TECNIACÚSTICA
CUENCA 2023

MODELADO DE RESPUESTAS AL IMPULSO DE SALAS MEDIANTE REDES NEURONALES PROFUNDAS

Gonzalo Atienza Selva [1], Fran Pastor Naranjo [1], Valery Naranjo Ornedo [1], Gema Piñero Sipán [2]*
[1] I3B, Universitat Politècnica de València, España
[2] ITEAM, Universitat Politècnica de València, España

RESUMEN

En este trabajo se describe un sistema para inferir respuestas al impulso acústicas (*room impulse responses*, RIR) entre dos ubicaciones específicas dentro de la misma sala. Dicho sistema está basado en redes neuronales profundas en las que la entrada es una RIR medida en la sala. La novedad de este modelo es que la entrada es directamente la respuesta al impulso en el dominio del tiempo. En general, los métodos para el modelado de RIRs que utilizan redes neuronales se han basado en la imagen obtenida por la *short-time Fourier transform* (STFT), o bien en la imagen espacial obtenida mediante la propagación del campo sonoro (*soundfield*). Ambas presentan varias limitaciones principalmente debido a la dificultad en reconstruir la fase. En este trabajo se proponen e implementan diferentes arquitecturas de redes neuronales profundas y se compara su rendimiento. Entre otras, el estudio de *autoencoders* basados en redes neuronales recurrentes, las cuales son adecuadas para procesar secuencias de datos, como el audio, ya que pueden capturar la dependencia temporal de los datos.

ABSTRACT

This paper describes a system for inferring room impulse responses (RIR) between two specific locations within the same room. Such a system is based on deep neural networks where the input is a measured RIR in the room. The novelty of this model is that the input is directly the impulse response in the time domain. In general, the modelling of RIRs using neural networks has been based on the image obtained by the Short Time Fourier Transform (STFT) or on the spatial image obtained by the sound field propagation. Both have several limitations, mainly due to the difficulty in reconstructing the phase. In this work, different deep neural network architectures are proposed and implemented, and their performance is compared. Among others, autoencoders based on recurrent neural networks are studied, which are suitable for processing sequences of data, such as audio, since they can capture the temporal dependence of the data.

Palabras Clave— Estimación de *room impulse response* (RIR), redes neuronales recurrentes (RNN), sistemas multicanal.

1. INTRODUCCIÓN

El mundo de las comunicaciones audiovisuales está experimentando una serie de cambios significativos, entre los que se distinguen dos campos: uno el visual, el cual se está llevando gran parte de la atención de los investigadores, y el campo auditivo, que no recibe esa misma dedicación por parte de los mismos, a pesar de ser igual de importante que la parte visual para la experiencia de los usuarios que gozan de estos servicios. Es por esto, que nace la necesidad de investigar y desarrollar aplicaciones de audio capaces de generar una experiencia inmersiva e incorporarlos, por ejemplo, en videojuegos o realidad virtual. Para ello se proponen sistemas de reproducción de sonido inteligentes mediante el uso de múltiples altavoces, conocidos como sistemas multicanal.

Mientras las aplicaciones de procesamiento de audio se han desarrollado teniendo en cuenta plataformas de alto rendimiento computacional para sistemas multicanal, el paradigma del *Internet of Things* (IoT) demanda el uso de aplicaciones más flexibles y eficientes energéticamente. Por ello, a partir de esta demanda de sistemas flexibles y eficientes surge el propósito de implementar Zonas Personales de Sonido mediante el uso de técnicas de *machine learning*.

Las Zonas Personales de Sonido [1] consisten en crear zonas de sonido personalizado en un espacio compartido, como por ejemplo una habitación. El propósito principal es brindar a cada individuo una experiencia auditiva única en un mismo espacio, minimizando las interferencias. Esto permite que cada persona pueda escuchar diferentes sonidos de manera independiente sin afectar a los demás, dándole la información completa al oyente sobre la fuente de audio y su entorno acústico. Esto es gracias al avance del procesamiento digital de señales que se ha conseguido que el entorno acústico pueda ser descrito mediante la denominada respuesta al impulso de la sala (RIR, *Room Impulse Response*), que puede ser modelada por un filtro de respuesta finita [2]. Para que este sistema funcione se requiere una RIR específica entre el altavoz y el punto de la sala donde esté el oyente. Sin embargo, si queremos que el oyente pueda tener libertad para desplazarse por la sala y seguir escuchando el sonido, será necesario introducir una nueva RIR que nuevamente represente la respuesta entre el altavoz y la nueva posición del oyente. Estas RIRs hay que medirlas, y si aspiramos a que estos sistemas puedan ser usados en cualquier sala, medir todas las RIRs no es una opción válida. Es por eso que en este trabajo se propone el uso de técnicas de *deep learning* (DL) para predecir las nuevas RIRs.

La mayoría de los trabajos previos que intentan predecir una RIR mediante DL se basan en la reconstrucción de la magnitud y fase de la STFT o bien en la reconstrucción del *soundfield* (Ver [3] y las referencias incluidas). El mayor reto es la reconstrucción de la fase,

* ***Autor de contacto***: gonatsel@cam.upv.es

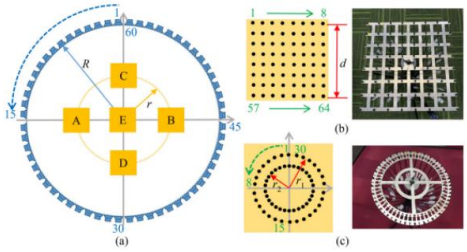

Figura 1. Representación de la disposición de altavoces y micrófonos en la base de datos

por lo que en este trabajo se ha optado por considerar la RIR en el dominio del tiempo.

2.BASE DE DATOS

Para realizar los experimentos se ha empleado una base de datos que contiene un subconjunto de mediciones de RIR realizadas en diversas salas [4]. Los creadores de la base de datos colocaron un conjunto de 60 altavoces dispuestos de manera uniforme en una estructura circular con un radio de R=1.5 m en seis habitaciones distintas. Además, se posicionaron dos arrays de micrófonos: uno en forma de matriz cuadrada de 8x8 con 64 micrófonos y otro en forma de matriz circular de doble capa con 60 micrófonos, en cinco zonas distintas dentro de la disposición de altavoces. Entre estas cinco zonas, los centros de las Zonas A, B, C y D se colocaron a lo largo de un círculo con un radio de r=0.4 m, mientras que la Zona E se ubicó en el centro de la disposición de altavoces, tal y como se muestra en la Figura 1 (a). El primer array de 64 micrófonos constaba de una matriz cuadrada de 8x8 con una dimensión de 28 cm, es decir, la separación de los micrófonos es de 4 cm, y se muestra en la Figura 1 (b). El segundo array de 60 micrófonos presentaba una disposición circular de doble capa, donde 30 micrófonos se distribuyen de manera uniforme a lo largo de los círculos exterior e interior con radios de r_1=12 cm y r_2=10 cm respectivamente, y se muestra en la Figura 1 (c).

La base de datos incluye un total de 260,400 RIRs de las cuales 134,400 provienen del array cuadrado y 126,000 del array circular. Cada RIR tiene una longitud de 43480 muestras, lo que equivale a aproximadamente 1 segundo, con una frecuencia de muestreo de 48 kHz. Para los siguientes experimentos se ha decidido usar las RIRs de la sala pequeña, tanto por sus dimensiones como por tiempo de reverberación T60=0.5seg.

3.MARCO TEÓRICO

En este apartado se explica la base teórica del *deep learning* necesaria para entender el funcionamiento del trabajo realizado.

A. Red Neuronal Simple

Una red neuronal es una arquitectura que trata de imitar el funcionamiento de una neurona del cerebro humano, y está compuesta por *n* entradas, las cuales representan las características de nuestros datos de entrada.

Para cada neurona en particular el funcionamiento es idéntico, se multiplica el valor de todas las conexiones de entrada a la neurona x_i por los pesos w_i y se suman todos los valores obtenidos. Los pesos son los parámetros de la red que representan la intensidad de la conexión entre neuronas [5]. A la suma ponderada se le aplica una función de activación, la cual es importante para introducir no linealidades y permitir que la red capture patrones y relaciones complejas entre los datos. Las funciones de activación más comunes son la sigmoide, que trabaja en un rango entre 0 y 1; la ReLU que es lineal respecto los valores positivos mientras anula los negativos; la tangente hiperbólica que tiene un rango entre -1 y 1; y también están otras más recientes como la ELU o la PReLU.

Una de las redes más simples es el perceptrón multicapa, en la cual las neuronas básicas se organizan en capas. Una red neuronal tiene siempre una capa de entrada, que representa las características, una o más capas intermedias u ocultas, y la capa de salida que representa la predicción. Las capas más comunes de una red neuronal son las capas densas o capas totalmente conectadas[6]. Estas implican que cada neurona en esta capa está interconectada con todas las neuronas de la capa anterior por los pesos, los cuales tienen que ser calculados mediante el entrenamiento para conseguir la salida óptima para todos los datos de entrenamiento.

B. Forward Backward Propagation

El entrenamiento se divide en dos pasos que se hacen iterativamente durante este proceso.

Primero el *forward-propagation*, que toma los datos de entrada y los propaga por la red hasta la salida, es decir, calcula el resultado de cada neurona utilizando los coeficientes que en ese momento están calculados.

Acto seguido el *backward-propagation*, que tras el cálculo del error entre la salida y el *ground-truth* (el verdadero valor de la salida) y mediante un algoritmo de optimización se ajustan los pesos de la red desde la capa de salida hasta la capa de entrada tratando de alcanzar el mínimo error posible a la salida de la red.

C. Red Neuronal Recurrente.

Las redes neuronales recurrentes (*Recurrent Neural Network*, RNN) [7] constituyen una arquitectura de redes neuronales diseñada especialmente para identificar patrones en secuencias de datos. Se aplican ampliamente en tareas como el modelado de lenguaje, la generación de texto y el reconocimiento de voz, entre otras.

Lo que diferencia a las Redes Neuronales Recurrentes de los perceptrones multicapa es la manera en que transmiten información a través de la red. Mientras que los perceptrones multicapa llevan a cabo la propagación sin ciclos, las RNN incorporan ciclos, permitiendo que la información fluya hacia atrás en la secuencia.

Esto les permite tener en cuenta también las entradas anteriores xt-1 y no solo la entrada actual xt, como se puede observar en la Figura 2.

Sin embargo, las RNN tradicionales tienen limitaciones en su capacidad para manejar y aprender relaciones a largo plazo en secuencias debido al problema conocido como desvanecimiento de gradientes (*vanishing gradient problem*) y explosión de gradientes (*exploding gradient problem*). Estos problemas ocurren cuando se intenta entrenar redes neuronales profundas, donde los gradientes

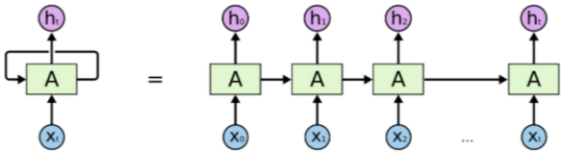

Figura 2. Red neuronal recurrente. Un fragmento de la red neuronal, A, examina una entrada xt y produce un valor ht. Un bucle permite que la información se transmita de un paso de la red al siguiente. Imagen obtenida en [8].

que se propagan a través de las capas se vuelven muy pequeños o grandes, lo que dificulta o incluso impide el entrenamiento efectivo de la red.

Aquí es donde entra en juego la arquitectura de *Long Short-Term Memory* (LSTM) [8], representada en la Figura 3, que fue diseñada para mitigar estos problemas y permitir que las redes neuronales recurrentes capturen relaciones a largo plazo en secuencias. Las LSTM son un tipo de RNN que incorporan unidades de memoria especializadas para retener y olvidar información en diferentes puntos de una secuencia. Tienen tres puertas principales: la puerta de entrada, la puerta de olvido y la puerta de salida, que controlan cómo la información fluye dentro de la unidad de memoria.

La clave de la LSTM es el estado de celda, la línea horizontal que atraviesa la parte superior del diagrama. La LSTM tiene la capacidad de eliminar o agregar información al estado de celda.

Para ello utilizamos una puerta de salida Ot para leer las entradas de la celda, una puerta de entrada It para introducir datos en la celda y una puerta de olvido Ft para restablecer el contenido de la celda. El primer paso en nuestra LSTM es decidir qué información vamos a descartar del estado de la celda y qué nueva información vamos a almacenar en el estado de la celda. Las siguientes ecuaciones ilustran el funcionamiento de la red:

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \qquad f_t \in [0,1] \quad (1)$$

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \qquad i_t \in [0,1] \quad (2)$$

$$\underline{C_t} = tanh(w_C[h_{t-1}, x_t] + b_C) \qquad \underline{C_t} \in [-1,1] \quad (3)$$

Donde w_f, w_i y w_c son los pesos que se entrenan, b_f, b_i y b_c son la constante bias, σ y *tanh* son las funciones de activación sigmoide y tangente hiperbólica. Luego de obtener la celda de memoria candidata $\underline{C_t}$ y la compuerta de olvido Ft, actualizamos el nuevo estado de la celda C_t .

$$C_t = f_t * C_{t-1} + i_t * \underline{C_t} \quad (4)$$

Finalmente, se decide qué partes específicas del estado de la celda serán producidas como salida, siendo w_o y b_o los pesos y el bias de la puerta de salida.

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \qquad o_t \in [0,1] \quad (5)$$

$$h_t = o_t * tanh(C_t) \qquad \underline{C_t} \in [-1,1] \quad (6)$$

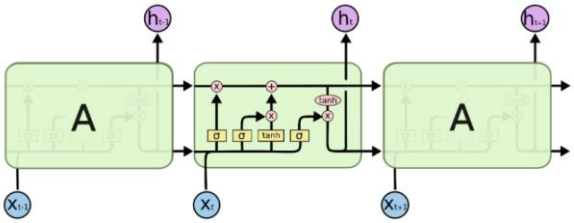

Figura 3. Long Short-Term Memory. Imagen obtenida en [8].

4. METODOLOGÍA

Hasta ahora, en otros trabajos similares para estimar o predecir respuestas al impulso en salas, se ha trabajado en el dominio frecuencial, en concreto utilizando la *Short-Time Fourier Transform* (STFT). Estos trabajos han dado buenos resultados prediciendo la magnitud de la STFT, pero no tanto a la hora de predecir su fase [9]. Es por eso que en este trabajo se explora la capacidad de las redes neuronales para estimar la RIR en el dominio del tiempo.

En esta sección, investigamos diversas estructuras de aprendizaje profundo con el propósito de descubrir un modelo con la capacidad de replicar una RIR determinada entre dos puntos de la sala. Comenzamos con enfoques simples y progresivamente aumentamos la complejidad. Para lograr esto, experimentamos con múltiples modelos en los cuales la salida buscada o *ground-truth* es la RIR inferida. Después, continuamos avanzando en el desarrollo utilizando el modelo que obtenga el menor error.

A. *Autoencoder Fully-connected*

El primer modelo que exploramos es el *autoencoder fully-connected* [10] que es una estructura que posibilita obtener una representación compacta de los datos de entrada en el espacio latente reduciendo la dimensionalidad de la RIR mediante capas densas. Se observa en la Figura 4.

El objetivo es que el *autoencoder* comprima la información relevante en la representación latente y luego la utilice para reconstruir los datos de entrada. Un *autoencoder* consta de tres partes principales:

- El codificador o *encoder* tiene el objetivo de reducir la dimensión de los datos de manera que la información relevante se conserve en la representación latente. El codificador consiste en capas de neuronas que aplican transformaciones lineales y funciones de activación para capturar patrones y características importantes en los datos de entrada.

- El espacio latente de un *autoencoder* es una representación de dimensionalidad reducida de los datos de entrada que se aprende durante el proceso de entrenamiento del *autoencoder*. Es el espacio donde se codifica la información más importante de los datos originales en una forma más compacta.

- El decodificador o *decoder* toma la representación latente generada por el codificador y la utiliza para reconstruir los datos originales. Al igual que el *encoder*, el *decoder* consiste en capas de neuronas que aplican transformaciones para generar una versión reconstruida de los datos. El objetivo es que la reconstrucción sea lo más fiel posible a los datos de entrada.

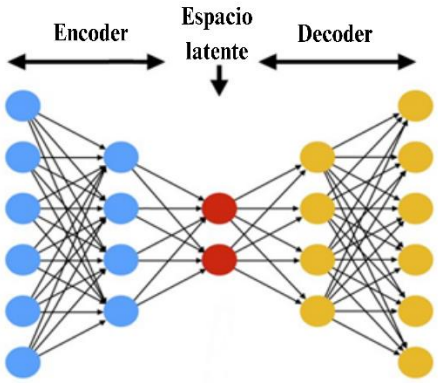

Figura 4. Autoencoder Fully-connected.

En un autoencoder el numero de entradas y salidas corresponde con el tamaño de la RIR que se pretende reconstruir, es decir, el numero de muestras de la RIR.

B. *Autoencoder* LSTM

El siguiente modelo es una modificación del *autoencoder fully-connected* del apartado anterior en el que se le aplica mayor complejidad añadiendo capas LSTM en el *encoder-decoder*. Las LSTM son de mucha utilidad para mapear secuencias de datos, como las RIR, ya que posibilita la capacidad de captar las dependencias temporales de la señal.

El modelo que se propone es una variante del modelo AE-FC (*autoencoder fully-connected*) que incorpora capas LSTM. A través de varias pruebas y experimentación, como se puede ver en la Figura 5, se ha determinado que la ubicación más efectiva para insertar estas capas LSTM es en el cuello de botella del modelo. En el diseño de la arquitectura se ha seguido un enfoque que comienza con una serie de capas densamente conectadas que conforman el *autoencoder*. Estas capas son responsables de la codificación y decodificación de los datos de entrada. En este punto, se puede introducir una o varias capas LSTM, que actúan como una extensión de la representación codificada. Ubicar las capas LSTM en el cuello de botella permite al modelo comprimir los datos de entrada en una representación compacta y contextualizada, en la que se capturan tanto las características importantes como las relaciones temporales.

C. Dual-Path Recurrent Neural Network

El *Dual-Path recurrent neural network* (DPRNN) es una arquitectura diseñada para modelar largas secuencias de audio en el dominio del tiempo mediante redes neuronales recurrentes (RNN). Debido a que las RNN muestran limitaciones en la representación de secuencias especialmente largas, en el artículo [11] se propone el DPRNN como solución basada en la reorganización de las capas RNN mediante dos RNN locales y globales. El modelo DPRNN se destaca por su simplicidad y baja parametrización, a pesar de sobrepasar el rendimiento del estado del arte en la tarea de separación de fuentes de audio mono, siendo este el objetivo del artículo original y la razón por la que la arquitectura DPRNN fue diseñada, y ha demostrado ser altamente eficaz en este contexto, superando las limitaciones de las RNN estándar. La arquitectura se muestra en la Figura 6 y consta de tres fases: segmentación, procesamiento de bloques y reconstrucción de la señal resultante mediante *overlap-add*:

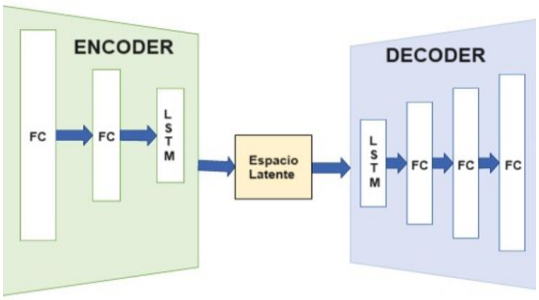

Figura 5. *Autoencoder* LSTM.

1. Segmentación

En el proceso de segmentación, se avanza a través de una RIR compuesta por L muestras, extrayendo gradualmente segmentos S de K muestras mediante un desplazamiento P *siendo* $P<K$. La uniformidad en la longitud de los segmentos es crucial, por lo que, si el último segmento contiene menos de K muestras, es posible completarlo con ceros al principio o al final de la secuencia completa de la RIR. En esta situación, para evitar la necesidad de agregar ceros, se ha optado por emplear un desplazamiento P y un número de K muestras por segmento que sea múltiplo de dos. Al tener una RIR con un número de muestras múltiplo de dos, se garantiza que todos los segmentos tengan una longitud uniforme.

La inclusión de un desplazamiento P en el proceso de segmentación resulta fundamental, ya que este aspecto contribuye significativamente a que las redes neuronales recurrentes del *Dual-PathRNN* capturen de forma más precisa las dependencias temporales presentes en la RIR, además de que ofrece la ventaja de ampliar el contexto tanto en la dirección pasada como en la futura de las secuencias temporales, enriqueciendo así la información disponible para el modelo.

2. Procesamiento de bloques

El DPRNN se compone de una etapa de procesamiento en bloques, donde cada bloque DPRNN se descompone en dos bloques adicionales para cada RNN. En el primer bloque, se emplea una LSTM bidireccional intra para el procesamiento de información local, mientras que el segundo bloque presenta una LSTM bidireccional inter, destinada a la elaboración de información global.

El bloque intra se encarga de modelar la secuencia de audio en un contexto local, procesando los datos de forma independiente. En contraste, los bloques inter se centran en extraer información de todos los segmentos para ejecutar un procesamiento global. Esto permite capturar relaciones de largo alcance y patrones de la secuencia completa, en lugar de solo considerar información local en cada segmento. Esto se logra transponiendo la salida del bloque intra antes de introducirlo al bloque inter.

3. Overlap-add

La fase final del DPRNN implica el uso del *overlap-add*, una técnica empleada en el procesamiento de señales que combina fragmentos superpuestos de señales, normalmente en el dominio temporal, para obtener la señal resultante.

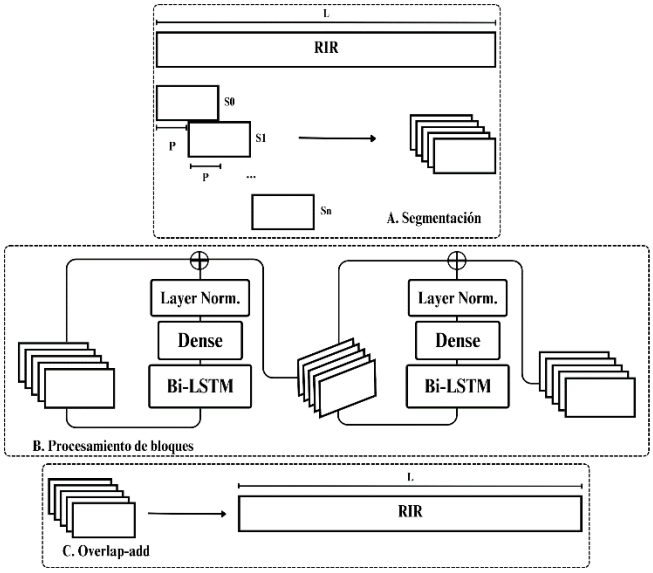

Figura 6. Esquema de las fases del DPRNN

En resumen, al igual que los autoencoders, el objetivo es que el modelo reconstruya la RIR introducida. Para ello, se introduce una RIR segmentada al modelo y mediante el procesamiento de bloques el modelo devuelve la RIR segmentada, la cual se reconstruye en la fase de overlap-add.

5. RESULTADOS

En esta sección se muestran los resultados obtenidos por las distintas redes neuronales profundas del apartado anterior a la hora de tratar de recrear la RIR inferida.

Las métricas utilizadas para el cálculo de los resultados es la función de pérdidas *mean square error* (MSE) en los datos de validación. Los datos de validación se utilizan para evaluar el rendimiento de un modelo en un conjunto de datos independiente que no se ha utilizado durante el entrenamiento. El MSE se define como:

$$MSE = \sum_i (y_i - \hat{y}_i)^2 \quad (7)$$

Donde y_i es la RIR *ground-truth* e $\hat{y}_i$ es la RIR inferida, siendo i el índice de la muestra temporal. En la tabla 1 se muestra el MSE obtenido para las distintas redes neuronales. Se puede ver como el DPRNN es el modelo que mejor se comporta reconstruyendo una RIR inferida.

Modelos	MSE
AE-FC	0.0146915
AE-LSTM	0.0026876
DPRNN	**2.0139009e-06**

Tabla 1. Resultados en la reconstrucción de la RIR inferida.

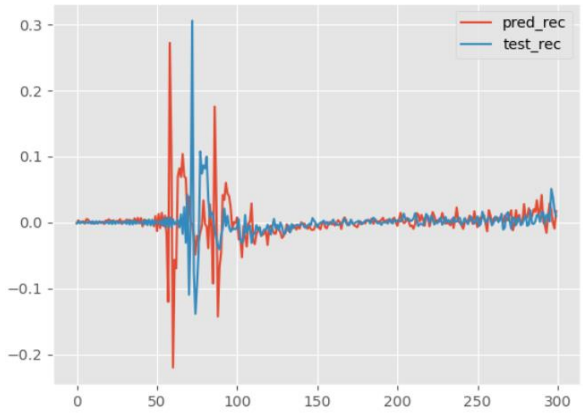

Figura 7. Comparación directa entre el *ground-truth* y la RIR predicha (solo las 300 primeras muestras). En azul el *ground-truth* y en rojo la RIR obtenida.

El DPRNN es el modelo que mejores resultados nos ha dado a la hora de modelar la RIR que introducimos y hemos decidido partir de esta arquitectura para predecir una RIR completamente nueva.

Por ello la propuesta que se plantea consiste en inferir dos RIR diferentes a partir de una misma matriz de RIRs, con el objetivo de predecir la RIR intermedia. En otras palabras, se emplearán dos RIR distintas como entradas del modelo, mientras que la RIR interpolada se considera la salida deseada. El objetivo es evitar que el modelo aprenda directamente de las salidas al introducir las RIRs que se desean predecir durante el proceso de entrenamiento.

El siguiente paso en el proceso consiste en inferir las dos RIRs de entrada utilizando dos bloques independientes de la arquitectura *Dual-Path Recurrent Neural Network*. Cada bloque DPRNN opera con una de las RIRs de entrada como su propia entrada y busca predecir la RIR intermedia correspondiente. Esto significa que cada bloque DPRNN se especializa en capturar patrones y características específicas de una RIR de entrada.

Una vez que ambas RIRs de entrada han pasado por sus respectivos bloques DPRNN y se han obtenido las predicciones de las RIRs intermedias, estas predicciones se combinan en un proceso de fusión. La fusión de las predicciones de las RIRs intermedias se realiza con el propósito de obtener una RIR final interpolada.

Los resultados de la Tabla 2 muestran cómo se comporta el modelo a la hora de inferir dos RIRs y combinarlas. Podemos ver que las pérdidas no son tan bajas, pero se alejan de la prueba anterior. No obstante, en la Figura 7 se observa un problema en la RIR predicha, y es que parece no ser capaz de decidir cuál de las dos RIRs tiene más peso que la otra.

Modelo	MSE
Double-DPRNN	5.9405043e-05

Tabla 2. Resultados de inferir dos RIR y predecir la RIR intermedia.

Al observar los resultados, queda claro que el modelo sigue siendo eficaz para capturar toda la información de las RIR inferidas, pero tiene dificultades al interpolar ambas RIR. Aunque existen técnicas de interpolación de RIRs, es importante destacar que estas son

computacionalmente más complejas en comparación con las técnicas de aprendizaje profundo. Recordar que esta es una de las razones por las cuales se optó por implementar soluciones basadas en redes neuronales profundas.

6. CONCLUSIONES

En este trabajo se han establecido los fundamentos para un nuevo estudio sobre metodologías de modelado y análisis de RIRs en el dominio del tiempo. La línea de investigación se mantiene abierta, y en el futuro se pretende seguir trabajando en el modelo DPRNN. Hay ideas como implementar mecanismos de atención [12], redes convolucionales unidimensionales [13] o ideas como acortar aún más las RIR.

ACKNOWLEDGEMENT

Este trabajo ha sido parcialmente financiado por MCIN/AEI/ 10.13039/501100011033, "ERDF A way of making Europe" a través del proyecto PID2021-124280OB-C21 y por PROGRAMA PROMETEO 2023-CIPROM/2022/20.

7. REFERENCIAS

[1] T. Betlehem et al. "Personal Sound Zones: Delivering interface-free audio to multiple listeners", *IEEE Signal Proc. Magazine*, *32*(2), 81–91, 2015.
[2] H. Kuttruff, Room Acoustics, Sixth Edition. Boca Raton, FL: CRC Press, 2016.
[3] Fernandez-Grande, E., Verburg, S. A., Karakonstantis, X. "Sound field reconstruction: towards large-scale spatial sensing." 10th Conv. of the European Acoustics Assoc., Torino, Italy, 2023.
[4] Zhao, S., Zhu, Q., Cheng, E., & Burnett, I. S. "A room impulse response database for multizone sound field reproduction". *The Journal of the Acoust. Soc. of America*, *152*(4), 2505–2512, 2022.
[5] Dasaradh S. K., "A Gentle Introduction to Math Behind Neural Networks" (2020)
[6] Yugesh Verma, "A Complete Understanding of Dense Layers in Neural Networks", 2021.
[7] Schmidt, R. M. "Recurrent neural networks (RNNs): A gentle introduction and overview", 2019
[8] Cristopher Olah. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[9] I. Martin et al., "Predicting Room Impulse Responses Through Encoder-Decoder Convolutional Neural Networks", 34th MLSP Conference, Rome, Italy, 2023.
[10] M. Sotaquirá, "Autoencoders: explicación y tutorial en Python" (2019)
[11] Luo, Y., Chen, Z., & Yoshioka, T., "Dual-Path RNN: Efficient Long Sequence Modeling for Time-Domain Single-Channel Speech Separation." *ICASSP*, 46–50, 2020.
[12] Subakan, C et al., "Attention Is All You Need In Speech Separation." *ICASSP*, 21–25, 2021.
[13] Q. Tao, F. Liu, Y. Li and D. Sidorov, "Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU," IEEE Access, vol. 7, pp. 76690-76698, 2019.