



INGENIERÍA DE SONIDO

**Dereverberación del habla a partir de algoritmos
de aprendizaje profundo†**

**Autor: Martín Bernardo Meza
Tutor: Ing. Leonardo Pepino**

(†) Tesis para optar por el título de ingeniero/a de Sonido.

Noviembre 2021

Índice

1. Introducción	1
1.1. Fundamentación	1
1.2. Objetivos	2
1.2.1. Objetivo general	2
1.2.2. Objetivos específicos	2
1.3. Estructura de la Investigación	2
2. Estado del Arte	4
3. Marco Teórico	7
3.1. Representación temporal y frecuencial de señales	7
3.1.1. Transformada de corto término de Fourier (STFT)	8
3.2. Respuesta al impulso y reverberación	11
3.2.1. Relación directo-reverberado (DRR)	14
3.3. Inteligibilidad y parámetros de calidad de percepción	15
3.3.1. Relación energía de modulación de voz a reverberación	16
3.3.2. Inteligibilidad objetiva de corto término extendida	16
3.3.3. Relación señal a distorsión	16
3.4. Redes neuronales y algoritmos de aprendizaje profundo	17
3.4.1. La neurona artificial	17
3.4.2. Modelos basados en redes neuronales	18
3.4.3. Entrenamiento y aprendizaje	19
3.4.4. Redes neuronales convolucionales	23
3.4.5. Autoencoders	27
3.5. Dereverberación por filtrado temporal-frecuencial	28
3.5.1. Máscaras de amplitud	28
3.5.2. Síntesis de audio a partir de espectrogramas	29

4. Metodología	32
4.1. Análisis de datos	32
4.2. Base de datos de respuestas al impulso	32
4.2.1. Respuestas al impulso reales	33
4.2.2. Respuestas al impulso generadas	34
4.2.3. Respuestas al impulso generadas por aumento	36
4.3. Bases de datos de señales de habla	39
4.3.1. Pre-procesamiento de datos	40
4.4. Modelo propuesto	41
4.5. Detalles de implementación	43
4.6. Evaluación del modelo	45
4.6.1. Combinaciones de bases de datos	45
4.6.2. Ordenamiento de los datos durante el entrenamiento	46
5. Resultados y Discusiones	48
5.1. Bases de datos de respuestas al impulso	48
5.2. Análisis cualitativo del sistema	50
5.3. Tratamiento de la Fase	53
5.4. Dereverberación del habla y manejo de datos	56
5.5. Aprendizaje por currículum	60
6. Conclusiones	65
7. Líneas futuras de investigación	66
Bibliografía	68
Anexos	75
A. Aumentación de tiempo de reverberación	75

Índice de figuras

1.	Proceso de obtención de la STFT. Extraído y adaptado de [38].	9
2.	Ventanas (a) rectangular, (b) triangular y (c) Hann, con sus respectivas respuestas en frecuencia. Extraído de [39].	10
3.	Espectrograma de una señal de audio.	10
4.	Efecto del solapamiento entre ventanas en la STFT.	11
5.	Secciones temporales de una respuesta al impulso. Extraído de [40].	13
6.	Modelos analíticos de cálculo de la respuesta al impulso de un recinto. Extraído de [40].	14
7.	Esquema de neurona artificial.	17
8.	Funciones de activación y sus derivadas. Extraído de [26].	18
9.	Ejemplo de red neuronal multicapa con alimentación hacia adelante.	19
10.	Diagrama de flujo del bucle de entrenamiento de una red neuronal.	21
11.	Funcionamiento de un filtro de convolución. Extraído de [56].	24
12.	Representación de la aplicación de un filtro bidimensional sobre una imagen. .	25
13.	Principales hiperparámetros de una capa convolucional. Extraído de [26]. . .	26
14.	Capas convolucionales con campos receptivos locales rectangulares.	26
15.	Estructura general de un autoencoder.	27
16.	Disposición de las entradas y salidas del modelo para la estimación de las máscaras de amplitud.	29
17.	Diagrama en bloques del algoritmo de Griffin-Lim.	30
18.	Recinto greathall y esquema de medición de respuestas al impulso.	33
19.	Recinto octagon y esquema de medición de respuestas al impulso.	33
20.	Recinto classroom y esquema de medición de respuestas al impulso.	34
21.	Recintos y puntos receptor-fuente generados para la simulación de respuestas al impulso.	35
22.	Señales involucradas en el proceso de aumento de DRR.	38
23.	Esquema básico de una red tipo 'U-NET' con conexiones de salto.	42
24.	Modelo de red neuronal convolucional implementado.	43

25. Conjunto de respuestas al impulso reales.	48
26. Conjunto de respuestas al impulso generadas.	49
27. Conjunto de respuestas al impulso aumentadas.	49
28. Ejemplo de procesamiento de audio reverberado.	51
29. Diferencias entre espectrograma anecoico y dereverberado.	52
30. Espectrogramas de magnitud y fase de los audios para entrenamiento.	53
31. Influencia del número de iteraciones del algoritmo de Griffin-Lim.	55
32. Variaciones de SDR para el primer conjunto de pruebas.	57
33. Variaciones de SRMR para el primer conjunto de pruebas.	57
34. Variaciones de ESTOI para el primer conjunto de pruebas.	58
35. Variaciones de SDR para el segundo conjunto de pruebas.	59
36. Variaciones de SRMR para el segundo conjunto de pruebas.	59
37. Variaciones de ESTOI para el segundo conjunto de pruebas.	60
38. Respuestas al impulso generadas por aumentación.	61
39. Comparación de SDR entre tipos de ordenamiento de datos durante el entrenamiento.	62
40. Comparación de SRMR entre tipos de ordenamiento de datos durante el entrenamiento.	62
41. Comparación de ESTOI entre tipos de ordenamiento de datos durante el entrenamiento.	63
42. Descomposición temporal de la respuesta al impulso a procesar durante la aumentación.	76
43. Banco de filtros Butterworth.	77
44. Sub-bandas obtenidas luego de aplicar el banco de filtros.	78
45. Estimación paramétrica de la pendiente de caída.	80
46. Respuesta original y extendida sin piso de ruido.	81
47. Aumentación del tiempo de reverberación alterando la envolvente de caída original.	81
48. Resultado del proceso de aumento del tiempo de reverberación de una respuesta al impulso.	82

Índice de tablas

1.	Tiempos de reverberación por bandas para cada recinto.	34
2.	Especificaciones de la arquitectura implementada.	44
3.	Configuración del primer conjunto de pruebas.	46
4.	Configuración del segundo conjunto de pruebas.	46
5.	Comparación de métodos de reconstrucción de espectrograma complejo para generar audio.	56
6.	Resultados de las métricas sobre los conjuntos reverberados.	57
7.	Medianas correspondientes a cada esquema de entrenamiento.	63

RESUMEN

En este trabajo se estudia la dereverberación de señales de habla a partir de algoritmos de aprendizaje profundo. Se implementa una red neuronal convolucional tipo autoencoder con conexiones de salto, basada en el estado del arte actual, para estimar máscaras de amplitud que realicen la dereverberación del habla en el dominio de la transformada de tiempo corto de Fourier. Uno de los problemas de esta tarea es la escasa cantidad de datos disponibles, por lo que se analizan técnicas de generación y aumentación de datos, evaluando su impacto en el desempeño del sistema. Además, se evalúa el efecto que tiene el ordenamiento de los datos de entrenamiento y el tratamiento de la información de fase. Los resultados indican que las técnicas de generación y aumentación de datos permiten mejorar el rendimiento final del sistema. A su vez, ordenar los datos de entrenamiento con un tiempo de reverberación creciente tuvo un impacto positivo en las métricas de evaluación. Por último, se proponen mejoras al enfoque utilizado, y líneas futuras de investigación.

Palabras clave: “Dereverberación del habla”; “Redes Neuronales Convolucionales”; “Respuestas al Impulso”

SUMMARY

In this research, speech dereverberation based on deep learning algorithms is studied. A convolutional neural network model called autocoder with skip connections is implemented, following the current state of the art. The neural network is developed to estimate amplitude masks that perform speech dereverberation in the short-time Fourier transform domain. One of the issues of this particular task is the lack of a large training dataset, so generation and augmentation techniques were analyzed, evaluating the impact on model's performance. In addition, ways to handle phase information and data during training were studied. The results show that the generation and augmentation techniques allow to improve the model's performance. Moreover, sorting the training data from smallest to largest reverberation time results in better evaluation metrics. Finally, improvements for the implemented model and future lines of research were proposed.

Keywords: "Speech dereverberation"; "Convolutional Neural Networks"; "Room Impulse Response"

AGRADECIMIENTOS

Esta tesis es el resultado de un año de trabajo con el que culmina un camino de formación profesional de 6 años. A lo largo de este camino, pude conocer, trabajar y verme acompañado de numerosas personas a las cuales me gustaría agradecer.

En primer lugar dar gracias a la Universidad Nacional de Tres de Febrero (UNTREF), a su Rector Lic. Anibal Jozami, a los docentes de la carrera de Ingeniería de Sonido y a su coordinador Ing. Alejandro Bibondo.

A Leonardo Pepino, que dispuso desmedidamente de su tiempo para transmitirme conocimientos y ayudarme a organizar las ideas que conforman este trabajo.

A mis compañeros y amigos, por alentarme y siempre interesarse por los proyectos en los que me involucro.

A la comunidad online, que aporta tiempo y dedicación a la generación de conocimiento libre y gratuito.

Por último, pero no menos importante, a mi familia. En especial a mis padres, Jorge y Mónica, por inculcarme desde pequeño el hábito del estudio, por el sacrificio que hicieron durante estos años para poder brindarme una educación de calidad y por el apoyo y confianza que me hicieron sentir desde el momento en que decidí venir a Buenos Aires a estudiar esta carrera. Al resto de mi familia, hermanos, primos, tíos, que se alegraron y festejaron conmigo cada pequeño logro a lo largo de estos años.

Martin Bernardo Meza

CAPÍTULO 1: INTRODUCCIÓN

1.1 FUNDAMENTACIÓN

Las tecnologías de procesamiento digital de señales de voz mostraron grandes avances en las últimas décadas, llegando a ocupar un rol importante en nuestro día a día. Las investigaciones realizadas en este campo impulsaron diversas aplicaciones basadas en el análisis de la voz humana [1][2]. Estas aplicaciones, en mayor o menor medida, deben lidiar con una característica intrínseca a cualquier emisión sonora dentro de un recinto: la reverberación. Esto se debe principalmente a que las señales de voz se obtienen a partir de un transductor que no siempre se encuentra cercano a la fuente que se desea registrar, provocando que la señal registrada contenga la reverberación propia del entorno. Esta reverberación interfiere con la señal de voz, produciendo una reducción en el rendimiento de aquellas aplicaciones que dependen de la integridad de dicha señal, como por ejemplo: Reconocimiento del habla [3], verificación del hablante¹ [4] y localización del hablante [5].

Si bien esta problemática fue abordada desde el enfoque de diversas técnicas de procesamiento de señales, en los últimos años ocurrieron grandes avances producto de la implementación de una tecnología emergente de amplio crecimiento en el ambiente científico: los algoritmos de aprendizaje profundo. La capacidad y robustez de estos métodos a la hora de resolver problemas pertinentes al procesamiento de imágenes y detección de patrones se vio también reflejada en el campo de la dereverberación de habla. Actualmente, los sistemas basados en modelos de aprendizaje profundo representan el estado del arte, tanto en dereverberación del habla como otras tareas de audio, desplazando a enfoques más clásicos del procesamiento de señales. Sin embargo, aún quedan desafíos por resolver, como por ejemplo: la falta de bases de datos masivas de señales acústicas específicas como respuestas al impulso reales, la selección de una representación óptima de las señales que permita explotar sus características intrínsecas, las formas de evaluar y cuantificar el desempeño de los sistemas, entre otros.

Por este motivo, esta investigación pretende realizar un análisis de estas problemáticas desde el punto de vista de la ingeniería de sonido, para comprender las limitaciones de los modelos

¹Debe distinguirse entre reconocimiento del habla y verificación del hablante. Lo primero refiere a poder distinguir qué palabras fueron dichas, y lo segundo refiere a identificar quién es el que está pronunciando las palabras.

actualmente utilizados en este campo de estudio, analizar alternativas posibles a la escasez de datos y poder aportar al progreso y mejora del rendimiento de dichos modelos.

1.2 OBJETIVOS

1.2.1 Objetivo general

El objetivo general de esta investigación es implementar un algoritmo de dereverberación de señales de voz a partir del uso de redes neuronales y algoritmos de aprendizaje profundo.

1.2.2 Objetivos específicos

- Realizar una revisión de las técnicas utilizadas para resolver el problema de dereverberación.
- Diseñar e implementar una arquitectura de red neuronal para dereverberación de señales de voz en lenguaje Python.
- Analizar técnicas de pre y post procesamiento de datos, estudiando el impacto que tienen en el rendimiento del algoritmo.
- Optimizar el sistema propuesto, y evaluar los resultados obtenidos de manera objetiva.
- Estudiar y analizar técnicas de generación y aumentación de datos, evaluando su impacto en el desempeño del sistema implementado.

1.3 ESTRUCTURA DE LA INVESTIGACIÓN

En el capítulo 2 se presenta el estado del arte referido a las técnicas de dereverberación de señales del habla. En el capítulo 3 se detalla el marco teórico necesario para el seguimiento y comprensión de este trabajo. En este se abordan tres temáticas principales: la representación de señales de audio en el dominio espectral mediante la transformada de tiempo corto de Fourier, el concepto de reverberación y su relación con la respuesta al impulso, y por último la aplicación de redes neuronales convolucionales y algoritmos de aprendizaje junto con las principales técnicas de procesamiento. En el capítulo 4 se especifica de manera detallada la

metodología seguida a lo largo de este trabajo, y se brinda toda la información necesaria para replicar los experimentos realizados. En el capítulo 5 se presentan los resultados de los experimentos y se hace un análisis crítico de los mismos. En el capítulo 6 se exponen las conclusiones generales del trabajo, y por último, en el capítulo 7 se proponen líneas futuras de investigación relacionadas con el presente trabajo.

CAPÍTULO 2: ESTADO DEL ARTE

En los últimos años se ha registrado un marcado desarrollo y progreso en el campo del procesamiento de señales del habla. En este campo, la dereverberación ocupa un rol crucial, debido al impacto negativo que genera la presencia de reverberación en muchas aplicaciones.

Los primeros enfoques que apuntaron a resolver el problema de la reverberación se orientaron al modelado o registro de las respuestas al impulso y la estimación de filtros inversos a partir de estas [6]. Como el efecto de la reverberación en una señal se puede pensar como el resultado de una convolución entre una señal anecoica y una respuesta al impulso, este enfoque apunta a estimar la respuesta al impulso con el fin de poder generar un filtro inverso que permita realizar una deconvolución de la señal para poder revertir el efecto de la respuesta del recinto, recuperando la señal en su estado anecoico. Sin embargo, esta metodología presenta varios inconvenientes, como el hecho de considerar que las respuestas al impulso son lineales e invariantes en el tiempo, lo cual no siempre se cumple en la práctica [7], o bien, el hecho de que la respuesta no siempre pueda ser deducida de manera directa y deba ser estimada.

También surgieron trabajos enfocados en modelar matemáticamente la señal de habla anecoica [8]. Algunos de estos trabajos consistían en estimar la señal de habla mediante predicción lineal y calcular el residuo, el cual contiene información sobre la reverberación en la señal. Esta señal de residuo se utilizó para estimar filtros variantes en el tiempo que al aplicarse a la señal de habla lograban eliminar parte de la reverberación [9]. Otro enfoque consistió en utilizar múltiples transductores y aplicar técnicas de factorización matricial, como la descomposición en valores singulares (SVD), sobre las señales captadas [10]. Algunas características propias de las señales de habla, como la estructura armónica [11] y el índice de modulación [12], fueron explotadas para eliminar los efectos de la reverberación.

Posteriormente, se aplicó la idea de la sustracción espectral [13] [14] que básicamente consiste en la estimación del espectro de potencia generado por la reverberación a partir de modelos estadísticos. En 2006, Wang et. al. aplicaron este enfoque combinado con el de la estimación de filtros inversos logrando presentar avances importantes en la efectividad de los algoritmos [15].

El uso de máscaras binarias ideales en el dominio temporal-frecuencial para extraer las se-

ñales buscadas [16] es un enfoque muy utilizado, el cual tiene su origen en el campo del análisis computacional de escenas auditivas [17]. Las máscaras se definen como ideales ya que su obtención requiere del conocimiento de la señal buscada y de la señal que interfiere. El uso de estas máscaras implica primero realizar una transformación de la señal de entrada de manera de trasladarla al dominio tiempo-frecuencia (por ejemplo, calculando un espectrograma o un cacleograma) y luego asignarle a cada punto del espacio temporal-frecuencial un valor de 1 cuando su energía mayormente pertenece a la señal objetivo, y un valor de 0 en el caso contrario. Roman et. al. [18] aplicaron este concepto para tratar el problema de dereverberación, donde se busca estimar la máscara binaria ideal tomando como señal objetivo la señal del habla en condiciones anecoicas y como interferencia a la parte reverberante. Para conseguir la dereverberación, este método requiere seleccionar de manera correcta parámetros como el punto desde el cual se distingue la parte temprana y tardía de la reverberación, y el nivel del umbral en base al cual se identifica a un punto específico como parte de la señal deseada o de la interferencia [19]. Hazrati et al. [20] propusieron estimar la máscara binaria a partir de un parámetro dependiente de la varianza de la señal, la cual define un umbral adaptativo, obteniendo mejores resultados.

A partir del año 2007, se comenzaron a aplicar redes neuronales en la tarea de dereverberación. Jin y Wang [21] utilizaron perceptrones multicapa para estimar las máscaras binarias necesarias para la separación de la componente reverberante en una señal voz. La red neuronal aprende a estimar máscaras binarias a partir de la representación tiempo-frecuencia de la señal reverberada. Mas adelante, con el avance de los modelos de aprendizaje profundo, esta técnica se iría perfeccionando reflejándose en mejores resultados en la tarea de dereverberación. Los enfoques basados en la estimación de máscaras tuvieron variantes como por ejemplo la estimación de máscaras ideales reales [22], máscaras ideales complejas [23] y máscaras sensibles a la fase [24].

En 2014 Kun et al. [25] proponen el uso de redes neuronales profundas para aprender el mapeo espectral de señales reverberantes hacia señales anecoicas. Esto quiere decir, en otras palabras, que se entrena una red neuronal profunda para que sea capaz de estimar el espectro anecoico a partir de la señal reverberada. Nuevamente se vuelve al planteo de la búsqueda del filtro inverso que permita la deconvolución de la señal reverberante para obtener su versión

anecoica, pero en este caso se utilizarán redes neuronales para estimar ese filtro.

Se han explorado una gran cantidad de arquitecturas y tipos de redes neuronales profundas. Las más utilizadas son las redes neuronales convolucionales (CNN), que surgieron del estudio de la corteza visual del cerebro y han sido muy exitosas en algunas tareas visuales complejas [26]. Las CNN en general trabajan sobre espectrogramas de magnitud, y tienen una estructura de codificador-decodificador [27], [28]. Estas son eficientes en términos de parámetros, aunque requieren de una gran cantidad de capas (o profundidad). Esto se debe a que cada capa convolucional modela su entrada de forma local, con un campo receptivo limitado, y es necesario colocar muchas capas en serie para ampliar ese campo receptivo y abarcar la totalidad del espectrograma de entrada. Otras arquitecturas han sido exploradas para deneverberación del habla, como por ejemplo redes recurrentes [29], combinaciones de redes recurrentes y convolucionales [30] y redes generativas adversarias [24].

Gran parte de los trabajos procesan la magnitud del espectrograma, ignorando la información de fase. En estos casos, al realizar la inversión del espectrograma estimado, algunos sistemas [25] utilizan el algoritmo de Griffin Lim [31], el cual permite invertir espectrogramas utilizando solamente su magnitud. Otros trabajos utilizan la fase original de la señal reverberada [27], [28], [32], [33], lo cual es una solución sencilla aunque subóptima. Por último, en trabajos recientes la información de fase es utilizada por las redes neuronales, ya sea porque trabajan con el espectrograma complejo [22], o porque modelan directamente la forma de onda [34].

CAPÍTULO 3: MARCO TEÓRICO

3.1 REPRESENTACIÓN TEMPORAL Y FRECUENCIAL DE SEÑALES

El sonido se genera cuando un disturbio que se propaga por un material elástico causa una alteración de la presión o un desplazamiento de las partículas del material que puedan ser reconocidos por una persona o por un instrumento [35]. En este trabajo se manipulan señales de audio digitales, por lo cual se realiza un muestreo periódico de la señal de audio analógica. La distancia temporal entre dos muestras contiguas se determina según la frecuencia máxima que se desea representar, acorde al teorema de Nyquist [36]. Entonces, una señal continua $x_c(t)$ que es muestreada a una frecuencia de f_s muestras por segundo, produce una señal discreta $x[n]$ como se expresa en la ecuación 1.

$$x[n] = x_c(nT_s) \quad -\infty < n < \infty \quad (1)$$

En donde T_s es el período de muestreo, y es equivalente al inverso de la frecuencia de muestreo f_s . Partiendo de esta representación temporal discreta de la señal, se puede obtener una representación frecuencial (espectro) de la misma a partir de la Transformada Discreta de Fourier (DFT) [36]. Dada una señal discreta $x[n]$ con N muestras, su transformada discreta de Fourier se define como:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jn2\pi k/N} \quad k = 0, 1, 2, \dots, N-1 \quad (2)$$

En donde X es la transformada de Fourier discreta de x , y cada muestra de $X[k]$ es el resultado del producto interno entre x y una exponencial compleja de frecuencia $2\pi k/N$. La DFT es invertible, por lo cual es posible recuperar la señal $x[n]$ a partir de $X[k]$ utilizando la transformada discreta de Fourier inversa (IDFT):

$$x[n] = 1/N \sum_{k=0}^{N-1} X[k] e^{jnk2\pi/N} \quad 0 \leq n \leq N-1 \quad (3)$$

Existen algoritmos eficientes para el cálculo de la DFT, que se denominan algoritmos de transformada rápida de Fourier (FFT) [37]. Estos consiguen realizar el cálculo de la DFT redu-

ciendo considerablemente la complejidad computacional, de $O(N^2)$ a $O(N \log(N))$.

3.1.1 Transformada de corto término de Fourier (STFT)

Las señales sonoras a analizar pueden ser no estacionarias. En este caso, la forma de onda de la señal nos brinda información del orden de aparición de los eventos sonoros, pero no sobre sus características frecuenciales. Por otro lado, la DFT nos permite tener información sobre el espectro de la señal, pero resignando información sobre la evolución temporal. Entonces, para representar adecuadamente este tipo de señales tanto en tiempo como en frecuencia se utiliza la transformada de corto término de Fourier (STFT). La misma consiste en calcular la DFT sobre una ventana temporal que limita la cantidad de muestras a utilizar. Esta ventana se desplaza a lo largo de la señal, de manera tal que el resultado final pueda representar las variaciones del espectro en el tiempo. Matemáticamente, la STFT se define como:

$$X[t, k] = \sum_{n=0}^{N-1} w[n]x[tH + n]e^{-j\frac{2\pi kn}{N}} \quad (4)$$

en donde:

- $X[t, k]$ es la transformada de corto término de Fourier de $x[n]$.
- t y k son los indices temporales y frecuenciales respectivamente.
- $w[n]$ es la ventana utilizada.
- N es el número de muestras de la ventana
- H es el número de muestras que se desplaza la ventana. Se lo denomina tamaño de salto o *hop size*, y determina el factor de solapamiento entre ventanas.

En la Figura 1 se ilustra el proceso descrito para la obtención de la STFT.

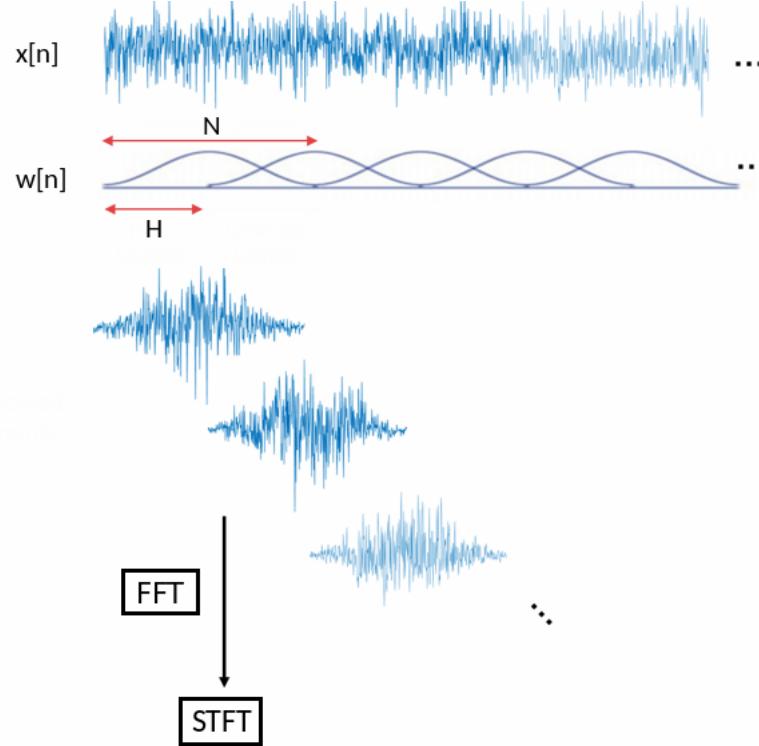


Figura 1: Proceso de obtención de la STFT. Extraído y adaptado de [38].

El resultado de la STFT depende tanto de la señal bajo análisis como del tipo de ventana utilizada. Esto se debe a que la señal y la ventana se multiplican en el dominio del tiempo, lo que equivale a una convolución en frecuencia. Es decir que el espectro frecuencial de la señal se verá distorsionado por el espectro frecuencial de la ventana utilizada. Sabiendo esto, las ventanas se diseñan para controlar la distorsiónpectral que producen sobre la señal bajo análisis. Existen dos tipos principales de distorsiónpectral: el manchado espectral y la fuga espectral. El manchado espectral refiere a una pérdida de resolución en frecuencia y está relacionado al ancho del lóbulo principal del espectro de la ventana utilizada. Por otro lado, la fuga espectral consiste en la aparición de componentes frecuenciales que no corresponden a la señal bajo análisis y está relacionada a la amplitud relativa del lóbulo principal con respecto a los lóbulos secundarios. En la Figura 2 se muestran algunas ventanas con sus respectivas respuestas en frecuencia.

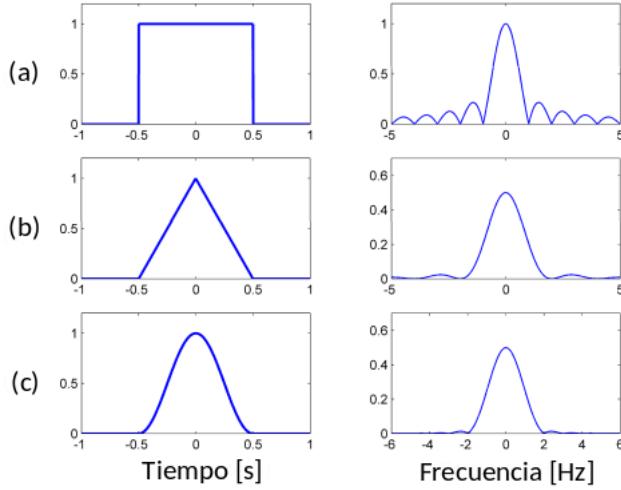


Figura 2: Ventanas (a) rectangular, (b) triangular y (c) Hann, con sus respectivas respuestas en frecuencia.
Extraido de [39].

El resultado de aplicar la STFT es una matriz de números complejos. Dicha matriz puede descomponerse en componentes de magnitud y fase. Comúnmente, en tareas de procesamiento de audio, la información de fase se descarta y se trabaja únicamente con la magnitud. Toman-do la magnitud de la STFT, las amplitudes suelen representarse en una escala de colores para producir una visualización como la que se muestra en la Figura 3 a la que se denomina espe-trograma.

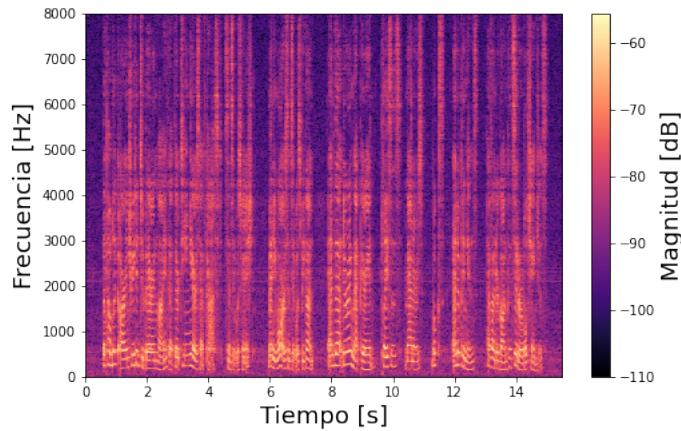


Figura 3: Espectrograma de una señal de audio.

La STFT es una transformación reversible. Para obtener una señal temporal partiendo de una STFT se aplica la técnica de solapamiento y suma (overlap-add). La misma consiste en cal-

cular la IDFT para cada cuadro temporal de la STFT y sumar las señales resultantes aplicando el mismo desplazamiento que se utilizó en el proceso de análisis. En términos matemáticos, esta operación se define como:

$$x[n] = \sum_{t=0}^{L-1} Shift_{tH} \left[\frac{1}{N} \sum_{k=0}^{N-1} X[t, k] e^{j \frac{2\pi kn}{N}} \right] \quad (5)$$

En donde L es la cantidad de cuadros temporales presentes en la STFT. Para que la reconstrucción de la señal sea correcta, la ventana utilizada tiene que cumplir con el criterio de solapamiento y suma constante (COLA):

$$\sum_{t=0}^{L-1} w[n - tH] = \alpha \quad \forall n \in \mathbb{Z} \quad (6)$$

En donde α es una constante. Cuando $\alpha = 1$ la reconstrucción es perfecta. Para otros valores de α se deben aplicar compensaciones de amplitud sobre la señal reconstruida. La ventana de Hann cumple el criterio COLA siempre que la relación $\frac{N}{H}$ sea un número entero mayor a 1. En la Figura 4 se muestran ejemplos de diferentes grados de solapamientos. Se puede ver que cuando no se cumple la relación anteriormente mencionada entre el largo de la ventana y el tamaño de salto, se produce una modulación en la amplitud de la señal recuperada.

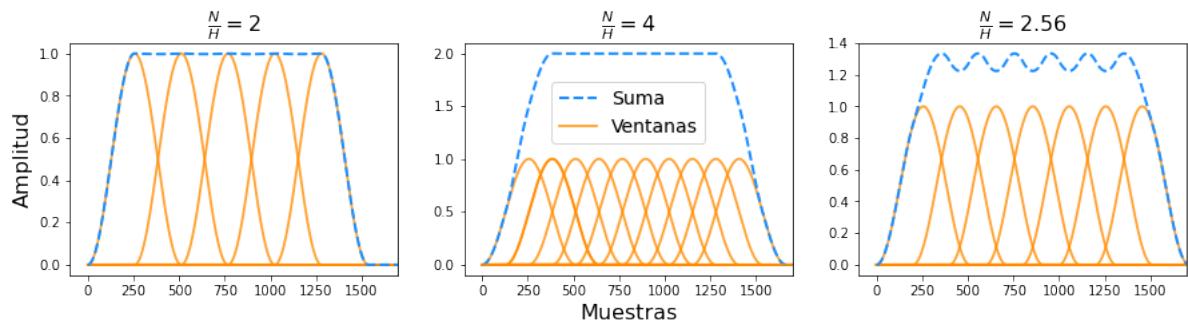


Figura 4: Efecto del solapamiento entre ventanas en la STFT.

3.2 RESPUESTA AL IMPULSO Y REVERBERACIÓN

Si en un recinto se tiene una fuente sonora y un micrófono captando a una cierta distancia, las ondas sonoras emitidas se reflejan en las paredes del recinto y alcanzan el micrófono inmediatamente después que la onda sonora directa. Cada instancia de reflexión supone una

disminución de la energía de la onda, principalmente causada por el efecto de absorción acústica de las superficies que producen las reflexiones. En un determinado tiempo, la energía sonora decaerá en todo el recinto hasta ubicarse por debajo del ruido de fondo. A este proceso se lo denomina reverberación. Al camino más corto entre la fuente y el punto de captura se lo denomina camino directo, y a la relación de nivel entre la presión sonora que genera la onda propia del camino directo y la presión que genera el efecto de reverberación se lo conoce como relación directo-reverberado.

Si el micrófono se ubica cerca de la fuente va a captar en mayor medida la señal correspondiente al camino directo, y una pequeña porción del sonido reverberado. Es decir, una relación directo-reverberado alta. A medida que el punto de captura se aleja de la fuente va a captar una menor cantidad del sonido correspondientemente al camino directo, mientras que el campo reverberado se mantendrá aproximadamente invariante. Esto se traduce en una disminución de la relación directo-reverberado. De esta manera, habrá una distancia específica para la cual el nivel de presión sonora generado por la fuente sera igual al nivel de presión sonora generado por el efecto de la reverberación. Esta distancia se conoce como distancia crítica, y depende tanto de las condiciones del recinto como de las características del micrófono utilizado.

Si pensamos a la fuente y el micrófono dentro del recinto como un sistema, es de interés estudiar su respuesta al impulso $h(n)$ para poder calcular una serie de parámetros que describan las características acústicas del recinto. Como su nombre lo indica, la respuesta al impulso equivale a la respuesta del sistema cuando se lo excita con un impulso infinitamente angosto (delta de Dirac). Dicha respuesta al impulso será diferente para cada par de puntos fuente-receptor dentro del recinto.

La Figura 5 muestra una respuesta al impulso junto con un esquema temporal de la misma. En dicho esquema podemos identificar 3 partes: primero, el nivel de sonido directo (producido por la onda que viaja a través del camino directo), las reflexiones tempranas (cuyo límite temporal vendrá definido por las características propias de cada recinto) y por último la cola reverberante. Se puede distinguir la parte de reflexiones tempranas y la cola reverberante partiendo de la suposición de que las reflexiones tempranas ocurren en un proceso determinístico, siendo altamente sensibles a pequeños cambios en la geometría del recinto, mientras que la cola reverberante es más bien un proceso estocástico, y al depender de un mayor número de

reflexiones es poco sensible a cambios de geometría.

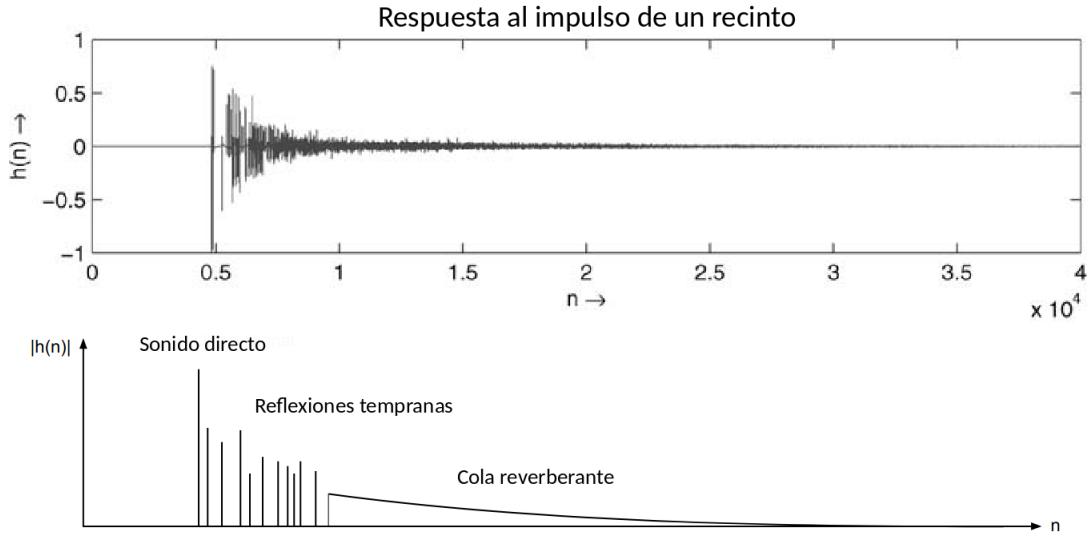


Figura 5: Secciones temporales de una respuesta al impulso. Extraído de [40].

Idealmente, el micrófono captura una señal que corresponde a la convolución entre la respuesta al impulso del recinto y la señal correspondiente a la fuente, como se ve en la ecuación 7. Esto equivale a una multiplicación en el dominio de la frecuencia de acuerdo con la transformada de Fourier, como se ve en la ecuación 8.

$$x(t) = h(t) * s(t) \quad (7)$$

$$X(f) = H(f)S(f) \quad (8)$$

De esta manera se puede ver que la respuesta al impulso conserva toda la información sobre la influencia de la reverberación del recinto sobre la señal captada por el micrófono.

La respuesta al impulso de un recinto se mide actualmente utilizando una técnica de barrido frecuencial [41]. Además, existen modelos geométricos que se utilizan para determinar la respuesta al impulso de manera analítica. Los principales son el modelo de trazado de rayos [42] y el modelo fuente imagen [43]. Estos modelos asumen la propagación del sonido como rayos en lugar de ondas. En la Figura 6 se ilustran ambos modelos.

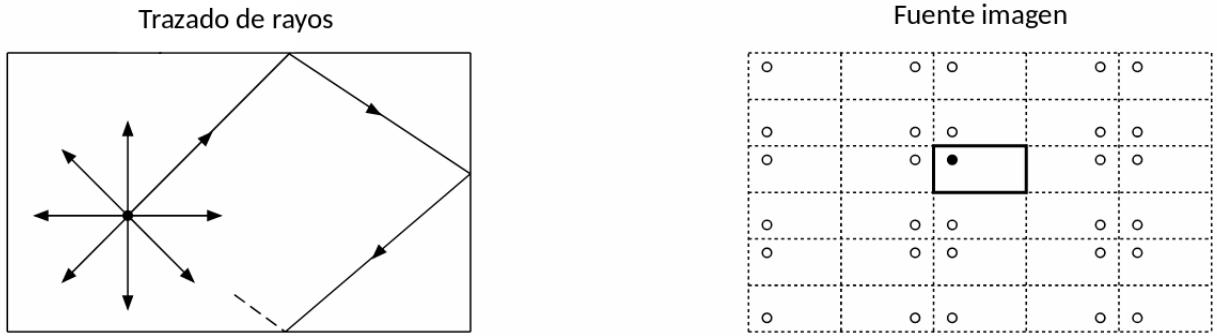


Figura 6: Modelos analíticos de cálculo de la respuesta al impulso de un recinto. Extraído de [40].

El trazado de rayos consiste en considerar un punto de fuente que emite radialmente. La longitud de los caminos de cada rayo y los coeficientes de absorción acústica de las superficies del recinto se utilizan para determinar la respuesta al impulso del recinto. Por otro lado, el modelo de fuente imagen se basa en el principio de que una reflexión especular puede ser definida geométricamente espejando la fuente respecto del plano de reflexión. De esta forma se genera una imagen especular de la fuente por cada superficie de reflexión, y esto se aplica de manera recursiva. La suma de todas las fuentes imágenes con sus respectivos retardos y atenuaciones conforman la respuesta al impulso estimada del recinto.

3.2.1 Relación directo-reverberado (DRR)

Es un descriptor acústico que se aplica sobre respuestas al impulso. Se define según la ecuación 9 en la cual $h(n)$ representa la respuesta al impulso discreta obtenida. Los índices desde cero hasta n_d representan las muestras correspondientes a la señal directa, y las muestras que continúan luego de n_d representan solo la reverberación producida por las reflexiones.

$$DRR[dB] = 10 \log_{10} \left(\frac{\sum_{n=0}^{n_d} h^2(n)}{\sum_{n=n_d+1}^{\infty} h^2(n)} \right) \quad (9)$$

Este parámetro es dependiente de la distancia entre el punto emisor y receptor, y del tiempo de reverberación del recinto. Como esta definición inicialmente se piensa en un dominio continuo, la primera intuición es pensar que el camino directo está fielmente representado por la mayor magnitud en la parte temprana de la respuesta al impulso. Sin embargo, esto solo es correcto cuando el tiempo de propagación entre la fuente y el receptor es un múltiplo entero

del período de muestreo. Por esto, trabajar con frecuencias de muestreo finitas (dominio discreto) en general deriva en que la representación del camino directo se produzca a través de una función seno cardinal ($Sinc$) correspondiente a la ventana de muestreo, centrada de acuerdo al retardo correspondiente al tiempo de propagación. En cambio, cuando se trata de respuestas al impulso sintéticas, el camino directo puede ser computado de forma separada del resto. Es decir, se puede determinar con exactitud el aporte del campo directo y del campo reverberado, lo que permite el cálculo del parámetro DRR con una mayor exactitud.

3.3 INTELIGIBILIDAD Y PARÁMETROS DE CALIDAD DE PERCEPCIÓN

Para caracterizar la señal del habla propagándose en condiciones reverberantes se utilizan métricas objetivas derivadas de la respuesta al impulso del recinto en cuestión, como por ejemplo el tiempo de reverberación o la relación energética entre la señal directa y el campo reverberado. En cambio, al considerar el proceso de dereverberación de estas señales las respuestas al impulso requieren ser estimadas, lo que usualmente conduce a una caracterización de baja calidad. Además, los algoritmos de dereverberación pueden introducir artefactos audibles a la señal voz, los cuales no son contemplados por las respuestas al impulso estimadas. Es por esto que es preciso utilizar métodos de medida de calidad basados en la señal dereverberada. Las pruebas subjetivas son el método más confiable para evaluar la calidad percibida de una señal de habla dereverberada. Sin embargo, este método es costoso y requiere mucho tiempo, por lo cual se vuelve inviable su aplicación para procesamientos en tiempo real. Para aplicaciones prácticas se definieron entonces métodos objetivos de medición de calidad basados en la señal dereverberada como reemplazo de las pruebas subjetivas. Estos métodos consisten en algoritmos que de manera objetiva y repetible buscan estimar la calidad percibida de la señal, por lo cual, un método resulta efectivo cuando logra obtener una alta correlación con las respuestas subjetivas. Estos métodos se clasifican en intrusivos o no intrusivos, dependiendo de si requieren o no una señal de referencia para realizar la estimación. Poder contar con una señal de referencia para realizar estas estimaciones es usualmente una dificultad, por lo cual se presta mayor interés en aquellos métodos no intrusivos.

3.3.1 Relación energía de modulación de voz a reverberación

Este parámetro de medida de calidad para señales dereverberadas se basa en obtener características de la reverberación partiendo del espectro de modulación de la señal [44]. La formulación de este parámetro se basa en el hecho de que la cola reverberante de una respuesta al impulso puede ser modelada como ruido blanco gaussiano exponencialmente amortiguado. Esta característica puede ser explotada en el análisis del espectro de modulación de la señal bajo análisis para obtener descriptores del efecto de la reverberación.

3.3.2 Inteligibilidad objetiva de corto término extendida

Este parámetro está basado en características extraídas a partir de la correlación de corto término entre la señal limpia y la señal procesada. Es aplicable para evaluar aquellos procesos que realizan transformaciones no lineales [45]. Su funcionamiento se basa en aplicar una ventana de análisis de 384 ms en las envolventes de amplitud de las subbandas de la señal analizada. Estas ventanas temporales se aplican en pos de contemplar frecuencias de modulación que son relevantes para la inteligibilidad. En estos lapsos temporales se calculan coeficientes de correlación espectrales que son luego promediados. De esta manera, este parámetro puede ser interpretado en términos de una descomposición ortogonal de espectrogramas energéticamente normalizados que son luego ordenados de acuerdo a su contribución a la inteligibilidad estimada.

3.3.3 Relación señal a distorsión

Este descriptor fue ampliamente utilizado en tareas de separación de fuentes y refuerzo de señales de habla. Esta basado en el cómputo de la relación señal a interferencia (SIR), y en la relación señal a artefacto (SAR) [46]. En las tareas de dereverberación, estas medidas pueden ser interpretadas como proporcionales a la supresión de componentes reverberantes tardías e inversamente proporcionales a la distorsión en la señal del habla, respectivamente. Contemplando estos valores, el parámetro final mide la calidad general de la señal dereverberada.

3.4 REDES NEURONALES Y ALGORITMOS DE APRENDIZAJE PROFUNDO

Las redes neuronales artificiales son modelos computacionales que comparten algunas propiedades con el funcionamiento de las neuronas biológicas [47]. Estas conforman el estado del arte actual para la resolución y modelado de problemas de alta complejidad en diversas disciplinas [48].

3.4.1 La neurona artificial

El bloque básico de los modelos de aprendizaje profundo es la neurona artificial. El esquema de una neurona artificial se puede ver en la Figura 7. Sus componentes principales son:

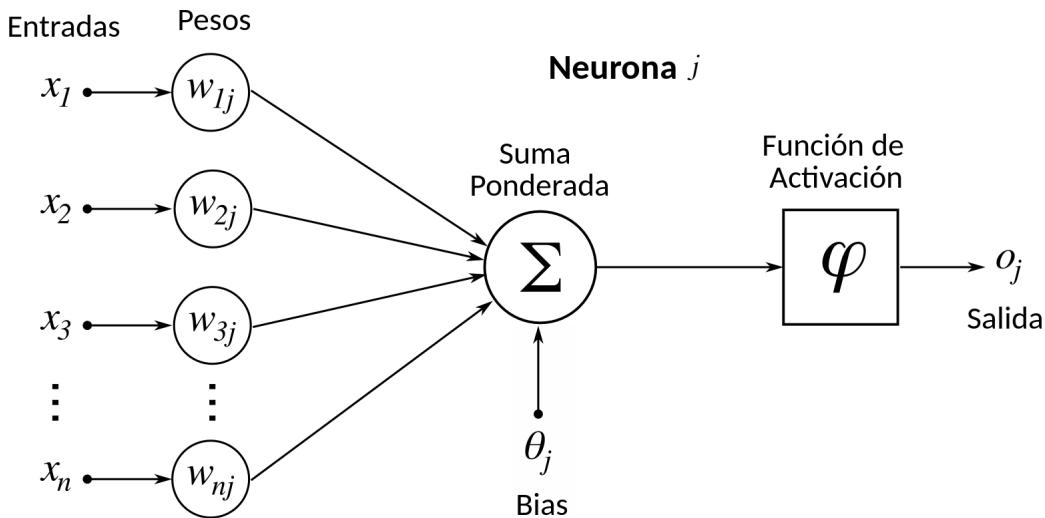


Figura 7: Esquema de neurona artificial.

- **Entradas:** son los datos que van a ser procesados en esta unidad.
- **Pesos sinápticos:** son parámetros asociados a cada entrada que se van ajustando durante la etapa de entrenamiento.
- **Umbral o Bias:** entrada externa a la red neuronal. Su valor se modifica durante el entrenamiento al igual que los pesos sinápticos.
- **Suma ponderada:** consiste en el producto interno entre el vector de entradas y el vector de pesos sinápticos, al cual se suma el valor del umbral o bias. Matemáticamente, se expresa como:

$$\sum_{i=1}^n x_i w_{ji} + \theta_j \quad (10)$$

donde x son las entradas, w los pesos sinápticos, θ el umbral o bias y n el número de entradas.

- **Función de activación:** Función que se aplica a la salida de la suma ponderada, y genera la salida de la neurona como se muestra en la ecuación 11. La misma introduce alinealidades en la neurona, haciendo que el modelo resultante sea no lineal. Algunos ejemplos de funciones de activación se pueden ver en la Figura 8.

$$o_j = \varphi\left(\sum_{i=1}^n x_i w_{ji} + \theta_j\right) \quad (11)$$

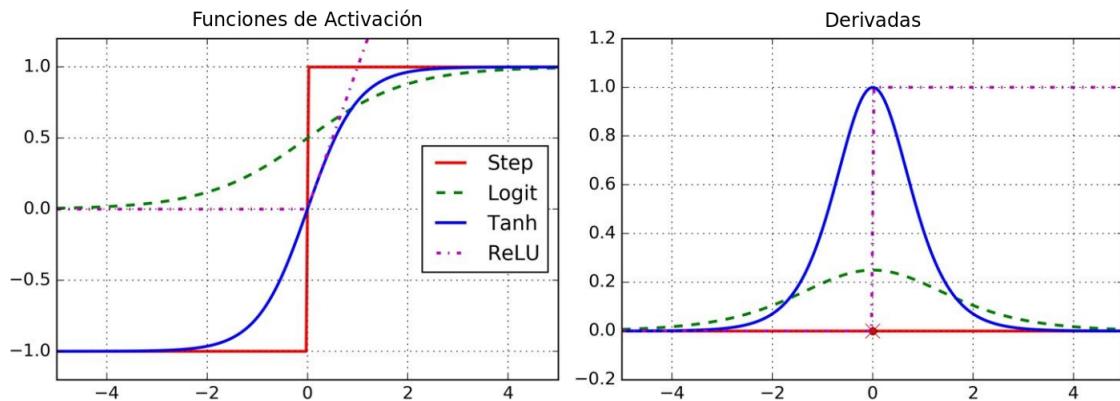


Figura 8: Funciones de activación y sus derivadas. Extraído de [26].

3.4.2 Modelos basados en redes neuronales

Una red neuronal se organiza en capas, las cuales poseen una cierta cantidad de neuronas. Las salidas de las neuronas de una capa suelen constituir las entradas de las neuronas de la capa siguiente. A la hora de diseñar una arquitectura de red neuronal, es necesario definir hiperparámetros, como cuántas capas se utilizarán, cuántas neuronas tendrá cada capa y qué tipo de capas serán, entre muchos otros.

Una de las redes neuronales más estudiadas es el perceptrón multicapa (*multilayer perceptron*, o MLP) [49]. Esta consta de una capa de entrada, una o varias capas ocultas, y una capa

de salida. Cada capa puede contener un número distinto de neuronas, y cada capa se encuentra completamente conectada a la capa adyacente. Esta red neuronal es capaz de representar cualquier función dado un número suficiente de neuronas. Un ejemplo de esta red se ilustra en la Figura 9.

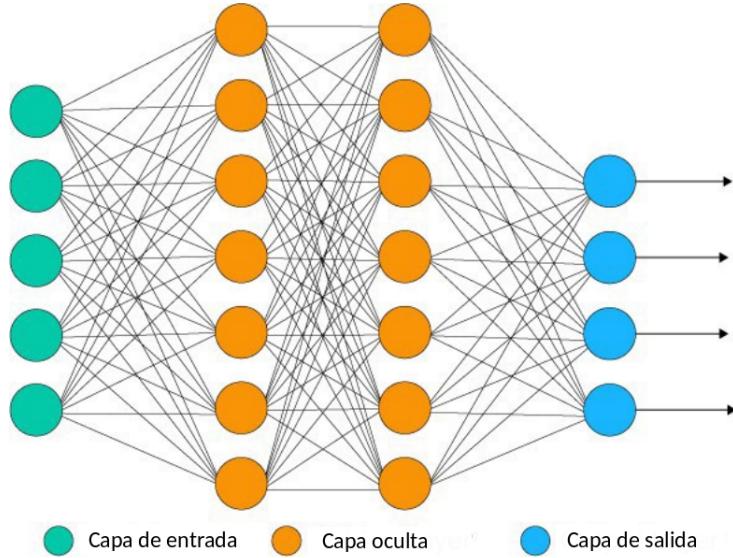


Figura 9: Ejemplo de red neuronal multicapa con alimentación hacia adelante.

Frecuentemente se cataloga a las redes neuronales con más de una capa oculta como algoritmos de aprendizaje profundo. Esto se debe a que contienen múltiples capas de procesamiento no lineal que aprenden diferentes niveles de representación formando una jerarquía de características, desde un nivel de abstracción más bajo a uno más alto [50].

3.4.3 Entrenamiento y aprendizaje

Los algoritmos de aprendizaje por máquina se entrena a partir del procesamiento de datos. Cuando se trata de un modelo de aprendizaje supervisado, los datos de entrenamiento se presentan de a pares (x, y) en donde y es el valor objetivo o la salida que se espera obtener para el valor de entrada x . En el contexto del entrenamiento de una red neuronal, se define al aprendizaje como la búsqueda de una determinada configuración de los parámetros entrenables de la red que produzcan que la entrada x genere la salida y . En general, inicialmente las entradas van a generar salidas \hat{y} aleatorias que difieren del valor objetivo y . Entonces, es necesario tener una medida de esta diferencia. De eso se encarga la función de costo (también denominada

función objetivo o función de pérdida).

La función de costo recibe las salidas de la red y las salidas esperadas, y luego calcula una medida de error a partir de una función matemática. Esta función se escoge de acuerdo a la tarea que se busca realizar. Entonces, para cada estimación de la red, la función de costo otorga un puntaje que explica cuan lejos está el valor estimado del valor objetivo.

El paso siguiente en el proceso de entrenamiento es utilizar la salida de la función de costo como una señal de realimentación para poder ajustar los parámetros entrenables de la red neuronal (pesos sinápticos, umbrales, etc) de manera tal de minimizar la función de costo. Esta tarea es realizada por la función de optimización. La misma aplica el algoritmo de propagación del error hacia atrás para computar el gradiente de la función de costo respecto a los parámetros entrenables de la red neuronal. En base a este gradiente y al valor de tasa de aprendizaje definido en la función de optimización, se puede determinar cómo modificar los parámetros entrenables para lograr disminuir el error de salida. Este bucle de entrenamiento se ilustra en el esquema de la Figura 10. Repetir este ciclo un número suficiente de veces conduce a la convergencia del valor de error.

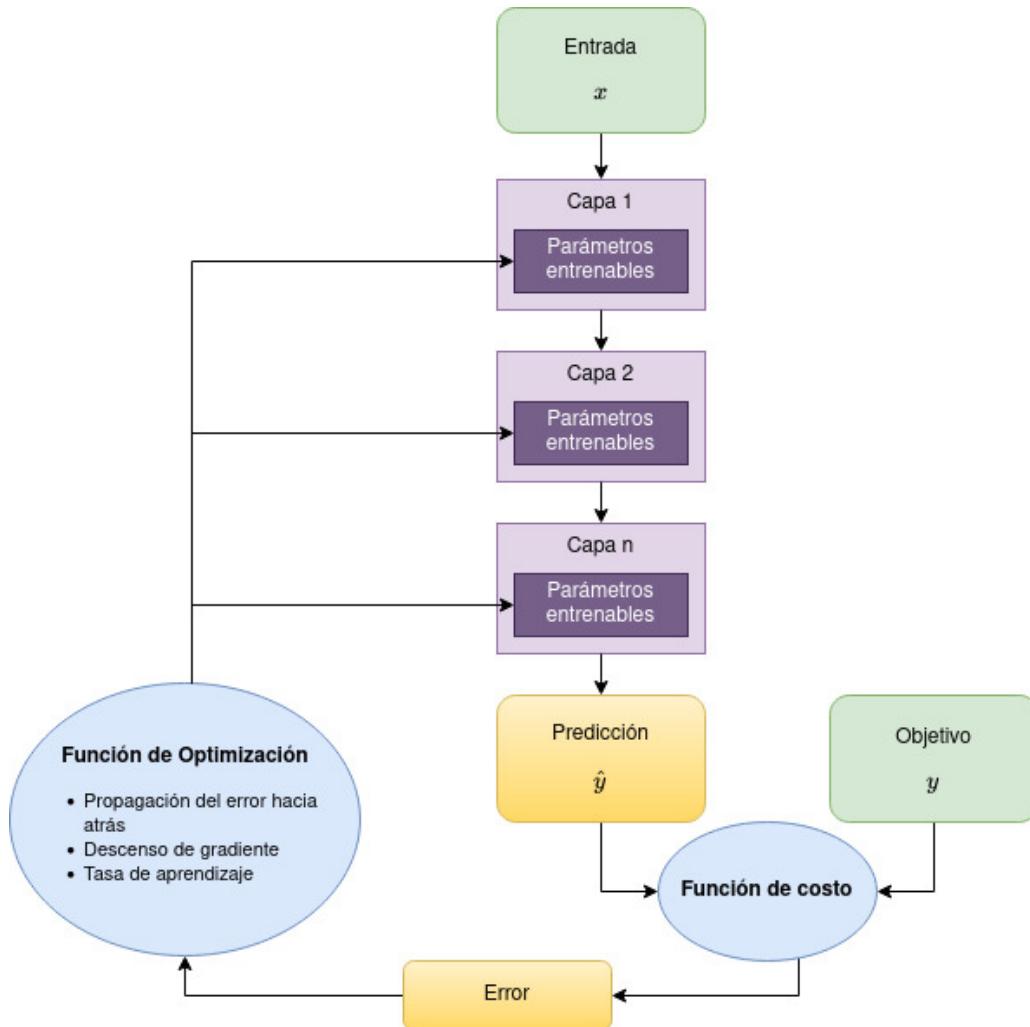


Figura 10: Diagrama de flujo del bucle de entrenamiento de una red neuronal.

En este trabajo, el conjunto de datos de entrenamiento se segmenta en lotes. En cada iteración de entrenamiento la red neuronal recibe un lote, lo procesa, aplica la función de costo y ajusta los parámetros de cada capa. Cuando la red procesó todos los lotes que componen el conjunto de datos de entrenamiento se dice que transcurrió una época. Por ende, tanto el tamaño de los lotes como la cantidad de épocas de entrenamiento son hiperparámetros a definir.

El orden en el que los datos son presentados ante el modelo puede influenciar positiva o negativamente en el resultado final del entrenamiento. Existen técnicas de optimización que se desarrollaron partiendo de cualidades propias del aprendizaje en seres humanos y animales, como el hecho de que el aprendizaje resulta mejor cuando las instancias de aprendizaje están organizadas en un orden significativo, añadiendo gradualmente mayor cantidad de conceptos

y por ende una mayor complejidad. Esta idea se traslada al entrenamiento de algoritmos de aprendizaje profundo con el desarrollo de una estrategia denominada aprendizaje por currículum [51]. La misma consiste en seleccionar cuales datos y en que orden presentarlos al sistema durante el aprendizaje, de manera de guiar el entrenamiento para que inicialmente se aprendan los conceptos más sencillos del problema, e ir gradualmente aumentando el grado de complejidad de la tarea. El aprendizaje por currículum se define como una estrategia de optimización global. Dependiendo de la tarea sobre la que se aplica, puede lograr en menor o mayor medida que un sistema logre un mejor nivel de generalización así como también llegar al punto de convergencia en un menor tiempo de entrenamiento.

Por último, la segmentación de los datos que la red neuronal recibe y procesa en cada etapa del desarrollo también influye en el desempeño de la misma. El objetivo final del sistema es alcanzar un grado de generalización que le permita procesar adecuadamente instancias de datos que no hayan sido reveladas ante la red en la etapa de entrenamiento. Por esto, el conjunto total de los datos se divide en tres subgrupos:

- **Conjunto de entrenamiento:** Este conjunto de datos es el que se utiliza en la etapa de entrenamiento para optimizar los parámetros de la red. Aquí se concentra el mayor volumen de datos.
- **Conjunto de validación:** Sobre este conjunto se mide el desempeño del sistema a lo largo de su entrenamiento. Los resultados obtenidos al evaluar en este conjunto sirven para ajustar las variables que requieren ser especificadas de manera previa al entrenamiento. Estas variables se denominan hiperparámetros.
- **Conjunto de prueba:** Este conjunto es el que se utiliza para medir el desempeño final del sistema. Como contiene instancias que no fueron utilizadas en las etapas de entrenamiento y ajuste de parámetros, la evaluación sobre este conjunto sirve para medir el nivel de generalización que el sistema logró alcanzar.

Durante el entrenamiento pueden ocurrir dos fenómenos indeseados: el sobreajuste y el subajuste. El sobreajuste refiere al caso en que el modelo obtenga muy buenos resultados sobre los datos de entrenamiento, pero malos resultados sobre los datos de validación y evaluación. Es

decir, el modelo no alcanzó el grado de generalización pretendido, y memorizó las instancias de entrenamiento. Esto ocurre cuando la complejidad del modelo es muy alta en comparación a la complejidad de la tarea que busca resolver, o no se utilizaron suficientes datos de entrenamiento. El subajuste, en cambio, hace referencia a un error elevado en la etapa de entrenamiento. Esto ocurre cuando el modelo es muy simple, o carece de la complejidad necesaria para realizar la tarea.

Existen varias estrategias para lidiar con los problemas de sobreajuste, como por ejemplo:

- **Aumento de datos de entrenamiento:** Proporcionar más datos a la etapa de entrenamiento produce una mejora en la capacidad de generalización del modelo. De todos modos, recolectar más datos no siempre es posible, y por eso existen técnicas de aumento, que consisten en manipular datos ya existentes para generar datos nuevos.
- **Dropout:** Es una técnica de regularización que consiste en anular aleatoriamente un determinado número de activaciones de una capa durante el entrenamiento [52]. Es equivalente a entrenar un número menor de neuronas, lo que se traduce en una reducción de la complejidad del modelo.
- **Interrupción temprana del entrenamiento:** Esta estrategia consiste en detener el entrenamiento antes de que el modelo comience a sobreajustar los datos [53]. Para determinar el punto en el que el modelo empieza a sobreajustar, comúnmente se monitorea el error que produce el modelo sobre el conjunto de validación a medida que transcurre el entrenamiento.

Otra técnica importante que se utiliza en este trabajo es la normalización por lotes [54]. Esta consiste en normalizar las entradas de una capa, restando la media y dividiendo por el desvío estándar del lote de datos actual. Esto suele resultar en un entrenamiento más estable, y una convergencia más rápida de las curvas de error del modelo.

3.4.4 Redes neuronales convolucionales

Las redes neuronales convolucionales emergen del estudio de la corteza visual del cerebro animal [55]. En los últimos años, estas estructuras fueron utilizadas para resolver tareas visuales

complejas (análisis de imágenes). El bloque básico de las redes neuronales convolucionales es la capa convolucional. La misma posee las siguientes características:

- **Campo receptivo limitado:** las neuronas que conforman la capa convolucional no están conectadas a todas las entradas, sino que solo se conectan con una porción de las mismas que se denomina campo receptivo. Esto hace que se modelen estructuras locales, es decir, presentes dentro de este campo receptivo.
- **Parámetros compartidos:** los pesos sinápticos de las neuronas se comparten. Por esto, los patrones aprendidos por las neuronas son invariantes a la translación. Esto quiere decir que si se aprende de un patrón ubicado en un lugar específico del volumen de entrada, este mismo patrón puede luego ser identificado en cualquier otra ubicación de la entrada.
- **Convolución:** se aplican filtros de convolución sobre las entradas, que comúnmente son imágenes. Si bien la literatura utiliza el término convolución, matemáticamente se realiza una correlación cruzada. La Figura 11 muestra en qué consiste este proceso, y como el filtro se desplaza sobre las entradas para generar las salidas. En cada paso, el filtro se multiplica por una sección de la imagen de entrada (producto interno) y el resultado corresponde a un solo valor en la imagen de salida. Las salidas de una capa convolucional se denominan mapas de características.

El funcionamiento a partir de campos receptivos y el hecho de que los parámetros se comparten entre neuronas producen que las capas convolucionales sean más eficientes en términos de cantidad de parámetros a entrenar.

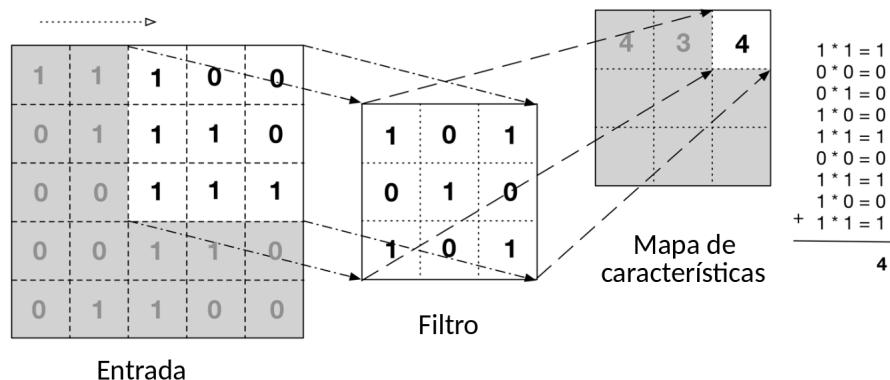


Figura 11: Funcionamiento de un filtro de convolución. Extraído de [56].

De esta manera, una capa convolucional aplica uno o varios filtros bidimensionales sobre la imagen de entrada, generando mapas de características que representan la presencia del patrón del filtro a lo largo de la imagen, como se puede apreciar en la Figura 12. En este tipo de capas, el aprendizaje se traduce en determinar los valores de los filtros que se deben aplicar para conseguir los resultados esperados.

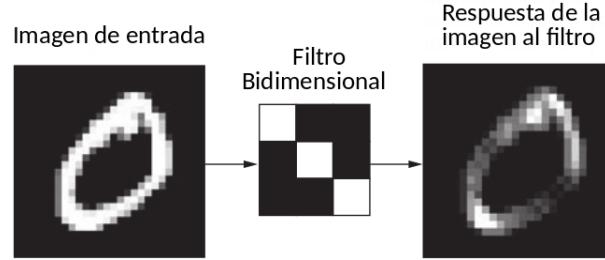


Figura 12: Representación de la aplicación de un filtro bidimensional sobre una imagen.

Los hiperparámetros que se deben definir en cada capa convolucional son:

- **Tamaño del filtro:** define el tamaño del campo receptivo de cada unidad de procesamiento de la capa. Valores comunes son 3×3 o 5×5 . En la Figura 13 se ve un ejemplo de un filtro de tamaño 3×3 .
- **Tamaño del salto:** determina la distancia horizontal y vertical entre campos receptivos de dos unidades contiguas. Hacer que este valor sea mayor a uno permite reducir las dimensiones de la imagen de entrada al atravesar la capa convolucional. Esto se puede apreciar en la Figura 13 en donde se aplica un tamaño de salto igual a dos (tanto en sentido vertical como horizontal).
- **Relleno de ceros:** cuando se pretende mantener invariables las dimensiones de entrada y salida de una capa convolucional, se suele aplicar un relleno con ceros en los contornos de la imagen. La cantidad de ceros agregados dependerá de las características del filtro a aplicar. Aplicar un relleno de ceros produce un fenómeno denominado efecto de borde [26].
- **Cantidad de filtros aplicados:** el número de filtros convolucionales que se aplican sobre la entrada. Es equivalente al número de mapas de características que se generan en cada

capa.

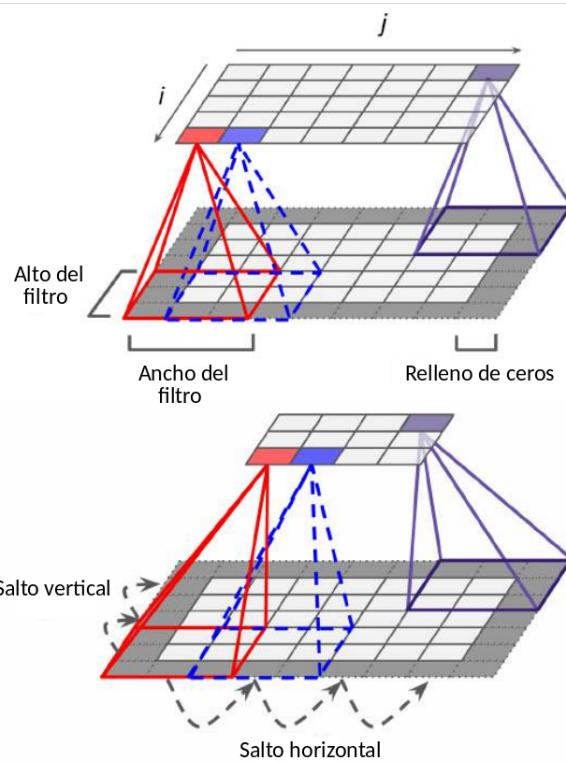


Figura 13: Principales hiperparámetros de una capa convolucional. Extraído de [26].

Para el análisis de imágenes, estas capas se organizan de manera que la primera capa no contempla todos los píxeles de la imagen, sino que solo se enfoca un número acotado de píxeles que caen dentro de su campo receptivo. De igual manera, las capas subsiguientes se enfocan en las salidas de un conjunto acotado de neuronas de la capa precedente. Este funcionamiento se ilustra en la Figura 14.

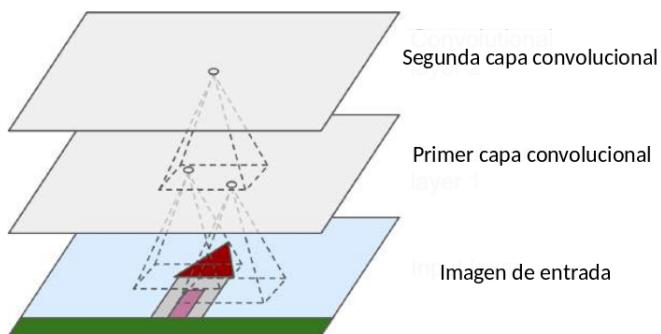


Figura 14: Capas convolucionales con campos receptivos locales rectangulares.

Formar esta estructura le permite a la red aprender diferentes patrones estructurales locales de manera jerárquica [26]. En conclusión, a diferencia de las redes completamente conectadas, las redes convolucionales logran la generalización de conceptos visuales complejos a partir de una menor cantidad de parámetros, distribuidos estratégicamente a lo largo de la arquitectura.

3.4.5 Autoencoders

Un autoencoder es una arquitectura de red neuronal cuyo objetivo es copiar su entrada en su salida. El esquema básico de un autoencoder se puede observar en la Figura 15, donde x representa a las variables de entrada, r representan las variables de salida y h representan las variables latentes.

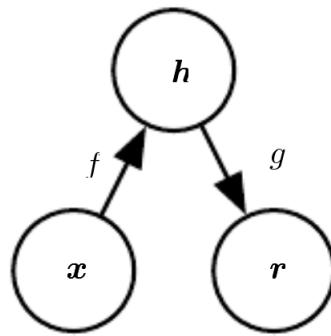


Figura 15: Estructura general de un autoencoder.

El esquema se compone de tres partes fundamentales:

- Una función de codificación f en donde las dimensiones de la variable de entrada se comprimen, idealmente descartando información irrelevante para la reconstrucción de la entrada. De esta forma, se logra reducir la dimensionalidad de la variable de entrada, conservando solo la información más relevante.
- Un espacio latente h (o espacio de representación), el cual es la representación comprimida de la entrada que genera la codificación f .
- Una función de decodificación en donde se aplica el proceso inverso que en la codificación, expandiendo las dimensiones tomadas del espacio latente para reconstruir la señal de entrada.

Comúnmente, este tipo de arquitecturas se entrenan restringiendo o condicionando los datos de entrada, para que la red se vea obligada a priorizar qué aspectos de la entrada deben copiarse, lo que a menudo implica aprender propiedades útiles de los datos.

3.5 DEREVERBERACIÓN POR FILTRADO TEMPORAL-FRECUENCIAL

3.5.1 Máscaras de amplitud

Existen numerosas maneras de representar las señales de audio para dereverberarlas. En la actualidad, los espectrogramas son la más utilizada ya que su cálculo es rápido y sencillo, son interpretables, es posible invertirlos y pueden ser fácilmente aprovechados por modelos de aprendizaje profundo como las redes neuronales convolucionales. Partiendo de una señal con reverberación, se extrae una representación en tiempo-frecuencia a partir de transformaciones como la transformada de corto término de Fourier (STFT). Una vez obtenido este espectrograma, lo que se busca es aprender el proceso necesario para obtener un nuevo espectrograma que se corresponda con la señal anecoica (descartando el efecto de la reverberación). Entonces, el proceso de dereverberación se puede resumir a la estimación de un filtro variable en el tiempo que se aplica sobre el espectrograma con reverberación. Estudios previos afirman que la fase no aporta información significativa para estas tareas de mejora del habla [57][58], por lo cual se suelen realizar estos procesos únicamente sobre la magnitud de los espectrogramas, utilizando la información de fase del audio original. Considerando esto, el proceso de dereverberación se reduce a la expresión de la ecuación 12, en donde $STFT_Y$ es la magnitud del espectrograma de la señal anecoica, $STFT_X$ es la magnitud del espectrograma de la señal con reverberación y M es la máscara ideal que representa el filtrado en el dominio tiempo-frecuencia. En otras palabras, la magnitud del espectro de la señal dereverberada se obtiene aplicando la máscara ideal sobre la magnitud del espectrograma de la señal con reverberación.

$$STFT_Y(t, f) = M(t, f)STFT_X(t, f) \quad (12)$$

Considerando que el espectro reverberado tendrá siempre mayor amplitud que el espectro anecóico, se trasladan los rangos de ambos espectros al intervalo $(0, 1]$. De esta manera, las máscaras resultantes estarán dentro del mismo intervalo. Este proceso de escalado de las en-

tradas, resulta beneficioso para los modelos de redes neuronales artificiales [26]. Luego, la red neuronal tiene el objetivo de estimar la máscara ideal a partir del espectrograma reverberado. En la Figura 16 se muestra un diagrama de cómo se estructura el sistema y las señales disponibles para lograr estimar la máscara de manera indirecta. El espectrograma con reverberación ingresa a la red neuronal artificial, la cual procesa esta entrada y produce una salida intermedia de las mismas dimensiones que la entrada. Luego, esta salida intermedia se multiplica con el espectrograma de entrada y el resultado es comparado con el espectrograma sin reverberación, mediante la función de costo. De esta manera, los pesos de la red neuronal se van a modificar en pos de que la salida del sistema sea lo más semejante posible al espectro sin reverberación. Cuando la función de costo tienda a cero, esto significará que la salida de la red neuronal tiende a igualarse con una máscara ideal, que al aplicarse sobre el espectrograma reverberado produce el espectrograma anecóico.

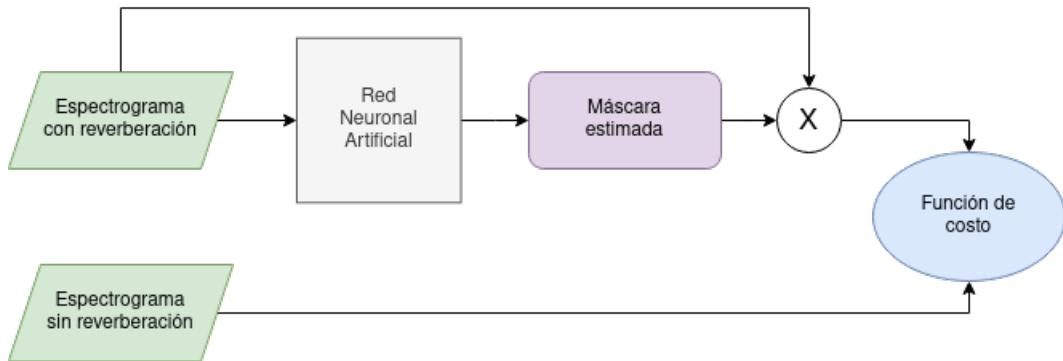


Figura 16: Disposición de las entradas y salidas del modelo para la estimación de las máscaras de amplitud.

3.5.2 Síntesis de audio a partir de espectrogramas

En este trabajo, como en muchas otras tareas de procesamiento de audio (y procesamiento de señales en general), se lleva a cabo la siguiente secuencia de pasos:

1. Tomar una señal en el dominio del tiempo $x[n]$ y convertirla en un espectrograma $X[t, f]$ a través de la STFT.
2. Modificar el espectrograma $X[t, f]$ para obtener $\hat{X}[t, f]$.
3. Convertir el espectrograma modificado $\hat{X}[t, f]$ nuevamente a una señal en el dominio del tiempo $\hat{x}[n]$ a través de la ISTFT.

El último paso de la lista conlleva un problema, ya que ciertos procesos pueden generar espectrogramas que no sean consistentes, es decir, que no haya ninguna señal en el dominio del tiempo cuyo espectrograma sea el generado. Para solucionar este problema se desarrollaron algoritmos que buscan estimar una señal temporal cuyo espectrograma sea el más cercano posible al espectrograma que se quiere invertir. Este es el caso del algoritmo propuesto por Griffin y Lim [31]. El algoritmo consiste en un bucle iterativo que busca minimizar el error cuadrático medio entre el espectrograma de la señal estimada y el espectrograma modificado. En la Figura 17 se muestra un diagrama de bloques que explica el funcionamiento básico del algoritmo.

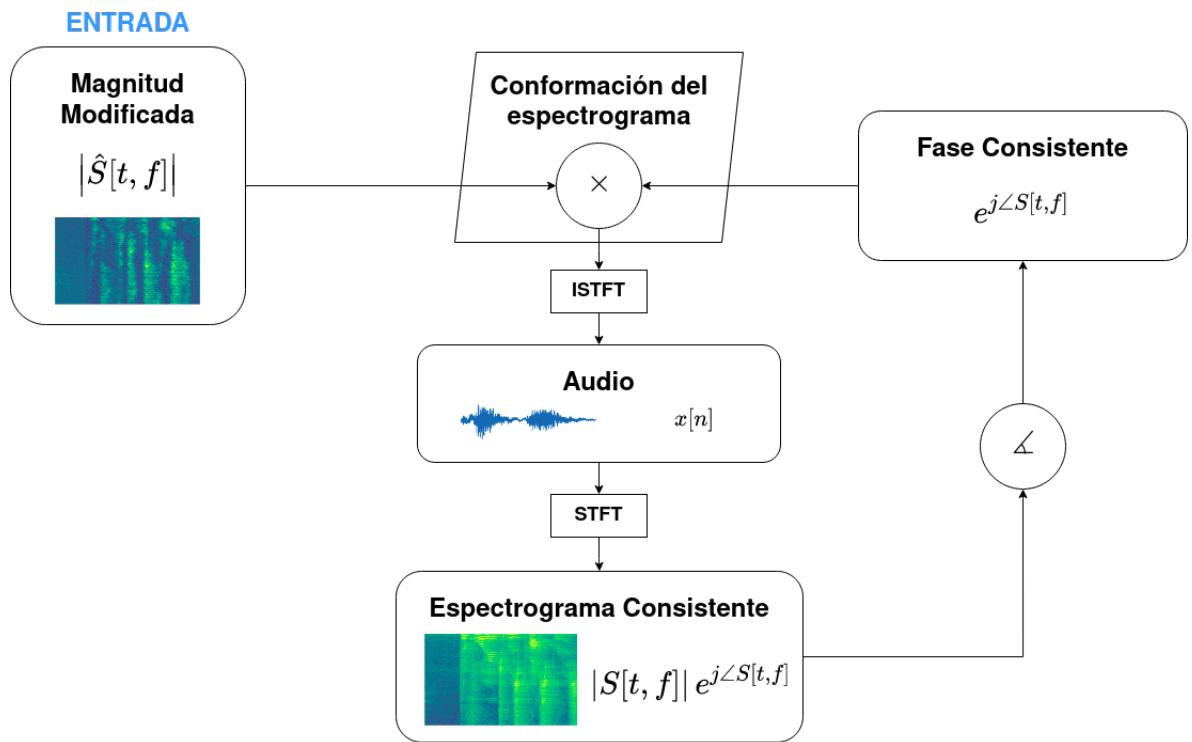


Figura 17: Diagrama en bloques del algoritmo de Griffin-Lim.

La magnitud del espectrograma de entrada $|\hat{S}[t, f]|$ inicialmente se combina con una fase aleatoria para formar un espectrograma complejo. Si se cuenta con una estimación previa de la fase, esta puede utilizarse en lugar de la fase aleatoria al iniciar el proceso. Luego, el espectrograma complejo conformado se antitransforma obteniendo una señal de audio, la cual vuelve a ser transformada obteniéndose un nuevo espectrograma complejo $|S[t, f]| e^{j\angle S[t,f]}$. Este espectrograma es consistente, pues deriva de la transformación de una señal de audio real. Se puede probar que combinar esta fase resultante $\angle S[t, f]$ con el espectrograma modificado de entra-

da $|\hat{S}[t, f]|$ disminuye el error cuadrático entre los espectrogramas evaluados (es decir, entre el espectrograma consistente y el inconsistente). Entonces, se combina la fase consistente con la magnitud del espectrograma de entrada y se vuelve a repetir todo el proceso. Se debe definir un criterio para determinar cuando dejar de iterar, y tomar al espectrograma consistente como el resultado final del proceso para antitransformarlo y obtener $\hat{x}[n]$. Un criterio podría ser calcular la convergencia espectral al final de cada iteración, y definir un valor de tolerancia. De esta manera, el bucle se repite hasta que la diferencia entre los espectrogramas evaluados sea menor que un valor definido como tolerancia. Otro criterio, por ejemplo, podría ser directamente fijar un número de iteraciones a realizar a partir del cual se asume la convergencia del espectro resultante. De igual manera se podrían definir otros criterios.

CAPÍTULO 4: METODOLOGÍA

4.1 GENERACIÓN DE DATOS

Para poder entrenar un algoritmo de aprendizaje profundo es recomendable contar con un conjunto de datos extenso. Este conjunto debe poder representar de la mejor manera posible el fenómeno que se quiere modelar. Además, se debe poder asegurar que todas las instancias que componen la base de datos comparten un mismo tipo de codificación (formato de audio, profundidad de bits, frecuencia de muestreo) que se adecúe a los procesos subsiguientes.

Para este trabajo se requieren grabaciones de voz anecoicas, y las mismas señales reverberadas. La forma más sencilla de obtener las señales reverberadas es utilizando respuestas al impulso. Mediante el proceso de convolución, es posible obtener las señales reverberadas a partir de las anecoicas y las respuestas al impulso. En este trabajo, se asumirá que los recintos son sistemas LTI y por lo tanto pueden ser modelados mediante un proceso de convolución con una respuesta al impulso.

Además, se debe tener en cuenta que se busca formar tres grandes conjuntos de datos: conjunto de entrenamiento, conjunto de validación y conjunto de prueba. Los conjuntos de validación y prueba deben servir para verificar que el sistema es capaz de generalizar a condiciones no vistas durante el entrenamiento, pero que si serán encontradas al utilizar el sistema de dereverberación. Por ejemplo, debe ser capaz de generalizar a recintos y voces que no vio durante el entrenamiento. Por esto, se utilizaron respuestas al impulso y señales de voz distintas en cada conjunto generado.

4.2 BASES DE DATOS DE RESPUESTAS AL IMPULSO

Para este trabajo se utilizaron respuestas al impulso reales y simuladas. A partir de este punto, se utilizarán los términos respuestas simuladas y respuestas generadas de manera indistinta. A su vez, también se trabaja con un tercer conjunto formado a partir de la aumentación de respuestas al impulso reales. Esto es, partiendo de un subconjunto de respuestas al impulso reales, se alteran estas señales de manera controlada para producir nuevas respuestas al impulso con diferentes características acústicas.

4.2.1 Respuestas al impulso reales

Las respuestas al impulso reales se obtienen del conjunto de datos C4DM [59]. Este conjunto consiste en una colección de respuestas al impulso que fueron registradas en tres recintos: una sala multipropósito con aproximadamente 800 asientos (*greathall*), un edificio victoriano construido en 1988 originalmente diseñado para ser una biblioteca (*octagon*), y una sala de clases de una universidad (*classroom*). Las Figuras 18, 19, 20 muestran imágenes de los recintos nombrados.

Tanto en la sala multipropósito como en la biblioteca se registraron un total de 169 respuestas al impulso, mientras que en la sala de clases se registraron 130 respuestas al impulso, como se puede observar en los esquemas de medición.

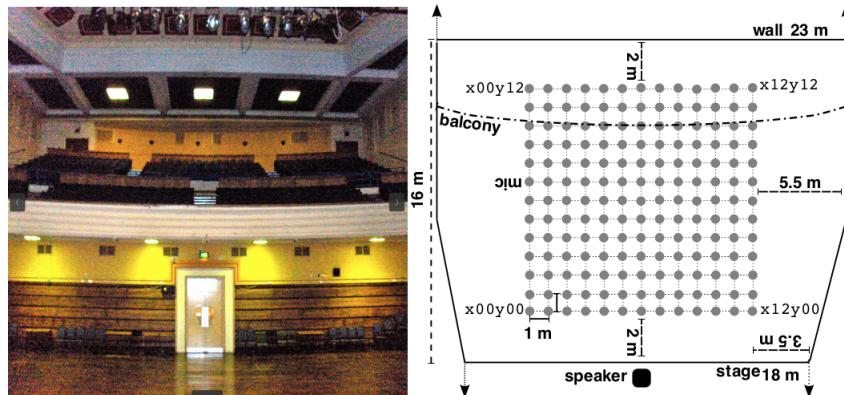


Figura 18: Recinto greathall y esquema de medición de respuestas al impulso.

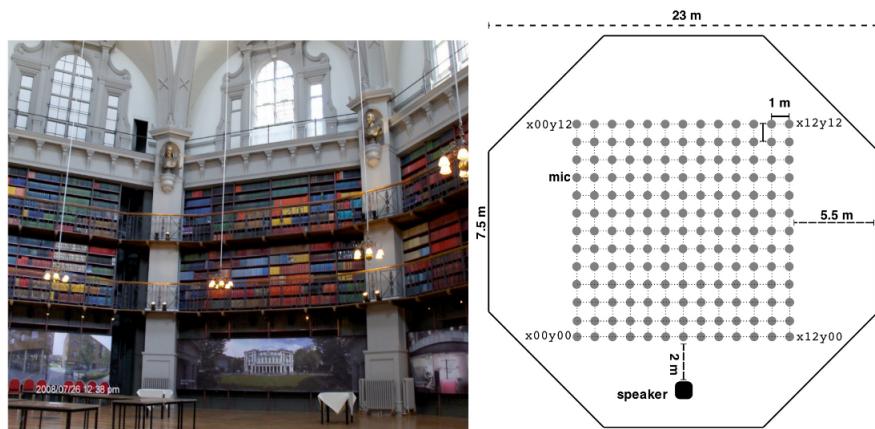


Figura 19: Recinto octagon y esquema de medición de respuestas al impulso.

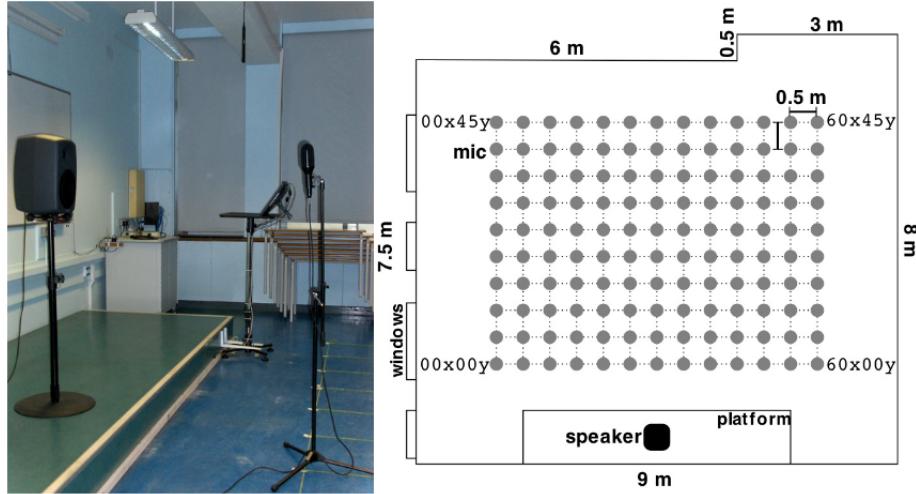


Figura 20: Recinto classroom y esquema de medición de respuestas al impulso.

Las mediciones fueron realizadas empleando la técnica del barrido frecuencial [41], utilizando un altavoz Genelec 8250A y un micrófono omnidireccional DPA 4006. El altavoz mencionado es un transductor de dos vías, con un driver de 8' para frecuencias bajas-medias y un driver de 1' para frecuencias altas. En la fotografía presente en la Figura 20 se puede observar tanto el altavoz como el micrófono dispuestos para realizar una medición. Se miden respuestas para una única posición de fuente y varias posiciones de micrófono. Las posiciones de micrófonos se definen formando una grilla de puntos equiespaciados dentro de los recintos para realizar un mapeo uniforme, como se observa en los esquemas de medición presentados. En la Tabla 1 se puede ver el tiempo de reverberación por bandas para cada recinto.

Tabla 1: Tiempos de reverberación por bandas para cada recinto.

	125 Hz	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz
Class room	1.80 ± 1.12	2.09 ± 0.12	2.05 ± 0.04	1.86 ± 0.02	1.99 ± 0.02	1.61 ± 0.013
Great Hall	2.19 ± 1.71	2.16 ± 0.29	2.40 ± 0.07	2.44 ± 0.06	2.30 ± 0.06	1.75 ± 0.06
Octagon	2.40 ± 1.73	2.34 ± 0.11	2.99 ± 0.05	3.26 ± 0.04	2.91 ± 0.03	2.23 ± 0.03

4.2.2 Respuestas al impulso generadas

En cuanto a las respuestas al impulso generadas, se utilizó la librería de Python 'PyRoomAcoustics' [60] para sintetizarlas. Esta biblioteca brinda un software de generación de respuestas al impulso basado en el método de fuente imagen [61]. El algoritmo está implementado en el

lenguaje de programación C, permitiendo una rápida simulación de la propagación del sonido en recintos poliédricos. Los parámetros que se deben indicar a la hora de generar una respuesta al impulso son:

- Dimensiones del recinto (largo, ancho y alto).
- Posiciones de fuente y receptor, en coordenadas tridimensionales.
- Coeficientes de absorción de las superficies.
- Orden máximo de reflexiones a computar.

Para generar los datos se proponen dos recintos, el primero de dimensiones $8m \times 6m \times 4m$ que se denominará 'Recinto 1', el segundo de dimensiones $6m \times 4m \times 3,5m$ que se denominará 'Recinto 2'. La cantidad de recintos generados y sus dimensiones se definen de acuerdo a los trabajos del estado del arte utilizados como referencia [27], [28]. En la Figura 21 se pueden visualizar ambos recintos generados.

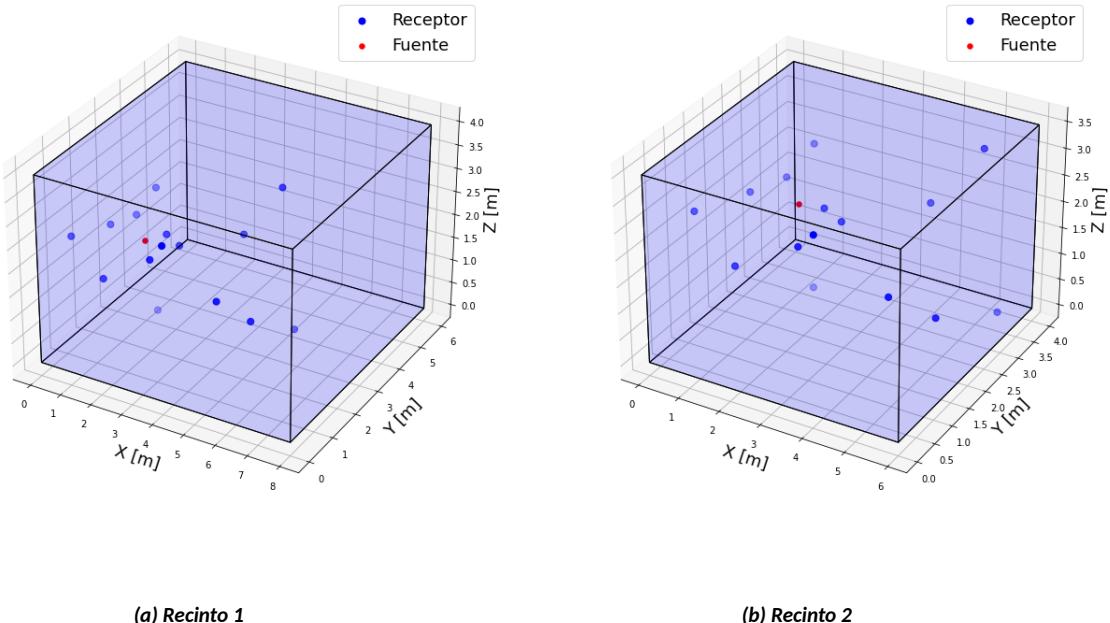


Figura 21: Recintos y puntos receptor-fuente generados para la simulación de respuestas al impulso.

Para controlar los demás parámetros que refieren a las condiciones del recinto, se subordina el orden máximo de reflexiones y los coeficientes de absorción a un tiempo de reverberación

esperado. Esto es, teniendo un cierto recinto se determina un valor de tiempo de reverberación T60 inicial. Este se utiliza para estimar un valor de un coeficiente de absorción promedio mediante la ecuación de Sabine y también en base a este tiempo se determina el orden de reflexiones necesario para poder representar la reverberación. Por último, las posiciones de fuente y receptor se generan aleatoriamente para poder generar diferentes respuestas al impulso a partir de un mismo recinto. De esta manera, los datos que se deben determinar son las dimensiones del recinto, un tiempo de reverberación inicial y la cantidad de respuestas al impulso que se busca generar.

Con esto, para formar el conjunto de respuestas al impulso generadas de entrenamiento se utilizó el recinto 1 para 3 tiempos de reverberación principales: 0,5s como reverberación baja, 0,75s como reverberación media y 1,0s como reverberación alta. Partiendo de estos tiempos, se generan 30 respuestas al impulso para cada uno, variando aleatoriamente los puntos de fuente y receptor. Esto resulta en un total de 90 respuestas al impulso con tiempos de reverberación de entre aproximadamente 0,5 segundos a 1,0 segundos. Para generar las respuestas destinadas a evaluación se realiza el mismo procedimiento pero utilizando el recinto 2 y generando 15 respuestas por cada tiempo de reverberación, formando un total de 45 respuestas al impulso generadas.

4.2.3 Respuestas al impulso generadas por aumentación

Este conjunto se obtiene a partir de respuestas al impulso reales, que generalmente son pocas y no logran una buena cobertura de los parámetros acústicos como el T60 y el DRR. Es posible, mediante técnicas de procesamiento de señales, alterar los parámetros acústicos de una respuesta al impulso, generando nuevas respuestas al impulso con distintas características acústicas. De esta forma, se puede ampliar considerablemente la cantidad de respuestas al impulso del conjunto de datos, a partir de alteraciones a un conjunto pequeño de respuestas al impulso reales, y lograr una mejor cobertura de los parámetros acústicos de interés [62]. En este trabajo, se utilizaron dos procesos para aumentar las respuestas al impulso reales: una alteración de amplitud en la parte temprana de la respuesta al impulso para controlar la relación directo-reverberado, y una alteración de envolvente de caída para controlar el tiempo de reverberación.

Para el primer proceso, a la parte correspondiente al sonido directo $h_e(t)$ ² se le aplica una ganancia definida por un factor α el cual se calcula para obtener el valor de DRR deseado generando una nueva señal $\tilde{h}_e(t)$. Para evitar generar discontinuidades durante el proceso, se aplican ventanas complementarias a la señal directa obteniendo una señal directa ventaneada y un residuo ventaneado. A partir de esto, la parte directa se puede definir según la ecuación 13.

$$h_e(t) = \alpha w_d(t) h_e(t) + [1 - w_d(t)] h_e(t) \quad (13)$$

En donde $w_d(t)$ corresponde a una ventana Hann de 5ms de longitud. De esta manera, partiendo de esta última definición junto con la expresión del parámetro DRR expresado en la ecuación 9 se plantea un sistema de ecuaciones a partir del cual se puede definir un valor pretendido de DRR y despejar el correspondiente valor de α . En la Figura 22 se puede observar una representación de una parte directa $h_e(t)$, las ventanas aplicadas, el efecto del factor de ganancia α y la nueva señal $\tilde{h}_e(t)$ generada. Finalmente, esta parte directa modificada se concatena con el resto de la respuesta al impulso completando así el proceso de aumentación referido a la relación directo-reverberado. Se debe tener en cuenta que para tiempos de reverberación cortos, puede ser un problema generar relaciones directo-reverberado demasiado bajas, ya que la energía de la parte tardía ya es de por si muy baja. Para esos casos, se definen valores límites para la ganancia aplicada a la parte temprana.

Luego, para la modificación del tiempo de reverberación T_{60} se trabaja únicamente con la parte tardía de la respuesta al impulso. Esta puede ser modelada como ruido Gaussiano con una caída de nivel exponencial dependiente de la frecuencia, sumado a un determinado piso de ruido. Como esta pendiente de caída varía con la frecuencia, se analiza la respuesta al impulso en bandas de tercio de octava para contemplar esta dependencia frecuencial. Este modelo se expresa en la ecuación 14.

$$h_m(t) = A_m e^{\frac{-(t-t_0)}{\tau_m}} n(t) u(t - t_0) + \sigma_m n(t) \quad (14)$$

²Comúnmente se considera que los primeros 2,5 ms corresponden al sonido directo. Esto representa una diferencia de camino de 0,85 m para una velocidad del sonido de 340 ms⁻¹. Las normas de medición de respuestas al impulso establecen condiciones de posición de manera tal que la primera reflexión ocurre luego de los 2,5 ms

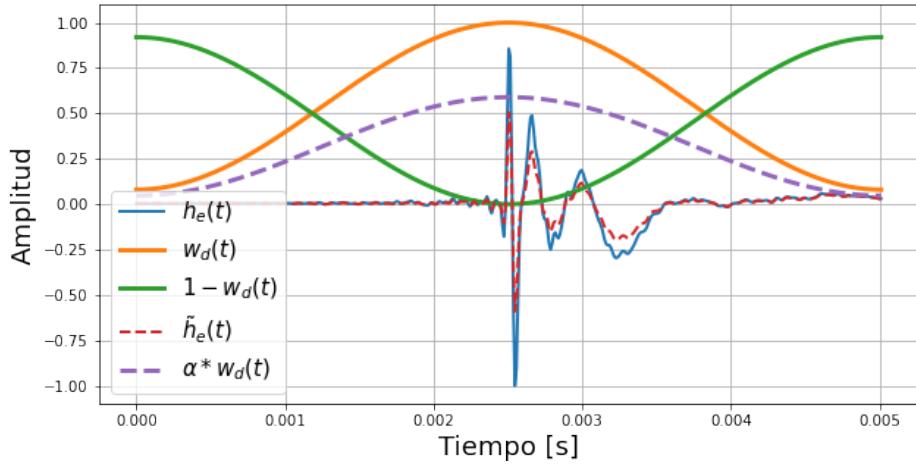


Figura 22: Señales involucradas en el proceso de aumento de DRR.

En donde A_m es la amplitud inicial, τ_m es la tasa de caída, σ_m es el nivel del piso de ruido, $n(t)$ es ruido Gaussiano de media cero y desvío estándar unitario, t_0 es el instante temporal en donde comienza la parte tardía de la respuesta al impulso, m es el índice que refiere a una sub-banda frecuencial y $u(t)$ es un escalón unitario. En este modelo, el tiempo de reverberación T_{60} se relaciona directamente con el parámetro τ según la ecuación 15.

$$T_{60} = \ln(1000)\tau T_s \quad (15)$$

En donde T_s es el período de muestreo. Dado este modelo, se aplican métodos de optimización no lineales para estimar el conjunto de parámetros $\{\hat{A}_m; \hat{\tau}_m; \hat{\sigma}_m\}$ que mejor aproximen la envolvente de caída de la respuesta al impulso. Con estos parámetros sumados a la tasa de caída deseada $\tau_{m,d}$ calculada a partir del tiempo de reverberación deseado, se modifica la parte tardía de la respuesta al impulso inicial multiplicándola por una envolvente exponencial creciente o decreciente según corresponda como se muestra en la ecuación 16.

$$h_m'(t) = h_m(t) e^{-(t-t_0) \frac{\hat{\tau}_m - \tau_{m,d}}{\hat{\tau}_m \tau_{m,d}}} \quad (16)$$

En donde $h_m'(t)$ representa a la nueva parte tardía de la respuesta al impulso generada para obtener el tiempo de reverberación deseado. En términos generales, el proceso consiste en modificar la pendiente de caída para obtener la pendiente de caída deseada por cada banda

de frecuencia. Al final, las sub-bandas generadas se suman para obtener el resultado final que contemple todo el espectro de la señal. Hasta aquí este proceso funciona satisfactoriamente cuando se generan tiempos de reverberación menores al de la respuesta al impulso inicial, es decir, siempre que se multiplica la respuesta al impulso por envolventes exponenciales decrecientes. En cambio, cuando se busca generar tiempos de reverberación mayores la envolvente por la que se multiplica la respuesta inicial es creciente, lo que produce una amplificación de la parte tardía de la respuesta al impulso. Esto muchas veces equivale a amplificar el piso de ruido presente en la señal, lo que puede producir pendientes de caída inestables que no se corresponden con el comportamiento propio de la respuesta al impulso ya que no es información del sistema acústico sino simplemente ruido. Para evitar este efecto adverso del proceso de aumentación anteriormente propuesto se debe estimar el piso de ruido de la respuesta al impulso. Esto se realiza a través del método iterativo propuesto por Lundeby et. al. [63]. Una vez estimado el piso de ruido de la señal, la respuesta final se obtiene haciendo un cross-fade en el inicio del piso de ruido entre la parte tardía generada y una cola reverberante sintética creada a partir de multiplicar ruido Gaussiano con una envolvente exponencial decreciente, utilizando los parámetros previamente calculados. Una explicación más detallada de este proceso se puede encontrar en el anexo A.

Para realizar este proceso se determinan límites de relación directo-reverberado y tiempo de reverberación medio. La relación directo-reverberado va desde -3 dB a 10 dB con saltos de 1 dB , lo que se considera una diferencia de nivel promedio acorde a la mínima perceptible por el oído humano. Con respecto al tiempo de reverberación, se generan desde $0,1\text{ s}$ a $1,2\text{ s}$ para estar dentro del rango de las respuestas al impulso generadas, con un paso de $0,05\text{s}$ basado en estudios previos realizados sobre la mínima diferencia perceptible entre tiempos de reverberación [64]. Una vez obtenido el conjunto de respuestas al impulso aumentadas, se seleccionan aleatoriamente 135 respuestas para equiparar al número de respuestas al impulso generadas.

4.3 BASES DE DATOS DE SEÑALES DE HABLA

Las señales de habla necesarias para formar los pares anecoico-reverberados se obtienen de la librería LibriSpeech [65], la cual consiste en un conjunto de datos que reúne 100 horas de audio correspondientes a lecturas en idioma inglés. Los datos corresponden a programas tipo

audiolibros. Las señales poseen bajo nivel de reverberación, y provienen de una aplicación en la cual la inteligibilidad es primordial, lo cual hace que esta base de datos sea adecuada para utilizarse en este trabajo.

4.3.1 Pre-procesamiento de datos

Partiendo de audios de voz y respuestas al impulso, el modelo de red neuronal propuesto requiere generar instancias de espectrogramas de magnitud correspondientes a audios anecoicos y reverberados. Para conseguir esto, se programa una cadena de procesamiento automatizada que realice esta transformación de los datos de entrada. En primer lugar se controla la uniformidad de frecuencias de muestreo aplicando las transformaciones de aumentación o decimado cuando sean requeridas. Se decide trabajar con una frecuencia de muestreo de 16000 muestras por segundo, considerando que se trata con señales de voz que concentran su información por debajo de la frecuencia de Nyquist de 8000 Hz. Luego, los audios de voz se convolucionan con las respuestas al impulso para formar pares de señales con y sin reverberación. El resultado de la convolución se recorta para descartar el retardo generado por la convolución, haciendo que los pares de señales sean sincrónicas. Luego, se toman ventanas rectangulares de 32640 muestras, lo que equivale a segmentos de audio de 2,04 segundos para la frecuencia de muestreo utilizada. Lo siguiente es aplicar la transformada de corto término de Fourier tanto a la señal limpia como a la señal convolucionada. La transformada se aplica con una ventana de 512 muestras y un salto de 128 muestras lo cual equivale a un solapamiento del 75 %, el cual permite una correcta reconstrucción de la señal. El tamaño de ventana y de salto se escoge de acuerdo a los trabajos del estado del arte [27], [28]. Se obtienen espectrogramas complejos, a los cuales se les calcula la magnitud, descartando la información de fase. Además, se aplica una normalización para acotar el dominio en valores que sean convenientes para el algoritmo de aprendizaje posterior.

Finalmente, las instancias finales de este proceso son el espectro de magnitud de la señal con reverberación (que corresponde a la variable de entrada de la red neuronal) y el espectro de magnitud de la señal sin reverberación (que corresponde a la salida deseada). Ambas instancias tienen las mismas dimensiones, que corresponden a 256 cuadros temporales y 257 cuadros frecuenciales (se conserva sólo la parte positiva del espectro frecuencial simétrico). Por último, se

descartan los puntos correspondientes al valor máximo de frecuencia. Esto se realiza para obtener dimensiones finales de 256×256 lo cual facilita el diseño de la red neuronal convolucional, al ser dimensiones múltiplos de 2. Se descarta la frecuencia más alta, lo cual no compromete la representación de la señal ya que no habrá información relevante de la misma en la frecuencia de Nyquist.

Cabe destacar que debido a este preprocesamiento aplicado, a la hora de evaluar el modelo se deberán aplicar una serie de procesos previos sobre el audio a procesar. Más precisamente, se deberá segmentar el audio y obtener espectros de magnitud de la STFT respetando los mismos parámetros que en el preprocesamiento. Luego, como la salida de la red es una máscara de amplitud comprimida, se debe descomprimir esta máscara, aplicarla sobre el espectro reverberado y luego combinar el espectro de amplitud modificado resultante con la fase original de la señal para poder finalmente obtener la información de audio de salida (ya sea a través de la aplicación del algoritmo de Griffin-Lim, o bien simplemente combinando la fase del espectrograma reverberado con la magnitud dereverberada).

4.4 MODELO PROPUESTO

El modelo propuesto se basa en una arquitectura de red neuronal completamente convolucional tipo 'autoencoder' inspirada en el trabajo de Ernst et. al. [27]. En este autoencoder, la señal de salida es la señal de entrada sin reverberación. Este tipo de variantes de autoencoder se denominan denoising autoencoders [66], y son un patrón de diseño muy común en problemas de mejora del habla. En este trabajo, en vez de estimar directamente el espectrograma anecoico, la red estima máscaras que al ser multiplicadas con la señal reverberada de entrada, dan como resultado el espectrograma anecoico. En estudios previos, se demostró que este método da mejores resultados que estimar directamente el espectrograma anecoico [67]. Para trabajar con espectros, las señales de entrada se transforman al dominio temporal-frecuencial a partir de la transformada de Fourier de corto término.

La estructura de red neuronal utilizada consiste en una U-NET con conexiones de salto, inspirada inicialmente en [27]. Este tipo de estructura consiste en tomar mapas bidimensionales de entrada y a partir de la aplicación sucesiva de capas convolucionales con valores de salto mayor a 1, reducir la dimensionalidad del mismo e ir aumentando el número de filtros utilizados

por las capas convolucionales. Un esquema básico de esta estructura se puede ver en la Figura 23, en donde se puede observar que las dimensiones de las capas siguen una forma de 'U', lo cual le da el nombre a estas estructuras.

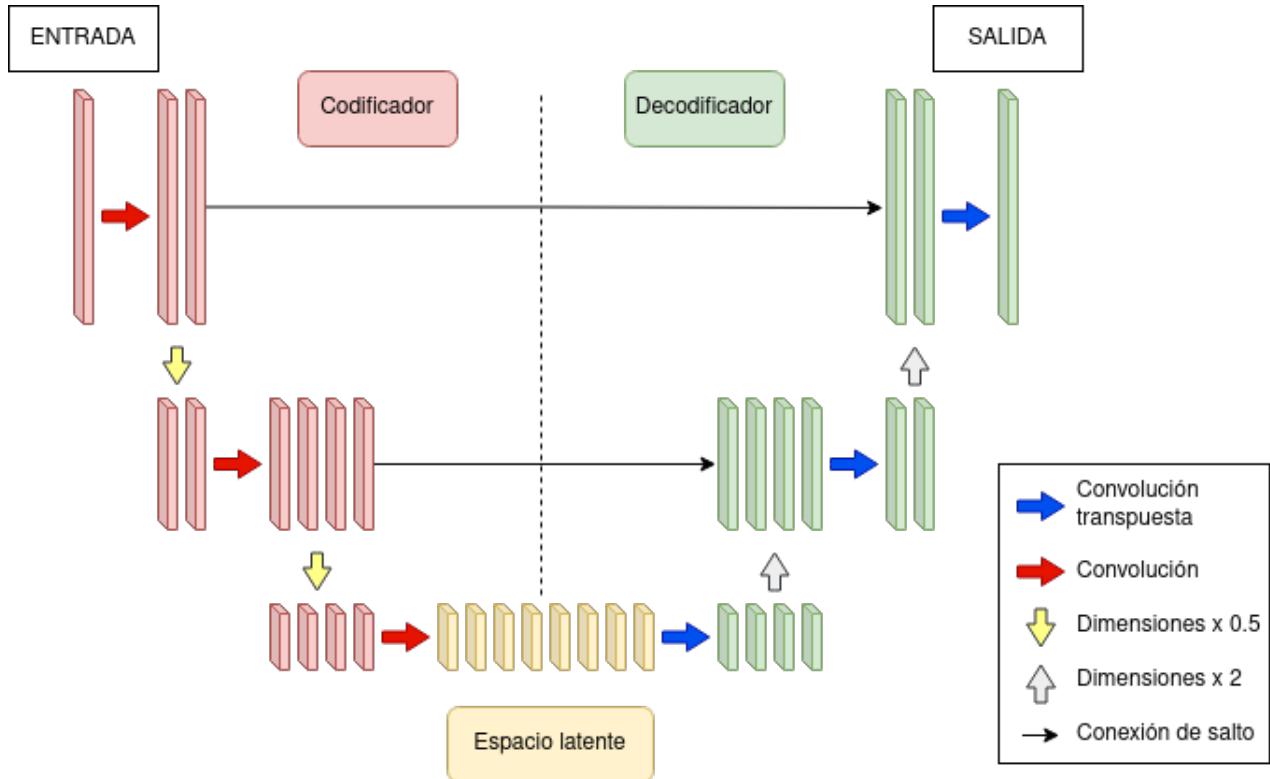


Figura 23: Esquema básico de una red tipo 'U-NET' con conexiones de salto.

A medida que se avanza en el modelo, las dimensiones de la imagen de entrada disminuyen y la cantidad de filtros utilizados aumentan. Esta primera etapa equivale a la función de codificación f de un autoencoder. El proceso se realiza sucesivamente hasta que las dimensiones de la imagen son de 1×1 , lo cual se corresponde con el espacio latente h . Luego, prosigue una etapa de decodificación en la cual se aplica el proceso inverso. Esto es, las dimensiones van aumentando con capas de convolución transpuesta configuradas con tamaños de saltos mayores a 1, y la cantidad de filtros utilizados va disminuyendo. Esto se repite hasta que las dimensiones del tensor sean las mismas que tenía a la entrada del codificador. Mediante este esquema de U-NET y el efecto del cuello de botella de las dimensiones, se consigue que la estimación de cada punto del espectrograma anecoico esté condicionado por todos los puntos que componen el espectrograma reverberado de entrada, y no solo por una región particular del mismo. Para poder pasar información de manera más directa desde el decodificador hacia el codificador, se

implementan conexiones de salto. La conexión de salto consiste en concatenar la salida de una capa del codificador con la de una capa del decodificador. Para poder hacerlo, la dimensión de concatenación (en este caso, las dimensiones del spectrograma) deben ser las mismas. De esta manera se logran decodificaciones más precisas, y se solucionan problemas como el desvanecimiento de gradiente [26].

Una representación gráfica del modelo final implementado se puede apreciar en la Figura 24. En cada capa se indican tres valores, donde el primero representa la dimensión temporal, el segundo la dimensión frecuencial y el tercero el número de canales (equivalente a la cantidad de filtros). En las primeras capas, las dimensiones se reducen a la mitad en cada instancia debido al uso de un desplazamiento de paso 2 en el cálculo de los filtros convolucionales. En las capas subsiguientes, las dimensiones sufren el efecto contrario hasta volver a obtener las dimensiones originales. El uso de las conexiones de salto permite que en el proceso de codificación, en el cual se reduce la dimensionalidad, no se pierda información detallada del spectrograma de entrada. Estos detalles presentes en las primeras capas del codificador, podrán ser aprovechados por las últimas capas del decodificador al existir las conexiones de salto. A su vez, el decodificador tendrá acceso a información global y general del spectrograma, la cual se encuentra codificada en el espacio latente.

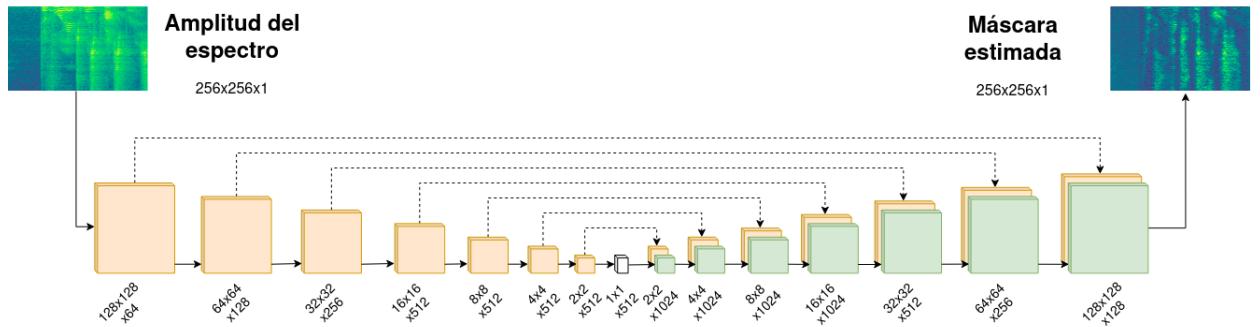


Figura 24: Modelo de red neuronal convolucional implementado.

4.5 DETALLES DE IMPLEMENTACIÓN

En la arquitectura implementada, se imitaron decisiones tomadas en el trabajo de Ernst et al. [27]. Se utilizan activaciones tipo ReLu combinadas con LeakyReLu de pendiente 0,2. Se escogen estas activaciones debido a que son favorables frente al problema de desvanecimiento

de gradiente, que suele ser común en redes muy profundas. Se emplean técnicas de regularización como dropout y también se realiza normalización por lotes (BatchNorm). En la Tabla 2 se especifican las características de la arquitectura implementada.

Tabla 2: Especificaciones de la arquitectura implementada.

Capa	ENCODER			DECODER		
	Filtros	Activación	Regularización/ Normalización	Filtros	Activación	Regularización/ Normalización
1	64	LeakyReLU	-	512	ReLU	BatchNorm-Dropout
2	128	LeakyReLU	BatchNorm	512	ReLU	BatchNorm-Dropout
3	256	LeakyReLU	BatchNorm	512	ReLU	BatchNorm-Dropout
4	512	LeakyReLU	BatchNorm	512	ReLU	BatchNorm
5	512	LeakyReLU	BatchNorm	256	ReLU	BatchNorm
6	512	LeakyReLU	BatchNorm	128	ReLU	BatchNorm
7	512	LeakyReLU	BatchNorm	64	ReLU	BatchNorm
8	512	ReLU	BatchNorm	1	ReLU ¹	BatchNorm

¹ Se cambia a ReLU limitada en 1 a la hora de hacer predicciones.

Las capas que componen el codificador son convolucionales, y las que componen el decodificador son capas deconvolutivas o de convolución transpuesta. Las capas de convolución transpuesta pueden generar artefactos indeseados en los mapas de características [68]. Una manera de solucionar este efecto es asegurándose de que los campos receptivos sean múltiplos enteros del tamaño de salto utilizado [69]. Sin embargo, aun teniendo esto en consideración, las capas de convolución transpuesta pueden seguir generando artefactos. Entonces, en lugar de utilizar capas de convolución transpuesta se opta por implementar una combinación de dos capas consecutivas: en primer lugar una capa que aumente las dimensiones del espectrograma generando nuevos puntos a partir de una interpolación entre los valores más cercanos, y luego una capa convolucional. De esta manera se logra aumentar la dimensionalidad de la entrada y utilizar una operación de convolución sin generar la aparición de artefactos indeseados [70]. En todas las capas tanto del codificador como del decodificador se utiliza un tamaño de filtro de 6x6 y un tamaño de salto igual a 2.

Finalmente, la función de costo utilizada para evaluar las predicciones realizadas por el modelo frente a las máscaras ideales en la salida es el error cuadrático medio (MSE), el cual se expresa en la ecuación 17. Para la optimización se utilizó el algoritmo de estimación adaptativa de momento (ADAM) [71] con un valor de tasa de aprendizaje de 0,001.

$$L_{MSE} = \sum_{i=1}^{N-1} (M_i(t, f) - \hat{M}_i(t, f))^2 \quad (17)$$

La arquitectura de red neuronal se implementó utilizando el lenguaje de programación Python (versión 3.7), particularmente haciendo uso de la biblioteca Tensorflow³, la cual permite el desarrollo y entrenamiento de modelos de aprendizaje por máquina.

4.6 EVALUACIÓN DEL MODELO

En este trabajo se evalúa el impacto que provoca la aumentación, simulación y ordenamiento de datos en el desempeño del modelo. Particularmente, se analiza el aporte de realizar aumentación de respuestas al impulso reales, y de sintetizar nuevas respuestas al impulso. También se evalúa el efecto del ordenamiento de los datos durante el entrenamiento de la red.

4.6.1 Combinaciones de bases de datos

En primer lugar, como se cuenta con respuestas al impulso reales, generadas y aumentadas, se prueban combinaciones de estos datos a la hora de formar los diferentes conjuntos de entrenamiento y evaluación. Es decir, se entrena el modelo utilizando un determinado conjunto, y luego se evalúa su funcionamiento sobre el total de los conjuntos. Esto resulta en 3 pruebas, en donde los conjuntos de entrenamiento y evaluación quedan determinados según la Tabla 3. Cabe aclarar que los conjuntos utilizados para entrenamiento no son los mismos que los utilizados para evaluación en cada prueba. Es decir, cuando por ejemplo se entrena con reales y se evalúa sobre reales, en cada proceso se utilizan conjuntos de datos distintos.

³Página oficial de Tensorflow: <https://www.tensorflow.org/>

Tabla 3: Configuración del primer conjunto de pruebas.

	Prueba 1	Prueba 2	Prueba 3
Conjunto de entrenamiento	Reales	Generadas	Aumentadas
Conjuntos de evaluación	Reales Generadas Aumentadas	Reales Generadas Aumentadas	Reales Generadas Aumentadas

Luego, se evalúa el desempeño del modelo propuesto al utilizar combinaciones de conjuntos en la etapa de entrenamiento. Como el objetivo principal es contar con una mayor cantidad y variedad de respuestas al impulso, en vez de solo utilizar respuestas grabadas, las combinaciones propuestas consisten en combinar las respuestas reales con las generadas y aumentadas como se indica en la Tabla 4.

Tabla 4: Configuración del segundo conjunto de pruebas.

	Prueba 1	Prueba 2	Prueba 3
Conjunto de entrenamiento	Reales + Aumentadas	Reales + Generadas	Reales + Aumentadas + Generadas
Conjuntos de evaluación	Reales Generadas Aumentadas	Reales Generadas Aumentadas	Reales Generadas Aumentadas

4.6.2 Ordenamiento de los datos durante el entrenamiento

Por otro lado, se evalúa la influencia del orden con el que las instancias de entrenamiento se le presentan a la red. Normalmente, durante el entrenamiento de una red neuronal, los datos se ordenan de forma aleatoria. Sin embargo, diversos trabajos [51], [72], [73], demuestran que ordenar los datos de forma creciente en dificultad es beneficioso y lleva a un mejor desempeño de los modelos. Esta técnica se denomina aprendizaje por currículum, y para este trabajo, el

criterio que se utilizó para medir la dificultad es que a mayor tiempo de reverberación (TR), más difícil es deneverberar una señal. En consecuencia, se evaluó el modelo entrenado con los datos en un orden creciente de TR (currículum), y para evaluar la efectividad del método, se comparó su desempeño con entrenar el modelo con un orden decreciente de TR (anti-currículum), y con un orden aleatorio.

Para esta evaluación del orden de los datos en el entrenamiento, es necesario generar un conjunto de datos anecoicos-reverberados de manera controlada, asegurando una distribución homogénea de los tiempos de reverberación presentes en el conjunto final. Para conseguir esto se decide conformar el conjunto de datos a partir de respuestas al impulso aumentadas, ya que estas se pueden generar controlando paramétricamente el TR medio y la DRR. Se busca cubrir el rango de tiempo de reverberación medio desde 0,1 s hasta 3,5 s con variaciones de entre –10 dB a 10 dB de relación directo-reverberado.

CAPÍTULO 5: RESULTADOS Y DISCUSIONES

El código desarrollado a lo largo de este trabajo se encuentra disponible en un repositorio público en línea junto con su correspondiente documentación [74].

5.1 BASES DE DATOS DE RESPUESTAS AL IMPULSO

Para tener una medida de la variedad de reverberación presente en los conjuntos de respuestas al impulso reales, generadas y aumentadas, se utilizaron los parámetros TR_{mid} y DRR . En las Figuras 25, 26, y 27 se muestran los parámetros acústicos anteriormente mencionados para cada conjunto de respuestas al impulso utilizado.

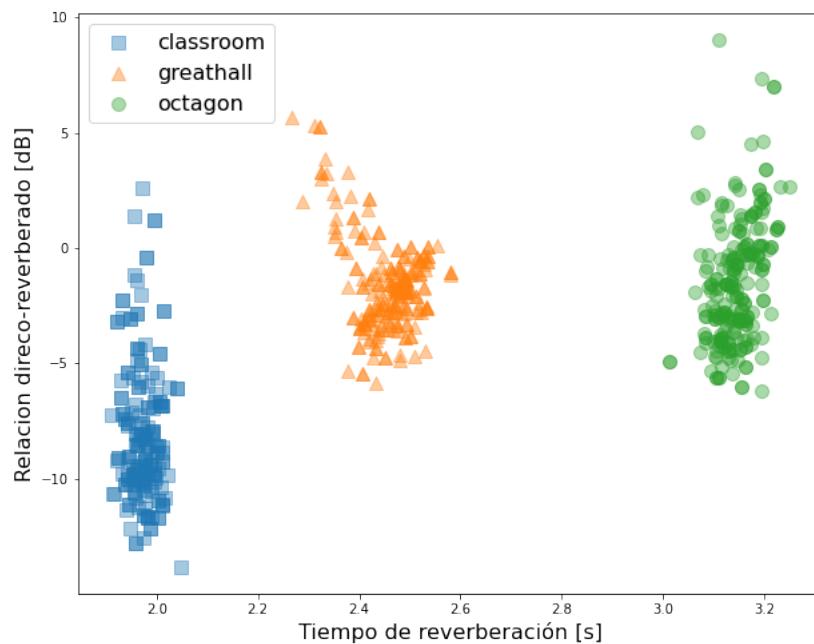


Figura 25: Conjunto de respuestas al impulso reales.

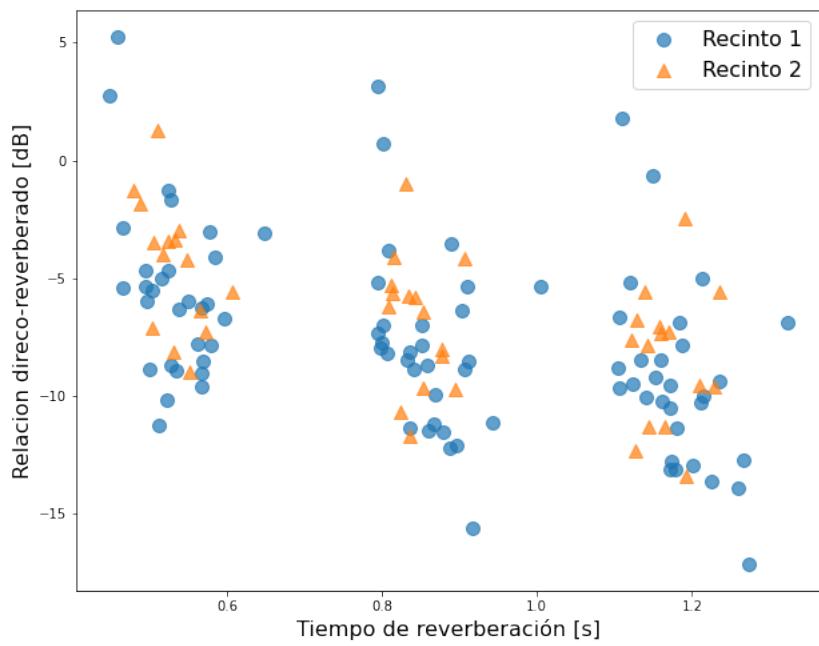


Figura 26: Conjunto de respuestas al impulso generadas.

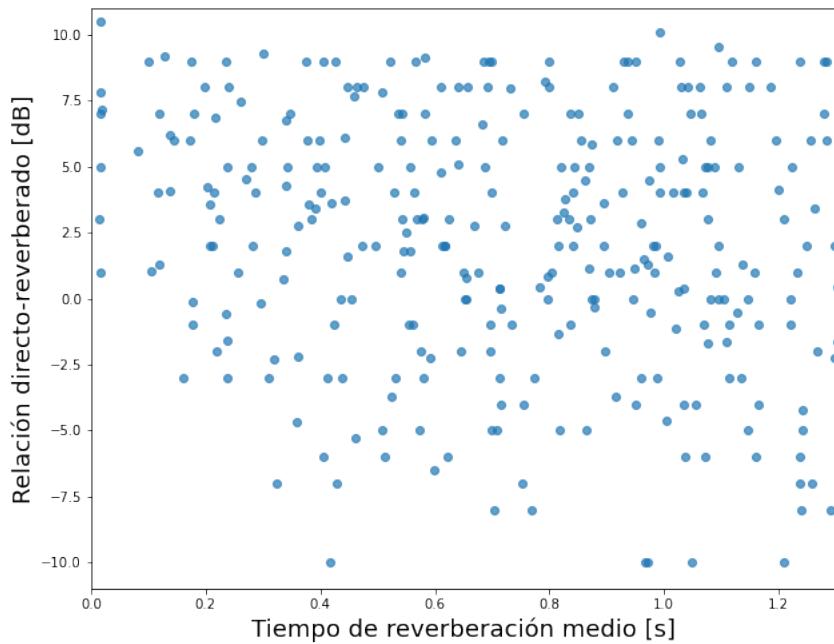


Figura 27: Conjunto de respuestas al impulso aumentadas.

Del análisis de estos conjuntos utilizando como medida la relación directo reverberado y el tiempo de reverberación medio se pueden observar algunas particularidades. Respecto a las respuestas al impulso reales, en la Figura 25 se pueden distinguir tres grandes grupos de puntos de acuerdo a los tres recintos de los cuales fueron obtenidas dichas respuestas.

Las variaciones de ambos parámetros acústicos ocurren debido a las diferentes posiciones de micrófono que han sido utilizadas. Los recintos poseen tiempos de reverberación medios de $1,97 \pm 0,03$ s, $2,46 \pm 0,06$ s y $3,14 \pm 0,04$ s. De igual manera, se producen variaciones de DRR en un rango de 15 dB, 11 dB y 16 dB respectivamente. Por ende, las respuestas al impulso reales cubren, de una forma poco homogénea, un rango de TR entre 1,97 s y 3,14 s. Algo similar ocurre con las respuestas al impulso generadas que se muestran en la Figura 26. Si bien en este caso se tiene control sobre los puntos centrales de los conjuntos de puntos (se generaron para tiempos de reverberación de 0,5 s, 0,75 s y 1,0 s) ocurre el mismo fenómeno que con las respuestas al impulso reales, en donde se forman grupos de puntos que no se dispersan uniformemente en el plano. Esto cambia para el tercer conjunto que corresponde a las respuestas al impulso generadas a partir del proceso de aumentación. La dispersión de estas respuestas se observa en la Figura 27. A primera vista se observa una mayor uniformidad de los puntos en el plano, ya que no se aprecian conjuntos separados sino más bien una aleatoriedad uniforme a lo largo del rango generado. La uniformidad de la dispersión la podemos atribuir al control que se tiene sobre estos parámetros a la hora de generar las respuestas aumentadas, y la aleatoriedad entre los puntos se debe al hecho de que siempre se parte de una respuesta al impulso real diferente para realizar la aumentación, lo cual produce que el margen entre los parámetros deseados y los obtenidos sea variable. Por otro lado, parece haber una menor cantidad de puntos en la esquina inferior izquierda del grafico, es decir, tiempos de reverberación bajos con relaciones directo-reverberado bajos. Esta es una limitación tanto propia del algoritmo de aumentación como también de la naturaleza de las respuestas al impulso reales, en donde para tiempos de reverberación bajos la energía de la parte tardía de la respuesta es de por sí baja.

5.2 ANÁLISIS CUALITATIVO DEL SISTEMA

En la Figura 28 se muestra una instancia de ejemplo del funcionamiento del algoritmo de dereverberación implementado. Durante el entrenamiento, el espectrograma reverberado in-

gres a la red neuronal para procesarse y generar una máscara de amplitud. Esta máscara se multiplica con el mismo spectrograma reverberado de entrada para generar el spectrograma dereverberado, que es la salida de la red. Luego, se calcula el error entre la salida de la red y el spectrograma anecoico correspondiente, y se propaga a los parámetros de la red. A la hora de hacer predicciones, solo se necesita ingresar un spectrograma reverberado para que la red estime una máscara de amplitud con la cual pueda generarse el spectrograma dereverberado. Cabe aclarar que a lo largo de estos procesos se trabaja únicamente sobre la magnitud de la STFT, a lo que se hace referencia como spectrograma.

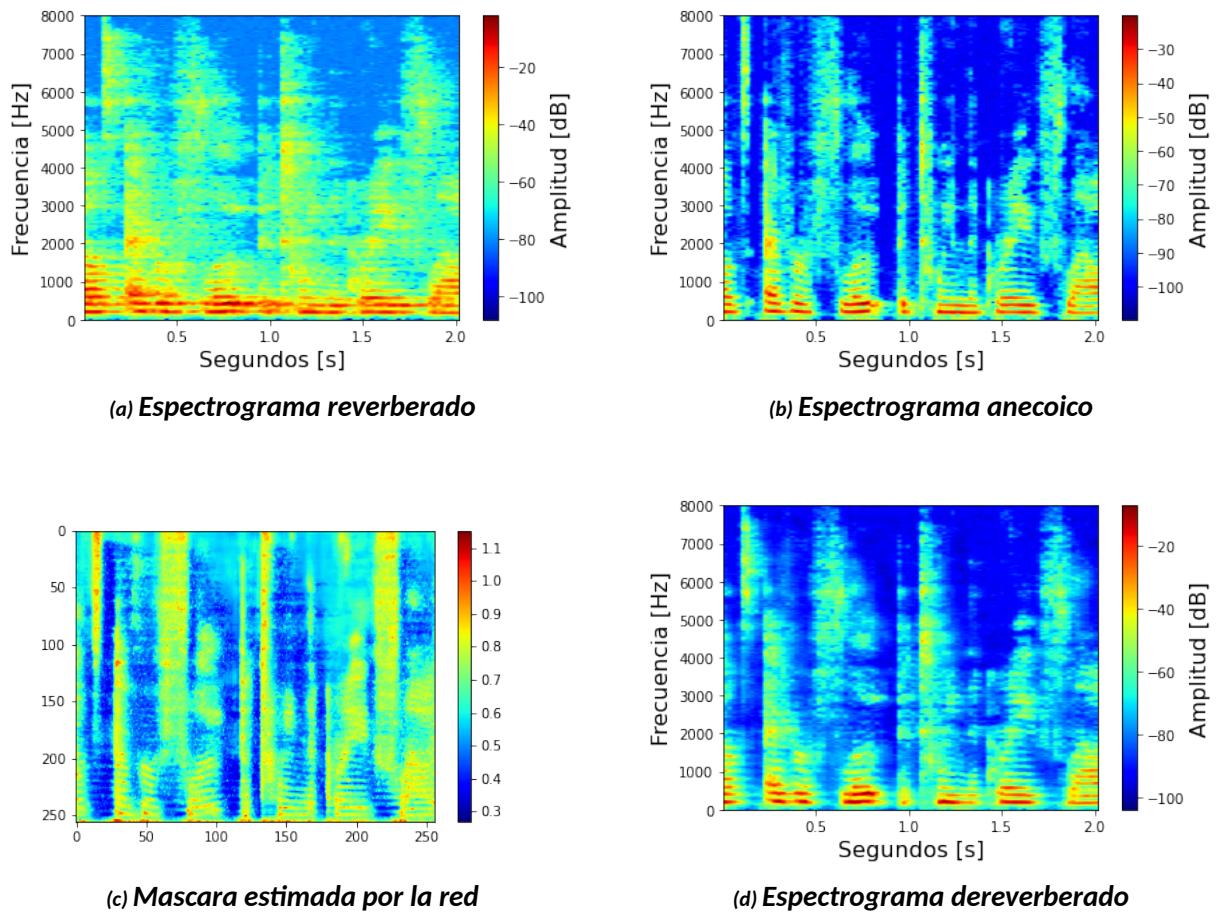


Figura 28: Ejemplo de procesamiento de audio reverberado.

Se puede observar en el ejemplo de la Figura 28, que efectivamente, el modelo propuesto luego de ser entrenado es capaz de estimar máscaras que reducen la reverberación conservando la señal de habla. La atenuación de las componentes reverberantes ocurre con mayor precisión para frecuencias bajas, en donde hay más energía. Esto puede deberse al uso del error

cuadrático medio (MSE) como función de costo, el cual prioriza la energía por lo que sesga al modelo a aprender a deneverberar en bajas frecuencias. Algo para tener en cuenta en un trabajo futuro sería evaluar otro tipo de funciones de costo que compensen este sesgo, como por ejemplo utilizar un error cuadrático medio normalizado por el espectrograma objetivo. A pesar de esto, se pueden observar rasgos del espectro reverberado aun presentes en el espectro deneverberado, lo que es de esperarse debido a que el proceso únicamente esta aplicando un filtro de amplitud por sobre la magnitud del espectro reverberado. En la Figura 29 se muestra con más detalle un espectrograma deneverberado con su correspondiente espectrograma anecoico y el resultado del error cuadrático entre ambos. Además de lo mencionado anteriormente, el espectrograma deneverberado conserva los armónicos propios de la señal de habla, evitando así su deterioro.

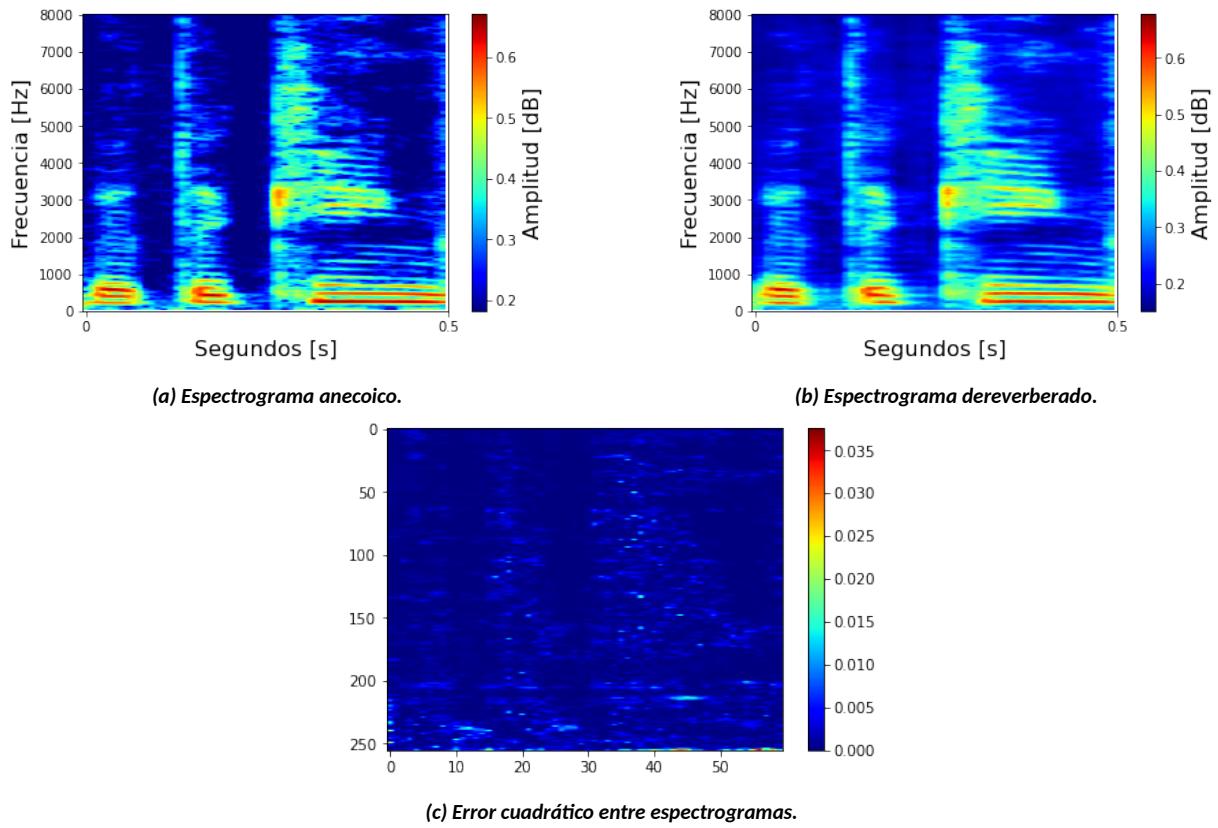


Figura 29: Diferencias entre espectrograma anecoico y deneverberado.

5.3 TRATAMIENTO DE LA FASE

El proceso de dereverberación de los audios sucede sobre la magnitud de los espectros STFT de los audios con reverberación. Una vez estimada la magnitud del espectro dereverberado, es necesario combinar esta magnitud con información de fase, para poder conformar un spectrograma complejo apto para antitransformarse y pasar del dominio temporal-frecuencial al dominio temporal (información de audio). Un ejemplo de los espectros de magnitud y fase para un audio con reverberación y sin reverberación se muestra en la Figura 30.

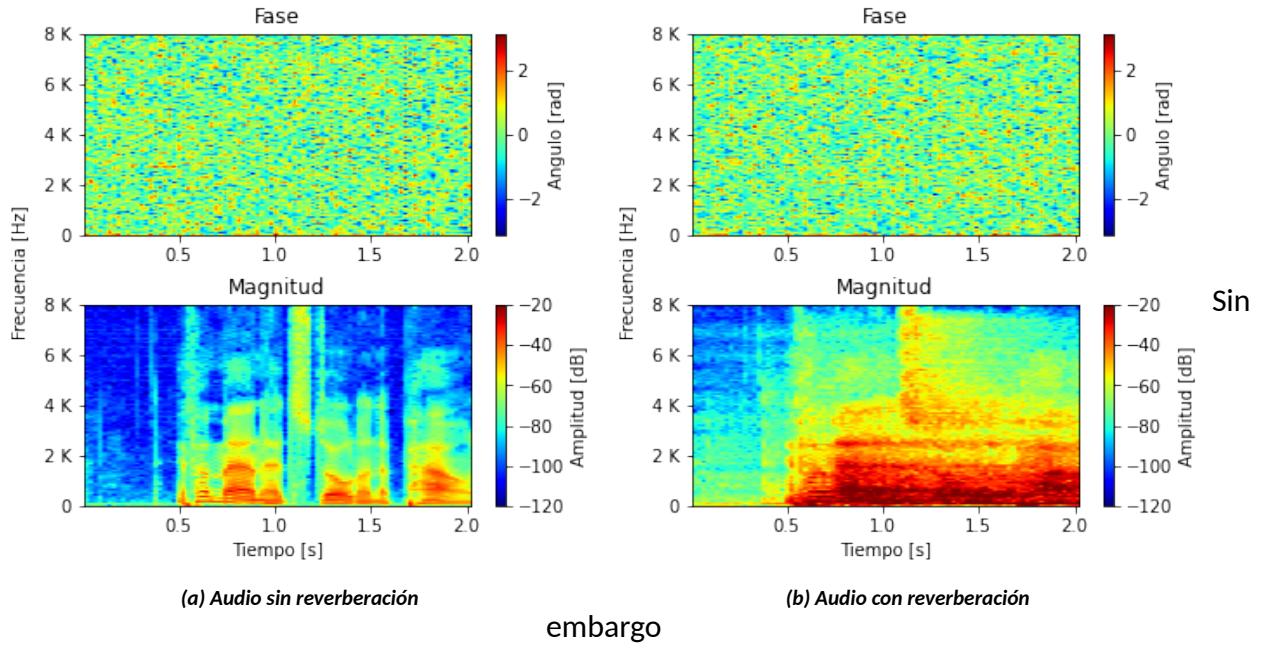


Figura 30: Espectrogramas de magnitud y fase de los audios para entrenamiento.

En tareas de procesamiento de audio para mejora del habla es común trabajar sobre la magnitud de los spectrogramas, dejando la fase inalterada. Esto se debe a que las componentes de magnitud tienen un mayor aporte frente a la inteligibilidad de la palabra [75], [76]. Además, se reduce el costo computacional teniendo que estimar únicamente la magnitud del spectrograma mejorado, en lugar de la magnitud y la fase. Es por esto que, en este trabajo, el proceso de dereverberación se realiza sólo sobre la magnitud de la STFT, como en [27], [28], [32], [33].

Por otro lado, se puede apreciar que los espectros de fase parecen estar menos estructurados a comparación de los espectros de magnitud. Tanto la fase del audio con reverberación como la fase del audio sin reverberación parecen contener información aleatoria, y por inspec-

ción visual son similares entre sí. Sin embargo, manipulando la fase para obtener otro tipo de representaciones, como por ejemplo la frecuencia instantánea (derivada de la fase respecto del tiempo) o el retardo de grupo (derivada de la fase respecto a la frecuencia), es posible obtener información estructural de la señal similar a la presente en la magnitud. De esta forma, estudios recientes que utilizan enfoques similares a los de este trabajo (modificación de espectrogramas mediante arquitecturas U-Net) toman en cuenta la información de fase con la misma importancia que la magnitud, mostrando resultados prometedores [77]. Esto marca un camino de interés para investigaciones futuras.

Para determinar la fase del nuevo espectro de magnitud estimado por la red (dereverberado), se consideraron dos alternativas: utilizar directamente la fase del espectro con reverberación o bien utilizar el método iterativo de Griffin-Lim para estimar la fase a partir de la magnitud dereverberada. Este último método iterativo puede inicializarse con una fase determinada (como la fase del audio reverberado) para aprovechar información existente de manera de mejorar la estimación o puede inicializarse de manera aleatoria. Para determinar el número necesario de iteraciones a utilizar en el algoritmo de Griffin-Lim se evaluó la evolución de las métricas utilizadas en este trabajo (SDR, SRMR y ESTOI) en función de la cantidad de iteraciones. En la Figura 31 se pueden observar estas relaciones para cada métrica, teniendo en cuenta que se utilizó el algoritmo inicializado desde una fase aleatoria. Se puede apreciar que los valores se estabilizan al aproximarse a 100 iteraciones, siendo este el número de iteraciones que se utilizó para las pruebas subsiguientes.

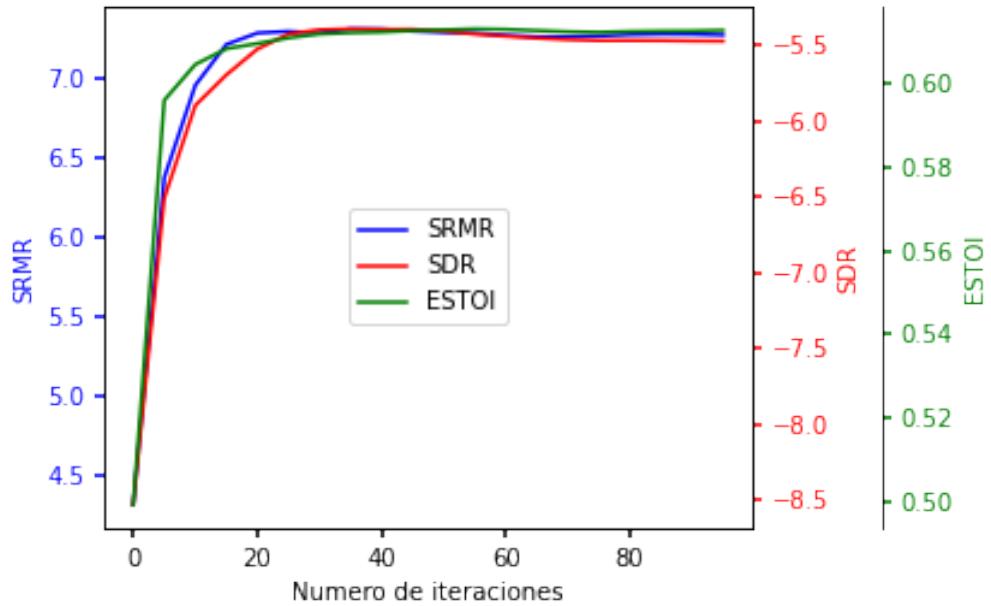


Figura 31: Influencia del número de iteraciones del algoritmo de Griffin-Lim.

En la Tabla 5 se comparan las métricas obtenidas al utilizar las distintas técnicas para invertir el espectrograma. Allí se expresan las variaciones de las métricas para cada alternativa con respecto al audio reverberado. Se puede observar que la principal diferencia ocurre sobre el parámetro SDR. La reconstrucción de fase utilizando el algoritmo de Griffin-Lim inicializado de manera aleatoria empeora el resultado de esta métrica, lo cual es un efecto contrario al deseado. Sin embargo, utilizando este algoritmo inicializado con la fase reverberada genera una mejora. Finalmente, la dereverberación utilizando directamente la fase reverberada produce una mejora mucho mayor que los métodos iterativos previamente mencionados. Para las otras dos métricas, SRMR y ESTOI, los métodos iterativos producen mejores resultados que la utilización directa de la fase reverberada, pero las diferencias entre las alternativas son menores.

Tabla 5: Comparación de métodos de reconstrucción de espectrograma complejo para generar audio.

	SDR	SRMR	ESTOI
Audio reverberado (referencia)	- 3.11	1.73	0.29
Δ Dereverberación con fase reverberada	+4.27	+4.53	+0.31
Δ Dereverberación Griffin-Lim iniciado con fase reverberada	+1.38	+5.13	+0.33
Δ Dereverberación Griffin-Lim iniciado con fase aleatoria	-2.92	+5.24	+0.33

Estos resultados revelan una cierta inconsistencia entre las métricas utilizadas. A su vez, al realizarse escuchas de los resultados, no siempre el método con mejores métricas era el que se percibía con una mayor calidad de dereverberación. Incluso, en algunos casos, las diferencias percibidas eran mínimas, mientras que las métricas mostraban grandes diferencias en sus valores. La utilización de la fase reverberada de manera directa resultó ser el método más robusto frente a las métricas. Esto, sumado a su utilización en otros trabajos del estado del arte hizo que esta alternativa sea la escogida a lo largo de este trabajo. Sin embargo, este análisis de fase deja en evidencia la falta de correlación de ciertas métricas como el SDR con la percepción auditiva de los resultados, lo cual se condice con análisis realizados en otros trabajos [78]. Por cuestiones como esta, continuamente se desarrollan nuevas métricas con el objetivo de lograr una mejor correlación con la percepción auditiva.

5.4 DEREVERBERACIÓN DEL HABLA Y MANEJO DE DATOS

Para las evaluaciones se tuvieron en cuenta tres conjuntos de datos de acuerdo al tipo de respuestas al impulso utilizadas para generar la reverberación: reales, generadas y aumentadas. Para medir el desempeño de la tarea de dereverberación, en una primera instancia se evaluaron las métricas sobre los conjuntos reverberados, obteniéndose los resultados de la Tabla 7. Luego, se compararon estas métricas con las obtenidas sobre los audios dereverberados.

Tabla 6: Resultados de las métricas sobre los conjuntos reverberados.

Conjunto	SDR	SRMR	ESTOI
Reales	-3.94	1.22	0.28
Generadas	2.89	2.53	0.46
Aumentadas	8.09	3.19	0.64

Se pueden observar en las Figuras 32, 33 y 34 las diferencias de las métricas SDR, SRMR y ESTOI respectivamente. Nótese que un valor positivo indica que la métrica aumentó al realizarse la dereverberación.

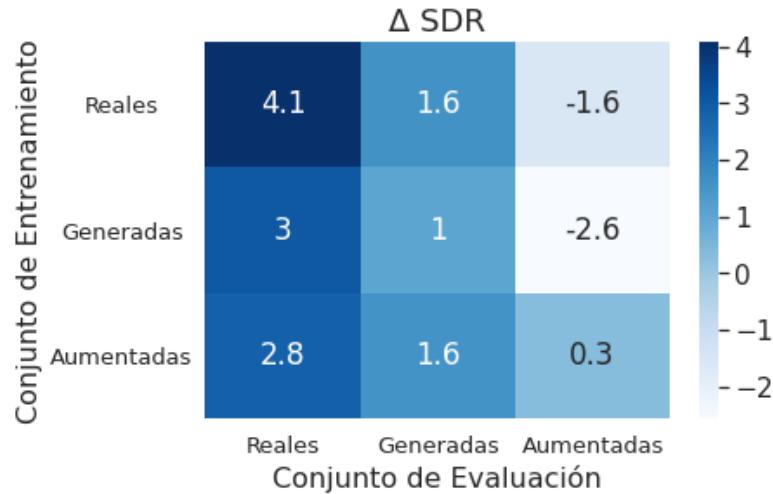


Figura 32: Variaciones de SDR para el primer conjunto de pruebas.

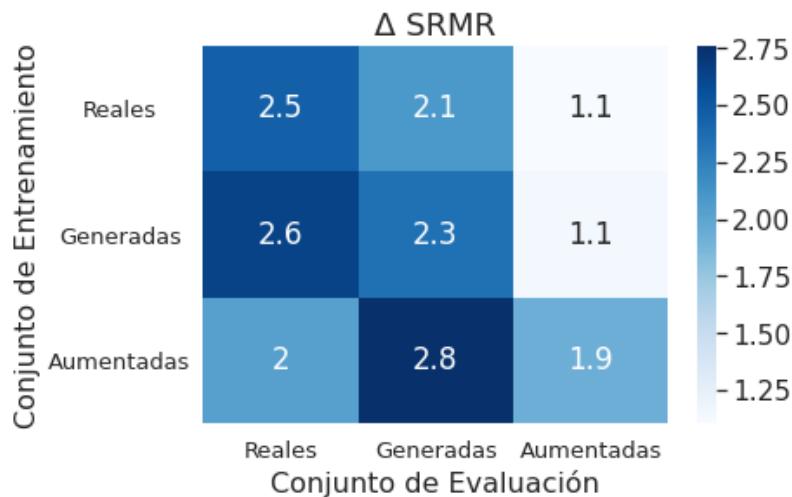


Figura 33: Variaciones de SRMR para el primer conjunto de pruebas.

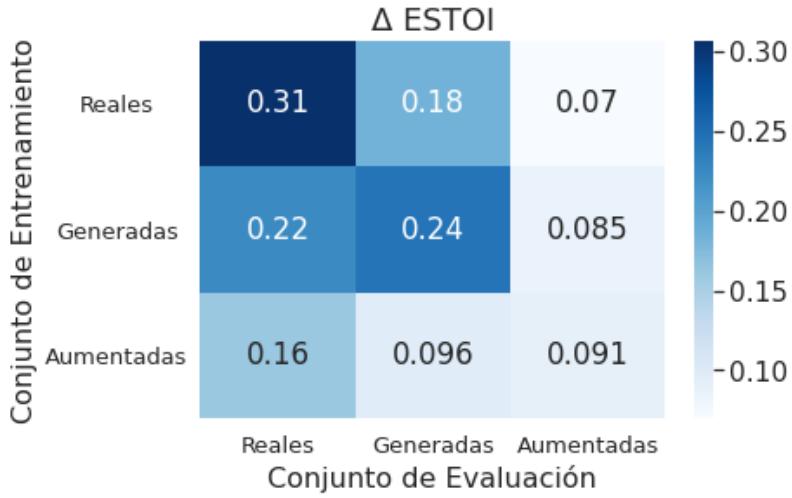


Figura 34: Variaciones de ESTOI para el primer conjunto de pruebas.

Del primer conjunto de pruebas se esperaba obtener los mejores resultados para aquellos casos en los que el conjunto de entrenamiento y el conjunto de evaluación coinciden. Esto ocurrió para la evaluación con la métrica ESTOI. Para las otras métricas, el comportamiento esperado ocurrió en general para los conjuntos formados con respuestas al impulso reales y aumentadas, pero no para las generadas. Particularmente, utilizar respuestas al impulso aumentadas durante el entrenamiento produjo mejores resultados al evaluar sobre respuestas al impulso generadas que usando respuestas al impulso generadas durante el entrenamiento. Esto puede deberse al hecho de que, si bien ambos conjuntos contienen tiempos de reverberación del mismo rango, las respuestas al impulso aumentadas tienen una distribución más uniforme a lo largo de este rango.

Los resultados correspondientes al segundo conjunto de pruebas definido en la Tabla 4 se muestran en las Figuras 35, 36 y 37 para las variaciones de las métricas SDR, SRMR y ESTOI respectivamente.

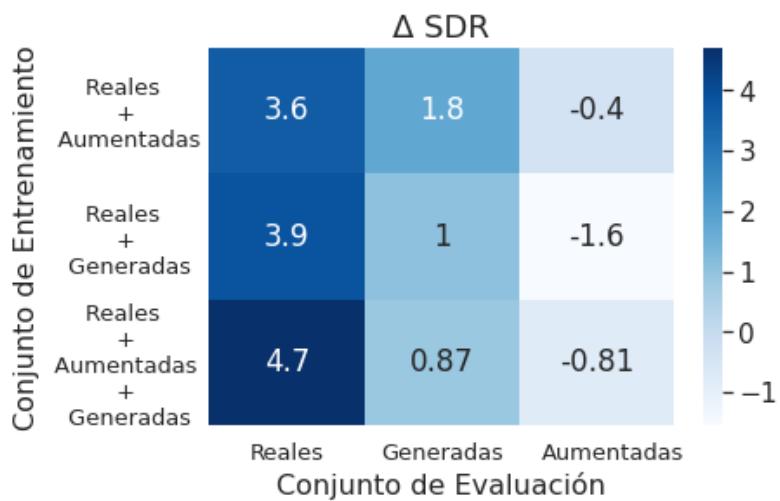


Figura 35: Variaciones de SDR para el segundo conjunto de pruebas.

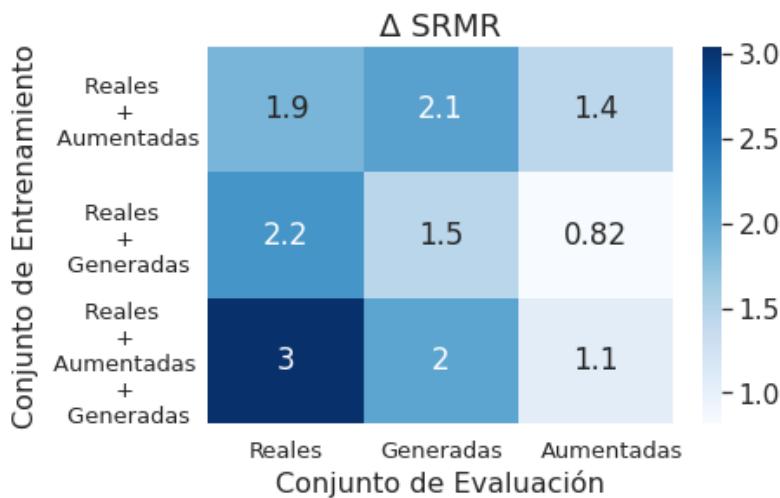


Figura 36: Variaciones de SRMR para el segundo conjunto de pruebas.

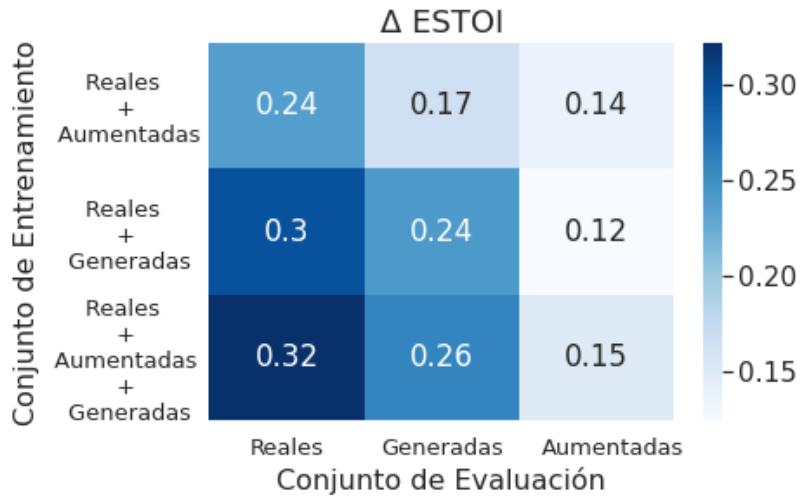


Figura 37: Variaciones de ESTOI para el segundo conjunto de pruebas.

Para el segundo conjunto de pruebas se combinaron tipos de respuestas al impulso en la conformación de los datos de entrenamiento y se volvió a evaluar en los mismos conjuntos de la primera prueba, asegurándose que el número de instancias de entrenamiento se mantenga fijo en todas las pruebas. Se debe tener en consideración que es de mayor importancia para éste trabajo valorar el rendimiento al evaluar sobre respuestas al impulso reales, ya que es el objetivo principal del sistema implementado. En esta prueba, para todas las métricas, los mejores resultados se obtuvieron al combinar los tres tipos de datos en la conformación del conjunto de entrenamiento. Es decir, una mayor diversidad de impulsos presentes a la hora de generar los datos de entrenamiento resulta en una mejora en el rendimiento del sistema. Además, la combinación de respuestas al impulso reales-generadas arrojó mejores resultados que la combinación reales-aumentadas para todas las métricas. Esto puede deberse al hecho de que las respuestas al impulso aumentadas si bien varían la pendiente de caída de la cola reverberante, mantienen el mismo perfilpectral que las respuestas al impulso reales de las que provienen. En trabajos posteriores sería de interés realizar una aumentación que no mantenga el perfil original del tiempo de reverberación, siendo el perfil resultante otra variable a controlar.

5.5 APRENDIZAJE POR CURRÍCULUM

Para evaluar la influencia del ordenamiento de los datos en el proceso de entrenamiento en primer lugar se generó una base de datos de respuestas al impulso asegurando una adecuada

dispersión de los parámetros acústicos de relación directo-reverberado y tiempo de reverberación medio. Para poder conseguir esto, se partió de la base de datos de respuestas al impulso reales C4DM y se aplicó el método de aumentación. Esta vez se generaron tiempos de reverberación medio desde $0,1\text{ s}$ a $3,5\text{ s}$ y relaciones directo-reverberado de -10 dB a 10 dB . En la Figura 38 se puede observar la dispersión de los parámetros acústicos mencionados en el conjunto de respuestas al impulso conformado.

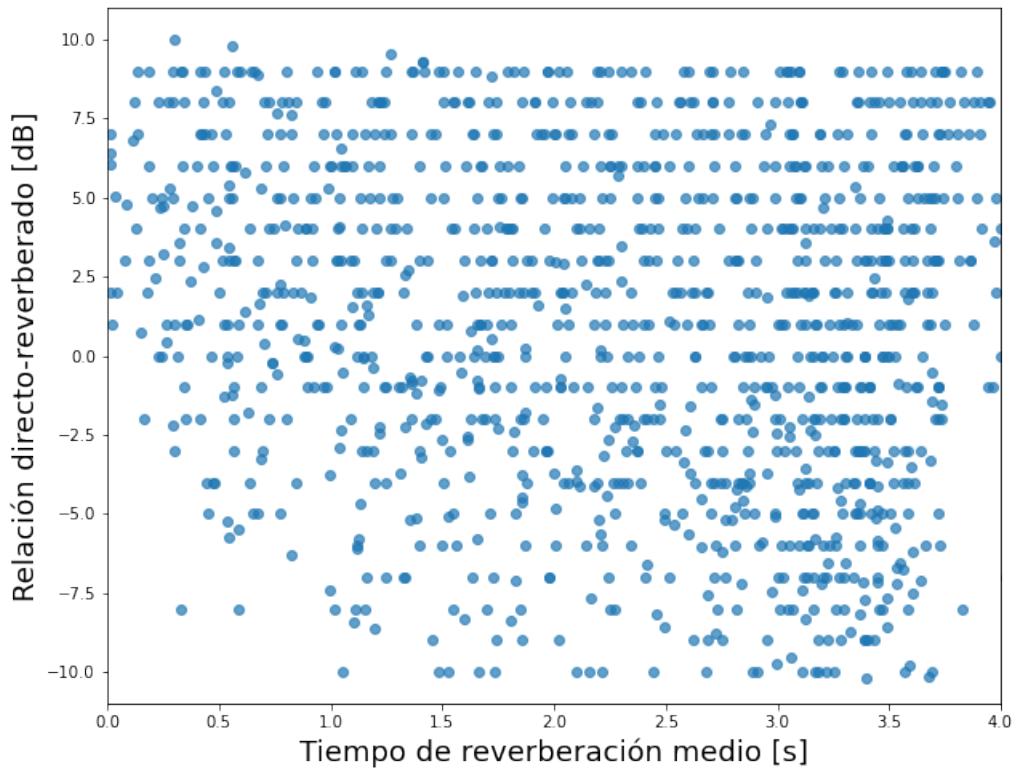


Figura 38: Respuestas al impulso generadas por aumentación.

Partiendo de la información del tiempo de reverberación medio de cada respuesta al impulso, se generaron pares de audios anecoicos-reverberados junto con un registro que indicaba cual tiempo de reverberación medio correspondía con cada audio reverberado. Este registro se utilizó para conformar los esquemas de entrenamiento que fueron evaluados. Entonces, se organizaron los datos de entrenamiento de tres maneras: con tiempos de reverberación crecientes, decrecientes y aleatorios. Se aumentaron los datos de entrenamiento de manera tal de

que, para cada método, la convergencia del error ocurra entrenando durante una sola época. Esto se consiguió generando 460 respuestas al impulso aumentadas y utilizando 360 horas de grabaciones de voz anecoicas. Una vez realizado el entrenamiento, se utilizó el modelo entrenado para hacer predicciones y se calcularon las métricas objetivas sobre los resultados de cada variante. En las Figuras 39, 40, y 41 se muestran los resultados obtenidos para las métricas SDR, SRMR y ESTOI respectivamente.

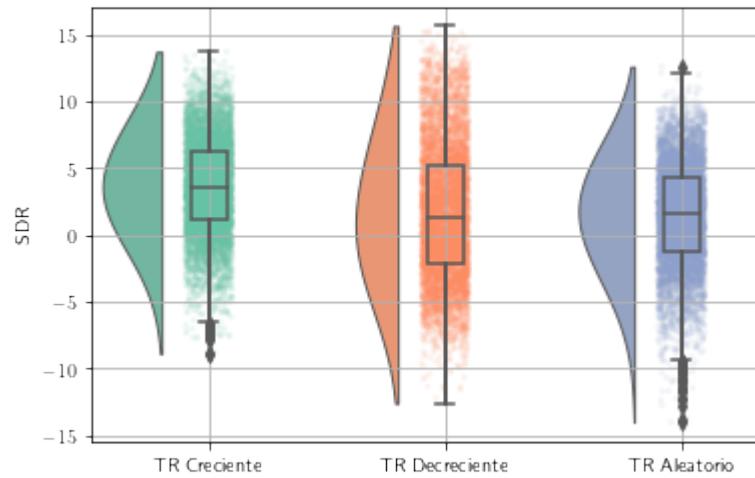


Figura 39: Comparación de SDR entre tipos de ordenamiento de datos durante el entrenamiento.

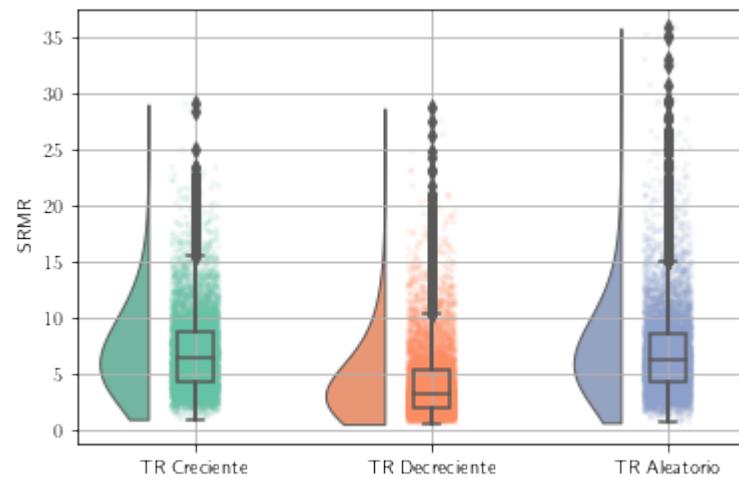


Figura 40: Comparación de SRMR entre tipos de ordenamiento de datos durante el entrenamiento.

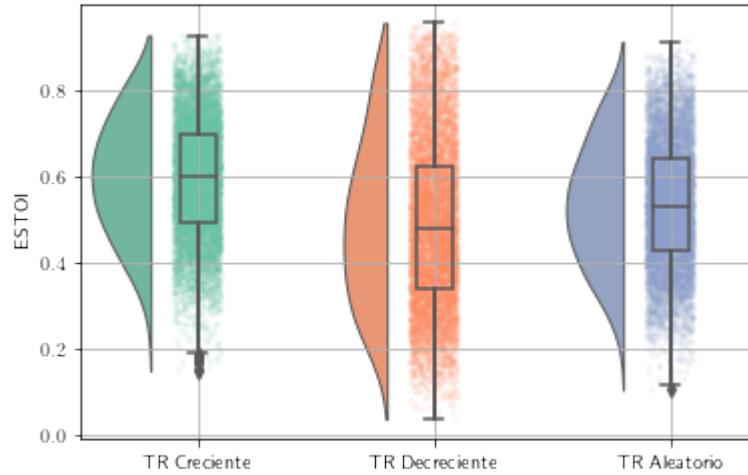


Figura 41: Comparación de ESTOI entre tipos de ordenamiento de datos durante el entrenamiento.

A primera vista se puede percibir que para todas las métricas el resultado obtenido para el entrenamiento con tiempos de reverberación crecientes es el mejor, el resultado obtenido para el entrenamiento con tiempos de reverberación decrecientes es el peor, y el resultado para el entrenamiento con tiempos de reverberación aleatorios esta en un punto medio entre los dos anteriores, en ocasiones más cerca del mejor y en ocasiones más cerca del peor. Para ilustrar mejor este comportamiento, en la Tabla 7 se muestran las medianas de las métricas obtenidas para cada esquema de entrenamiento.

Tabla 7: Medianas correspondientes a cada esquema de entrenamiento.

	\tilde{X}_{SDR}	\tilde{X}_{SRMR}	\tilde{X}_{ESTOI}
TR Creciente	3.54	6.37	0.59
TR Decreciente	1.24	3.25	0.47
TR Aleatorio	1.61	6.24	0.53

Finalmente, con estos resultados se pudo corroborar el impacto del ordenamiento de los datos de entrenamiento en el rendimiento final del sistema. En este caso, el criterio de ordenamiento de los datos se asoció al tiempo de reverberación de cada instancia. Con esto, al entrenar con datos ordenados por tiempo de reverberación de manera creciente, es decir, iniciando con tiempos de reverberación bajos y aumentando progresivamente el tiempo de reverberación se obtuvieron mejores resultados para todas las métricas. También se comprobó que el orde-

namiento inverso produce el peor resultado de los tres, y el orden aleatorio cae en un punto medio entre estos dos casos. Esto es consistente con trabajos previos aplicando aprendizaje por currículum [72], [73]. En una primera etapa del entrenamiento, para tiempos de reverberación bajos, es sencillo para la red neuronal mantener el sonido directo que es predominante en la señal debido a la escasa reverberación (la máscara ideal se asemeja a una máscara con unos en todos sus elementos). Es posible que el aprendizaje por currículum permita al modelo aprender a dereverbar de manera progresiva, aprovechando mejor las instancias de entrenamiento y guiando al proceso de optimización.

CAPÍTULO 6: CONCLUSIONES

En este trabajo se implementó un algoritmo de aprendizaje profundo para la tarea de dereverberación de señales de habla. Se utilizó una estructura de tipo autoencoder con conexiones de salto, la cual se entrenó para estimar máscaras de amplitud que al aplicarse sobre un espectro de magnitud reverberado generen un espectro de magnitud deneverberado. Luego, para obtener la señal temporal a partir de la magnitud del espectrograma deneverberado, se utilizó la fase del espectrograma reverberado y se invirtió mediante la transformada inversa de Fourier y la técnica de suma y solapamiento.

Principalmente, se puso foco en analizar la influencia de ampliar el conjunto de datos de entrenamiento con técnicas de aumentación y síntesis de respuestas al impulso. También se analizó el efecto del ordenamiento de los datos de entrenamiento, y de la técnica utilizada para la inversión del espectrograma deneverberado.

Se pudo comprobar que una mayor diversidad de respuestas al impulso para la generación de los datos de entrenamiento genera mejores resultados, y que tanto las respuestas al impulso generadas como las aumentadas son útiles como método de aumentación de datos de entrenamiento. Por otro lado, se pudo comprobar que aplicar aprendizaje por currículum, ordenando los datos con un TR creciente, permite obtener mejores resultados de deneverberación.

Estos resultados son un aporte importante para el estudio de las tareas de deneverberación de audio, para las cuales las bases de datos de respuestas al impulso existentes son escasas, poco diversas y en ocasiones difíciles de conseguir o muy costosas.

CAPÍTULO 7: LÍNEAS FUTURAS DE INVESTIGACIÓN

Si bien los resultados obtenidos en este trabajo son satisfactorios y demuestran el potencial del uso de redes neuronales profundas y técnicas de aumentación de datos, algunas limitaciones fueron notadas. Por un lado, la información de fase fue omitida, y trabajos recientes sugieren que pueden obtenerse mejoras expandiendo el procesamiento para que se consideren también las componentes complejas de la STFT, es decir, realizar la dereverberación en amplitud y fase. Esto se podría conseguir estimando máscaras complejas o bien estimando máscaras independientes para magnitud y fase. Por otro lado, en este enfoque se procesa de la misma manera la totalidad del espectro. Sabiendo que la reverberación aporta más energía en bajas frecuencias, podría segmentarse el espectro en bandas y procesar cada banda con configuraciones diferentes, o utilizar otras funciones de costo que compensen la energía en alta frecuencia. De manera más general, también podrían modelarse de forma eficiente dependencias temporales presentes en la reverberación introduciendo capas recurrentes.

Otro aspecto a considerar es el referido a las métricas utilizadas tanto para el entrenamiento como para la evaluación de los modelos. Se deben analizar y proponer nuevas métricas que se correlacionen de manera directa con la percepción auditiva de la tarea de dereverberación [79]. De esta forma el entrenamiento se realizará en pos de una mejora en la percepción auditiva de los resultados, y los resultados de las evaluaciones podrán ser utilizados para tomar decisiones que mejoren la calidad del proceso de dereverberación de manera consistente y objetiva.

Respecto al proceso de aumentación, es de interés tener un cierto control sobre el perfil espectral del tiempo de reverberación con el que se generan las respuestas al impulso aumentadas. En este trabajo, las respuestas al impulso aumentadas poseen el mismo perfil espectral de TR que las originales. Una mayor diversidad de respuestas podría conseguirse si se logra controlar también la forma general del perfil de tiempo de reverberación.

Perfeccionar las técnicas de aumentación de respuestas al impulso y recolectar nuevas podría llevar en un futuro a la creación de un corpus de datos de dominio libre destinado específicamente al desarrollo de sistemas de dereverberación del habla, estandarizando los métodos de evaluación de los estudios que se realicen en el área. Generar esta base de datos de acceso libre puede asegurar que exista una correcta comparación entre las soluciones propuestas por

diferentes trabajos, de forma tal que cada resultado logre contribuir de manera objetiva a la mejora general de las técnicas de reverberación.

Bibliografía

- [1] L. Deng, G. Hinton y B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.
- [2] H. Chung, T. Kim, E. Plourde y B. Champagne, "Noise-adaptive deep neural network for single-channel speech enhancement," *Proc. Int. Workshop on Machine Learning for Signal Process. (MLSP)*, 2018.
- [3] J. Pearson, J. Flanagan, B. Devries y L. Jin, "Robust distant-talking speech recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996.
- [4] P. J. Castellano, S. Sradharan y D. Cole, "Speaker recognition in reverberant enclosures," *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996.
- [5] J. H. DiBiase, H. F. Silverman y M. S. Brandstein, *Microphone Arrays: Signal Processing Techniques and Applications*, 1.^a ed. Springer, Berlin, 2001.
- [6] M. Miyoshi e Y. Kaneda, "Inverse Filtering of Room Acoustics," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, n.^o 2, págs. 145-152, 1988.
- [7] S. T. Neely y J. B. Allen, "Invertibility of a room impulse response," *Journal of the Acoustical Society of America*, vol. 66, n.^o 1, págs. 165-169, 1997.
- [8] L. R. Rabiner y R. W. Schafer, *Theory and Applications of Digital Speech Processing*, 1.^a ed. Pearson, California, 2010.
- [9] B. Yegnanarayana y P. Satyanarayana, "Enhancement of Reverberant Speech Using LP Residual Signal," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.
- [10] S. Gannot y M. Moonen, "Subspace Methods for Multimicrophone Speech Dereverberation," *EURASIP journal on advances in signal processing*, 2003.

- [11] N. Roman y D. Wang, "Pitch-based monaural segregation of reverberant speech," *Journal of Acoustical Society of America*, vol. 120, n.^o 1, págs. 458-459, 2006.
- [12] M. Avendano y H. Hermansky, "Study on the dereverberation of speech based on temporal envelope filtering," *Proceeding of Fourth International Conference on Spoken Language Processing*, 1996.
- [13] K. Lebart y J. M. Boucher, "A New Method Based on Spectral Subtraction for Speech Dereverberation," *Acta Acustica united with Acustica*, vol. 87, n.^o 3, págs. 359-366, 2001.
- [14] K. Lebart, J. Boucher y P. Denbigh, "A new method based on spectral subtraction for speech dereverberation," *Acta Acustica united with Acustica*, vol. 87, págs. 359-366, 2001.
- [15] M. Wu y D. Wang, "A two-stage algorithm for one-microphone reverberant speech enhancement," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 14, n.^o 3, págs. 774-784, 2006.
- [16] D. Wang y P. Divenyi, *Speech Separation by Humans and Machines*, 1.^a ed. Kluwer Academic Publishers, Norwell, 2005.
- [17] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*, 1.^a ed. MIT Press, Massachussets, 1994.
- [18] N. Roman y J. Woodruff, "Intelligibility of reverberant noisy speech with ideal binary masking," *Journal of Acoustical Society of America*, vol. 130, n.^o 4, págs. 2153-2161, 2011.
- [19] N. Roman y J. Woodruff, "Speech intelligibility in reverberation with ideal binary masking: Effects of early reflections and signal-to-noise ratio threshold," *Journal of Acoustical Society of America*, vol. 133, n.^o 3, págs. 1707-1717, 2013.
- [20] O. Hazrati, J. Lee y P. C. Loizou, "Blind binary masking for reverberation suppression in cochlear implants," *Journal of Acoustical Society of America*, vol. 133, n.^o 3, págs. 1607-1614, 2013.
- [21] Z. Jin y D. L. Wang, "A supervised learning approach to monaural segregation of reverberant speech," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

- [22] D. S. Williamson y D. Wang, "Time-frequency masking in the complex domain for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, n.^o 7, págs. 1492-1501, 2017.
- [23] I. Kodrasi y H. Bourlard, "Single-channel late reverberation power spectral density estimation using denoising autoencoders," *Proc. Interspeech*, 2018.
- [24] C. Li, T. Wang, S. Xu y B. Xu, "Single-channel speech dereverberation via generative adversarial training," *Proc. Interspeech*, 2018.
- [25] K. Han, Y. Wang y D. Wang, "Learning spectral mapping for speech dereverberation," *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2014.
- [26] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2.^a ed. O'reilly media, California, 2019.
- [27] O. Ernst, S. E. Chazan, S. Gannot y J. Goldberger, "Speech Dereverberation Using Fully Convolutional Networks," *22nd International Conference on Digital Signal Processing*, 2017.
- [28] H. Chung, V. S. Tomar y B. Champagne, "Deep convolutional neural network-based inverse filtering approach for speech de-reverberation," *IEE International Workshop on Machine Learning for Signal Processing*, 2020.
- [29] F. Weninger, S. Watanabe, Y. Tachioka y B. Schuller, "Deep recurrent de-noising autoencoder and blind de-reverberation for reverberated speech recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.
- [30] J. F. Santos y T. H. Falk, "Speech Dereverberation with Context-aware Recurrent Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, n.^o 7, págs. 1232-1242, 2018.
- [31] D. W. Griffin y J. S. Lim, "Signal Estimation for Modified Short-Time Fourier Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, n.^o 2, págs. 236-243, 1984.
- [32] D. S. Wang, Y. X. Zou y W. Shi, "A Deep Convolutional Encoder-Decoder Model for Robust Speech Dereverberation," *22nd International Conference on Digital Signal Processing*, 2017.

- [33] V. Kothapally, W. Xia, S. Ghorbani, J. H. Hansen, W. Xue y J. Huang, "SkipConvNet: Skip Convolutional Neural Network for Speech Dereverberation using Optimally Smoothed Spectral Mapping," *Proc. Interspeech*, 2020.
- [34] J. Su, Z. Jin y A. Finkelstein, "HiFi-GAN: High-Fidelity Denoising and Dereverberation Based on Speech Deep Features in Adversarial Networks," *Proc. Interspeech*, 2020.
- [35] L. L. Beranek, *Acústica*, 2.^a ed. MIT, Massachusetts, 1969.
- [36] A. V. Oppenheim, R. W. Shafer y J. R. Buck, *Discrete-Time Signal Processing*, 2.^a ed. Prentice-Hall, New Jersey, 1989.
- [37] E. O. Brigham, *The Fast Fourier Transform And Its Applications*, 1.^a ed. Pearson, New Jersey, 1988.
- [38] Documentación oficial de Mathworks del módulo "DSP" de Matlab, <https://la.mathworks.com/help/dsp/ref/dsp.stft.html>, Extraido el 15 de Octubre de 2021.
- [39] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. 1.^a ed. Springer, Germany, 2021.
- [40] U. Zölzer, *Digital Audio Signal Processing*, 2.^a ed. Willey, Hamburg, 2017.
- [41] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept sine technique," *108th AES Convention*, 2000.
- [42] M. R. Schroeder, "Digital Simulation of Sound Transmission in Reverberant Spaces," *Journal of the Acoustical Society of America*, vol. 47, págs. 424-431, 1970.
- [43] J. B. Allen y D. A. Berkeley, "Image Method for Efficient Simulating Small RoomAcoustics," *Journal of the Acoustical Society of America*, vol. 65, págs. 943-950, 1979.
- [44] T. H. Falk, C. Zheng y W.-Y. Chan, "A Non-intrusive Quality Measure of Dereverberated Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, n.^o 7, págs. 1766-1774, 2010.
- [45] J. Jensen y C. H. Taal, "An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, n.^o 11, págs. 2009-2022, 2016.

- [46] E. Vincent, R. Gribonval y C. Févotte, "Performance Measurement in Blind Audio Source Separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, n.º 4, págs. 1462-1469, 2006.
- [47] C. Liu, L. Wang y J. Dang, "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol. 5, págs. 115-133, 1943.
- [48] I. Basheer y M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of Microbiological Methods*, vol. 43, n.º 1, págs. 3-31, 2000.
- [49] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, n.º 2, págs. 251-257, 1991.
- [50] F. Chollet, *Deep Learning with Python*, 2.^a ed. Manning Publications, New York, 2021.
- [51] Y. Bengio, J. Louradour, R. Collobert y J. Weston, "Curriculum Learning," *Proceedings of the 26th International Conference on Machine Learning*, 2009.
- [52] N. Srivastava y G. Hinton, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, págs. 1929-1958, 2014.
- [53] I. Goodfellow, Y. Bengio y A. Courville, *DEEP LEARNING*, 1.^a ed. The MIT Press, Massachusetts, 2016.
- [54] S. Ioffe y C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *International Conference on Machine Learning*, vol. 37, págs. 448-456, 2015.
- [55] M. Eickenberg, A. Gramfort, G. Varoquaux y B. Thirion, "Seeing it all: Convolutional network layers map the function of the human visual system," *NeuroImage*, vol. 152, págs. 184-194, 2017.
- [56] J. Patterson y A. Gibson, *Deep Learning: A Practitioner's Approach*, 1.^a ed. O'Reilly, Beijing, 2008.
- [57] D. L. Wang y J. S. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, n.º 4, págs. 679-681, 1982.

- [58] Y. Ephraim y D. Malah, "Speech enhancement using a minimummean-square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, n.^o 6, págs. 1109-1121, 1984.
- [59] R. Stewart y M. B. Sandler, "Database of omnidirectional and B-format impulse responses," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process. (ICASSP)*, 2010.
- [60] R. Scheibler, E. Bezzam e I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," *Proc. IEEE ICASSP*, 2018.
- [61] J. B. Allen y D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, n.^o 4, págs. 943-950, 1979.
- [62] N. J. Bryan, "Impulse response data augmentation and deep neural networks for blind room acoustic parameter estimation," *Adobe Research*, 2019.
- [63] A. Lundeby, T. E. Vigran, H. Bietz y M. Vorlander, "Uncertainties of Measurements in Room Acoustics," *Acta Acustica united with Acustica*, vol. 81, n.^o 4, págs. 344-355, 1995.
- [64] M. G. Blevins, A. T. Buck, Z. Peng y L. M. Wang, "Quantifying the just noticeable difference of reverberation time with band-limited noise centered around 1000 Hz using a transformed up-down adaptive method," *International Symposium on Room Acoustics*, 2013.
- [65] V. Panayotov, G. Chen, D. Povey y S. Khudanpur, "Libris-peech: an asr corpus based on public domain audio books," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.
- [66] P. Vincent, H. Larochelle, Y. Bengio y P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," *Proceedings of the 25th international conference on Machine learning*, 2008.
- [67] Y. Wang, A. Narayanan y D. L. Wang, "On Training Targets for Supervised Speech Separation," *IEEE/ACM Trans Audio Speech Lang Process*, vol. 22, n.^o 12, págs. 1849-1858, 2014.
- [68] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang y W. Shi, "Checkerboard artifact free sub-pixel convolution," *ArXiv - Computer Science*, 2017.

- [69] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert y Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [70] C. Dong, C. C. Loy, K. He y X. Tang, "Image super-resolution using deep convolutional networks," *International Conference on Intelligent Computing*, 2014.
- [71] D. P. Kingma y J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, 2014.
- [72] K. J. Geras y C. Sutton, "Scheduled denoising autoencoders," *International Conference on Learning Representations*, 2015.
- [73] S. Zheng, G. Liu, H. Suo e Y. Lei, "Autoencoder-based Semi-Supervised Curriculum Learning For Out-of-domain Speaker Verification," *Proc. Interspeech*, 2019.
- [74] Martin Bernardo Meza, *Repositorio del trabajo "Dereverberación del habla a partir de algoritmos de aprendizaje profundo"*, <https://github.com/martinBmeza/deep-dereverb>, Extraido el 6 de Octubre de 2021.
- [75] D. O'Shaughnessy, *Speech Communications: Human and Machine*, 2.^a ed. Wiley-IEEE Press, 1987.
- [76] D. Wang y J. S. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1982.
- [77] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha y K. Lee, "Phase-aware speech enhancement with deep complex U-net," *International Conference on Learning Representations*, 2019.
- [78] J. L. Roux, S. Wisdom, H. Erdogan y J. R. Hershey, "SDR - half-baked or well done?" *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- [79] P. Manocha, Z. Jin, R. Zhang y A. Finkelstein, "CDPAM: Contrastive learning for perceptual audio similarity," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

ANEXO A: AUMENTACIÓN DE TIEMPO DE REVERBERACIÓN

El proceso de aumento de respuestas al impulso puede realizarse tomando como parámetros del proceso ciertos descriptores acústicos como el tiempo de reverberación TR_{60} . Esto es, partiendo de una respuesta al impulso con un TR_{60} determinado, se busca generar nuevas respuestas al impulso cuyo TR_{60} pueda ser controlado paramétricamente para ocupar de manera homogénea y balanceada un rango de valores de interés. El TR_{60} se relaciona directamente con la forma de la envolvente de caída de nivel exponencial presente en la parte tardía de las respuestas al impulso. El proceso de aumento equivale a modificar esta pendiente de caída, multiplicando la señal original por una nueva envolvente que produzca el efecto deseado en la envolvente resultante. Los pasos a seguir para realizar este proceso son:

- Acondicionamiento de la respuesta al impulso de entrada.
- Filtrado por bandas de octava o bandas de tercio de octava.
- Estimación de piso de ruido.
- Estimación de envolvente de caída.
- Sintetizar una señal aplicando la envolvente estimada con piso de ruido cero a una señal de ruido Gaussiano.
- Realizar el cross-fade entre la señal sintetizada y la señal original en el punto inicial del piso de ruido.
- Aumentación de la envolvente de caída multiplicando la señal por la correspondiente envolvente exponencial creciente/decreciente.
- Suma de las sub-bandas para obtener la señal resultante en su espectro completo.
- Integración de la parte tardía aumentada con la parte directa de la respuesta al impulso inicial.

A continuación se explica cada paso del algoritmo en mayor profundidad. En primer lugar, se define una determinada frecuencia de muestreo y profundidad de bits para trabajar con la señal

de entrada, en este caso una respuesta al impulso real. Una vez asegurada la homogeneidad de estas características, la señal se normaliza para trabajar en un rango de amplitud acotado en el intervalo $[-1, 1]$ y luego se separa la parte directa de la parte tardía de la respuesta al impulso. Esto último se realiza utilizando una ventana de tolerancia de $t_0 = 2,5\text{ms}$. Para los pasos siguientes se trabaja únicamente modificando la parte tardía, y la parte directa se almacena para ser utilizada en el paso final a la hora de reconstruir la respuesta completa. En la Figura 42 se muestra la respuesta al impulso desde la que se parte, distinguiendo la descomposición temporal de la misma y luego la parte tardía de la respuesta al impulso aislada con la que se va a trabajar durante el proceso.

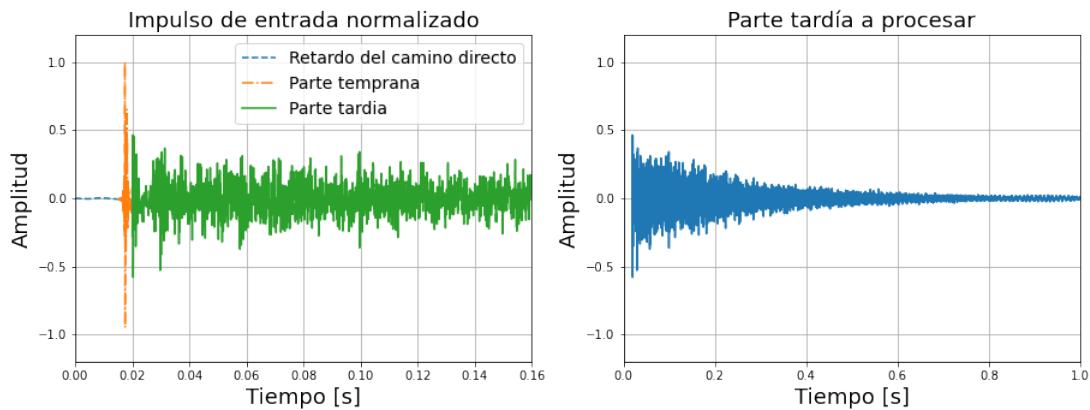


Figura 42: Descomposición temporal de la respuesta al impulso a procesar durante la aumentación.

Luego, el siguiente paso consiste en descomponer la señal en bandas de octava o tercios de octava. En esta demostración se trabaja con bandas de octava desde 125 Hz hasta 4000 Hz teniendo en cuenta que se utiliza una frecuencia de muestreo de 16000 Hz . Es necesario trabajar en sub-bandas frecuenciales para contemplar la dependencia del tiempo de reverberación con la frecuencia, y mantener esa característica en las señales a generar en este proceso. Para conseguir esta descomposición en sub-bandas frecuenciales se crea un banco de filtros. El mismo se compone de filtros Butterworth que van siendo creados a partir de las frecuencias centrales que se quiera obtener en cada banda. El proceso consiste en crear filtros pasa-banda para generar una banda de paso alta y una banda de paso baja. Luego, se toma la banda de paso alta y se la vuelve a dividir en banda de paso alta y baja aplicando un nuevo par de filtros pasa-banda. Esto se repite hasta completar todas las frecuencias de corte necesarias. De esta forma se ob-

tiene el prototipo IIR de cada filtro que compone el banco de filtros. Una vez obtenido esto, se crean respuestas de tipo FIR para cada filtro a través de pasar un impulso ideal por cada filtro. En la Figura 43 se puede observar la respuesta en frecuencia del banco de filtros. Un detalle importante a considerar es que la suma del efecto de todos los filtros es unitaria para todo el rango frecuencial analizado, siendo esta una característica necesaria para poder descomponer la señal en bandas y luego re-componer la señal sumando las bandas sin generar ningún tipo de distorsión en el proceso. Para lograr esto, los coeficientes de los filtros deben ser elevados al cuadrado (lo que equivale a poner dos filtros en cascada) para lograr que en las intersecciones entre los filtros la suma de amplitud sea unitaria (en la frecuencia de corte se genera una caída de 6 dB en lugar de los 3 dB que presentaría un filtro Butterworth simple).

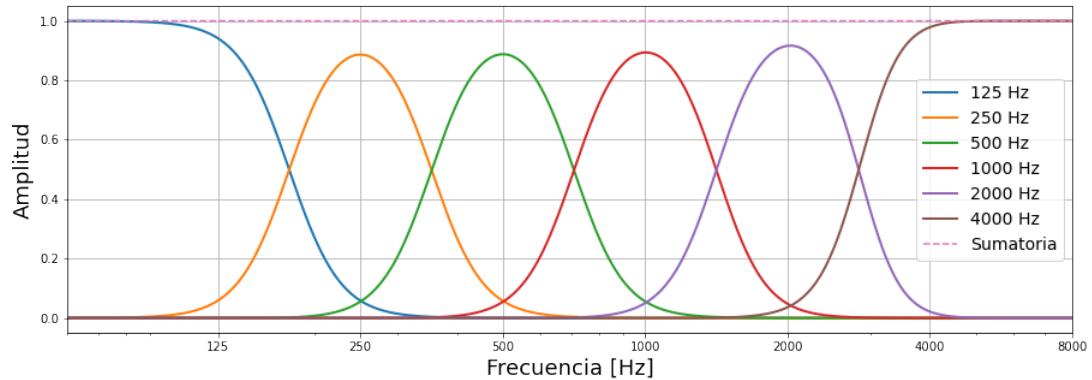


Figura 43: Banco de filtros Butterworth.

A partir de aplicar este banco de filtros, la señal de entrada se descompone en 6 sub-bandas como se muestra en la Figura 44. En este gráfico se puede apreciar como la pendiente de caída varía según la banda de frecuencia que se observa. Todos los procesos subsiguientes se aplican de manera independiente para cada banda, y luego al final las bandas se suman para volver a tener una señal correspondiente al espectro completo original. De ahora en adelante, por una cuestión de simplicidad se muestran los gráficos pertenecientes a la banda de 1000 Hz de manera ilustrativa.

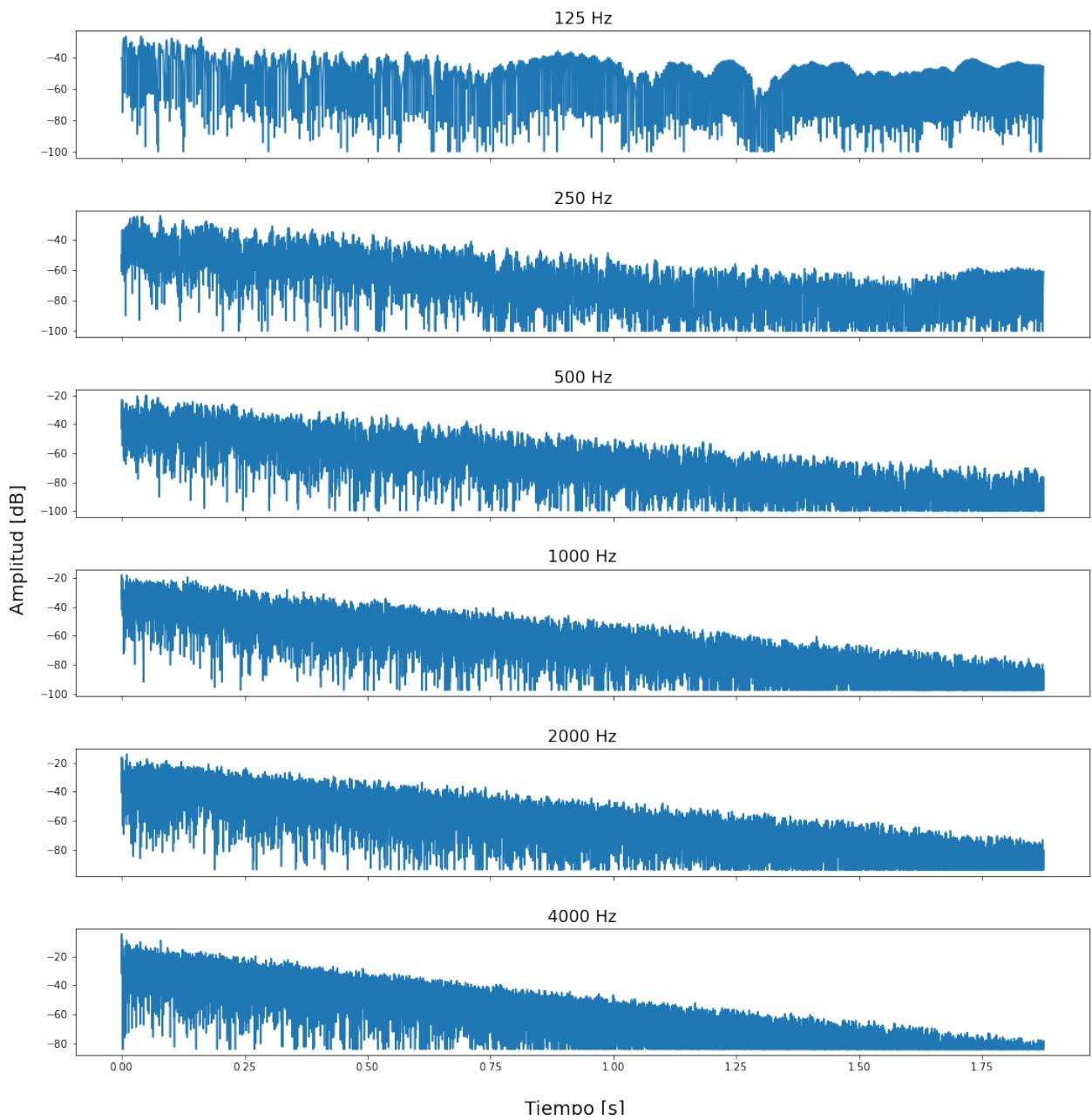


Figura 44: Sub-bandas obtenidas luego de aplicar el banco de filtros.

El paso siguiente consiste en determinar el piso de ruido de la señal. Detectar el piso de ruido permite asegurar que el método de aumentación no amplifique ruido cuando se busca obtener un tiempo de reverberación mayor al inicial propio de la respuesta al impulso de entrada. Para determinar el punto donde predomina el ruido en la respuesta al impulso se utiliza el método iterativo de Lundeby [63]. El mismo consta de los siguientes pasos:

1. La respuesta al impulso al cuadrado es promediada en intervalos de tiempo locales de entre 10 ms y 50 ms para obtener una curva 'suavizada', es decir, disminuir las variaciones instantáneas sin perder las pendientes cortas.
2. Se hace una primera estimación del piso de ruido. Para hacerlo se toma el segmento correspondiente al último 10% de la respuesta al impulso.
3. La pendiente de caída se estima aplicando una regresión lineal sobre el intervalo de tiempo que contiene la respuesta entre el pico de 0 dB y el primer intervalo 5 – 10 dB por encima del ruido de fondo.
4. Se determina un punto de cruce provisorio en la intersección entre la pendiente de caída estimada y el nivel de piso de ruido.
5. Se obtiene un nuevo intervalo de tiempo de acuerdo a la pendiente calculada, de manera que haya entre 3 y 10 intervalos por cada 10 dB de caída.
6. Se vuelve a promediar localmente el impulso al cuadrado de acuerdo al nuevo intervalo temporal calculado previamente .
7. Se estima el ruido de fondo nuevamente. El segmento a evaluar debe corresponder a 5 – 10 dB luego del punto de cruce (siguiendo la curva estimada previamente), o bien, un mínimo del 10% de la señal total (en el caso de tener que optar por el 10% de nuevo, el resultado sería el mismo que antes, y el punto encontrado previamente sería el definitivo).
8. Se estima la pendiente de caída para un rango dinámico de entre 20 dB y 10 dB, empezando desde un punto 5 – 10 dB por encima del nivel de ruido.
9. Se encuentra un nuevo punto de cruce.
10. Los pasos 7-9 se repiten hasta que el valor del piso de ruido converja, tolerando un máximo de 6 iteraciones.

El paso siguiente es estimar la pendiente paramétrica que mejor se aproxime a la pendiente de caída real. La estimación se basa en el modelo de la ecuación 14. Por lo tanto, los parámetros que se busca estimar son la amplitud inicial, la tasa de caída y el nivel de piso de ruido. La estimación se realiza aplicando un algoritmo de ajuste no lineal por cuadrados mínimos. El resultado de la estimación para la banda de 1000 Hz se muestra en la Figura 45.

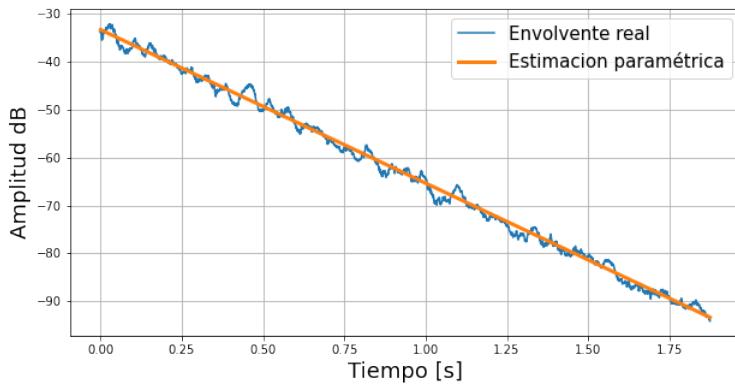


Figura 45: Estimación paramétrica de la pendiente de caída.

Con estos parámetros estimados se genera una nueva envolvente de caída pero llevando el nivel de piso de ruido a cero, y se aplica esta envolvente sobre una señal de ruido Gaussiano de media cero y desvío estándar unitario. Con esta señal sintética y la señal original se hace una transición cruzada en el punto estimado del piso de ruido. De esta manera se elimina el ruido de la señal original, reemplazándolo por la caída exponencial determinada por la envolvente paramétrica. En la Figura 46 se puede observar como se extiende la respuesta luego del punto de inicio del piso de ruido de la señal original.

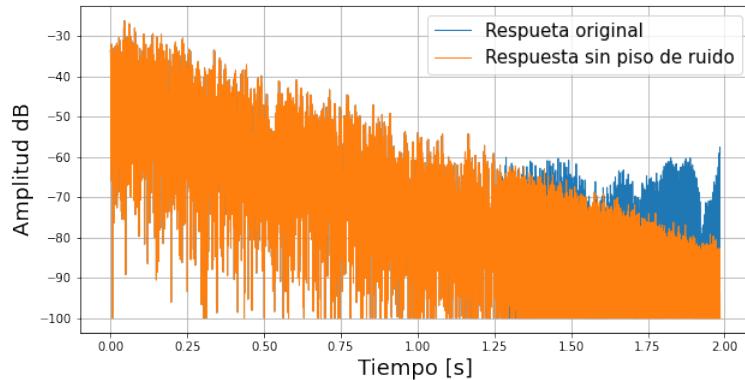


Figura 46: Respuesta original y extendida sin piso de ruido.

Finalmente, teniendo las bandas extendidas y habiéndose eliminado el piso de ruido, se prosigue con el proceso de aumentación del tiempo de reverberación para cada sub-banda, multiplicando cada respuesta por la correspondiente envolvente exponencial creciente o decreciente según corresponda, para obtener la envolvente de caída necesaria para generar el tiempo de reverberación deseado, como se indica en la ecuación 16. Un ejemplo del resultado de la aumentación sobre una sub-banda de frecuencia se muestra en la Figura 47, en donde el tiempo de reverberación objetivo es menor que el tiempo de reverberación de la respuesta original, por lo cual la pendiente aumentada resulta mas atenuada que la pendiente original.

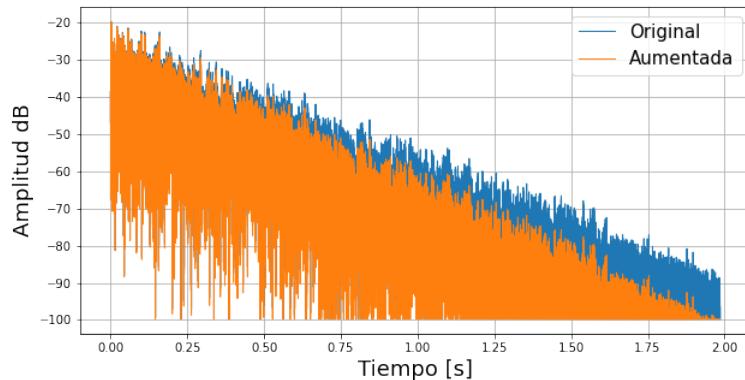


Figura 47: Aumentación del tiempo de reverberación alterando la envolvente de caída original.

Una vez realizada la alteración de la envolvente de caída para todas las bandas frecuenciales, estas se suman para conformar nuevamente la parte tardía de la respuesta al impulso de espectro completo. El paso final consiste en concatenar los segmentos de la respuesta al impul-

so original que no fueron alterados, es decir, el delay del camino directo y la parte directa de la respuesta. Con esto, el proceso de aumentación termina, y se obtiene como resultado una nueva respuesta al impulso con el tiempo de reverberación deseado. En la Figura 48 se muestra la respuesta al impulso original comparada con la nueva respuesta generada a partir del proceso anteriormente descrito.

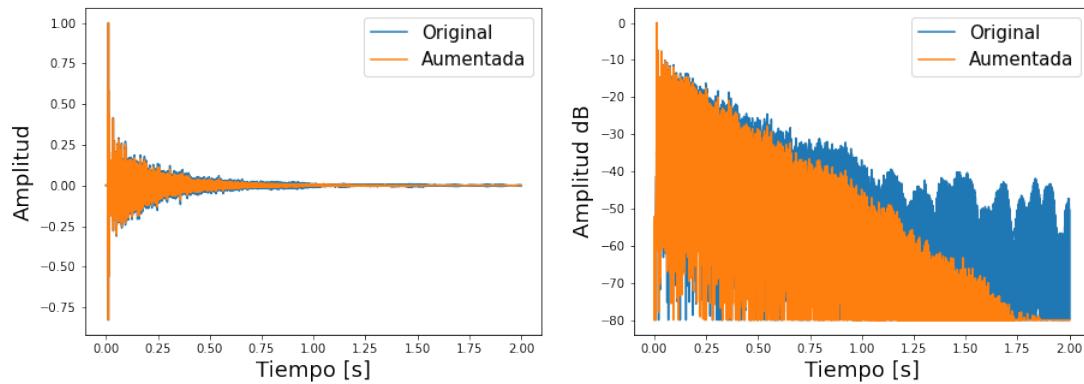


Figura 48: Resultado del proceso de aumento del tiempo de reverberación de una respuesta al impulso.