro clases de ladridos inducidos en diferentes contextos: muy agresivo, agresivo, juguetón y otros.

En la parte de clasificación, ellos utilizaron algunos algoritmos de aprendizaje automático como lo son: máquina de soporte vectorial, bosque aleatorio, C45 y otros. Estos algoritmos están implementados en el software ‘Weka’ especializado para el aprendizaje automático, desarrollado en la Universidad de Waikato.

Este trabajo alcanzó una exactitud de 79 % en el conjunto de prueba con el algoritmo de bosque aleatorio. Estos resultados superan a estudios previos como el de Yin *et al* (2004)., en el que se alcanzó 50.4 % de exactitud; (Molnár *et al.*, 2008) obtuvieron 52 % de exactitud sobre el conjunto de prueba con un método para generar características llamado Greedy Stepwise y un clasificador Bayesiano.

Los trabajos anteriores resumen los esfuerzos que se han hecho para llegar al estado del arte en el problema del reconocimiento de ladridos. Estos trabajos se utilizan

como cimientos de la presente investigación, ya que se usan las mismas herramientas y técnicas realizadas en esos trabajos. En esta tesis se estudian nuevas estrategias. Primero se evalúan distintas bases de datos en los sistemas de los trabajos previos que les hacía falta explorar su desempeño en otros conjuntos de datos, incluyendo todas las tareas. Después se analizan las distintas maneras de representar la información de la señal de audio, con lo que se busca replantear las ventajas y desventajas que ofrecen. Finalmente, se exploran diferentes arquitecturas de redes neuronales profundas existentes con el fin de averiguar cuál se adapta mejor a este problema.

1.3. Objetivos de la investigación

El objetivo general de este trabajo de investigación es crear una herramienta de clasificación competente y viable de implementar que clasifique ladridos con base en aprendizaje profundo. Esta herramienta debe aprovechar las características acústicas del ladrido y mejorar los resultados en comparación a otros métodos de clasificación de ladridos reportados previamente.

1.3.1. Objetivos específicos

1. Implementar y comprobar los resultados de trabajos previos que utilizan las técnicas siguientes: máquina de soporte vectorial, bosque aleatorio y C45. se utilizaron descriptores acústicos como base de comparación para las tareas de clasificación.
2. Evaluar distintos métodos de generación de características acústicas.
3. Identificar las características óptimas para mejorar la clasificación.
4. Implementar la arquitectura de red neuronal más pertinente para resolver las tareas de clasificación.
5. Desarrollar un método para encontrar la mejor combinación de parámetros de afinación para red neuronal profunda.

6. Generar modelos que, por medio de la fusión de diferentes bases de datos, incrementen la cantidad de datos, para cada tarea.
7. Combinar todas las etapas para cada modelo y obtener los resultados de predicción para cada uno de estos modelos.
8. Obtener una tabla comparativa del desempeño del método de caracterización y clasificación propuesto contra los demás métodos y bases de datos.

Capítulo 2. Marco Teórico

Este capítulo explica el sustento teórico y conceptual del presente trabajo de investigación. Primero, se describe el preprocesamiento de los datos con los que se trabaja; para este caso, son los ladridos emitidos por los perros.

En esta parte se explican las formas de representar la información contenida en el audio. Estas representaciones se consiguen a través de distintos métodos. Sólo se mencionan los que se utilizaron en este trabajo y se describen las ventajas y desventajas que pueden llegarse a presentar en distintos escenarios.

En la segunda parte se utilizan algoritmos que corresponden al campo del aprendizaje automático. Principalmente se discuten las redes neuronales, así como otros algoritmos que se utilizan de referencia para comparar el desempeño y de apoyo al analizar los patrones que existen en los datos a utilizar. Debido a que estos algoritmos presentan distintos parámetros con los que se trabaja, es importante comprender su funcionamiento. Así que, dependiendo de la naturaleza de los datos y del problema, se ajusta su configuración.

Por último, se describe el modelo de Russell y James (1980). Este modelo ayuda a conceptualizar, interpretar y medir emociones, para la tarea de clasificación de contextos.

2.1. Procesamiento digital de señales de audio

El procesamiento de audio abarca varios campos involucrados en representar los sonidos escuchados por el humano. Gran parte del procesamiento digital se refiere a la computación y matemáticas realizada por algoritmos que se aplican a la señal de audio, además de manipular la información contenida en los datos para cierto fin.

A continuación se resumen algunos métodos utilizados para representar el audio y la información que contiene. Estos métodos forman parte de la etapa de caracterización del audio.

2.1.1. Señal de audio cruda

El sonido se captura mediante un dispositivo electrónico; el cual puede ser cualquier dispositivo de grabación que tenga micrófono. El audio es la representación digital del sonido, el cual es una señal analógica acústica y representa el movimiento físico de las partículas en el aire o medio. El sensor capta variaciones de presión en el aire y las convierte a una señal digital por medio de un convertidor analógico digital (ADC).

De forma inversa, una señal de audio digital se puede convertir a analógica por medio de un convertidor digital analógico (DAC). Después, la señal analógica se amplifica con la energía necesaria para activar una bocina de manera que se pueda escuchar.

La señal digitalizada se captura en el ADC con cierta velocidad de muestreo. Cada medida analógica de voltaje de la señal queda registrada con un número binario determinado durante cada fracción de tiempo. La frecuencia de muestreo se mide en Hertz (Hz) y depende del ancho de banda de la señal de entrada.

En el proceso de captura del sonido, lo ideal es no tener ruido de fondo; el sonido puede venir de múltiples direcciones y para recibir un volumen balanceado se necesita tener un conjunto de micrófonos en distintas posiciones a la distancia correcta; también es bueno evitar que el ambiente genere eco. Todo ésto con el propósito de que se capture la esencia del sonido individual, aislado de factores externos que puedan interferir. Aunque lograr estas condiciones es difícil, al lograrlo, se evita el uso de técnicas exhaustivas de procesamiento de audio. Para corregir el audio se pueden usar filtros y ecualizadores con el fin de eliminar particularidades de la grabación que estorban o simplemente no es necesario tener presentes.

En el proceso grabación es importante tomar en cuenta el teorema fundamental de muestreo. El teorema de Nyquist Shannon (1949) demuestra que es matemáticamente posible reconstruir una señal periódica con base en sus muestras, siempre y cuando la tasa de muestreo sea superior al doble de la máxima frecuencia en la señal de entrada. Esto se asegura que no se pierda información de la señal. En esta investigación se trabajó con archivos con frecuencia de muestreo 44100 Hz o mayores. Después se propuso utilizar 8820 Hz para reducir el tamaño de los archivos de audio. Basándose

en el teorema de muestreo, si se utilizan 8820 Hz como frecuencia de muestreo, la frecuencia máxima de la senal sería de 4410 Hz. Note que el ancho de banda del ladrido es menor a 3000Hz, de esta forma se conservan todas las frecuencias menores a este valor. En el trabajo de (Sakamoto *et al.*, 2014) se observa que los ladridos de distintos perros se encuentran en el intervalo de frecuencia de 1000 Hz a 2000 Hz en su mayoría. También en el trabajo de (Frommolt, 2004) se registraron chillidos de perros que varían en un intervalo de frecuencia de 4000 a 5000 Hz. No obstante, sólo se utilizan ladridos para estos experimentos.

El audio crudo es la representación más básica de la señal de audio en bits; sin embargo, para poder trabajar con ella y visualizarla es aconsejable convertirla en un formato decimal. Por esta razón se utiliza el formato WAVE, cuyos archivo tienen extensión .wav. Este formato de archivo de audio digital no tiene pérdida de calidad, generalmente se graba a 44.1KHz y por cada minuto de grabación se ocupan unos 10 megabytes. En consecuencia, se puede llegar a ocupar mucho espacio en memoria cuando se trabaja con muchos archivos al mismo tiempo. De las representaciones de audio que se mencionan y se utilizan en este trabajo, ésta es la más difícil y lenta de procesar, ya que no se ha aplicado ningún método que comprima o reduzca la información. Por esta razón también es versátil, ya que no hay que realizar un preprocesamiento anterior a la clasificación. Lo anterior significa que cuando se captura la señal en tiempo real, ésta se procesa de manera más rápida sin ningún retardo.

La señal de audio crudo representa la amplitud del sonido en cada instante. Si estos datos se representan de forma gráfica, se puede visualizar la forma de onda, picos, ruido y pausas, entre otras características. Trabajar con la señal de audio cruda es la forma más directa de trabajar con el sonido; puede ser complicado debido a que visualmente contiene muchos detalles que son difíciles de apreciar para el ojo humano. Existen muchas otras características del audio crudo que pueden ser de interés y algunas de éstas no han sido aún exploradas.

2.1.2. Técnicas de preprocesamiento

En el análisis de audio inteligente, después de capturar y convertir la señal en audio crudo, se determina si requiere alguna técnica de preprocesamiento, según se necesite para su finalidad. Algunas de las técnicas utilizadas en este trabajo son las siguientes:

1. Segmentación: se corta un audio de larga duración en partes más cortas, generando nuevos archivos de audio. Se puede realizar la segmentación por interés en cierto evento que contiene la señal o simplemente por la duración que se necesita.

2. Normalización: para evaluar y comparar los datos, se debe manejar una misma escala. Ya que las grabaciones de audio pueden variar en su nivel de volumen, se busca normalizar para todas las muestras de audio; Para lograrlo se aplica la relación dada por la ecuación 1:

$$x_{normalizada} = \frac{x_n - x_{min}}{x_{max} - x_{min}} \tag{1}$$

   Donde x_{max} representa el valor más alto dentro del vector y x_{min} el valor más bajo que se haya encontrado. Lo anterior se aplica para cada valor de x_n en el vector de datos. El resultado es la misma señal ya normalizada dentro de un intervalo unitario de 1 a 0.

3. Reducción de frecuencia de muestreo: este proceso toma muestras de la señal original para generar la señal reducida. Se utiliza en caso de que se necesite reducir el tamaño del archivo de audio. Se usa una tasa de muestreo menor que sea mayor al doble de la frecuencia máxima. Así se evita que se pierda información de la señal y se puede realizar el suavizado de contorno de la señal correctamente.

2.1.3. Caracterización

Los algoritmos inteligentes de análisis de audio buscan información relevante en la señal de audio que ayude en las tareas de reconocimiento, algunos ejemplos son: la emoción, el estilo de conversación, tipo de música, el estado de ánimo, los instrumentos, la progresión de acordes musicales o distintos eventos de sonido. En general, estas propiedades se caracterizan por la dinámica de la señal en el tiempo.

Se busca facilitar el trabajo del clasificador dándole características relevantes con las que se puedan distinguir propiedades de la señal de audio exitosamente. Algunas técnicas para caracterizar el audio que se utilizan en este trabajo son las siguientes:

2.1.3.1. Espectrograma y transformada de Fourier

Las señales de audio se componen de una combinación de sonidos. Cada uno de estos sonidos vibra a distinta frecuencia que le da la nota o tono musical que se escucha. Estas frecuencias se encuentran encimadas entre sí. También, cada una puede tener diferente volumen que se puede visualizar en la amplitud de la onda.

Por otro lado la transformada de Fourier es una transformación matemática empleada para transformar señales desde el dominio del tiempo hacia el dominio de la frecuencia. La operación de transformación produce la función que representa todas las frecuencias del tiempo durante el cual existió la señal.

Al aplicar la transformada de Fourier sobre una señal de audio que se encuentra en el dominio del tiempo, gráficamente se obtiene la representación de la señal en el dominio de la frecuencia. Asi se puede visualizar la energía contenida en cada frecuencia. En este dominio, un eje representa la escala de frecuencia y otro la amplitud.

El espectrograma es la representación tridimensional que resulta al procesar la información contenida en una ventana de tiempo de la señal de audio. En esta representación, un eje corresponde al tiempo, otro al de la frecuencia, y por último una escala de color para representar la intensidad de la energía.

Esta representación contiene patrones visuales que se utilizan para interpretar

comportamiento o características de los sonidos. Algunos de sus usos son: identificar sonidos fonéticos y procesado del habla, en telecomunicaciones, radares y en las redes neuronales como representación del audio de entrada.

2.1.3.2. Coeficientes cepstrales en escala de Mel

Estos coeficientes se usan para representar de manera más adecuada los valores de la señal de audio respecto a la percepción auditiva del humano. Se usan mucho en el área de reconocimiento de audio automático ya que poseen las características componentes de la señal de audio, lo cual es útil en la caracterización.

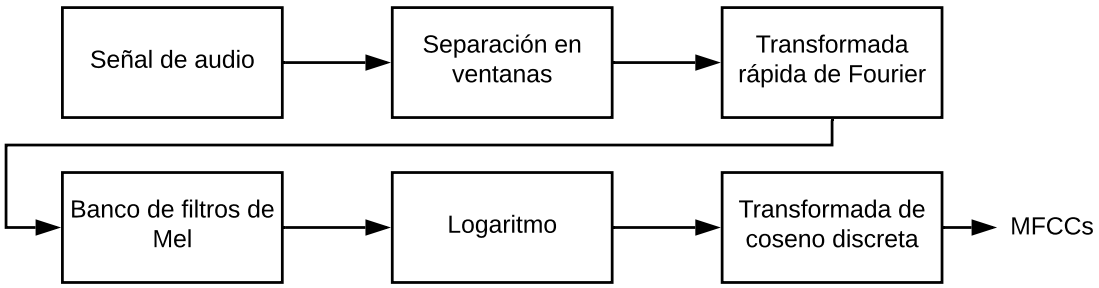

Figura 1. Diagrama de pasos para obtener los coeficientes cepstrales en las frecuencias de Mel

Para calcular los coeficientes cepstrales en escala de Mel (MFCCs) se utiliza el siguiente procedimiento (ver la figura 1).

1. Se divide la señal en pequeños tramos que se les conoce como ventanas.
2. A cada tramo o ventana se le aplica la transformada de Fourier discreta, con lo que se obtiene la potencia espectral de la señal.
3. Se aplica un banco de filtros triangulares donde los centros están esparcidos a lo largo de la escala de Mel.
4. A cada uno de los espectros resultantes se le suma la energía de la señal contenida en el ancho de banda de cada filtro.
5. Se obtiene el logaritmo de todas las energías de cada frecuencia de Mel.

6. Se aplica la transformada de coseno discreta a los resultados previos.
7. Se obtiene el resultado final estos valores son los coeficientes que representan a los MFCCs.

Es común que estas características se usen en sistemas de reconocimiento del habla. Cada vez se les otorga más importancia y se cree que son las características fundamentales que representan mejor a la voz humana. Se ha descubierto que también se pueden usar en otras aplicaciones como en el campo de la recuperación de información musical, la clasificación de géneros, y como medida de similitud de audio.

Una desventaja que presentan estos coeficientes es que su sensibilidad al ruido es alta. Los valores de los MFCCs no son muy robustos ante la presencia de ruido aditivo, por ello es común la normalización de los valores en los sistemas de reconocimiento de voz para reducir la influencia de dicho ruido.

2.1.3.3. Grupos de descriptores acústicos

Se le llama descriptores acústicos al conjunto de características distintas de la señal que se pueden extraer mediante diferentes técnicas. Existe un solo método que puede proporcionar diversas sub-características que pueden ser útiles.

A continuación se describen algunas características del audio que se pueden extraer al analizar el audio crudo. Estas características se extraen individualmente de cada una de las ventanas en la que se divide la longitud del audio.

1. Melspec: Se obtiene al aplicar un banco de filtros triangulares equidistantes en la escala de Mel sobre el espectro de frecuencias.
2. MFCC: son los primeros 12 coeficientes cepstrales de frecuencia de Mel en la banda crítica del espectro.
3. Energía: es la representación del cálculo de energía de señal logarítmica y raíz media cuadrada (RMS).

4. Bandas espectrales: contienen la energía dada por cierta banda espectral al realizar una sumatoria de todas las ventanas obtenidas por la transformada rápida de Fourier en esa banda.
5. Barrido espectral: resulta de la frecuencia de caída dentro del espectro. Se puede utilizar para distinguir entre sonidos de armónicos.
6. Probabilidad de vocalización: se calcula un porcentaje estimado de la energía para cada armónico; después de un umbral establecido, se determina si existe vocalización o no.
7. F0: se refiere a la frecuencia fundamental definida como la frecuencia más baja en la forma de onda de la señal donde se produce el primer armónico.
8. Envolvente F0: suavizado del decaimiento exponencial de la frecuencia fundamental.
9. ZCR: representa la velocidad a la que se producen los cruces por cero. Es una medida simple del contenido de frecuencias en una señal.

2.2. Aprendizaje automático

En el campo de las ciencias de la computación existe la rama de la inteligencia artificial, la cual está inspirada en la capacidad de la mente humana por aprender, adaptarse y resolver problemas. Ya que en la práctica, la inteligencia humana como tal no se puede recrear. Se le llama inteligencia artificial al proceso por el cual una máquina imita alguna de las funciones cognitivas de la mente humana.

El aprendizaje automático es una rama de la inteligencia artificial que se orienta hacia el desarrollo de técnicas de aprendizaje para computadoras. Estos algoritmos son capaces de interpretar los datos que reciben y, en el proceso de evaluar la información, se podría decir que mejoran con la experiencia. Tienen la finalidad de generalizar un comportamiento o ser capaces de inferir una decisión.

El aprendizaje de máquina se basa en el análisis de datos. La inferencia estadística se involucra en gran parte ya que por medio de algunos métodos y procedimientos

se determinan algunas propiedades estadísticas para obtener conclusiones y modelar patrones.

Existe una gran cantidad de algoritmos de aprendizaje de máquina disponible. Cada algoritmo pertenece a una clase o grupo que se puede identificar por su estilo de aprendizaje y similitud de acuerdo a su funcionamiento.

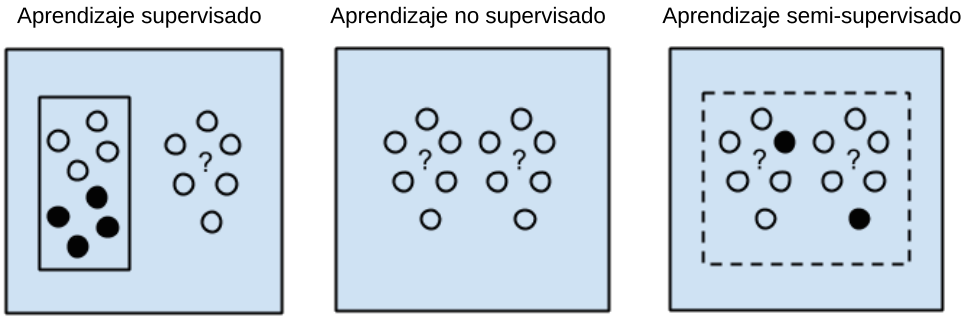

Figura 2. Tipos de aprendizaje. Fuente: (Jason Brownlee, November 25, 2013, A Tour of Machine Learning Algorithms in Understand Machine Learning Algorithm)

En la Figura 2 se puede ver que existen tres tipos de grupos que se dividen por su distinta forma de aprendizaje sobre los datos:

1. Aprendizaje supervisado: los datos que se utilizan se llaman conjunto de entrenamiento y la información está etiquetada con la categoría correspondiente. El modelo se prepara a través de un proceso de entrenamiento en el que se requiere hacer una predicción. Luego se evalúa si esta predicción es correcta o no. El proceso de entrenamiento se repite hasta que el modelo logra clasificar, a un nivel de precisión adecuada, los datos que se utilizaron.

2. Aprendizaje no supervisado: los datos de entrada no están etiquetados y no se conoce su resultado. El modelo se prepara para deducir las estructuras generales presentes en los datos. Lo anterior se logra a través de métodos matemáticos para reducir la redundancia o organizándolos por sus similaridades.

3. Aprendizaje semi-supervisado: los datos de entrada son una combinación de datos etiquetados y sin etiqueta. Se desea resolver el problema de predicción, pero el modelo debe aprender las estructuras para organizar los datos e inferir los

resultados de predicción.

Los algoritmos de aprendizaje de máquina también se pueden agrupar por su funcionamiento. A pesar de que existen algoritmos que pueden pertenecer a múltiples categorías, las técnicas que utilizan estos algoritmos son bastante distinguibles entre sí, aunque no se esté familiarizado con el funcionamiento en detalle. En la figura 3 se muestran los diversos grupos de algoritmos de aprendizaje de máquina. Debido que algunos de estos algoritmos están relacionados a la lectura, se mencionan los ejemplo más relevantes y utilizados, además de una breve y simple descripción.

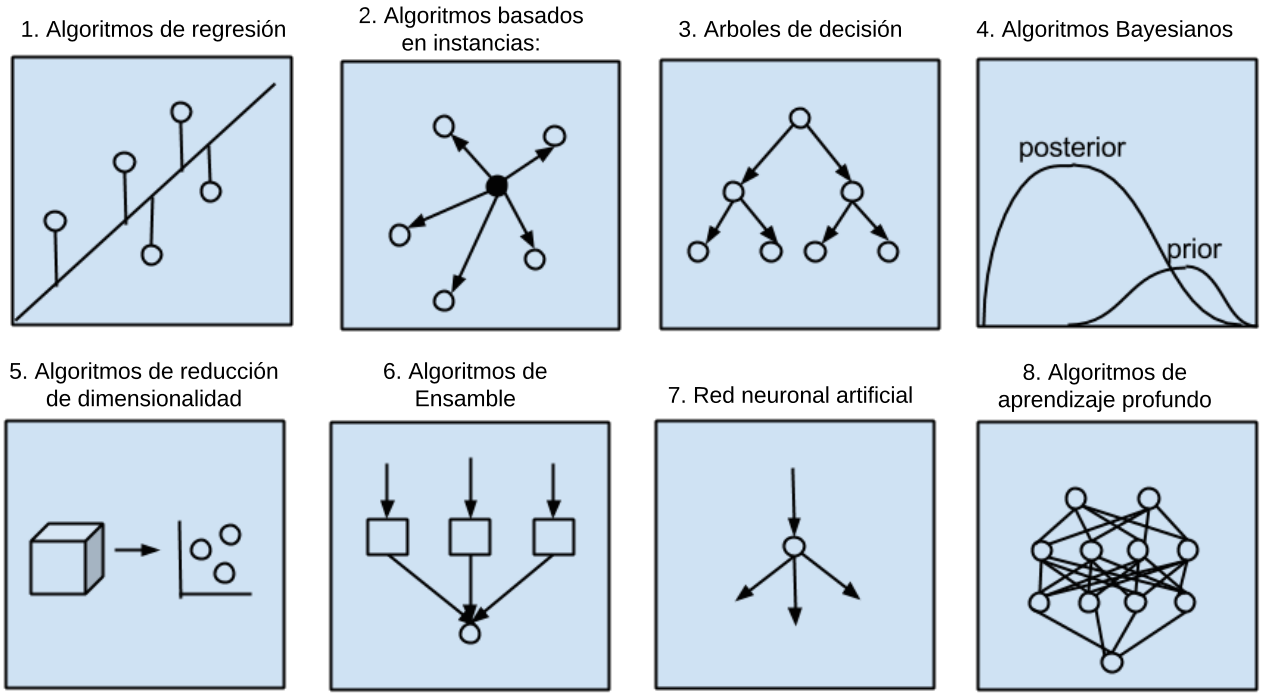

Figura 3. Grupos de algoritmos. Fuente: (Jason Brownlee, November 25, 2013, A Tour of Machine Learning Algorithms in Understand Machine Learning Algorithm).

1. Algoritmos de regresión: utilizan la relación entre variables e iterativamente refinan la medida del error que predice el modelo. Entre los más populares están la regresión lineal, la máquina de soporte vectorial y la regresión logística.

2. Algoritmos basados en instancias: estos algoritmos contienen ejemplos de datos para comparar contra nuevos datos usando una medida de similitud y buscan

encontrar la mejor coincidencia para hacer la predicción. Algunos ejemplos son el mapa auto organizado y los *k* vecinos más próximos.

3. Árboles de decisión: construyen un modelo basado en decisiones de acuerdo al valor actual de los atributos que presentan los datos y su estructura se ramifica hasta una predicción o decisión dada. Los ejemplos más comunes son los árboles de decisión condicional, C4.5 y CART.

4. Algoritmos Bayesianos: En estos métodos se aplica el teorema de Bayes para problemas de clasificación y regresión. Algunos ejemplos son el clasificador de Naive Bayes, La red Bayesiana y el clasificador Gaussiano de Naive Bayes.

5. Algoritmos de reducción de dimensionalidad: son parecidos a los métodos de agrupamientos pero buscan reducir la cantidad de información y explotar la estructura inherente en los datos, en este caso, de una maneras no supervisada para describir los datos usando menos información. Los más comunes son el PCA, t-SNE y MDS.

6. Algoritmos de ensamble: son métodos compuestos de modelos débilmente entrenados independientemente y que sus predicciones en combinación suman una predicción total. Algunos ejemplos son el bosque aleatorio y el bootstrap.

7. Red neuronal artificial: este modelo está inspirado en la función biológica de las neuronas. Mediante las neuronas de entrada se alimentan los datos que se conectan a las demás neuronas. Durante el entrenamiento, se ajusta el peso relacionado a la decisión del patrón de neuronas de entrada que activa mayormente a la salida que corresponde. El ejemplo más simple corresponde al del perceptrón multicapa.

8. Algoritmos de aprendizaje profundo: Son una combinación de distintas redes neuronales artificiales con mayor complejidad y tamaño que explotan el poder de cómputo y gran tamaño de los datos de entrada.

En esta investigación se utiliza principalmente el aprendizaje supervisado para el clasificador, ya que los algoritmos de redes neuronales profundas requieren de datos supervisados de entrenamiento con etiquetas ya definidas. También, se utilizaron en la etapa de caracterización algoritmos basados en instancias para evaluar los atributos

más importantes mediante el algoritmo de selección de alivio (relief). Sin embargo, también se trabajó con algoritmos de decisión, de regresión lineal y reducción de dimensionalidad para otras tareas como: comparar desempeño, analizar la complejidad de los datos y buscar patrones en la información.

2.2.1. Redes neuronales artificiales y aprendizaje profundo

Actualmente el aprendizaje profundo ha sido sobresaliente entre todos los algoritmos de aprendizaje automático, debido a que logra una exactitud superior en tareas importantes, como lo son el reconocimiento de imágenes, texto, voz y lenguaje, entre otras. Comparado con los antiguos métodos de aprendizaje de máquina, existe una notable mejora en el desempeño al aumentar la cantidad de datos disponibles para el entrenamiento.

El aprendizaje profundo es una rama del aprendizaje de máquina basado completamente en redes neuronales. A diferencia de una red neuronal tradicional, la cual sólo contiene dos o tres capas ocultas, las redes profundas pueden llegar a tener 10 capas o más, junto a diferentes arquitecturas de redes implementadas. Esta serie de capas con distintas arquitecturas es lo que hace alusión al concepto de profundidad.

Lo que hace llamativo a estos métodos es que aprenden directamente a partir de los datos, sin necesidad de una extracción manual de características, aunque ésto no es siempre el caso para todos los problemas.

2.2.1.1. Estructura de una red neuronal

Para comprender cómo se construyen estos modelos, se debe tener claro su concepto. Una red neuronal es un modelo matemático inspirado en el funcionamiento biológico de las neuronas; sin embargo, es solamente una analogía a la estructura básica que presentan.

Primero se debe hablar de una neurona, la cual es una unidad básica de la red

neuronal. La neurona recibe entradas, realiza una operación con los datos recibidos y produce una salida. La figura 4 presenta un esquema funcional de una neurona.

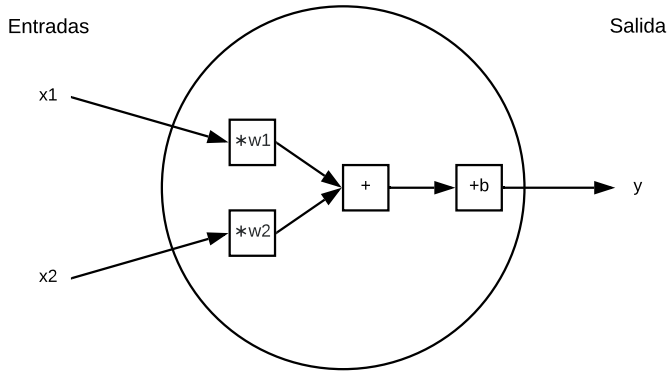

Figura 4. Operación básica de una neurona. Fuente: Victor Zhou Machine Learning for Beginners: An Introduction to Neural Networks

En el ejemplo de la figura 4, la neurona recibe dos valores de entrada que se multiplica por un peso *w*. Después se pesan todas las entradas y se le agrega un factor de parcialidad *b*. Por último se suma el total y el resultado y se obtiene de evaluar la función de activación. Como se muestra en la ecuacion 2.

$$
\begin{aligned}
x_1 &\longrightarrow x_1 * w_1 \\
x_2 &\longrightarrow x_2 * w_2 \\
(x_1 * w_1) + (x_2 * w_2) + b \\
y = f(x_1 * w_1 + x_2 * w_2 + b)
\end{aligned}
\tag{2}
$$

La función de activación se usa para convertir la entrada en una salida delimitada. Existe una variedad de funciones que se utilizan en diferentes situaciones, algunas de ellas son:

1. Sigmoide: esta función transforma los valores que se introducen a una escala dentro del intervalo (0,1), donde los valores altos tienden, de forma asintótica, a uno y los valores muy bajos tienden, de manera asintótica, a cero. Esta función no está centrada en cero, por lo que se utiliza casi siempre en la última capa por

su buen rendimiento. La función sigmoide se define en la ecuación 3.

$$f(x) = \frac{1}{1 - e^{-x}} \tag{3}$$

2. Tanh: la función tangente hiperbólica toma los valores de la entrada y los evalúa en una escala con intervalo (-1,1), donde los valores altos tienden de manera asintótica, a uno y los valores muy bajos tienden de manera asintótica, a menos uno.

   Se utiliza comúnmente en redes recurrentes, ya que sirve para decidir entre una opción o la contraria. La función tangente hiperbólica se define en la ecuación 4.

$$f(x) = \frac{2}{1 - e^{-2x}} - 1 \tag{4}$$

3. ReLU: la función de unidad lineal rectificada (rectificadora) transforma los valores linealmente anulando los valores negativos y dejando los positivos tal y como entran. Sólo se activa con positivos, no está acotada y tiene buen desempeño en redes convolutivas con imágenes. La función ReLU se define en la ecuación 5.

$$f(x) = max(0, x) = \begin{Bmatrix} 0 \leftarrow x < 0 \\ x \leftarrow x \geq 0 \end{Bmatrix} \tag{5}$$

4. Softmax: esta función transforma las salidas a una representación de probabilidades, de forma que el resultado del sumatorio de todas las probabilidades de las salidas es uno. Se utiliza cuando se quiere tener una representación en forma de probabilidades y para normalizar múltiples clases. La función softmax se define en la ecuación 6.

$$f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{6}$$

Una red neuronal típica se compone desde un par de neuronas hasta miles que se conectan entre sí. Estas neuronas se organizan dentro de la red en una serie de capas, cada una conectada entre sí. En la figura 5 se muestra una red neuronal simple y una red neuronal profunda compuesta por una mayor número de capas que también puede estar compuesta por diferentes tipos de neuronas y arquitecturas.

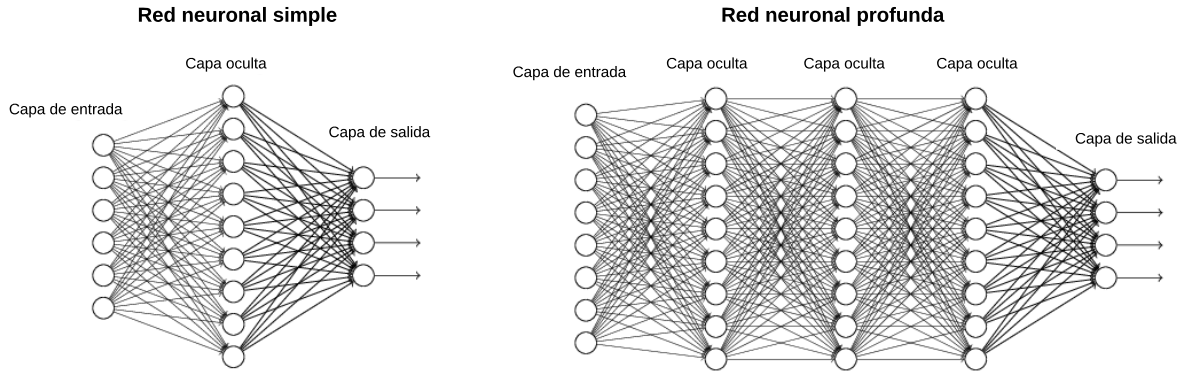

Figura 5. Capas en redes neuronales

La red comienza con la primera capa o capa de entrada, la cual recibe la información de las muestras. Generalmente está construida con la misma cantidad de neuronas que presenta el tamaño de las muestras en los datos de entrada.

Del otro lado se localiza la última capa de salida, la cual representa el número de neuronas que corresponden al número de categorías que se presentan en el problema. Es en esta capa donde se obtiene el resultado de la predicción.

Opcionalmente se puede incluir una o más capas ocultas que son las capas que se conectan entre las capas de entrada y salida. La cantidad de neuronas que tienen estas capas y el número de capas que se eligen, son hiper-parámetros que se deben ajustar dependiendo del problema que se tenga.

2.2.1.2. Entrenamiento de una red neuronal

Para el entrenamiento de la red neuronal es esencial comprender las dos etapas que existen la propagación hacia delante y la retro-propagación. Al momento de iniciar el entrenamiento de una red neuronal, los valores de los pesos de cada conexión entre neuronas se inicia aleatoriamente. La función de activación es la que determina el valor de activación de cada neurona en la red neuronal.

En la propagación hacia adelante, la información de una capa pasa a la siguiente. En cada neurona se suman los valores de los pesos anteriores y se calcula la función de activación generando una salida con el valor que tendrá el peso en la siguiente capa. La retro-propagación sucede después de haber alimentando a la red neuronal con un lote de entrada completo.

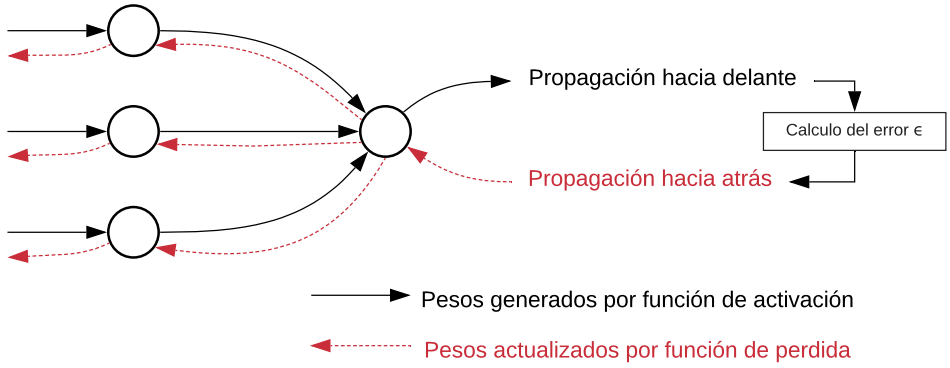

Figura 6. Flujo de señales de activación y de propagación del error

Al momento de calcular el error en la última capa, comienza el proceso de ajustar los pesos de la red neuronal en función de la tasa de error, es decir, la pérdida obtenida en la época anterior, es decir, la iteración. Después, dependiendo de la función de optimización que se utilice, se determina en qué dirección se deben ajustar los pesos para obtener una pérdida menor que la que calculó la función de pérdida, la cual va ligada directamente al optimizador que se utilizó. La figura 6 muestra el diagrama que ilustra el flujo de la señales en la red.

El optimizador implementa el algoritmo de propagación de errores y proporciona un método eficiente para calcular las derivadas del error para cada peso. El ajuste adecuado de los pesos garantiza tasas de error más bajas, lo que hace que el modelo sea confiable al aumentar su generalización.

2.2.1.3. Hiper-parámetros

Los hiper-parámetros son valores que se proponen antes de que el algoritmo inicie su proceso de aprendizaje. Éstos se dividen en los hiper-parámetros referentes a

la arquitectura del modelo y los hiper-parámetros relacionados al entrenamiento del modelo. Algunos de estos valores determinan qué tanto tiempo tomará al algoritmo en terminar y pueden ser sutiles, en ciertos casos es difícil saber cómo influyen. Para seleccionar los valores óptimos de estos hiper-parámetros se requiere tener cierto criterio, que se adquiere a través de prueba y error, ya que cada problema puede tener sus propias variaciones.

Sin embargo, existen algunas técnicas que se pueden usar como: búsqueda por cuadrícula y búsqueda aleatoria. No obstante, si el espacio de búsqueda es demasiado grande, ésto puede ser muy tardado, por lo cual se requiere establecer los límites de la búsqueda. Con la experiencia se va formando un criterio para determinar cuáles hiper-parámetros y qué intervalo de valores ayudan a tener buenos resultados. A continuación se introducen lo hiper-parámetros más comunes.

Hiper-parámetros relacionados a la arquitectura de la red neuronal:

1. Cantidad de capas: una red neuronal sin capas ocultas puede resolver fácilmente un problema de clasificación donde las dos clases se pueden separar linealmente. En la mayoría de los problemas en los que se tiene interés de resolver con redes neuronales, son casos que no son linealmente separables. Por esta razón aumentar la cantidad de capas, ayuda a separar las clases más complejamente separadas.

2. Cantidad de neuronas: el uso de muy pocas neuronas en las capas ocultas da lugar a lo que se denomina insuficiencia. La insuficiencia impide detectar adecuadamente las señales en un conjunto de datos complicado. El uso de demasiadas neuronas en las capas ocultas puede dar lugar a varios problemas. Primero, pueden resultar en un sobreajuste. Un consejo común es que el número de neuronas ocultas debe ser inferior al doble del tamaño de la capa de entrada y no menor al número de neuronas en la capa de salida.

Parámetros relacionados al entrenamiento:

1. Épocas (epochs): es el número de veces que el conjunto de entrenamiento com-

pleto se muestra a la red neuronal mientras entrena.

2. Tamaño de lote (batch size): es el número que define cuántas muestras tiene el subconjunto que se alimenta a la red. Ésto se repite hasta terminar de pasar todo el conjunto de entrenamiento. Un tamaño pequeño del lote conducirá a un comportamiento estocástico; en cambio, un tamaño de lote grande afectará los requisitos de memoria y el tiempo de cálculo. El tamaño de lote es mayormente una cuestión computacional por lo que generalmente se maneja en potencias de dos.

3. Optimizador (optimizer): existe una cantidad de algoritmos distintos que optimizan el proceso en que los valores de los pesos y la parcialidad se van actualizando. Estos algoritmos buscan disminuir la función de error o maximizar la función objetivo. Usando valores del gradiente con respecto a los parámetros, la derivada indica si la función va incrementando o disminuyendo en cierto punto con una línea tangente en su superficie. El optimizador define la función de pérdida u objetivo a utilizar y su comportamiento depende de la tasa de aprendizaje

4. Tasa de aprendizaje (learning rate): La tasa de aprendizaje determina qué tan rápidamente las actualizaciones del gradiente siguen la dirección del gradiente. Si la velocidad de aprendizaje es muy pequeña, el modelo converge muy lentamente; si la tasa de aprendizaje es demasiado grande, el modelo diverge. Para las redes neuronales comunes se establece típicamente entre 0.1 y 0.00001. Este parámetro va altamente relacionado al tipo de optimizador que se utiliza.

5. Drop out: omite una cantidad de neuronas al azar al momento en que los pesos de las neuronas previamente conectadas a la neurona están por evaluarse. Si la neurona llegará activarse o no, ésta no dependería totalmente de una sola neurona sino de muestras del grupo que varían aleatoriamente. Esto generaliza el modelo y lo hace más confiable evitando el sobre ajuste. Este proceso reduce la varianza del modelo ya que no dependerá de que las neuronas estén acomodas de forma exacta en un caso particular.

2.2.1.4. Diferentes tipos de arquitecturas de redes neuronales

1. **Red neuronal densa:** es la red neuronal más común y también se le conoce como red totalmente conectada. Cada neurona se encuentra completamente conectada con todas las otras y cada unidad realiza la operación básica de activarse cuando los pesos la favorecen. Esta red siempre se encuentra en la parte final ya que cumple con la función de clasificar cuando se activan las neuronas de la última capa.

2. **Redes neuronales recurrentes (RNN):** las redes neuronales recurrentes son un tipo de red neuronal donde las conexiones entre unidades (celdas o neuronas) forman un grafo dirigido a lo largo de una secuencia temporal. Lo anterior le permite exhibir un comportamiento dinámico temporal. Pueden usar su estado interno para procesar secuencias de entrada como un tipo de memoria. sus aplicaciones son tareas como el reconocimiento de escritura manuscrita o el reconocimiento de voz.

   Las RNN sufren de memoria a corto plazo. Si una secuencia es bastante larga, les resulta difícil transportar información de los pasos anteriores a los posteriores. Entonces, si está tratando de procesar un párrafo de texto para hacer predicciones, los RNN pueden omitir información importante desde el principio. Durante la retro-propagación (back-propagation), las redes neuronales recurrentes sufren el problema del gradiente de fuga. Los gradientes son valores utilizados para actualizar los pesos de una red neuronal. El problema del gradiente de fuga es que el gradiente se reduce a medida que se propaga a través del tiempo. Si un valor de gradiente se vuelve extremadamente pequeño, no contribuye al aprendizaje. Dentro de la familia de redes neuronales recurrentes se discuten dos de ellas en especial. La red neuronal de unidad con compuerta recurrente (GRU) y la red neuronal de memoria a largo-corto plazo (LSTM). Estas redes sustituyen la unidad de operación de la neurona clásica y la remplazan con un unidad llamada celda que esta compuesta de distintos bloques operacionales que funcionan de manera similar pero con algunas diferencias.

3. **Red neuronal GRU:** por sus siglas en ingles: Gated recurrent unit (GRU). La red GRU conceptualmente tiene una compuerta de reinicio y olvido que ayuda a garantizar que su memoria no sea controlada por el seguimiento de dependencias a

corto plazo. La red aprende a usar sus puertas para proteger su memoria de modo que sea capaz de hacer predicciones a largo plazo. Pueden ignorar secuencias según sea necesario. En ciertos casos, las redes GRU son más rápidas y fáciles de entrenar con menos datos en comparación con las redes LSTM.

4. **Red neuronal LSTM:** por sus siglas en ingles: Long-short term memory (LSTM). La red LSTM, a diferencia de las redes neuronales tradicionales, en lugar de poseer neuronas de manera clásica, tiene como neuronas bloques de memoria que están conectados a través de capas. Estos bloques de memoria facilitan la tarea de recordar valores para largos o cortos períodos de tiempo. Por lo tanto, el valor almacenado no se reemplaza (al menos a corto plazo) de forma iterativa en el tiempo. A su vez, el término de gradiente no tiende a desaparecer cuando se aplica la retro-propagación durante el proceso de entrenamiento, tal y como acontece en el uso de las redes neuronales clásicas.

5. **Red neuronal convolutiva:** esta red neuronal aplica la operación de convolución mediante un filtro a una entrada que resulta en una activación. La aplicación repetida del mismo filtro a una entrada da como resultado un mapa de activaciones denominado mapa de características. Estas redes tienen la capacidad de aprender automáticamente una gran cantidad de filtros específicos para un conjunto de datos de entrenamiento. Esta característica es bastante efectiva para problemas de modelado predictivo específico, como la clasificación de imágenes. El resultado es un conjunto de características altamente específicas que pueden detectarse en cualquier lugar en las imágenes de entrada.

    a) Capas convolutivas: se refiere a la cantidad de capas que se utilizan en el modelo que realizan la operación de convolución sobres los datos. Por lo menos debe existir una. Este número depende de la complejidad del conjunto de entrenamiento. Cada capa de convolución reduce el número de características de entrada que se entrega a la red neuronal densa; sin embargo, después de dos o tres capas, la mejora de la exactitud que se pueda ganar es cada vez menor, hasta ser relativamente pequeña, y realmente se debe preocupar por el tiempo de ejecución del entrenamiento. Cada problema es diferente y se recomienda incrementar el número de capas hasta estar satisfecho con los resultados.

b) Strides: se refiere al valor numérico que determina la cantidad de pasos a la que se mueve el filtro de convolución sobre la matriz de datos. Si el valor predeterminado de uno se aumenta, a otro valor, éste es el número de renglones y columnas que brinca cada vez que el filtro se mueva de región después de haber calculado la convolución para esa zona. Entonces, en lugar de que el filtro se mueva de uno en uno por cada dato en el que quepa el filtro, éste se mueve en una cantidad de acuerdo a este parámetro. Esto podría hacer que el cálculo se haga más rápido en una imagen grande, pero esto influye en la resolución que podría aportar pequeños detalles de los datos al cálculo total del mapa de características final o filtro que se guarda. En este trabajo sólo se maneja el valor predeterminado de uno para este parámetro y no se cambia en ningún momento.

c) Profundidad de filtro (Filter depth): define la cantidad de filtros que se aplican en paralelo sobre la entrada. Esta cantidad define la longitud de la dimensión adicional que se genera después de finalizar todo el proceso de la capa convolutiva. Al añadir mayor cantidad de filtros, incrementa la cantidad de operaciones a realizar, lo que también aumenta el tiempo de ejecución del entrenamiento de la red neuronal convolutiva. Tampoco existe un regla que defina cuántos filtros se deben usar; por lo general, se incrementa este parámetro mientras ayude a mejorar la exactitud en la clasificación del modelo.

d) Bordeado (Padding): al momento de tomar la ventana y empezar a deslizarla de izquierda a derecha y de arriba hacia abajo por los datos, se debe colocar en la esquina superior de donde comienzan los datos. Existen dos posibilidades, si se selecciona ’padding same’, el centro de la ventana se tiene que recorrer horizontal y verticalmente algunos lugares para que se pueda insertar la ventana. Este efecto ocasiona que las dimensiones del filtro aprendido pierdan unas cuantas unidades. Si de lo contrario, se quiere evitar esta pérdida, se debe seleccionar la opción ’zero padding’. Esta opción crea una marco con el valor de cero alrededor de la matriz de los datos, de forma que la ventana pueda deslizarse completamente sobre la entrada sin perder datos en los bordes. Para este trabajo se conserva la opción de ’zero padding’ como predeterminada.

2.3. Reducción de características

En esta etapa se define un tamaño u objetivo que facilite el trabajo del clasificador. Se eligen aquellos parámetros de mayor correlación para maximizar la exactitud en el resultado de la clasificación.

El espacio original de características se transforma para reducir la covarianza, que es la medida estadística que indica el grado de dependencia entre los datos, es decir qué tan relacionados están.

Generalmente se utiliza el análisis de componente principal (PCA) o análisis discriminante lineal (LDA). Estos métodos minimizan la dispersión de clases eligiendo los componentes con valores propios más altos.

Para reducir aún más el tamaño del espacio de características en esta etapa, se decide qué características se quedan y cuáles se descartan. Generalmente se escoge a mano con base en la observación y experiencia o con ayuda de algún algoritmo.

2.3.1. Algoritmo de alivio (relief) para selección de características

Originalmente, este algoritmo se diseñó para su aplicación a problemas de clasificación binaria con características discretas o numéricas. Este algoritmo se comporta como un filtro para la selección de características, el cual es notablemente sensible a las interacciones de éstas. El filtro calcula una puntuación para cada función que luego se puede aplicar para clasificar y seleccionar las funciones de puntuación más altas.

Estas puntuaciones pueden aplicarse como ponderaciones de características para el modelado posterior. La puntuación de las características de alivio se basa en la identificación de las diferencias de valor de las características entre los pares de las instancias vecinas más cercanas. Si se observa una diferencia en el valor de la característica en un par de instancias vecinas con la misma clase, un acierto, la puntuación de la característica disminuye. Por el otro lado, si se observa una diferencia en el valor de la característica en un par de instancias vecinas con diferentes valores de clase, un desacierto, la puntuación de la característica aumenta.

2.3.2. Análisis de componentes principales (PCA)

El PCA corresponde al campo de aprendizaje de máquina como algoritmo de reducción de dimensiones. Es un método de reducción de dimensionalidad que se usa a menudo para reducir la dimensionalidad de grandes conjuntos de datos, al transformar un gran conjunto de variables en uno más pequeño que aún contiene la mayor parte de la información del gran conjunto.

La reducción de la cantidad de variables de un conjunto de datos naturalmente conlleva un costo de precisión, pero el truco en la reducción de la dimensionalidad es cambiar un poco de precisión por simplicidad. Debido a que los conjuntos de datos más pequeños son más fáciles de explorar y visualizar, esta técnica hace que el análisis de los datos sea mucho más fácil y rápido para los algoritmos de aprendizaje automático sin variables extrañas para procesar.

2.3.3. Incrustación estocástica de vecinos de distribución-T (t-SNE)

El t-SNE lo desarrollaron Laures van der Maaten y Geoffrey Hinton en 2008. Corresponde al campo de aprendizaje de máquina como algoritmo de reducción de dimensiones. Se utiliza tanto para el pre-procesamiento de un gran conjunto de datos como para la clasificación y visualización, donde se busca encontrar una forma de reducir su dimensionalidad.

El objetivo es tomar un conjunto de puntos en un espacio de alta dimensión y encontrar una representación fiel de esos puntos en un espacio de menor dimensión en un plano. Esta capacidad sirve para crear e interpretar “mapas” bidimensionales y tridimensionales que sean convincentes a partir de datos con cientos o incluso miles de dimensiones.

El algoritmo no es lineal y se adapta a los datos subyacentes, realizando diferentes transformaciones en diferentes regiones. Esas diferencias pueden ser una fuente importante de confusión, lo que si no se interpreta bien, puede generar complicaciones. Para aprovechar al máximo el t-SNE, existen hiper-parámetros adicionales relacionados con el proceso de optimización. Ésto puede significar analizar múltiples gráficos

con diferentes valores. Dentro de los hiper-parámetros sintonizables más importantes del t-SNE se encuentran: perplejidad, número de pasos y la tasa de aprendizaje.

1. Perplejidad: rige el equilibrio entre los aspectos locales y globales de sus datos. Este parámetro es una conjetura sobre el número de vecinos cercanos que tiene cada punto. Los valores típicos varían entre 5 y 50, valores más pequeños o grandes pueden causar comportamientos inesperados.
2. Número de pasos: el algoritmo t-SNE no siempre produce resultados similares en ejecuciones sucesivas. Por experiencia, si se ve una gráfica con formas extrañas "pellizcadas", es probable que el proceso se haya detenido demasiado pronto. Desafortunadamente, no hay un número fijo de pasos que produzca un resultado estable. Diferentes conjuntos de datos pueden requerir diferentes números de iteraciones para converger, lo más importante es iterar hasta alcanzar una configuración estable.
3. Tasa de aprendizaje: también conocida como épsilon. Con una mayor tasa de aprendizaje, los datos convergen con mayor rapidez en menos iteraciones. Lo anterior ayuda en grandes cantidades de datos que tardan en converger. De lo contrario, en cantidades pequeñas de datos, se podría brincar algún punto deseado y evitar que se encontrara un mejor resultado.

A veces se pueden ver algunas formas de agrupamiento. Ésto es lo que los separa de confundirlas con ruido. El t-SNE exagera enormemente el tamaño del grupo más pequeño de puntos. Es raro que los datos se distribuyan en una simetría perfecta. Para la topología, es posible que necesite más de un gráfico. A veces, puede leer información topológica en un gráfico t-SNE, pero eso generalmente requiere vistas en múltiples perplejidades.

El tamaño de los agrupamientos de datos y las distancias entre ellos no significan nada. El algoritmo t-SNE adapta su noción de ’distancia’ a las variaciones de densidad regional en el conjunto de datos. Como resultado, naturalmente expande los grupos densos y contrae los dispersos, igualando los tamaños de los grupos. Para que quede claro, éste es un efecto diferente al hecho de que cualquier técnica de reducción de dimensionalidad distorsione las distancias.

El uso del t-SNE se ha vuelto popular debido a su flexibilidad y puede encontrar una estructura donde otros algoritmos de reducción de la dimensionalidad no pueden. Desafortunadamente, esa misma flexibilidad hace que sea difícil de interpretar. Fuera de la vista del usuario, el algoritmo hace todo tipo de ajustes que ordenan sus visualizaciones. Al estudiar cómo se comporta el t-SNE en casos simples, es posible desarrollar una intuición que explique lo que está sucediendo.

2.4. Modelo circumplejo del afecto

En esta investigación se trabajó con la tarea de clasificar ladridos de perros que fueron inducidos en distintos contextos, los cuales pueden ser interpretados de distintas maneras. Se trata de utilizar las etiquetas mayormente relacionadas al estado emocional del perro. Por lo tanto, es necesario apoyarse de algo de teoría y modelos de emociones que puedan ser útiles para estos experimentos.

El modelo circumplejo del afecto (MCA) se considera un paradigma en el campo de la neurociencia afectiva. (Russell y James, 1980) postula dos dimensiones de afecto interrelacionadas que hacen referencia al nivel de activación de la emoción y a la valoración de la misma; los autores denominan a estas dimensiones excitación y valencia, respectivamente. Puesto que el objetivo del MCA es categorizar las emociones, puede ayudar a mostrar los estados afectivos del perro en distintos contextos. La representación gráfica de los dominios afectivos se hacen en términos polarizados de tipo positivo o negativo, o en relación a animales no humanos en términos adaptativos o desadaptativos. Siguiendo esta línea, Russell y sus colaboradores consideran que los estados emocionales están mejor representados en un círculo o en dos dimensiones bipolares, mencionadas anteriormente. La función de esta categorización se basa en cómo las personas interpretan el afecto, en la tendencia de categorizar expresiones emocionales verbales y no verbales, y en las estructuras cognitivas del afecto. La figura 7 muestra un diagrama del modelo circumplejo de afecto.

El MCA resulta pertinente ya que valora aspectos no verbales en la expresión de la afectividad. No obstante, incluso las conductas no verbales, como los indicadores fisiológicos en animales, pueden ser de ayuda para inferir su estado afectivo. Conviene

resaltar que no existen métodos objetivos de medición que lo hagan con exactitud.

Además, el MCA ofrece nuevos enfoques teóricos y empíricos para estudiar el desarrollo de los estados afectivos, así como los fundamentos genéticos y cognitivos del procesamiento afectivo dentro del sistema nervioso central.

El MCA dentro de la investigación con animales enfatiza cómo influyen las estructuras primitivas y subcorticales en el procesamiento de las emociones. Mientras que la investigación en humanos demuestra la importancia de las estructuras neocorticales en la experiencia emocional (Berridge y Kent, 2003).

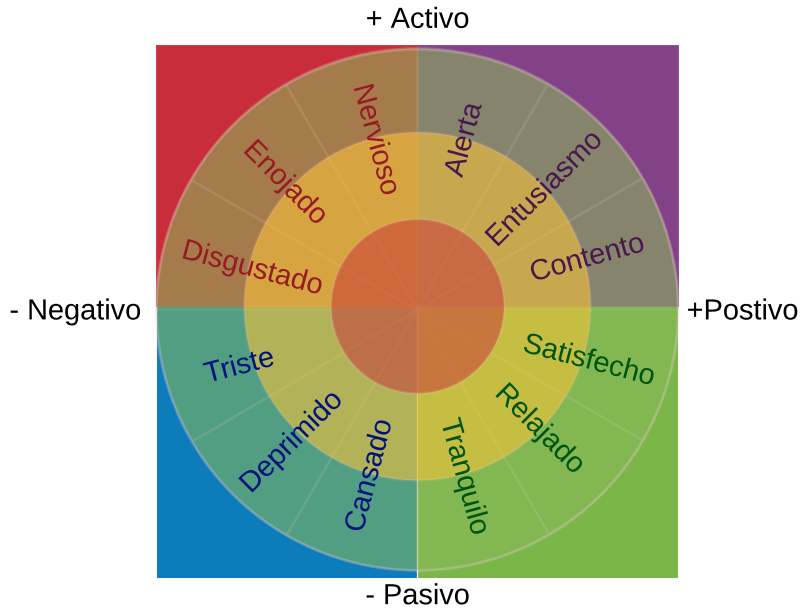

Figura 7. Modelo Circumplejo del Afecto, adaptación de las revisiones de Posner, Russell y Peterson (2005).

De hecho, los abundantes datos de los estudios de lesiones y las investigaciones de neuroimagen demuestran que la activación de la corteza prefrontal participa centralmente en la experiencia de emociones positivas y negativas (Davidson *et al.*, 1990).

Al estimular selectivamente las vías neuronales y observar comportamientos posteriores, o al contrario, al provocar conductas en circunstancias experimentales altamente limitadas y medir la actividad neuronal, los investigadores en animales han construido taxonomías de las emociones básicas y han propuesto vías neuronales específicas asociadas con cada emoción básica (Panksepp, 1998).

En conclusión, el enfoque de este modelo experimental ha ayudado a los investigadores a comenzar a explorar las bases neuronales de la emoción. No obstante, permanece limitado en la información que proporciona acerca de las experiencias afectivas y los sistemas neuronales que las respaldan. Los investigadores se ven obligados a atribuir estados afectivos a los animales en función de los comportamientos que exhiben los animales.

Sin embargo, las conductas afectivas no son suficientes ni necesarias para caracterizar los estados emocionales (Kagan, 2003). Se han documentado casos en que un animal podría experimentar una emoción sin demostrar ningún cambio manifiesto en el comportamiento y, a la inversa, a través de la manipulación experimental, el animal podría mostrar comportamientos afectivos sin ninguna emoción asociada.

2.5. Conclusiones del marco teórico

En conclusión, se mezclan temas y conocimiento de distintas áreas desde procesamiento digital de señales que involucra el procesado de la señales de audio hasta la inteligencia artificial, utilizando algoritmos de aprendizaje de máquina para encontrar patrones y clasificar tareas. Estas áreas tienen en común algoritmos y matemáticas que corresponden al campo de las ciencias computacionales con aplicación a un problema del mundo real en el área de la etología. La etología estudia hasta las bases neurológicas asociadas al comportamiento de los animales, creando modelos que puedan servir para explicar las conductas asociadas a distintos contextos.

Aquí termina el marco teórico, el cual cubre la mayor parte del conocimiento general que se utilizó en esta investigación para poder comprender los conceptos; sin embargo, en el siguiente capítulo, al mencionar los conceptos, se espera dejar claro con más detalle la utilidad y el caso en el que se utilizan.

Capítulo 3. Metodología

Esta sección describe con detalle la metodología llevada a cabo.

3.1. Preliminares y estrategia general

El método propuesto se desarrolló como una herramienta que facilita la identificación del ladrido del perro y reconoce al individuo, especie, edad, sexo y contexto asociado a cada ladrido, con la finalidad de estimar información que aporte un perfil más amplio del animal. Se construyeron cinco modelos distintos para cada una de las tareas de clasificación. El método propuesto se divide en tres etapas, las cuales se describen a continuación con la ayuda de la figura 8.

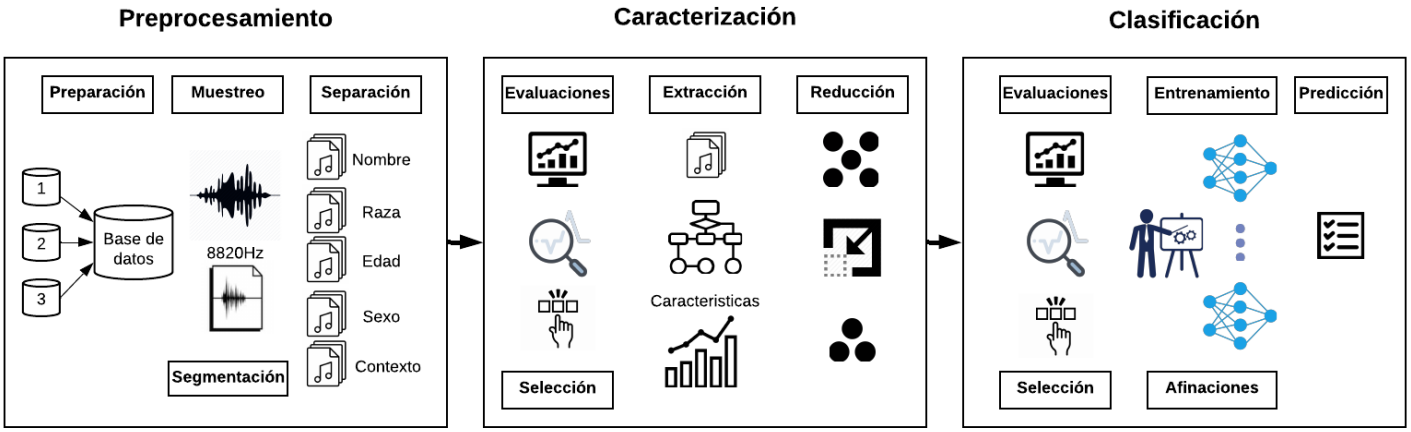

Figura 8. Esquema del método completo

Primero, se tiene una etapa de preprocesamiento donde se realizaron todos los procesos necesarios para presentar los datos con el formato adecuado, con los cuales se construyeron cada uno de estos modelos.

Posteriormente, en la etapa de caracterización, se evalúan distintas técnicas y se selecciona la de mejor desempeño, con la que se extraen los atributos más importantes y se preparan los datos para la siguiente etapa.

Por último, se presenta la etapa de clasificación donde se construye y se entrena cada modelo. Se utilizó la mejor arquitectura de redes neuronales profundas dentro de las que fueron evaluadas. Además, se aplica una técnica mediante la cual se afinan los mejores hiper-parámetros para cada modelo. Una vez que se tiene el modelo afinado y entrenado, se consiguen los resultados finales de predicción.

En la etapa de procesamiento existe una sección de segmentación y muestreo que una vez que se realizó para cada archivo en las bases de datos, no es necesario que se repita el proceso para cada modelo.

En la etapa de caracterización y clasificación se repite la sección de evaluación y selección. En la evaluación se investigó y se probaron distintos métodos de los cuales se midió el rendimiento que aportaron en los resultados finales, ésto se realizó para ambos casos. Con los resultados se concluyó cuál de todos los métodos fue el de mejor desempeño, ésto se realizó como parte de la investigación previa antes de implementar el método completo. En la selección ya conocemos cúal es el método que dará mejores resultados y éste sólo se implementa cada vez que que se construye el modelo con los métodos óptimos para cada tarea.

De la misma manera en la etapa de clasificación se hace la afinación, la cual fue un trabajo de investigación de prueba y error, hasta que se consiguió la mejor combinación de hiper-parámetros. Ya que se conoce el método de afinación y los valores de hiper parámetros óptimos, sólo se entrena para cada modelo de una tarea de forma que se obtengan los resultados más altos en la predicción final de cada uno.

3.1.1. Preparación de los datos

Se cuenta con una colección de archivos de audio, donde cada grabación contiene un ladrido individual. Esta colección consiste de tres distintas bases de datos que son las siguientes: perros mudi, mescalina 2015 y mescalina 2017. Dichas bases de datos se describen con mayor detalle a continuación.

3.1.2. Base de datos de perros mudi

La base de perros mudi es una de las bases de datos de ladridos de perros que tiene más tiempo siendo usada por diversos trabajos de investigación. La base la reportaron en el trabajo Hungaro (Molnár *et al.*, 2008). Las condiciones en que los ladridos se grabaron son las siguientes:

1. Extraño: El experimentador, un sujeto masculino de 23 años, tomaba el rol de ser un extraño para todos los perros. Él aparecía en el jardín del dueño o en la puerta de entrada de su departamento en ausencia del dueño. El experimentador registró los ladridos del perro durante su aparición durante 2-3 minutos.

2. Pelea: Para que los perros actúen en esta situación, el entrenador alienta al perro a ladrar agresivamente y morder el guante que tiene en el brazo del entrenador. Mientras tanto, el propietario mantiene al perro con correa.

3. Caminata: Se le pidió al propietario que se comportara como si se estuviera preparando para salir a pasear con el perro. Por ejemplo, el propietario tomó la correa del perro en su mano y le dijo al perro nos vamos ahora".

4. Solo: El dueño ató al perro a un árbol con una correa en el parque y se alejó, fuera de la vista del perro.

5. Comida: El dueño sostiene el plato de comida para perros a 1.5 m delante del perro.

6. Pelota: El dueño sostuvo una pelota (o un juguete favorito del perro) a una altura de aproximadamente 1.5 m delante del perro.

7. Juego: se le pidió al dueño que jugara con el perro un juego habitual, como tirar y aflojar, persecución o luchas suaves. El experimentador registró los ladridos emitidos durante esta interacción.

Distribución de clases:

En esta base de datos se tienen 6614 ladridos segmentados a nivel individual con un promedio de duración 1 segundo. La figura 9 presenta la cantidad de ladridos por clase en la base de datos de los perros mudi.

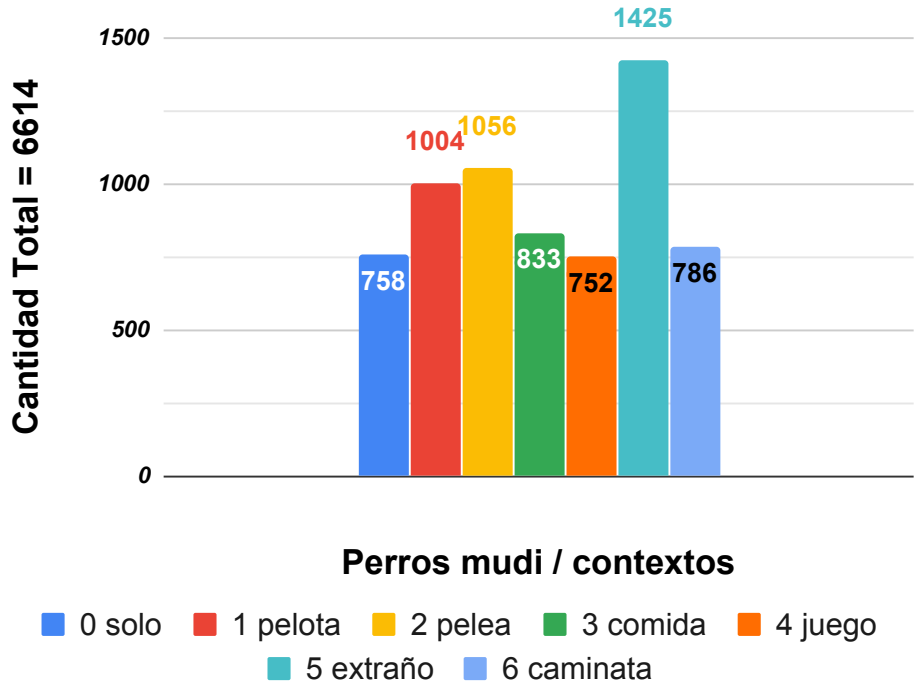

Figura 9. Distribución de clases para contextos en perros mudi

La base de datos tiene once individuos con sus nombres correspondientes todos de la raza Mudi. Como se muestra en la figura 10.

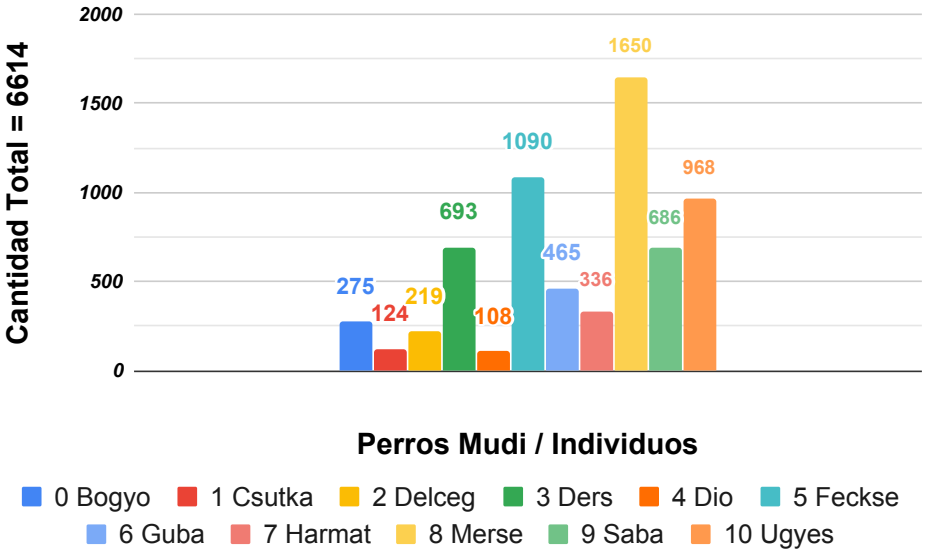

Figura 10. Distribución de clases para individuos en perros mudi

En esta base de datos, las edades de perros se etiquetaron entre 1 hasta 11 años y se aplicó el agrupamiento de edad respectivo a la raza. Esto se muestra en la figura 11.

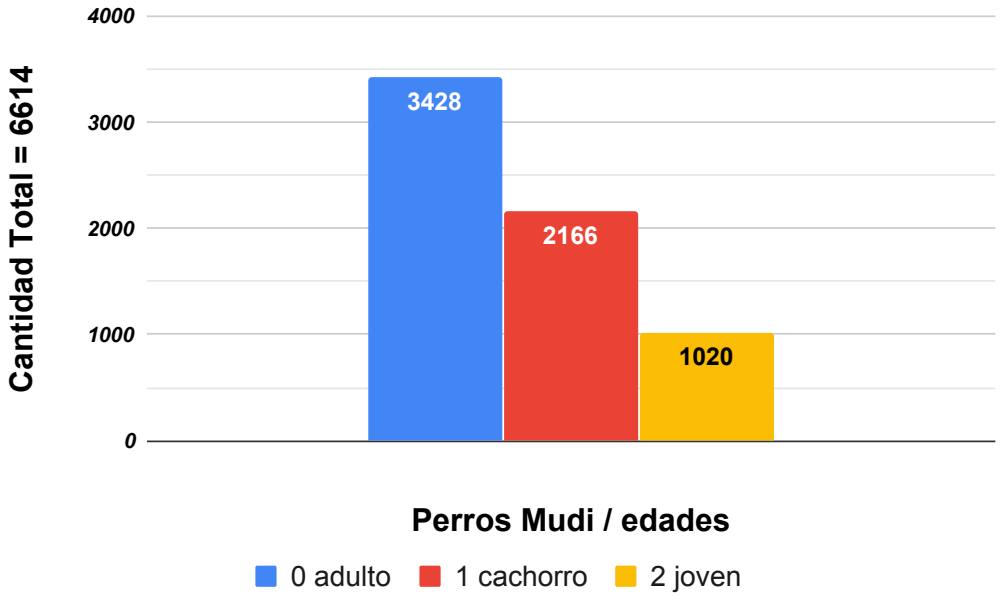

Figura 11. Distribución de clases para grupos de edad en perros mudi

La figura 12 muestra la distribución por sexo de las perros mudi.

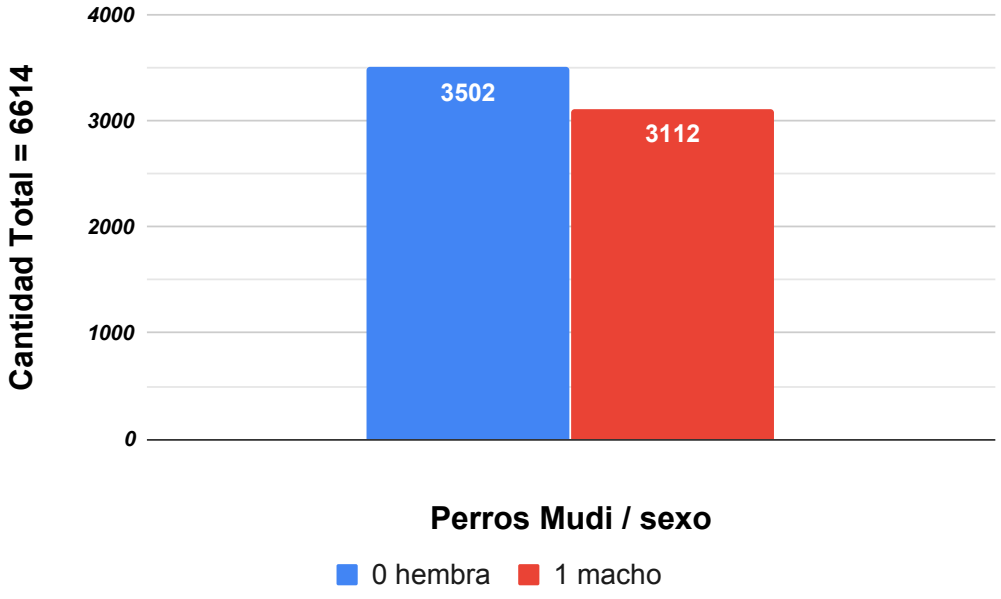

Figura 12. Distribución de clases para sexo en perros mudi

3.1.3. Base de datos de perros Mescalina 2015

La base de datos Mescalina 2015 la grabó el CICESE-UT[3] y pertenece a la empresa mexicana Mescalina y se utilizó en el trabajo de Pérez *et al* (2015). Se pidió la autorización de los propietarios para poder grabar a sus mascotas y someterlos a una serie de estímulos de acuerdo al protocolo que se explica a continuación:

1. L-P Juego: El propietario estimula al perro utilizando los objetos o juguetes con los que normalmente juega, esperando lograr vocalizaciones.
2. L-S1 Alerta normal ante presencia de un extraño: se comienza a tocar repetidamente el timbre del domicilio y dar golpes lo suficientemente fuertes a la puerta esperando provocar que el perro ladre. Esto se repite las veces que sea necesario.
3. L-S2 Agresión: Una persona entra al domicilio, llamando la atención del perro y acercándose amenazadoramente a él, golpeando los pies contra el piso y dando palmadas para lograr que el perro se sienta agredido.
4. L-S3 Agresión al dueño dentro del domicilio: se solicita al propietario que se acerque y se simula un ataque, haciendo también ruidos como palmadas y golpes al piso. Para lograr una mejor reacción de parte del perro, de ser necesario, se le pide al dueño que grite pidiendo auxilio o el nombre del perro.

Distribución de clases:

En esta base de datos se tienen 6077 ladridos segmentados a nivel individual con un promedio de duración 1 segundo. La figura 13 muestra las cuatro clases de contexto de esta base de datos.

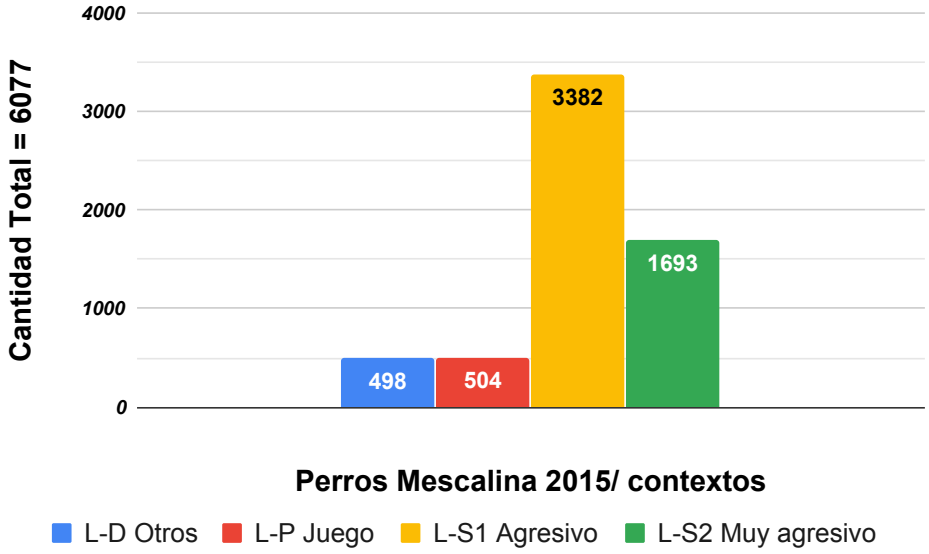

Figura 13. Distribución de clases para contextos de perros Mescalina 2015

La base de datos Mescalina 2015 tiene treinta y siete individuos con sus nombres correspondientes, como se muestra en la figura 14.

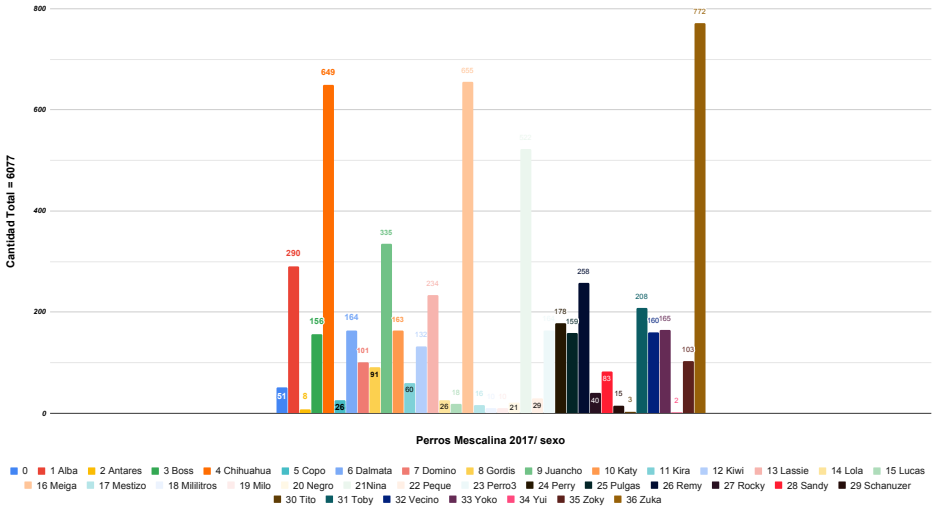

Figura 14. Distribución de clases para individuos de perros Mescalina 2015

Los ladridos corresponden a las grabaciones de 11 razas de perros comunes en los

hogares. Esto se muestra en la figura 15.

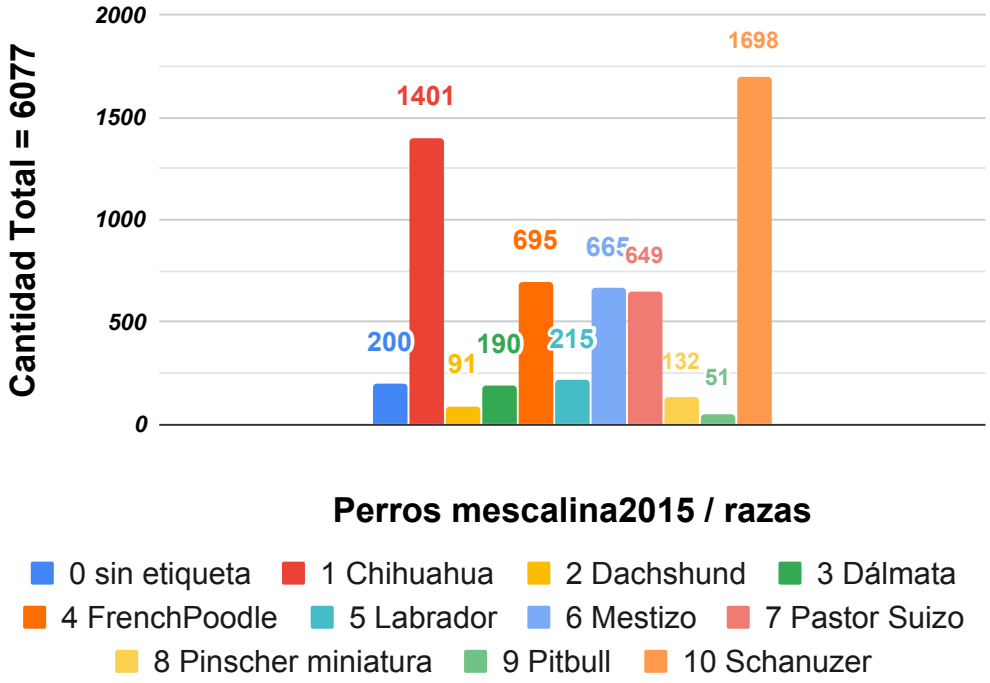

Figura 15. Distribución de clases para razas de perros Mescalina 2015

En esta base de datos las edades de perros se etiquetaron en meses. Desde los 6 meses hasta 72 y se aplicó el agrupamiento de edad respectivo a la raza. Esto se muestra en la figura 16.

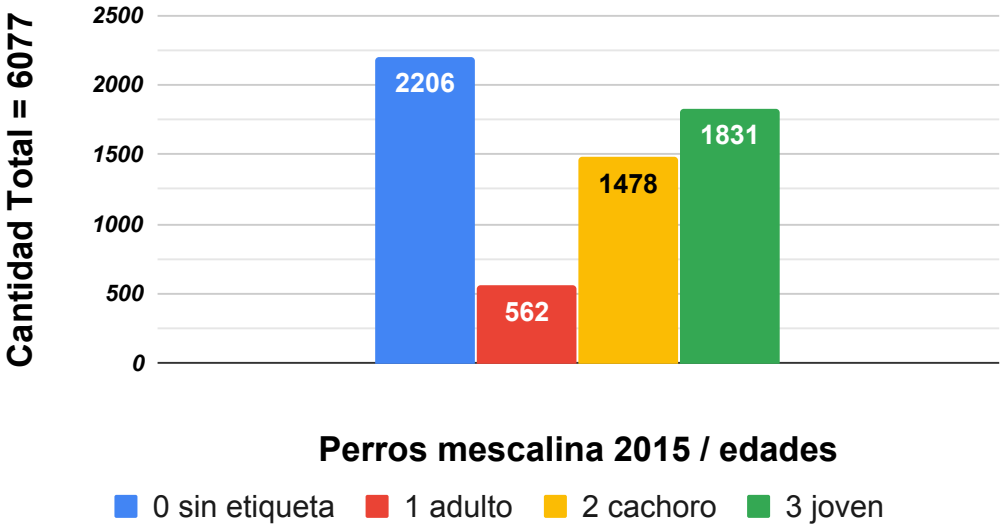

Figura 16. Distribución de clases para grupos de edades de perros Mescalina 2015

La figura 17 muestra la distribución de sexo de esta base de datos.

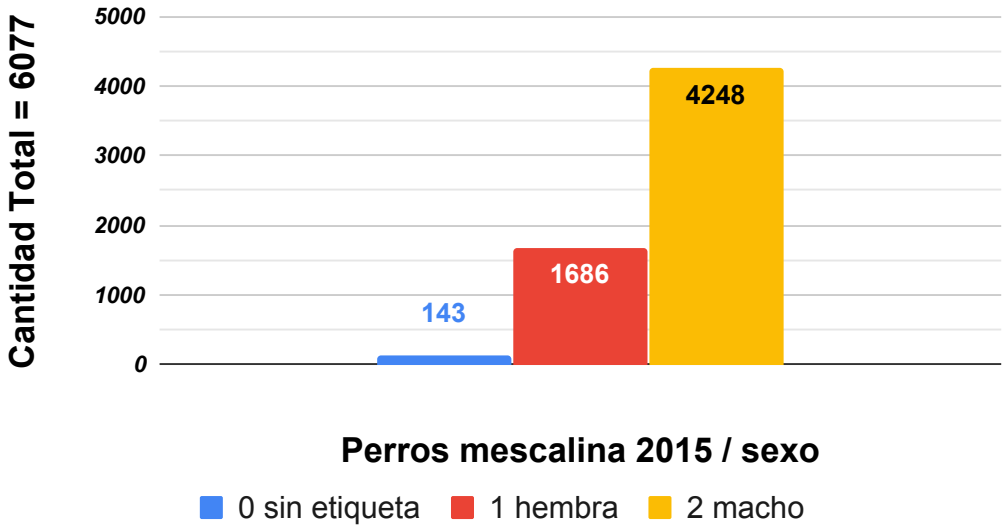

Figura 17. Distribución de clases para sexo de perros Mescalina 2015

3.1.4. Base de datos de perros Mescalina 2017

La base de Mescalina 2017 la grabó la empresa mexicana Mescalina. Con el fin añadir más recursos para la investigación, se incrementaron la cantidad de muestras. En estas grabaciones se añaden nuevos contextos junto a los anteriores, los cuales se describen a continuación.

1. L-H llegada a casa: se solicita al propietario que salga del domicilio, se aleja con el experimentador, posteriormente regresan, y desde afuera de la casa le habla cariñosamente a su perro, pero sin abrir la puerta.
2. L-PA simulación de paseo: el propietario realiza la rutina normal que precede al paseo. De ser necesaria una mejor reacción, se estimula al perro con llamadas cálidas o simulando abrir la puerta hasta que se emocione y produzca vocalizaciones.
3. L-TA tristeza/ansiedad por separación: el propietario ata al perro con la correa a un árbol y se aleja de su vista. Para lograr un mejor resultado se le pide al dueño que se despida de él.
4. L-A asustado en el parque: mientras el perro continúa atado al árbol, el experimentador o algún otro extraño se acerca amenazadoramente al perro.

Distribución de clases:

En esta base de datos se tienen 6948 ladridos segmentados a nivel individual con un promedio de duración 1 segundo. La figura 18 muestra la distribución de contextos.

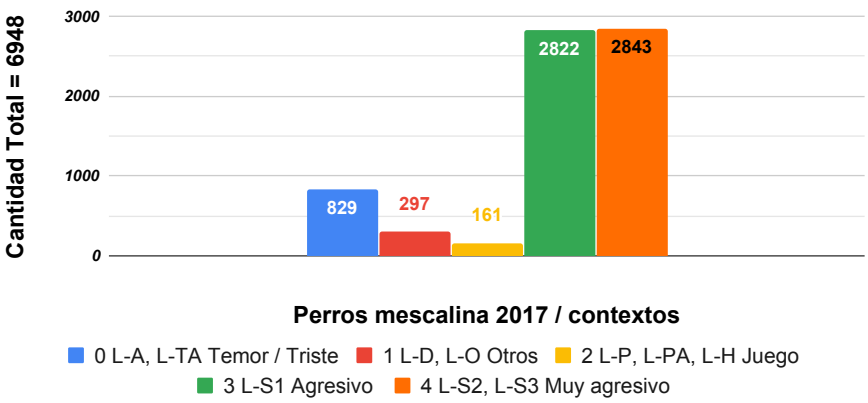

Figura 18. Distribución de clases para contextos de perros Mescalina 2017

La base de datos Mescalina 2017 tiene sesenta y cinco individuos que se muestran en la figura 19.

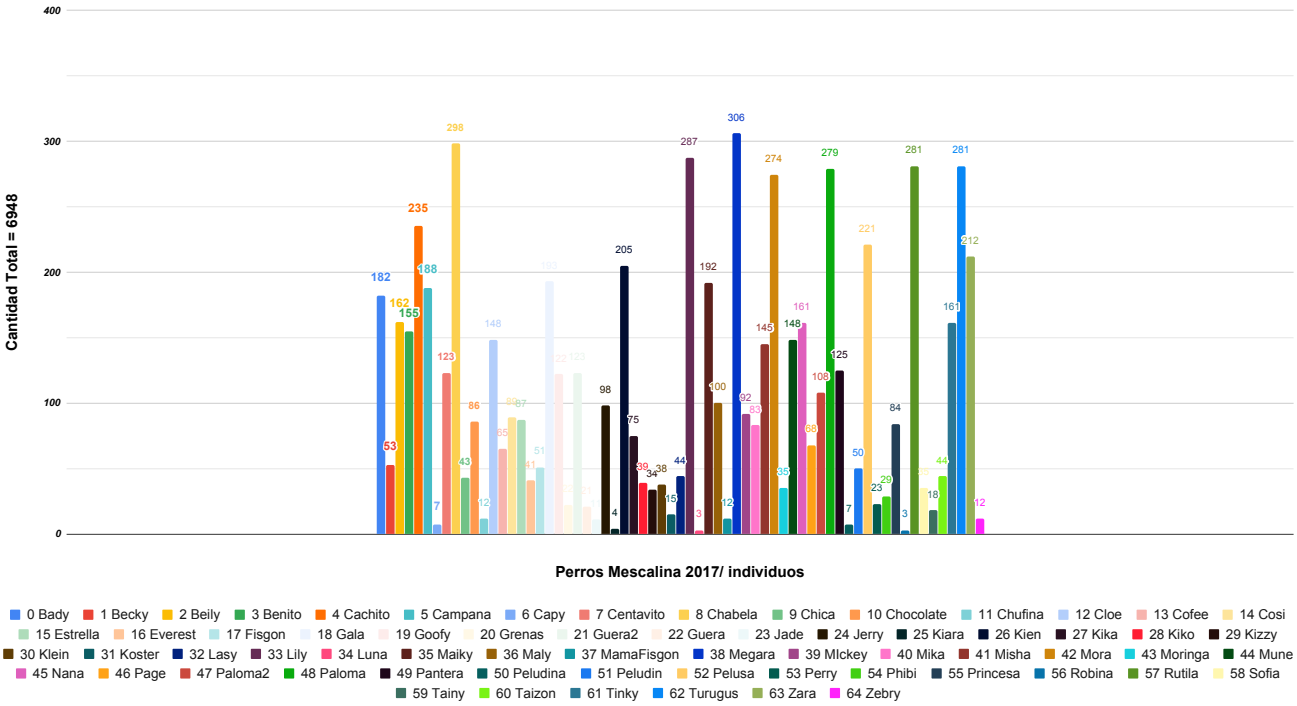

Figura 19. Distribución de clases para individuos de perros Mescalina 2017

Los ladridos corresponden a las grabaciones de cuatro razas de perros comunes en

los hogares, las cuales se muestran en la figura 20.

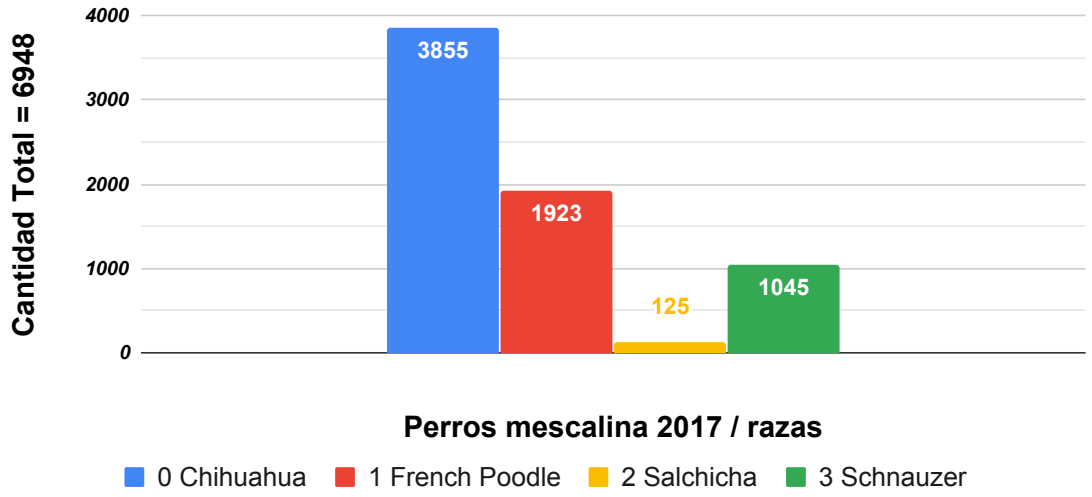

Figura 20. Distribución de clases para razas de perros Mescalina 2017

La figuras 21 muestra la distribución de clases para sexo de perros de la basa de datos Mescalina 2017.

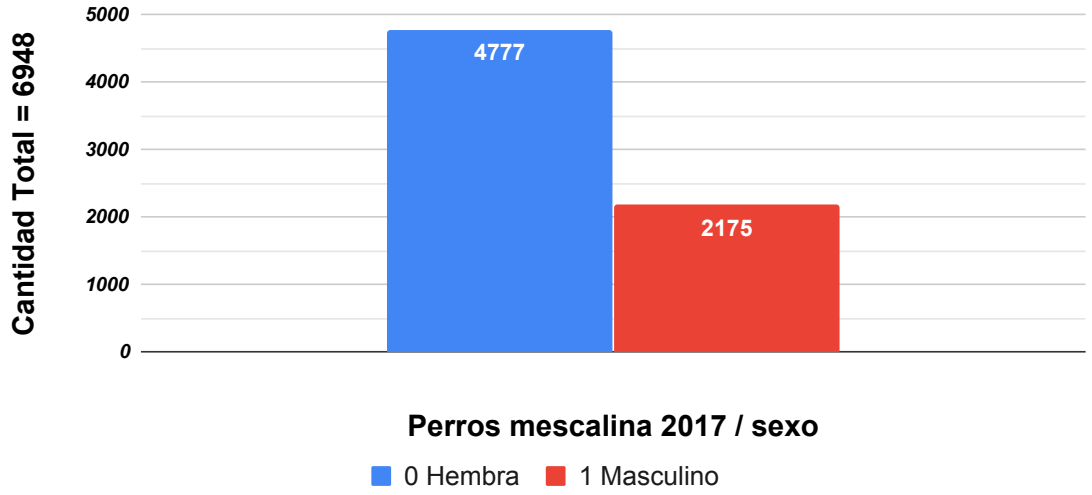

Figura 21. Distribución de clases para sexo de perros Mescalina 2017

3.1.5. Unión de bases de datos para el método propuesto

Al concatenar las tres bases de datos se forma una colección de 19643 muestras. En esta base de datos se incluyen las etiquetas correspondientes a cada una de las cinco distintas tareas que se busca identificar; sin embargo, algunas de las bases de datos se etiquetaron parcialmente, dándose el caso de no tener etiquetas para algunas muestras o simplemente no incluyen las etiquetas para cierta tarea en específico.

Puesto que para el entrenamiento, validación y prueba del modelo se necesita contar con todas las etiquetas, se separan en cinco distintos. Estos conjuntos si están completamente etiquetados para cada tarea específica. Estos conjuntos se describen con detalle en la siguiente sección.

3.1.6. Propuesta de modelo de clasificación de contextos

La clasificación de contextos utilizando ladridos de perros depende en gran medida del contexto de la etiqueta que se le asigne a cada ladrido. Las condiciones en las se grabó el ladrido corresponden a una situación específica, la cual se planeó para inducir el ladrido en respuesta a un estimulo. Ésto depende del diseño del experimento de captura y etiquetado de los ladridos. El experimento se diseñó con una metodología confiable por etólogos expertos en el área. Sin embargo, siempre puede existir algún factor no tomado en cuenta que pueda afectar el experimento Por tal razón no se puede garantizar con exactitud si el perro está mostrando la reacción esperada y puede variar mucho de un individuo a otro.

Organizar los datos para este experimento es complicado debido que, al concatenar las distintas bases de datos, se tiene el conjunto de etiquetas mezcladas incluyendo las similitudes y coincidencias de etiquetado. Esto quiere decir que en ciertos casos la etiqueta con diferente nombre corresponde a un contexto muy similar o igual.

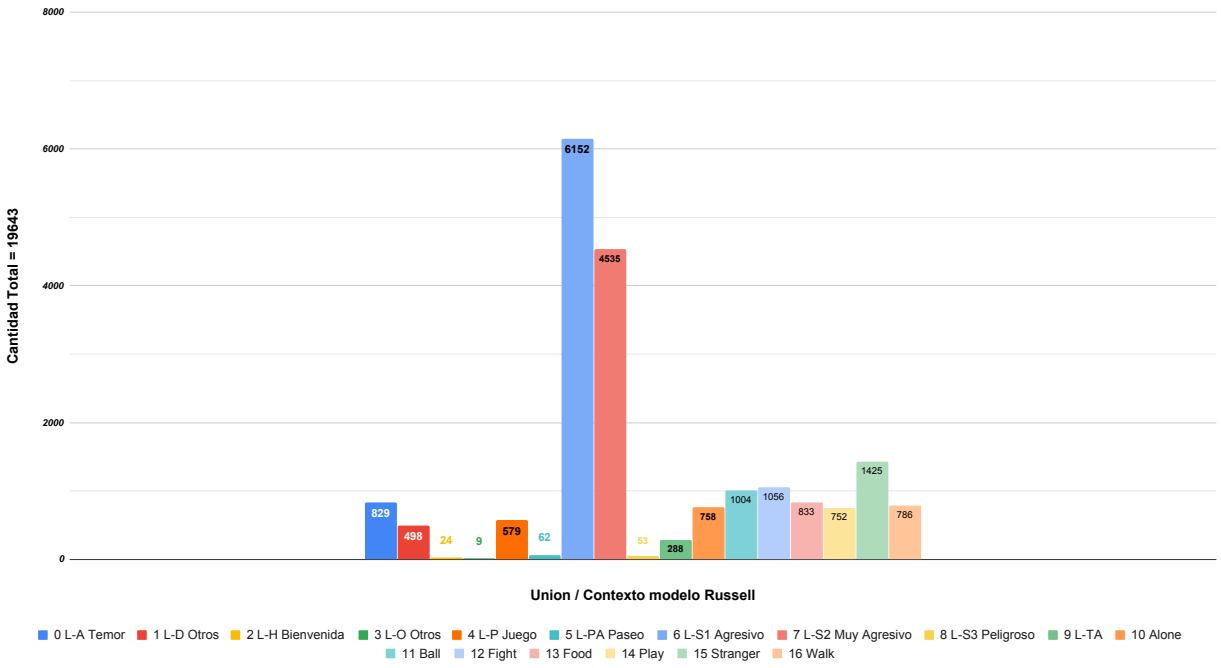

Figura 22. Distribución de clases para contextos de unión propuesta

En la sección 4.6 se puede observar que la tarea de identificar el contexto obtuvo menor desempeño en la precisión de clasificación comparado con las demás tareas, individuos, raza, edad, sexo. Por lo que se propuso un modelo que unifique las etiquetas de contextos similares en un solo grupo de categorías.

3.1.7. Ubicación de contextos dentro del modelo afectivo de Russell

Al analizar las condiciones de cada grabación se observó que hay grupos con categorías similares que corresponden al mismo sector dentro del modelo de emociones de Russell. Esta selección se hizo por criterio propio al inferir la emoción y estado que el perro presenta ante las situaciones de ladrido inducido. Así se re-etiqueta la base de datos usando las cuatro categorías correspondientes a:

1. N-A Negativo activo
2. N-P Negativo pasivo
3. P-A Positivo activo

4. P-P Positivo pasivo

Por lo que se propone agrupar las etiquetas de la siguiente manera en la figura 23.

Stranger - Extraño	Entrando al departamento en ausencia del dueño
L-S1 - Extraño	Ladridos normales ante un extraño tocando el timbre
L-S2 - Agresivo	Ladridos muy agresivos ante un extraño dentro del hogar provocando al perro con gestos
L-S3 - Peligroso	Ladridos de temor ante un extraño atacando al dueño y el dueño pidiendo ayuda
Fight - Pelea	Entrenador alienta al perro a ladrar agresivamente y morder un guante de entrenamiento

Play - Juego	Se le pidió al dueño que jugara con el perro un juego habitua
Ball - Pelota	El dueño sostuvo una pelota a una altura de aproximadamente 1,5 m delante del perro
Food - Comida	El dueño sostiene el plato de comida para perros a 1,5 m delante del perro.
L-P - Juego	Ladridos durante juego utilizando los objetos o juguetes con los que normalmente juega
L- H - Bienvenida	Ladridos de llegada del dueño a casa

Alone - Solo	El dueño ató al perro a un árbol con una correa en el parque y se alejó, fuera de la vista del perro
L -A Temor	Ladridos por agresión al dueño en un parque fuera de casa
L- TA Triste	Ladridos de Tristeza/Ansiedad el propietario ata al perro con la correa a un árbol y se aleja de su vista.

Walk - Caminar	Se le pidió al propietario que se comportara como si se estuviera preparando para salir a pasear con el perro
L - PA - Paseo	Ladridos por estimulación al paseo

Figura 23. Agrupamiento de categorías basadas en modelo de emociones de Russell para modelo de contexto

Basándose en las condiciones en las cuales se grabó el ladrido para cada una de los contextos en las base de datos, se mapeó la etiqueta de contexto a la emoción que esté más relacionada dentro de las secciones del modelo de Russell, como se muestran en la figura 24.

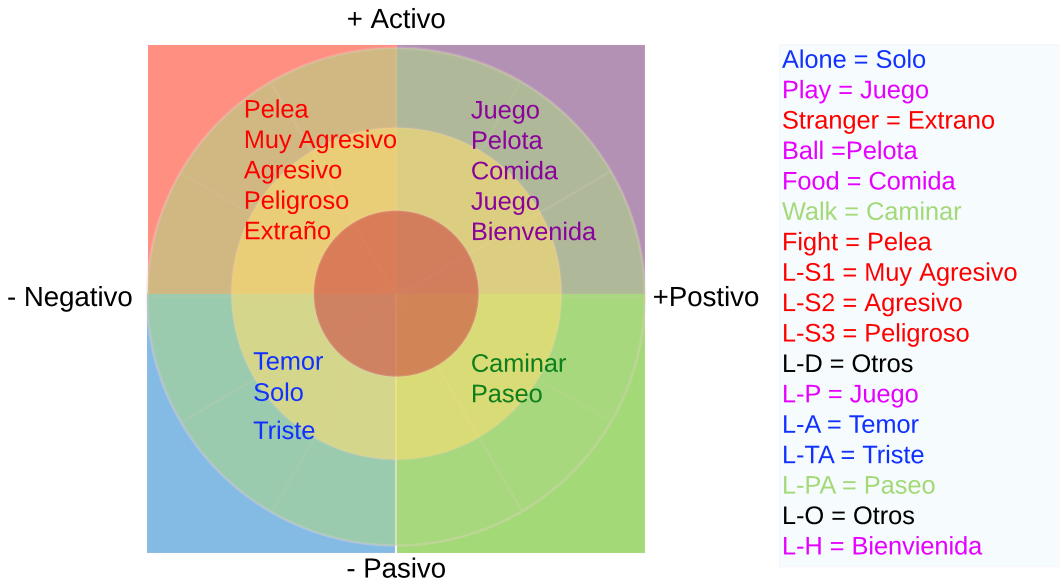

Figura 24. Asignación de etiquetas de contexto a modelo de emociones de Russell

Sin embargo, esta agrupación tiene un costo, ya que al usar cuatro grupos se reduce la cantidad de contextos a clases menos específicas. Con lo anterior el clasificador puede lograr un mejor desempeño. La distribución de los grupos de contextos basados en el modelo de Rusell se muestra en la figura 25.

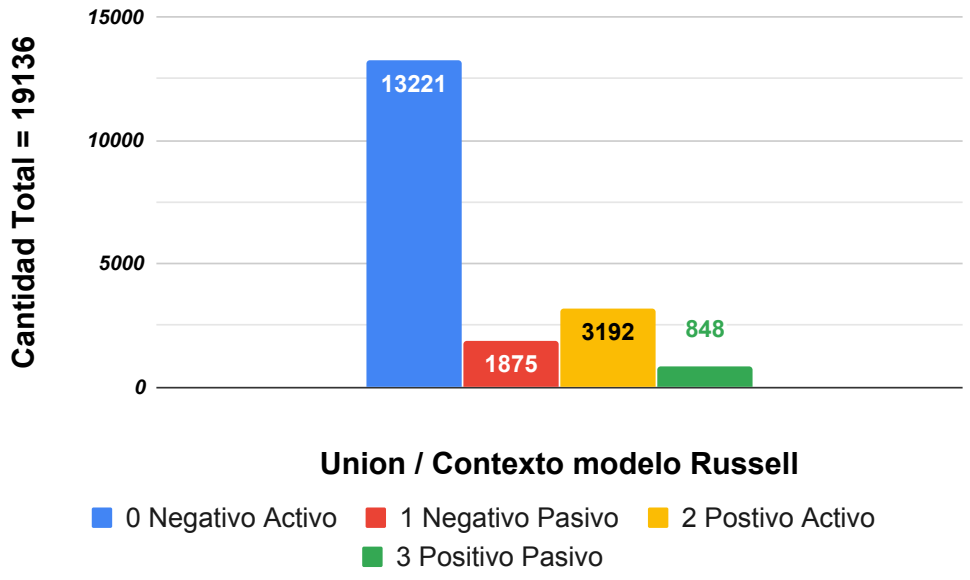

Figura 25. Distribución de clases para modelo de contextos de unión propuesta

3.1.8. Propuesta de modelo de clasificación de individuos

Para la tarea de clasificación de individuos la cantidad de etiquetas es de 113, como se muestra en la figura 26. Se descartaron las categorías con menos de tres muestras, debido a que al momento de dividir cada uno de los tres subconjunto es necesario tener por lo menos un caso por clase, de lo contrario se ocasiona un error.

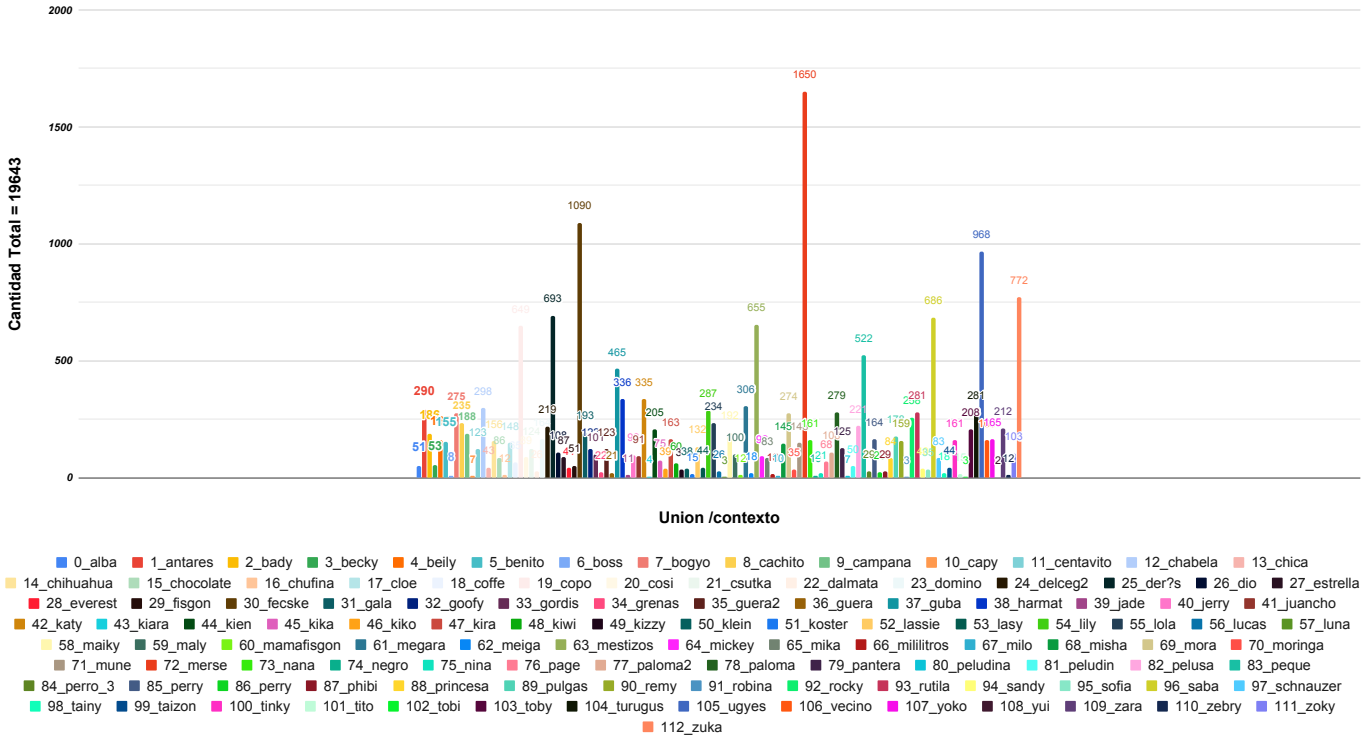

Figura 26. Distribución de clases para individuos de unión propuesta

3.1.9. Propuesta de modelo de clasificación de razas

La figura 27 muestra la distribución de razas en la base de datos de unión propuesta.

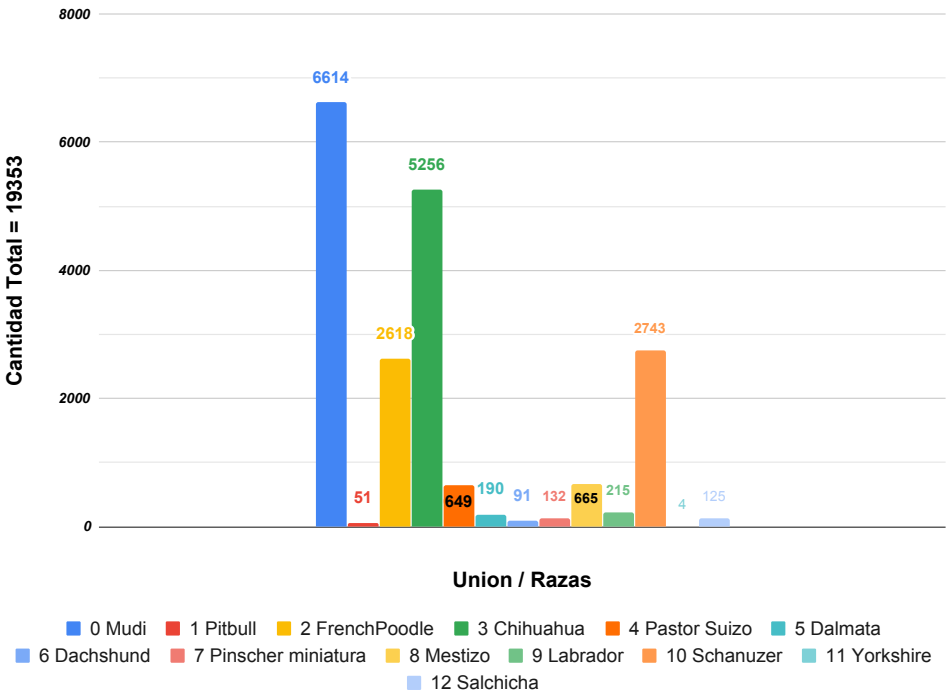

Figura 27. Distribución de clases para razas de unión propuesta

3.1.10. Propuesta de modelo de clasificación de sexo

La figura 28 muestra la distribución de sexo en la base de datos de unión propuesta.

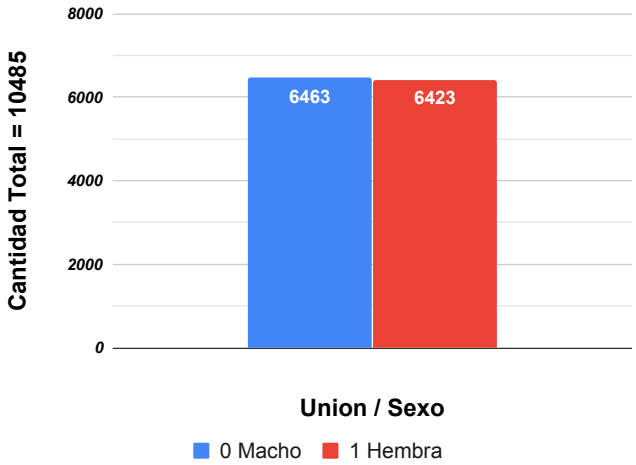

Figura 28. Distribución de clases para sexo de unión propuesta

3.1.11. Propuesta de modelo de clasificación de grupos de edad

Para crear el modelo de perros, se tomó de las bases de datos todas las muestras etiquetadas, ya sea con meses o con años. Las muestras se agruparon correspondiendo al tamaño de la raza en la categoría de acuerdo a la figura 29.

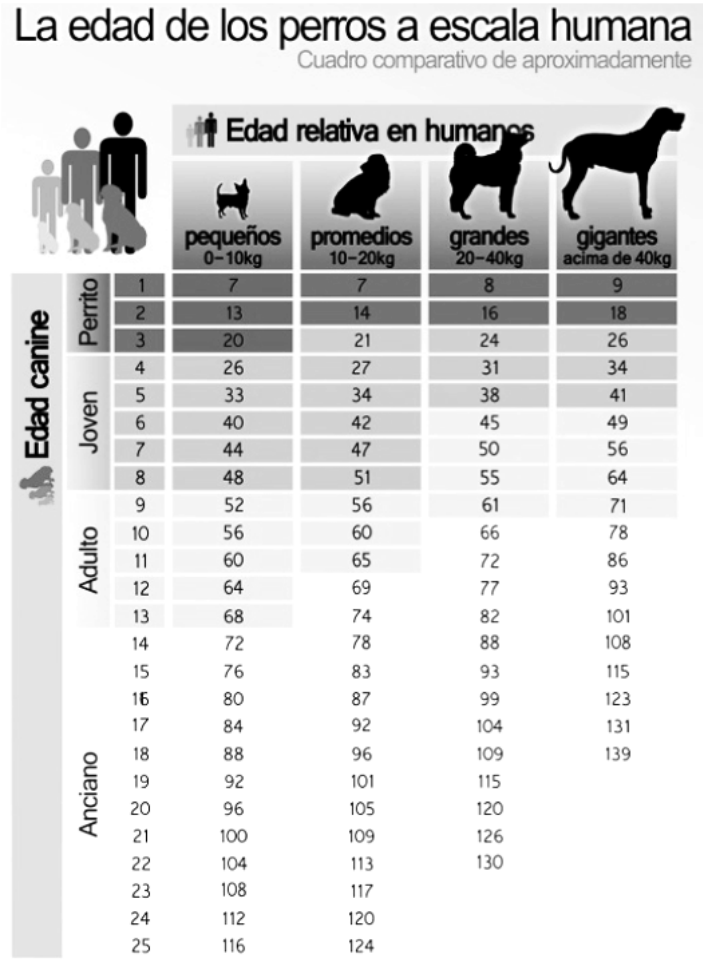

Figura 29. Tabla de grupos de edades correspondientes al tamaño del perro.

Podemos dividir a los perros en cuatro grandes grupos y con diferentes relaciones de equivalencia en años con respecto a un humano: los perros pequeños, que son los de hasta 10 kilogramos. Los promedios, entre 10 y 20 kilogramos. Los grandes, entre 20 y 40 kilogramos y los gigantes, los que superan los 40 kilogramos.

La figura 30 muestra la distribución de los grupos de edades para la base de datos de unión propuesta.

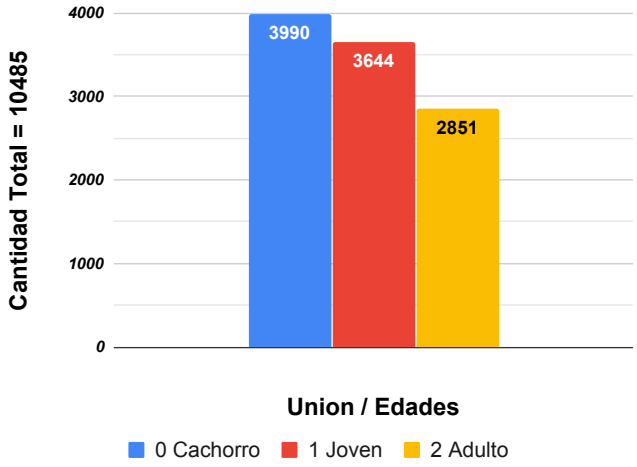

Figura 30. Distribución de clases para grupos de edad de unión propuesta

3.2. Preprocesado y preparación de archivos de audio crudo

Se reorganizaron los archivos en el directorio, ya que algunas bases de datos organizan sus archivos de maneras diferentes. Ahora, todos los archivos individuales de cada ladrido se encuentran en una sola carpeta y en el mismo formato. Ya que hay casos en los que la base de datos organiza los ladridos en subcarpetas que tienen en el nombre la información del grupo al que pertenecen, se recorren todos los directorios leyendo los archivos, añadiendo el nombre de la subcarpeta al nombre del archivo y guardándolos en una carpeta nueva.

En el momento de la lectura, puede ser que el archivo se encuentre en un formato distinto. Entonces, se convierte en archivo de extensión Wave (.wav), por medio la librería 'librosa' de python. Adicionalmente, para reducir el peso del archivo y facilitar el procesado, se reduce la tasa de muestreo original que en ciertos casos varía para la base de datos. Se busca la tasa de muestreo mínima que retenga la información importante del ladrido sin ninguna pérdida.

Según Sakamoto *et al.* (2014), los ladridos de todas las razas de perro tienen componentes de frecuencia entre 1000–2000 Hz. En (Frommolt, 2004), se registran chillidos de perros que varían entre 4000-5000 Hz. Para este trabajo se toma la frecuencia

más común de 44100 Hz y se divide entre cinco con lo que se tiene 8820 Hz. Basándose en el teorema de muestreo, al utilizar esta frecuencia de muestreo, queda la frecuencia máxima de 4410 Hz. Ási se asegura que se puedan reproducir todas las frecuencias menores a ésta, lo cual no debería dar problemas para el caso de los ladridos de perro. Sin embargo, hay una pérdida mínima para las frecuencias más altas, las cuales podrían contener información relevante de alguna otra vocalización más aguda del perro; no obstante, en este trabajo se tiene interés en los ladridos solamente. La figura 31 muestra el espectrograma de mel en dB del resultado de haber aplicado el proceso anterior en la señal.

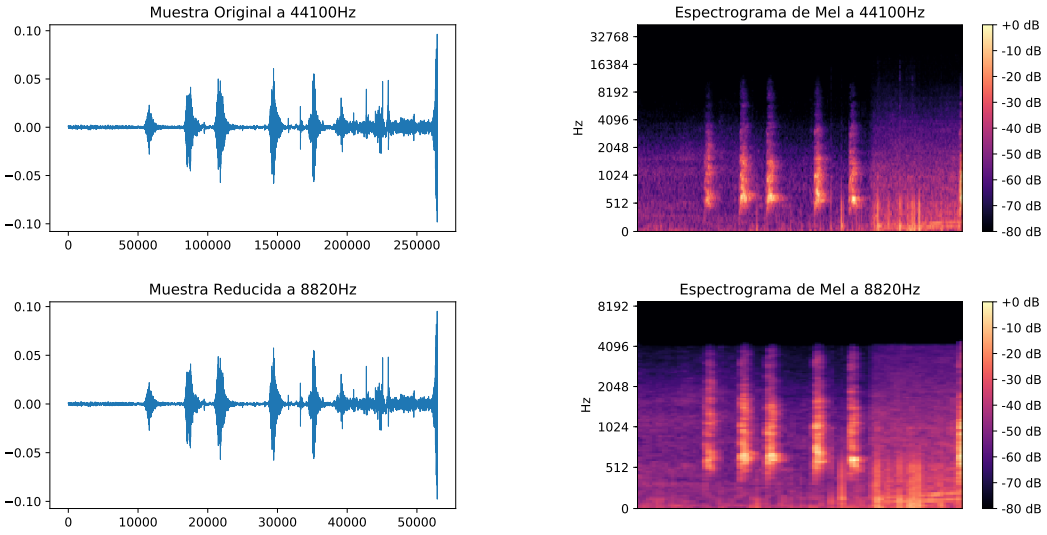

Figura 31. Reducción de la tasa de muestreo en archivo de audio que contiene ladridos de juego con duración de 6 segundos

El presente trabajo se enfoca principalmente en utilizar ladridos individuales, por lo que se extraen cada uno de estos eventos individuales de las grabaciones de audio más largas. Ya que la duración del ladrido es variable en el tiempo, se utilizó una ventana de un segundo como el tamaño predeterminado para todos los archivos. Como ningún ladrido individual dura más de un segundo, se toma el evento de ladrido capturado y se inserta en la ventana, donde el tiempo restante corresponde al silencio en valores de cero. Esto no debería ser ningún problema ya que las redes neuronales convolutivas tiene la característica de aprender el patrón de las datos independientemente de la localización o la región en que se encuentra el evento. Con ésto termina la preparación del audio crudo.

El programa comienza con la lectura de los datos mediante un archivo CSV (valores separados por comas), el cual se encuentra organizado de la siguiente forma:

Cada renglón corresponde a la longitud de la información del archivo de audio. El último valor corresponde a la etiqueta de la clase que corresponde ese archivo. De manera que la última columna contiene todas las etiquetas de cada archivo de audio

La longitud de renglones es igual a la cantidad de archivos de audio que existen en la base de datos. La longitud de las columnas menos uno, es el tamaño de la información extraída de cada archivo de audio, donde el último dato es la categoría enumerada a la que corresponde. Opcionalmente se genera un archivo separado que relaciona las categorías enumeradas con el texto que tiene el nombre de cada una, ésto con el fin de visualizar las clases por su nombre y no su número al momento de utilizar representaciones gráficas.

Para este programa, fue necesario preparar los datos con el formato anterior. De esta forma se facilita la lectura de distintas bases de datos y diferentes representaciones de la información que contiene el audio. Para este caso, el audio crudo debe parecer una matriz con longitud de 8821 columnas por el numero de renglones de cada archivo en la base de datos.

3.3. Procesado de audio para obtener espectrogramas en escala de Mel

Para obtener el espectrograma es necesario procesar cada archivo de audio individualmente, lo cual se realiza sobre cada archivo de audio crudo que anteriormente se preparó con longitud de un segundo y tasa de muestro en 8820 Hz. La operación se realiza con ayuda de la librería ‘Librosa’. En este caso se seleccionó un número de bandas de Mel de 128 para esta función, que es la máxima cantidad de bandas que se puede generar. Ésto define la cantidad horizontal del intervalo de valores que representan las frecuencias en escala de Mel, ya que una mayor cantidad da mayor resolución y eso beneficia encontrar pequeños detalles o cambios dentro de la imagen. Por lo cual se utilizó este valor de Mels por predeterminado.

```
librosa.feature.melspectrogram(y=audio_wav, sr=8820,mels=128)
```

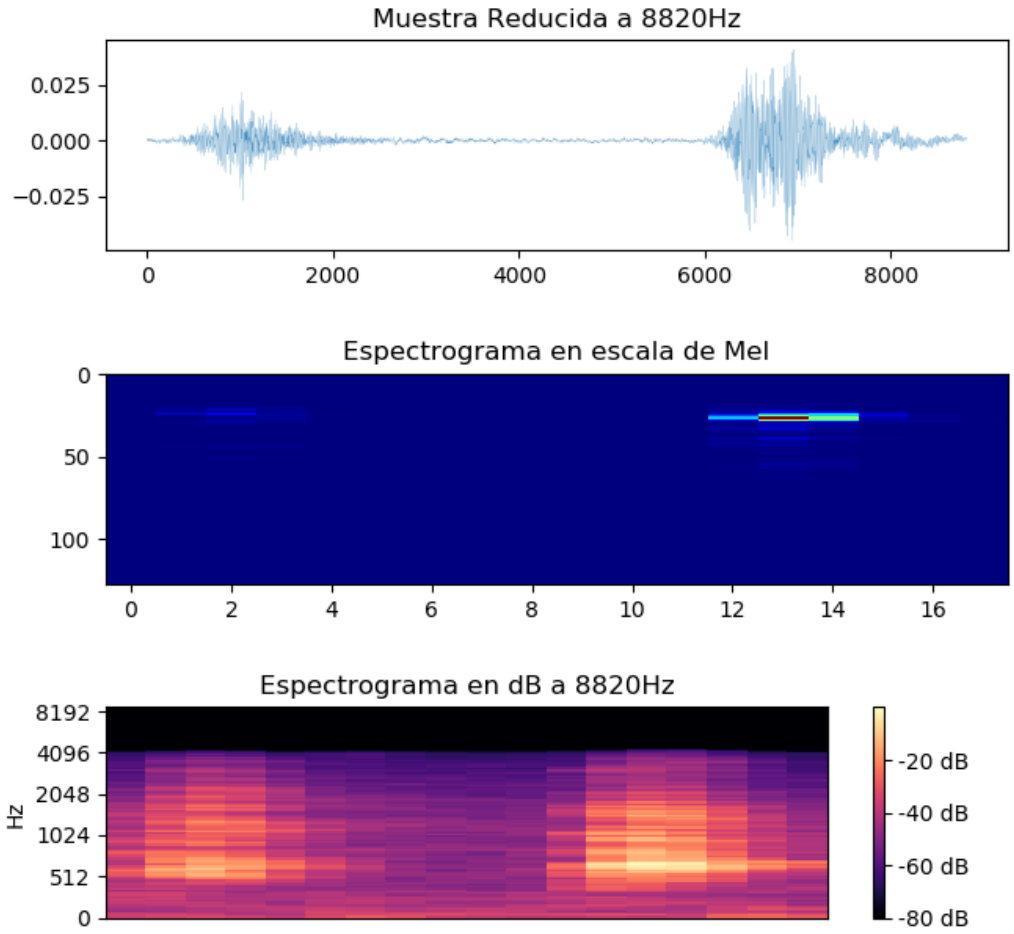

Figura 32. Conversión de audio a espectrogramas en escala de Mel

Una vez terminado el procesado, se recibe una matriz o imagen con las dimensión de 128 x 18, donde las 18 ventanas corresponden a un procesado de la señal con duración de 55.55 milisegundos. De esta forma, las 18 ventanas suman la duración total de un segundo, por otra parte, los 128 representan los valores dentro de la escala de Mel.

3.4. Algoritmo de caracterización utilizado

Existen muchos atributos extraíbles de la señal de audio. En algunos trabajos se utiliza el audio crudo de forma directa, por lo cual se evaluó su potencial al utilizarlo en las pruebas. También en otros trabajos se utiliza el espectrograma de las señal y se reportan mejores resultados al utilizar ese mismo espectrograma en escala de Mel. Dentro

de las evaluaciones realizadas se incluye esta técnica. Sin embargo, entre todas las distintas técnicas de caracterización que se evaluaron, ver tabla 2, se seleccionaron los descriptores acústicos, ya que demostraron tener un desempeño sobresaliente.

Los descriptores acústicos pueden utilizarse para diseñar especificaciones acústicas comparables y para verificar el efecto de los diferentes procedimientos con relación a la audición acústica de un fenómeno. Dentro de los descriptores acústicos existen características de varios niveles; de ellas, las características más cercanas a la señal se les conoce como de bajo nivel (LLDs). Las características de bajo nivel se relacionan a valores medibles de la señal y se suelen agrupar en familias dependiendo del funcionamiento del algoritmo con el que se extraen. Estas características se describen en la sección del marco teórico.

Para el presente trabajo de investigación, la extracción de estos descriptores se selecciona mediante un archivo de configuración con el software OpenSmile. En este software se presentan algunos archivos de configuración que contienen la lista de atributos utilizados en distintas tareas y trabajos publicados; algunos de éstos son: IS09, IS10, IS11, I-13, emobase, emo-large, chroma, entre otros.

En esta sección se explica cuáles de estos grupos de atributos son los de mayor aportación para el desempeño en este problema de clasificación. Para poder cubrir todo el rintervalo de atributos disponibles se utiliza emo-large, el cual contiene 6669 atributos = (57 LLDs + 144 funciones delta) × 39 funcionales. Este archivo de configuración contiene todos los descriptores acústicos que se encuentran por separado en los otros archivos de configuración. Los atributos extraídos se dividen de acuerdo a como se indica en la tabla 1:

A estos LLDs se adicionan los coeficientes delta de velocidad y doble delta de aceleración. Después de procesar todo lo anterior, se calculan los 39 funcionales estadísticos por cada uno los descriptores. Estos funcionales incluyen valores como la media, la desviación estándar, los percentiles y cuartiles, los funcionales de regresión lineal o los locales relacionados con mínimos / máximos. Lo anterior da un total de 6669 descriptores acústicos.

Tabla 1. Grupos de descriptores acusticos de bajo nivel extraídos por la configuración emo-large

57 descriptores acusticos de bajo nivel 57 LLDs = 13 MFCCs + 35 Espectro + 6 Energía + 3 Vocalización
Características Cepstrales (13)
MFCCs 0 - 12
Rasgos espectrales (35)
rangos de espectro espectral 0–25, tasa de cruce cero, 25 %, 50 %, 75 % y 90 % en puntos de caída espectral, flujo espectral, centroides, posición relativa del máximo y mínimo
Características de energia (6)
energía logarítmica, energía en bandas de: 0 a 250 Hz, 0 a 650 Hz, 250 a 650 Hz, 1 a 4 kHz, 3010 a 9123 Hz
Características relacionadas a las vocalizaciones (3)
F0 (suma subarmónica (SHS) seguida por el suavizado de Viterbi), envolvente del F0, probabilidad de cambio en vocalización

3.4.1. Reducción de características

Ya que trabajar con todos estos valores puede resultar muy tardado y complicado, se hace una prueba para buscar cuáles son los mejores y cuál sería el tamaño ideal. Con lo anterior se reduce la dimensionalidad del vector de características buscando mejorar el desempeño del clasificador y eligiendo las características que estén altamente correlacionadas con el problema.

Para encontrar los atributos más importantes, a través del software "Weka", se utiliza el algoritmo de evaluación de alivio de atributos ’ReliefFAttributeEval’, descrito en el marco teórico. Este Algoritmo evalúa el valor de un atributo muestreando repetidamente un punto en el espacio. Se considera el valor del atributo dado para el punto más cercano de la misma clase y la diferente.

Con el procedimiento anterior se obtiene un lista de las características de mayor

a menor calificación. De esta lista, se realiza una corte en diferentes grupos de LLDs para ver cúal tamaño elegir.

Con este corte se busca reducir el espacio de 6669 a los mejores 1000, 750, 500 y 250 atributos. Para ello, se utilizan distintos algoritmos de aprendizaje de máquina para la evaluación y se realiza la validación cruzada de 10 pliegues para todos los casos.

Se realiza una evaluación utilizando diversos algoritmos de clasificación con ayuda del software ’Weka’ para una rápida implantación. Entre ellos, se utilizan la máquina de vector de soporte (SVM), el bosque aleatorio (Random Forest) y el árbol de decisión J48. En la tabla 3 se pueden observar los resultados de la evaluación. Después de obtener los resultados de la evaluación, se toma el vector del tamaño que tuvo la mejor puntuación entre todos los demás. El corte de 500 mejores atributos es la mejor opción para utilizar en la siguientes pruebas, como se muestra en la figura 36 en la sección de resultados.

3.5. Algoritmo de clasificación

En este sección se plantea el uso de redes neuronales profundas para entrenar cada uno de los modelos para las tareas propuestas y usarlos para clasificar. Sin embargo, existen distintas arquitecturas de redes neuronales que podrían adaptarse a la naturaleza de los datos.

Existen tendencias populares que recomiendan usar las redes convolucionales para los problemas de reconocimiento de imágenes y vídeos. Por otra parte, se recomienda utilizar redes neuronales recurrentes para el reconocimiento del lenguaje natural hablado; sin embargo, nada establece firmemente que una sea mejor que la otra para casos específicos.

Por lo anterior, en este trabajo se realiza una comparativa entre ellas para el problema de clasificación de ladridos.

Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN) son buenas para procesar datos que se puedan representar en forma de secuencia para realizar las predicciones, pero tienen memoria de corto plazo. Para esta razón, las redes GRU y LSTM se crearon como un método para mitigar la memoria a corto plazo utilizando mecanismos llamados compuertas.

Las compuertas son elementos operacionales que regulan el flujo de información que fluye a través de la celda. Las GRU y LSTM se utilizan en aplicaciones de aprendizaje profundo como el reconocimiento de voz, síntesis de voz, comprensión del lenguaje natural. Lo único que cambia de una red a la otra es el tipo de unidad.

Red neuronal convolutiva

Las redes neuronales convolutivas se utilizan principalmente para clasificar imágenes, agrupar imágenes por similitud y reconocer objetos dentro de escenas. Su eficacia es la razón por la cual el aprendizaje profundo es famoso.

Se han aplicado directamente al análisis de texto y al procesamiento del lenguaje natural. También se han aplicado al sonido cuando se representa visualmente como en un espectrograma. Para este último caso, en este trabajo se prueba con imágenes y audio crudo como anteriormente se realizó con las otras redes.

Combinación entre distintas redes neuronales

Se tiene interés en juntar las etapas de estas redes en una red neuronal profunda compuesta de diferentes arquitecturas.

En primer lugar, con la intención de reducir el tamaño de características y conservar las características aprendidas de mayor aportación, se utiliza la red convolutiva. Después se agrega cualquiera de las dos redes recurrentes, GRU o LSTM, con el objetivo de aprender u olvidar las características dentro de la secuencia alimentada.

Por último, se agrega una red neuronal densa totalmente conectada que se encarge de reducir la cantidad de neuronas hasta la última capa, donde se realiza la clasificación. Para poder alimentar los datos de la red convolutiva a la red recurrente en forma de secuencia se utiliza la función de tiempo distribuido. Esta función se encarga de añadir una dimensión extra al tensor de los datos que corresponde al espacio tempo-

ral que utiliza la red recurrente para procesar los datos.

3.5.1. Evaluación de distintas redes

Para validar la selección de la arquitectura más adecuada entre las cinco redes anteriores: GRU, LSTM, Convolucional, ConvGRU y ConvLSTM, se evaluó el desempeño de cada arquitectura de red neuronal en esta lista para cada tarea. Los parámetros se seleccionaron manualmente hasta encontrar una configuración de valores estable con buen desempeño sobre los ejemplos de referencia como se muestra en la sección 4.4.

Después de realizar la evaluación para cada una de las cinco arquitecturas seleccionadas, se obtuvo (ver tabla 4) que la arquitectura más adecuada con base en su desempeño de clasificación es la red convolucional.

Además, para comprobar que el algoritmo de red neuronal convolutiva es la mejor elección entre diversos algoritmos de aprendizajes de máquina, se comparó con algunos de ellos, como lo son: SVM, bosque aleatorio y J48.

3.5.2. Afinación de redes

La mayoría de los algoritmos de aprendizaje automático incluyen hiper-parámetros, que son valores que controlan las variables dentro de las funciones del algoritmo. Estos valores se establecen antes de comenzar la construcción y entrenamiento del modelo. Es común seleccionar los valores por recomendaciones generales o predeterminadas antes de buscar optimizar los parámetros del modelo.

La configuración de los valores de hiper-parámetros se entiende como una selección específica para el modelo, es decir, qué modelo resulta más eficiente de todo el conjunto hipotético de modelos posibles. Los hiper-parámetros a menudo se configuran a mano mediante prueba y error; en ocasiones se seleccionan mediante un algoritmo de búsqueda o por una metodología desarrollada.

Las redes neuronales pueden tener muchos hiper-parámetros, incluidos aquellos

que especifican la estructura de la red en sí y aquellos que determinan cómo se entrena la red. El problema radica en que el espacio de hiper-parámetros es muy grande para explorar, por lo que con tiempo y recursos limitados se debe implementar alguna estrategia que encuentre una buena selección de forma eficiente.

Ciertos parámetros se eligen por experiencia, de la cual se pueden inferir los parámetros que afectan significativamente o no el desempeño; otros parámetros requieren seleccionarse con más cuidado, como los que definen la estructura de la arquitectura de la red. Se identifican ciertas estrategias propuestas para encontrar la mejor combinación de hiper-parámetros, las más comunes se listan a continuación:

1. Búsqueda manual:
   Se empieza con alguna combinación de parámetros y gradualmente se cambia cada parámetro hasta decidir la mejor configuración.
2. Búsqueda en cuadrícula:
   Generalmente se define un intervalo para cada hiper-parámetro y se buscan todas las combinaciones entre estos valores.
3. Búsqueda aleatoria:
   Se generan muestras aleatorias dentro de un intervalo determinado de hiper-parámetros.

En este trabajo se utiliza una estrategia de búsqueda en cuadrícula dividida en tipos de objetivos. Primero, se encuentra la mejor configuración de hiper-parámetros referentes a la arquitectura del modelo. Segundo, se encuentra la configuración óptima de parámetros relacionados al entrenamiento del modelo. Esta segunda configuración se obtiene una vez que se tienen los primero parámetros que definen la arquitectura de la red neuronal.

Debido a lo exhaustivo que puede ser intentar todas las combinaciones, se seleccionan solamente los parámetros más relevantes para cada tipo de objetivo. Los que se seleccionados por experiencia se explican a continuación.

3.5.3. Elección de hiper-parámetros

Hiper-parámetros relacionados al modelo:

A continuación se describen los hiper-parámetros relacionados a la arquitectura y la forma de la red neuronal. Estos parámetros influyen fuertemente en la cantidad de memoria que se requiere y en el tiempo de ejecución, ya que se correlaciona con la cantidad de operaciones requeridas.

1. Cantidad de unidades-neuronas

   ```
   hidden_neurons=[1000,750,512,256,128,90]
   ```

   Se usan los valores propuestos en esta lista; sin embargo, el intervalo máximo está limitado por la memoria de vídeo disponible y en el caso de la red GRU o LSTM es necesario reducirlos aún más. También existe la libertad de modificar y proponer valores conforme se observa a prueba y error; este parámetro es uno de los más laboriosos de ajustar. No existe un número determinado; sin embargo, se recomienda utilizar un intervalo entre el doble de neuronas de entrada como máximo y pocas más del doble de neuronas de la capa de salida como mínimo. No obstante pueden existir algunas excepciones para ciertos casos. El número de neuronas también va ligado a la varianza, que evita o causa el sobre-ajuste del modelo, por lo cual hay que mantener un balance. La insuficiencia se produce cuando hay muy pocas neuronas en las capas ocultas para detectar adecuadamente las señales en un conjunto de datos de mayor complejidad. El uso de demasiadas neuronas en las capas ocultas puede resultar en un sobre-ajuste.

2. Cantidad de capas ocultas

   ```
   dense_layers=[3,1,0]
   ```

   Este valor corresponde a números enteros positivos que representan la cantidad de capas ocultas contenidas en la red. Se recomienda utilizar los valores de la lista; sin embargo, están limitados por la cantidad de memoria de vídeo disponible. El número de capas ocultas está relacionado con la complejidad que se necesita para separar los datos del problema. Una red neuronal sin capas ocultas puede resolver fácilmente un problema de clasificación para dos o más clases que se pueden separar linealmente. Al aumentar el número de capas ocultas, aumenta

la flexibilidad de moldear los pesos a los datos que se presenten, lo cual también puede ser contraproducente ya que existe la posibilidad de un sobre-ajuste. La mayoría de los problemas que interesa resolver en este trabajo no son linealmente separables, por lo cual, aumentando la cantidad de capas ayuda a separar las clases más complejamente separadas.

```
-----------------------------------------------------------------------------------------
    #parámetros de convolución sólo se aplican si se selecciona la red convolucional
-----------------------------------------------------------------------------------------
```

3. Cantidad de capas convolutivas

```
conv_layers=[3,2,1]
```

   Este parámetro configura la cantidad de capas que realizan la operación de convolución. Se utiliza para reducir las características de los datos de entrada. Por lo menos se debe seleccionar una capa. Este parámetro tiene un costo computacional alto y puede llegar a ser tardado, por lo que debe seleccionarse el número de capas que ayude a tener un buen resultado sin que el entrenamiento demore mucho.

4. Tamaño de la ventana

```
kernel_size=[6,5,3]
```

   Este hiper-parámetro configura el tamaño de la ventana que se desliza sobre los datos de entrada realizando la convolución. Un menor tamaño se enfoca en características de menor área; por el contrario, un mayor valor de tamaño de la ventana abarca mayor área. Los valores deben ser enteros mayor a uno y se recomienda ir aumentando el valor hasta estar satisfecho con los resultados; si ya no hay mejora, se mantiene el valor, ya que se puede retardar el proceso de entrenamiento.

Parámetros relacionados al entrenamiento

A continuación se describe la configuración de los parámetros que van relacionados al entrenamiento del modelo. Estos parámetros se afinan después de que se encuentra la mejor combinación de hiper-parámetros que definen la arquitecturas de la red. Se debe tener cuidado al momento de elegir valores demasiados altos que puedan ocasionar un error de memoria insuficiente en el sistema de cómputo.

1. Tamaño del lote

```
#batch_size=[256,128,64,32]
```

   El tamaño del lote, principalmente, es una cuestión computacional. Se divide el conjunto de entrenamiento en un número de muestras que es preprocesado por el modelo. Puede ser cualquier valor numérico; sin embargo, por convención se maneja en potencias de dos. Este proceso va relacionado a la cantidad de memoria disponible en el GPU. Un tamaño más grande de lote termina los cálculos más rápido, pero ocupa mayor cantidad en la memoria. Esto limita la memoria de vídeo disponible en el GPU, lo que pueda causar un error en la ejecución por memoria insuficiente. Si se quiere tener resultados rápidamente, lo ideal es incrementar este valor hasta donde permita la memoria de vídeo. Este parámetro puede llegar a ocupar demasiada memoria comparado con todos los demás.

2. Drop out

```
drop_out=[0.7,0.5,0.2]
```

   Este parámetro se configura en la escala de ceroa uno. Se utiliza para evitar el sobre-ajuste y omite aleatoriamente una fracción de las neuronas que previamente conectan con la siguiente para reducir la dependencia a la activación de una neurona en específico. Este parámetro no influye en el peso de la memoria.

Hiper-parámetros descartados

Durante la selección de hiper-parámetros a escoger dentro de la afinación, algunos de estos parámetros no mejoran el desempeño.

1. Epocas

```
#epochs = 120
```

   Este parámetro de entrenamiento define el número de iteraciones completas del conjunto de entrenamiento en la red, puede ser cualquier valor numérico, pero se recomienda un valor alrededor de 45 a 75 para no sobre-ajustar la red. Ya que dentro de la opciones se utilizó la función ’early stopping’, no importa tener un valor muy alto dado que el entrenamiento de la red neuronal se detiene en cuanto no haya mejoras en la validación, evitando el sobre-ajuste.

2. Función de activación

```
#activation=['relu','softmax','tanh','sigmoid','exponential','linear']
```

   La experimentación con este parámetro va ligada al tipo de datos de entrada. Se considera si existen valores negativos o respuestas no lineares. Después de analizar el tipo de datos utilizados, se concluyó que la función de activación 'softmax' en la capa de salida y la función de activación 'relu' en las demás capas, son las mejores opción para las redes neuronales utilizadas ya que conserva un comportamiento lineal a partir de los valores positivos y se descartaron las otras funciones que no tuvieron mejora alguna.

3. Optimizador

```
#optimizer=['adam','SGD','rmsProp']
```

   Cada optimizador funciona con diferentes valores de tasa de aprendizaje. La afinación de este parámetro ayuda durante el entrenamiento a que el modelo converja más rápido. En la mayoría de locasos, el optimizador Adam termina siendo el más eficiente y, al utilizarlo, su función de aprendizaje ésta predeterminada para el mejor de los casos generales.

4. Tasa de aprendizaje

```
#learning_rate=[1e-2,1e-3,0.75e-3]
```

   Este valor define el tamaño del cambio en cada uno de los pasos, qué tan rápido o lento el modelo aprende el problema. Este parámetro va ligado a la función de optimización. Los valores en la lista están dentro de un intervalo estandarizado. Este parámetro determina que tán rápidamente las actualizaciones del gradiente siguen la dirección del gradiente. Si la velocidad de aprendizaje es muy pequeña, el modelo converge muy lentamente; si la tasa de aprendizaje es demasiado grande, el modelo diverge. Para las redes neuronales comunes se establece típicamente entre 0.1 y 0.00001.

5. Tamaño de reducción

```
pooling_size=[4,2,1]
```

Este valor ayuda a reducir el tamaño de los datos y reducir su dimensionalidad. Lo anterior ayuda a un procesamiento más ligero. Aunque también existe el riesgo de descartar valores que no entren dentro del promedio o máximos, lo cual repercute en pérdida de información de interés. Para el caso de los descriptores acusticos utilizados, que son datos ya reducidos, no conviene reducir aún más esta información.

6. Profundidad de filtros

```
filter_depth=[12,24,36]
```

Esta cantidad incrementa el tamaño del tensor con una dimensión adicional. Al añadir una mayor cantidad de filtros, se incrementa la cantidad de operaciones a realizar. Lo anterior aumenta el tiempo de ejecución del entrenamiento de la red neuronal convolutiva. Tampoco existe un regla que defina cuántos filtros se deben usar; por lo general, se incrementa este parámetro mientras ayude a mejorar la exactitud en la clasificación del modelo.

3.6. Diseño del experimento

Para buscar la mejor configuración de cada modelo es necesario comparar el desempeño entre distintas combinaciones de hiper-parámetros. En cada ejecución se prueba una configuración de valores distinta para los modelos de manera independiente. Dado que este proceso se vuelve laborioso, se automatiza la ejecución por medio de una lista que incluye todos los parámetros. Este proceso se ejecuta de manera iterativa, para cada modelo, siguiendo el criterio de afinación de parámetros descrito en las dos etapas anteriores.

```
Hiper-Parámetros-Convolución    ---------------------------------------------
                                conv_layers=[3,2,1]
                                kernel_size=[6,5,3]
Hiper-Parámetros                ---------------------------------------------
                                hidden_neurons=[1000,750,512,256,128]
                                dense_layers=[3,1,0]
Parámetros de entrenamiento     ---------------------------------------------
                                batch_size=[256,128,32]
                                drop_out=[0.7,0.5,0.2]
                                ---------------------------------------------
```

El proceso anterior genera 3x3x5x3 + 3x3 = 144 combinaciones para la redes convolutivas y 3x3x5x3 + 3x3 = 24 combinaciones para las redes recurrentes. Además, este proceso se realiza para cada una de los cinco modelos.

Este experimento registra las curvas de aprendizaje mientras se construyen los modelos mediante TensorBoard, como se muestra en la figura 33. Se puede analizar la exactitud en el conjunto de validación a través de las épocas del entrenamiento para cada combinación de parámetros.

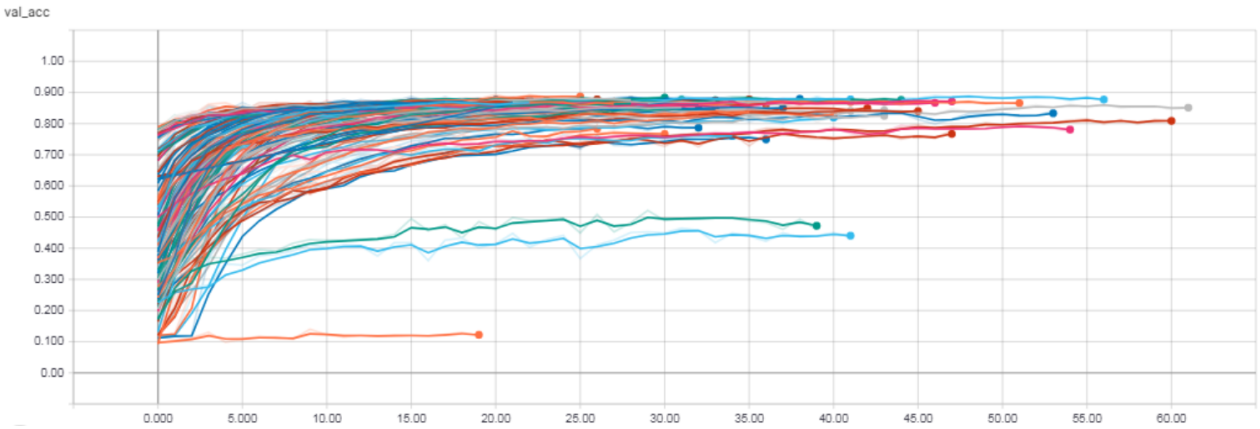

Figura 33. Exactitud en conjunto de validación para cada combinación a través cada época.

Con esto se supervisa el comportamiento de las redes. Aquí es donde se puede observar qué parámetros resultan en entrenamientos más rápidos e influyen en la mejora de la validación; así se crea un criterio que se utiliza para las futuras afinaciones. Una vez que se termina todo el proceso iterativo, se obtiene una lista con las evaluaciones con más alta exactitud sobre el conjunto de prueba, para cada modelo.

Una vez que se obtienen los resultados anteriores, se pueden utilizar las mejores configuraciones conseguidas para cada modelo; sin embargo, para ampliar la búsqueda de hiper-parámetros aún más afinados, se puede realizar una búsqueda ligera adicional.

Se propone otro intervalo de parámetros, que con base en la observación, se pueda inferir un intervalo alrededor de dónde se concentran los resultados con valores de exactitud más altos.

También, si el intervalo máximo o mínimo anteriormente propuesto limita encontrar una mejora en la exactitud, se extiende el intervalo. De esta forma se verifica si puede

o no mejorar aún más el modelo.

```
Hiper-Parámetros-Convolución    ---------------------------------------------
                                conv_layers=[4,3]
                                kernel_size=[3]
Hiper-Parámetros                ----------------------------------------------
                                hidden_neurons=[1200,1000,850]
                                dense_layers=[3,0]
Parámetros de entrenamiento     ----------------------------------------------
                                batch_size=[256]
                                drop_out=[0.5]
                                ----------------------------------------------
```

Los resultados de esta segunda búsqueda se presentan en la sección 4.6. De dichos valores, se toman los modelos finales ya entrenados con la mejor configuración de parámetros.

3.6.1. Herramientas y especificaciones técnicas

1. Lenguaje de programación: se utilizó Python como lenguaje de programación para este trabajo, ya que se ha convertido en el lenguaje más utilizado para redes neuronales. Actualmente existe gran cantidad de paquetes especialmente diseñadas para trabajar en el área de ciencia de datos y aprendizaje automático. Python, además de la versatilidad que se tiene para manipular los directorios al crear y leer archivos desde imágenes hasta audio, presenta también sintaxis sencilla que ayuda a leer claramente e implementar el código más eficientemente.

2. Paquetes (módulos de librerías) utilizados:

   a) Tensorflow-gpu: es una biblioteca de fuente abierta para computación numérica que usa el flujo de operaciones con tensores en un grafo, originalmente desarrollada por 'Google' para redes neuronales y aprendizaje de máquina. Tensorflow puede ejecutarse en procesadores CPU's, tarjetas gráficas GPU's y TPU's, que es un hardware especializado para realizar matemáticas con tensores. Un tensor se define como un simple arreglo de números multidimensional, como lo puede ser un vector o una matriz. Un tensor incrementa

su rango por la cantidad de dimensiones que presentan los datos. Tensorflow facilita las instrucciones para declarar la arquitectura y dimensión del modelo de una red neuronal junto con las operaciones que se realizan. Las instrucciones se distribuyen al motor de ejecución en C++ para que el hardware procese los datos.

b) Keras: es una interfaz de aplicación de alto nivel creada con la intención de facilitar al usuario trabajar con distintos motores de ejecución como: 'theano' y 'tensorflow'. Las instrucciones son sencillas y tiene una estructura modular para declarar cada capa del modelo y compilarlo. Además, Keras se puede extender con nuevas funciones. Keras es una interfaz que trata de unificar distintos ambientes de trabajo de redes neuronales en uno, haciendo más sencillo el desarrollo de modelos de aprendizaje profundo independientemente del motor de ejecución optimizado.

c) Numpy: es un paquete para python que agrega instrucciones que soportan operaciones de vectores y matrices. Esta biblioteca contiene más funciones matemáticas de alto nivel para operar con esos vectores o matrices.

d) Matplotlib: esta biblioteca sirve para generar gráficos a partir de los datos contenidos en listas, arreglos o matrices.

e) Pandas: es una biblioteca escrita como extensión de numpy para manipulación y análisis de datos en computación y ciencia de datos. La biblioteca ofrece estructuras de datos (data frames) y operaciones para manipular tablas numéricas, lo cual facilita operar con grandes bases de datos de manera más sencilla.

f) Sklearn: es una biblioteca orientada al aprendizaje automático para python. Sklearn incluye varios algoritmos de clasificación, regresión lineal y otras herramientas para facilitar la manipulación de los datos. Esta biblioteca trabaja también en conjunto con numpy.

g) Librosa: es un paquete de python especialmente diseñado para el análisis musical y de audio. Tiene los bloques necesarios para crear sistemas que obtengan la información del audio.

h) Picklet: este módulo contiene protocolos binarios para comprimir y descomprimir la estructura de un objeto de python.

i) Time: este módulo incluye las instrucciones que se encargan de todas las tareas relacionadas con el tiempo.

j) Operator: este módulo proporciona funciones que son equivalentes a las de los operadores de Python. Estas funciones son útiles en los casos en que ciertos valores deben almacenarse, usarse como argumentos, o devolverse como resultados de la función. Las funciones en operador tienen los mismos nombres que los métodos especiales.

3. Tensorboard: este software incluye un conjunto de herramientas de visualización que se vincula con todos los cálculos que se realizan con Tensorflow. El entrenamiento de una red neuronal profunda masiva puede ser complejo y confuso. Por lo tanto, para facilitar la comprensión, la depuración y la optimización de los programas, se puede usar TensorBoard para visualizar su gráfico, trazar métricas cuantitativas sobre la ejecución del gráfico y mostrar datos adicionales como las imágenes que pasan a través de él. TensorBoard funciona al leer los archivos de eventos de TensorFlow que contienen datos del resumen y se generan al ejecutar TensorFlow. Lo anterior se logra declarando en el gráfico de TensorFlow del que se desea recopilar datos y se decide qué nodos se desea anotar con las operaciones del sumario.

4. Weka: Es un software libre implementado en Java que contiene una colección de algoritmos para aprendizaje de máquina y para tareas de minería de datos. Estas herramientas incluyen la preparación, clasificación, regresión, agrupamiento, y visualización para datos, además de reglas de asociación para minar.

5. OpenSmile: es un software de fuente libre que contiene un conjunto de herramientas para analizar audio. Se utiliza en proyectos académicos y científicos, permite extraer una gran cantidad de espacios de características del audio en tiempo real, combina las características del reconocimiento de lenguaje hablado y la extracción de información musical. Esto se logra a través de una simple archivo de configuración.

6. Hardware: se trabajó con distintas computadoras y servidores, los cuales estaban equipados con distintos procesadores, capacidad de memoria Ram y GPU’s. El equipo utilizado tenia un nivel de rendimiento suficiente para realizar las tareas.

Las especificaciones son la siguientes: para entrenar las redes neuronales se utilizaron las tarjetas: nvidia tesla k20 6gb, nvidia gtx 980 ti 6gb, nvidia gtx 1050 ti 4gb, entre otras. Para los demás algoritmos y procesos se utilizó el procesador intel i7 de tercera generación y Ryzen 2400g. Además es importante contar con por lo menos, 16 Gb de memoria Ram para poder manipular esta gran cantidad de datos.

3.6.2. Descripción del método y funcionamiento

El método general propuesto se resume en la creación de cinco modelos optimizados para cada una de las tareas.

Se utilizó el método de caracterización más sobresaliente entre las evaluaciones realizadas. Se redujo el número de características a un tamaño manipulable que conserva la información más importante de la señal de audio.

Cada modelo se entrenó con la arquitectura de red neuronal más eficiente, utilizando la mejor combinación de hiper-parámetros para lograr la mejor exactitud en la clasificación del conjunto de prueba.

Una vez que se construyeron los cinco modelos, el resultado es un archivo por modelo que contiene los valores de la configuración final de los pesos para cada conexión de la red neuronal. Es con este modelo con el que se clasifican los datos de prueba.

Cada uno de los datos de prueba se compara con el modelo final para cada una de las tareas, obteniendo un resultado de predicción con un porcentaje de exactitud para la etiqueta de la tarea de clasificación correspondiente.

Lo anterior se realiza para cada una de las tareas de clasificación, aun si el dato que se prueba no contiene etiqueta para cierto modelo. Este modelo predice la etiqueta con mayor probabilidad de parecido.

3.6.3. Procedimiento para comparar algoritmos

Para comparar el algoritmo se realizaron pruebas individuales con las base de datos de perros Mudi y Mescalina 2015 y tareas de clasificación de individuos y contexto. También se comparó con cada uno de los distintos métodos de caracterización de la señal propuesto.

Cada una de esta pruebas se realizó usando la búsqueda de parámetros óptima para obtener la mejor afinación para cada tarea, cada base de datos y cada método de caracterización de la señal.

Se comparó el resultado de la exactitud de clasificación contra trabajos previos que utilizaron las mismas bases de datos en tareas similares. Los resultados se presentan en la sección 4.5, donde se pueden observar como en la mayoría de los casos, el uso de redes neuronales profundas sobresale. Sin embargo, al utilizar una menor cantidad de datos, los algoritmos de aprendizaje automático como los son: la maquina de soporte vectorial o el bosque aleatorio, pueden tener un muy buen desempeño. Estos algoritmos en ciertos casos superan a algunas redes neuronales que no tengan suficientes datos para entrenarse.

3.6.4. Métrica utilizada para evaluar la clasificación

Para la validación dentro del entrenamiento se utilizó un conjunto de validación que contiene 10 % de las muestras de la base de datos extraídas de manera aleatoria y manteniendo la misma distribución de clases. Similarmente, para la prueba final de resultados se utilizó un conjunto de prueba del 10 % de los datos tomados de la misma base de datos, aleatoriamente, y respetando la distribución de clases, como se observa en la figura 34.

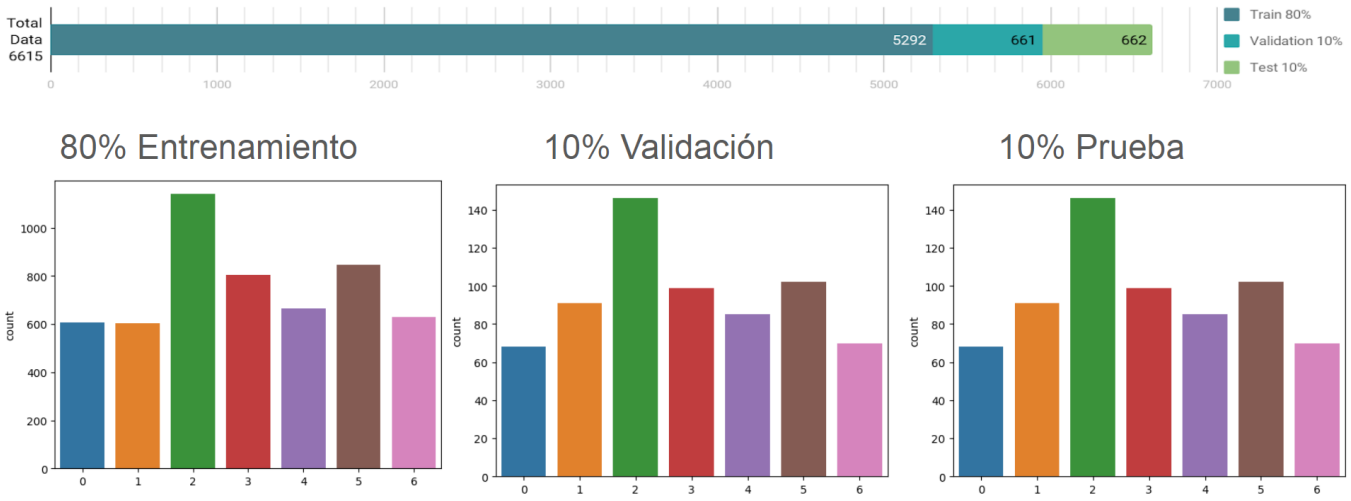

Figura 34. División de conjuntos de prueba, validación y entrenamiento, ejemplo utilizado en base de datos de perros Mudi.

La medida de exactitud utilizada es el porcentaje de aciertos sobre el conjunto de prueba. En las predicciones se utiliza esta medida de exactitud como el porcentaje de confianza que se tiene de que pertenezcan a cierta clase entre todas, donde se toma la de mayor valor.

Capítulo 4. Resultados

4.1. Resultados experimentales

A continuación se presentan todos los resultados preliminares de las pruebas previas al desarrollo del método final.

4.2. Resultado de evaluación de características

En esta sección se presentan las tablas e imágenes relacionadas con el análisis de cada una de las técnicas de caracterización utilizada sobre la señal de audio. El objetivo fue determinar cuál de todas estas técnicas es el mejor candidato para la clasificación. Los experimentos fueron seleccionados para las tareas y bases de datos de las que se tenia referencia con trabajos previos para poder comparar los resultados que se obtuvieron.

Tabla 2. Resultados de exactitud de clasificación con distintas técnicas de caracterización de la señal

Tipo de caracterización / Exactitud de clasificación	Audio crudo	Espectrograma en escala de Mel	Grupo de descriptores acústicos de bajo nivel
Red convolutiva Mudi Contextos	33.9 %	57.1 %	**84.6 %**
Red convolutiva Mescalina 2015 Contextos	58.5 %	76.5 %	**82.9 %**
Red convolutiva Mudi Individuos	23.9 %	88.3 %	**95.5 %**
Red convolutiva Mescalina 2015 Individuos	37.8 %	86.5 %	**87.7 %**

La tabla 2 muestra los resultados de exactitud en el conjunto de prueba para cuatro pruebas específicas de comparación que se seleccionaron con base en los previos trabajos: (Molnár *et al.*, 2008), (Larrañaga *et al.*, 2015), (Pérez *et al* 2015), (Pérez *et al* 2016), (Pérez *et al* 2018). Los cuales se utilizan como referencia para los resultados.

Se puede observar en la columna derecha cómo todos los casos los descriptores acústicos de bajo nivel sobresalen ante los otros métodos que se utilizaron. Por lo cual no fue necesario realizar más pruebas para las demás bases de datos y tareas de clasificación.

4.3. Reducción de dimensionalidad y análisis de complejidad

Se realizó una reducción de dimensionalidad para analizar la correlación entre los datos. Se buscó identificar si los datos se agruparon de forma visible.

En teoría esto sirve para revelar patrones en los datos y ayuda a la clasificación. Si en la gráfica se distinguen nubes de datos agrupados, donde se encuentran datos de la misma clase cercanos entre si, se vuelve a reagrupar y etiquetar los datos para mejorar el desempeño del clasificador.

Para esta prueba se utilizaron los datos de la base de datos de los perros mudi y la tarea de clasificación de contextos la cual contiene siete etiquetas: 0-extraño, 1-pelea, 2-caminata, 3-solo, 4-comida, 5pelota y 6-juego. La cual se seleccionó de referencia por que tiene una distribución equilibrada entre clases.

Primero se realizo el experimento utilizando el PCA ya que este algoritmo es más adecuado y eficiente para cuando se tiene una gran cantidad de datos.

Se puede observar en las figura de las siguientes páginas desde la figura 35 hasta la figura 40 cómo la dispersión de las clases es más compleja para el audio crudo y en el caso de los descriptores acústicos, el patrón es menos aleatorio.

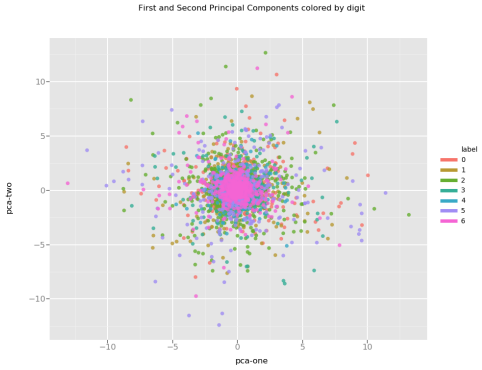

Figura 35. Resultados de PCA en segunda dimensión para audio crudo

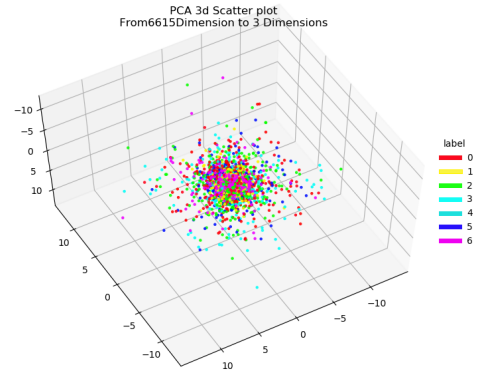

Figura 36. Resultados de PCA en tercera dimensión para audio crudo

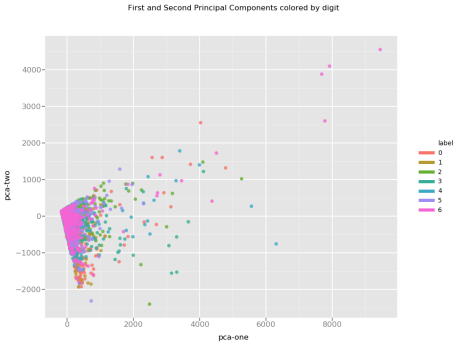

Figura 37. Resultados de PCA en segunda dimensión para espectrograma de Mel

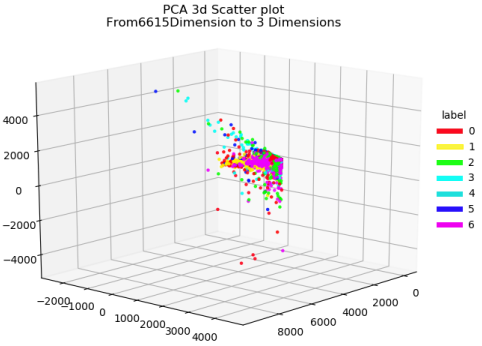

Figura 38. Resultados de PCA en tercera dimensión para espectrograma de Mel

Figura 39. Resultados de PCA en segunda dimensión para descriptores acústicos de bajo nivel

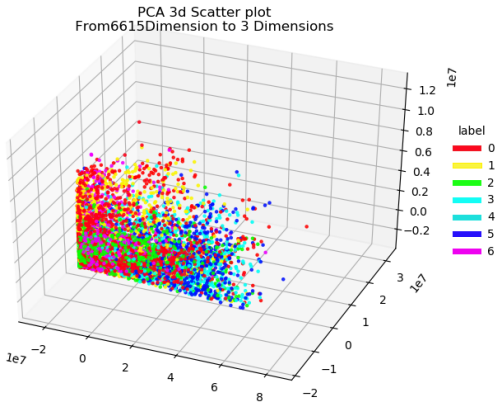

Figura 40. Resultados de PCA en tercera dimensión para para descriptores acústicos de bajo nivel

Después se realizo el experimento utilizando el t-SNE ya que este algoritmo es más flexible con los datos y ayuda a encontrar patrones en los datos con mayor complejidad, aunque requiere mayor cantidad de cálculos y se debe ajustar sus hiperparámetros, lo cual requiere mayor capacidad de cómputo.

En las figura de las siguientes páginas desde la figura 40 hasta la figura 46 se muestran los resultados de aplicar el t-SNE en ambos casos de segunda y tercera dimensión, la dispersión de las clases mejora, aun así, el audio crudo sigue siendo el caso más complejo y los descriptores acústicos, conservan un patrón menos aleatorio.

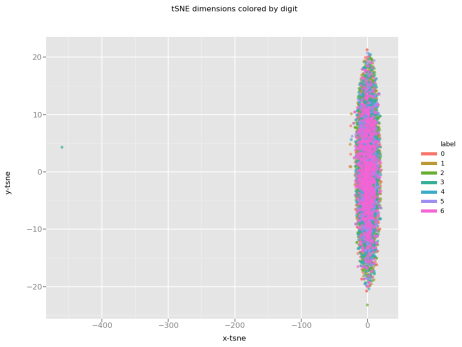

Figura 41. Resultados de tSNE en segunda dimensión para audio crudo

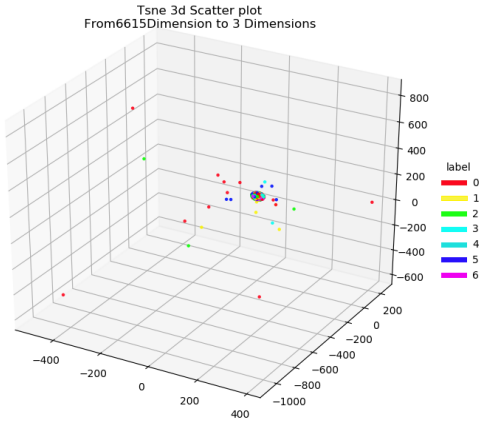

Figura 42. Resultados de tSNE en tercera dimensión para audio crudo

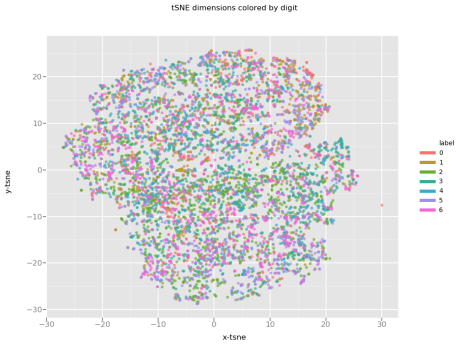

Figura 43. Resultados de tSNE en segunda dimensión para espectrograma de Mel

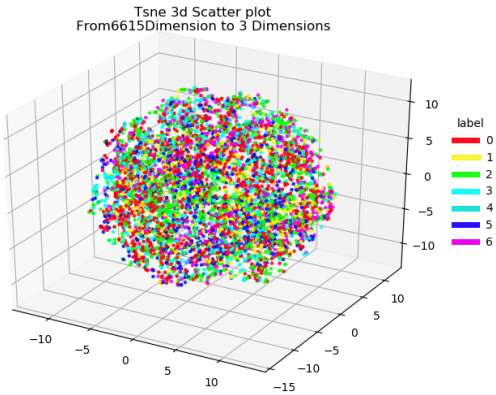

Figura 44. Resultados de tSNE en tercera dimensión para espectrograma de Mel

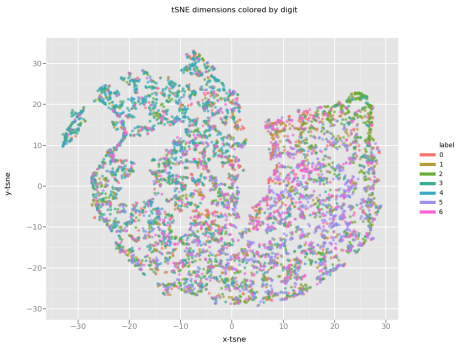

Figura 45. Resultados de tSNE en segunda dimensión para descriptores acústicos de bajo nivel

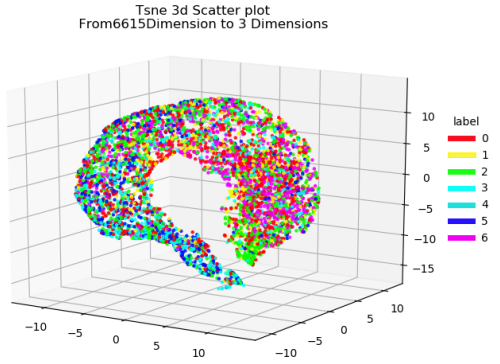

Figura 46. Resultados de tSNE en tercera dimensión para para descriptores acústicos de bajo nivel

4.3.1. Selección de descriptores acústicos de bajo nivel

Los resultados que se presentan a continuación se obtuvieron al utilizar distintos tamaños de reducción en los descriptores acústicos. En esta prueba se seleccionaron los intervalos de tamaño que se muestran en la tabla 3 con el objetivo de manipular un menor tamaño para ayudar al procesamiento y conservar un buen desempeño.

Cada porcentaje representa la exactitud de predicción de cada algoritmo de aprendizaje automático, mediante una validación cruzada de 10 pliegues, donde se clasificaron las etiquetas de contexto de la base de perros mudi. Se puede ver en esta tabla que los 500 mejores atributos son los que lograron mejores resultados.

Tabla 3. Evaluación de distintos tamaños de lista de atributos

Clasificador	250 mejores	500 mejores	750 mejores	1000 mejores
J48	53.1 %	54.1 %	52.9 %	52.3 %
SVM	53.1 %	55.2 %	53.3 %	54.0 %
RandomForest	73.6 %	75.3 %	73.2 %	74.8 %

En la figura 47 se muestra el resultado de la reducción de características extraídas, con lo cual se conservan sólo las de mayor correlación al problema. El resultado es un vector con longitud 500 conformado por distintos grupos de descriptores acústicos de bajo nivel. Cada uno de estos grupos se muestra en menor o mayor cantidad a través

del vector manteniendo el orden que se muestra.

Este vector es el resultado de la reducción y el algoritmo de selección utilizado. El vector se probó con la combinación de todas las bases de datos para que fuera una evaluación generalizada sobre todos los datos que se presentaron. Así se puede utilizar la misma selección de grupo y división en cada uno de los modelos.

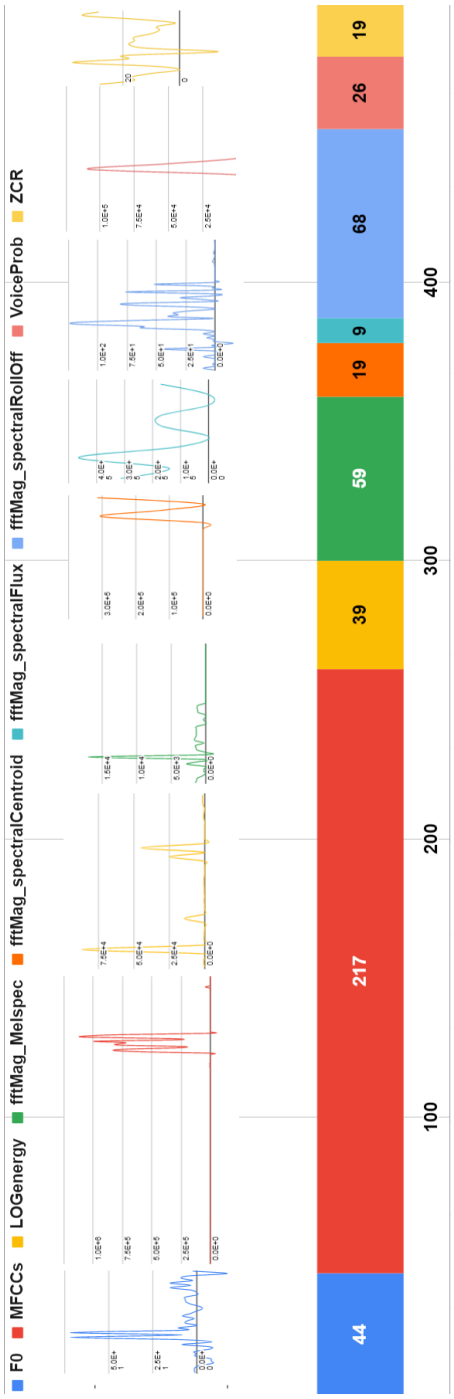

Figura 47. Distribución de diferentes grupos de descriptores a través del vector de los 500 descriptores mejor calificados.

4.4. Resultados de evaluación de redes neuronales

Para esta evaluación se presentan los resultados en la tabla 4. Cada renglón corresponde a cada una de las arquitecturas propuestas y en las columnas se mide su desempeño en la exactitud del conjunto de prueba para cada experimento.

Tabla 4. Resultados en la exactitud de clasificación con distintas arquitecturas y bases de datos

	Mudi contexto	Mudi individuos	Mescalina 2015 contexto	Mescalina 2015 individuos
GRU	65.2 %	69.3 %	58.5 %	58.9 %
LSTM	55.7 %	65.1 %	75.3 %	77.8 %
Conv	**84.6 %**	**95.5 %**	**82.9 %**	**87.7 %**
Conv-GRU	81.5 %	94.7 %	76.5 %	84.5 %
Conv-LSTM	77 %	93.8 %	80.3 %	86.5 %

Se observa que en la mayoría de los casos la arquitectura de red neuronal convolutiva (tercer renglón, con números en negritas) por si sola superó a las demás arquitecturas propuestas. Se concluye que esto sucede debido a que el formato de nuestra información sólo presenta un evento de ladrido individual. Las redes neuronales recurrentes se benefician cuando la información se presenta con un tipo de secuencia; un ejemplo es el caso del habla humano, donde las palabras se forman con una secuencia de vocales y consonantes que se repiten individualmente. La redes convolutiva aprenden diferentes características sobre la entrada unidimensional independiente de la posición, por lo cual el desempeño con el vector de descriptores acústicos fue satisfactorio.

4.5. Comparación de resultados con trabajos previos

Se realizó una comparación para verificar que la red neuronal implementada es mejor opción que usar algunos de los algoritmos de aprendizaje automático clásicos, como lo son: SVM, J48 y bosque aleatorio. Estas pruebas se realizaron para tareas específicas en las mismas bases de datos utilizadas en trabajos previos para poder comparar de una manera más adecuada a el experimento de clasificación.

Además, en previas investigaciones Pérez *et al* (2016) realizaron una evaluación de dichos algoritmos, por lo que se reprodujo dichos resultados.

La tabla 5 compara las bases de datos y las tareas previamente seleccionadas. En el lado izquierdo se tienen los resultados de exactitud de los experimentos realizados con el conjunto prueba de la base de datos, utilizando la red neuronal convolutiva que se implementó y se compara contra el lado derecho donde se muestran los resultados de clasificación de los trabajos previos, mencionando el método que se utilizó.

Tabla 5. Comparación de resultados presentes contra resultados en trabajos previos

Conjunto de datos utilizado	Exactitud / Algoritmo	Referencia del trabajo previo	Exactitud / Algoritmo
Perros Mudi 7 contextos	84.5 % CNN	Pérez *et al* (2015)	75 % SVM
Perros Mudi 11 individuos	95.5 % CNN	(Molnár *et al.*, 2008)	52 % Bayesiano (espectrograma)
Perros Mescalina 2015 4 contextos	82.9 % CNN	Pérez *et al* (2016)	79 % Bosque aletorio (LLDs)
Perros Mescalina 2015 37 individuos	87.7 % CNN	Pérez *et al* (2018)	90.5 % SVM (LLDs)
Perros Mudi sexo 3 grupos de edad 7 contextos 8 individuos	97 % sexo 95 % edad 85 % contexto 95 % individuo CNN	(Larrañaga *et al.*, 2015)	85 % sexo 80 % edad 55 % contexto 68 % individuo Naive Bayes, k-vecinos cercanos y regresion lineal

Se observa que en la mayoría de los casos el método que se desarrolló en este trabajo de investigación supera las implementaciones de los algoritmos utilizados en los otros trabajos; sin embargo, en el trabajo previo más reciente se logra un desempeño muy alto con el SVM. Para la investigación que se menciona se utilizan descriptores

acústicos muy parecidos a los que también se utilizaron con redes neuronales, pero cabe recordar que estas redes se benefician cuando se tiene una gran cantidad de datos disponible.

4.6. Resultados finales

Se presentan los resultados de exactitud de predicción en el conjunto de prueba para cada uno de los modelos utilizados. Los modelos que se presentan ya fueron afinados de manera óptima para cada una de las tareas que se presentan en las bases de datos.

Se utilizó la caracterización mediante descriptores acústicos y la clasificación con la red neuronal convolutiva para alcanzar el resultado final de esta investigación.

Los mejores resultados se reflejan en la tabla 6, donde se muestran los porcentajes de desempeño en la exactitud de predicción de las base de datos por separado y finalmente en el ultimo renglón, la de los modelos finales en la base de datos del método propuesto.

El formato del resultado se encuentra ya redondeado debido a que puede variar unos cuantos decimales al volverse a entrenar, siguiendo la metodología anterior. La idea es quedarse con el modelo con mayor porcentaje de confianza que se tuvo y utilizar dicho modelo en predicciones futuras.

Tabla 6. Resultados finales para cada modelo con afinación óptima

	Contexto	Individuo	Raza	Edad	Sexo
Perros Mudi	85 %	95 %	n/a	95 %	97 %
Mescalina 2015	80 %	85 %	88 %	91 %	91 %
Mescalina 2017	68 %	80 %	86 %	87 %	n/a
Método propuesto	**85 %**	**93 %**	**92 %**	**97 %**	**88 %**

En la tabla 7 se muestran los resultados de la búsqueda para encontrar la mejor configuración de hiper-parámetros para cada modelo. Así se obtuvieron conseguir los valores óptimos para las condiciones que se presentaron en el problema, además del tiempo y equipo con el que se hizo esta investigación. Estos valores pueden ser de guía para conseguir mejorar el desempeño en un modelo de características similares.

Tabla 7. Resultados de configuración de hiper-parámetros para cada modelo con afinación óptima

modelo	Número de neuronas	Número de capas ocultas	Número de capas convolutivas	Tamaño de lote (batch size)	Tamaño de ventana (kernel size)	Abandono (drop out)
contexto 85 %	1000	3	2	256	3	0.5
individuo 93 %	1000	3	2	256	3	0.7
raza 92 %	1150	0	2	512	5	0.5
edad 97 %	950	3	2	256	3	0.2
sexo 88 %	1000	3	2	256	5	0.5

Capítulo 5. Conclusiones

En este trabajo se probó el aprendizaje profundo para automatizar la caracterización junto con la clasificación directamente sobre el audio crudo. Se buscó descubrir cuáles son las características del ladrido que las redes neuronales pueden aprender directamente del audio sin extraerlas. Sin embargo, en la experimentación los resultados no fueron tan favorables para tal caso.

Uno de los puntos importantes del aprendizaje profundo es la gran cantidad de datos que se necesitan y la gran capacidad de cómputo necesaria para procesar grandes cantidades de información, lo cual se logra con equipo de supercómputo especializado. El audio crudo tiene la naturaleza de ser caótico, ya que varía en frecuencia, tiempo y amplitud y es difícil encontrar patrones estables. Sin embargo, muchos trabajos de investigación encuentran formas más adecuadas de representar la información contenida en el audio, como los son el espectrograma, MFCC's, espectrograma de Mel, promedios de amplitud, frecuencia fundamental y muchos más atributos extraíbles de la señal. Con ayuda de estas técnicas, la dificultad del problema se reduce y hace posible que ordenadores personales modernos, con poder de cómputo suficiente, sean capaces de realizar experimentos de este tipo.

Con técnicas de caracterización que se implementan manualmente, separadas del clasificador, aún se puede explotar el potencial que tiene el aprendizaje profundo. Por tal razón, este trabajo se orientó en resolver el problema en distintas etapas haciendo una evaluación de las mejores técnicas de caracterización y usando la mejor configuración de redes neuronales profundas para clasificar los datos.

Se analizaron distintas técnicas de caracterización y entre todas ellas sobresalieron los descriptores acústicos de bajo nivel. En esta investigación se concluye que los MFCC's, junto a las demás características, y utilizando una red convolutiva profunda, hacen que este método tenga un desempeño sobresaliente en comparación de resultados previos publicados por otros autores.

Con los descriptores acústicos de bajo nivel, como método de caracterización, se lograron buenos resultados; sin embargo, se conjetura que es posible mejorar el clasificador ya que las redes neuronales profundas se benefician de la aumentación de datos. En este caso, se contó con distintas bases de datos, que tenían el potencial de

incrementar la diversidad de información que se puede obtener de los ladridos con más categorías para diferentes tareas.

En la práctica, un clasificador del mundo real puede encontrar casos muy diversos y si no se entrena con alguno de esos casos, el clasificador no podrá acertar correctamente. Por lo que para cada tarea se analizó la manera de tratar con estos casos.

La tarea de clasificar el contexto del ladrido es el mayor reto, ya que este depende de la personalidad del perro y los contextos pueden producirse por emociones muy especificas. La mejor forma de abordar el problema es utilizando un modelo que agrupe los contextos a una definición más amplia y general de los estados emotivos del perro.

En la tarea de clasificar al individuo se obtuvieron buenos resultados. El reto yace en que si un modelo no incluye muestras suficientes o no las contiene, el individuo no se reconocerá y para una aplicación del modelo tendría que realizarse la parte de capturar datos de nuevos perros y registrar el nombre del individuo.

En la tarea de clasificación de raza se lograron buenos resultados para las etiquetas disponibles. Sin embargo, el reto yace en que existe una amplia lista de razas en el mundo real, la cual requeriría de un gran esfuerzo de colectar y registrar.

En la tarea de clasificación de edades se obtuvieron buenos resultados al agrupar por edades respectivas a la raza del perro. El reto yace en predecir la edad exacta en tiempo, lo cual es menos preciso y requiere de un modelo para cada raza.

En la tarea de clasificación del sexo del perro los resultados fueron bastante favorables, esta tarea parece que se pudiera resolver de una manera más directa. Sin embargo, la variación entre razas en los datos genera equivocaciones al momento de predecir.

Se propone que en trabajos futuros se siga incrementando la cantidad de datos y la variedad de etiquetas para las distintas tareas que se busca clasificar. También existe el potencial de utilizar otras representaciones del audio más complejas o incluso el audio crudo si se tiene el equipo de supercómputo suficiente. También podrían investigarse modelos más complejos de lenguaje y emociones para estos animales.

El mayor logro en este trabajo es que a pesar de que las bases de datos cuentan

con pocas muestras y clases desbalanceadas, se obtiene un gran desempeño con este método. Los logros más relevantes fueron los siguientes:

1. Trabajar con una gran cantidad de información, optimizando la distribución de los archivos y reduciendo el tamaño en el preprocesamiento.
2. Implementar diferentes técnicas de caracterización y encontrar la mas adecuada para extraer la información más útil con lo que se reduce el costo computacional.
3. Haber implementado exitosamente un clasificador con la arquitectura de redes neuronales profundas adecuada para la naturaleza del problema.
4. Haber implementado una estrategia de afinamiento de hiper-parámetros de redes neuronales para obtener el mejor desempeño en la clasificación, en un espacio de búsqueda muy grande.

Este método desarrollado se encuentra en el estado del arte respecto al problema de clasificación de ladridos, aporta resultados sobresalientes, y clasifica nuevas tareas antes no propuestas. El archivo del modelo contiene el mapa de los pesos aprendidos. Éstos se comparan con nuevos datos para realizar predicciones que pueden servir en la aplicación de un problema del mundo real. Los modelos de clasificación se pueden utilizar en la práctica con tecnologías destinadas al usuario, las que hacen uso de los archivos de cada modelo.

En conclusión, el método propuesto puede servir como una herramienta que facilita la identificación del ladrido del perro y reconoce al individuo, especie, edad, sexo y contexto asociado a cada ladrido. El método propuesto tiene la finalidad de estimar información que aporte un perfil más amplio del animal. Además, estos modelos pueden servir como base para futuras implementaciones que busquen ampliar las casos de clasificación con más diversidad de datos.

Literatura citada

Berridge y Kent, C. (2003). Comparing the emotional brains of humans and other animals. *Handbook of affective sciences*, pp. 22–51.

Davidson, Ekman, Saron, C., Senulis, J., y Friesen, W. (1990). Approach/withdrawal and cerebral asymmetry: Emotional expression and brain physiology. *Journal of Personality and Social Psychology*, **58**: 330–341.

Frommolt, Karl-Heinz Gebler, A. (2004). Directionality of dog vocalizations. *The Journal of the Acoustical Society of America*, pp. 561–5.

Gutiérrez-Serafín y bradly (2018). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, **17**(3): 715–734.

Kagan, J. (2003). Behavioral inhibition as a temperamental category. *Handbook of affective sciences*, pp. 320–331.

Larrañaga, Bielza, C., Pongrácz, P., Faragó, T., Bálint, A., y Larrañaga, P. (2015). Comparing supervised learning methods for classifying sex, age, context and individual mudi dogs from barking. *Animal Cognition*, **18**(2): 405–421.

Molnár, C., Kaplan, F., Roy, P., Pachet, F., Pongrácz, P., Dóka, A., y Miklósi, A. (2008). Classification of dog barks: a machine learning approach. *Animal Cognition*, **11**(3): 389–400.

Molnár, C., Pongrácz, P., Faragó, T., Dóka, A., y Miklósi, A. (2009). Dogs discriminate between barks: The effect of context and identity of the caller. *Behavioural processes*, **82**(2): 198–201.

Panksepp (1998). Affective neuroscience: The foundations of human and animal emotions. *Oxford University Press*.

Pongrácz, Molnar, C., y Miklosi, A. (2005). Human listeners are able to classify dog (canis familiaris) barks recorded in different situations. *Journal of Comparative Psychology*, **119**(2): 136–144.

Posner, J., Russell, J. A., y Peterson, S. B. (2005). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, **17**(3): 715–734.

ez2015

Pérez-Espinosa, H., Pérez-Martınez, J. M., Ángel Durán-Reynoso, J., y Reyes-Meza, V. (2015). Automatic classification of context in induced barking. *Research in Computing Science*, **100**: 63–74.

ez2016

Pérez-Espinosa, H., Himer Avila-George, J. R.-J., Cruz-Mendoza, H. A., Martínez-Miranda, J., y Espinosa-Curiel, I. (2016). Tuning the parameters of a convolutional artificial neural network by using covering arrays. *Research in Computing Science*, **121**: 69–81.

Pérez-Espinosa, H., Meza, R., Benitez, A., y Rosas, S. (2018). Automatic individual dog recognition based on the acoustic properties of its barks. **34**(5): 3273–3280.

Russell y James, A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, **39**(6): 1161–1178.

Sakamoto, Toyoshima, Y., Murayama, N., Miyairi, T., Hoshino, A., y Narumi, T. (2014). Sound attenuation devices for dogs barking (estimation of aperture ratio and experimental study of silencer). *International Journal of Mechanical Engineering and Applications*, (1): 18–24.

Shannon, C. (1949). Communication in the presence of noise. *Proceedings of the IRE*, (1): 10–21.

Yin, S. y McCowan, B. (2002). A new perspective on barking in dogs (canis familaris.). *Journal of Comparative Psychology*, **116**(2): 189–193.

Yin, S. y McCowan, B. (2004). Barking in domestic dogs: context specificity and individual identification. *Animal Behaviour*, **68**(2): 343–355.