UNTREF UNIVERSIDAD NACIONAL DE TRES DE FEBRERO

INGENIERÍA DE SONIDO

Dereverberación del habla a partir de algoritmos de aprendizaje profundo†

Autor: Martin Bernardo Meza
Tutores: Ing. Leonardo Pepino

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

Agosto 2021

Índice

Índice de figuras

Índice de tablas

1. Introducción

1.1. Fundamentación

Las tecnologías que explotan el procesamiento digital de señales de voz mostraron grandes avances en las ultimas décadas, llegando a ocupar roles de primera importancia en nuestro día a día. Las investigaciones realizadas en este campo fueron impulsando diversas aplicaciones basadas en el análisis de la voz humana [1][2]. Estas tareas, en mayor o menor medida, deben lidiar con una característica intrínseca a cualquier emisión sonora dentro de un recinto: la reverberación. Las señales de voz que reciben las aplicaciones anteriormente nombradas por lo general se obtienen a partir de un transductor que no siempre se encuentra cercano a la fuente que desea registrar, provocando que la señal resultante capte la reverberación propia del entorno de origen de la señal. Esta reverberación interfiere en detrimento la señal de voz, produciendo una reducción en el rendimiento de aquellas aplicaciones que dependen de la integridad de dicha señal, como ser:

- Reconocimiento del habla [3]
- Verificación del hablante[1] [4]
- Localización del hablante [5]
- Inteligibilidad de la palabra

Si bien esta problemática fue abordada desde el enfoque de diversas técnicas de procesamiento de señales, en los últimos años este campo de estudio tuvo 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 que esta técnica demostró a la hora de resolver problemas pertinentes al procesamiento de imágenes y detección de patrones frente a los enfoques clásicos la pusieron al frente de las herramientas utilizadas para resolver problemas de este ámbito. Sin embargo, las tareas relacionadas al procesamiento de audio aun son un campo de estudio reciente para estas tecnologías, en donde todavía se presentan obstáculos para lograr una implementación plena de estas técnicas como por ejemplo:

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

la falta de bases de datos masivas de señales acústicas, la selección de una manera de representación óptima de las señales que permita explotar sus características intrínsecas, las maneras de medir el rendimiento de los procesos, entre otros.

Por este motivo, esta investigación pretende realizar un análisis de esta problemática desde el punto de vista de la ingeniería de sonido, para comprender las limitaciones de los modelos actualmente utilizados en este campo de estudio, 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

Los objetivos específicos son

- Realizar una revisión de las técnicas utilizadas para resolver el problema de dereverberación.
- Diseñar e implementar una estructura de red neuronal para dereverberación de señales de voz en lenguaje Python.
- Analizar las técnicas de pre y post procesamiento de datos, estudiando el impacto que tienen en el rendimiento del algoritmo.
- Optimizar el sistema propuesto, y comparar los resultados obtenidos con los modelos actuales de manera objetiva.
- Diseñar e implementar una interfaz gráfica en donde se permita visualizar los efectos del proceso de dereverberación aplicados a una señal particular.

1.3. Estructura de la Investigación

2. Estado del Arte

En los últimos años se ha registrado un marcado desarrollo y progreso en el campo de el procesamiento de señales del habla. En este campo, la dereverberación ocupa un rol crucial debido a que es una característica que influye en la mayoría de las aplicaciones del procesamiento de señales del habla.

Los primeros enfoques que apuntaron a resolver el problema de la dereverberación fueron aquellos referidos a 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 enfoques basados en los modelos matemáticos de la generación del habla [8]. Se implementaron algoritmos que se basaban en el estudio de la señal de residuo obtenida luego de la predicción lineal del habla. Se detectó que esta señal residuo contenía información sobre los efectos de la reverberación, por lo cual se la utilizó para excitar filtros variantes en el tiempo que al aplicarse sobre la señal del habla mostraban una mejora respecto a la eliminación de los efectos reverberantes [9]. También se realizaron análisis de dereverberación a partir del uso de varios transductores, aplicando técnicas de descomposición sobre el conjunto de señales captadas [10]. Otras características propias de la señal del habla fueron explotadas en pos de eliminar los efectos de la reverberación, tales como la estructura armónica [11], y el espectro de modulación [12].

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].

A partir del año 2006 en el campo del estudio de la separación de fuentes se popularizó un enfoque al problema denominado Análisis Computacional de la Escena Auditiva [16] que está inspirado en la teoría perceptiva del Análisis de la Escena Auditiva [17] la cual intenta explicar la capacidad del sistema auditivo de descomponer una señal captada en varias señales correspondientes a diferentes fuentes de sonido. Este enfoque trajo consigo el uso de máscaras binarias ideales en el dominio temporal-frecuencial para extraer las señales buscadas [18]. Las máscaras se definen como ideales ya que su obtención requieren 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 un espectrograma, o un cocleograma) y luego asignarle a cada punto del espacio temporal-frecuencial un valor de 1 cuando su energía pertenece a la señal objetivo, y un valor de 0 en el caso contrario. Roman et. al. [19] 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 [20]. Hazrati et al. [21] 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.

Los primeros antecedentes de la implementación de redes neuronales en la tarea de la dereverberación se encuentran desde el año 2007. Jin y Wang [22] aplicaron la estructura de perceptrón multicapa para estimar las mascaras binarias necesarias para la separación de la componente reverberante en una señal voz. La estructura de red neuronal debía realizar el mapeo entre características extraídas de la señal de entrada y cada unidad temporal-frecuencial de la señal de salida. 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. En 2014 Kun et al. [23] 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 de una señal reverberante. 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 será la red quien aprenda la forma de ese filtro inverso. Entonces, esto se logra entrenando una red que tiene como entrada el espectro de la señal reverberante y como salida (es decir, como objetivo) el espectro de la señal anecoica. Se implementan soluciones desde el post-procesamiento para lograr reconstruir la fase de la señal estimada. Por otro lado, Weninger et al. [24] implementaron redes neuronales recurrentes bidireccionales de larga memoria de corto término en la tarea de la dereverberación en pos de conservar la continuidad del habla. Luego, se pasan a utilizar redes neuronales convolucionales [25]. Estas ofrecen una mayor habilidad de modelado, permitiendo considerar patrones locales presentes en la representación temporal-frecuencial. Además, utilizan menos parámetros y distribuyen de manera mas eficiente los pesos sinápticos, lo que se traduce en un menor costo computacional de procesamiento. Globalmente, los modelos de redes convolucionales logran ser mas eficientes y mas precisos en sus resultados. Por lo general se utilizan estructuras secuenciales, formando estructuras denominadas codificadores-decodificadores, en las cuales la información temporal-frecuencial de entrada sufre una compresión que disminuye su dimensión derivándose a un espacio latente, para luego a partir de este espacio poder estimar el espectro objetivo que corresponde a la señal dereverberada. Como este tipo de redes consideran cada punto de tiempo-frecuencia en un contexto de pocos puntos contiguos, el siguiente paso fue implementar redes completamente convolucionales [26], en las cuales se contempla el conjunto completo de puntos de tiempo-frecuencia. Este último presentó mejores resultados que los modelos mas acotados. Entre los estudios mas recientes, se encuentran enfoques que proponen la combinación de métodos utilizados previamente que demostraron un buen funcionamiento en algún aspecto para lograr que en conjunto logren minimizar el error que producen por separado [27].

3. Marco Teórico

3.1. Representación temporal y frecuencial de señales

Fundamentalmente, el audio esta compuesto por formas de ondas. Cuando un objeto vibra genera ondas de presión que cuando alcanzan nuestros oídos son percibidas como sonido [28]. Si bien entonces podemos definir a una señal de audio como una variación continua, para su análisis digital nos interesa estudiar este tipo de señales en un dominio discreto. Para lograrlo, se toman muestras equiespaciadas de la señal continua, en un proceso que se denomina muestreo. La distancia temporal entre dos muestras contiguas es determinado según la frecuencia máxima que se desea representar, acorde al teorema de Nyquist [29]. Entonces, una señal continua $x_a(t)$ que es muestreada a una frecuencia de $f_s = 1/T_s$ muestras por segundo, produce la señal discreta $x(n)$ que se puede definir a partir dela señal continua a partir de la ecuación 1, que equivale a la representación vectorial vista en la ecuación 2, siendo N el número de muestras tomadas.

$$x(n) = x_a(nT_s) \tag{1}$$

$$x_n = [x(0), x(1), ..., x(N-1)]^T \tag{2}$$

Partiendo de esta representación temporal de la señal, se puede obtener una representación frecuencial de la misma a partir de la Transformada Discreta de Fourier (DFT) [29] que matemáticamente equivale a la expresión 3, lo cual es útil para poder realizar un análisis mas profundo de la señal. La DFT permite entonces representar a la señal a partir de componentes frecuenciales complejas. Es decir que para cada punto se tiene un valor de amplitud y un valor de fase. A su vez, esta transformación supone un proceso reversible, por lo cual la señal temporal puede ser recuperada partiendo de la señal frecuencial, aplicando la expresión 4.

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

$$x[n] = 1/N \sum_{k=0}^{N-1} X[k] e^{jnk2\pi/N} \qquad 0 \leq n \leq N-1 \tag{4}$$

El cálculo de esta transformación es costoso en términos computacinoales, y hay un alto grado de redundancia en este proceso. Por esto, comúnmente se utiliza una implementación definida como transformada rápida de Fourier que permite optimizar el cómputo de esta transformación [30].

Cuando se trabaja con señales no estacionarias es de interés evaluar la variación del espectro de frecuencias en el tiempo. Para esto se utiliza una transformación denominada transformada de Fourier de corto plazo (STFT por sus siglas en inglés) la cual consiste en una representación tridimensional formada al calcular la transformada de Fourier para sub-intervalos temporales de la señal, y luego representarlos de manera contigua [28]. De esta manera se obtiene un gráfico con dimensiones de tiempo, frecuencia y amplitud en donde se puede ver la evolución del espectro en función del tiempo. Un ejemplo de un espectrograma donde se conserva solo la magnitud se puede ver en la Figura 1.

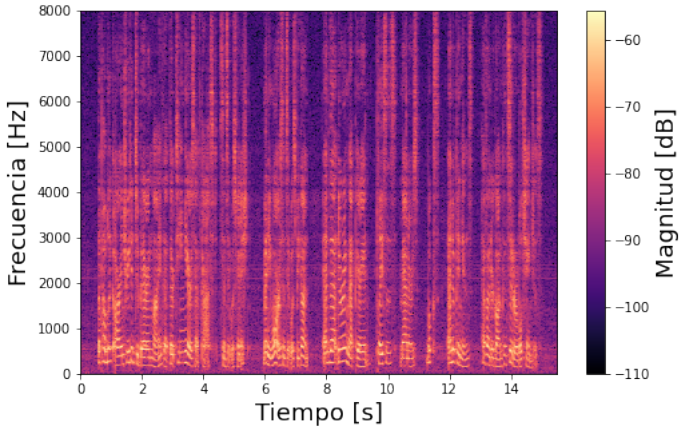


Figura 1: Espectrograma de una señal de audio

3.2. Respuesta al impulso y reverberación

Si en un recinto se tiene una fuente y un micrófono captando a una cierta distancia de la fuente, las ondas sonoras que emite la fuente se reflejarán en las paredes del recinto y alcanzarán el micrófono inmediatamente después que la onda sonora directa. Las reflexiones continúan

ocurriendo, y cada instancia de reflexión supone una disminución de la energía sonora 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 mas corto entre la fuente y el punto de captura se 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. Esta depende tanto de las condiciones del recinto como de las características del micrófono.

La función de transferencia entre la fuente emisora y el micrófono se define como la respuesta al impulso del recinto y usualmente se denota como $h(t)$. Este será diferente para cualquier punto en el espacio dentro del recinto. Haciendo un análisis temporal de una respuesta al impulso, podemos identificar 3 partes: en primer lugar el nivel de sonido directo (producido por la onda que viaja a través de camino directo), las reflexiones tempranas (cuyo limite temporal vendrá definido por las características propias de cada recinto) y por último la cola reverberante. Esto se ve representado en la Figura 2. 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 mas bien un proceso estocástico, y al depender de un mayor número de reflexiones no varia drásticamente frente a pequeños cambios de geometría. Analíticamente, la parte temprana y tardia de una respuesta al impulso se define segun las ecuaciones 5 y 6 respectivamente.

$$h_e(t) = \begin{cases} h(t) & t_d - t_0 \leq t \leq t_d + t_0 \\ 0 & e.o.c \end{cases} \tag{5}$$

$$h_l(t) = \begin{cases} h(t) & t < t_d - t_0 \\ h(t) & t > t_d + t_0 \\ 0 & e.o.c. \end{cases} \tag{6}$$

En donde, $h(t)$ corresponde a la respuesta al impulso, $h_e(t)$ corresponde a la parte temprana, $h_l(t)$ corresponde a la parte tardia, t_d es el tiempo de retardo del camino directo y t_0 es el parámetro que define el largo temporal de la ventana de tolerancia. Comúnmente se utiliza un valor de $t_0 = 2{,}5ms$.

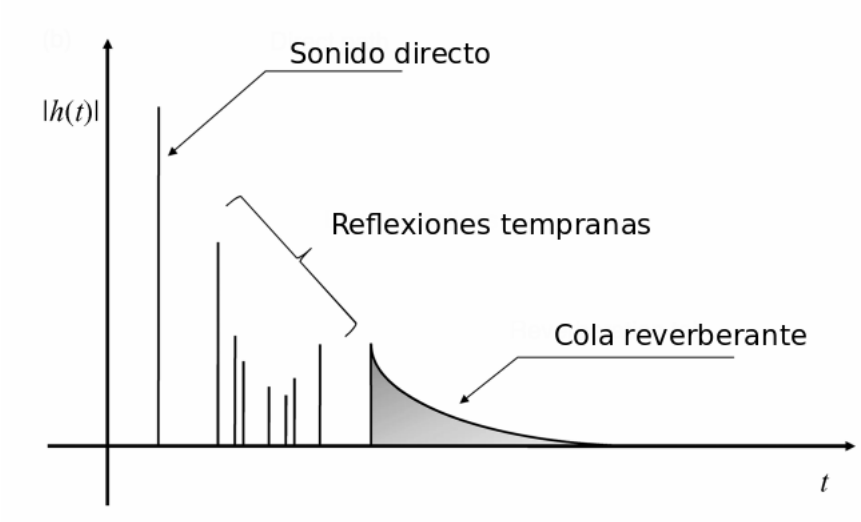


Figura 2: Secciones temporales de una respuesta al impuulso

Idealmente, el micrófono captura una señal que corresponde a la convolución entre la respuesta al impulso del recinto y la señal 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.

$$x(t) = h(t) * s(t) \tag{7}$$

$$X(f) = H(f)S(f) \tag{8}$$

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.

3.2.1. Relación directo-reverberado

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 trayectoria directa, y las muestras que continúan luego de n_d representan solo la reverberación producida por las trayectorias reflejadas.

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

Este parámetro es dependiente de la distancia entre el punto emisor y receptor, y del tiempo de reverberación del recinto. Realizando un análisis energético [31] se puede obtener una expresión equivalente para este descriptor, la cual se muestra en la ecuación 10.

$$DRR[dB] = 10Log_{10}(\frac{QR}{16\pi D^2}) \tag{10}$$

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 [32]. La formulación de este parámetro se basa en el hecho de que la cola reverberante de cualquier respuesta al impulso puede ser modelada como ruido blanco Gaussinano 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 termino 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 [33]. 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) [34]. 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 contempla la calidad general de la señal dereverberada.

3.4. Redes neuronales y algoritmos de aprendizaje

La base de los algoritmos de aprendizaje profundo se encuentra en la neurona artificial. Esta consiste en un modelo que parte de los principios de funcionamiento de las neuronas biológicas [35]. El perceptrón [36] fue de las primeras arquitecturas formalmente implementadas y que se considera la unidad básica de estos algoritmos. Un esquema de una neurona artificial básica se puede ver en la Figura 3. Las componentes básicas de una neurona artificial son:

- **Entradas**: Recibe los datos que van a ser procesados en esta unidad.
- **Pesos sinápticos**: Parámetros de ponderación. Cada entrada se asocia a uno de estos

parámetros. Es el valor que se va ajustando cuando el modelo se encuentra en la etapa de entrenamiento. En ellos se ve reflejado la propagación del error.

- **Suma ponderada**: Los pesos sinápticos se asocian a cada entrada a partir de una regla de propagación que consiste en una suma ponderada.
- **Función de activación**: Función que se aplica a la salida de la suma ponderada, cuya salida representa la salida final de la unidad. Esta se determina de manera de poder agregar complejidad al modelo. Algunos ejemplos de funciones de activación se pueden ver en la figura 4.
- **Salida**: Es el resultado de aplicar el proceso completo al conjunto de entradas. En una estructura, esta salida puede ser la entrada de una o varias unidades subsiguientes.

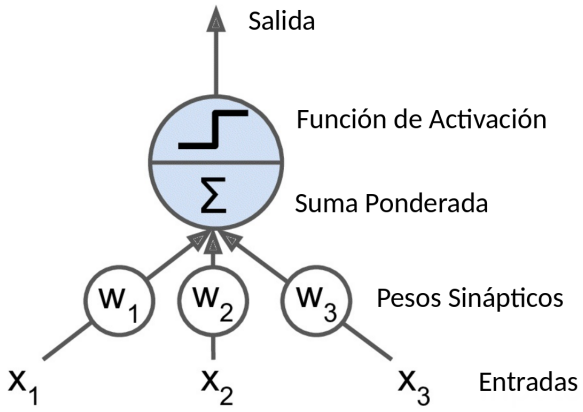


Figura 3: Esquema de neurona artificial

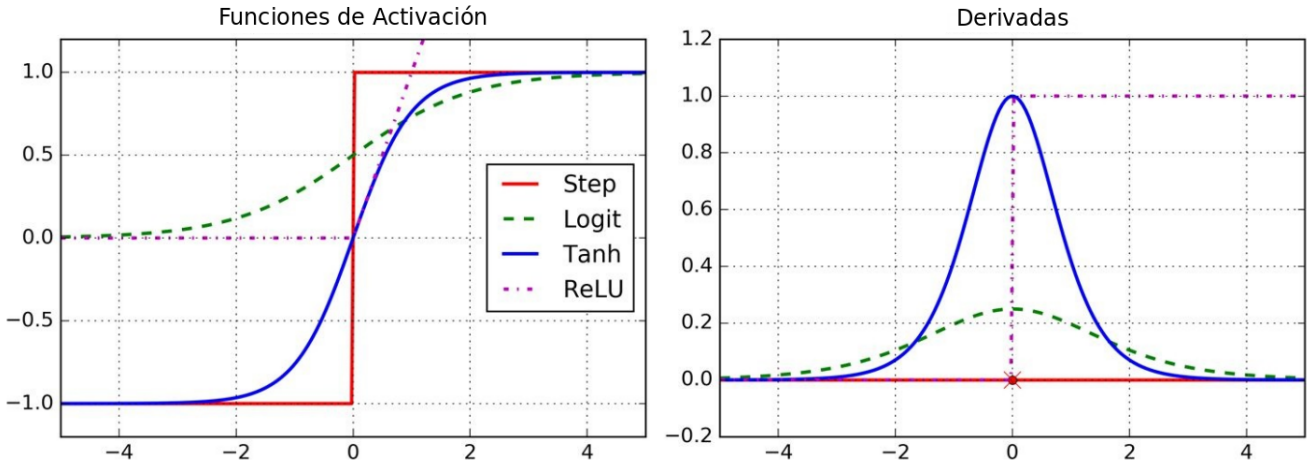


Figura 4: Funciones de activación y sus derivadas

3.4.1. Modelos basados en redes neuronales

Los modelos basados en redes neuronales son sistemas compuestos por capas que agrupan unidades computacionales (neuronas artificiales). En una capa, las entradas y salidas de las neuronas artificiales que la componen están agrupadas. Diferentes capas se relacionan formando sistemas de acuerdo al problema que se busque resolver. En general, un sistema se compone por una capa de entrada, una capa de salida, y un número finito de capas ocultas intermedias. De esta forma, el sistema recibe valores de entrada, los procesa a través de las distintas capas que componen la red, y otorga valores de salida. Un esquema de este funcionamiento se puede ver en la Figura 5.

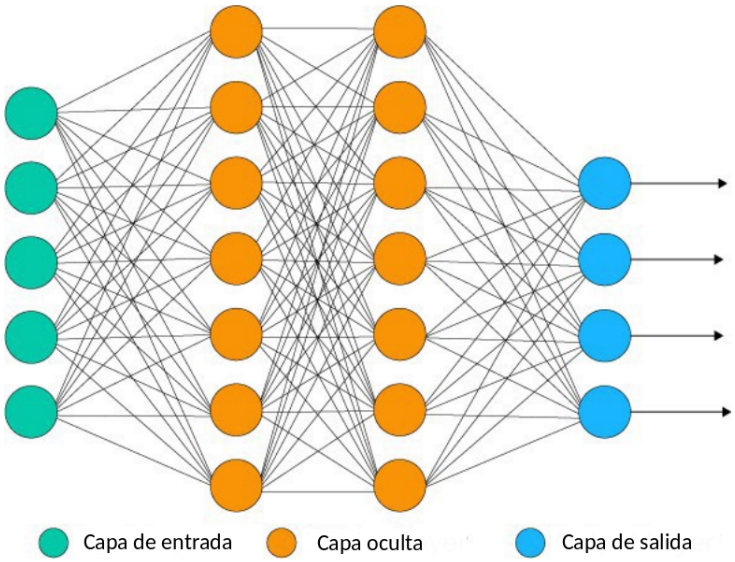


Figura 5: Esquema básico de red neuronal

Catalogar a estos sistemas como de aprendizaje 'profundo' hace referencia al hecho de tener sucesivas capas de representación [37]. Cuando un modelo se compone de un mayor numero de capas, se lo considera mas 'profundo'.

Estas estructuras de redes neuronales aprenden automáticamente al ser expuestas a un conjunto de datos. El proceso de aprendizaje se puede pensar como la evaluación de un mapeo de valores de entrada a ciertos valores objetivos de salida. Esto es, se toman valores de entrada, se los transforman a lo largo de las capas que componen la red produciendo valores de salida, y se comparan estas salidas con los valores de salida objetivos. Entonces, la especifi-

cación del proceso que está siendo implementado por el sistema se encuentra reflejado en los pesos sinápticos de las neuronas que componen cada capa. El aprendizaje se obtiene a partir de poder modificar estos pesos sinápticos acorde a las diferencias que se obtengan entre las salidas producidas por la red y las salidas que se tienen como objetivo. Para lograr esto último, los algoritmos de redes neuronales utilizan determinadas funciones:

- **Función de costo**: También denominada función objetivo, o función de pérdida. Recibe las salidas de la red y las salidas esperadas y evalúa que tanto difieren entre sí. Estas diferencias las traduce a una medida de distancia a partir de una expresión matemática que se define en función del problema que se busca resolver. Entonces, para cada estimación de la red, esta función otorga un puntaje que explica cuan lejos está el valor estimado del valor pretendido.

- **Función de optimización**: Esta función aplica el algoritmo de propagación del error hacia atrás, que es una parte fundamental de un algoritmo de aprendizaje profundo. Este cálculo permite estimar el aporte que tiene cada peso sináptico en el error final de la estimación de la red, y por lo tanto permite ajustar los valores de estos pesos sinápticos estratégicamente para conseguir minimizar la distancia computada por la función de costo.

Entonces, la salida de la función de costo se utiliza como realimentación del sistema a través de la función de optimización. De este modo, el entrenamiento consiste en un bucle en el cual en cada iteración el sistema evalúa una instancia de los datos de entrenamiento (valores de entrada y de salida), y ajusta los pesos sinápticos en pos de reducir el error calculado. Un esquema que expone este funcionamiento se ve en la Figura 6. Repetir este ciclo un número suficiente de veces conduce a la convergencia del valor de error.

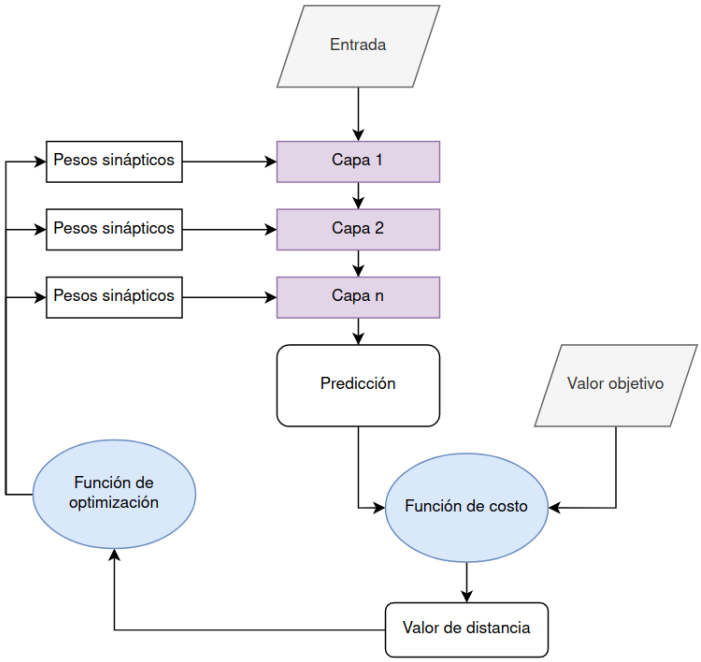


Figura 6: Diagrama de flujo del bucle de entrenamiento de una sistema de red neuronal

Por último, la manera en que el sistema de red neuronal recibe y procesa los datos 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 del procesamiento de este conjunto sirven para ajustar variables que requieren ser especificadas de manera previa al entrenamiento. Estos parámetros se denominan hiper parámetros.
- **Conjunto de prueba**: Este conjunto es el que se utiliza para medir el rendimiento final del sistema. Como contiene instancias que no fueron utilizadas en las etapas de entrena-

miento y ajuste de parámetros, el análisis del procesamiento de este conjunto sirve para medir el nivel de generalización que el sistema logró alcanzar.

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 pesos sinápticos de cada capa. Cuando la red procesó todos los lotes que componen el conjunto de datos de entrenamiento se dice que transcurrió una época. El proceso de entrenamiento depende en cierta medida del tamaño de los lotes [37]. Si consideramos la curva de la función de costo de un parámetro como la de la Figura 7, vemos que existen mínimos locales y mínimos globales a lo largo de la misma. En el proceso de entrenamiento se busca minimizar este valor de costo. Si se toman lotes muy pequeños, lo que se traduce en desplazamientos pequeños a lo largo de esta curva, se corre el riesgo de quedar confinado en un mínimo local. De igual manera, un conjunto demasiado grande produciría saltos demasiado grandes en comparación a las fluctuaciones de esta curva, haciendo que se obtengan valores de costo aleatorios.

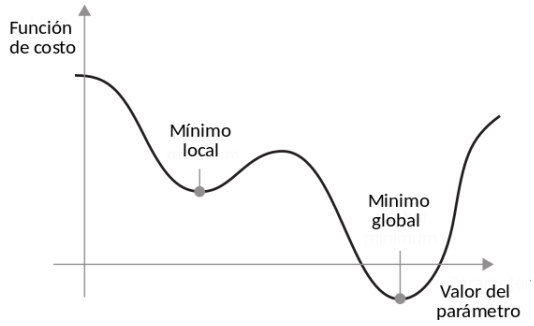


Figura 7: Curva de costo para un parámetro

El orden en el que los datos son presentados ante el modelo puede también influenciar positiva o negativamente en su rendimiento. Al igual que los humanos y animales algunos algoritmos logran aprender de manera más eficaz cuando los ejemplos no se les presentan de manera aleatoria sino mas bien organizados siguiendo un orden significativo en donde la complejidad de la representación a aprender aumente gradualmente. Esta estrategia de entrenamiento se conoce como ’curriculum learning’ [38] y puede lograr que ciertos modelos logren mejores niveles de generalización o bien logran converger mas rápidamente. Se la puede pensar como una estrategia de optimización global.

3.4.2. Redes neuronales convolucionales

Las redes neuronales convolucionales emergen del estudio de la corteza visual del cerebro. En los últimos años, estas estructuras fueron utilizadas para resolver tareas visuales complejas (análisis de imágenes). El componente principal es la capa convolucional. Para el análisis de imágenes, estas capas se concatenan de manera que la primera capa no contempla cada píxel de la imagen, sino que solo se enfoca un número acotado de píxeles que caen dentro de su campo perceptivo. 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 8.

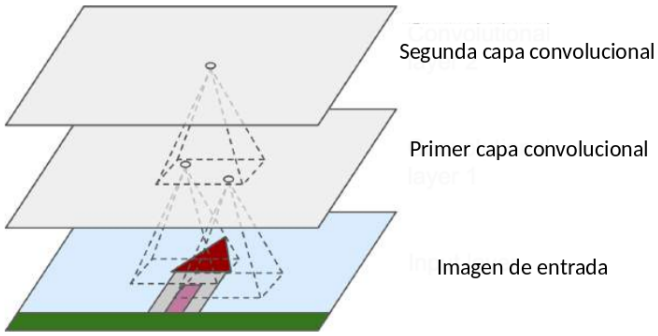


Figura 8: Capas convolucionales con campos receptivos locales rectangulares

Formar esta estructura le permite a la red aprender diferentes patrones estructurales locales de manera jerárquica [39]. Estas capas se distinguen por dos propiedades fundamentales:

- Los patrones que se aprenden son invariantes al desplazamiento. Esto quiere decir, que si se aprende de un patrón ubicado en un lugar específico de una imagen, este mismo patrón puede ser identificado en cualquier otra ubicación dentro de la imagen.
- Cuando estas capas se concatenan formando redes logran aprender jerarquías espaciales de patrones. Esto les permite aprender de manera eficiente conceptos visuales cada vez más complejos y abstractos.

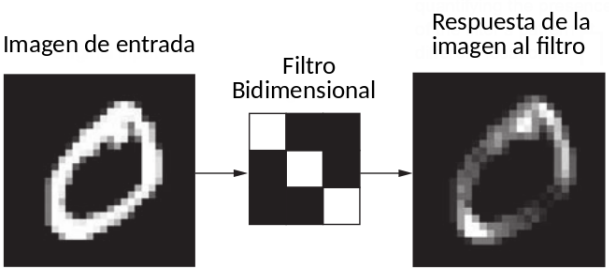


Figura 9: Representación de la aplicación de un filtro bidimensional sobre una imagen

El procesamiento que ocurre en una capa convolucional consiste en la aplicación de uno o varios filtros bidimensionales sobre la imagen de entrada, lo cual genera mapas de respuestas que representan la presencia del patrón del filtro a lo largo de la imagen, como se puede apreciar en la Figura 9. En este tipo de capas, el aprendizaje se traduce en determinar la forma de los filtros que se deben aplicar para conseguir los resultados esperados. Teniendo en cuenta este proceso, las variables que se deben definir en cada capa son:

- **Tamaño del filtro**: Define el tamaño del campo perceptivo de cada unidad de procesamiento de la capa. Valores comunes son $3x3$ o $5x5$. En la Figura 10 se ve un ejemplo de un filtro de tamaño $3x3$.

- **Tamaño del salto**: Determina la distancia horizontal y vertical entre campos perceptivos 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 10 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 [39].

- **Cantidad de filtros aplicados**: El número de filtros computados por la convolución.

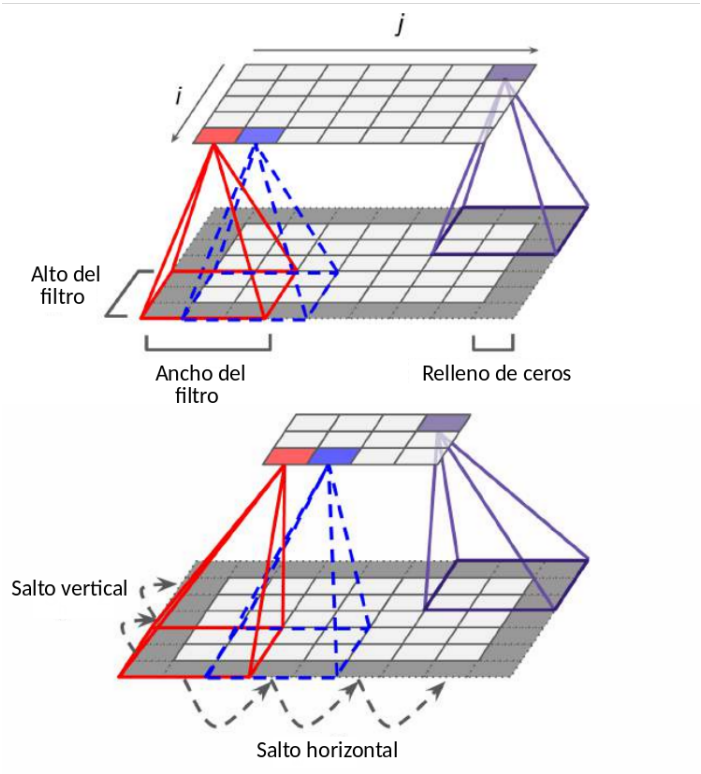


Figura 10: Parámetros de procesamiento en capas convolucionales

De esta manera, las capas convolucionales permiten construir modelos con menos cantidad de parámetros a entrenar, distribuidos adecuadamente para conseguir la generalización de conceptos visuales complejos.

3.5. Dereverberación por filtrado temporal-frecuencial

3.5.1. Máscaras de amplitud

Entre los numerosos métodos estudiados para abordar la tarea de la dereverberación, en la actualidad el análisis del espectro temporal-frecuencial ha tomado una importancia central debido a la posibilidad de explotar este enfoque utilizando herramientas como redes neuronales convolucionales y otros algoritmos de aprendizaje profundo. 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 descifrar el proceso necesario para obtener un nuevo espectro que se corresponda con la señal anecoica (descartando el efecto de la reverberación). Entonces, el pro-

ceso de dereverberación se puede resumir a la estimación de un filtro variable con el tiempo que se aplica sobre el espectrograma con reverberación. Estudios previos demostraron que la fase no aporta información significativa para estas tareas [40][41], por lo cual se pueden realizar estos procesos únicamente sobre la magnitud de los espectrogramas, descartando la información de fase. Considerando esto, el proceso de dereverberación se reduce a la expresión de la ecuación 11, en donde $STFT_Y$ es el espectrograma de amplitud la señal anecoica, $STFT_X$ es el espectrograma de amplitud de la señal con reverberación y M es la máscara ideal que representa el filtrado en el dominio tiempo-frecuencia. En otras palabras, el espectro de amplitud de la señal dereverberada se obtiene aplicando la máscara ideal sobre el espectro de amplitud con reverberación.

$$STFT_Y(t,f) = M(t,f)STFT_X(t,f) \tag{11}$$

$$M(t,f) = \frac{STFT_Y(t,f)}{STFT_X(t,f)} \tag{12}$$

Por otro lado, de la ecuación 12 se puede inferir que la mascara ideal puede tomar valores en el dominio $[-\infty, +\infty]$. Esto puede ser un contraproducente para la utilización de algoritmos de aprendizaje supervisado, donde lo conveniente es trabajar con instancias acotadas en un rango de valores del dominio $[-1, +1]$. Para conseguir cambiar el dominio de las máscaras se realiza una compresión de los valores a través de una función tangencial hiperbólica. La misma se expresa en la ecuación 13, en donde $M^{'}(t,f)$ corresponde a la máscara comprimida. Luego de aplicar esta transformación, el dominio resultante pasa a ser $[-Q, +Q]$. El parámetro C controla la pendiente de la tangente hiperbólica.

$$M^{'}(t,f) = Q\frac{1 - e^{CM(t,f)}}{1 + e^{CM(t,f)}} \tag{13}$$

De igual manera, partiendo de una máscara comprimida se puede recuperar la máscara original a través de la ecuación 14.

$$M(t,f) = \frac{-1}{C} log(\frac{Q - M^{'}(t,f)}{Q + M^{'}(t,f)}) \tag{14}$$

3.5.2. Sintesis de audio a partir de espectrogramas

Para realizar procesamiento de audio a partir de modificar espectros temporales-frecuenciales es necesario poder pasar de información de audio temporal a un dominio temporal-frecuencial (utilizando una transformación como la transformada de corto término de Fourier) y también poder recuperar información de audio partiendo de un espectrograma que ha sido modifcado. Esto último puede ser un problema, porque ciertos procesos pueden generar espectrogramas que no sean consistentes, es decir, que no haya ninguna señal en el dominio temporal que se condiga con el espectrograma generado. Para solucionar esta cuestión se desarrollaron algoritmos que buscan estimar una señal temporal cuyo espectrograma sea el mas cercano posible al espectrograma que se quiere antitransformar. Este es el caso del algoritmo propuesto por Griffin et. al. [42]. El algoritmo consiste en un bucle iterativo que busca minimizar el error cuadrático medio entre la señal estimada y el espectrograma modificado. En la figura 11 se muestra un diagrama de bloques que explica el funcionamiento básico del algoritmo.

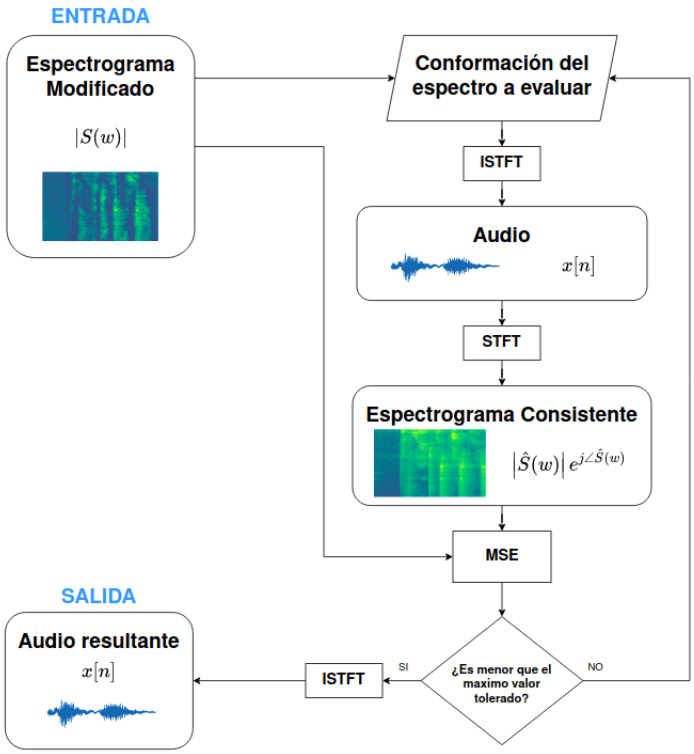


Figura 11: Diagrama en bloques del algoritmo de Griffin-Lim.

El espectrograma de entrada $|S(w)|$ inicialmente se combina con una fase aleatoria para

formar un espectrograma complejo. Este se antitransforma obteniendo una señal de audio, la cual vuelve a ser transformada obteniendose un nuevo espectrograma complejo $\left|\hat{S}(w)\right| e^{j\angle\hat{S}(w)}$. Este espectrograma es consistente, pues deriva de la transformación de una señal de audio real. Se puede probar que combinar esta fase resultante $\angle\hat{S}(w)$ con el espectrograma modificado de entrada $|S(w)|$ disminuye el error cuadrático entre los espectrogramas evaluados (es decir, entre el espectrograma consistente y el inconsistente). De esta manera, este proceso se repite hasta lograr que el error cuadratico medio descienda hasta un cierto valor deseado. Cuando esta condición se cumple, el espectrograma complejo que causa esta condición se antitransforma generando la señal de audio resultante final.

4. Metodología

4.1. Análisis de datos

Para poder entrenar un algoritmo de aprendizaje profundo se requiere un conjunto de datos extensos. Este conjunto debe poder representar de la mejor manera posible el fenómeno que se quiere procesar. Además, se debe poder asegurar que todas las instancias que componen la base de datos tengan características homogéneas que se adecuen a los procesos subsiguientes.

Para este trabajo, es necesario partir de un conjunto de datos conformados por dos tipos de elementos principales: Respuestas al impulso, y grabaciones de voz. Con estos elementos, es posible generar instancias que comprendan información de audio con reverberación y su correspondiente versión anecoica, siendo esta última la que un sistema de dereverberación tiene como objetivo.

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. Las características de estos conjuntos deberán variar de acuerdo al propósito de cada uno para lograr optimizar cada etapa, o bien, para evaluar ciertos aspectos de estos procesos.

4.2. Base de datos de respuestas al impulso

Para este trabajo se utilizan respuestas al impulso reales y simuladas. A su vez, tambien 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

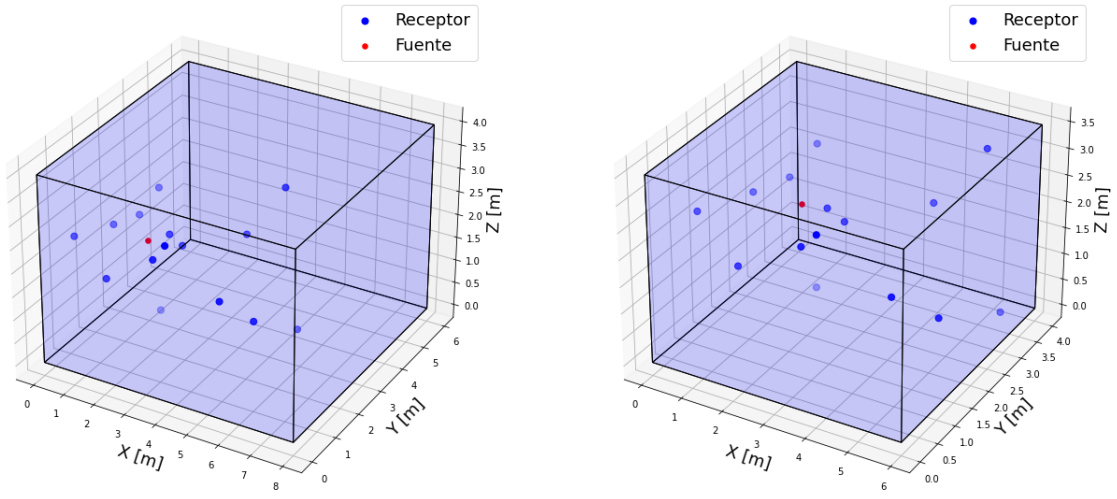
Las respuestas al impulso reales se obtienen del conjunto de datos C4DM [43]. Este conjunto consiste en una colección de respuestas al impulso que fueron medidas en tres recintos: una sala multipropósito con aproximadamente 800 asientos, un edificio victoriano construido en 1988 originalmente diseñado para ser una biblioteca, y una sala de clases de una universidad. Las mediciones fueron realizadas utilizando la técnica del barrido frecuencial [44]. Para todas estas respuestas al impulso, el tiempo de reverberación es de aproximadamente 2 segundos.

4.2.2. Respuestas al impulso simuladas

En cuanto a las respuestas al impulso simuladas, se utiliza la librería de Python 'PyRoomAcoustics' [45] para generarlas. Esta librería brinda un software de generación de respuestas al impulso basado en el método de fuente imagen [46]. 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 $8m\,x\,6m\,x\,4m$ que se denominará 'Recinto 1' , el segundo de dimensiones $6m\,x\,4m\,x\,3{,}5m$ que se denominará 'Recinto 2'. En la figura 12 se pueden visualizar ambos recintos generados.



(a) Recinto 1 *(b) Recinto 2*

Figura 12: 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.

4.2.3. Respuestas al impulso generadas por aumentación

Este conjunto se genera partiendo de un subconjunto de respuestas al impulso reales. El proposito de este proceso es partir de un conjunto de impulsos escasos con determinadas características, y generar un conjunto mucho mas grande de respuestas al impulso controlando de manera paramétrica ciertos descriptores acústicos como el tiempo de reverberación $T60$ y

la relación directo-reverberado DRR, de manera tal que se pueda asegurar un cierto balance en el conjunto conformado [47]. El proceso de aumentación entonces se divide en dos procesos principales: 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 caida para controlar el tiempo de reverberación.

Para el primer proceso, a la parte temprana de la respueta al impulso $h_e(t)$ 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 parte temprana obteniendo una parte temprana ventaneada y un residuo ventaneado. A partir de esto, la parte temprana se puede definir segun la ecuación 15.

$$h_e(t) = \alpha w_d(t) h_e(t) + [1 - w_d(t)] h_e(t) \tag{15}$$

En donde $w_d(t)$ corresponde a una ventana Hann de $5ms$ de longitud. De esta manera, partiendo de esta ultima 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 13 se puede observar una representacion de una parte temprana $h_e(t)$, las ventanas aplicadas, el efecto del factor de ganancia α y la nueva señal $\tilde{h}_e(t)$ generada. Finalmente, esta parte temprana modificada se concatena con el resto de la respuesta al impulso completando asi el proceso de aumentación referido a la relación directo-reverberado.

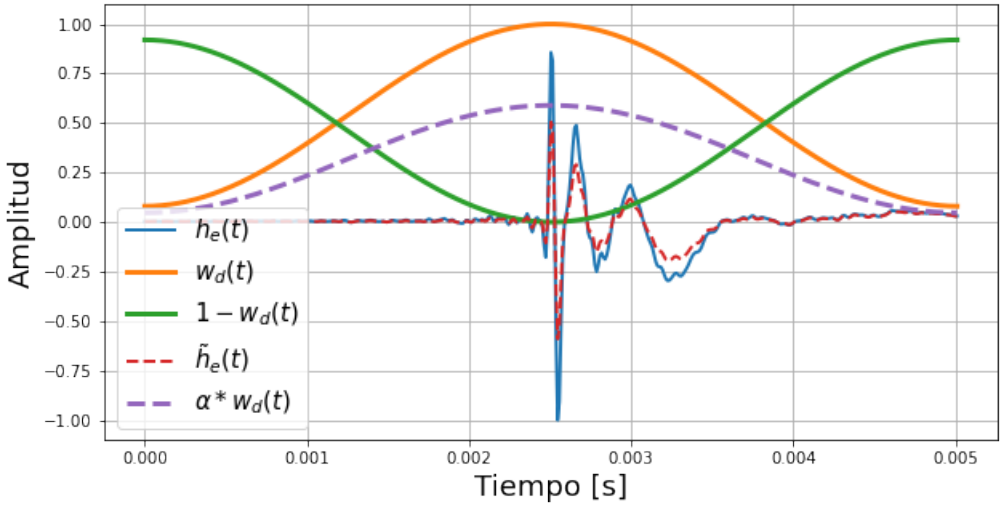


Figura 13: Señales involucradas en el proceso de aumentación de DRR

Luego, para la modificación del tiempo de reverberación $T60$

4.3. Bases de datos de señales del habla

Las señales del habla necesarias para formar los pares anecoico-reverberados se obtienen de la librería LibriSpeech [48] 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 y máscaras ideales para poder entrenarse. 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 tratan con señales de voz que concentran su información por

debajo del límite de representación frecuencial de $8000Hz$ impuesto por esta decisión. 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 $75\,\%$. Esto permite la correcta reconstrucción de la señal al antitrasformar. 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. Con las magnitudes de los espectros anecoicos y reverberados, se calculan máscaras de amplitud ideales y luego se comprimen aplicando una función tangencial hiperbólica para la cual se definen los parámetros $Q = 1$ y $C = 0,5$. Finalmente, las instancias finales de este proceso son en el espectro de magnitud de la señal con reverberación (que corresponde a la variable de entrada de la red neuronal) y la máscara de magnitud ideal comprimida (que corresponde a la salida de la red, es decir, el objetivo que el modelo busca estimar). Ambas instancias tienen las mismas dimensiones, que corresponden a 256 cuadros temporales y 257 valores posibles de frecuencia (se conserva solo 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 $256x256$ lo cual facilita el proceso de compresión y expansión de los espectros al ser dimensiones múltiplos de 2. Se descarta la frecuencia más alta ya que por la característica de la fuente no contendrá información crucial para la representación.

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. Mas 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 a través de la aplicación del algoritmo de Griffin-Lim.

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.[26]. Más precisamente, un autoencoder es una estructura que tiene como objetivo aprender niveles de representación de la información de entrada, para luego poder reconstruir una instancia similar descartando la información no deseada o considerada ruido". En este caso, la señal no deseada corresponde a la reverberación. El esquema básico del algoritmo se puede observar en la figura 14, donde la variable x representa a las variables de entrada, e y representa la variable de salida.

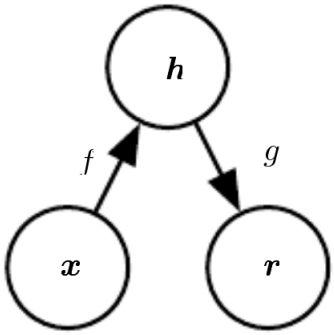


Figura 14: Estructura general de un autoencoder.

El esquema se compone de tres partes fundamentales:

- Una función de codificación f en donde las dimensiones de la variable de entrada se comprimen y las características más relevantes son aprendidas. Esta función realiza el mapeo de la variable de entrada al espacio latente.
- Un espacio latente h (o espacio de representación), en donde se concentran las representaciones internas aprendidas a partir de la compresión de la variable de entrada.
- Una función de decodificación en donde se aplica el proceso inverso que en la codificación, expandiendo las dimensiones tomadas del espacio latente para formar una representación que minimice el error de reconstrucción.

El sistema propuesto consiste en la estimación del espectro de la señal anecoica a partir del espectro de la señal reverberada. Para conseguirlo, en lugar de hacer un mapeo directo entre ambos espectros, se opta por estimar una máscara de amplitud. Se decidió trabajar con máscaras ya que estudios previos demostraron que con este método se obtienen mejores resultados que realizando estimaciones de mapeos directos entre dos espectros. Para trabajar con espectros, las señales de entradas se transforman al dominio temporal-frecuencial a partir de la transformada de Fourier de corto término. Se utiliza una ventana temporal de 512 muestras, con un solapamiento del 75%. La estructura de red neuronal utilizada consiste en una U-NET con conexiones de saltos, inspirada inicialmente en [26]. Este tipo de estructuras 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 la capa convolutiva. Un esquema básico de está estructura se puede ver en la figura 15, en donde se puede ver que las dimensiones de las capas siguen una forma de 'U', lo cual le da el nombre a estas estructuras.

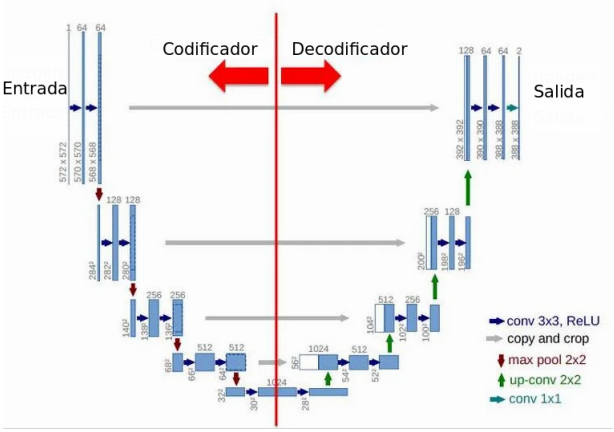


Figura 15: Esquema básico de una red tipo 'U-NET'.

A medida que se avanza en el modelo, el tamaño del espectro de entrada va disminuyendo y la cantidad de filtros utilizados va aumentando. Esta primera parte se puede pensar como un codificador, ya que el sistema esta tomando información del espectro de manera jerárquica. Este proceso se realiza sucesivamente hasta que la imagen de entrada alcanza una dimensión de 1x1. Luego, prosigue una etapa de decodificación en la cual se aplica el proceso inverso. Esto es, la dimensión del espectro se va aumentando con capas convolutivas transpuestas de

saltos mayores a 1, y la cantidad de filtros utilizados va disminuyendo. Esto se repite hasta que las dimensiones del espectro 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 pixel de la imagen resultante esté condicionado por todos los pixeles que componen la imagen de entrada. Se puede decir entonces que la estimación de cada punto del espectro final depende de todo el espectro de entrada, y no solo de una región determinada.

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 una capa del decodificador. Para poder hacerlo, la dimensión de concatenación (en este caso, las dimensiones del espectrograma) deben ser las mismas. De esta manera se logran decodificaciones más precisas.

Una representación gráfica del modelo final implementado se puede apreciar en la figura 16. En cada capa se indican tres valores, donde el primero representa la dimension temporal, el segundo la dimension frecuencial y el tercero el numero de canales. En las primeras capas, las dimensiones se reducen a la mitad en cada instancia debido al uso de un desplazamiento de paso 2 en los filtros convolucionales, lo que realiza la compresión de la información. En las capas subsiguientes, las dimensiones sufren el efecto contrario hasta volver a obtener las dimensiones originales. Este tipo de estructura tiende a perder información importante de bajo nivel durante el proceso de compresión. Como por lo general las variables de entrada y de salida comparten información estructural, se puede mejorar el funcionamiento de estas estructuras implementando conexiones entre las capas del codificador y el decodificador. Esto quiere decir que los mapas de características de las capas que conforman el codificador se van a concatenar directamente con los mapas de características en el decodificador, es decir, la salida de la capa i se concatena con la salida de la capa $N - i$ donde N es el número de capas. Estos saltos evitan que las activaciones pasen por el cuello de botella permitiendo la propagación de esta información estructural que estas estructuras tienden a perder.

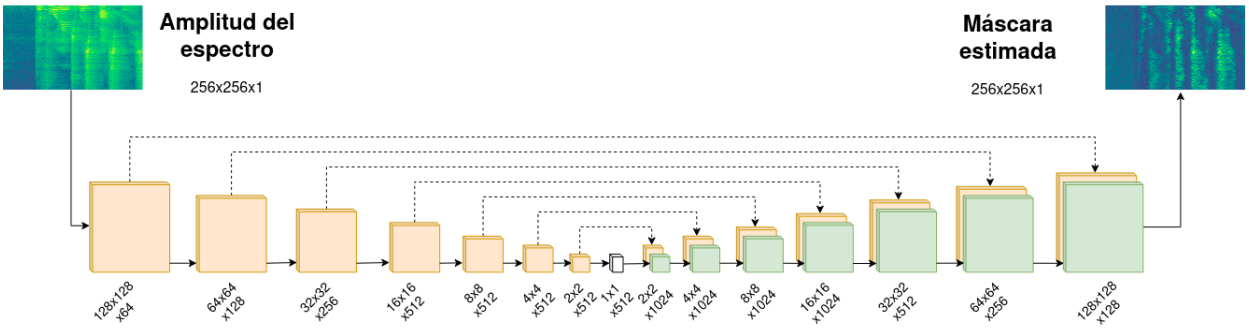


Figura 16: Modelo de red neuronal convolucional implementado

De esta forma, el modelo realiza una compresión de los datos de entrada, buscando minimizar el error en la reconstrucción al comparar con las instancias que se le presentan como objetivo. Para este caso particular, las entradas del modelo son espectrogramas, es decir, valores en el dominio tiempo-frecuencia. Las dimensiones del espectrograma de entrada son comprimidas hasta llegar a dimensiones de $1x1$ y luego son expandidas nuevamente, lo que produce un aumento de los campos perceptivos que permite propagar información global tanto en tiempo como en frecuencia. Esto significa que el cómputo de cada pixel de salida se verá influenciado por la totalidad del espectrograma de entrada. Estudios previos demostraron que la estimación directa de espectros, o lo que es equivalente, el mapeo frecuencial-temporal genera resultados menos precisos comparado con aquellos modelos que en lugar de estimar espectros estiman máscaras espectrales[49]. Por esto, las entradas del modelo son espectrogramas correspondientes a señales con reverberación, y las salidas u objetivos son máscaras de amplitud previamente calculadas en una etapa de preprocesamiento.

4.5. Especificaciones de la arquitectura implementada

Como se indicó anteriormente, esta arquitectura se basa en el entrenamiento de un modelo completamente convolucional. Sin embargo, la etapa de codificación y decodificación requieren evaluar ciertos detalles a la hora de su implementación.

En la etapa de codificación, la primera capa consiste en una capa convolucional utilizando una activación del tipo Leaky-Relu con una pendiente de $0,2$. Se escoge esta activación por sobre la rectificación lineal debido a que es favorable frente al problema de desvanecimiento de gradiente, lo cual puede ser un problema al trabajar con una arquitectura de red tan profunda.

Luego, las seis capas subsiguientes son también capas convolucionales con la misma función de activación pero con el agregado de que implementan normalización por lotes. Finalmente, la última capa de esta etapa es convolucional con normalización por lotes y función de activación ReLU.

En la etapa de decodificación es importante tener en cuenta que el tamaño de los campos perceptivos de las capas deben ser un múltiplos enteros del tamaño de salto para que no se produzcan artefactos indeseados durante el proceso inverso al cuello de botella que ocurre al utilizar capas deconvolutivas con tamaño de salto mayor a 1. Sin embargo, aun teniendo esto en consideración, las capas de deconvolución pueden generar artefactos indeseados. En lugar de utilizar capas convolutivas 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 mas cercanos, y luego una capa convolutiva. De esta manera se obtiene el efecto del aumento de dimensiones junto con el análisis convolutivo. Esta deconvolución se combina con un drop-out del 50 % y una función de activación ReLU en las primeras tres capas del decodificador. Luego, continuan 4 capas idénticas pero omitiendo el drop-out. Finalmente, la última capa del decodificador que a la vez es la capa de salida de la red consiste también en una deconvolución sin drop-out pero utilizando una función de activación tangencial, lo que significa que los valores de salida estarán acotados en el intervalo $[-1, 1]$. En todas las capas convolucionales y deconvolucionales 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 16 y para la optimización se utilizó el algoritmo de estimación adaptativa de momento (ADAM) [50] con un valor de tasa de aprendizaje de $0,001$.

$$L_{MSE} = \sum_{i=1}^{N-1} \overline{(M_i(t,f) - \hat{M}_i(t,f))^2} \tag{16}$$

4.6. Manejo de datos a evaluar

En este trabajo se ponen a prueba cuestiones relativas al manejo de datos utilizados para entrenar el algoritmo de aprendizaje profundo. Las pruebas se diferencian entre combinaciones

Tabla 1: Conformación de los distintos conjuntos de datos utilizados.

	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Conjunto de validación	RI simuladas	RI reales	RI simuladas	RI reales
Conjunto de prueba	RI simuladas	RI reales	RI reales	RI simuladas

entre datos simulados o reales, y en el ordenamiento de la complejidad de los datos durante el entrenamiento.

En primer lugar, como se cuenta con respuestas al impulso reales y simuladas, se prueban combinaciones de estos datos a la hora de formar los diferentes conjuntos requeridos para el entrenamiento (conjunto de entrenamiento y conjunto de prueba). A la hora de formar los conjuntos, se debe poder conseguir una determinada dispersión de ciertos descriptores acústicos para poder asegurar que el conjunto es representativo de las distintas variantes posibles en el efecto de la reverberación. Para analizar esta cuestión se decide trabajar con los parámetros tiempo de reververación medio TR_{mid} y relación directo-reverberado DRR. Estos descriptores son fácilmente controlables en las respuestas al impulso simuladas, debido a que se pueden controlar los parámetros en el algoritmo que las genera. Sin embargo, obtener esta heterogeneidad puede ser un problema al tratarse de respuestas al impulso reales debido a la escasez de bases de datos extensas. Para resolver esta cuestión se aplican procesamientos propuestos en [51] para lograr una aumentación de datos y poder formar un conjunto representativo de respuestas al impulso reales. Teniendo en cuenta esto, se definen como se va a conformar los conjuntos de datos de entrenamiento y de prueba para cada evaluación. Cabe aclarar que el conjunto de validación se conforma tomando una porción del conjunto de entrenamiento. Entonces, las pruebas a realizar quedan definidas por las distintas combinaciones entre datos simulados y reales y los diferentes conjuntos de datos como se muestra en la tabla **??**.

Para formar el conjunto de respuestas al impulso simuladas se utilizan 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, resultando en un total de 90 respuestas al impulso con tiempos de reverberación de entre aproximadamente $0{,}5$ segundos a 1 segundos.

Por otro lado, también es interesante evaluar la influencia del orden con el que las instancias de entrenamiento se le presentan a la red, siguiendo los lineamientos de la técnica denominada

curriculum learning. En este caso se considera que una instancia es más compleja cuando posee un mayor tiempo de dereverberación y a su vez una relación directo-reverberado baja debido a que la distorsión que sufre la señal sin reverberación es mayor, y es conocida la correlación que tienen estos parámetros con la disminución de la inteligibilidad. Para evaluar esto se utiliza conjuntos de datos simulados en la etapa de entrenamiento ya que estos permiten tener un mayor control sobre los parámetros acústicos. Entonces, se decide comparar el rendimiento del sistema al recibir instancias con tiempos de reverberación aleatoriamente seleccionados contra el rendimiento obtenido al ordenar las instancias de entrenamiento de menos a más complejas, es decir, tiempos de reverberación de menores a mayores, siempre partiendo del mismo conjunto de datos.

5. Resultados

5.1. Influencia del armado de datos

5.1.1. Variables del sistema

6. Discusión de los resultados

7. Conclusiones

8. Lineas futuras de investigación

Bibliografía

[1] L. Deng, G. Hinton y B. Kingsbury. “New types of deep neural network learning for speech recognition and related applications: An overview”. En: *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process (ICASSP)* (2013).

[2] H. Chung y col. “Noise-adaptive deep neural network for single-channel speech enhancement”. En: *Proc. Int. Workshop on Machine Learning for Signal Process. (MLSP)* (2018).

[3] Pearson J. y col. “Robust distant-talking speech recognition”. En: *IEEE International Conference on. Vol. 1* (1996). Acoustics, Speech, and Signal Processing.

[4] Castellano Pierre J., S. Sradharan y David Cole. “Speaker recognition in reverberant enclosures”. En: *IEEE International Conference on. Vol. 1* (1996). Acoustics, Speech, and Signal Processing.

[5] DiBiase Joseph H., Harvey F. Silverman y Michael S. Brandstein. “Robust localization in reverberant rooms”. En: *Springer Berlin Heidelberg* (2001). Microphone Arrays.

[6] Masato Miyoshi y Yutaka Kaneda. “Inverse Filtering of Room Acoustics”. En: *IEEE Transactions on Acoustics, Speech and Signal Processing* (1988).

[7] Stephen T. Neely y Jont B. Allen. “Invertibility of a room impulse response”. En: *Acoustics Research Department, Bell Laboratories, Murray Hill, New Jerse* (1997).

[8] Laurence R. Rabiner y W. Schafer Ronald. *Digital Speech Processing*. Prentice- Hall, 1975.

[9] Yegnanarayana B. y Satyanarayana P. “Enhancement of Reverberant Speech Using LP Residual Signal”. En: *IEEE Transactions on Speech and Audio Processing* (2000).

[10] Gannot S. y Moonen M. “Subspace Methods for Multimicrophone Speech Dereverberation”. En: *EURASIP journal on advances in signal processing* (2003).

[11] N. Roman y D. L. Wang. “Pitch-based monaural segregation of reverberant speech”. En: *Journal of Acoustical Society of America* (2006).

[12] M. Avendano y H. Hermansky. “Study on the dereverberation of speech based on temporal envelope filtering”. En: *Proc. of ICSLP* (1996).

[13] K. Lebart y J. M. Boucher. “A New Method Based on Spectral Subtraction for SpeechDereverberation”. En: *Acta Acustica united with Acustica* (2001).

[14] K. Lebart, J.-M. Boucher y P. Denbigh. “A new method based on spectral subtraction for speech dereverberation”. En: *Acta Acustica united with Acustica* (2001).

[15] M. Wu y D. L. Wang. “A two-stage algorithm for one-microphone reverberant speech enhancement”. En: *IEEE Transactions on Acoustics, Speech and Signal Processing* (2006).

[16] D. L. Wang y G. J. Brown. “Computational Auditory Scene Analysis: Principles, Algorithms, and Applications”. En: *Proc. of ICSLP* (1996).

[17] A. S. Bregman. “Auditory Scene Analysis”. En: *Cambridge, MA: MIT Press* (1990).

[18] D. Wang. *On ideal binary mask as the computational goal of auditory scene analysis*. Kluwer, 2005, págs. 181-197.

[19] Nicoleta Roman y John Woodruff. “Intelligibility of reverberant noisy speech with ideal binary masking”. En: *Journal of Acoustical Society of America* (2011).

[20] Nicoleta Roman y John Woodruff. “Speech intelligibility in reverberation with ideal binary masking: Effects of early reflections and signal-to-noise ratio threshold”. En: *Journal of Acoustical Society of America* (2013).

[21] O. Hazrati, J. Lee y P. C. Loizou. “Blind binary masking for reverberation suppression in cochlear implants”. En: *Journal of Acoustical Society of America* (2013).

[22] Z. Jin y D. L. Wang. “A supervised learning approach to monaural segregation of reverberant speech”. En: *IEEE Transactions on Acoustics, Speech and Signal Processing* (2009).

[23] Kun Han, Yuxuan Wang y DeLiang Wang. “Learning spectral mapping for speech dereverberation”. En: *IEEE International Conference on Acoustic, Speech and Signal Processing* (2014).

[24] F. Weninger y col. “Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition”. En: *IEEE International Conference on Acoustics, Speech and Signal Processing* (2014).

[25] D. S. Wang, Y. X. Zou y W. Shi. “A Deep Convolutional Encoder-Decoder Model for Robust Speech Dereverberation”. En: *22nd International Conference on Digital Signal Processing* (2017).

[26] Ori Ernst y col. “Speech Dereverberation Using Fully Convolutional Networks”. En: *22nd International Conference on Digital Signal Processing* (2017).

[27] Chunlei Liu, Longbiao Wang y Jianwu Dang. “Deep Learning-Based Amplitude Fusion for Speech Dereverberation”. En: *Discrete Dynamics in Nature and Society* (2020).

[28] Joshua D. Reiss y Andrew P. McPherson. *Audio Effects: Theory, Implementation and Application*. CRC Press, 2014.

[29] Alan V. Oppenheim, Ronald W. Shafer y John R. Buck. *Discrete-Time Signal Processing*. Prentice- Hall, 1989.

[30] E. Oran Brigham. *The Fast Fourier Transform And Its Aplications*. Prentice- Hall, 1988.

[31] Patrick A. Naylor y Nikolay D. Gaubitch. *Speech Dereverberation - Signals and Communication Technology*. Springer, 2010.

[32] Tiago H. Falk y Wai-Yip Chan. “A Non-intrusive Quality Measure of Dereverberated Speech”. En: *Department of Electrical and Computer Engineering* (2008). Queen’s University, Kingston, Ontario, Canada.

[33] Tiago H. Falk y Wai-Yip Chan. “A Non-intrusive Quality Measure of Dereverberated Speech”. En: *Department of Electrical and Computer Engineering* (2008). Queen’s University, Kingston, Ontario, Canada.

[34] Emmanuel Vincent, Rémi Gribonval y Cédric Févotte. “Performance Measurement in Blind Audio Source Separation”. En: *IEEE Transactions on Audio, Speech and Language Processing* (2006).

[35] Chunlei Liu, Longbiao Wang y Jianwu Dang. “A Logical Calculus of Ideas Immanent in Nervous Activity”. En: *Bulletin of Mathematical Biophysics* (1943).

[36] Frank Rosenblatt. “The Perceptron: A Perceiving and Recognizing Automaton”. En: *Cornell Aeronautical Laboratory* (1957).

[37] Francois Chollet. *Deep Learning with Python*. Manning Publications, 2017.

[38] Y. Bengio y col. “Curriculum Learning”. En: *Proceedings of the 26thInternational Conference on Machine Learning* (2009).

[39] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O’reilly media, 2019.

[40] D. L. Wang y J. S. Lim. “The unimportance of phase in speechenhancement”. En: *IEEE Trans. Acoust. Speech Signal Process* (1982).

[41] Y. Ephraim y D. Malah. “Speech enhancement using a minimummean-square error short-time spectral amplitude estimator”. En: *IEEE Trans. Acoust. Speech Signal Process* (1984).

[42] Daniel W. Griffin y Jaa S. Lim. “Signal Estimation for Modified Short-Time Fourier Transform”. En: *IEEE Trans. Acoust. Speech Signal Process* (1984).

[43] R. Stewart y M. Sander. “Database of omnidirectional and B-format impulse responses”. En: *Proc. IEEE Int. Conf. Acous-tics, Speech, and Signal Process. (ICASSP)* (2010).

[44] Farina Angelo. “Simultaneous measurement of impulse response and distortion with a swept sine technique”. En: *108th AES Convention* (2000).

[45] R. Scheibler, E. Bezzam e I. Dokmanić. “Pyroomacoustics: A Python package for audio room simulations and array processing algorithms”. En: *Proc. IEEE ICASSP* (2018).

[46] J. B. Allen y D. A. Berkley. “Image method for efficiently simulating small-room acoustics”. En: *J. Acoust. Soc. Am., vol. 65* (1979).

[47] Nicholas J. Bryan. “Impulse response data augmentation and deep neural networks for blind room acoustic parameter estimation”. En: *Adobe Research* (2019). San Francisco, CA, USA.

[48] Panayotov V. y col. “Libris-peech: an asr corpus based on public domain audio books”. En: *In Acoustics,Speech and Signal Processing (ICASSP)* (2015).

[49] Yuxuan Wang, Arun Narayanan y De Liang Wang. “On Training Targets for Supervised Speech Separation”. En: *IEEE/ACM Trans Audio Speech Lang Process* (2014).

[50] Kingma D. P. y Ba J. “Adam: A Method for Stochastic Optimization”. En: *ICLR* (2014).

[51] Nicholas J. Bryan. “Impulse response data augmentation and deep neural networks for blind eoom acoustic parameter estimation”. En: *Adobe Research* (2019).