UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – PROCESAMIENTO DIGITAL DE SEÑALES

SEPARACIÓN MONOAURAL DE SEÑALES DE VOZ EMPLEANDO APRENDIZAJE PROFUNDO

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
MAURICIO MICHEL OLVERA ZAMBRANO

TUTOR
M.I. LARRY HIPÓLITO ESCOBAR SALGUERO
FACULTAD DE INGENIERÍA

CIUDAD UNIVERSITARIA, CDMX, SEPTIEMBRE 2019

JURADO ASIGNADO:

Presidente:	Dr. Savage Carmona Jesús
Secretario:	Dr. Pérez Alcázar Pablo Roberto
Primer vocal:	M.I. Escobar Salguero Larry
Segundo vocal:	Dra. Medina Gómez Lucía
Tercer vocal:	Dr. Rivera Rivera Carlos

Lugar donde se realizó la tesis:

Laboratorio de Procesamiento Digital de Señales,
Edificio T “Bernardo Quintana”, segundo piso.
Posgrado de Ingeniería, UNAM.

TUTOR DE TESIS:

M.I. LARRY HIPÓLITO ESCOBAR SALGUERO

_____

FIRMA

Separación Monoaural de Señales de Voz Empleando Aprendizaje Profundo

Mauricio Michel Olvera Zambrano

19 de agosto de 2019

Resumen

En este trabajo de tesis se presenta el diseño e implementación de un sistema digital capaz de separar las voces simultáneas de dos hablantes contenidas en una mezcla de audio monoaural. El sistema implementado incorpora un método novedoso de aprendizaje profundo denominado *Deep Clustering*, el cual produce una representación multidimensional del espectrograma de la mezcla de audio para separar linealmente las unidades tiempo-frecuencia y generar máscaras binarias tal que aplicadas al espectrograma de la mezcla se obtenga una estimación del espectrograma de cada uno de los hablantes. La calidad de la separación del sistema es evaluada en términos de las relaciones señal a distorsión, señal a artefactos y señal a interferencias y se compararon los resultados con un sistema de referencia de separación de hablantes basado en la aplicación de máscaras binarias ideales.

Abstract

In this thesis work the design and implementation of a digital system capable of separating the simultaneous voices of two speakers within a monaural audio mixture is presented. The implemented system is based on a novel deep learning method called *Deep Clustering*, which produces a multidimensional representation of the spectrogram to linearly separate the time-frequency bins and generate binary masks such that, when applied to the spectrogram of the input mixture, an estimatate of the spectrogram of each of the speakers is obtained. The quality of the speaker separation system is evaluated in terms of the signal to distortion, signal to artifacts and signal to interferences ratios and the results are compared to a speaker separation baseline system based on the application of ideal binary masks.

Contenido

Tablas

Figuras

1

Introducción

Actualmente, en espacios acústicos se tiene especial interés en la separación de señales provenientes de diferentes fuentes que se encuentran altamente correlacionadas entre sí, tal como en el *efecto fiesta de cóctel*, donde existen señales de voz de distintas personas, las cuales se encuentran hablando al mismo tiempo. En este caso, el sistema de audición humana es capaz de focalizar su atención únicamente en la voz del hablante de su interés, al mismo tiempo que ignora el resto de las voces interferentes [1]. Los métodos computacionales que intentan replicar esta habilidad humana empleando información de un solo canal de audio se enfrentan a dos dificultades principalmente: la permutación de hablantes, la cual ocurre en instantes de tiempo en que varias personas hablan simultáneamente; y el número de hablantes en la mezcla es generalmente desconocido. Sin embargo, gracias a la inclusión de métodos de separación monoaural de señales de voz basados en algoritmos de aprendizaje profundo se han podido obtener resultados de vanguardia que superan las dificultades anteriores.

1.1. Objetivo

Diseñar e implementar a partir de la teoría del procesamiento digital de señales y la teoría del aprendizaje profundo, un sistema capaz de separar las señales de voz simultáneas de dos hablantes presentes en una mezcla de audio monoaural.

1.2. Metodología

El sistema de separación de hablantes implementado en este trabajo de tesis está constituido por tres módulos. En el primer módulo una mezcla de audio con voces simultáneas que se traslapan es traslada al dominio tiempo-frecuencia (TF) mediante la Transformada de Fourier en Tiempo Corto (STFT). En el segundo módulo, una red neuronal recurrente bidireccional (Bi-LSTM) se encarga de producir una representación multidimensional linealmente separable del espectrograma de la mezcla. Mediante la aplicación del algoritmo de k-medias se agrupan las unidades TF pertenecientes a cada uno de los hablantes y se generan máscaras binarias que aplicadas al espectrograma de la mezcla permiten obtener una estimación de los espectrogramas de cada uno de los hablantes. Finalmente, el tercer módulo se encarga de convertir los espectrogramas estimados al dominio del tiempo mediante la Transformada de Fourier en Tiempo Corto Inversa (ISTFT).

1.3. Organización del trabajo

En el capítulo 2: *Técnicas de Separación de Voz*, se describen los trabajos previos en el área de separación de señales simultáneas de hablantes y los avances de vanguardia en torno a la resolución del problema de la fiesta de cóctel empleando aprendizaje profundo.

En el capítulo 3: *Procesamiento Digital de la Señal de Voz*, se aborda la teoría fundamental de procesamiento de digital de señales necesaria para describir los métodos computacionales más importantes para manipular señales de voz en los dominios del tiempo y frecuencia.

En el capítulo 4: *Redes Neuronales Recurrentes*, se presenta una breve introducción al aprendizaje profundo y se describen los conceptos matemáticos fundamentales de la arquitectura de las redes neuronales recurrentes y sus variantes para el modelado eficiente de la información contextual de secuencias de datos.

En el capítulo 5: *Diseño e Implementación del Sistema de Separación de Señales de Voz*, se describen detalladamente los módulos de procesamiento digital de señales y aprendizaje profundo que integran el sistema de separación de hablantes implementado.

En el capítulo 6: *Pruebas y Resultados*, se describen las distintas configuraciones del sistema de separación de hablantes implementado y se presentan los resultados de separación en términos de las métricas de separación ciega de fuentes `BSS_eval` y su comparación con un sistema de separación de hablantes ideal basado en máscaras binarias ideales.

Finalmente, en el capítulo 7: *Conclusiones y Trabajo Futuro*, se presentan las conclusiones de este trabajo de tesis con base en los resultados obtenidos en el capítulo 6 y se describen brevemente las posibles mejoras para incrementar la eficiencia del sistema de separación de señales de voz implementado.

2

Técnicas de Separación de Voz

El trabajo pionero en la separación de fuentes data de la década de 1970. Desde sus inicios, la investigación en esta área ha buscado separar señales de audio de interferencias con la mejor calidad posible. En la actualidad, el campo de la separación de fuentes es más amplio e involucra una gran variedad de escenarios. Específicamente, la investigación actual en la separación de señales de voz de múltiples hablantes consiste en replicar computacionalmente los procesos subyacentes al mecanismo de separación de voces llevado a cabo por los humanos. Con base en esta idea, históricamente se han propuesto técnicas tanto en el dominio del tiempo como en la frecuencia, siendo esto últimos los que hasta hace unos años habían obtenido un mejor desempeño, sin embargo, gracias a la gran cantidad de información y poder de cómputo actual, el éxito de la incorporación de métodos de inteligencia artificial en el área de procesamiento de voz, ha inspirado el desarrollo de métodos de aprendizaje profundo que ofrecen resultados sorprendentes que intentan resolver el problema de la fiesta de cóctel.

2.1. El problema del efecto fiesta de cóctel

El término *fiesta de cóctel* fue propuesto por Collin Cherry en su clásico artículo titulado *Some experiments on the recognition of speech, with one and with two ears* [2]. En este artículo estudió si los humanos pueden seleccionar una señal de voz sobre otra, si retienen algo acerca de la señal que no es de interés, y la manera en que focalizan su atención entre señales. Cerca de cuatro décadas después, Albert Bregman [3] fue pionero en estudiar cómo los humanos realizan la separación de sonidos en ambientes complejos, actividad que fue llamada *análisis de escenas auditivas*. De hecho, la mayoría del trabajo pasado y reciente en el problema de la fiesta de cóctel se enfoca en el primer problema, concerniente a la separación de sonidos.

El *efecto fiesta de cóctel* es un fenómeno que se refiere a la capacidad que posee el ser humano para reconocer y focalizar su atención auditiva en la voz de una persona específica cuando múltiples personas hablan simultáneamente y cuando otro ruido de fondo está involucrado en la conversación. Este fenómeno ha sido ampliamente observado a lo largo de muchas décadas y emularlo mediante una computadora habilita muchos escenarios y aplicaciones donde no se pueden ignorar las voces traslapadas presentes en una conversación [4].

Aunque el efecto fiesta de cóctel ha demostrado ser un problema difícil de resolver por las computadoras, para los humanos resulta una tarea trivial, pues somos capaces de segregar eficientemente los sonidos presentes en un ambiente ruidoso y asistir a reconocer fácilmente aquellos que son de nuestro interés. Por ejemplo, en una conversación dentro de una sala ruidosa, como lo es típicamente una fiesta de cóctel, los asistentes pueden concentrarse sin dificultad en la voz de las persona con la que se encuentra conversando, y pueden focalizar su atención de manera instantánea hacia otros sonidos, como puede ser la música de fondo, el ruido del exterior, o bien, otra conversación donde circunstancialmente han pronunciado su nombre. Esta habilidad, sin embargo, es compartida por el ser humano con algunas especies animales [5], quienes ante situaciones de amenaza y de peligro son capaces de identificar los sonidos procedentes ya sea de sus semejantes o enemigos en ambientes donde los animales vocalizan al mismo tiempo.

Para que una máquina iguale la capacidad que tienen los seres humanos para separar los sonidos presentes en una mezcla, esencialmente se necesita resolver dos problemas. En el marco de una escena auditiva compleja, como una fiesta de cóctel típica ilustrada en la figura 2.1, el primer problema consiste en encontrar una mecanismo para separar los sonidos de la mezcla, la cual se asume como la superposición de diversas fuentes de sonido. Comúnmente, el interés del ser humano se concentra en solamente una o dos fuentes al mismo tiempo, por lo que es fundamental separar estos sonidos de la mezcla. El segundo problema a tener en cuenta tiene que ver con la capacidad de seguir una fuente particular de sonido y mantener la atención entre fuentes. En la mayoría de los casos, estos dos problemas son dependientes uno del otro, pues por un lado la atención a la fuente de interés se beneficia por un buen mecanismo de separación, y la separación se beneficia por un buen mecanismo de seguimiento de la fuente.

Figura 2.1: Ejemplo de una escena auditiva compleja, tal como lo es una fiesta de cóctel típica. (*The Great Gatsby, Paramount, 2013*)

2.2. Técnicas convencionales de separación monoaural

Históricamente se han propuesto diferentes enfoques para resolver el problema del efecto fiesta de cóctel. Entre los más importantes se destacan los basados en técnicas de procesamiento de señales, en métodos de análisis computacional de escenas auditivas (CASA), en la factorización matricial no negativa (NMF), y también en técnicas de formador de haz empleando arreglos de micrófonos. No obstante, pocos de estos enfoques han logrado resultados robustos con una separación de alta fidelidad, especialmente cuando solamente un solo canal de la mezcla de señales se encuentra disponible o cuando los hablantes están orientados en la misma dirección.

2.2.1. Análisis computacional de escenas auditivas

En psicoacústica, el proceso perceptual de separar los sonidos presentes en una mezcla de señales de audio se conoce como *análisis de escenas auditivas* (ASA) [3]. La investigación en este campo motivó el *Análisis Computacional de Escenas Auditivas* (CASA),[6, 7, 8], que tiene como objetivo igualar mediante una computadora el desempeño humano entorno al análisis de escenas auditivas (ASA) empleando grabaciones de la escena acústica realizadas con uno o dos micrófonos [9]. Esta restricción hace a este método biológicamente relevante, ya que el ser humano proporciona información al sistema auditivo a partir de las señales acústicas recibidas por uno o dos oídos. Con base en esta idea, CASA busca estudiar la forma en que los humanos separan los sonidos e intenta aprender de ellos.

En CASA se tienen ciertas reglas de segmentación basadas en señas de agrupamiento perceptual, las cuales son comúnmente diseñadas a mano mediante observaciones humanas, para operar en características de bajo nivel con el objetivo de estimar una máscara de tiempo-frecuencia (TF) que aísle los componentes de la señal que pertenecen a diferentes hablantes, y luego poder

utilizar esta máscara para reconstruir la señal.

Aunque CASA simula la conducta de alto nivel de la escucha humana, tiene varias desventajas, las cuales se listan a continuación:

- Desde una perspectiva más amplia de separación de fuentes, donde no solamente la señal de voz está involucrada, sino también otras fuentes de audio, el método falla pues únicamente se concentra en la voz humana.
- La mayoría de las reglas de segmentación que utiliza CASA son diseñadas manualmente y están basadas en un número limitado de observaciones, por lo que su capacidad para generalizar es deficiente.
- Dado que la separación final está basada en la segmentación de unidades TF (i.e., cada unidad TF pertenece solamente a una fuente origen), el mejor resultado posible es el obtenido con una máscara binaria ideal (IBM), la cual es subóptima en la mayoría de los escenarios [7].
- Los sistemas basados en CASA dependen fuertemente de la eficiencia del seguidor de *pitch*, el cual no es robusto bajo condiciones acústicas complejas.
- Al no ser un enfoque basado en datos, está limitado al no poder emplear técnicas de aprendizaje automático.

No obstante, a pesar de que CASA fue propuesto hace más de una década, actualmente se siguen desarrollando técnicas basadas en los mismos principios de este enfoque.

2.2.2. Factorización matricial no negativa

Debido a la limitada capacidad de CASA para incorporar técnicas de aprendizaje automático, el método de *factorización matricial no negativa* (NMF) [10], junto con otros modelos basados en datos fueron propuestos para encontrar características complejas de las observaciones. NMF y otros métodos de descomposición matricial fueron construidos bajo la suposición de que el espectrograma de audio posee una estructura de rango bajo que puede ser representada a partir de un pequeño número de funciones base. Bajo ciertas condiciones, la descomposición en NMF es única y no se necesita otra suposición de independencia u ortogonalidad.

Específicamente en NMF,

$$\boldsymbol{Y} = \sum_{s} \boldsymbol{W}_s \boldsymbol{H}_s \tag{2.1}$$

donde cada fuente s es modelada por la aproximación de bajo rango de matrices no negativas $\boldsymbol{W}_s$ y $\boldsymbol{H}_s$ y, luego sumadas para formar la mezcla $\boldsymbol{Y}$. Debido a la no negatividad de la descomposición de matrices, no se presenta cancelación entre fuentes en la reconstrucción del espectro de la mezcla $\boldsymbol{Y}$, la cual modela la aditividad entre fuentes mezcladas.

En la etapa de entrenamiento, cada fuente limpia, e.g., voz, ruido, o música, se descompone y se mapea a una serie funciones base y activaciones, y se forma un diccionario específico $\boldsymbol{W}$ para cada fuente. Durante la etapa de prueba, todos los diccionarios aprendidos, se mezclan

para forma un diccionario único. Los valores de este diccionario combinado se mantienen fijos y solamente la activación $\boldsymbol{H}$ es optimizada para cada fuente, en cuyo caso la optimización es convexa y se puede alcanzar un valor óptimo global. Después, cada fuente en la mezcla es reconstruida a partir de las funciones base y las activaciones correspondientes. De forma general la descomposición matricial NMF puede definirse como

$$\min_{\boldsymbol{W},\boldsymbol{H}} D(\boldsymbol{Y}||\boldsymbol{W}\boldsymbol{H}), \qquad \boldsymbol{W},\boldsymbol{H} \geq 0 \tag{2.2}$$

Se han propuesto muchas variantes del método de descomposición matricial NMF, entre las cuales se incluyen: la descomposición NMF dispersa [11] que forza la activación $\boldsymbol{H}$ a ser dispersa; la descomposición NMF convolucional [12], donde el espectrograma es descompuesto en una convolución (en lugar de una multiplicación) entre las funciones base y la activación; la descomposición NMF robusta [13] que combina el método NMF con análisis de componentes principales robusto (RPCA).

Algunas de las desventajas del método de descomposición NMF se listan a continuación:

- El método está limitado por las funciones base y no se explotan todos los atributos y regularidades de la señal de voz, tales como las propiedades dinámicas de los sonidos del habla.
- La formulación lineal del modelo evita que tenga un desempeño con una calidad de separación alta.
- Su aplicación en escenarios de tiempo real está limitada debido a la complejidad de la descomposición matricial durante la etapa de prueba, la cual requiere de un alto costo computacional.
- Los parámetros del modelo están determinados por las fuentes que se tienen en el conjunto de entrenamiento y su tamaño incrementa linealmente con el número de fuentes.
- En la etapa de prueba, cada fuente de sonido debe tener un diccionario aprendido en la etapa de entrenamiento, lo cual no es factible en la mayoría de las aplicaciones del mundo real.

2.2.3. Modelos generativos

Los *modelos generativos* fueron propuestos para abordar la limitación de la descomposición NMF para modelar las características dinámicas de la señal de voz. La mayoría de estos métodos combinan *modelos de mezclas Gaussianas* y *modelos ocultos de Markov* (GMM-HMM), un modelo generativo popular en el reconocimiento voz. Entre los modelos de separación GMM-HMM, el modelo oculto de Markov factorial (FHMM) es el que tiene un mejor desempeño. Esta técnica modela cada fuente de sonido con un HMM. Para cada señal s, si se define la señal fuente (libre de ruido) como $\{\boldsymbol{x}_t^s\}$ $(t \in \{1, 2, ..., T\})$, los estados ocultos como $\{\boldsymbol{v}_t^s\}$, y el estado de la mezcla

discreta como $\{\boldsymbol{m}_t^s\}$, el modelo HMM tiene las características:

$$p(\boldsymbol{v}_t^s|\boldsymbol{v}_{1:t-1}^s) = p(\boldsymbol{v}_t^s|\boldsymbol{v}_{t-1}^s) \tag{2.3}$$

$$p(\boldsymbol{x}_t^s|\boldsymbol{v}_{1:T}^s) = p(\boldsymbol{x}_t^s|\boldsymbol{v}_t^s) \tag{2.4}$$

$$= \sum_{\boldsymbol{s}_t^s} p(\boldsymbol{x}_t^s|\boldsymbol{m}_t^s)p(\boldsymbol{m}_t^s|\boldsymbol{v}_t^s) \tag{2.5}$$

donde la ecuación 2.3 describe la probabilidad de transición y la ecuación 2.5 representa la probabilidad de observación bajo la suposición de independencia de Markov. Dada la mezcla de sonidos $\{\boldsymbol{y}_t\}$ compuesta de S señales fuente, el nuevo modelo generativo, llamado *modelo de interacción*, se define como

$$p(\{\boldsymbol{y}_t\},\{\boldsymbol{x}_t\},\{\boldsymbol{m}_t\},\{\boldsymbol{v}_t\}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\{\boldsymbol{x}_t\}) \prod_{t=1}^{T}\prod_{s=1}^{S} p(\boldsymbol{x}_t^s|\boldsymbol{m}_t^s)p(\boldsymbol{m}_t^s|\boldsymbol{v}_t^s)p(\boldsymbol{v}_t^s|\boldsymbol{v}_{t-1}^s) \tag{2.6}$$

donde $\{\boldsymbol{x}_t^s\}$ no es observable.

Después se busca inferir la secuencia de estado oculta $\{\hat{\boldsymbol{v}}_t^{(s)}\}$ para cada fuente s usando el criterio máximo a posteriori (MAP), para lo cual se requiere calcular $p(\boldsymbol{y}_t|\{\boldsymbol{v}_{(t)}^s\})$ como

$$p(\boldsymbol{y}_t|\{\boldsymbol{v}_{(t)}^s\}) = \sum_{\boldsymbol{m}_t^1,\boldsymbol{m}_t^2,\ldots,\boldsymbol{m}_t^S} p(\boldsymbol{y}_t|\{\boldsymbol{m}_{(t)}^i\}) \prod_s p(\boldsymbol{m}_t^s|\boldsymbol{v}_t^s) \tag{2.7}$$

$$= \sum_{\{\boldsymbol{m}_{(t)}^i\}} p(\boldsymbol{y}_t|\{\boldsymbol{m}_{(t)}^i\}) \prod_s p(\boldsymbol{m}_t^s|\boldsymbol{v}_t^s) \tag{2.8}$$

donde

$$p(\boldsymbol{y}_t|\{\boldsymbol{m}_{(t)}^s\}) = \int\int\cdots\int p(\boldsymbol{y}_t,\{\boldsymbol{x}_{(t)}^s\}|\{\boldsymbol{m}_{(t)}^s\})\mathrm{d}\boldsymbol{x}_t^1\mathrm{d}\boldsymbol{x}_t^2\ldots\mathrm{d}\boldsymbol{x}_t^S \tag{2.9}$$

El proceso de cálculo es muy complicado debido a que todas las estimaciones están acopladas sobre los estados de los hablantes, por lo que se han desarrollado varios métodos para aproximar la función de interacción y poder resolver la integral de la ecuación 2.9 analíticamente. Este cálculo puede dividirse en dos partes: en la primera se calculan las probabilidades de estado acústico $p(\boldsymbol{y}_t|\{\boldsymbol{m}_{(t)}^s\})$ y en la segunda se combinan estas probabilidades para inferir la configuración de las variables de estado dinámico $\{\hat{\boldsymbol{v}}_t^s\}$.

La tabla 2.1 compara el modelo generativo FHMM con otras técnicas convencionales para la tarea separación y reconocimiento de las voces simultáneas de dos hablantes grabadas con un solo micrófono llevado a cabo en el *Speech Separation and Recognition Challenge* (SSC) de 2010. Todos los modelos generativos (ALGONQUIN y Max-model) superaron los modelos CASA y NMF e incluso tuvieron un mejor desempeño que los humanos en esta tarea [14].

Aunque el desempeño de los modelos generativos es mejor con respecto a otras técnicas convencionales de separación, el método tiene las siguientes limitantes:

- El costo computacional es muy alto durante la etapa de inferencia, incluso con los métodos analíticos que aproximan la función de interacción.

Método	WER (%)
ALGONQUIN [15]	**22.7**
Max-model [15]	**23.7**
Humanos [14]	28.5
PMC iterative Viterbi [16]	35.1
CASA [17]	38.2
NMF [11]	44.2

Tabla 2.1: Comparación de la tasa de error de reconocimiento de palabras (WER) de métodos de convencionales de separación y reconocimiento de voces simultáneas de dos hablantes en el *Speech Separation and Recognition Challenge* (SSC) de 2006. [14].

- A medida que el número de hablantes en la mezcla se incrementa, el modelo se vuelve exponencialmente más complejo.
- El modelo no maneja eficientemente a los hablantes y ambientes acústicos desconocidos, ya que su información debe ser descrita previamente con un modelo HMM.

2.2.4. Otras técnicas de separación de fuentes

Entre los métodos de separación de fuentes que emplean técnicas de procesamiento de señales se destacan principalmente los basados en filtrado espacial y en el análisis de componentes independiente. Otra técnica de separación muy popular, pero con un objetivo más específico es el mejoramiento de la señal de voz. A continuación se describen cada una de estas técnicas.

2.2.4.1. Formador de haz

Los métodos de separación que emplean formadores de haz logran la separación de fuentes empleando el principio de filtrado espacial. El filtrado espacial tiene como objetivo potenciar la señal procedente de una dirección específica mediante la configuración apropiada de un arreglo de micrófonos, y al mismo tiempo atenuar las señales interferentes procedentes de otras direcciones [1].

El formador de haz más simple es el formador de haz de retraso y suma (también conocido como formador de haz de Barlett), el cual retrasa las señales del arreglo de micrófonos de tal manera que al sumarlas, las señales procedentes de la dirección deseada se encuentren en fase y se realice una suma constructiva que potencie la señal de interés, mientras que las diferencias de fase atenúen las señales procedentes de otras direcciones. A medida que el número de micrófonos en el arreglo aumenta, la atenuación de las señales interferentes es mayor.

Otro tipo de formador de haz, es el formador de haz adaptable, el cual atenúa las señales de interferencia a través de una serie de pesos adaptables calculados mediante un proceso de entrenamiento. En general, un formador de haz adaptable con L micrófonos puede cancelar únicamente $L - 1$ señales de interferencia diferentes. Sin embargo, esta limitante es resuelta por versiones sub-banda del formador de haz adaptable, que permite cancelar más fuentes de ruido,

cuyo espectro no se traslapa sustancialmente.

Algunas de las ventajas de los formadores de haz son las siguientes:

- Producen una separación de alta fidelidad cuando el arreglo de micrófonos utilizado está configurado adecuadamente.
- Poseen la habilidad de atenuar la reverberación, debido a que la señal de interés procede de una dirección específica, mientras que las señales reverberantes arriban desde diferentes direcciones.

Por otra parte algunas de las desventajas de esta técnica son las siguientes:

- Poseen una configuración de estacionaridad, por lo que es difícil separar ya sea una señal de interés en movimiento o una que alterna entre diferentes fuentes de sonido.
- La separación de la señal de interés de una mezcla no es posible cuando múltiples sonidos provienen de una misma dirección o se generan en locaciones cercanas.

2.2.4.2. Análisis de componentes independientes

La *separación ciega de fuentes* es un enfoque de separación que emplea *análisis de componentes independientes* (ICA). Este método combina filtrado adaptable y técnicas de aprendizaje automático. Al igual que en el formador de haz, una mezcla es modelada como una superposición lineal de fuentes de sonido, es decir, se asume un modelo de la forma $x(t) = As(t)$, donde $s(t)$ es un vector de señales fuente desconocidas, A es la matriz de mezcla, y $x(t)$ es un vector que contiene las señales mezcladas grabadas por múltiples micrófonos. El empleo de ICA supone que las fuentes de sonido son estadísticamente independientes, por lo que el problema de separación consiste en estimar los valores de una matriz que realice la separación, i.e., la matriz inversa de A. Esta formulación requiere de una serie de suposiciones acerca del proceso de mezclado y del número de micrófonos utilizados. Cuando estas suposiciones son ciertas, la separación de señales se realiza con una alta fidelidad. Algunas limitantes de este método son las siguientes:

- La matriz de mezcla A debe ser estacionaria por un periodo determinado de tiempo para realizar la estimación de un gran número de parámetros, lo cual es muy difícil de satisfacer en situaciones donde las fuentes de sonido están en movimiento.
- Al igual que en el filtrado espacial, las fuentes de sonido deben originarse en direcciones espaciales diferentes.

2.2.4.3. Mejoramiento de la señal de voz

El *mejoramiento de la señal de voz* es un área con diversos métodos de separación donde específicamente las fuentes de sonidos son señales de voz y ruido, por lo que su objetivo es mejorar la inteligibilidad de la voz en presencia de ruido. Los métodos de mejoramiento de la señal de voz son aplicados generalmente a mezclas grabadas con un solo micrófono y de manera general se busca estimar la señal de voz a partir de la mezcla ruidosa mediante un análisis estadístico

de ambas señales. Entre los métodos más populares se encuentran: el método de *sustracción espectral* y la *estimación del error cuadrático medio*. El primero consiste en sustraer del espectro de la mezcla la densidad espectral de potencia de la señal de ruido estimada [18], mientras que el segundo modela el espectro de la señal de voz y el ruido como dos variables aleatorias Gaussianas estadísticamente independientes, y estima óptimamente la señal de voz libre de ruido de acuerdo al criterio del error cuadrático medio mínimo (MSEE).

Las desventajas de los métodos de mejoramiento de la señal de voz son las siguientes:

- Se requiere de un buen estimador de la interferencia, lo cual es muy difícil de conseguir a menos que la interferencia sea estacionaria. Esta limitante comúnmente se ameniza empleando algoritmos de detección de silencios y una estimación subsecuente del ruido presente en esos intervalos.

- El modelo no puede tratar con la perspectiva de ASA, en la cual una mezcla es una escena auditiva en la que convive una amplia gama de sonidos, los cuales aparecen y desaparecen de manera impredecible, ya que en el modelo una mezcla se representa únicamente como la suma de voz y ruido.

2.3. Técnicas de separación basadas en aprendizaje profundo

Gracias al éxito de los métodos de aprendizaje profundo en el área del reconocimiento de voz, estos métodos se han introducido en la separación de fuentes para resolver el problema de la fiesta de cóctel. La mayoría de las investigaciones en esta área se han llevado a cabo en tareas de separación monoaural de voz.

Los enfoques de separación de fuentes basados en arquitecturas de aprendizaje profundo son capaces de descubrir las estructuras ocultas y características en diferentes niveles de abstracción de los datos de las fuentes de sonido. En contraste con los métodos convencionales, estos métodos son rápidos y eficientes en la tarea específica de separación de voz.

No obstante, la desventaja principal de estas técnicas novedosas, es que se enfocan en condiciones ideales, libres de ruido para las etapas de entrenamiento y prueba, por lo que para cualquier aplicación práctica, con ruido de fondo debido a fuentes de sonido interferentes o micrófonos no ideales, los resultados deben ser esperados, por lo que todavía está por verse cuál es el desempeño de estas técnicas, bajo condiciones ruidosas que reflejen un escenario de uso real. A continuación se describen brevemente tres métodos novedosos para la separación de voz de múltiples hablantes.

2.3.1. Deep Clustering

La técnica de *Deep Clustering* (DPCL) [19], la cual es estudiada más a detalle en el capítulo 5 por ser el método implementado en esta tesis, supone que cada unidad TF del espectrograma de una mezcla de voces pertenece a un hablante. Durante el entrenamiento del sistema, cada unidad tiempo-frecuencia es convertida en un vector multidimensional, denominado *embedding* que representa nuevas características de la señal. Los nuevos vectores de esta representación son optimizados de tal manera que en el espacio multidimensional las unidades TF que pertenecen

a un hablante se encuentran juntas unas de otras y las que son de diferentes hablantes se encuentran alejadas. Durante la etapa de inferencia, un algoritmo de agrupamiento es aplicado en el espacio de los *embeddings* para generar particiones de las unidades TF y construir máscaras binarias que permitan estimar los espectrogramas de los hablantes presentes en la mezcla de audio.

2.3.2. Deep Attractor Network

La técnica de *Deep Attractor Network* (DANet) [20], al igual que DPCL, crea un espacio con nuevas características a partir de el espectro de las señales de voz, pero a diferencia de DPCL, crea grupos centrales en el espacio de los *embeddings*, llamados *atractores*, que como su nombre lo indica, buscan atraer a las unidades TF que pertenecen a cada hablante. La principal limitante de DANet es la estimación de los *atractores* durante la etapa de inferencia.

2.3.3. Permutation Invariant Training

La técnica *Permutation Invariant Training* (PIT) [21], entrena un modelo de aprendizaje profundo para estimar máscaras, cada una de las cuales es aplicada a la mezcla de voces para producir una estimación de los espectrogramas de los hablantes. En esta técnica los hablantes que constituyen una mezcla de audio son tratados como un conjunto. Durante el entrenamiento, PIT determina la asignación de salida para cada hablante de interés y luego determina el error dada la salida asignada. La asignación con el error más bajo es elegida y el modelo es entrenado para minimizar el error correspondiente. En el caso particular de una mezcla de audio con dos hablantes, durante las etapas de entrenamiento e inferencia el modelo toma un segmento de la mezcla, y estima dos hablantes para el segmento. Ya que las dos salidas del modelo no corresponden a un hablante en específico, un mismo hablante puede cambiar de una salida a otra a través de segmentos consecutivos. Por consiguiente, las fuentes estimadas a este nivel de segmentos, necesitan ser organizadas secuencialmente a menos que los segmentos sean tan grandes como una palabra completa. Aunque este método es más simple que DPCL y DANet, tiene resultados similares a éstos.

2.4. Métricas de evaluación en la separación de voz

Para evaluar el desempeño de los sistemas de separación de voz, se han propuesto varias métricas para medir la capacidad de separación. Dichas métricas pueden agruparse en dos clases: a nivel de señal y a nivel de percepción. A nivel de señal, las métricas tienen como objetivo cuantificar el grado de mejora de la señal de voz o bien, la reducción de las interferencias. Adicionalmente a la tradicional *relación señal a ruido* (SNR), otras métricas comúnmente utilizadas son: la *relación señal a distorsión* (SDR), la *relación señal a artefactos* (SAR) [22] y la *inteligibilidad objetivo en tiempo corto* (STOI) [23]. A nivel de percepción la métrica más común es la *evaluación perceptual de la calidad de voz* (PESQ) [24].

En algunos escenarios, la medición del rendimiento del sistema es una tarea altamente dependiente de la aplicación. Por ejemplo, en la tarea de reconocimiento de voz de múltiples ha-

blantes, la separación de voz es solamente un paso intermedio y la métrica esencial del sistema es la eficiencia de reconocimiento medida con, e.g., la *tasa de error de reconocimiento de palabras* (WER). Otro ejemplo es en la tarea de identificación de múltiples hablantes, donde generalmente se emplea la *relación de error igual* (EER) para evaluar el desempeño de la separación de voz en ambientes ASA típicos [4].

2.5. Resumen

En este capítulo se han descrito las técnicas más relevantes que intentan solucionar el problema del efecto fiesta de cóctel: una tarea trivial para el ser humano que consiste localizar y focalizar su atención en una fuente de sonido de interés en un ambiente ruidoso y al mismo tiempo ignorar el resto de las señales. No obstante, esta habilidad resulta muy difícil de igualar mediante una computadora, pero que sin embargo la investigación en esta área ha otorgado resultados importantes en torno a la resolución de este problema. Por un lado, se han descrito las técnicas convencionales para la separación monoaural de señales de voz tales como CASA, NMF y los modelos generativos, siendo estos últimos los de mayor desempeño. También se han descrito métodos de separación de fuentes tales como el formador de haz y la separación ciega de fuentes basada en en el análisis de componentes independientes, los cuales incorporan información espacial a la solución del problema debido a su configuración multicanal. Por otro lado, se han descrito las técnicas novedosas recientes de DPCL, DANet y PIT para la solución de la separación monoaural de voz de múltiples hablantes, los cuales superan en desempeño a cualquier técnica convencional y representan el estado del arte en el área de separación de voz. Finalmente se han descrito las métricas de evaluación de sistemas de separación de fuentes más comunes, las cuales son altamente dependientes del problema a resolver.

3

Procesamiento Digital de la Señal de Voz

El intercambio de información a través de la voz es una de las formas más convenientes en la interacción hombre-máquina, ya que es el método de comunicación más natural y por consecuencia el más ampliamente utilizado por los humanos. La señal de voz posee una representación tanto en el dominio del tiempo como en el dominio de la frecuencia. Adicionalmente, dadas las características espectrales altamente fluctuantes de la voz al paso del tiempo, estas representaciones pueden mezclarse para determinar una nueva representación tiempo-frecuencia. En cada uno de estos dominios se puede analizar la señal de voz para revelar las características del tracto vocal e información lingüística relevante del hablante.

3.1. Análisis de Fourier en el tiempo discreto

La representación de Fourier de señales continuas y discretas es muy importante en el procesamiento de señales, ya que proporciona un método para trasladar señales a otro dominio en el cual puedan ser manipuladas [25]. Este otro dominio corresponde al *dominio de la frecuencia*, cuya representación y métodos resultantes, históricamente emergieron de la transformación de señales en el dominio del tiempo en otras formas útiles. Dos de las más notables son la *Serie de Fourier* y la *Transformada de Fourier*, desarrolladas por el matemático francés Jean-Baptiste Joseph Fourier.

3.1.1. Transformada de Fourier en Tiempo Discreto

La respuesta en frecuencia de un sistema discreto lineal e invariante en el tiempo puede determinarse mediante una suma de multiplicaciones de la respuesta al impulso $h(n)$ con una exponencial compleja $e^{-jn\omega}$, evaluada para cada valor de n, donde $-\infty < n < \infty$. En este sentido, la *transformada de Fourier en tiempo discreto* (DTFT, por sus siglas en inglés) de una secuencia $x(n)$, se define como

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{3.1}$$

donde $x(n)$ puede representarse mediante una integral de Fourier de la forma,

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega \tag{3.2}$$

La ecuación 3.2 representa la *transformada de Fourier en tiempo discreto inversa* o , que puede ser vista como una descomposición de la secuencia $x(n)$ en una combinación lineal de exponenciales complejas cuyas frecuencias se encuentran en un intervalo de longitud 2π.

En el tiempo continuo, las contrapartes naturales de las ecuaciones 3.1 y 3.2 son

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\mathrm{d}t \tag{3.3}$$

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}\mathrm{d}\omega \tag{3.4}$$

Las ecuaciones 3.3 y 3.4 definen a la *transformada de Fourier* y *transformada de Fourier inversa* en el tiempo continuo, respectivamente.

Sin embargo, para que la DTFT de una secuencia exista, la sumatoria en la ecuación 3.1 debe converger, es decir, se requiere que $x(n)$ sea absolutamente sumable [26]

$$\sum_{n=-\infty}^{\infty} |x(n)| = S < \infty \tag{3.5}$$

De manera que la respuesta en frecuencia de un sistema discreto lineal e invariante en el tiempo, $H(e^{j\omega})$, es la DTFT de la respuesta al impulso $h(n)$, la cual puede representarse mediante una integral de Fourier como

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}\mathrm{d}\omega \tag{3.6}$$

donde $H(e^{j\omega})$, de acuerdo a la ecuación 3.1 se calcula como

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-jn\omega} \tag{3.7}$$

Dadas las secuencias discretas $x(n)$ y $y(n)$, y sus respectivas transformadas de Fourier $X(e^{j\omega})$ y $Y(e^{j\omega})$, algunas de las propiedades más importantes de la DTFT se listan en la tabla 3.1.

Propiedad	**Secuencia**	**DTFT**
Linealidad	$ax(n) + by(n)$	$aX(e^{j\omega}) + bY(e^{j\omega})$
Desplazamiento	$x(n - n_0)$	$e^{-jn_0\omega}X(e^{j\omega})$
Reflexión	$x(-n)$	$X(e^{-j\omega})$
Modulación	$e^{jn\omega_0}x(n)$	$X(e^{j(\omega-\omega_0)})$
Convolución	$x(n) * y(n)$	$X(e^{j\omega})Y(e^{j\omega})$
Conjugación	$x^*(n)$	$X^*(e^{-j\omega})$
Derivación	$nx(n)$	$j\frac{\mathrm{d}X(e^{j\omega})}{\mathrm{d}\omega}$
Multiplicación	$x(n)y(n)$	$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(\omega-\theta)})\mathrm{d}\theta$

Tabla 3.1: Propiedades de la Transformada de Fourier en Tiempo Discreto [25].

3.1.2. Transformada Discreta de Fourier

Es posible que una secuencia de duración finita $0 \leq n \leq (N-1)$ y longitud N en el dominio del tiempo, sea trasladada al dominio de la frecuencia como una secuencia $X(k)$ de la misma longitud N, donde $k = 0, 1, ...(N-1)$. La transformación se realiza al dominio de la frecuencia dado que $X(k)$, para cualquier valor de k, representa el coeficiente de Fourier para la exponencial compleja discreta con frecuencia igual al k-ésimo harmónico de la frecuencia fundamental $\frac{2\pi}{N}$. Con base en lo anterior, se define la *Transformada Discreta de Fourier* (DFT, por sus siglas en inglés), como

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} \qquad k = 0, 1, 2, ..., (N-1) \tag{3.8}$$

Dado que la DFT es una transformación reversible, la *transformada discreta de Fourier inversa* o *IDFT* se define como:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi kn}{N}} \tag{3.9}$$

Por conveniencia en notación, la DFT y la IDFT son comúnmente escritas en términos de la cantidad compleja W_N, definida por

$$W_N = e^{-j\frac{2\pi}{N}} \tag{3.10}$$

De forma que las ecuaciones que definen la DFT y la IDFT pueden ser expresadas como

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \tag{3.11}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \tag{3.12}$$

La comparación de la definición de la DTFT en la ecuación 3.1 con la definición de la DFT en la ecuación 3.8, resulta en que los coeficientes de la DFT son muestras de la DTFT, es decir

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} = \sum_{n=-\infty}^{\infty} x(n) e^{-j\frac{2\pi kn}{N}} = X(e^{j\omega})|_{\omega=\frac{2\pi k}{N}} \tag{3.13}$$

En la tabla 3.2 se muestran algunas propiedades importantes de la DFT.

Propiedad	**Secuencia**	**DFT**
Linealidad	$ax(n) + by(n)$	$aX(k) + bY(k)$
Periodicidad	$x(n + mN) = x(n)$	$X(k + mN) = X(k)$
Reflexión	$x(-n)$	$X(-k)$
Desplazamiento	$x(n - n_0)$	$X(k)e^{-j\frac{2\pi n_0 k}{N}}$
Modulación	$x(n)e^{j\frac{2\pi k_0 n}{N}}$	$X(k - k_0)$
Multiplicación	$x(n)y(n)$	$\frac{1}{N}[X(k) \otimes Y(k)]$

Tabla 3.2: Propiedades de la Transformada Discreta de Fourier [27].

3.1.3. Transformada Rápida de Fourier

La *transformada rápida de Fourier* (FFT, por sus siglas en inglés), es un algoritmo ingenioso para calcular la DFT, pero en un tiempo mucho menor. Por consiguiente, la FFT reduce la complejidad computacional de la DFT del orden $O(N^2)$ al orden $O(N \log_2 N)$.

La historia de la FFT comienza en 1805, cuando Carl Friedrich Gauss intentó determinar la órbita de ciertos asteroides. De este modo, desarrolló la transformada discreta de Fourier, incluso antes de que Joseph Fourier publicara sus resultados en 1822. Para calcular la DFT, Gauss inventó un algoritmo que es equivalente al de Cooley y Tukey publicado en 1965, tiempo en el que la milicia estadounidense estaba interesada en un método para detectar pruebas soviéticas nucleares. Entre 1805 y 1965, varios científicos desarrollaron métodos eficientes para calcular la

DFT, pero ninguno de ellos fue tan general como el método desarrollado por Gauss o Cooley y Tukey. La tabla 3.3 resume los principales descubrimientos de métodos eficientes para el cálculo de la DFT.

Investigador(es)	Fecha	Aplicación
C. F. Gauss	1805	Interpolación de órbitas de cuerpos celestes
F. Carlini	1828	Análisis armónico de presión barométrica
A. Smith	1846	Corrección de desviaciones de brújulas en barcos
J. D. Everett	1860	Modelado de desviaciones de temperatura bajo tierra
C. Runge	1903	Análisis armónico de funciones
K. Stumpff	1939	Análisis armónico de funciones
Danielson y Lanczos	1942	Difracción de rayos X en cristales
L. H. Thomas	1948	Análisis armónico de funciones
I. J. Good	1958	Análisis armónico de funciones
Cooley y Tukey	1965	Análisis armónico de funciones
S. Winograd	1976	Uso de la teoría de la complejidad para análisis armónico

Tabla 3.3: Principales descubrimientos de métodos eficientes para el cálculo de la DFT [28].

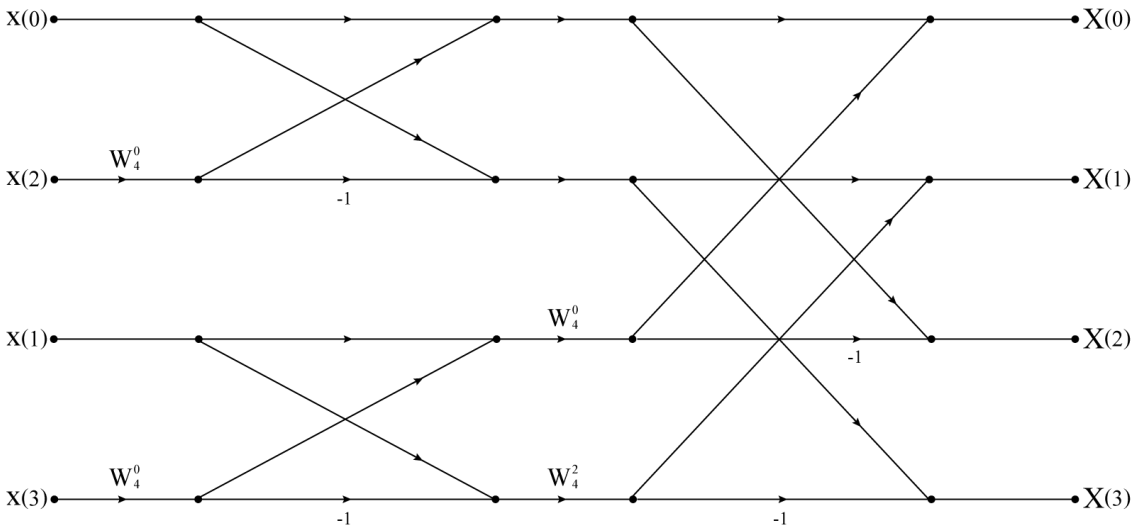


Figura 3.1: Diagrama de mariposa de la FFT de cuatro puntos.

Existen diversas variantes para el cálculo rápido de la DFT. La eficiencia de los algoritmos consiste en realizar cálculos más pequeños de la DFT, explotando las propiedades de simetría y periodicidad de la exponencial compleja $W_N^{kn} = e^{-j(2\pi/N)kn}$. A los algoritmos que descomponen la secuencia discreta $x(n)$ en subsecuencias más pequeñas, son conocidos como algoritmos de

decimación en el tiempo. El algoritmo de la FFT de *Cooley* y *Tukey* está basado en este enfoque. La figura 3.1 muestra el diagrama del cálculo de una DFT de cuatro puntos ($N = 4$) empleando un algoritmo de decimación en el tiempo. El cálculo se realiza en dos etapas; primero se calculan dos DFTs de dos puntos, y luego una de cuatro puntos. El cálculo básico realizado en cada etapa consiste en tomar dos números a y b, multiplicar b por W_N^r, y luego sumar y restar el producto de a para formar un nuevo par de números A y B. A este cálculo se le conoce como *mariposa*, ya que el diagrama de flujo se asemeja a una mariposa.

También es posible derivar algoritmos que primero descomponen la secuencia de salida $X(k)$ en subsecuencias sucesivas más pequeñas. Estos algoritmos se denominan algoritmos de *decimación en frecuencia*. El algoritmo de *Sande* y *Tukey* es un ejemplo del uso de este enfoque. Otra clase de FFTs subdivide el conjunto inicial de datos de longitud N, en pequeñas potencias de 2, por ejemplo si $N = 2$, se obtienen FFTs de *base 4*, o si $N = 8$, se obtienen FFTs de *base 8*. Estas pequeñas transformaciones se llevan a cabo por secciones de código altamente optimizado que toman ventaja de las simetrías existentes en ese particular N. Por ejemplo, si $N = 4$, los valores de las funciones seno y coseno que ingresan al algoritmo son todos ± 1 o 0, de manera que muchas multiplicaciones son eliminadas, dejando en gran parte sumas y restas.

Existen también algoritmos de la FFT para conjuntos de datos de longitud N que no son potencia de 2, los cuales consisten en subdividir la secuencia inicial en subsecuencias sucesivas más pequeñas, pero no mediante factores de 2, sino por cualquier factor primo que divida a N. Entre más grande es el factor primo de N, más bajo es el rendimiento del método. Si la longitud de la secuencia discreta es un número primo, entonces no es posible realizar una subdivisión, y la transformada de Fourier se lleva a cabo empleando N^2 operaciones.

Los algoritmos de *Winograd* son otros algoritmos para el cálculo rápido de la DFT. Son análogos a las FFTs de base-4 y base-8. Winograd derivó códigos altamente óptimos para tomar transformadas discretas de Fourier de una longitud pequeña, e.g., para $N = 2, 3, 4, 5, 7, 8, 11, 13, 16$. El algoritmo también emplea una forma ingeniosa para combinar los subfactores, pues involucra un método para el reordenamiento de los datos antes y después del procesamiento, permitiendo una reducción significativa en el número de multiplicaciones en el algoritmo [29].

3.2. Muestreo y reconstrucción de señales

El muestreo es una de las operaciones fundamentales en el procesamiento digital de señales, ya que proporciona un mecanismo para convertir señales continuas en señales discretas. Esta conversión se lleva a cabo tomando de la señal continua un número suficiente de muestras adquiridas cada cierto intervalo de tiempo, con las cuales la señal puede ser representada en su totalidad. Este hecho se establece en el *teorema del muestreo*, que de manera más específica, determina la frecuencia mínima con la cual una señal continua debe ser muestreada para convertirla en una señal discreta sin que exista pérdida de información y que a su vez, de acuerdo a esta condición, la señal original pueda ser reconstruida perfectamente a partir de sus muestras [30]. Una manera conveniente para representar el muestreo de una señal continua y entender los efectos producidos en el domino de la frecuencia es a través de los operadores $\mathrm{comb}[\cdot]$ y $\mathrm{rep}[\cdot]$ [31], cuya definición se establece en las secciones siguientes.

3.2.1. Operador comb$[\cdot]$

El operador comb$[\cdot]$ aplicado a una función $f(t)$, consiste en multiplicar dicha función por un tren de impulsos periódico $p(t)$ con periodo T, es decir,

$$\text{comb}[f(t)] = f(t)p(t) \tag{3.14}$$

donde $p(t)$ se define matemáticamente como:

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3.15}$$

o bien, mediante su representación en una serie de Fourier compleja como:

$$p(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{jn\frac{2\pi}{T}t} \tag{3.16}$$

Sustituyendo la ecuación 3.15 en la ecuación 3.14,

$$\text{comb}[f(t)] = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3.17}$$

La ecuación 3.17 se puede reescribir como:

$$\text{comb}[f(t)] = \sum_{n=-\infty}^{\infty} f(t)\delta(t - nT) \tag{3.18}$$

Empleando la propiedad de multiplicación de la función impulso $f(t)\delta(t - t_0) = f(t_0)\delta(t - t_0)$, el operador comb$[\cdot]$ aplicado a una función $f(t)$, también puede definirse como:

$$\text{comb}[f(t)] = \sum_{n=-\infty}^{\infty} f(nT)\delta(t - nT) \tag{3.19}$$

3.2.2. Operador rep$[\cdot]$

El operador rep$[\cdot]$ denota un proceso mediante el cual una función es replicada periódicamente. Aplicado a una función $f(t)$, consiste en convolucionar dicha función con un tren de impulsos periódico $p(t)$, con periodo T, es decir,

$$\text{rep}[f(t)] = f(t) * p(t) \tag{3.20}$$

Sustituyendo la ecuación 3.15 en la ecuación 3.20,

$$\text{rep}[f(t)] = f(t) * \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3.21}$$

La ecuación 3.21 se puede reescribir como:

$$\text{rep}[f(t)] = \sum_{n=-\infty}^{\infty} f(t) * \delta(t - nT) \tag{3.22}$$

Empleando la propiedad de convolución de la función impulso $f(t) * \delta(t - t_0) = f(t - t_0)$, el operador rep$[\cdot]$ aplicado a una función $f(t)$, también puede definirse como:

$$\text{rep}[f(t)] = \sum_{n=-\infty}^{\infty} f(t - nT) \tag{3.23}$$

3.2.3. Teorema del muestreo

El muestreo de una señal continua limitada en banda, es decir, de una señal cuya transformada de Fourier es exactamente cero fuera de una banda finita de frecuencias ($X(w) = 0\, para\, |\omega| \geq \omega_M$), puede llevarse a cabo aplicando el operador $comb$ a la señal que se desea muestrear. El resultado de esta operación, y cuyo desarrollo se muestra gráficamente en la figura 3.2 (a)-(c), es una señal muestreada $x_s(t)$, que consiste en un tren de impulsos cuya amplitud de cada impulso es igual a las muestras de la señal original $x(t)$, en intervalos equiespaciados un valor T, es decir,

$$x_s(t) = \text{comb}[x(t)] = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) \tag{3.24}$$

Analíticamente, los efectos en el dominio de la frecuencia del muestreo en el dominio del tiempo, se pueden conocer al aplicar la transformada de Fourier a la ecuación 3.24, esto es,

asumiendo que

$$\mathscr{F}\{x(t)\} \longrightarrow X(\omega) \tag{3.25}$$

entonces,

$$\mathscr{F}\{\text{comb}[x(t)]\} = \mathscr{F}\{x(t)p(t)\} \tag{3.26}$$

De acuerdo al teorema de la convolución en la frecuencia, el cual afirma que

$$\mathscr{F}\{f_1(t)f_2(t)\} = \frac{1}{2\pi}F_1(\omega) * F_2(\omega) \tag{3.27}$$

y la transformada de Fourier del tren de impulsos $p(t)$, dada por

$$\mathscr{F}\{p(t)\} = \omega_o \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0), \qquad \omega_0 = \frac{2\pi}{T} \tag{3.28}$$

se tiene que la ecuación 3.26 puede ser expresada como:

$$\mathscr{F}\{\text{comb}[x(t)]\} = \frac{1}{2\pi}\left[X(\omega) * \omega_o \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0)\right] \tag{3.29}$$

Sustituyendo ω_0 y reacomodando los términos del lado derecho de la ecuación 3.29, se tiene que,

$$\mathscr{F}\{\text{comb}[x(t)]\} = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(\omega) * \delta(\omega - n\omega_0) \tag{3.30}$$

Finalmente, empleando la propiedad de convolución de la función impulso se obtiene la transformada de Fourier de comb$[x(t)]$:

$$\mathscr{F}\{\text{comb}[x(t)]\} = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(\omega - n\omega_0) \tag{3.31}$$

Comparando el lado derecho de la ecuación 3.31 con la definición del operador rep$[\cdot]$ aplicado a una función $f(t)$, dada en la ecuación 3.23, se observa que,

$$\mathscr{F}\{\text{comb}[x(t)]\} = \frac{1}{T}\text{rep}[X(\omega)] \tag{3.32}$$

por lo cual,

$$x_s(t) \xrightarrow{\mathscr{F}} X_s(\omega) \qquad X_s(\omega) = \frac{1}{T}\text{rep}[X(\omega)] \tag{3.33}$$

La ecuación 3.33 muestra que la transformada de Fourier de $x_s(t)$, es una función que replica periódicamente el espectro de la señal original $x(t)$. La figura 3.2 (d)-(f) ilustra gráficamente este proceso.

De acuerdo a la figura 3.2 (f), el espectro de una señal muestreada idealmente es una repetición periódica escalada en amplitud del espectro original. Ya que la señal $x(t)$ es una señal limitada en banda y ha sido muestreada de manera que la frecuencia de muestreo es al menos dos veces mayor que la frecuencia máxima contenida en la señal, las repeticiones periódicas de $X(\omega)$ no se traslapan. Cuando el muestreo no cumple con esta condición, se produce un fenómeno conocido como *aliasing*, el cual produce traslapes en el espectro de la señal muestreada tal como se muestra en la figura 3.3, ocasionando inevitablemente la pérdida de información. Esta idea se establece en el *teorema del muestreo*, o *teorema de Nyquist-Shannon*, en el cual también se enuncia que si la condición anterior matemáticamente expresada como:

$$\omega_s > 2\omega_M, \qquad \omega_s = \frac{2\pi}{T} \tag{3.34}$$

se cumple, entonces que la señal puede ser reconstruida perfectamente a partir de sus muestras. Este procedimiento se lleva a cabo empleando un filtro pasobajas con ganancia T y frecuencia de corte igual a $\omega_s/2$ para recuperar $X(\omega)$ de $X_s(\omega)$, es decir,

$$X(\omega) = \begin{cases} TX_s(\omega), & |\omega| \leq \frac{\omega_s}{2} \\ 0, & \text{otro caso} \end{cases} \tag{3.35}$$

En la figura 3.4 (a)-(d) se muestra el proceso de reconstrucción de una señal continua a partir de sus muestras utilizando un filtro ideal pasobajas. Sin embargo, la condición de Nyquist para reconstruir perfectamente una señal continua a partir de sus muestras es solamente una condición suficiente más no necesaria, pues existen señales que a pesar de no cumplir con este teorema pueden ser reconstruidas en su totalidad. La figura 3.4 (e)-(h) muestra un ejemplo de reconstrucción perfecta cuando no se cumple la condición de Nyquist.

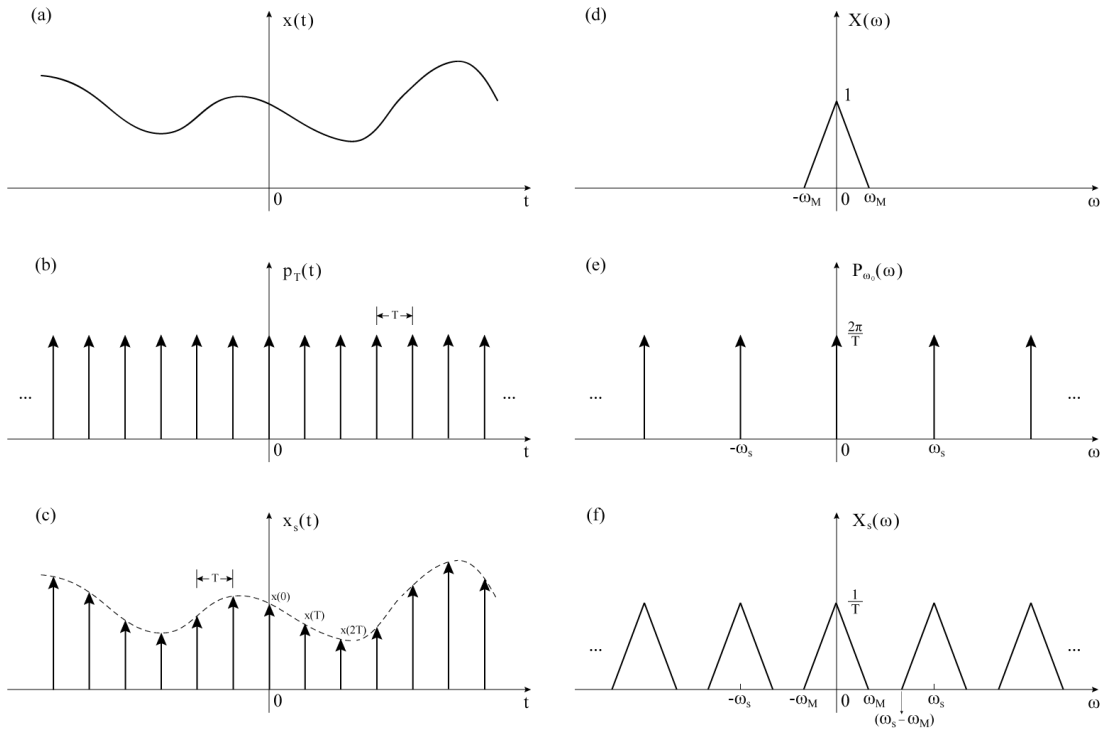


Figura 3.2: Muestreo de una señal continua. (a) Señal continua. (b) Tren periódico de impulsos. (c) Señal muestreada. (d) Espectro de una señal continua. (e) Espectro de un tren periódico de impulsos. (f) Espectro de una señal muestreada.

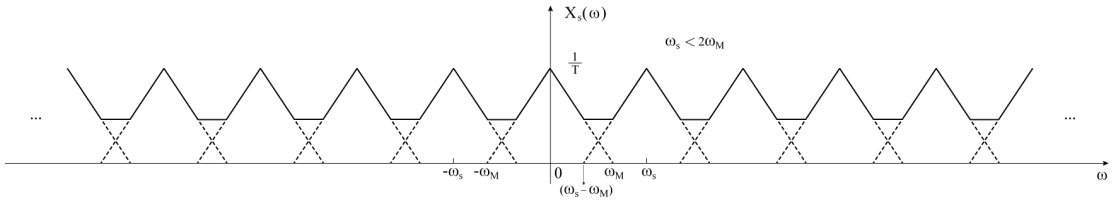


Figura 3.3: Efecto de aliasing en el dominio de la frecuencia debido a un submuestreo.

3.3. Análisis de la señal de voz en el dominio del tiempo

El análisis de la señal de voz en el dominio del tiempo permite una interpretación física sencilla y simplicidad en el cálculo de parámetros relevantes del habla [32]. Entre las características más importantes que se encuentran con facilidad en el análisis temporal están las estadísticas de la forma de onda de la señal, la frecuencia fundamental, la energía, así como la tasa de cruces por cero y la autocorrelación, las cuales proporcionan detalles espectrales sin emplear necesariamente métodos formales de análisis espectral [33].

En la mayoría de los esquemas de procesamiento de voz, existe una suposición fundamental que considera que las propiedades de la señal de voz cambian de forma relativamente lenta

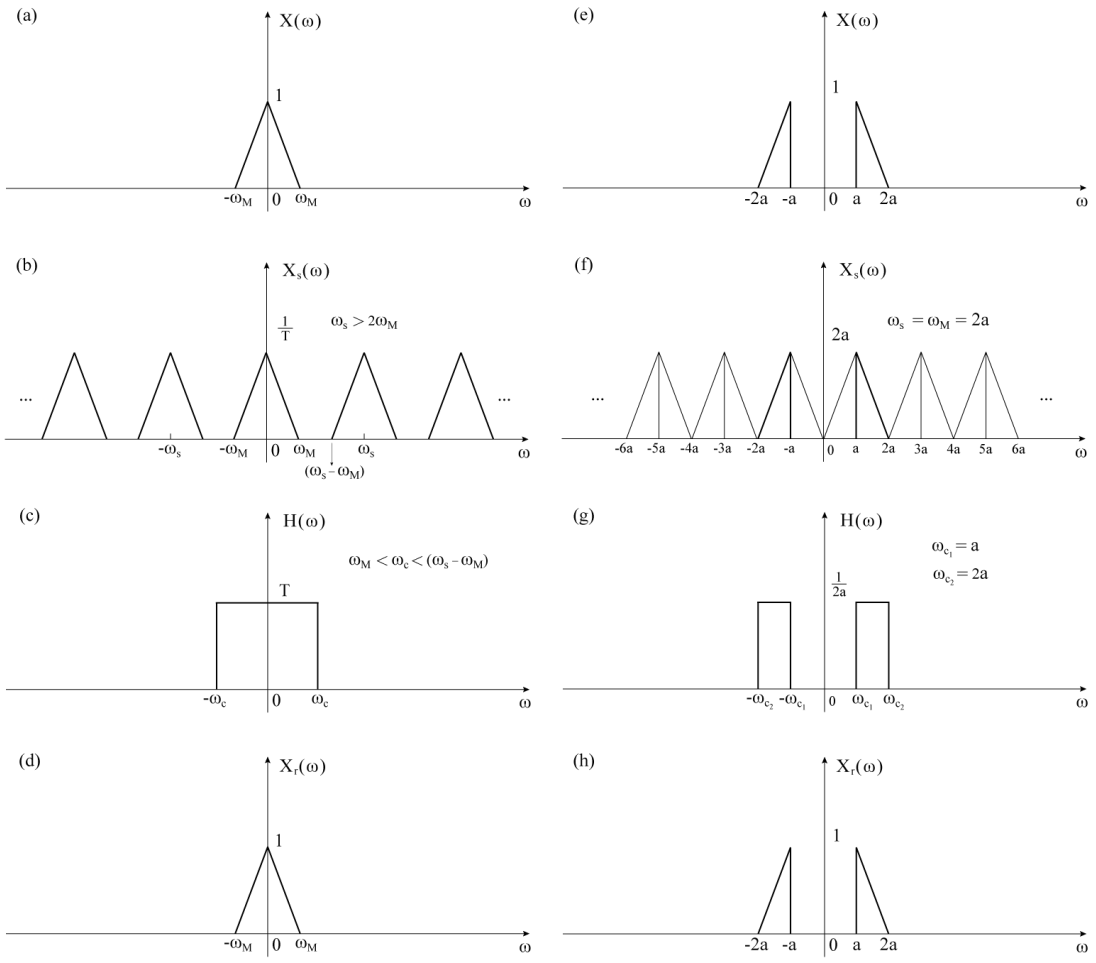


Figura 3.4: Reconstrucción ideal de una señal continua. (a) Espectro $X(\omega)$ de una señal continua $x(t)$. (b) Espectro $X_s(\omega)$ de una señal muestreada $x_s(t)$. (c) Espectro $H(\omega)$ de un filtro pasobajas ideal. (d) Espectro $X_r(\omega)$ de una señal continua recuperada a partir de sus muestras. (e) Espectro $X(\omega)$ de una señal pasobanda. (f) Espectro $X_s(\omega)$ de una señal pasobanda muestreada a una frecuencia menor a la establecida por la condición de Nyquist. (g) Espectro $H(\omega)$ de un filtro pasobanda. (h) Espectro $X_r(\omega)$ de la señal pasobanda recuperada a partir de sus muestras.

con respecto al tiempo. De manera que cuando la señal de voz es analizada sobre periodos de tiempo entre 5 y 100 ms, sus características se pueden consideran estacionarias [34], es decir, que sus parámetros estadísticos tales como la media, varianza y potencia de las componentes espectrales, entre otros, se mantienen constantes. Esta suposición conduce a una variedad de métodos de procesamiento en *tiempo corto*, en los cuales segmentos muy pequeños de la señal de voz son aislados y procesados como si fueran segmentos cortos de un sonido prolongado con propiedades fijas [35]. Del procesamiento de la señal resulta una secuencia distinta que sirve como una nueva representación de la señal de voz.

3.3.1. Entramado y ventaneo

Debido a la naturaleza no estacionaria de la señal de voz, su análisis en el dominio del tiempo se lleva a cabo tomando pequeñas porciones de la señal a la vez. Cada segmento de voz seleccionado se conoce como *trama de análisis*, por lo que la señal de voz es procesada trama por trama, comúnmente en intervalos superpuestos, hasta que toda la región de voz es cubierta por al menos una de las tramas. La figura 3.5 muestra la segmentación en tramas superpuestas de una señal de voz.

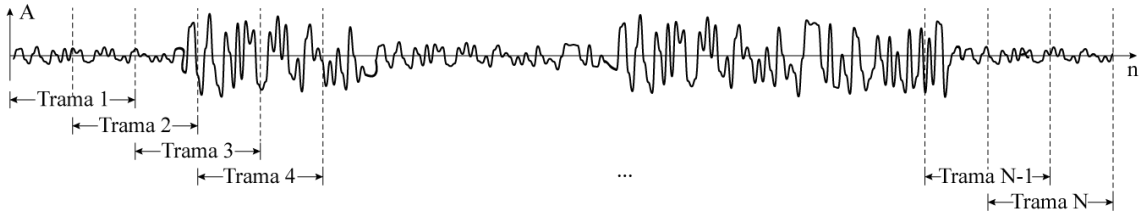


Figura 3.5: Entramado de una señal de voz. Las tramas se superponen 50 %.

De forma más específica, a la técnica empleada para segmentar una señal en tramas de un número finito de muestras se le conoce como *ventaneo*, que consiste en la multiplicación de una señal con una función denominada *ventana* cuya amplitud es cero excepto en la región de interés. El uso de ventanas es importante, ya que es necesario considerar cómo tratar los bordes de las tramas para reducir los componentes espectrales generados por el proceso de segmentación. En la figura 3.6 se ilustran los tipos de ventanas más empleadas en el procesamiento digital de señales y en la tabla 3.4 se muestra su definición matemática.

3.3.2. Energía

La energía de una señal discreta se define matemáticamente como:

$$E_x = \sum_{n=-\infty}^{\infty} |x(n)|^2$$

Sin embargo, la ecuación anterior para el cálculo de la energía tiene poca utilidad si la señal discreta es tiene propiedades estadísticas variantes en el tiempo. Por consiguiente, su uso sobre una señal de voz carece totalmente de significado, pues ofrece poca información acerca de las propiedades de la señal de voz que son dependientes del tiempo.

Para conocer las variaciones en el tiempo de la energía en una señal de voz, es necesario tener una representación diferente de la señal. Es por ello que la mayoría de las técnicas de análisis de la señal de voz en el dominio del tiempo se representan matemáticamente mediante la siguiente forma:

$$Q_n = \sum_{m=-\infty}^{\infty} T\{x(m)\}w(n-m) \tag{3.36}$$

en donde una transformación $T\{\cdot\}$ que puede ser lineal o no lineal, aplicada a una señal de voz, denotada por $x(n)$, es convolucionada con una ventana $w(n)$, usualmente de longitud finita. En la figura 3.7 se muestra el diagrama de bloques de este proceso.

Propiedad	Definición matemática
Rectangular	$w(n) = \begin{cases} 1, & \lvert n\rvert \leq \frac{N-1}{2} \\ 0, & \text{otro caso} \end{cases}$
Triangular	$w(n) = \begin{cases} 1 - \frac{\lvert n\rvert}{N/2}, & \lvert n\rvert \leq \frac{N-1}{2} \\ 0, & \text{otro caso} \end{cases}$
Hanning	$w(n) = \begin{cases} 0.5\left[1 + \cos\left(\frac{2\pi}{N}n\right)\right], & \lvert n\rvert \leq \frac{N-1}{2} \\ 0, & \text{otro caso} \end{cases}$
Hamming	$w(n) = \begin{cases} 0.54 + 0.46\cos\left(\frac{2\pi}{N}n\right), & \lvert n\rvert \leq \frac{N-1}{2} \\ 0, & \text{otro caso} \end{cases}$
Blackman	$w(n) = \begin{cases} 0.42 + 0.5\cos\left(\frac{2\pi n}{N}n\right) + 0.08\cos\left(\frac{2\pi}{N}2n\right), & \lvert n\rvert \leq \frac{N-1}{2} \\ 0, & \text{otro caso} \end{cases}$
Kaiser-Bessel	$w(n) = \begin{cases} \frac{I_0\left(\beta\sqrt{1-(\frac{n}{N/2})^2}\right)}{I_{0(\beta)}}, & \lvert n\rvert \leq \frac{N-1}{2} \\ 0, & \text{otro caso} \end{cases}$

Tabla 3.4: Tipos de ventanas y su definición matemática [36].

Si la transformación $T\{\cdot\}$ en la ecuación 3.36 realiza la operación de elevar al cuadrado, es decir, que aplicada a una señal $x(n), T\{x(n)\} = x^2(n)$, entonces Q_n corresponde a la *energía* en tiempo corto. Debido a que la señal es elevada al cuadrado, los valores presentes en la señal cuya amplitud es alta, son enfatizados con el cálculo de la energía, lo cual ayuda a reflejar la variación en amplitud de sonidos voceados y no voceados. Matemáticamente, la energía en tiempo corto se define como:

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2 \tag{3.37}$$

donde a diferencia de la ecuación 3.3.2, el resultado es un vector de valores de energía para cada instante de análisis.

3.3.3. Magnitud

Una de las desventajas de la función de energía definida en la ecuación 3.37, es que es muy sensible a grandes niveles de energía, por lo que muestra a muestra las variaciones grandes en la señal $x(n)$ son enfatizadas. Una forma simple de aligerar el problema es que la transformación $T\{\cdot\}$ de la ecuación 3.36, en lugar de elevar al cuadrado la función sobre la cual opera, calcule

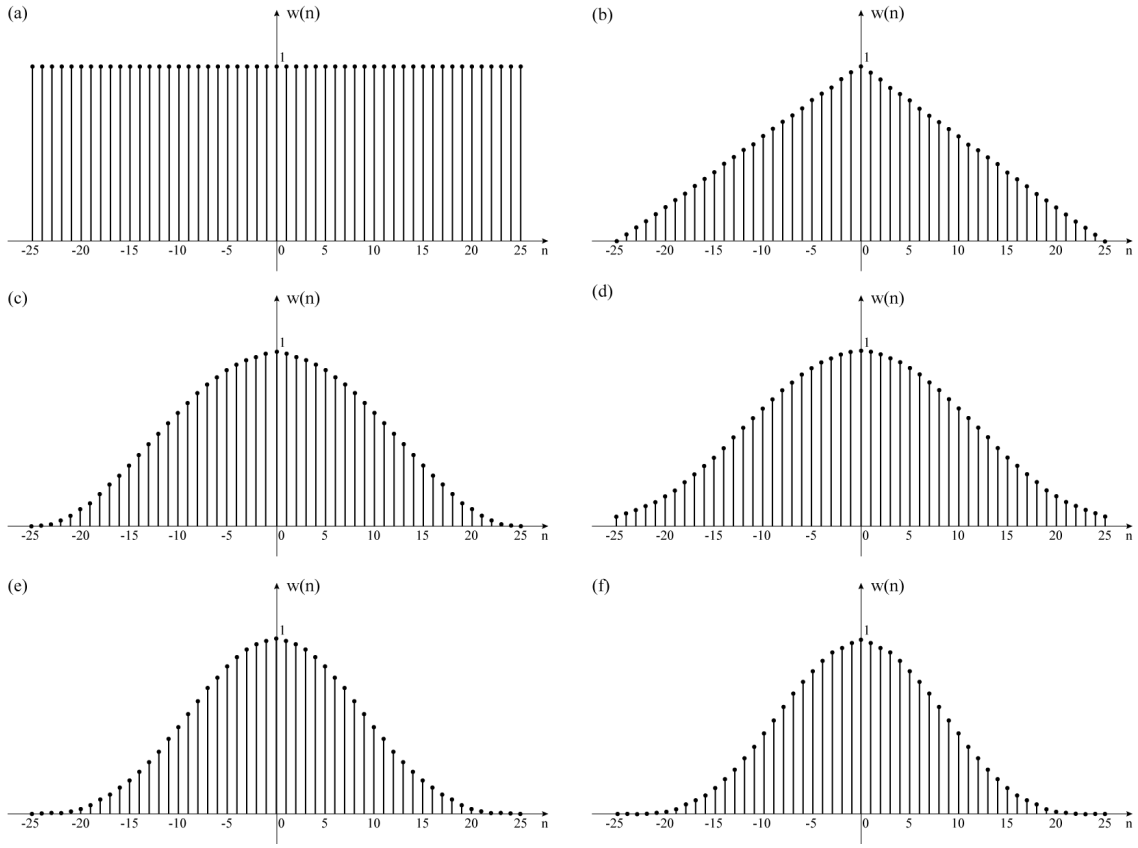


Figura 3.6: Diferentes tipos de ventanas ($N = 25$). (a) Rectangular. (b) Triangular. (c) Hanning. (d) Hamming. (e). Blackman (f). Kaiser-Bessel. Adaptado de [36].

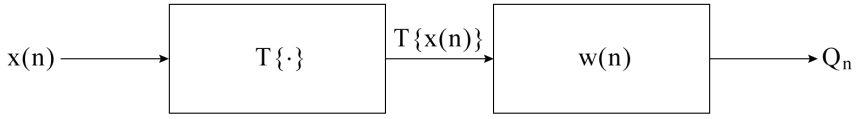


Figura 3.7: Análisis de la señal de voz en el dominio del tiempo.

su magnitud absoluta, es decir, que $T\{x(n)\} = |x(n)|$. Por consiguiente, la función de *magnitud* puede definirse como:

$$M_n = \sum_{m=-\infty}^{\infty} |x(m)|w(n-m) \tag{3.38}$$

3.3.4. Cruces por cero

Los *cruces por cero* indican el número de veces que una señal atraviesa el nivel cero en cualquiera de los dos sentidos [37]. Un cruce por cero se determina cuando dos muestras sucesivas de una señal tienen signos algebraicos diferentes. Matemáticamente, la tasa de cruces por cero

se define como:

$$Z_n = \frac{1}{2} \sum_{m=-\infty}^{\infty} |\text{sgn}[x(m)] - \text{sgn}[x(m-1)]| w(n-m) \tag{3.39}$$

donde,

$$\text{sgn}[x(n)] = \begin{cases} 1, & x(n) \geq 0 \\ -1, & x(n) \leq 0 \end{cases} \tag{3.40}$$

El cálculo de los cruces por cero es simplemente una medida del contenido frecuencial de una señal. El modelo de producción de voz sugiere que la energía de un sonido voceado se concentra por debajo de los 3 kHz, mientras que en los sonidos no voceados, la mayoría de la energía se encuentra en las frecuencias altas. Debido a que las altas frecuencias implican tasas de cruces por cero altas, y bajas frecuencias implican tasas de cruces por cero bajas, existe una fuerte correlación entre la tasa de cruces por cero y la distribución de la energía, con la frecuencia. Con base en lo anterior, una generalización razonable, sin embargo imprecisa, es que si la tasa de cruces por cero es alta, entonces la señal de voz es no voceada, mientras que si la tasa de cruces por cero es baja, la señal de voz es voceada [35].

3.3.5. Función de autocorrelación

Dadas dos señales de energía finita $x(n)$ y $y(n)$, la *correlación* entre $x(n)$ y $y(n)$, denotada por la función r_{xy}, se define como:

$$r_{xy}(\ell) = \sum_{n=-\infty}^{\infty} x(n) y(n-\ell) \tag{3.41}$$

donde el índice ℓ es un parámetro de desplazamiento y el subíndice xy indica que la secuencia $x(n)$ permanece fija mientras la secuencia $y(n)$ se desplaza ℓ unidades. La correlación es útil para medir el grado en que dos señales son similares. La técnica basada en el cálculo de la correlación entre una señal y una versión retardada de la misma, es decir, el caso especial donde $y(n) = x(n)$ es llamada *autocorrelación*, que analíticamente se define como:

$$r_{xx}(\ell) = \sum_{n=-\infty}^{\infty} x(n) x(n-\ell) \tag{3.42}$$

En el procesamiento digital de señales de voz, la función de autocorrelación puede ser empleada para encontrar la frecuencia fundamental o la tonalidad (pitch) de una señal, sin embargo, se requiere el uso de su forma analítica en tiempo corto, la cual se define como:

$$R_n(\ell) = \sum_{m=-\infty}^{\infty} x(m) w(n-m) x(n-\ell) w(n-m+\ell) \tag{3.43}$$

3.4. Análisis de la señal de voz en el dominio de la frecuencia

Al igual que las propiedades temporales de la señal de voz se mantienen fijas en intervalos cortos de tiempo, las propiedades espectrales de la señal de voz también pueden asumirse invariables bajo el mismo análisis en tiempo corto. La caracterización de las propiedades espectrales de las señales de voz se logra mediante la representación variante en el tiempo de la transformada de Fourier.

3.4.1. Transformada de Fourier en Tiempo Corto

Las propiedades espectrales de la señal de voz pueden ser analizadas mediante la *transformada de Fourier en tiempo corto* (STFT, por sus siglas en inglés) *STFT* definida matemáticamente por:

$$X(n,k) = \sum_{m=0}^{L-1} w(m)x(n+m)e^{-j2\pi km/N} \tag{3.44}$$

donde L es la longitud de la ventana $w(n)$ y $k = 0, 1, ..., N-1$. La STFT es una representación bidimensional de la señal unidimensional $x(n)$, ya que es una función de dos variables: el índice de tiempo n, y el índice de frecuencia k, por lo que $X(n,k)$ es una colección de DFTs de N puntos de una trama de $x(m)$ que comienza en $m = n$ y termina en $m = n+L-1$. La ventana determina la porción de la señal que se enfatiza y está fija en el intervalo de $m = 0$ a $m = L-1$. A medida que el índice n cambia, la señal $x(n+m)$ se desplaza y la ventana extrae un segmento diferente de la señal para el análisis. Básicamente, para cada valor de n se extrae un segmento de la señal $x(m)$ y se evalúa el espectro *local* [35].

En el cálculo de la STFT se emplean ventanas separadas un determinado número de muestras, el cuál puede especificarse en términos de la superposición o traslape existente entre ventanas sucesivas. El criterio para decidir la cantidad de traslape incluye la longitud de la ventana, la resolución deseada en el tiempo, y la rapidez con la que las características de la señal cambian con el tiempo.

3.4.2. Efecto del ventaneo en la STFT

El objetivo principal de la ventana en la STFT es limitar la extensión de la secuencia digital a ser transformada de manera que las características espectrales sean cuasi-estacionarias a lo largo de la duración de la ventana. Entre más rápido cambien las características de la señal, más corta debería ser la duración de la ventana, sin embargo, a medida que la ventana se vuelve más corta, la resolución en frecuencia disminuye. Además, a medida que la longitud de la ventana disminuye, aumenta la capacidad de resolver cambios en el tiempo. Por consiguiente, la elección del tamaño de la ventana se vuelve en un compromiso entre la resolución en frecuencia y la resolución en el tiempo.

En general, el ventaneo de una señal en el dominio del tiempo introduce dos tipos de distorsión espectral:

- El efecto predominante del ancho del lóbulo principal es *esparcir* el espectro original, con lo cual se tiene una pérdida de resolución. Por ejemplo, una línea espectral en el espectro

original tendrá un ancho de alrededor de $2\pi/T_0$ después del ventaneo, y dos señales sinusoidales de igual amplitud cuyas frecuencias están separadas menos de $2/T_0$ se mezclarán una con la otra y podrían parecer una sola señal sinusoidal.

- El mayor efecto de los lóbulos laterales es que transfieren potencia desde bandas de frecuencia que contienen grandes cantidades de potencia de la señal a bandas que contienen poca o nula potencia. A esta transferencia de potencia se le conoce como *fuga* (en inglés *leakage*), y puede crear picos en frecuencias erróneas, crear picos que no existen o cambiar la amplitud de los picos existentes.

Los efectos de esparcimiento y fuga son críticos en espectros donde las diferencias de magnitud entre picos y valles son muy grandes, pero son despreciables en espectros relativamente planos, donde las diferencias de magnitud no son significativas. En este sentido, idealmente el espectro de una buena ventana debería tener un lóbulo principal angosto para minimizar el esparcimiento del espectro, y lóbulos laterales de baja magnitud para minimizar la fuga espectral. No obstante, como consecuencia del *principio de incertidumbre* de las transformadas de Fourier es imposible satisfacer ambos requerimientos simultáneamente. Por consiguiente, si se desea tener la mejor resolución espectral se debe elegir una ventana rectangular. Sin embargo, la ganancia en resolución espectral es a costa de fuga espectral. Para reducir este efecto de fuga, se requiere una ventana con lóbulos laterales bajos con un decaimiento rápido del espectro [26].

3.4.3. Espectrograma de voz

Durante el habla, los sonidos de voz cambian, de otra manera no sería posible comunicarse. En este sentido, un espectro que proporciona información frecuencial acerca de un sonido individual, no es suficiente para representar la naturaleza cambiante de los sonidos que se producen, y que es fundamental para comprender un mensaje. Lo que se requiere es una representación visual de la forma en que el espectro de sonido cambia con el tiempo, y esto se logra a través de un *espectrograma* [38]. Entonces, un espectrograma es una gráfica que muestra cómo el contenido frecuencial de una señal cambia con respecto al tiempo. Además, proporciona información más compleja que la que puede ser obtenida mediante la forma de onda de la señal en el dominio del tiempo, tal como la intensidad de las diferentes frecuencias presentes en la señal, cuya representación está dada por marcas de color blanco y negro o también multicolor; entre más intensa sea una frecuencia particular contenida en la señal más oscura es la marca.

Un espectrograma es generado a partir de la STFT, donde la colección de DFTs que difieren por la posición de la ventana que segmenta la señal de voz, es colocada verticalmente en una imagen, asignando una columna diferente a cada segmento de tiempo. Como convención, normalmente la frecuencia aumenta de abajo hacia arriba, y el tiempo de izquierda a derecha. El valor del pixel en cada punto de la imagen es proporcional a la magnitud (o magnitud al cuadrado) del espectro en una cierta frecuencia en algún punto particular del tiempo.

Para señales cuasi-periódicas los espectrogramas son divididos en dos categorías de acuerdo a la longitud de la ventana que segmenta la señal. En voz, los espectrogramas de *banda ancha* utilizan una ventana con una longitud comparable a la de un solo periodo de pitch, lo cual resulta en una alta resolución en el dominio del tiempo, pero una baja resolución en el dominio de la

frecuencia. Este tipo de espectrogramas se caracteriza por tener bandas verticales en los intervalos de sonidos voceados (figura 3.8-b), que corresponden a las regiones de alta y baja energía dentro de un solo periodo de la señal, ya que al desplazarse la ventana, algunas veces ésta incluye muestras de gran amplitud en su mayoría y otras veces muestras de amplitud baja. Por otro lado, en los espectrogramas de *banda estrecha*, la ventana es lo suficientemente grande para capturar varios periodos del pitch durante intervalos voceados. Como resultado, el espectrograma no es muy sensible a las variaciones rápidas de tiempo, pues la resolución en el tiempo es menor para dar una mayor resolución al contenido espectral, por lo que no se muestran bandas verticales. En lugar, en este tipo de espectrogramas los armónicos de la frecuencia fundamental pueden observarse como bandas horizontales (figura 3.8-c).

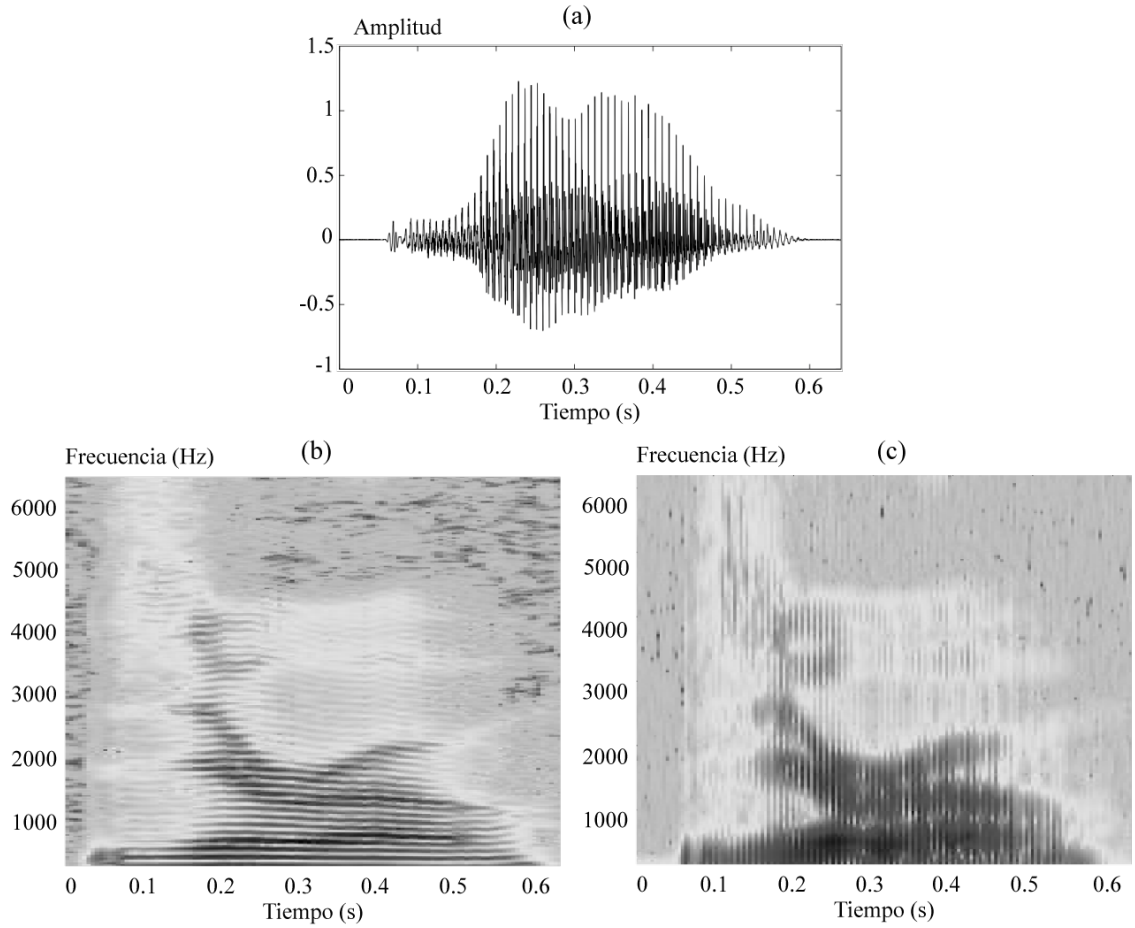


Figura 3.8: Espectrogramas de un segmento de voz. (a) Forma de onda de la palabra inglesa "zero". (b) Espectrograma de banda estrecha con ventana de 41 ms. (c) Espectrograma de banda ancha con ventana de 5 ms.

3.4.4. Síntesis de la señal de voz

La conversión de la señal de voz al dominio del tiempo a partir de su representación en un espectrograma, se lleva a cabo matemáticamente mediante una ecuación de síntesis que dependiendo de la resolución del muestreo en frecuencia, expresa la secuencia digital en términos de su STFT [39]. Desde la perspectiva de la transformada de Fourier, el método más simple para reconstruir la señal consiste en calcular para cada de tiempo de la STFT, la DFT inversa (IDFT) y dividir el resultado por la función de la ventana de análisis. Sin embargo, este método no es el más óptimo en aplicaciones prácticas ya que cualquier perturbación en la STFT puede resultar en una señal sintetizada muy diferente a la original.

Un método clásico para invertir la STFT de una señal es el método de *traslape y suma* (*OLA*, por sus siglas en inglés). En este método también se calcula la IDFT para cada trama de tiempo de la STFT, pero en lugar de dividir el resultado por la ventana de análisis, se realiza una operación de traslape y suma entre las secciones de tiempo. Este método funciona siempre y cuando la ventana de análisis esté diseñada de tal manera que la redundancia de las muestras dentro de los segmentos traslapados y el promedio de las muestras eliminen mediante la operación de traslape y suma, el efecto de la ventana de análisis.

Dada una STFT $X(n,k)$, el método OLA sintetiza una secuencia $\hat{x}(n)$ mediante la siguiente ecuación,

$$\hat{x}(n) = \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} \left[\frac{1}{N} \sum_{k=0}^{N-1} X(p,k) e^{j2\pi kn/N} \right] \tag{3.45}$$

donde $W(0) = \sum_{-\infty}^{\infty} w(n)$ y el término dentro de los paréntesis cuadrados es una DFT inversa, con la cual se obtiene para cada p,

$$f_p(n) = x(n)w(p-n) \tag{3.46}$$

siempre y cuando la DFT de longitud N sea más grande que la longitud de la ventana N_w, es decir, que no haya *aliasing* producido por la IDFT.

Ya que la IDFT es cero fuera del intervalo $[0, N)$, la expresión para $\hat{x}(n)$ se puede expresar como

$$\hat{x}(n) = \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} x(n)w(p-n) \tag{3.47}$$

que a su vez se puede reescribir como

$$\hat{x}(n) = x(n) \left(\frac{1}{W(0)} \right) \sum_{p=-\infty}^{\infty} w(p-n) \tag{3.48}$$

En la ecuación 3.48, $\hat{x}(n)$ es igual a $x(n)$ siempre que

$$W(0) = \sum_{p=-\infty}^{\infty} w(p-n) \tag{3.49}$$

3.5. Resumen

En este capítulo se han presentado las operaciones fundamentales más importantes involucradas en el procesamiento digital de señales de voz tanto en el dominio del tiempo como en la frecuencia. Se han presentado las diversas transformadas de Fourier en tiempo discreto, tales como la DTFT, la DFT y la FFT, y en el dominio del tiempo se han presentado los análisis de la señal de voz correspondientes a la energía, magnitud, cruce por ceros y la función de autocorrelación. También Se discutió la naturaleza del habla que consiste en una secuencia de diferentes eventos. Esta variación en el tiempo corresponde a características espectrales dinámicas que si son analizadas mediante una sola transformada de Fourier no es posible capturar este contenido espectral variante, por lo que se requiere de una nueva representación denominada tiempo-frecuencia, donde un mapa bidimensional muestra el comportamiento del espectro al paso del tiempo. En este nuevo significado, bajo el principio de incertidumbre debe existir un compromiso entre la resolución en el tiempo y la resolución en la frecuencia.

4

Redes Neuronales Recurrentes

Las neuronas son unas de las células más fascinantes de los organismos biológicos con sistemas nerviosos. En los humanos, son responsables de las acciones más elementales que realizamos día con día, como ver o caminar, pero también de las más elaboradas como generar pensamientos profundos sobre el origen del universo. Su complejidad nos proporciona nuestra personalidad y nuestra conciencia. Este gran poder expresivo de las neuronas es lo que ha inspirado desde el surgimiento de las computadoras el desarrollo de modelos basados en la representación matemática y computacional de estas unidades celulares fundamentales con el objetivo de replicar muchas de las capacidades del ser humano y otros seres biológicos con la mayor eficiencia posible. Aplicaciones recientes involucran robótica, control, reconocimiento de voz, visión computacional, análisis de proteínas, predicciones financieras, separación de fuentes de sonido y muchos otros problemas que involucran secuencias de datos.

4.1. Redes neuronales biológicas

El sistema nervioso humano y el de otros seres vivos está compuesto por células especializadas que se encuentran interconectadas. Estas células incluyen principalmente células nerviosas o *neuronas* y células gliales o *neuroglías*. Las neuronas son las unidades básicas fundamentales del sistema nervioso, y generan señales eléctricas que les permiten transmitir información rápidamente. En el cerebro hay una cantidad extraordinaria de neuronas y aunque pueden ser clasificadas en una gran variedad de tipos, todas comparten la misma estructura básica [40]. En el cerebro humano existen aproximadamente 100 billones de neuronas y por más de medio siglo se pensó que un trillón de células gliales, resultando en una relación neuroglías:neuronas de 10:1. No obstante, un método reciente de conteo demostró que el número de células gliales es casi proporcional al número de neuronas en el cerebro, con un total de menos de 100 billones de células gliales, cuya función principal es dar soporte a las neuronas. [41].

4.1.1. Anatomía de una neurona

En la figura 4.1 se muestra un diagrama de dos neuronas conectadas. Cada uno de los elementos que las conforman tiene un rol distinto en la generación y transmisión de señales.

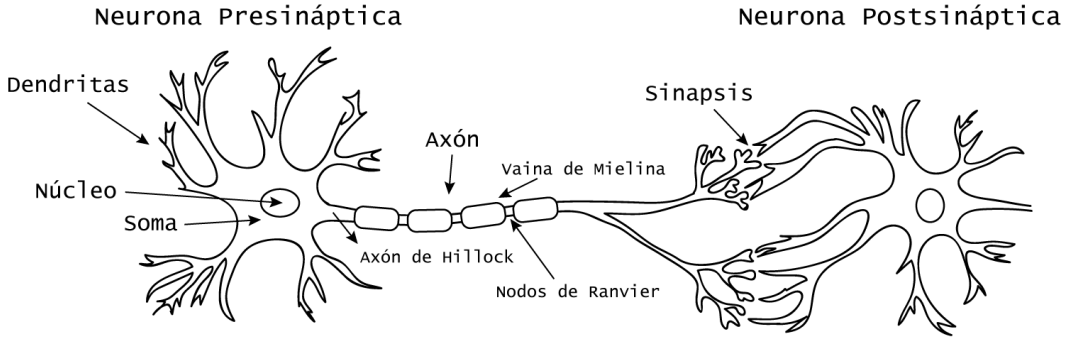


Figura 4.1: Anatomía de una neurona. Las neuronas están compuestas de un cuerpo celular también denominado *soma*, con *dendritas* que actúan como cables conectores para conectarse con otras neuronas. En la mayoría de los casos, una neurona tiene un *axón*, capaz de transmitir activamente corrientes eléctricas a otras células conectadas. Las conexiones entre neuronas son establecidas mediante *sinapsis* localizadas al final del axón. Adaptado de [40].

Como cualquier otra célula, las neuronas poseen un cuerpo celular denominado *soma*, donde se encuentra el núcleo y todos los organelos que mantienen viva a la célula. Del cuerpo de la célula emergen unas salientes ramificadas conocidas como *dendritas*, y una saliente típicamente más larga conocida como *axón*. Para comunicarse, las neuronas emplean dendritas y axones, y las regiones de conexión entre éstos se denominan *sinapsis*. A continuación se describen cada uno de los elementos fundamentales de una neurona.

- **Dendritas**: Las *dendritas*, al igual que el núcleo de la mayoría de las neuronas se encargan de recibir y procesar la información de señales procedentes de otras células. Las dendritas

tienen una estructura de árbol con ramificaciones de diámetros variables y están cubiertas por pequeñas protuberancias llamadas *espinas*. Una neurona puede tener más de un conjunto de dendritas, por lo que tiene la capacidad de recibir simultáneamente señales de miles de neuronas. La combinación de toas estas señales entrantes puede producir una respuesta binaria que genere o no un impulso nervioso que viaja a lo largo del axón.

- **Axón**: El *axón* de una neurona, a diferencia de las dendritas, mantiene el mismo diámetro en la mayoría de su longitud y no tiene espinas. El axón surge del cuerpo de la célula en un área llamada *axón de hillock* y (en varios tipos de neuronas) está cubierto con una sustancia aislante llamada *mielina*. Este recubrimiento aísla los nervios e incrementa la velocidad con la cual viajan los impulsos nerviosos. La mielina es creada por las *células de Schwann* en el sistema nervioso periférico y por *oligodendrocitos* en el sistema nervioso central. Los pequeños huecos en el recubrimiento de mielina son llamados *nodos de Ranvier*. Hacia el final, el axón se ramifica y desarrolla terminales nerviosas con formas de protuberancias bulbosas, las cuales se conectan con otras células.

  Un axón pude transmitir señales a lo largo de distancias que van desde unas cuantas fracciones de milímetros hasta longitudes de poco más de un par de metros. Estas señales se conocen como *potenciales de acción* y son impulsos eléctricos binarios, rápidos y transitorios con un valor de amplitud de 100 mV y una duración de aproximadamente 1 ms. [40]. Los potenciales de acción se generan en el axón de Hillock y desde allí son conducidos a lo largo del axón sin fallas o distorsión a una tasa de 1 a 100 metros por segundo.

- **Sinapsis**: Las neuronas se conectan unas con otras para transmitir información; sin embargo, sus fibras nerviosas no se tocan físicamente, existe siempre una brecha entre las células llamada *sinapsis*.

  Las sinapsis son los sitios en donde la información se transmite desde las fibras nerviosas de la primera neurona (*neurona presináptica*) a la siguiente (*neurona postsináptica*). Las sinapsis pueden ser de dos tipos: eléctricas o químicas. En las *sinapsis químicas* la información se transmite mediante *neurotransmisores*. Cuando un potencial de acción viaja a lo largo del axón y llega a la terminal nerviosa, se liberan neurotransmisores desde la neurona presináptica que actúan sobre las proteínas presentes en las membranas receptoras de la neurona postsináptica, transmitiendo una señal con efectos excitadores o inhibidores. En el caso de las *sinapsis eléctricas* la transmisión de información entre neuronas es virtualmente instantánea debido a que las terminales de las neuronas están conectadas a través de canales intercelulares que permiten el paso directo de iones e impulsos eléctricos; la respuesta de la neurona postsináptica al estímulo recibido de la neurona presináptica se da en fracciones de milisegundos [42]. Las sinapsis también efectúan una acción selectiva, pues en algunas ocasiones bloquean señales débiles a la vez que transmiten señales con mayor intensidad, mientras que en otras seleccionan y amplifican señales débiles que con frecuencia son transmitidas de forma multidireccional.

4.2. Redes neuronales recurrentes artificiales

De forma general una red neuronal artificial tiene como objetivo modificar la representación multidimensional de una señal de entrada aplicando tanto combinaciones de transformaciones lineales como no lineales a sus componentes. El resultado de las transformaciones puede ser empleado para obtener alguna clasificación, una predicción, una nueva señal o una representación equivalente de los datos de entrada.

Una red neuronal típica de arquitectura *feedforward* como la que se muestra en la figura 4.2 actúa como una función de mapeo, donde cada dato de entrada está asociado con una salida. En esta arquitectura convencional la información de entrada sigue un flujo de transformaciones hacia adelante a través de la red: comenzando desde los nodos de entrada, pasando por los nodos de las capas ocultas y finalizando en los nodos de salida. Esta naturaleza unidireccional del modelo obliga a que la información nunca pase dos veces por un mismo nodo, lo que supone que los datos de entrada son independientes entre sí y que si bien la red es capaz de aprender a partir de información previa, el entrenamiento no se lleva a cabo en un contexto temporal. Esto implica que las redes neuronales convencionales no tienen memoria y por consiguiente el conocimiento aprendido no puede ser compartido.

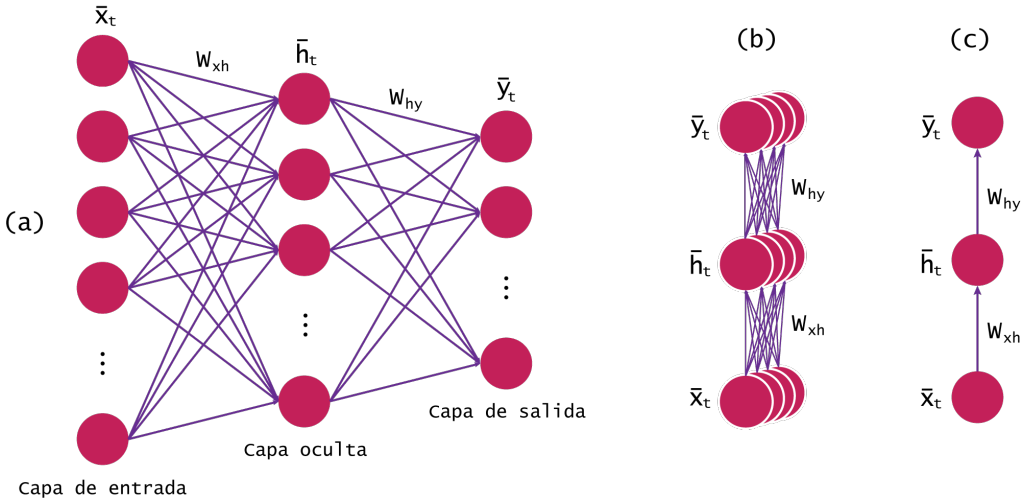


Figura 4.2: Diagrama de una red neuronal artificial feedforward. (a) Representación convencional. (b)-(c) Representación simplificada.

La abundancia de conexiones recurrentes entre las neuronas de la corteza cerebral sugiere que las neuronas en el cerebro humano no constituyen una arquitectura de red neuronal feedforward, sino más bien una arquitectura recurrente. Cabe mencionar que aunque la importancia de este modelo recurrente es muy grande, es sabido que los cálculos efectuados por una red neuronal artificial son solamente una aproximación muy vaga de los que se realizan en una red neuronal biológica [43]. A pesar de ello, desde este punto de vista dos o más entradas en la red pueden compartir conocimiento y la información puede fluir en más de una dirección. En la computación neurocientífica, esta particularidad ha permitido modelar muchos procesos

cerebrales complejos tales como la percepción, la memoria y la atención. En la inteligencia artificial, este principio es la base de muchas aplicaciones basadas en aprendizaje profundo, donde es esencial procesar la información de forma secuencial dentro de un contexto temporal. Dichas aplicaciones incluyen conversión de texto a voz, traducción automática de idiomas y transcripción de vídeos. Es evidente que el uso de las redes neuronales típicas no es adecuado para estas tareas, sin embargo, sus principios fundamentales pueden emplearse para conformar un nuevo tipo de arquitectura capaz de manejar secuencias de datos. Las redes neuronales recurrentes al igual que las redes neuronales típicas, poseen un número fijo de parámetros, pero a diferencia de éstas tienen la habilidad de procesar entradas de longitud variable y además retienen cualquier tipo de información que está relacionada con otras entradas.

4.2.1. Éxito y popularidad de las RNNs

Las redes neuronales recurrentes (RNN, por sus siglas en inglés) son relativamente antiguas. Fueron introducidas inicialmente en la década de 1980 con la invención de una arquitectura de red neuronal conocida como *red Hopfield*, inspirada en las propiedades de la memoria asociativa del cerebro humano. Años más tarde, durante la segunda mitad de la década de 1990, los investigadores Hochreiter y Schmidhuber propusieron un modelo avanzado de RNN denominado *LSTM* (*Long-Short Term Memory*, memoria de corto-largo plazo), con el objetivo de resolver los problemas principales del modelo simple [44]. No obstante, fue hasta hace unos pocos años que con el creciente poder computacional disponible y la cantidad masiva de información que tenemos hoy en día, las RNNs mostraron su potencial para resolver problemas que involucran secuencias de datos. Recientemente, en 2014, Chung et al. introdujeron una nueva mejora a la arquitectura RNN simple llamada *GRU* (Gated Recurrent Unit, unidad recurrente cerrada), la cual resuelve los mismos problemas que la arquitectura LSTM, pero de una manera más simple [45]. En cualquiera de sus variantes, la memoria interna de una RNN es capaz de retener las dependencias de los datos que recibe con el fin de lograr un entendimiento profundo del contexto de la información, lo que las convierte en una herramienta poderosa para realizar predicciones. Por esta razón son la arquitectura preferida para procesar series de tiempo, datos financieros, textos, señales de audio, vídeo, etc. Actualmente esta arquitectura robusta de redes neuronales representa el estado del arte y está detrás de los muchos de los logros más sorprendentes del aprendizaje profundo de los últimos años.

4.2.2. Los modelos ocultos de Markov como alternativa a las RNNs

Históricamente se han utilizado métodos alternativos a las redes neuronales recurrentes para resolver problemas que involucran secuencias de datos, tales como los *Modelos Ocultos de Markov* (HMM, por sus siglas en inglés). De manera general, un HMM es un modelo secuencial probabilístico cuyo objetivo es clasificar mediante etiquetas cada uno de los elementos de una secuencia de datos. El modelo calcula la probabilidad de cada posible secuencia y elige la más probable.

En la tabla 4.1 se muestran las ventajas y desventajas de los HMM con respecto a las RNNs. Ambos modelos entregan resultados excelentes, pero dependiendo de la aplicación, la cantidad de datos y recursos computacionales disponibles, un modelo suele ser mejor que el otro. En

resumen, dado un conjunto de datos muy grande y una configuración óptima de parámetros, las redes neuronales recurrentes resultan ser una mejor opción sobre los HMM. No obstante, con un conjunto de datos de tamaño limitado, los HMM representan una mejor alternativa sobre las RNNs [46].

	HMM	**RNN**
Ventajas	▪ La implementación es menos compleja que una RNN. ▪ El desempeño es tan rápido y eficiente como con una RNN en problemas de dificultad media.	▪ Su desempeño es muy eficiente y tiene un costo computacional muy bajo en tareas complejas que requieren grandes cantidades de datos.
Desventajas	▪ El costo computacional es exponencialmente caro cuando se desea aumentar la precisión de los resultados. ▪ El desempeño es lento en tareas tareas complejas que requieren grandes cantidades de datos.	▪ Establecer la configuración correcta de los parámetros de la red para resolver un problema específico es una tarea compleja. ▪ Su desempeño es muy bajo cuando se cuenta con un conjunto de datos muy pequeño.

Tabla 4.1: Comparación de las ventajas y desventajas de los HMM y las RNNs [46].

4.3. Descripción de la arquitectura RNN

A continuación se presenta la descripción de la arquitectura de red neuronal recurrente con la estructura y notación presentada en [47]. En la figura 4.3 (a) se muestra un diagrama de una RNN. A diferencia del diagrama una red neuronal convencional representada en la figura 4.2 (c), la RNN cuenta con un bucle de realimentación que ocasiona que el estado oculto de la red cambie después de la entrada de cada elemento de la secuencia. Las posiciones de los elementos de la secuencia se conocen como *marcas de tiempo*, pero este concepto de tiempo t es completamente diferente al empleado en el área de Procesamiento Digital de Señales, pues en el contexto de las redes neuronales se refiere al hecho de que cada elemento de una secuencia se procesa secuencialmente y no en paralelo [48]. La representación de la figura 4.3 (a) tiene un diagrama equivalente tal como se muestra en la figura 4.3 (b), en donde el bucle de realimentación se despliega para mostrar los elementos de la red en cada marca de tiempo. Esta representación es básicamente una simplificación de la conexión recurrente de varias redes neuronales feedfor-

ward cuyas matrices de pesos en las distintas marcas de tiempo son compartidas para aplicar las mismas operaciones a cada elemento de la secuencia. La figura clarifica de una mejor manera este concepto de recurrencia.

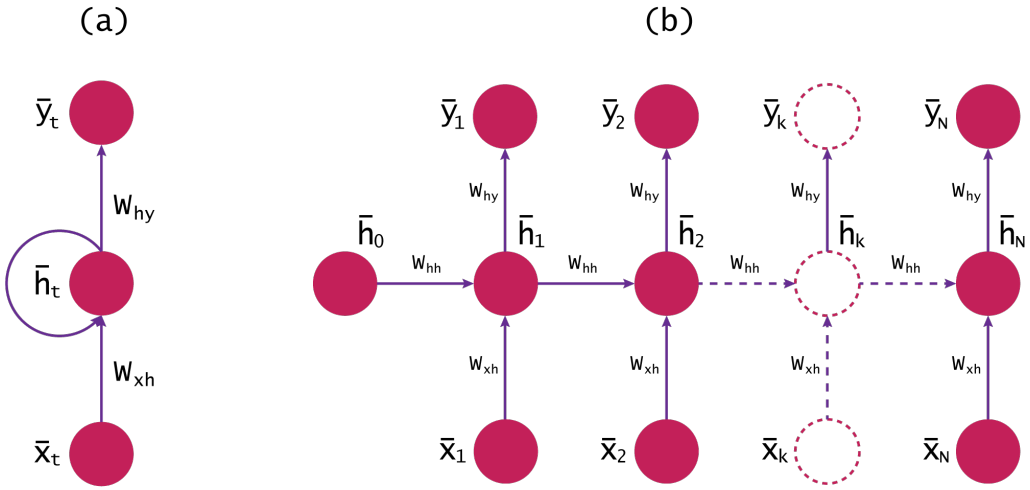


Figura 4.3: Diagrama de una red neuronal recurrente. (a) Diagrama simplificado. (b) Diagrama con el bucle desenrrollado.

La figura 4.4 sugiere que cada instante de tiempo se tiene una entrada, un estado oculto y una salida. Sin embargo, dependiendo de la aplicación en cuestión, puede existir o no una unidad de entrada o salida para cada marca de tiempo. En la tabla 4.2 se presentan las diversas variaciones de arquitectura de redes neuronales recurrentes con entradas y salidas faltantes y sus principales aplicaciones.

En el caso más simple de una red neuronal recurrente en el que están presentes todas las unidades de entrada y salida para cada marca de tiempo, el estado oculto en el tiempo t está dado en función del vector de entrada en el instante t y el vector oculto en el instante $(t-1)$, es decir:

$$\bar{h}_t = f(\bar{h}_{t-1}, \bar{x}_t) \tag{4.1}$$

Esta función (4.1) está definida por una función de activación fija $f(\cdot,\cdot)$, que actúa sobre las matrices de pesos, cuyos valores también son fijos en cada marca de tiempo, y sobre los vectores de entrada. Por otro lado, una función $\bar{y}_t = g(\bar{h}_t)$ se utiliza para aprender las salidas a partir de los estados ocultos.

La matriz W_{xh} de dimensiones $p \times d$ representa la matriz de pesos que conectan los datos de entrada con los nodos de la capa oculta, la matriz W_{hh} de dimensiones $p \times p$, representa la matriz de pesos entre las capas ocultas y la matriz W_{hy} de dimensiones $d \times p$ representa la matriz de pesos entre la capa oculta y la capa de salida. De acuerdo a lo anterior, la ecuación 4.1 puede expandirse y escribirse como:

$$\bar{h}_t = \tanh(W_{xh}\bar{x}_t + W_{hh}\bar{h}_{t-1}) \tag{4.2}$$

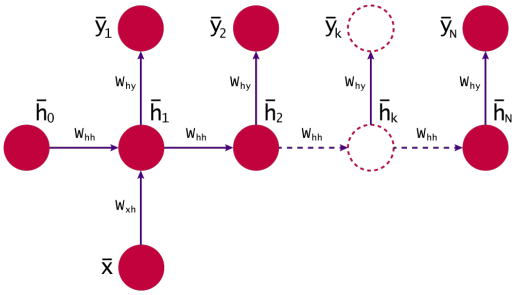
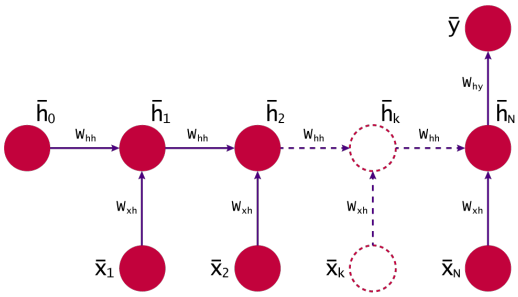
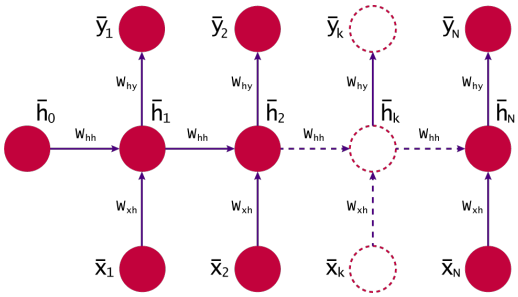
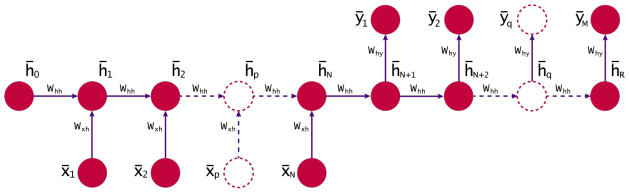
Variante	Arquitectura	Aplicaciones
Una a muchas		▪ Descripción textual ▪ Generación de música
Muchas a una		▪ Clasificación de sentimientos ▪ Predicción de la bolsa de valores
Muchas a muchas		▪ Reconocimiento de entidades nombradas
Muchas a muchas		▪ Traducción de idiomas ▪ Modelado del lenguaje

Tabla 4.2: Variantes de la Arquitectura Simple de una Red Neuronal Recurrente. [46].

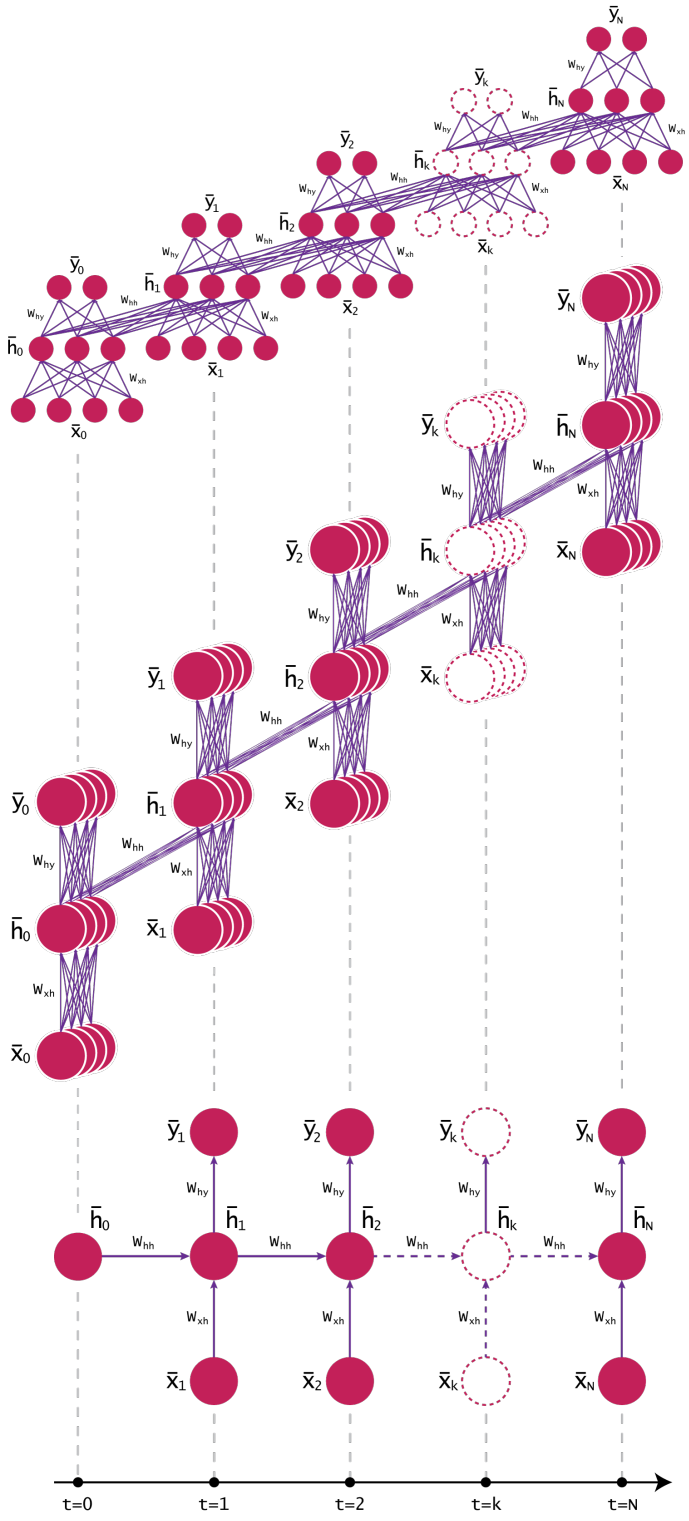


Figura 4.4: Conceptualización de una red neuronal recurrente a partir de la conexión recurrente de varias redes neuronales feedforward.

Por otra parte la condición para las salidas se escribe como:

$$\bar{y}_t = W_{hy}\bar{h}_t \tag{4.3}$$

En la ecuación 4.3 se emplea la función de activación *tanh*, que se aplica punto a punto al vector columna p-dimensional para crear otro vector de la misma dimensión con elementos en el intervalo $[-1, 1]$, sin embargo, podría utilizarse cualquier otra función de activación de la figura 4.5.

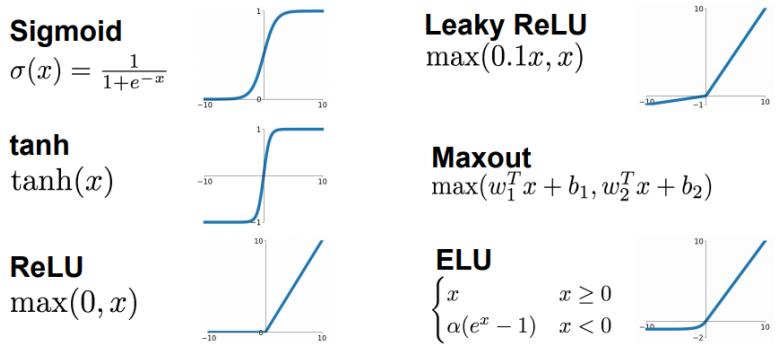


Figura 4.5: Principales funciones de activación utilizadas en las redes neuronales.

En el instante $t = 1$, correspondiente al inicio de la serie de tiempo, no existe información previa procedente de alguna capa oculta, por lo que una condición inicial es definir el estado h_{t-1} como un vector de valores constantes, aunque este vector también pueden ser aprendido en la etapa de entrenamiento. En los siguientes instantes de tiempo $t = 1, 2, \ldots, N$, los estados ocultos h_{t-i}, $i = 1, 2, \ldots, N$ cambian, pero debido a la condición recurrente de la red, las matrices de pesos permanecen constantes. El vector de salida $\bar{y}_t$ representa un conjunto de valores continuos, al cual se le puede aplicar una función de activación softmax para interpretar los resultados como probabilidades.

La ecuación 4.1 otorga a la red neuronal la capacidad de calcular una función de entradas de longitud variable. Esta recurrencia se puede expandir para definir los estados $\bar{h}_t$ en términos de t entradas. Comenzando en el estado $\bar{h}_0$, cuyo valor puede ser una constante, se tiene que,

$$\begin{aligned}
\bar{h}_1 &= f(\bar{h}_0, \bar{x}_1) \\
\bar{h}_2 &= f(f(\bar{h}_0, \bar{x}_1), \bar{x}_2) \\
\bar{h}_3 &= f(f(f(\bar{h}_0, \bar{x}_1), \bar{x}_2), \bar{x}_3) \\
\bar{h}_4 &= f(f(f(f(\bar{h}_0, \bar{x}_1), \bar{x}_2), \bar{x}_3), \bar{x}_4) \\
&\vdots \\
\bar{h}_t &= f(f(\cdots(f(f(f(\bar{h}_0, \bar{x}_1), \bar{x}_2), \bar{x}_3), \bar{x}_4), \ldots, \bar{x}_{t-1}), \bar{x}_t)
\end{aligned}$$

Nótese que $\bar{h}_1$ es una función solamente de $\bar{x}_1$, mientras que $\bar{h}_2$ es una función de $\bar{x}_1$ y $\bar{x}_2$, etc. En general, $\bar{h}_t$ es una función de $\bar{x}_1, \ldots, \bar{x}_t$.

Por otro lado, ya que la salida $\bar{y}_t$ es una función de $\bar{h}_t$, ésta se puede escribir como:

$$\bar{y}_t = F_t(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_t) \tag{4.4}$$

4.3.1. Retropropagación a través del tiempo

En las redes neuronales feedforward el algoritmo de retropropagación comienza calculando el error entre el valor obtenido a la salida de de la última capa de la red y un valor de referencia (*ground truth*). Después una porción del error se propaga capa a capa hacia atrás en dirección a las entradas. Para tale efecto, en cada paso se calculan las derivadas parciales del error con respecto los pesos $\partial E/\partial W$. Luego, mediante algún método de optimización (e.g descenso del gradiente), se emplean los resultados de las derivadas para ajustar el valor de los pesos en la dirección que reduce el error. [49].

En las redes neuronales recurrentes, al igual que en las de tipo feedforward, se puede utilizar directamente el algoritmo de retropropagación para calcular los gradientes con respecto a los pesos que conectan cada una de las capas ocultas. No obstante se debe tener en cuenta que las matrices de pesos son las mismas en cada marca de tiempo, por lo que se debe modificar el algoritmo para manejar los pesos compartidos.

La solución consiste simplemente en suponer que los parámetros en las diferentes marcas de tiempo son independientes. Bajo esta suposición, las matrices de pesos en el instante de tiempo t se definen como $W_{xh}^{(t)}$, $W_{hh}^{(t)}$ y $W_{hy}^{(t)}$. Al algoritmo de retropropagación que trabaja con esta notación se le conoce como algoritmo de *retropropagación a través del tiempo* (BPTT, por sus siglas en inglés) y fue propuesto Paul J Werbos en 1990 [50]. Los pasos del algoritmo son los siguientes:

- Secuencialmente, en cada marca de tiempo se ingresa la entrada y se calcula el error.
- Se calculan los gradientes de los pesos en la última marca de tiempo en dirección hacia atrás asumiendo independencia entre las matrices de pesos $W_{xh}^{(t)}$, $W_{hh}^{(t)}$ y $W_{hy}^{(t)}$ en cada instante de tiempo t. Utilizando el algoritmo de retropropagación de forma convencional se obtiene:
$$\frac{\partial L}{\partial W_{xh}^{(t)}}, \frac{\partial L}{\partial W_{hh}^{(t)}}, \text{ y } \frac{\partial L}{\partial W_{hy}^{(t)}}$$
- Finalmente se suman todos los pesos compartidos correspondientes a cada marca de tiempo, es decir,

$$\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_{xh}^{(t)}} \tag{4.5}$$

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_{hh}^{(t)}} \tag{4.6}$$

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_{hy}^{(t)}} \tag{4.7}$$

4.3.2. Redes neuronales recurrentes bidireccionales

La condición recurrente de las RNNs descrita por las ecuaciones 4.2 y 4.3 permite que en cada marca de tiempo t, el estado oculto h_{t-i}, $i = 1, 2 \ldots, N$ almacene información correspondiente a las entradas anteriores x_1, $x_2, \ldots x_j$, $j < i$. Sin embargo, en algunas aplicaciones como en el modelado de lenguaje o en el reconocimiento de voz es de gran utilidad analizar la información en un contexto bidireccional, tomando en cuenta también las entradas futuras de la secuencia de entrada. Bajo esta necesidad bastaría con emplear dos redes neuronales recurrentes independientes donde los elementos de la secuencia pasen sin alteraciones en el orden a través de la primera red, y pasen invertidos a través de la segunda. El problema de esta solución radica en la integración de los estados ocultos para calcular las salidas. Debido a que las redes están separadas, los estados ocultos no interactúan entre sí. Una solución más adecuada consiste en entrenar conjuntamente los estados ocultos del par de redes recurrentes independientes para obtener las salidas en función de los estados de ambas redes.

La arquitectura de red neuronal recurrente que toma en cuenta las consideraciones anteriores se conoce como *red neuronal recurrente bidireccional*. Esta red analiza para cada marca de tiempo tanto información de entradas anteriores como entradas futuras o posteriores de una secuencia, por lo que se tienen dos conjuntos independientes de estados ocultos en dos direcciones: $\bar{h}_t^{(f)}$ para las direcciones hacia adelante y $\bar{h}_t^{(b)}$ para las direcciones hacia atrás. Ya que los estados $\bar{h}_t^{(f)}$ y $\bar{h}_t^{(b)}$ son dos conjuntos separados, sus elementos solamente interactúan entre sí. No obstante, ambos conjuntos reciben como entrada el mismo vector $\bar{x}_t$. En la figura **??** se ilustra una arquitectura de red neuronal recurrente bidireccional.

Por consiguiente, en las RNNs bidireccionales se tienen matrices de parámetros en las direcciones hacia adelante y hacia atrás. Las matrices hacia adelante de los pesos que conectan la capa de entrada con la capa oculta se denotan mediante $W_{xh}^{(f)}$, los que conectan las capas ocultas mediante $W_{hh}^{(f)}$ y los que conectan la capa oculta con la capa de salida mediante $W_{hy}^{(f)}$. De manera similar, las matrices de pesos para la dirección hacia atrás se denotan mediante $W_{xh}^{(b)}$, $W_{hh}^{(b)}$ y $W_{hy}^{(b)}$, respectivamente. Las condiciones de recurrencia están dadas por las siguientes ecuaciones:

$$\bar{h}_t^{(f)} = \tanh(W_{xh}^{(f)}\bar{x}_t + W_{hh}^{(f)}\bar{h}_{t-1}^{(f)}) \tag{4.8}$$

$$\bar{h}_t^{(b)} = \tanh(W_{xh}^{(b)}\bar{x}_t + W_{hh}^{(b)}\bar{h}_{t+1}^{(b)}) \tag{4.9}$$

$$\bar{y}_t = W_{hy}^{(f)}\bar{h}_t^{(f)} + W_{hy}^{(b)}\bar{h}_t^{(b)} \tag{4.10}$$

Para esta arquitectura de red recurrente bidireccional hay un total de N marcas de tiempo, igual al número de elementos en la secuencia de entrada. En el primer y último instante de tiempo, $t = 1$ y $t = N$, los estados ocultos hacia adelante y hacia atrás no están definidos, respectivamente, por lo que se pueden inicializar con un valor constante o su valor puede ser aprendido en la etapa de entrenamiento.

Para calcular las salidas de la red, primero se introducen los datos de entrada en la dirección hacia adelante para calcular los estados ocultos en esa misma dirección. Después se vuelve a

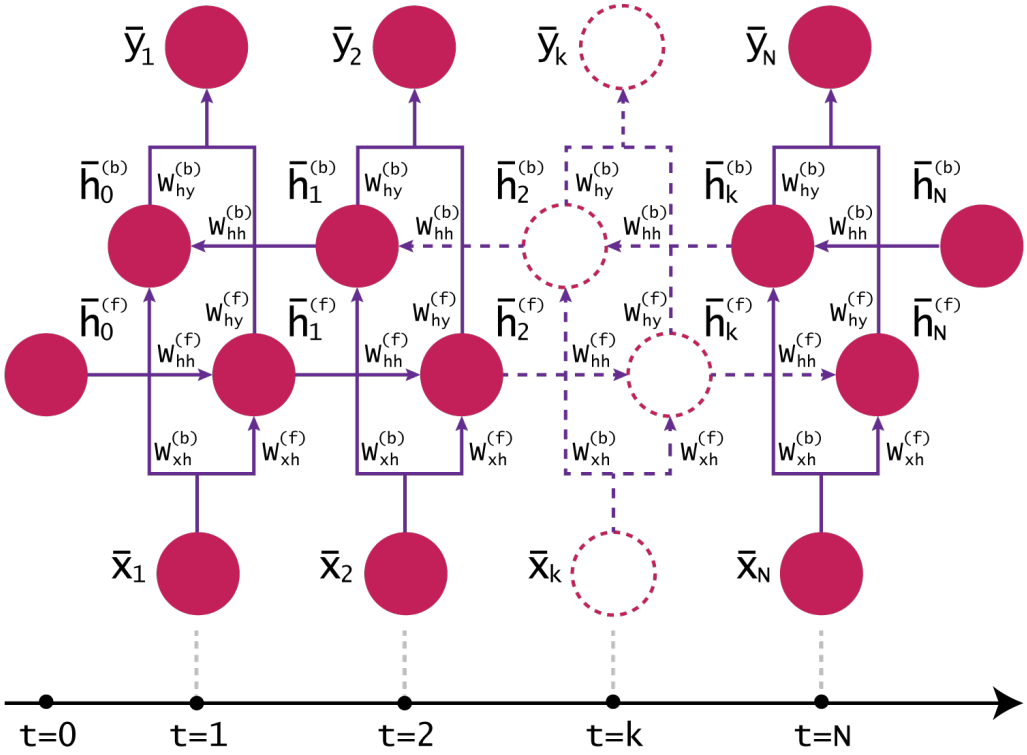


Figura 4.6: Red neuronal recurrente bidireccional.

introducir la secuencia, pero esta vez en la dirección hacia atrás. Una vez obtenidos ambos conjuntos de estados, los estados de salida se calculan a partir de la suma de los estados en ambas direcciones.

Después de calcular las salidas, se aplica el algoritmo BPTT con una modificación simple para el caso de redes bidireccionales. A continuación se resume el algoritmo:

- Se calculan los estados ocultos hacia adelante y hacia atrás en etapas separadas e independientes.
- Se calculan los estados de salida a partir de los estados ocultos obtenidos en ambas direcciones.
- Se calculan las derivadas parciales de la función costo con respecto a los estados de salida y a cada copia de los parámetros de salida.
- Se calculan las derivadas parciales de la función de costo con respecto a los estados en ambas direcciones de manera independiente empleando el algoritmo de retropropagación convencional. Estos cálculos se utilizan para evaluar las derivadas parciales con respecto a cada copia de los parámetros en ambas direcciones.
- Se suman los resultados de las derivadas parciales con respecto a los parámetros compartidos.

4.3.3. Redes neuronales recurrentes multicapa

En aplicaciones prácticas se emplean comúnmente redes neuronales recurrentes tanto convencionales (unidireccionales) como bidireccionales de múltiples capas, comúnmente de dos a cuatro capas con el objetivo de construir modelos de mayor complejidad. Emplear un número mayor de capas requiere una mayor cantidad de datos para el entrenamiento de la red para evitar sobreajuste. En la figura 4.7 se ilustra un ejemplo de red neuronal recurrente bidireccional multicapa.

En este modelo de capas múltiples, las capas superiores de la red reciben información procedente de las capas inferiores. La relación existente entre los estados ocultos puede generalizarse a partir de la ecuación 4.1, la cual puede reescribirse como

$$\begin{aligned} \bar{h}_t &= \tanh(W_{xh}\bar{x}_t + W_{hh}\bar{h}_{t-1}) \\ &= \tanh\left(W \begin{bmatrix} \bar{x}_t \\ \bar{h}_{t-1} \end{bmatrix}\right) \end{aligned} \tag{4.11}$$

donde la matriz W incluye a las matrices de pesos W_{xh} y W_{hh} y puede escribirse como

$$W = [W_{xh}, W_{hh}]$$

El conjunto de estados ocultos de cada una de las capas de la red se identifica mediante un superíndice $k, k = 2, \ldots, K$, donde K representa el número total de capas de la red. Por lo tanto, el estado oculto de la k-ésima capa en la marca de tiempo t está dado por $\bar{h}_t^{(k)}$. Análogamente, las matrices de pesos de la k-ésima capa se denotan mediante $W^{(k)}$, cuyos sus valores permanecen constantes en cada instante de tiempo t y no son compartidos entre capas.

Los estados ocultos de la primera capa, $k = 1$, reciben información de la capa de entrada en la marca de tiempo t y el estado oculto adyacente de la marca de tiempo previa $t - 1$. Por consiguiente, la dimensión de las matrices $W^{(1)}$ es de $p \times (d + p)$, donde d es la dimensión del vector de entrada $\bar{x}_t$ y p es la dimensión del vector oculto $\bar{h}_t$. La dimensión d típicamente es diferente a p. La condición de recurrencia para las capas ocultas posteriores, $k \geq 2$ se puede expresar como

$$\bar{h}_t^{(k)} = \tanh\left(W^{(k)} \begin{bmatrix} \bar{h}_t^{(k-1)} \\ \bar{h}_{t-1}^{(k)} \end{bmatrix}\right) \tag{4.12}$$

En este caso, la dimensión de la matriz $W^{(k)}$ es $p \times (p + p) = p \times 2p$. Finalmente, la transformación de la última capa oculta a la capa de salida permanece igual que en las redes recurrentes de una sola capa.

4.3.4. Problemas en el entrenamiento de las RNNs

En muchos problemas prácticos, tales como en los del área de Procesamiento Natural del Lenguaje o en el Reconocimiento de Voz, las secuencias que alimentan a una RNN pueden ser muy largas. Si las secuencias son muy largas entonces la arquitectura de la red se vuelve más

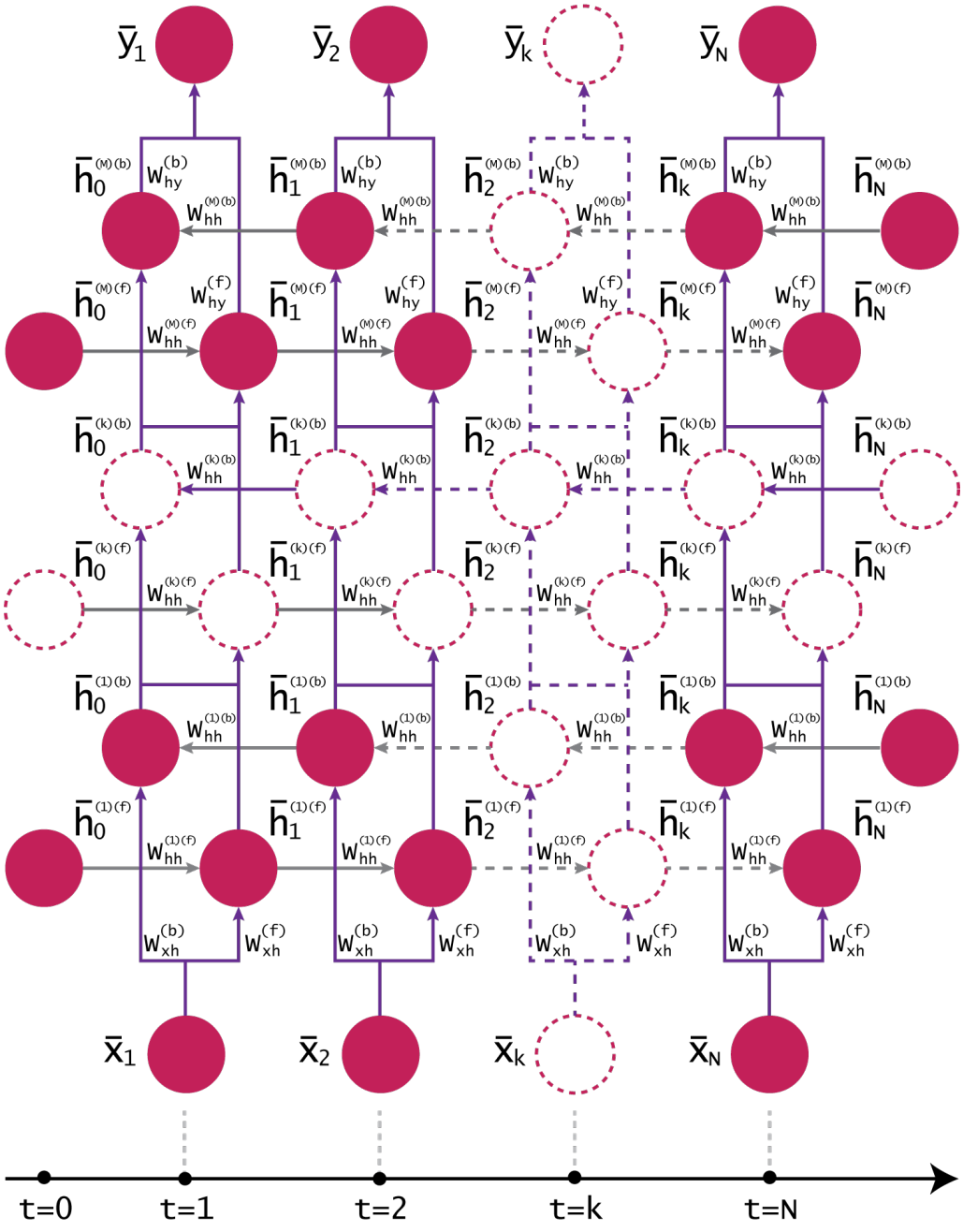


Figura 4.7: Red neuronal recurrente bidireccional de múltiples capas.

profunda. La profundidad puede pensarse como la cantidad total de marcas de tiempo o la cantidad total de elementos en la secuencia. Por consiguiente, cada vez se vuelve más complicado actualizar los valores de las matrices de pesos, en especial aquellos que se encuentran en las primeras capas de la red. En estas capas los pesos se actualizan cada vez menos debido a la

multiplicación sucesiva de valores en el intervalo $[0, 1]$, por lo que en este punto ya no se logra aprender más información de los datos de entrenamiento, o bien, el aprendizaje se lleva a cabo lentamente. A este problema se le conoce como *desvanecimiento del gradiente*.

4.3.5. Celdas LSTM

La arquitectura de RNN con celdas LSTM propuesta por Hochreiter y Schmidhuber proporcionó una solución al problema del desvanecimiento del gradiente [44] de las RNNs convencionales.

Las celdas LSTM son unidades de memoria con capacidad de almacenar información de secuencias de un mayor número de elementos, lo que permite a la red seguir aprendiendo dentro de un contexto histórico de entradas mucho más amplio. En su estructura incorporan varias compuertas que se encargan de controlar el flujo de información con la que se actualizan los estados ocultos de la red. En la figura 4.8 se muestra la estructura de una celda LSTM.

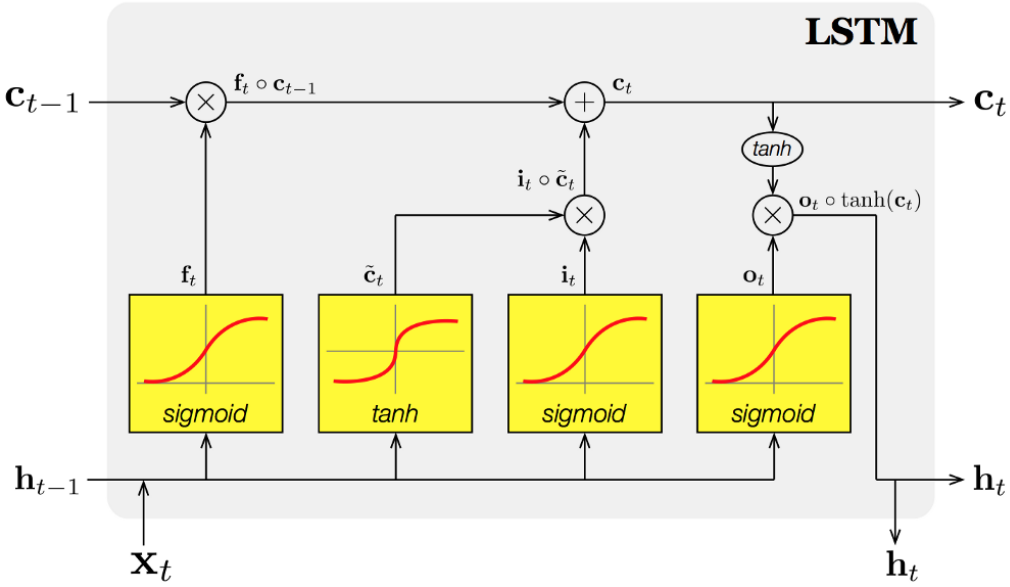


Figura 4.8: Estructura de una celda LSTM.

Esta mejora a la arquitectura RNN convencional se lleva a cabo modificando las condiciones de recurrencia de los estados ocultos descrita por la ecuación 4.1. La celda LSTM se incorpora a la arquitectura como un vector de dimensión p, denotado por $\bar{c}_t^{(k)}$. Los estados de la celda son actualizados de tal forma que el almacenamiento de la información es persistente. Esta persistencia otorga estabilidad a la red en el cálculo de los gradientes en el proceso de actualización de parámetros, por lo que de esta manera evita que se vuelvan o extremadamente pequeños o exponencialmente grandes.

Al igual que en una RNN convencional, en una arquitectura recurrente LSTM la matriz de pesos se denota mediante $W^{(k)}$ y se emplea para multiplicar el vector columna $[\bar{h}_t^{(k-1)}, \bar{h}_{t-1}^{(k)}]$.

Esta matriz es de dimensiones $4pp$, excepto para la capa de entrada ($k = 1$), cuya dimensión es $4p \times (p + d)$ y donde $\bar{h}_t^{(k-1)} = \bar{x}_t$. En el caso donde $k \geq 2$, la multiplicación de $W^{(k)}$ con un vector de tamaño $2p$ resulta en un vector de dimensión $4p$. Esto requiere cuatro vectores intermedios p-dimensionales para actualizar tanto los estados ocultos $\bar{h}_t^{(k)}$ como el estado de la celda $\bar{c}_t^{(k)}$. Conceptualmente estos vectores son considerados valores binarios por lo que son referidos como *compuertas*. En la práctica, representan compuertas binarias desde un enfoque probabilístico, ya que para llevar a cabo el entrenamiento de la red es fundamental trabajar con funciones continuas para asegurar la diferenciabilidad requerida para la obtención de los gradientes en la etapa de actualización, por lo que los elementos de los vectores intermedios no son binarios, sino más bien poseen un valor continuo en el intervalo $[0, 1]$. A continuación se define cada uno de los vectores intermedios:

Compuerta de entrada :
$$\bar{i} = \text{sigm}\left(W^{(k)} \begin{bmatrix} \bar{h}_t^{(k-1)} \\ \bar{h}_{t-1}^{(k)} \end{bmatrix}\right) \tag{4.13}$$

Compuerta de olvido :
$$\bar{f} = \text{sigm}\left(W^{(k)} \begin{bmatrix} \bar{h}_t^{(k-1)} \\ \bar{h}_{t-1}^{(k)} \end{bmatrix}\right) \tag{4.14}$$

Compuerta de salida :
$$\bar{o} = \text{sigm}\left(W^{(k)} \begin{bmatrix} \bar{h}_t^{(k-1)} \\ \bar{h}_{t-1}^{(k)} \end{bmatrix}\right) \tag{4.15}$$

Estado de la celda :
$$\bar{c} = \tanh\left(W^{(k)} \begin{bmatrix} \bar{h}_t^{(k-1)} \\ \bar{h}_{t-1}^{(k)} \end{bmatrix}\right) \tag{4.16}$$

$$\tag{4.17}$$

La actualización del estado de la celda $\bar{c}_t^{(k)}$ y el estado oculto oculto de la red $\bar{h}_t^{(k)}$ en el tiempo t se definen a partir de los vectores $\bar{i}$, $\bar{f}$ y $\bar{o}$ como,

$$\bar{c}_t^{(k)} = \bar{f} \odot \bar{c}_{t-1}^{(k)} + \bar{i} \odot \bar{c} \tag{4.18}$$

$$\bar{h}_t^{(k)} = \bar{o} \odot \tanh\left(\bar{c}_t^{(k)}\right) \tag{4.19}$$

El vector $\bar{c}$ representa la información actualizada del estado de la celda, mientras que la compuerta de entrada $\bar{i}$ y la compuerta de olvido $\bar{f}$ aseguran la persistencia de la memoria de largo plazo de la celda LSTM regulando cuánto cambia esta información con respecto al estado anterior de la celda. Básicamente estos vectores se encargan de decidir si la información de las marcas de tiempo anteriores se debe descartar, en cuyo caso se reinicia el estado de la celda y se *olvida* el pasado, o bien, si la información se debe incorporar a la memoria.

A continuación se describe de manera más detallada cada uno de los procedimientos efectuados con las compuertas de una celda LSTM, para clarificar el proceso de memorización de la información a largo plazo.

- **Compuerta de entrada**: La compuerta de entrada se encarga de "filtrar"la información de entrada en la marca de tiempo t. A través de la ecuación 4.13 decide que información tomar o descartar de la entrada actual.

- **Compuerta de olvido**: La compuerta de olvido decide que información del estado $\bar{c}_{t-1}^{(k)}$ se debe mantener o descartar. Esto se realiza mediante la ecuación 4.14 y luego multiplicando ese resultado por el estado previo de la celda $\bar{c}_{t-1}^{(k)}$. El resultado $\bar{f}$ representa la información memorizada a partir de los estados previos de la celda que aportan información útil al valor de la entrada actual.

- **Compuerta de actualización**: El siguiente paso consiste en actualizar el estado de la celda de $\bar{c}_{t-1}^{(k)}$ a $\bar{c}_{t}^{(k)}$. La memoria seleccionada del estado anterior ($\bar{f} \odot \bar{c}_{t-1}^{(k)}$) se combina la versión filtrada de la entrada en la marca de tiempo actual ($\bar{i} \odot \bar{c}$), dando lugar a la ecuación 4.18.

- **Compuerta de salida**: La compuerta de salida $\bar{o}$ decide selectivamente qué información del estado actual de la celda $\bar{c}_{t}^{(k)}$ representa el estado estado oculto en el tiempo actual. Esto se lleva a cabo mediante la multiplicación punto a punto entre la compuerta de salida $\bar{o}$ y la función tanh aplicada al estado actual de la celda $\bar{c}_{t}^{(k)}$, tal y como se establece en la ecuación 4.19.

4.4. Resumen

En este capítulo se ha presentado la arquitectura de red neuronal recurrente, la cual está inspirada en los procesos recurrentes del cerebro humano. Este tipo de arquitectura a diferencia de una red neuronal convencional de tipo feedforward, puede trabajar no solamente con datos de longitud variable, sino también es capaz de modelar las dependencias existentes entre los datos de entrada. Esta habilidad resulta ser deseada por aplicaciones como el reconocimiento de voz, la traducción automática de idiomas o el procesamiento de lenguaje natural, donde es fundamental *memorizar* las características dinámicas de la información. Se han abordado las variantes de RNNs cuando no se cuenta con la presencia de algunas unidades de entrada o salida en la arquitectura y se ha presentado la estructura de una celda LSTM capaz de retener las dependencias entre los elementos de una secuencia más larga, al mismo tiempo que evita el problema de las RNNs típicas: el desvanecimiento del gradiente. Gracias a la cantidad masiva de información existente hoy en día y al gran poder de cómputo, esta arquitectura recurrente representa actualmente una herramienta muy poderosa para efectuar tareas de predicción y clasificación. Muchos de los progresos actuales de la inteligencia artificial se han llevado a cabo con el apoyo de las redes neuronales recurrentes.

5

Diseño e Implementación del Sistema de Separación de Voz

En este capítulo se presenta el diseño de un sistema de separación de hablantes basado en el método de *Deep Clustering*, en el cual una transformación no lineal aprendida por una red neuronal profunda genera *embeddings*: una representación multidimensional de las unidades tiempo-frecuencia del espectrograma de la mezcla. La aplicación de un método de clustering en el espacio de los *embeddings* resulta en la asignación de las unidades tiempo-frecuencia a cada uno de los hablantes. Esta información es luego utilizada para estimar máscaras binarias que al ser aplicadas sobre el espectrograma de la mezcla revelan la información espectral de cada una de las señales de voz que constituyen la mezcla. Teóricamente no existe un limite para el numero de señales de voz que pueden ser estimadas a partir de este método, pero en la practica este numero esta limitado por la cantidad de datos empleada para desarrollar el sistema, la precisión que se puede alcanzar y los requerimientos computacionales y de memoria. Debido a estas limitaciones, se presentan de manera particular los elementos que integran un sistema de separación de señales de voz de dos hablantes.

5.1. Modelo de mezcla de hablantes

De manera general todas las mezclas de audio pueden considerarse lineales y convolutivas [51]. El modelo de mezcla más simple es el modelo *lineal*, el cual asume que las fuentes que integran la mezcla son sumadas linealmente. Además, si en el proceso de mezcla se modifica la amplitud de cada una de las fuentes, entonces el modelo se considera *lineal e instantáneo*.

En este capítulo se asume que todas las señales son digitales, por lo que la variable de tiempo t es discreta. También se desprecian los efectos de cuantización con el objetivo de operar sobre amplitudes continuas.

Formalmente, una mezcla de audio $x(t)$ es modelada a partir de la siguiente ecuación

$$x(t) = \sum_{j=1}^{J} a_{ij} s_j(t-k), \quad i = 1, \dots I \tag{5.1}$$

donde $a_{i,j}$ es el coeficiente de mezcla que modifica la amplitud de la fuente $s_j(t)$ en el canal i. El número total de canales y de fuentes presentes en la mezcla, se representa por I y J, respectivamente. Para el caso particular de una mezcla monoaural de audio donde $I = 1$, el índice i se puede descartar, por lo que la ecuación 5.1 se puede reescribir como

$$x(t) = \sum_{j=1}^{J} a_j s_j(t-k) \tag{5.2}$$

En el contexto de la ecuación 5.2, la separación monoaural de fuentes se refiere a la tarea de recuperar una o más fuentes de interés $s_j(t)$ dada únicamente la mezcla $x(t)$ como observación.

De forma general, el modelo de mezcla de hablantes es establecido por la ecuación 5.2 si se consideran como fuentes de sonido, $s_j(t)$, señales de voz. Para el caso particular, donde dos personas hablan simultáneamente en un solo micrófono, la señal digital de la mezcla, representada por $x(t)$, es simplemente la suma de las dos señales de voz individuales, es decir

$$x(t) = s_1(t) + s_2(t) \tag{5.3}$$

Las señales $s_1(t)$ y $s_2(t)$ en la ecuación 5.3 se suman tal cual, por lo que no están escaladas por ningún factor, es decir, los coeficientes de mezcla a_1 y a_2 son igual a 1.

Sea $S_1(\omega)$ el espectro de potencia de $s_1(t)$, el cual se define como

$$S_1(\omega) = |\mathcal{F}\{s_1(t)\}|^2 \tag{5.4}$$

donde $\mathcal{F}\{\cdot\}$ representa la transformada de Fourier, y la operación $|\cdot|^2$ calcula elemento a elemento la magnitud al cuadrado del resultado de $\mathcal{F}\{s_1(t)\}$.

De forma similar, $S_2(\omega)$ y $X(\omega)$ denotan el espectro de potencia de s_2 y $x(t)$, respectivamente. Si se supone que las señales de voz $s_1(t)$ y $s_2(t)$ no están correlacionadas entre sí, se tiene que

$$X(\omega) = S_1(\omega) + S_2(\omega) \tag{5.5}$$

La relación sugerida por la ecuación 5.5 es estrictamente válida solamente en el largo plazo y no hay garantía de que se cumpla en el cálculo de espectros de potencia medidos a partir de ventanas de análisis de longitud finita, por lo que a medida que la longitud de la ventana se incrementa, la ecuación 5.5 adquiere mayor validez.

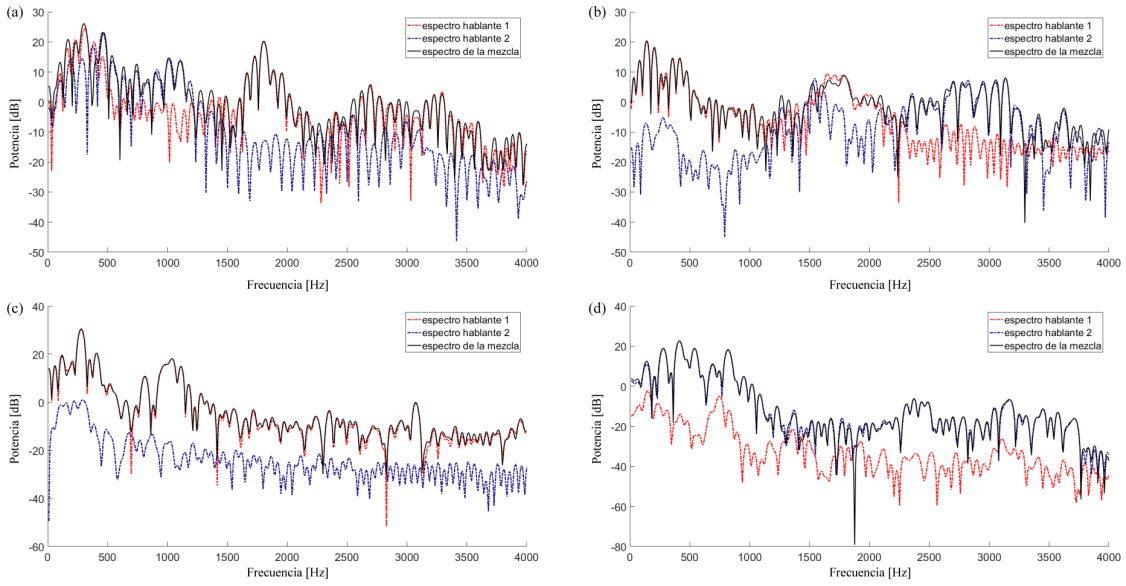


Figura 5.1: Aproximación *log-máx* en cuatro segmentos de 25 ms de una mezcla de voz de dos hablantes. En todos los casos se observa cómo el valor del espectro logarítmico de la mezcla de dos hablantes es muy cercano al valor más grande de los espectros logarítmicos de los hablantes.

Sean $s_1(\omega)$, $s_2(\omega)$ y $x(\omega)$ los logaritmos de $S_1(\omega)$, $S_2(\omega)$ y $X_(\omega)$, respectivamente. De la ecuación 5.5, se tiene que

$$x(\omega) = \log\left(e^{s_1(\omega)} + e^{s_2(\omega)}\right) \tag{5.6}$$

La ecuación 5.6 puede reescribirse como

$$x(\omega) = \max\left(s_1(\omega), s_2(\omega)\right) + \mathcal{X} \tag{5.7}$$

donde $\mathcal{X}$ se define como

$$\mathcal{X} = \log\left(1 + e^{\min\left(s_1(\omega), s_2(\omega)\right) - \max\left(s_1(\omega), s_2(\omega)\right)}\right) \tag{5.8}$$

El máximo valor de $\mathcal{X}$ es $\log(2) = 0.69$, cuando $s_1(\omega)$ y $s_2(\omega)$ son iguales. En general, es altamente improbable que ambos hablantes produzcan una cantidad idéntica de energía en la misma banda de frecuencia dentro de una ventana de análisis lo suficientemente corta. Por consiguiente, para ventanas de análisis cortas, $s_1(\omega)$ y $s_2(\omega)$ son usualmente significativamente diferentes, por lo que el valor de $\mathcal{X}$ es demasiado pequeño. Como resultado, la potencia logarítmica en cada frecuencia ω de la mezcla de señales $x(\omega)$ es dominada solamente por un hablante. La aproximación *log-máx* hace uso de esta observación al modificar la ecuación 5.7, despreciando el término

$\mathcal{X}$, es decir

$$x(\omega) = \max\left(s_1(\omega), s_2(\omega)\right) \tag{5.9}$$

A diferencia de la ecuación 5.5, la cual es válida a largo plazo, la ecuación 5.9 es aplicable principalmente a cálculos instantáneos de potencia espectral. A medida que la longitud de la ventana de análisis se incrementa, las características espectrales de los hablantes son más similares, por lo que la aproximación *log-máx* pierde validez. Por lo tanto, para determinar el tamaño de la ventana de análisis para estimar el espectro de potencia de las señales debe existir un compromiso entre los requerimientos de las ecuaciones 5.5 y 5.9.

En la figura 5.1 se ilustra la aproximación *log-máx* con la gráfica con los espectros de potencia de dos señales de voz y el espectro de potencia correspondiente a la suma de ambas señales. Se puede observar que para una ventana de análisis de 25 ms la validez de la ecuación 5.9 se mantiene en gran medida.

En general, la aproximación *log-máx* establece que los espectros logarítmicos de una mezcla de señales obtenida a partir de cualquier número de hablantes es simplemente el valor máximo del logaritmo del espectro de las señales de los hablantes individuales. En la práctica el error entre el espectro logarítmico real y el predicho por la aproximación *log-máx* es muy pequeño. Para una mezcla de N hablantes, el error máximo resulta ser $\log(N+1)$.

5.1.1. Aproximación *log-máx* en el dominio tiempo-frecuencia

En el dominio tiempo-frecuencia la aproximación *log-máx* se basa en la observación de que las señales de voz son *escasas* en este dominio conjunto, esto quiere decir que únicamente una pequeña porción de las unidades TF poseen una amplitud significativa con la cual se capta toda la energía de la señal [51]. Esta propiedad de escasez se relaciona directamente con el fenómeno de ortogonalidad disjunta-W (W-DO, W-Disjoint Orthogonality) [52], la cual establece que la probabilidad de que dos señales de voz independientes tengan una amplitud significativa en el mismo compartimento tiempo-frecuencia es muy pequeña.

De la sección anterior, la ecuación 5.3 establece que una mezcla $x(t)$ de dos hablantes $s_1(t)$ y $s_2(t)$ está dada por la suma de las señales individuales de voz

$$x(t) = s_1(t) + s_2(t)$$

Asumiendo que la señales $s_1(t)$ y $s_2(t)$ no están correlacionadas entre sí, la descomposición de la mezcla de señales en el dominio tiempo-frecuencia está dado por

$$X(t,f) = S_1(t,f) + S_2(t,f) \tag{5.10}$$

donde $X(t,f)$ es el espectrograma de la mezcla y $S_1(t,f)$ y $S_2(t,f)$ son los espectrogramas de $s_1(t)$ y $s_2(t)$, en los cuales cada una de los compartimentos tiempo-frecuencia (unidades TF) denotan el espectro de potencia del canal f y trama de tiempo t de cada uno de los hablantes.

Si se toma el logaritmo de todas las unidades TF y se usa la aproximación *log-máx* para modelar la relación entre la mezcla y las señales de voz, entonces se tiene la suposición de que en el dominio tiempo-frecuencia, la potencia de cada unidad TF del espectrograma de la mezcla corresponde a la del hablante dominante. Por consiguiente, la ecuación 5.10 puede aproximarse como

$$x(t,f) \approx \max\left(s_1(t,f), s_2(t,f)\right) \tag{5.11}$$

donde $s_1(t,f)$, $s_2(t,f)$, y $x(t,f)$ representan los logaritmos de $S_1(t,f)$, $S_2(t,f)$ y $X(t,f)$ respectivamente.

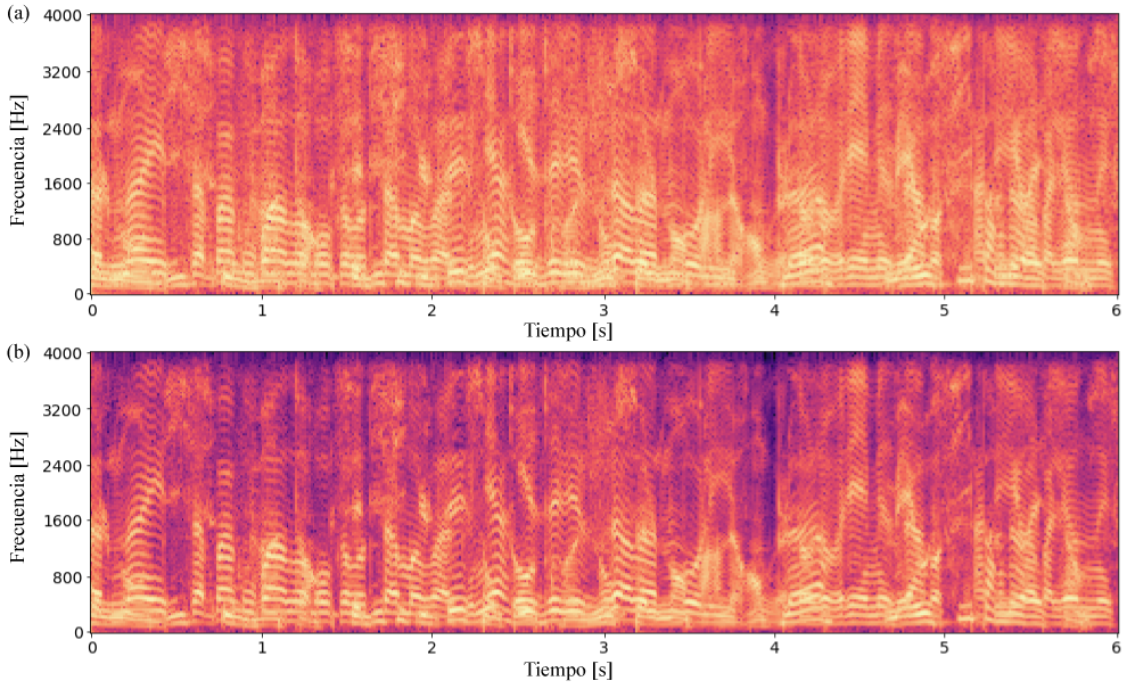


Figura 5.2: Aproximación *log-máx* en el dominio tiempo-frecuencia. (a) Espectrograma de una mezcla de dos hablantes. (b) Espectrograma de una mezcla de dos hablantes construido a partir de la aproximación *log-máx*. Cada unidad TF del espectrograma logarítmico corresponde al valor del espectro logarítmico del hablante dominante. Se puede observar que los espectrogramas (a) y (b) son similares, no obstante existe pérdida de información ya que (b) es una aproximación de (a) cuando la mezcla de hablantes representa un modelo lineal e instantáneo.

5.2. Problema de permutación y desajuste de la salida

En los sistemas de mejoramiento de la señal de voz, las propiedades estadísticas del ruido y de la voz son muy distintas, lo que permite emplear métodos basados en redes neuronales que discriminen eficientemente las fuentes, clasificándolas en dos clases. Sin embargo, en el caso de la separación de señales de voz, la red neuronal no tiene muy claro cómo realizar la tarea, ya que al tratarse únicamente de señales de voz, el procedimiento resulta contra intuitivo, ya que al ser señales de la misma clase poseen las mismas características.

Aún suponiendo que la red neuronal fuera capaz de separar eficientemente mezclas de hablantes, ésta se enfrenta al *problema de permutación y desajuste de la salida* [53]. En principio porque es difícil determinar si, por ejemplo, una mezcla de dos hablantes fue construida como $x(t) = s_1(t) + s_2(t)$ o como $x(t) = s_2(t) + s_1(t)$. Además, en el caso en que se tienen tres hablantes, s_a, s_b y s_c, y se desea separar una mezcla compuesta por los hablantes (s_a, s_b) o (s_a, s_c),

el espectrograma del hablante s_a es asignado en ambos casos a la primera posición de la salida de la red neuronal y el de los hablantes s_b y s_c a la segunda posición. Sin embargo, en caso de separar una mezcla entre los hablantes (s_b,s_c) existe una confusión, ya que para mantener la consistencia en la salida, tanto el espectrograma del hablante s_b como el del hablante s_c deben ser asignados a la segunda posición [53].

Bajo el esquema anterior es evidente que el modelo de separación no es flexible en caso de querer separar una mezcla con un número variable de hablantes, ya que la red neuronal tiene una dimensión de salida fija. En general, para poder separar N hablantes en una mezcla se requiere a la salida de la red neuronal un número de nodos fijo para N espectrogramas, uno para cada hablante presente en la mezcla.

Una de las direcciones a seguir en torno a la solución del problema de permutación y desajuste de la salida es formular el problema de manera diferente a la que siguen los sistemas de mejoramiento del habla. Generalmente estos sistemas formulan el problema de separación de fuentes como una tarea de clasificación, por lo que el entrenamiento de la red neuronal se lleva a cabo de manera supervisada a partir de ejemplos etiquetados con base en una función objetivo de la forma

$$\mathcal{C}(\Theta) = \sum_i |v_i - y_i|^2 \tag{5.12}$$

donde v_i es la predicción de la red para el ejemplo i y y_i la etiqueta de referencia del ejemplo i.

Ya que un sistema de separación de hablantes no puede seguir este esquema, en lugar de aprender las características de las señales de voz, las cuales son esencialmente las mismas por ser de la misma clase, la red neuronal puede ser entrenada para producir una nueva representación del espectrograma basándose en la forma en que está particionado. Esto quiere decir que en lugar de entrenar la red neuronal con una función objetivo como la de la ecuación 5.12, se puede entrenar con una función objetivo basada en particiones de la forma

$$\mathcal{C}(\Theta) = \sum_{i,j} f(v_i, v_j|y_i = y_j) + g(v_i, v_j|y_i \neq y_j) \tag{5.13}$$

donde el término $\sum_{i,j} f(v_i, v_j|y_i = y_j)$ compara si dos compartimentos tiempo-frecuencia del espectrograma pertenecen al mismo hablante, mientras que el término $g(v_i, v_j|y_i \neq y_j)$ compara si son hablantes diferentes. En la siguiente sección se presenta el desarrollo para obtener la nueva representación de datos.

5.3. Modelo *Deep Clustering*

El sistema de separación de hablantes desarrollado en esta tesis implementa el método *Deep Clustering* [19]. La aproximación *log-máx* y las propiedades de escasez y ortogonalidad disjunta-W de las señales de voz en el dominio tiempo-frecuencia, son el fundamento de este método basado en *clustering* que predice el hablante dominante en cada unidad TF del espectrograma y justifica la construcción de las máscaras binarias empleadas en el proceso de enmascaramiento del espectrograma de la mezcla, tal como se ilustra en la figura 5.3.

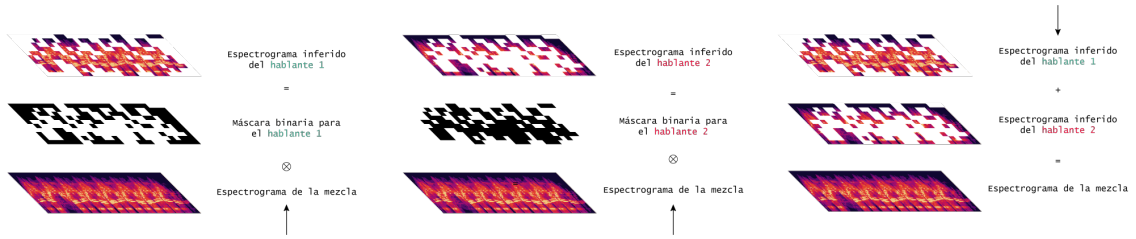


Figura 5.3: Proceso de separación de hablantes mediante el enmascaramiento del espectrograma de la mezcla. El espectrograma de la mezcla de hablantes se multiplica punto a punto con las máscaras binarias de los hablantes para estimar su información espectral.

De acuerdo a la ecuación 2, una mezcla lineal e instantánea de dos hablantes está dada por

$$x(t) = s_1(t) + s_2(t)$$

Los espectrogramas complejos de los componentes que constituyen la mezcla son calculados mediante la STFT, de manera que,

$$\begin{aligned} X(t,f) &= \mathrm{STFT}\{x(t)\} \\ S_1(t,f) &= \mathrm{STFT}\{s_1(t)\} \\ S_2(t,f) &= \mathrm{STFT}\{s_2(t)\} \end{aligned}$$

Las máscaras binarias ideales, IBM_s de los hablantes $s_1(t)$ y $s_2(t)$ se generan comparando la magnitud de su espectro en cada unidad TF

$$\mathrm{IBM}_1 = \begin{cases} 1, & \text{si} \quad s_1(t,f) > s_2(t,f) \\ 0, & \text{otro caso} \end{cases} \tag{5.14}$$

$$\mathrm{IBM}_2 = \begin{cases} 1, & \text{si} \quad s_2(t,f) > s_1(t,f) \\ 0, & \text{otro caso} \end{cases} \tag{5.15}$$

En la figura 5.4 se muestran las máscaras binarias de los hablantes presentes en la mezcla de señales de voz de la figura 5.2 empleando las ecuaciones 5.14 y 5.15.

Para simplificar la notación del dominio tiempo-frecuencia la pareja de índices (t, f) se puede expresar como $i = (t, f)$, $i \in 1, \ldots, \mathcal{N}$, donde $\mathcal{N}$ representa el número total de unidades TF del espectrograma. De acuerdo a la notación anterior

$$\begin{aligned} X_i &= X(t,f) \\ S_i &= S(t,f) \end{aligned}$$

Se busca seccionar los compartimentos i del espectrograma en conjuntos de unidades TF en los que cada hablante domina. Una vez estimadas las particiones, éstas se pueden emplear para construir máscaras binarias tal que aplicadas a la mezcla X_i se obtenga una estimación del

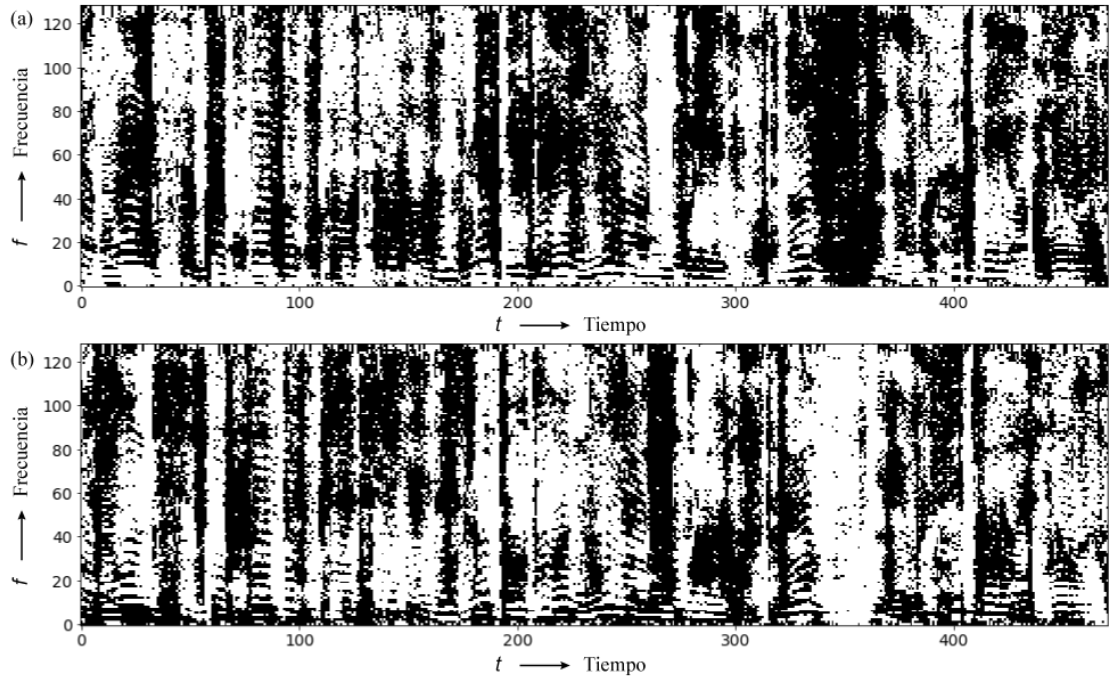


Figura 5.4: Máscaras binarias de las señales de voz de dos hablantes. Aplicadas al espectrograma de la mezcla de los dos hablantes es posible obtener una estimación del espectrograma de cada hablante.

espectrograma de cada uno de los hablantes. Posteriormente, se puede invertir la STFT para transformar las señales de voz de cada hablante al dominio del tiempo. A continuación se describe el procedimiento para conseguir tal objetivo.

A partir de las máscaras binarias, IBM_1 y IBM_2, se conforma una matriz de etiquetas, Y, que indica a qué hablante pertenece cada compartimento tiempo-frecuencia,

$$Y_{i,c} = \begin{cases} 1, & \text{si} \quad X_i \in c \\ 0, & \text{si} \quad X_i \notin c \end{cases} \tag{5.16}$$

donde $i \in 1, \ldots, \mathcal{N}$ y $c = 1, \ldots, C$, correspondiendo C al número de particiones en la mezcla, teniéndose una partición por cada hablante. Una observación importante es que las columnas de la matriz Y son ortogonales.

Luego, tal como se ilustra en la figura 5.5, con la matriz de etiquetas Y se construye una matriz de afinidad ideal binaria A, en la que se representan las asignaciones de las unidades TF a cada una de las particiones c. La matriz A de dimensiones $\mathcal{N} \times \mathcal{N}$ se calcula mediante el producto externo de Y

$$A = YY^T \tag{5.17}$$

La matriz A indica qué unidades TF del espectrograma pertenecen a la misma partición, por lo que $a_{m,n} = 1$ si los elementos m y n se encuentran en la misma partición y $a_{m,n} = 0$ si no.

Además, la matriz A tiene la propiedad de ser invariante a la permutación, es decir

$$A = (YP)(YP)^T \tag{5.18}$$

donde P es una matriz de permutación obtenida al permutar los renglones de una matriz identidad, I. En general para una mezcla de C hablantes se tienen $C!$ matrices de permutación cuadradas de tamaño $C \times C$. En el caso particular de dos hablantes, las matrices de permutación son

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{5.19}$$

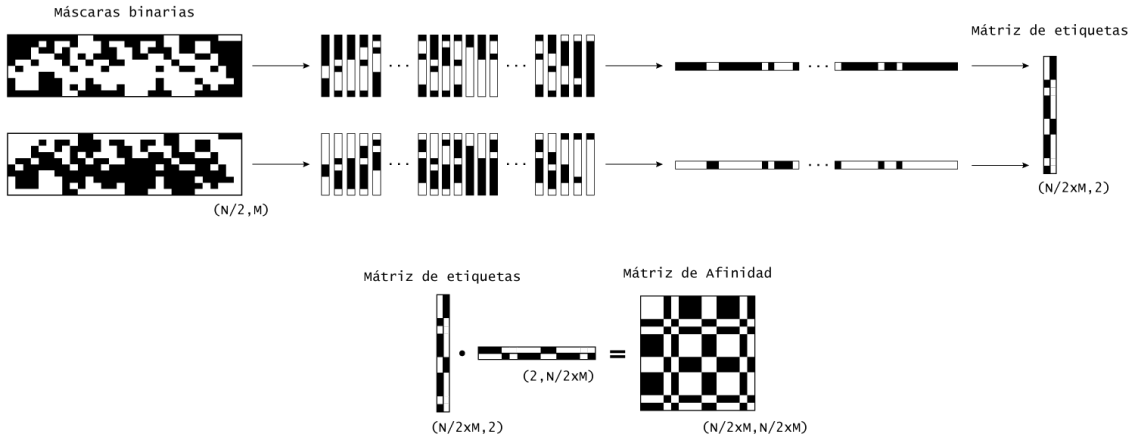


Figura 5.5: Construcción de la matriz de afinidad a partir de la matriz de etiquetas.

Para estimar las particiones se generan *embeddings* D-dimensionales mediante una función $V = f_\theta(x) \in \mathbb{R}^{N \times D}$, parametrizada por θ, tal que al aplicar un algoritmo de *clustering* en el espacio D-dimensional de los *embeddings* se encuentran las particiones de unidades TF dominadas por cada uno de los hablantes. Este hecho se fundamenta en el *teorema de Cover* acerca de la separabilidad de patrones [54], el cual establece que los problemas de clasificación de patrones tienen una mayor probabilidad de ser linealmente separables cuando son trasladados a un espacio multidimensional mediante una transformación no lineal.

La función V es obtenida mediante una red neuronal profunda (DNN) que es una función global de la mezcla X_i. La transformación V toma en cuenta propiedades globales de la entrada, y los *embeddings* resultantes pueden ser considerados como una nueva representación de las características de la mezcla en cada instante de tiempo t y frecuencia f.

Cada uno de los *embeddings* tienen norma unitaria, es decir

$$\sum_{d=1}^{D} |v_{i,d}|, \quad \forall i \tag{5.20}$$

Los *embeddings* V pueden emplearse para construir implícitamente una matriz de afinidad estimada $\hat{A} = VV^T$ de dimensiones $\mathcal{N} \times \mathcal{N}$.

La matriz de afinidad $\hat{A} = VV^T$ es aprendida por la red de tal forma que coincida con la matriz de afinidad ideal $A = YY^T$, minimizando la siguiente función costo con respecto a V

$$\mathscr{C}_Y(V) = \left\|\hat{A} - A\right\|_F^2 \tag{5.21}$$

$$= \left\|VV^T - YY^T\right\|_F^2 \tag{5.22}$$

$$= \sum_{i,j}(\langle v_i, v_j\rangle - \langle y_i, y_j\rangle)^2 \tag{5.23}$$

$$= \sum_{i,j:y_i=y_j}(|v_i - v_j|^2 - 1) + \sum_{i,j}\langle v_i, v_j\rangle^2 \tag{5.24}$$

donde $\|\cdot\|_F^2$ es el cuadrado de la norma de Frobenius.

En la ecuación 5.24, el término $\sum_{i,j:y_i=y_j}(|v_i-v_j|^2-1)$ permite a los *embeddings* permanecer juntos si se encuentran en la misma partición, mientras que el término $\sum_{i,j}\langle v_i, v_j\rangle^2$ los mantiene alejados si pertenecen a particiones diferentes.

La función costo $\mathscr{C}_Y(V)$ suma todos los pares de puntos de datos i, j, sin embargo, su naturaleza de bajo rango conlleva a una implementación eficiente,

$$\mathscr{C}_Y(V) = \left\|V^TV\right\|_F^2 - 2\left\|V^TY\right\|_F^2 + \left\|Y^TY\right\|_F^2 \tag{5.25}$$

En la práctica, $\mathcal{N}$ es de órdenes de magnitud más grande que D, por ejemplo, a partir de una señal de audio de 10 segundos muestreada a 8 kHz, si se calcula un espectrograma de magnitud de 129 puntos empleando una STFT de ventanas de 256 muestras con traslape de 176 muestras (10 ms), se tienen $\mathcal{N} = 129,000$ unidades TF, por lo que la matriz de afinidad estimada cuenta con alrededor de 17 billones de entradas. La ecuación 5.25 evita construir explícitamente las matrices de afinidad de tamaño $\mathcal{N} \times \mathcal{N}$, llevando a una aceleración significativa. Las derivadas con respecto a V son también obtenidas eficientemente debido a la estructura de bajo rango de $\mathscr{C}_Y(V)$:

$$\frac{\partial \mathscr{C}_Y(V)}{\partial V^T} = 4V(V^TV) - 4Y(Y^TV) \tag{5.26}$$

En la etapa de inferencia, los *embeddings* $V = f_\theta(x)$ se calculan a partir de la mezcla a separar y los renglones $v_i \in \mathbb{R}^D$ son agrupados mediante el algoritmo de k-medias.

5.4. Detector de unidades TF activas

Procesar todas las unidades tiempo-frecuencia del espectrograma puede llevar a dos situaciones críticas durante el proceso de entrenamiento de la red neuronal: la primera es que se asignen *embeddings* a las regiones con baja energía del espectrograma y la segunda es que en presencia de ruido, los *embeddings* además de representar las características de la señal de los hablantes, también representan implícitamente las características del ruido. Estas dos situaciones se pueden evitar mediante el uso de un detector de unidades TF activas, con el cual se impide la asignación de *embeddings* a regiones silenciosas del espectrograma, aligerando la carga computacional en las etapas de entrenamiento e inferencia, y también se descarta el ruido presente en las regiones ausentes de información espectral específica de las señales de voz.

En la implementación del sistema para conformar el detector de unidades TF activas, se empleó un umbral en decibles de -40 dB con respecto al pico máximo de potencia del espectrograma de la mezcla, reteniendo únicamente aquellas unidades TF en las cuales el valor de potencia supera el valor del umbral. Por consiguiente, el detector de actividad de unidades TF puede ser visto como una máscara binaria de tamaño $N \times M$ cuyo valor es 1 si $x(m,n) > \max(x(t,f)) - 40$ dB, donde $x(m,n) = 20\log(|X(m,n)|)$, correspondiente a la unidad TF de la m-ésima trama y n-ésimo canal de frecuencia del espectrograma, y $x(t,f) = 20\log(|X(t,f)|)$ corresponde al espectrograma de la mezcla $x(t)$. El valor de la máscara es 0 si la condición anterior no se satisface. En la figura 5.6 se ilustra la aplicación del detector de unidades TF activas para determinar las regiones activas del espectrograma.

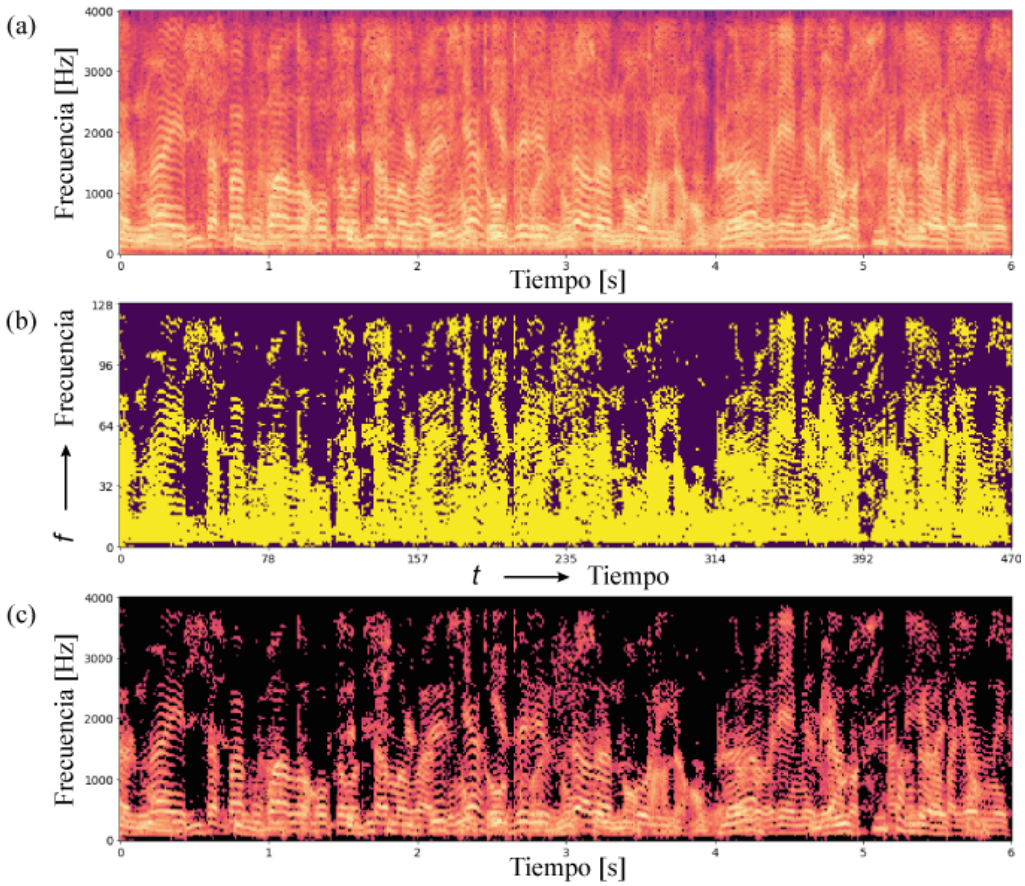


Figura 5.6: Máscaras de unidades TF activas (b). Al aplicarse sobre el espectrograma de la mezcla (a) se segmentan las unidades TF útiles para en las etapas de entrenamiento e inferencia de la red (c).

5.5. Datos para entrenamiento del modelo

El corpus de voz TED-LIUM 3 [55], fue el corpus elegido para la generación de las mezclas de hablantes y para el entrenamiento y validación del sistema. En la tabla 5.1 información relativa al número de hablantes presentes en el corpus, así como la duración total del conjunto de archivos de audio.

Elementos del corpus TED-LIUM 3		
Duración [hrs]	Total	452
	Hombres	316
	Mujeres	134
Duración media [min]		11.5
Número de hablantes		2028
Número de charlas		2351

Tabla 5.1: Características del corpus TED-LIUM 3 [55].

Del corpus TED-LIUM 3 se eligió un subconjunto de archivos de audio para crear los *datasets* de entrenamiento, validación y prueba del modelo *Deep Clustering*, cuya duración se especifica en la tabla 5.2.

Archivos de audio para entrenamiento del modelo		
Duración [hrs]	Total	10
	Hombres	5
	Mujeres	5
Entrenamiento [hrs]		6
Validación [hrs]		3
Evaluación [hrs]		2

Tabla 5.2: Características de los archivos de audio para entrenamiento del modelo.

5.5.1. Generación de las mezclas de hablantes

Las mezclas de audio utilizadas para el entrenamiento de la red neuronal se generaron con la información del *dataset* de entrenamiento obtenido del corpus TED-LIUM 3, a partir de la suma ponderada de la señales de voz de dos hablantes, a partir del siguiente procedimiento analítico. Primero las señales de los hablantes son normalizadas respecto a su valor máximo absoluto de amplitud y después su amplitud es modificada por un factor k tal que el SNR calculado entre la señal normalizada y la señal de amplitud modificada se encuentra entre -3 y 3 dB. El factor de

amplitud k es calculado a partir de la definición de la relación señal a ruido

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log \left(\frac{P_s}{P_n} \right) \tag{5.27}$$

Particularmente la relación P_s y P_n es la relación entre la potencia de la señal de amplitud modificada P_{mod} y la potencia de la señal normalizada, P_{norm}. Dicha relación es igual al factor k, es decir

$$k = \frac{P_{\mathrm{mod}}}{P_{\mathrm{norm}}} \tag{5.28}$$

por consiguiente, la ecuación 5.27 se puede reescribir en términos del factor k como

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log (k) \tag{5.29}$$

Despejando k de la ecuación 5.29 se obtiene el factor de amplitud en términos del valor SNR en decibeles deseado

$$k = 10^{\left(\frac{\mathrm{SNR}_{\mathrm{dB}}}{20} \right)} \tag{5.30}$$

Finalmente, la modificación de la amplitud de las señales de voz está dada por

$$s_{\mathrm{mod}}(t) = \sqrt{k} \cdot s_{\mathrm{norm}}(t) \tag{5.31}$$

Una vez obtenido el valor k, el valor del SNR deseado se puede comprobar mediante la ecuación

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log \left(\frac{P_{\mathrm{mod}}}{P_{\mathrm{norm}}} \right) \tag{5.32}$$

donde $-3 < \mathrm{SNR}_{\mathrm{dB}} < 3$

La potencia de la señal de amplitud modificada P_{mod} se calcula como

$$P_{\mathrm{mod}} = \frac{1}{N} \sum_{n=0}^{N-1} |s_{\mathrm{mod}}(t)|^2 \tag{5.33}$$

o bien, despejando P_{mod} de la ecuación 5.28 como

$$P_{\mathrm{mod}} = k \cdot P_{\mathrm{norm}} \tag{5.34}$$

De acuerdo al procedimiento anterior, la mezcla de hablantes se expresa como

$$x(t) = \tilde{s}_1(t) + \tilde{s}_2(t) \tag{5.35}$$

donde

$$\tilde{s}_1(t) = \sqrt{k_1} \cdot s_{1_{\mathrm{norm}}}(t) \tag{5.36}$$

$$\tilde{s}_2(t) = \sqrt{k_2} \cdot s_{2_{\mathrm{norm}}}(t) \tag{5.37}$$

5.6. Arquitectura de la red neuronal profunda

La arquitectura de la red neuronal profunda (DNN) fue implementada en *Tensorflow* [56] y es consistente con la arquitectura presentada en [19]. La DNN consiste en cuatro capas de redes neuronales recurrentes bidireccionales con celdas de largo-corto plazo (LSTM) seguida de una capa feedforwad (FFN). Cada capa BLSTM contiene 600 celdas ocultas y la capa FFN corresponde con la dimensión D de los *embeddings*.

En el proceso de entrenamiento, para la actualización de los pesos de la red neuronal profunda se emplea un método de optimización basado en descenso estocástico del gradiente como momentum 0.9 y una tasa de aprendizaje fija de 10^{-5}. Los pesos fueron inicializados de forma aleatoria de acuerdo a una distribución normal con media cero y varianza 0.1 y en cada paso de actualización se les añadió a los pesos ruido Gaussiano con media cero y varianza 0.6 para evitar mínimos locales.

5.7. Inferencia de las señales de voz

La inferencia de las señales de voz presentes en una mezcla de hablantes sigue de manera general los procesos ilustrados en el diagrama de bloques de la figura 5.7. La estimación de las señales de voz se realiza en el dominio tiempo-frecuencia con sustento en la aproximación *log-máx*. El primer paso en la separación de hablantes en una mezcla monoaural de audio consiste en obtener el espectrograma de la mezcla de hablantes. A continuación se describe el procedimiento que sigue una mezcla de hablantes lineal e instantánea $x(t) = \sum_{j=1}^{J} s_j(t-k)$ para ser transformada al dominio tiempo-frecuencia como $X(t,f)$.

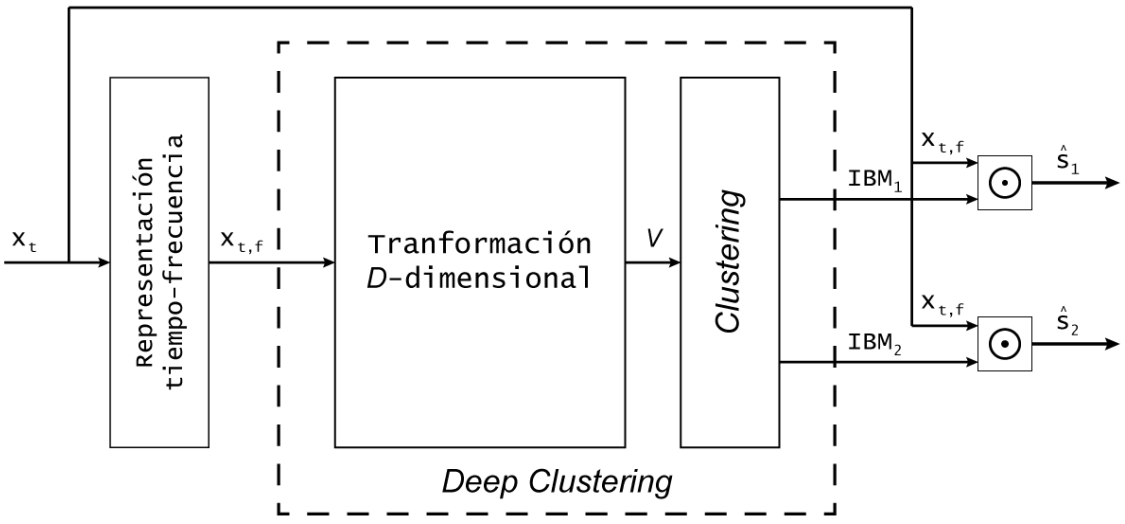


Figura 5.7: Diagrama de bloques de la inferencia de señales de voz en una mezcla de dos hablantes.

5.7.1. Representación tiempo-frecuencia

La intensidad de la señales de voz presentes en la mezcla de hablantes $x(t)$ no es uniforme en todas las frecuencias de su ancho de banda, ya que ésta disminuye aproximadamente 6 dB por octava a medida que la frecuencia aumenta. Esta caída espectral es causada naturalmente por el pulso glotal y la radiación de los labios. Con el objetivo de compensar la intensidad de las frecuencias altas de la señal de voz se emplea un filtro de pre-énfasis correspondiente a un filtro de respuesta finita al impulso (FIR) paso altas de primer orden, el cual es comúnmente utilizado en sistemas de reconocimiento de voz: $x(t) \longrightarrow x(t) - 0.95x(t-1)$.

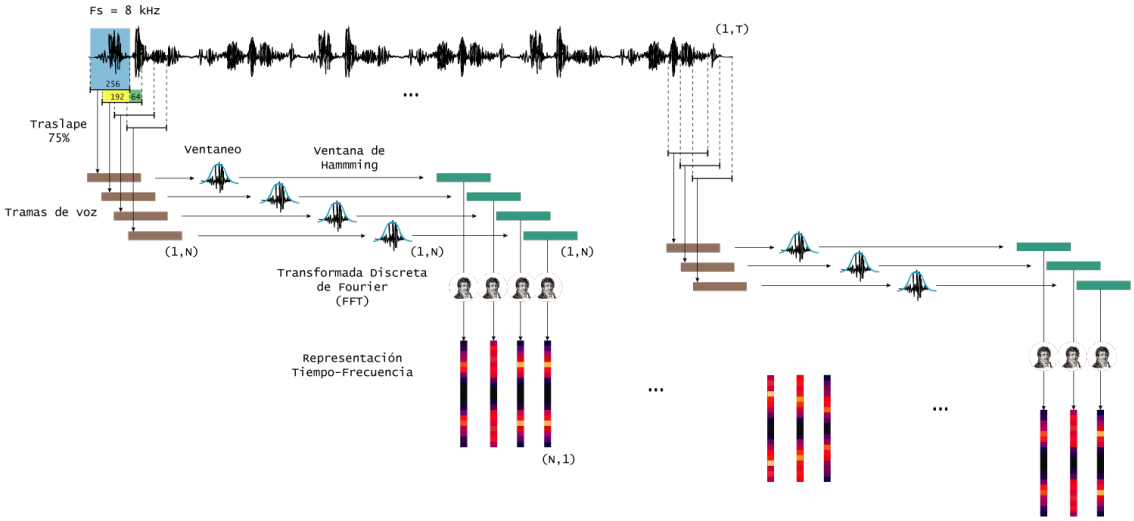


Figura 5.8: Diagrama de la obtención del espectrograma de la mezcla de hablantes.

Las señales de audio correspondientes a las mezclas de hablantes poseen una frecuencia de muestreo $F_s = 8$ kHz. La señal se particiona en tramas de corta duración para mantener sus propiedades estadísticas cuasi-estacionarias. Cada trama tiene 256 muestras, lo que es equivalente a una duración de 32 ms. Las tramas se encuentran traslapadas un $75\,\%$, lo que corresponde a un desplazamiento de 8 ms y una superposición de 192 muestras. Cada una de las tramas de la señal es multiplicada por una ventana de Hamming para suavizar las discontinuidades abruptas en los bordes y evitar la aparición de componentes espectrales no deseados.

Después, las tramas de la señal de voz son transformadas al dominio de la frecuencia mediante la transformada discreta de Fourier (DFT), la cual se calcula eficientemente mediante la transformada rápida de Fourier (FFT). Para el cálculo de la FFT se emplean 256 puntos, los cuales coinciden con el número de muestras por trama. Del cálculo resulta una transformación simétrica de 256 componentes frecuenciales con una resolución de 31.25 Hz.

Finalmente, se calcula el logaritmo de la magnitud del espectro para hacer valida la aproximación log-max, con la que cada unidad TF es dominada por un hablante.

Ya que la señal $x(t)$ es una señal real, su transformación al dominio de la frecuencia a partir de la FFT resulta en un vector simétrico, solamente se conservan $N/2+1$ puntos para el proceso

de transformación D-dimensional y *clustering*, pues $N/2$ puntos son redundantes. Esta simplificación del espectrograma se ilustra en la figura 5.9 Por otra parte, la información concerniente a la fase, dada por la ecuación 5.38

$$e^{j\angle X(t,f)} = \frac{X(t,f)}{|X_(t,f)|} \tag{5.38}$$

se calcula y almacena para emplearse posteriormente en la etapa de reconstrucción de las señales de los hablantes. En la figura 5.8 se muestra el diagrama de la obtención del espectrograma de la mezcla.

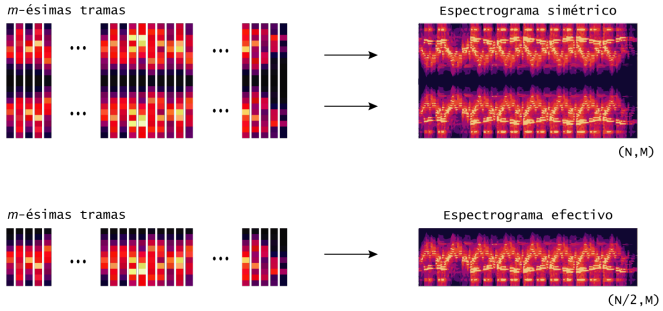


Figura 5.9: Simplificación del espectrograma.

5.7.2. Transformación multidimensional y *clustering*

Una vez transformada la mezcla de hablantes $x(t)$ al dominio tiempo-frecuencia $X(t,f)$, se procede a procesar la información con la red neuronal profunda.

Se calcula el espectro logarítmico de potencia $x(t,f) = 20\log|X(t,f)|$, y de acuerdo al detector de unidades TF activas, se seleccionan únicamente las unidades TF cuya magnitud es superior a la proporción seleccionada, en este caso -40 dB respecto al pico máximo de potencia del espectrograma logarítmico.

Luego, cada una de las unidades TF activas del espectrograma es trasladada a un espacio D-dimensional mediante la función aprendida por la red neuronal profunda, $V = f_\theta(x(t,f))$, con la cual se generan los *embeddings* D-dimensionales. En la figura 5.10 se ilustra este proceso.

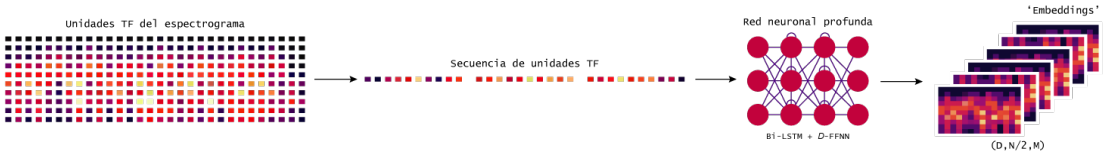


Figura 5.10: Generación de los *embeddings*.

Con el algoritmo de k-medias se transforman los *embeddings*, correspondientes a los renglones $v_i \in \mathbb{R}^D$, en un conjunto de etiquetas de particiones $c = 1, \ldots C$, donde $C = 2$, teniéndose

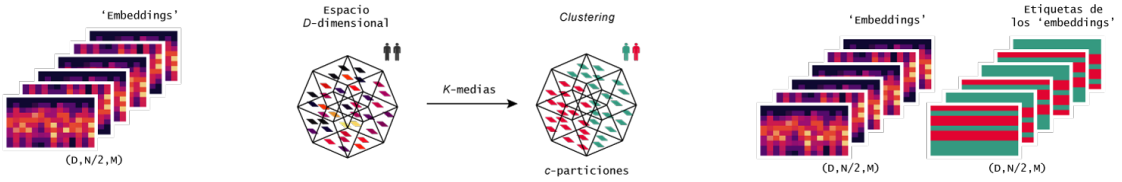


Figura 5.11: Transformación multidimensional y clustering.

una partición por cada hablante, de manera que cada *embedding* tiene asignada la etiqueta del hablante al que pertenece. En la figura 5.11 se muestra el proceso de transformación y *clustering*.

La asignación de particiones resultante es utilizada después para construir las máscaras binarias útiles para estimar el espectrograma de las señales de voz de cada uno de los hablantes. Este proceso se ilustra en la figura 5.12.

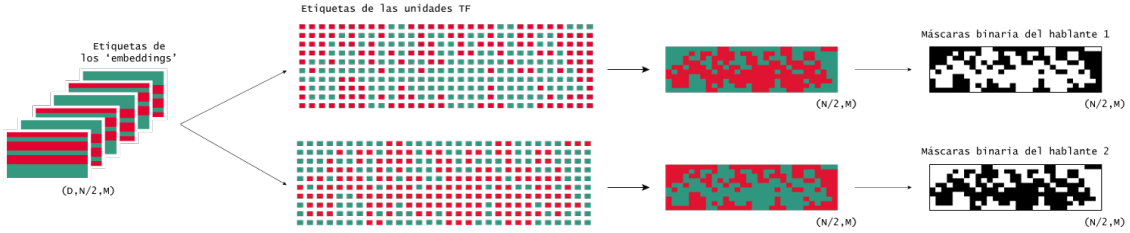


Figura 5.12: Generación de las máscaras binarias.

5.7.3. Reconstrucción de las señales de voz

Después de la obtención de las máscaras binarias para estimar el espectro de cada uno de los hablantes presentes en la mezcla, la transformación de las señales al dominio del tiempo produce las señales de voz estimadas $\hat{s}_c(t)$, $c = 1 \dots C$. Este proceso se se lleva a cabo de acuerdo al procedimiento ilustrado en la figura 5.13.

El proceso de reconstrucción se realiza trama a trama, multiplicando la m-ésima trama de la mezcla de hablantes con la m-ésima trama de la máscara binaria del hablante c cuya señal de voz se desea estimar. Este par de tramas se multiplica punto a punto, operación de la cual resulta la m-ésima del espectrograma estimado, $\hat{s}_c(t, f)$.

El espectrograma estimado del hablante c representa un espectrograma de potencia logarítmico, es decir, $\hat{s}_c(t, f) = 20 \log(|\hat{S}_c(t, f)|)$ por lo cual es convertido a un espectrograma de magnitud lineal mediante la siguiente operación

$$|\hat{S}_c(t, f)| = 10^{\frac{\hat{s}_c(t,f)}{20}} \tag{5.39}$$

Ya que por cada trama de análisis se tienen $N_{ef} = 129$ puntos de frecuencia correspondientes al espectrograma efectivo, es necesario reconstruir los 128 puntos restantes junto con la fase de la señal de la mezcla de hablantes, para tener un espectrograma inferido de $N = 256$ puntos de valores complejos con el cual se pueda transformar la señal de voz al dominio del tiempo mediante el cálculo de la ISTFT.

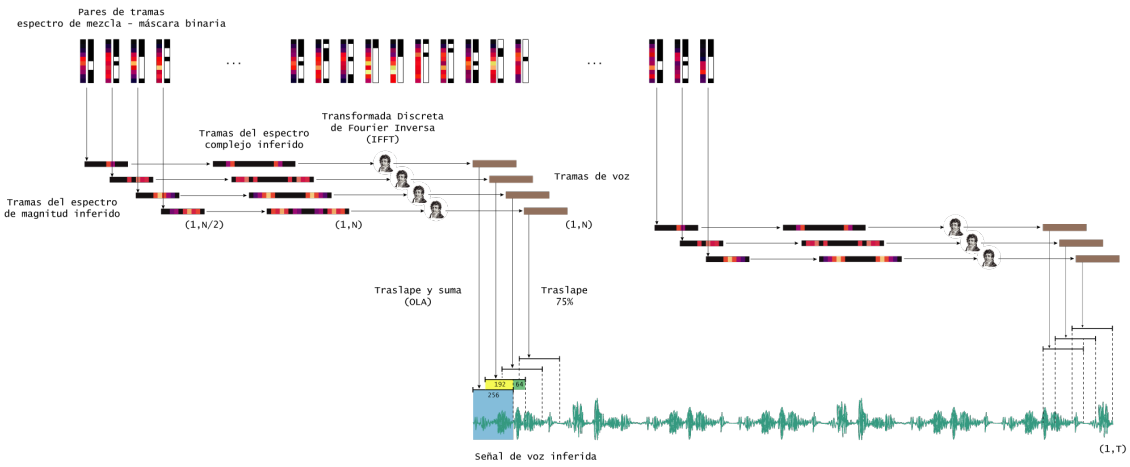


Figura 5.13: Reconstrucción de las señales de voz estimadas.

De acuerdo a la propiedad de escasez y al fenómeno de la ortogonalidad disjunta-W , ya que solamente un hablante domina en cada unidad TF, la fase de la mezcla es típicamente cercana a la de la hablante dominante. Esta es la motivación principal por la cual se asigna la fase de la mezcla a la fuente de voz estimada, de manera que el espectrograma estimado de valores complejos, $\hat{S}_c(t, f)$, es obtenido mediante la siguiente ecuación

$$\hat{S}_c(t, f) = |\hat{S}_c(t, f)| \angle X(t, f) \tag{5.40}$$

Para reconstruir la señal del hablante c, se calcula la transformada de Fourier inversa (IFFT) de la m-ésima trama de $\hat{S}_c(t, f)$ para producir

$$\hat{s}_m(t), \quad t = 0, \ldots, N-1 \tag{5.41}$$

Después, consecutivamente las tramas de tiempo se sobreponen y suman mediante el método de traslape y suma (OLA), traduciendo la m-ésima trama de salida al tiempo mR como

$$\hat{s}_m(t) = \hat{s}_m(t - mR) \tag{5.42}$$

donde $R = 64$, corresponde al tamaño de salto, o bien, al traslape de 75 % empleado en el cálculo de la STFT para la obtención del espectrograma.

Finalmente, $\hat{s}_m(t)$ se suma a la señal acumulada de salida $\hat{s}_c(t)$.

5.8. Resumen

En este capítulo se presentaron las consideraciones teóricas y prácticas para el diseño de un sistema de separación de hablantes en mezclas monoaurales de audio basado en aprendizaje profundo. De forma general, el sistema descrito en este capítulo se fundamenta en la aproximación *log-máx*, que establece que cada unidad tiempo-frecuencia del espectrograma de la mezcla es dominada por un hablante. La propiedad de escasez de las señales de voz en el dominio tiempo-frecuencia y el fenómeno de la ortogonalidad disjunta-W son la clave para la utilidad práctica de esta aproximación: existe una probabilidad muy baja de que dos señales independientes tengan el mismo valor de potencia en la mismos compartimentos tiempo-frecuencia. Bajo estos principios, se emplea el método de *Deep Clustering*, cuyo objetivo es trasladar la información del espectrograma de la mezcla a un espacio multidimensional mediante una función aprendida por una red neuronal profunda Bi-LSTM, cuyo entrenamiento se lleva a cabo optimizando una función costo cuya naturaleza de bajo rango permite, junto con su gradiente, ser formulada para ser computacionalmente eficiente. Los *embeddings* generados por la red neuronal son particionados por el algoritmo de *clustering* de K-medias y la información de las particiones encontradas es útil para generar las máscaras binarias que aplicadas al espectrograma de la mezcla revelan la información espectral de cada uno de los hablantes. Finalmente, con el método de traslape y suma, trama a trama se transforman al dominio del tiempo las señales de voz inferidas.

6

Pruebas y Resultados

En este capítulo se describen las condiciones experimentales bajo las cuales se lleva a cabo la evaluación del sistema de separación de hablantes implementado en esta tesis. Dichas condiciones evalúan el sistema de acuerdo al dominio de la información de prueba e incluyen diversas configuraciones de redes neuronales profundas a fin de explorar la arquitectura óptima para la tarea de separación. Se presentan resultados de separación de mezclas de hablantes de todas las combinaciones de género, mostrando una comparativa entre los espectrogramas de referencia y los espectrogramas estimados por el sistema, y objetivamente, la calidad de la separación de las señales de voz se realiza de una manera simple e intuitiva mediante el uso de las métricas de desempeño de separación ciega de fuentes denominada `BSS_eval`, las cuales miden la distorsión de las señales estimadas y su relación con las interferencias remanentes y artefactos producto del algoritmo de separación.

6.1. Configuración experimental

El sistema de separación de hablantes implementado en este trabajo de tesis se evaluó bajo dos condiciones experimentales. En la primera condición, denominada *condición cerrada* (CC), las mezclas de audio empleadas para la evaluación del sistema incluyeron hablantes cuyas voces fueron utilizadas para entrenar el modelo. Por otro lado, en la segunda condición, denominada *condición abierta* (CA) se emplearon mezclas de hablantes cuya información no fue incluida en el conjunto de datos de entrenamiento. Como se mencionó en el capítulo 1, los métodos tradicionales de separación de hablantes fueron desarrollados bajo condiciones cerradas, ya que son métodos que dependen de la información de las voces involucradas en las mezclas. Sin embargo, la novedad del método *deep clustering* permite realizar la tarea de separación en condiciones abiertas, es decir, independientemente de los hablantes aparezcan en una mezcla de audio, por lo que no se necesita información a priori de los hablantes para estimar las señales de voz.

Se generaron diversas arquitecturas de redes neuronales para la generación de modelos de separación de hablantes, todas fueron entrenadas para generar *embeddings* de $D = 40$ dimensiones. El objetivo de las variaciones en la arquitectura, tal como el número de capas en la red Bi-LSTM o el número de neuronas por capa, fue encontrar una arquitectura de red neuronal profunda de parámetros óptimos para realizar la tarea de separación bajo la condición abierta, la cual se asemeja a condiciones reales y es por lo tanto la más relevante para evaluar el sistema en términos de la calidad de separación. Las configuraciones implementadas se especifican a continuación

- **300x2**: Red neuronal Bi-LSTM de 2 capas con 300 celdas LSTM en cada dirección.
- **600x2**: Red neuronal Bi'LSTM de 2 capas con 600 celdas LSTM en cada dirección.
- **300x4**: Red neuronal Bi-LSTM de 4 capas con 300 celdas LSTM en cada dirección.

En la evaluación de las configuraciones anteriores se presentan también los resultados de un sistema denominado **IBM** de separación de hablantes basado en máscaras binarias ideales, cuyo objetivo es proporcionar una referencia del mejor resultado de separación posible que se podría obtener en la tarea de separación de hablantes bajo las mismas condiciones de prueba.

El conjunto de datos para evaluar el sistema de separación de hablantes en ambas condiciones experimentales consistió en 100 mezclas de dos hablantes para cada una de las combinaciones de género: hombre-mujer (MF), hombre-hombre (MM) y mujer-mujer (FF); 100 mezclas con las combinaciones conjuntas del mismo genero (FF/MM) y finalmente el promedio de todas las posibles combinaciones de genero (AVG) para evaluar el desempeño general del sistema.

6.1.1. Entorno de desarrollo

La implementación de las arquitecturas de redes neuronales se llevó a cabo empleando la plataforma para la creación de modelos de aprendizaje de máquina de código libre *TensorFlow* v1.10 con soporte para GPU. La computadora que se utilizó para el entrenamiento y pruebas del modelo cuenta con un procesador intel i7-4790 con 8 Gb de memoria RAM y una tarjeta gráfica NVIDIA GeForce GTX con 6 Gb de memoria RAM. El sistema operativo de la computadora de

trabajo es Ubuntu 16.04 y los requerimientos de software incluyeron la instalación de los controladores de la tarjeta gráfica y la paquetería de CUDA para el correcto funcionamiento del entorno de desarrollo. Inicialmente se realizaron pruebas de entrenamiento de los modelos en la computadora con las características de hardware descritas anteriormente, pero sin la tarjeta gráfica. El tiempo de entrenamiento de los modelos utilizando un solo CPU tomó varias semanas, sin embargo, la integración de la GPU aceleró el tiempo de entrenamiento de los modelos, reduciendo el tiempo de entrenamiento a solamente unas pocas horas.

6.2. Entrenamiento

La arquitectura de red neuronal base empleada para la generación de los modelos de separación de hablantes consistió en una red neuronal recurrente bidireccional (Bi-LSTM). Como se mencionó en la configuración experimental se emplearon tres configuraciones basadas en dicha arquitectura. La elección de las variaciones respecto al número de capas y al número de celdas LSTM en cada dirección fue basada en las configuraciones presentadas en [57] con el objetivo de encontrar un modelo óptimo de separación. A continuación se presentan las curvas de aprendizaje de las tres configuraciones de redes neuronales implementadas y posteriormente el resultado de la evolución de los *embeddings* en el espacio 40-dimensional a medida que progresa el entrenamiento.

6.2.1. Curvas de aprendizaje

A continuación se presentan las curvas de aprendizaje por cada una de las configuraciones de red neuronal. El objetivo de estas curvas es proporcionar una medida objetiva de la evolución del aprendizaje de la red neuronal al paso del tiempo. Por lo que el cálculo de las curvas para cada configuración indica qué arquitectura de red neuronal obtuvo un mejor desempeño en la tarea de separación de hablantes.

Las *curvas de entrenamiento* fueron calculadas a partir del conjunto de datos de entrenamiento, lo que da una idea de qué tan bien está *aprendiendo* la red. Por otro lado, las *curvas de validación*, que se calculan a partir del conjunto de datos de validación, dan una idea de qué tan bien está *generalizando* el modelo.

En el entrenamiento de redes neuronales que realizan tareas de clasificación comúnmente se utilizan funciones costo que se puedan minimizar, de forma que un número muy cercano a cero indica un aprendizaje óptimo por parte de la red neuronal. Bajo este principio, un número igual a cero indica que el conjunto de datos de entrenamiento fue aprendido perfectamente y que no se realizó ningún error. Sin embargo, en el caso del entrenamiento del sistema de separación implementado, ya que la función costo utilizada genera un modelo basado en particiones, y no en clasificación, la minimización de dicha función muy difícilmente se aproxima a cero, ya que intuitivamente, en la ecuación 5.21 de la función costo, se intenta generar una matriz binaria de millones de entradas con una matriz de números reales. Por lo cual, a pesar de que el valor de la función costo decrementa con el tiempo, dicho valor es muy grande.

En la figura 6.1 se muestra los pares de curvas de aprendizaje para cada una de las configuraciones de redes neuronales implementadas. Se puede observar que el modelo más óptimo en la tarea de separación de hablantes es la configuración 300x4 correspondiente a la arquitectura

de red red neuronal Bi-LSTM de cuatro capas con 300 neuronas por capa. El modelo óptimo de esta configuración es alcanzado en la *epoch* 289, a partir de la cual el modelo comienza a experimentar un problema de *sobreajuste*.

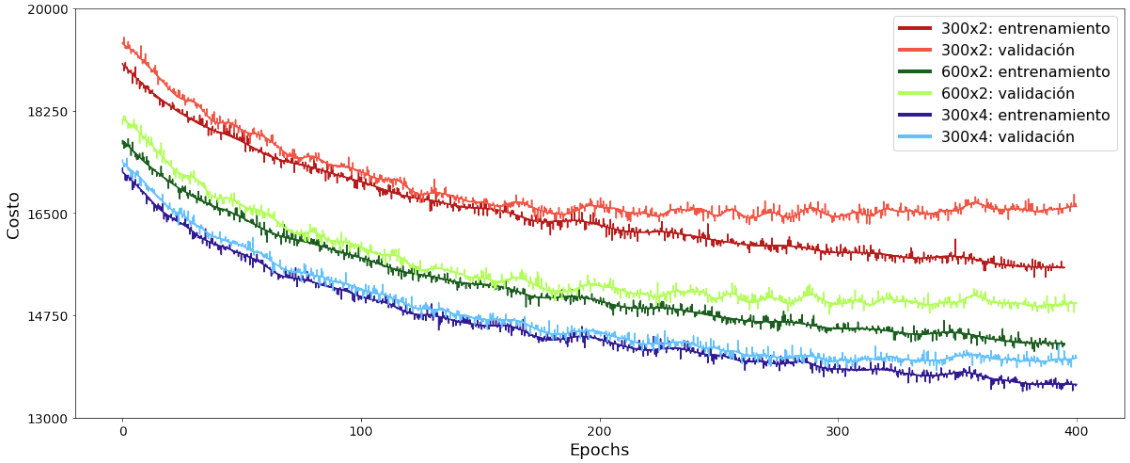


Figura 6.1: Curvas de aprendizaje de las configuraciones de redes neuronales implementadas. La configuración 300x2 genera un modelo óptimo en la *epoch* 189, donde alcanza un valor de costo de 16900. Por otro lado, la configuración 600x2 genera un modelo más óptimo aproximadamente en el mismo número de *epochs* (192) alcanzando un valor de costo de 1510. La tercera configuración: 300x4 genera el modelo más óptimo, alcanzando un valor de costo de 14100, no obstante requiere de un mayor número de *epochs* para su entrenamiento (284).

6.2.2. Visualización del espacio multidimensional

Como se mencionó en el capítulo 5, el modelo *deep clustering* tiene como objetivo realizar una transformación multidimensional de las unidades tiempo-frecuencia del espectrograma de una mezcla de hablantes, de tal forma que los *embeddings* generados se agrupen juntos en el espacio D-dimensional si pertenecen al mismo hablante o se alejen unos de otros si corresponden a hablantes diferentes. En la figura 6.2 se muestra una evolución de la transformación del las unidades TF de una mezcla de dos hablantes (hombre + mujer) al espacio de los *embeddings* de $D = 40$ dimensiones. A medida que progresa el entrenamiento de la red neuronal profunda de arquitectura óptima (red recurrente bidireccional de 4 capas y 600 neuronas por capa). La visualización de los *embeddings* se lleva a cabo calculando los primeros tres componentes principales (PCA) del espacio de 40 dimensiones.

Una vez entrenado el modelo de separación, el algoritmo de k-medias es aplicado en el espacio de los *embeddings* para etiquetar las unidades tiempo-frecuencia correspondientes a cada uno de los hablantes y posteriormente generar las máscaras binarias que permiten estimar el espectrograma de cada uno de los hablantes presentes en la mezcla. En la figura 6.3 se muestra el resultado de la aplicación del algoritmo de k-medias aplicado en el espacio de los embeddings.

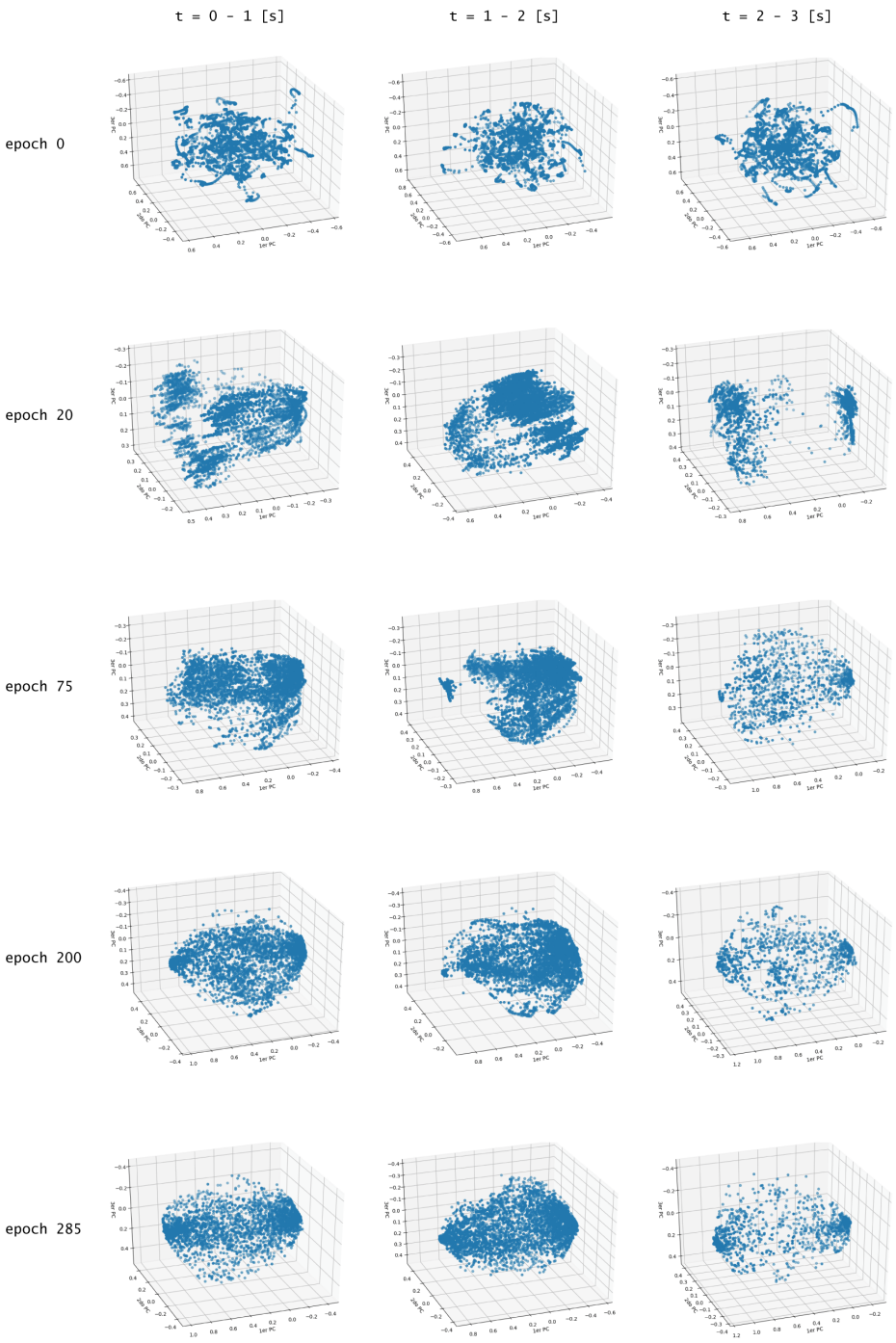


Figura 6.2: Evolución los *embeddings* en el espacio multidimensional calculados sobre una mezcla de hablantes de tres segundos de duración. A medida que el número de *epochs* incrementa, los *embeddings* pertenecientes a un mismo hablante se agrupan juntos en el espacio.

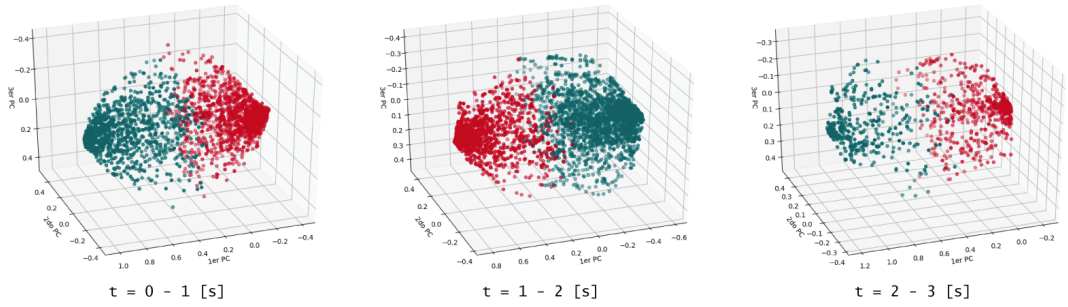


Figura 6.3: *Clustering* de los *embeddings* de una mezcla de hablantes de tres segundos de duración en el espacio multidimensional mediante el algoritmo de k-medias.

6.3. Evaluación del sistema

La evaluación del sistema de separación de hablantes implementado se llevó a cabo empleando las medidas de desempeño para separación ciega de fuentes propuestas en [22]. Éstas son medidas numéricas objetivas que miden la relación entre las señales de referencia y las señales estimadas por el sistema de separación. La validez de estas métricas está fundamentada en la forma en que los seres humanos percibimos las señales de voz, razón por la cual se propone la descomposición de la señal estimada a la salida del sistema de separación de hablantes como la suma de la señal deseada, las señales de interferencia, el ruido y artefactos, es decir

$$\hat{s}_j = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}} \tag{6.1}$$

donde s_{target} es la señal estimada por el sistema de separación de hablantes del j-ésimo hablante. Los términos e_{interf}, e_{noise} y e_{artif}, representan respectivamente los términos de error de las interferencias, ruido y artefactos. A continuación se presenta el procedimiento para obtener los componentes de la ecuación 6.1 de acuerdo a la descripción de [1].

La descomposición de $\hat{s}_j$ está basada en proyecciones ortonormales $\Pi\{\cdot\}$, los cuales proyectan la señal $\hat{s}_j$ en el subespacio de la señal deseada. Se consideran los siguientes tres proyectores ortogonales

$$P_{s_j} = \Pi\{s_j\}, \tag{6.2}$$
$$P_s = \Pi\{(s_{j'})_{1\leq j'\leq N}\}, \tag{6.3}$$
$$P_{s,\eta} = \Pi\{(s_{j'})_{1\leq j'\leq N},\ (\eta_i)_{1\leq i\leq M}\} \tag{6.4}$$

Con el conjunto de ecuaciones anteriores, la señal estimada $\hat{s}_j$ se descompone como la suma de los siguiente cuatro términos

$$s_{\text{target}} = P_{s_j}\hat{s}_j, \tag{6.5}$$
$$e_{\text{interf}} = P_s\hat{s}_j - P_{s_j}\hat{s}_j, \tag{6.6}$$
$$e_{\text{noise}} = P_{s,\eta}\hat{s}_j - P_s\hat{s}_j, \tag{6.7}$$
$$e_{\text{artif}} = \hat{s}_j - P_{s,\eta}\hat{s}_j \tag{6.8}$$

Con el conjunto de ecuaciones anteriores, la señal estimada $\hat{s}_j$ se descompone como la suma de los siguiente cuatro términos

$$s_{\text{target}} = P_{s_j}\hat{s}_j, \tag{6.9}$$
$$e_{\text{interf}} = P_s\hat{s}_j - P_{s_j}\hat{s}_j, \tag{6.10}$$
$$e_{\text{noise}} = P_{s,\eta}\hat{s}_j - P_s\hat{s}_j, \tag{6.11}$$
$$e_{\text{artif}} = \hat{s}_j - P_{s,\eta}\hat{s}_j \tag{6.12}$$

Ya que s_{target} es la proyección de la señal estimada en el subespacio de la señal deseada, se puede emplear el producto interno para su cálculo, por lo tanto

$$s_{\text{target}} = \frac{\langle \hat{s}_j, s_j\rangle s_j}{||s_j||^2} \tag{6.13}$$

Si las señales estimadas y las señales de referencia son mutuamente ortogonales, entonces

$$e_{\text{interf}} = \sum_{j'\neq j} \frac{\langle \hat{s}_j, s_{j'}\rangle s_{j'}}{||s_{j'}||^2} \tag{6.14}$$

si no,

$$P_s\hat{s}_j = \sum_{j'=1}^{N} \tilde{c}_{j'}s_{j'} = \mathbf{c}^H\mathbf{s} \tag{6.15}$$

donde $\mathbf{c} = \mathbf{R}_{\mathbf{ss}}^{-1}[\langle \hat{s}_j, s_1\rangle, \ldots, \langle \hat{s}_j, s_N\rangle]^H$, siendo $\mathbf{R}_{\mathbf{ss}}$ la matriz de Gram de las fuentes por $(\mathbf{R}_{\mathbf{ss}})_{jj'} = \langle s_j, s_{j'}\rangle$. Por consiguiente, e_{interf} está dado por

$$e_{\text{interf}} = \mathbf{c}^H\mathbf{s} - s_{\text{target}} \tag{6.16}$$

Si se considera que las señales de referencia son ortogonales a las señales de ruido, entonces $P_{s,\eta}$ se puede obtener como

$$P_{s,\eta}\hat{s}_j \approx P_{s_j}\hat{s}_j + \sum_{i=1}^{M} \frac{\langle \hat{s}_j, \eta_i\rangle \eta_i}{||\eta_i||^2} \tag{6.17}$$

A partir de lo cual

$$e_{\text{noise}} = \sum_{i=1}^{M} \frac{\langle \hat{s}_j, \eta_i\rangle \eta_i}{||\eta_i||^2}, \tag{6.18}$$
$$e_{\text{artif}} = \hat{s}_j - \mathbf{c}^H\mathbf{s} - e_{\text{noise}} \tag{6.19}$$

La descomposición de $\hat{s}_j$ permite definir un criterio de desempeño numérico mediante el cálculo de proporciones de energía expresadas en decibeles (dB). Por lo tanto, se definen cuatro

medidas inspiradas en la definición usual de la relación señal a ruido (SNR), pero con algunas modificaciones

$$\text{Relación Señal a Distorsión:} \quad \text{SDR} = 10\log_{10}\frac{||s_{\text{target}}||^2}{||e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}||^2} \tag{6.20}$$

$$\text{Relación Señal a Artefactos:} \quad \text{SAR} = 10\log_{10}\frac{||s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}||^2}{||e_{\text{artif}}||^2} \tag{6.21}$$

$$\text{Relación Señal a Ruido:} \quad \text{SNR} = 10\log_{10}\frac{||s_{\text{target}} + e_{\text{interf}}||^2}{||e_{\text{noise}}||^2} \tag{6.22}$$

$$\text{Relación Señal a Interferencias:} \quad \text{SIR} = 10\log_{10}\frac{||s_{\text{target}}||^2}{||e_{\text{interf}}||^2} \tag{6.23}$$

A continuación se muestran los resultados de los experimentos de separación de hablantes llevados a cabo de acuerdo a la configuración experimental descrita en la sección 6.1, en términos del promedio de la relación señal a distorsión (SDR), la relación señal a artefactos (SAR) y la relación señal a interferencias (SIR).

6.3.1. Condición abierta

En esta condición la evaluación del sistema se realizó con información de prueba que no formó parte del entrenamiento, es decir, las mezclas utilizadas para las pruebas de separación incluyen hablantes que no fueron incluidos en el entrenamiento de la red neuronal, pero que sí pertenecen al corpus de entrenamiento. Al ser esta condición la más representativa de un ambiente real (estilo fiesta de cóctel) el análisis de las métricas de calidad de separación de las señales de voz se detalla de acuerdo a las distintas combinaciones de género de los hablantes. Los resultados de la evaluación se muestran en la figura 6.1.

	FM			FF			MM			FF/MM			AVG		
conf	SDR	SAR	SIR	SDR	SAR	SIR	SDR	SAR	SIR	SDR	SAR	SIR	SDR	SAR	SIR
IBM	14.8	16.5	23.7	13.4	14.1	18.2	13.6	14.5	17.2	13.5	14.3	17.7	14.1	15.4	20.7
300x2	9.3	11.1	14.8	7.1	7.7	11.2	6.9	7.3	10.8	7.0	7.5	11.0	8.2	9.3	12.9
600x2	10.1	11.7	16.4	7.6	8.5	11.5	7.4	8.1	11.7	7.5	8.3	11.6	8.8	10.0	14.0
300x4	**11.4**	**14.9**	**19.3**	**8.6**	**11.3**	**14.4**	**8.2**	**10.8**	**13.9**	**8.4**	**11.1**	**14.2**	**9.9**	**13.1**	**16.8**

Tabla 6.1: Evaluación del sistema en condición abierta.

Los resultados de la tabla 6.1 muestran que el sistema tiene un mejor desempeño cuando se realiza separación de hablantes de distinto género, mientras que en el caso de la separación de hablantes del mismo género (MM, FF), las mejoras de las relaciones SDR, SAR y SIR son muy semejantes. En general, las configuraciones que involucran hablantes del mismo género representa una tarea más compleja que la separación de hablantes de géneros diferentes, ya que las características del tracto vocal de hablantes del mismo género son similares y la tonalidad de las voces o *pitch*, se encuentra en el mismo rango de frecuencias.

6.3.2. Condición cerrada

En esta configuración experimental se realizaron pruebas de separación con mezclas de hablantes que fueron incluidos en el entrenamiento de la red neuronal. Esta información fue empleada únicamente para validar el entrenamiento del modelo y optimizar sus parámetros. En la tabla **??** se muestran los resultados de las métricas de separación empleando la configuración 300x4 (DNN Bi-LSTM de 4 capas y 300 celdas LSTM por capa) y se comparan con los resultados de la evaluación de la misma configuración (300x4) bajo la condición abierta.

	AVG		
	SDR	SAR	SIR
Configuración IBM	14.1	15.4	20.7
Condición abierta	9.9	13.1	16.8
Condición cerrada	10.1	13.3	16.8

Tabla 6.2: Evaluación del sistema en condición cerrada.

La similitud de los resultados de la figura 6.2 para ambas condiciones demuestra la capacidad de la red neuronal para generalizar la tarea de separación a partir de la información dada en el entrenamiento, lo cual es esperado dado que el modelo de configuración (300x4) fue entrenado óptimamente como se ilustra en las curvas de aprendizaje de la figura 6.2. Esta generalización le otorga al modelo la capacidad de separar hablantes de quienes no se tiene información a priori, resultando en un sistema de separación independiente de los hablantes que aparecen en una mezcla.

6.3.3. Condición abierta experimental

En esta condición la evaluación se realizó con información de prueba fuera del dominio de entrenamiento, de manera que las mezclas utilizadas para las pruebas de separación incluyen hablantes que no fueron incluidos en el entrenamiento de la red neuronal y que tampoco pertenecen al corpus de entrenamiento TED-LIUM 3. Para la evaluación del sistema bajo esta condición se emplearon 100 mezclas de hablantes del corpus de voz en español *Acoustic Interactions for Robot Audition* (AIRA)

Condición	AVG		
	SDR	SAR	SIR
Abierta	10.1	13.3	16.8
Abierta experimental	9.2	10.1	14.2

Tabla 6.3: Evaluación del sistema en condición abierta experimental.

Los resultados de la tabla 6.3 muestran que aunque en la condición abierta experimental se obtuvieron resultados de separación por debajo de la condición abierta con información dentro del dominio de entrenamiento, las mejoras de las relaciones SDR, SAR y SIR son muy cercanas. Esto significa que el modelo de separación de hablantes implementado es capaz de generalizar

la separación de señales de voz simultáneas de información de otros dominios, es decir, la separación de hablantes en mezclas de señales de voz grabadas bajo condiciones diferentes. Además, una observación importante es que en este sentido de generalización, el sistema de separación de hablantes es independiente del idioma hablado, ya que TED-LIUM 3 es un corpus con voces en inglés y el corpus AIRA está compuesto por señales de voz en español.

6.4. Ejemplos de separación de hablantes

A continuación se presentan los resultados de la separación de hablantes realizada por el sistema implementado en cuatro mezclas de audio de tres segundos de duración. En cada una de las mezclas de audio, dos personas se encuentran hablando de manera simultánea. Tres de las mezclas incluyen hablantes que no fueron considerados en el entrenamiento de la red neuronal, pero que si pertenecen al corpus de entrenamiento, lo cual corresponde a una evaluación de *condición abierta*, mientras que la mezcla restante incluye hablantes del corpus AIRA [58], lo cual corresponde a una evaluación de condición *condición abierta experimental*.

6.4.1. Separación en condición abierta

En la generación de las mezclas para evaluación en condición abierta se consideraron a los siguientes hablantes: Brian Cox, Isabel Allende, Bill Gates, Michelle Obama, Elon Musk y Nadia López. Cada mezcla generada toma en cuenta una combinación de genero diferente como se indica a continuación:

- **MM**: Bill Gates + Elon Musk
- **FF**: Isabel Allende + Michelle Obama
- **MF**: Brian Cox + Nadia López

Los resultados presentados gráficamente a continuación incluyen la comparación de las señales estimadas con las señales de referencia en el dominio del tiempo, así como la comparación en el dominio tiempo-frecuencia. También se presentan las máscaras binarias estimadas por el método de *deep clustering* (DPCL) y éstas se comparan con las máscaras binarias ideales (IBM).

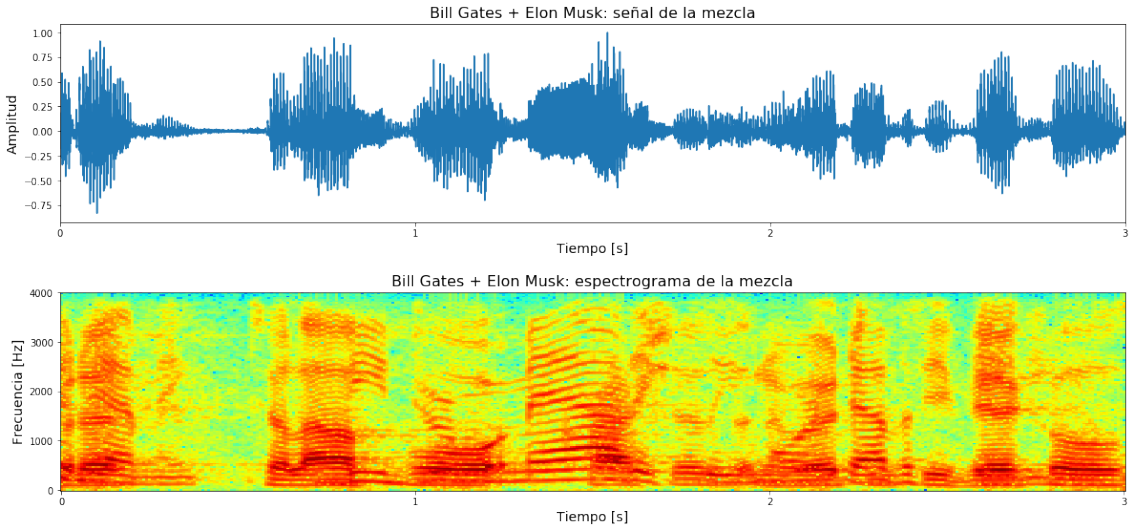


Figura 6.4: Mezcla de dos hablantes hombres: Bill Gates + Elon Musk. (Arriba) Señal de la mezcla en el dominio del tiempo. (Abajo) Espectrograma de la mezcla.

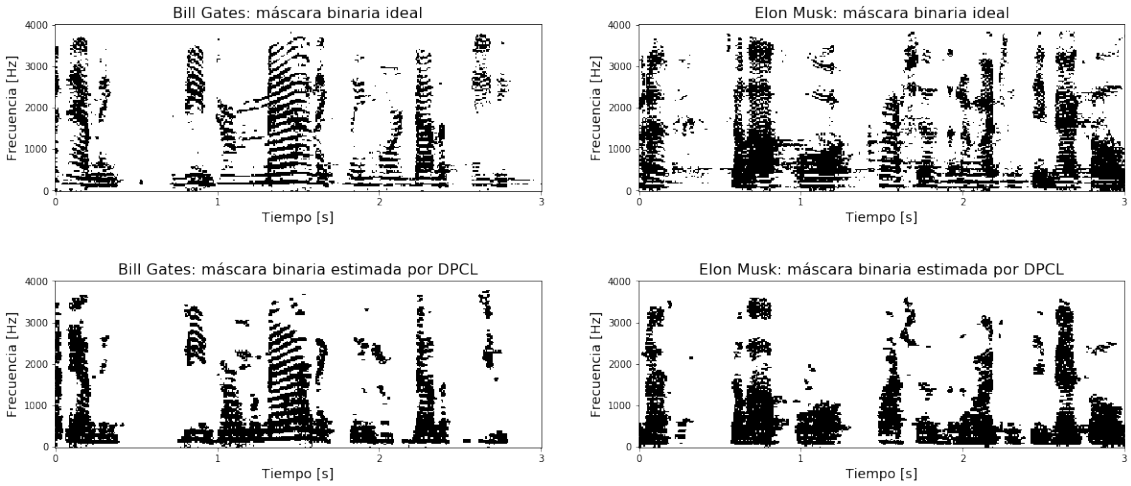


Figura 6.5: Mezcla de dos hablantes hombres: máscaras binarias estimadas. (Arriba) Máscaras binarias ideales. (Abajo) Máscaras binarias estimadas por el sistema (método DPCL).

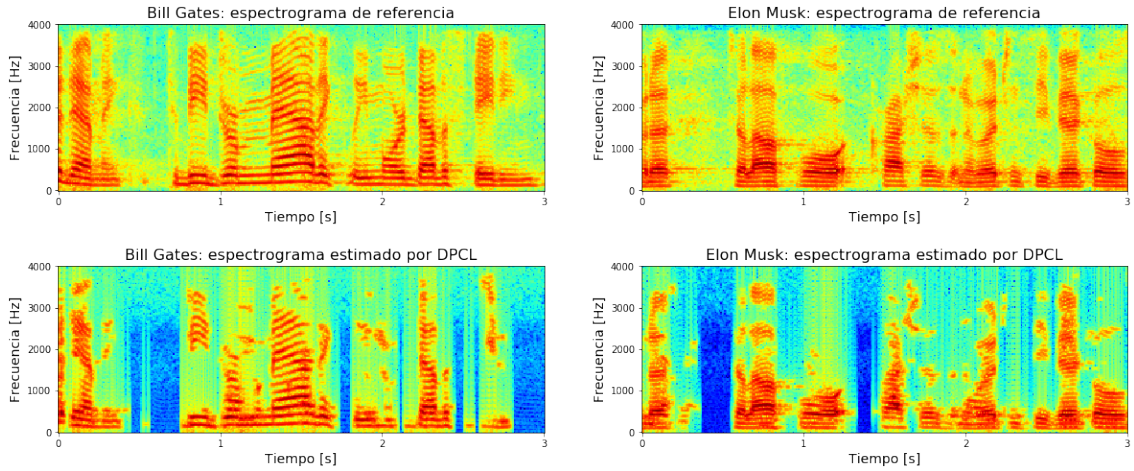


Figura 6.6: Mezcla de dos hablantes hombres: espectrogramas estimados. (Arriba) Espectrogramas de referencia. (Abajo) Espectrogramas estimados.

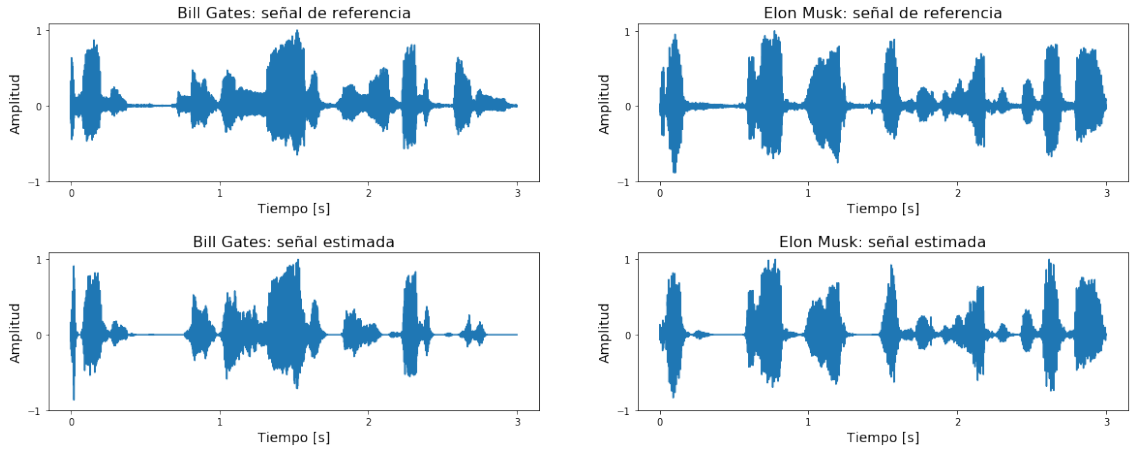


Figura 6.7: Mezcla de dos hablantes hombres: señales estimadas. (Arriba) Señales de referencia en el dominio del tiempo. (Abajo) Señales estimadas en el dominio del tiempo.

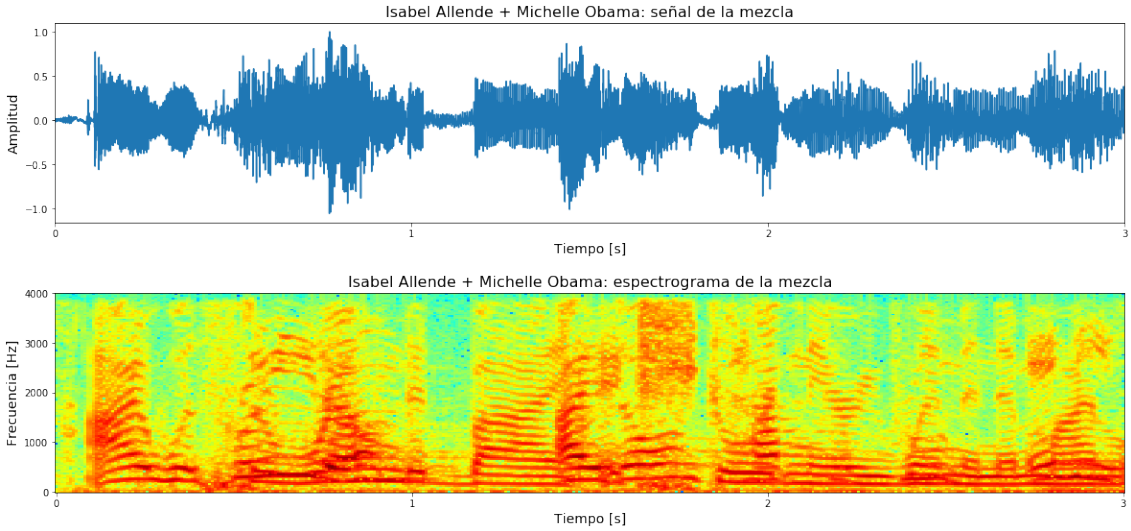


Figura 6.8: Mezcla de dos hablantes mujeres: Isabel Allende + Michelle Obama. (Arriba) Señal de la mezcla en el dominio del tiempo. (Abajo) Espectrograma de la mezcla.

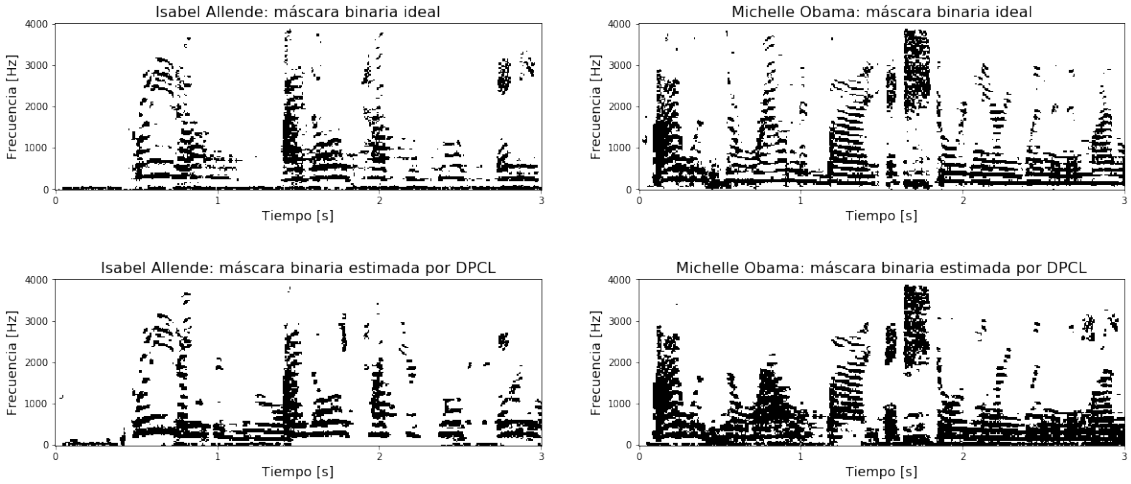


Figura 6.9: Mezcla de dos hablantes mujeres: máscaras binarias estimadas. (Arriba) Máscaras binarias ideales. (Abajo) Máscaras binarias estimadas por el sistema (método DPCL).

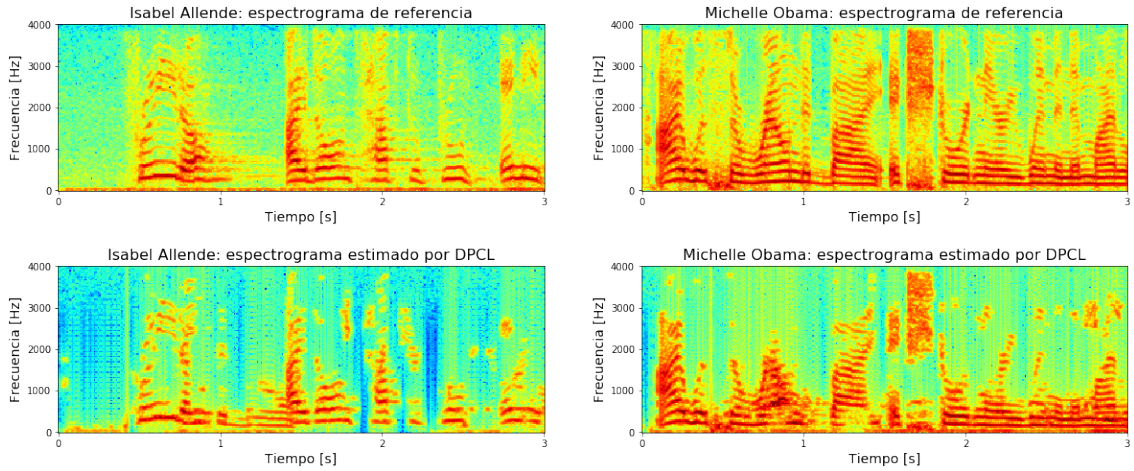


Figura 6.10: Mezcla de dos hablantes mujeres: espectrogramas estimados. (Arriba) Espectrogramas de referencia. (Abajo) Espectrogramas estimados.

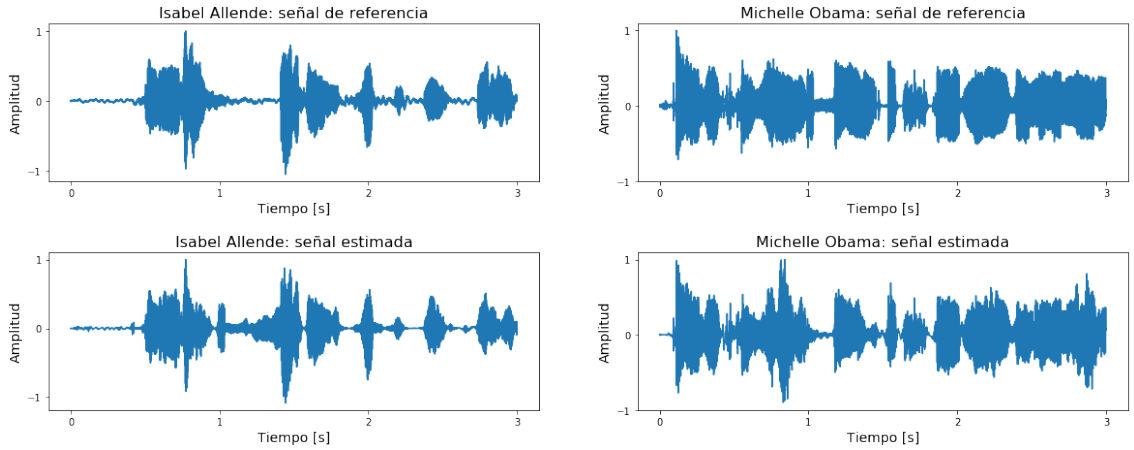


Figura 6.11: Mezcla de dos hablantes mujeres: señales estimadas. (Arriba) Señales de referencia en el dominio del tiempo. (Abajo) Señales estimadas en el dominio del tiempo.

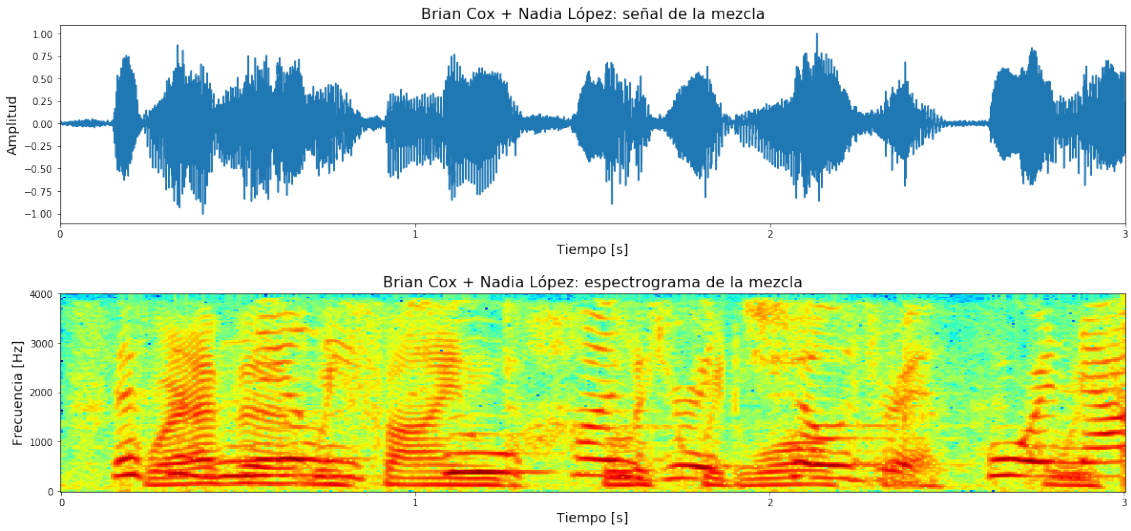


Figura 6.12: Mezcla de dos hablantes hombre-mujer:Brian Cox + Nadia López. (Arriba) Señal de la mezcla en el dominio del tiempo. (Abajo) Espectrograma de la mezcla.

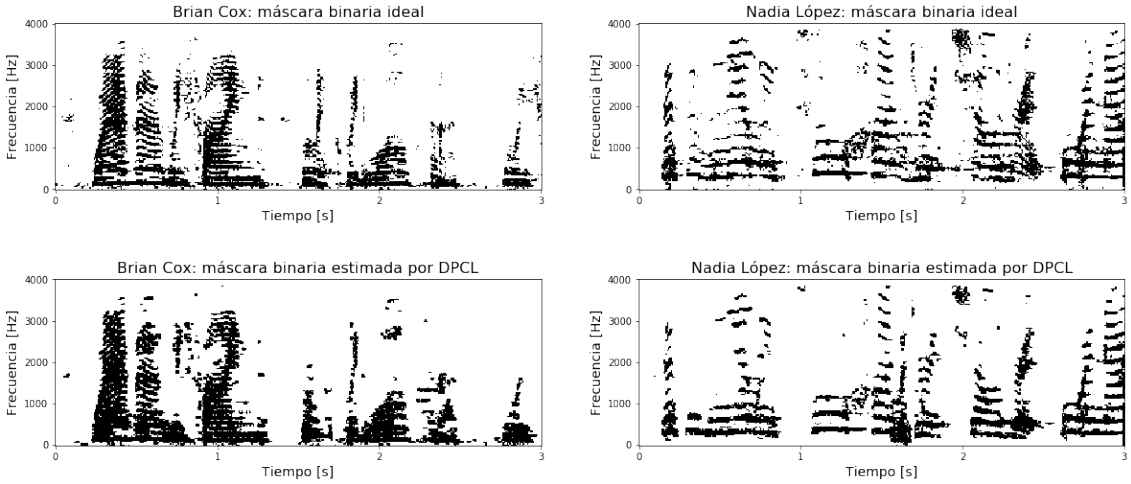


Figura 6.13: Mezcla de dos hablantes: hombre-mujer: máscaras binarias estimadas. (Arriba) Máscaras binarias ideales. (Abajo) Máscaras binarias estimadas por el sistema (método DPCL).

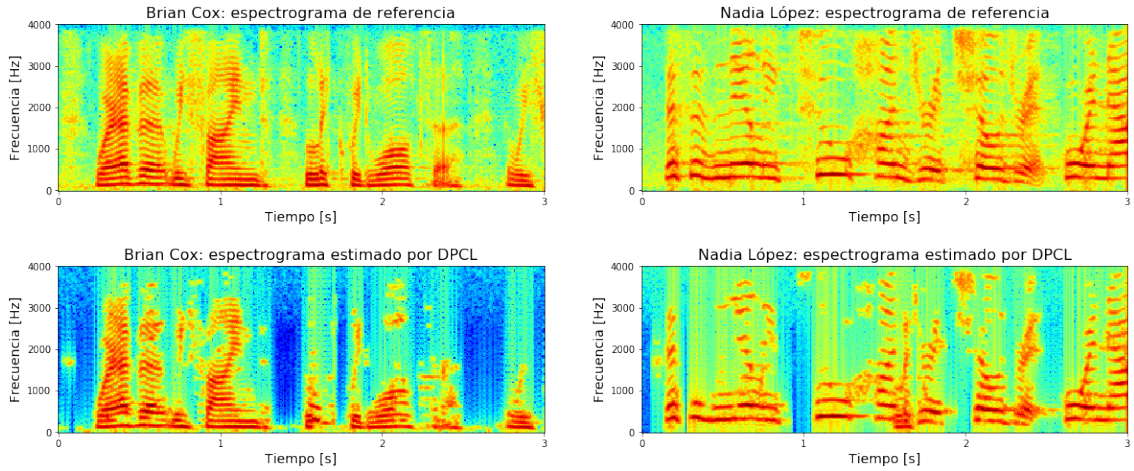


Figura 6.14: Mezcla de dos hablantes: hombre-mujer: espectrogramas estimados. (Arriba) Espectrogramas de referencia. (Abajo) Espectrogramas estimados.

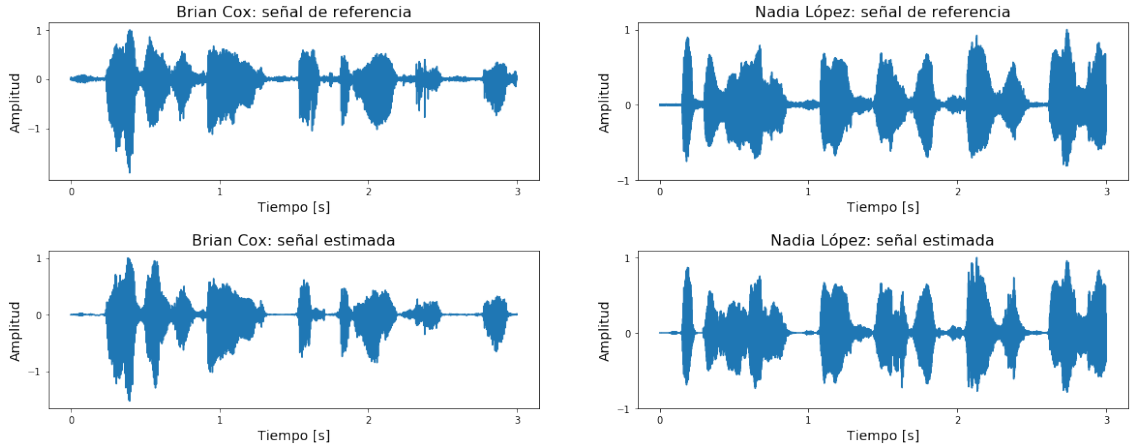


Figura 6.15: Mezcla de dos hablantes: hombre-mujer: señales estimadas. (Arriba) Señales de referencia en el dominio del tiempo. (Abajo) Señales estimadas en el dominio del tiempo.

6.4.2. Separación en condición abierta experimental

Bajo la condición abierta, las mezclas fueron generadas con información de hablantes que no fueron incluidos en el entrenamiento de la red neuronal, pero que pertenecen al corpus TED-LIUM 3. Los resultados de la separación bajo esta condición se muestran en la tabla 6.1. No obstante, también se evaluó el desempeño del sistema en una condición abierta experimental empleando señales de voz del corpus AIRA, con el objetivo de evaluar el desempeño del sistema de separación con información fuera del dominio de la información de entrenamiento. A continuación se muestra el resultado de la separación de una mezcla de hablantes: un hombre y una mujer, pertenecientes a dicho corpus.

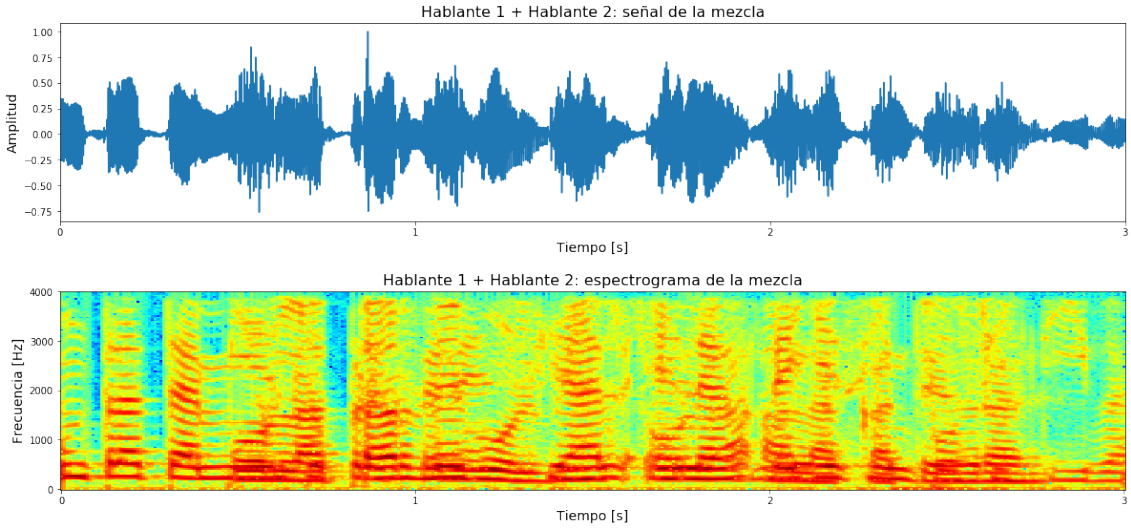


Figura 6.16: Mezcla de dos hablantes: hombre-mujer. (Arriba) Señal de la mezcla en el dominio del tiempo. (Abajo) Espectrograma de la mezcla.

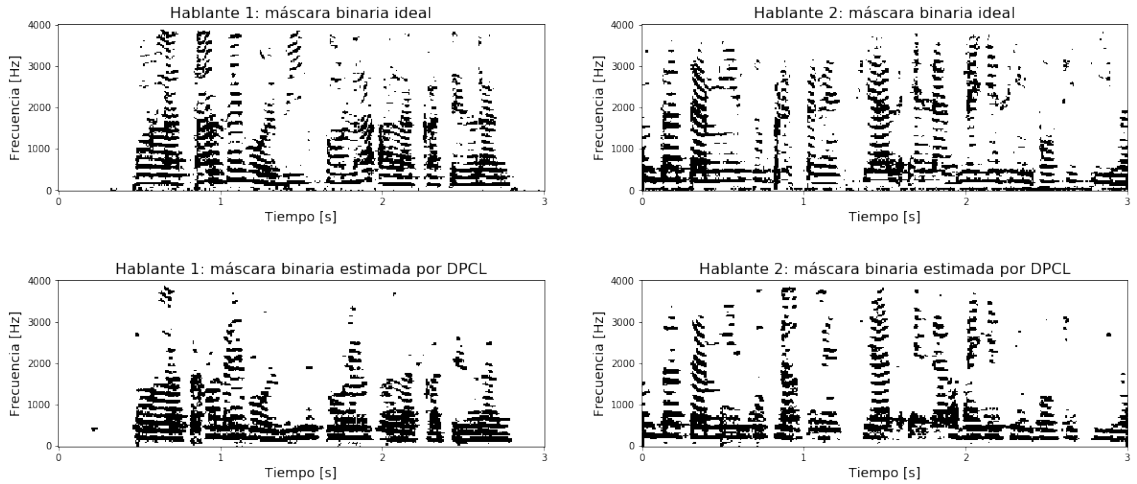


Figura 6.17: Mezcla de dos hablantes: hombre-mujer: máscaras binarias estimadas. (Arriba) Máscaras binarias ideales. (Abajo) Máscaras binarias estimadas por el sistema (método DPCL).

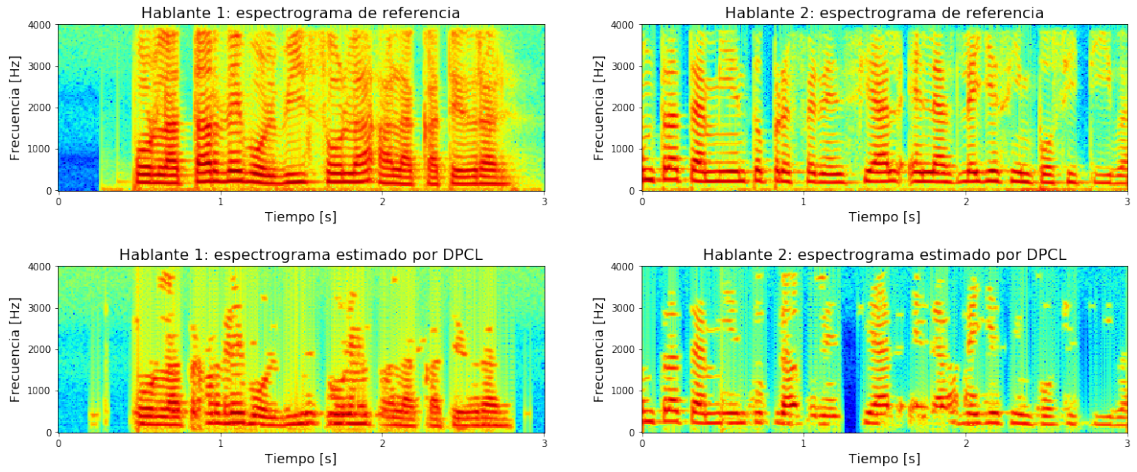


Figura 6.18: Mezcla de dos hablantes: hombre-mujer: espectrogramas estimados. (Arriba) Espectrogramas de referencia. (Abajo) Espectrogramas estimados.

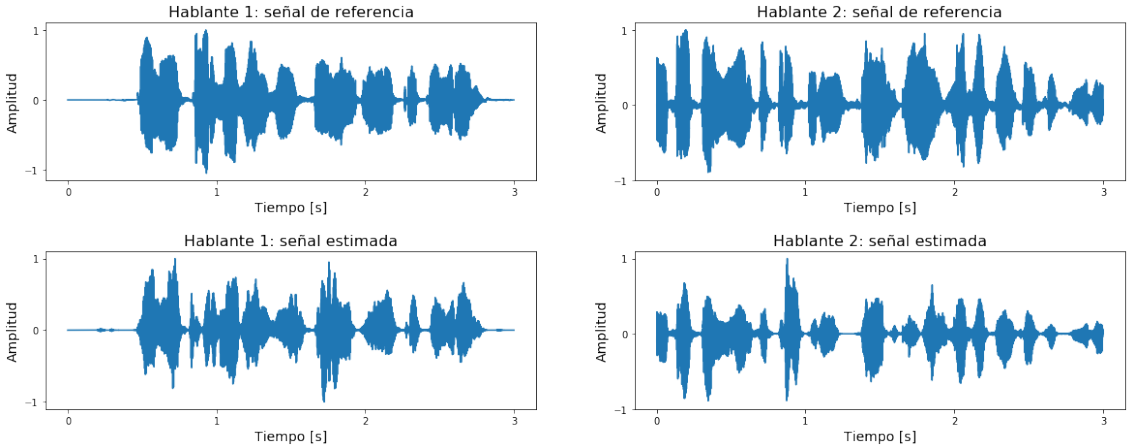


Figura 6.19: Mezcla de dos hablantes: hombre-mujer: señales estimadas. (Arriba) Señales de referencia en el dominio del tiempo. (Abajo) Señales estimadas en el dominio del tiempo.

6.5. Resumen

Se llevó a cabo la evaluación del sistema de separación de hablantes implementado en este trabajo de tesis bajo dos condiciones experimentales que dependen de la información empleada en la etapa de prueba del modelo. La calidad de la separación de cada una las configuraciones de redes neuronales profunda implementadas fue medida mediante las métricas de la librería `BSS_eval`, en términos del promedio de la relación señal a distorsión (SDR), señal a interferencia (SIR) y relación señal a artefactos (SAR).

La evaluación bajo la *condición cerrada*, fue empleada para validar la generalización del modelo para separar mezclas que incluyen hablantes cuya información no es incluida en el modelo. Por otro lado, En la *condición abierta*, la cual asemeja condiciones reales típicas de una fiesta de cóctel donde existe la presencia de hablantes desconocidos de distinto género, la evaluación concluyó que no es necesaria información a priori de los hablantes presentes en una mezcla de audio. No obstante, a pesar de la pérdida natural de información espectral en la estimación de los espectrogramas de los hablantes, la propiedad de escasez de la señal de voz y la redundancia de información en el dominio tiempo frecuencia, la reconstrucción de las señal de voz son en gran medida inteligibles.

La configuración de red neuronal 300x4 conformada por una red neuronal profunda Bi-LSTM de cuatro capas y 600 neuronas por capa, resultó ser la arquitectura óptima para la tarea de separación, obteniendo una mejora general en las relaciones SDR, SAR y SIR de 9.9, 13.1 y 16.8 dB, respectivamente. El sistema entrenado es capaz de separar mezclas de hablantes del corpus TED-LIUM 3, pero también es capaz de separar hablantes de información de otros dominios, como se demostró con la evaluación del sistema en la *condición abierta experimental* empleando mezclas de hablantes del corpus AIRA.

7

Conclusiones y Trabajo Futuro

7.1. Conclusiones

El gran auge de los sistemas de reconocimiento automático de voz en aplicaciones comerciales ha aumentado la necesidad de robustez de dichos sistemas bajo condiciones adversas. Un problema particularmente desafiante es el que ocurre comúnmente en conversaciones espontáneas, donde la señal de voz está corrompida no solamente por ruido, sino también por la interferencia de múltiples hablantes. Debido a que este tipo de interferencia se trata también de voz, se ha demostrado que es muy difícil de reducir, lo que inevitablemente reduce el rendimiento de los sistemas considerablemente. La dificultad de separar señales de la misma clase, es decir, señales de voz de señales de voz, fue la principal motivación de la elaboración de este trabajo de tesis, por lo que en torno a la resolución del problema descrito con anterioridad, se presentó la teoría de procesamiento digital de señales y de aprendizaje profundo fundamental para diseñar e implementar un sistema monoaural de señales de voz basado en aprendizaje profundo. Específicamente se implementó el método novedoso de *Deep Clustering*, el cual representa el estado del arte para la separación de hablantes cuyas voces se encuentran traslapadas en una mezcla de audio de un solo canal. Además se emplearon ilustraciones de cada una de las etapas del sistema de separación para su mejor entendimiento y posible réplica.

Se exploraron diversas configuraciones de redes neuronales para la generación óptima de *embeddings*, siendo una red neuronal recurrente bidireccional de cuatro capas y 300 neuronas por cada dirección, la mejor configuración para la transformación multidimensional no lineal de las unidades tiempo-frecuencia del espectrograma. De acuerdo al teorema de Cover respecto a la separabilidad de patrones, en este espacio multidimensional los *embeddings* que representan las unidades TF pertenecientes a un mismo hablante se localizan juntas, lo cual permitió aplicar un algoritmo simple de agrupamiento como K-medias para separar la información espectral de los hablantes en una mezcla de audio y crear máscaras binarias que aplicadas al espectrograma de la mezcla de audio, permitieron estimar los espectrogramas de cada uno de los hablantes presentes en la mezcla.

La calidad de la separación del sistema implementado fue evaluada en términos de las métricas de separación ciega de fuentes de la librería `BSS_eval`, cuyos resultados demostraron mejoras en las relaciones SDR, SAR y SIR de 9.9, 13.1 y 16.8 dB, respectivamente. Perceptualmente, estos resultados se traducen en una calidad de separación, donde las voces estimadas se encuentran en su mayoría libres de interferencias y con poca distorsión. Estos resultados fueron comparados con los de un sistema de separación de hablantes basado en la aplicación de máscaras binarias ideales, en el cual se obtuvo una mejora en las relaciones SDR, SAR y SIR de 14.1, 15.4 y 20.7 dB, respectivamente. Estos valores representan los mejores resultados de separación que podrían ser obtenidos.

Se demostró que el sistema de separación es capaz de trabajar en condiciones abiertas, por lo que es posible separar voces de hablantes que no fueron incluidos en la etapa de entrenamiento del sistema. Además, el sistema puede separar mezclas de hablantes procedentes de otros dominios de información, es decir, es capaz de separar mezclas de audio de distintas características a las empleadas durante el entrenamiento de la red neuronal. Debido a la optimización del modelo para generalizar las similitudes y diferencias entre dos señales de voz, señales de voz simultáneas en distintos idiomas también pueden ser separadas. Finalmente, todas estas capacidades de separación contribuyen a la solución del problema del efecto fiesta de cóctel.

7.2. Trabajo Futuro

A continuación se describen diversas mejoras al sistema de separación de hablantes implementado en este trabajo de tesis. Debido a que el entrenamiento de redes neuronales profundas es una tarea que requiere demasiado tiempo, diversas adaptaciones, pruebas y experimentos se han dejado para el futuro, mientras que otras se proponen para aquellas personas que tengan interés en trabajar con el método de *Deep Clustering*.

El objetivo principal del método de *Deep Clustering* presentado en [19] e implementado en este trabajo, consiste en estimar máscaras binarias para separar las voces simultáneas de dos hablantes en una mezcla de audio monoaural. Inicialmente el sistema podría adaptarse para separar tres o más hablantes. No obstante, la multiplicación punto a punto del espectrograma con las máscaras binarias produce distorsión en el espectrograma estimado, por lo que una extensión al método consiste en incorporar una etapa de *mejoramiento* del espectrograma estimado, con la finalidad de reducir dicha distorsión a la vez que se intentan recuperar las zonas faltantes debido al traslape de la información espectral de los hablantes. Esta modificación se plantea en [57] y

consiste en procesar cada espectrograma estimado con el espectrograma de la mezcla mediante una red neuronal recurrente bidireccional y una red feedforward con función de activación *softmax* para producir una máscara suave de valores entre 0 y 1.

Una segunda mejora consiste en la aplicación de otros enfoques para construir la máscara de actividad de unidades tiempo-frecuencia. En la implementación del sistema de separación se empleó una umbralización dura (hard thresolding), donde las unidades TF del espectrograma se consideran activas si su valor de potencia es mayor al pico máximo de potencia del espectrograma menos 40 dB. La desventaja de esta umbralización global es que muchas unidades tiempo-frecuencia del espectrograma con bajos niveles de energía, pero con gran cantidad de información útil son descartadas, particularmente las que se encuentran en las frecuencias altas de la señal de voz. Por lo tanto, en lugar de establecer un umbral global, se podría emplear un valor de umbral local definido para cada canal de frecuencia. Bajo este mismo principio también se podría aplicar una umbralización basada en la potencia promedio de cada una de las bandas de frecuencia.

En lo que respecta al aprendizaje de los *embeddings*, el entrenamiento se lleva a cabo a partir del espectrograma de potencia de las mezclas de hablantes, sin embargo podrían emplearse otras formas de representación tiempo-frecuencia para explorar el desempeño del sistema de separación. Por ejemplo, si se toma en cuenta que la distribución de frecuencias del sistema auditivo humano no es lineal, entonces podría emplearse la transformada Q constante [59] en lugar de simplemente la STFT para lograr una mejor simulación del poder de resolución en frecuencia del oído. De manera similar podría emplearse el espectrograma Mel, cuya escala no lineal emula la forma en que el oído humano percibe el sonido, siendo más sensible a frecuencias bajas y menos discriminativo a frecuencias altas [60].

Otra posible mejora consiste en explorar otras arquitecturas de redes neuronales diferentes a la Bi-LSTM para la generación de *embeddings*. Por ejemplo, en lugar de emplear una arquitectura recurrente bidireccional, podría emplearse una red neuronal convolucional (CNN) como en [61], con mecanismos de compuertas similares a las celdas LSTM, denominadas *gated linear units* (GLU) para modelar las dependencias temporales de la representación tiempo-frecuencia de la información de entrenamiento.

Existe la posibilidad de implementar el sistema de separación *en línea* como en [62]. En este sentido, la separación de hablantes debe realizarse con una latencia baja y para lograrlo podría emplearse una ventana de análisis de entre 8 y 25 ms procesada por una red neuronal recurrente de una sola dirección en lugar de una red Bi-LSTM. Bajo este esquema la posición de los centroides de los clústers formados en el espacio de los *embeddings* debe permanecer fija durante todo el proceso de separación, por lo que inicialmente podrían ser calculados mediante un búfer que procese los primeros 300 ms de información.

En el caso en que se cuenta con información de múltiples micrófonos, el método de *Deep Clustering* podría utilizarse de manera auxiliar en sistemas de separación de hablantes multicanal para mejorar su desempeño [63]. La información espacial para localizar las voces de los hablantes podría combinarse con el resultado de la separación monoaural para reducir la distorsión e interferencia de las señales de voces de hablantes provenientes de alguna dirección en particular.

Finalmente, el método de *Deep Clustering* podría adaptarse a la separación de voz de pistas musicales, de instrumentos musicales, sonidos ambientales o sonidos de animales.

Bibliografía

[1] L. A. Álvarez Fernández, *Filtrado direccional de señales de voz en tiempo real.* Universidad Nacional Autónoma de México, 2017.

[2] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[3] A. S. Bregman *et al.*, "Auditory scene analysis," 1990.

[4] Y.-m. Qian, C. Weng, X.-k. Chang, S. Wang, and D. Yu, "Past review, current progress, and challenges ahead on the cocktail party problem," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 40–63, 2018.

[5] J. H. McDermott, "The cocktail party problem," *Current Biology*, vol. 19, no. 22, pp. R1024–R1027, 2009.

[6] G. Hu and D. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation," *IEEE Transactions on neural networks*, vol. 15, no. 5, pp. 1135–1150, 2004.

[7] D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," *Speech separation by humans and machines*, pp. 181–197, 2005.

[8] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications.* Wiley-IEEE press, 2006.

[9] D. Wang and G. J. Brown, "Fundamentals of computational auditory scene analysis," *Computational auditory scene analysis: Principles, Algorithms, and Applications*, pp. 1–44, 2006.

[10] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, pp. 556–562, 2001.

[11] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Ninth International Conference on Spoken Language Processing*, 2006.

[12] S. Behnke, "Discovering hierarchical speech features using convolutional non-negative matrix factorization," in *Proceedings of the international joint conference on neural networks*, vol. 4, pp. 2758–2763, 2003.

[13] L. Zhang, Z. Chen, M. Zheng, and X. He, "Robust non-negative matrix factorization," *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 2, pp. 192–200, 2011.

[14] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Computer Speech & Language*, vol. 24, no. 1, pp. 1–15, 2010.

[15] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer Speech & Language*, vol. 24, no. 1, pp. 45–66, 2010.

[16] T. Virtanen, "Speech recognition using factorial hidden markov models for separation in the feature space.," in *Interspeech*, 2006.

[17] J. Barker, N. Ma, A. Coy, and M. Cooke, "Speech fragment decoding techniques for simultaneous speaker identification and speech recognition," *Computer Speech & Language*, vol. 24, no. 1, pp. 94–111, 2010.

[18] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[19] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 31–35, IEEE, 2016.

[20] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," *arXiv preprint arXiv:1611.08930*, 2016.

[21] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[22] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[23] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4214–4217, IEEE, 2010.

[24] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[25] M. H. Hayes, *Schaum's Outline of Digital Signal Processing*. McGraw-Hill, Inc., 1998.

[26] D. G. Manolakis and V. K. Ingle, *Applied digital signal processing: theory and practice*. Cambridge University Press, 2011.

[27] C. A. Bouman, "Discrete fourier transform." EE438. Class notes on Digital Signal Processing with Applications. Purdue University, October, 2007.

[28] M. Heideman, D. Johnson, and C. Burrus, "Gauss and the history of the fast fourier transform," *IEEE ASSP Magazine*, vol. 1, no. 4, pp. 14–21, 1984.

[29] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, "Numerical recipes in fortran 77: The art of scientific computing, 933 pp," 1992.

[30] R. J. Beerends, *Fourier and Laplace transforms*. Cambridge University Press, 2003.

[31] J. P. Allebach, "Analysis of sampling." EE438. Lecture notes on Digital Signal Processing with Applications. Purdue University, October, 2014.

[32] L. Deng and D. O'Shaughnessy, *Speech processing: a dynamic and optimization-oriented approach*. CRC Press, 2003.

[33] D. O'shaughnessy, *Speech communication: human and machine*. Universities press, 1987.

[34] L. Rabiner and B.-H. Juang, "Fundamentals of speech recognition," 1993.

[35] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*. Prentice Hall, 1978.

[36] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[37] J. Bernal Bermúdez, J. Bobadilla Sancho, and P. Gómez Vilda, "Reconocimiento de voz y fonética acústica," 2000.

[38] D. M. Howard and D. T. Murphy, *Voice science, acoustics, and recording*. Plural Publishing, 2007.

[39] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice*. Pearson Education India, 2006.

[40] E. R. Kandel, J. H. Schwartz, T. M. Jessell, D. of Biochemistry, M. B. T. Jessell, S. Siegelbaum, and A. Hudspeth, *Principles of neural science*, vol. 4. McGraw-hill New York, 2000.

[41] C. S. von Bartheld, J. Bahney, and S. Herculano-Houzel, "The search for true numbers of neurons and glial cells in the human brain: a review of 150 years of cell counting," *Journal of Comparative Neurology*, vol. 524, no. 18, pp. 3865–3895, 2016.

[42] D. Purves, G. Augustine, D. Fitzpatrick, L. Katz, A. LaMantia, J. McNamara, and S. William, "Neuroscience 2nd edition," *Sunderland (MA): Sinauer Associates*, 2001.

[43] S. Bitzer and S. J. Kiebel, "Recognizing recurrent neural networks (rrnn): Bayesian inference for recurrent neural networks," *Biological cybernetics*, vol. 106, no. 4-5, pp. 201–217, 2012.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[46] S. Kostadinov, *Recurrent Neural Networks with Python Quick Start Guide*. Packt Publishing Ltd, 2018.

[47] C. C. Aggarwal, *Neural networks and deep learning*. Springer, 2018.

[48] U. Michelucci, *Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks*. Apress, 2018.

[49] A. Bhardwaj, W. Di, and J. Wei, *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing Ltd, 2018.

[50] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[51] E. Vincent, T. Virtanen, and S. Gannot, *Audio source separation and speech enhancement*. John Wiley & Sons, 2018.

[52] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on signal processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[53] Z. Chen, *Single Channel auditory source separation with neural network*. PhD thesis, Columbia University, 2017.

[54] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE transactions on electronic computers*, no. 3, pp. 326–334, 1965.

[55] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*, pp. 198–208, Springer, 2018.

[56] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[57] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[58] C. Rascon, I. V. Meza, A. Millan-Gonzalez, I. Velez, G. Fuentes, D. Mendoza, and O. Ruiz-Espitia, "Acoustic interactions for robot audition: A corpus of real auditory scenes," *The Journal of the Acoustical Society of America*, vol. 144, no. 5, pp. EL399–EL403, 2018.

[59] J. C. Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[60] M. M. Olvera Zambrano, *Sistema de identificación automática del lenguaje hablado en archivos multimedia de voz*. Universidad Nacional Autónoma de México, 2017.

[61] L. Li and H. Kameoka, "Deep clustering with gated convolutional networks," in *Proc. ICASSP*, pp. 16–20, 2018.

[62] S. Wang, G. Naithani, and T. Virtanen, "Low-latency deep clustering for speech separation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 76–80, IEEE, 2019.

[63] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2018.

Acrónimos

ASA (Auditory Scene Analysis) Análisis de Escenas Auditivas

Bi-LSTM Red Neuronal Recurrente Bidireccional de Celdas LSTM

BPTT (Backpropagation Through Time)

CASA (Computational Auditory Scene Analysis) Análisis Computacional de Escenas Auditivas

DANet Deep Attractor Network

DFT (Discrete Fourier Transform) Transformada Discreta de Fourier

DPCL Deep Clustering

DTFT (Discrete-Time Fourier Transform) Transformada de Fourier en Tiempo Discreto

EER (Equal Error Rate) Tasa de Error Igual

FFT (Fast Fourier Transform) Transformada Rápida de Fourier

FHMM (Factorial Hidden Markov Models) Modelos Ocultos de Markov Factoriales

FIR (Finite Impulse Response) Respuesta Finita al Impulso

GMM (Gaussian Mixture Model) Modelo de Mezclas Gaussianas

GRU (Gated Recurrent Unit) Unidad Recurrente Cerrada

IBM (Ideal Binary Mask) Máscara Binaria Ideal

ICA (Indepent Component Analysis) Análisis de Componentes Independientes

IDFT (Inverse Discrete Fourier Transform) Transformada Discreta de Fourier Inversa

IFFT (Inverse Fast Fourier Transform) Transformada Rápida de Fourier Inversa

ISTFT (Inverse Short-Time Fourier Transform) Transformada de Fourier en Tiempo Corto Inversa

LSTM (Long-Short Term Memory) Memoria de Corto-Largo Plazo

NMF (Non-negative Matrix Factorization) Factorización Matricial No Negativa

OLA (Overlap and Add) Traslape y Suma

PCA (Principal Component Analysis) Análisis de Componentes Principales

PESQ (Perceptual Evaluation of Speech Quality) Evaluación Perceptual de la Calidad de Voz

PIT Permutation Invariant Training

RNN (Recurrent Neural Network) Red Neuronal Recurrente

RPCA (Robust Principal Component Analysis) Análisis de Componentes Principales Robusto

SAR (Signal to Artifacts Ratio) Relación Señal a Artefactos

SDR (Signal to Distortion Ratio) Relación Señal a Distorsión

SIR (Signal to Interference Ratio) Relación Señal a Interferencia

SNR (Signal to Noise Ratio) Relación Señal a Ruido

STFT (Short-Time Fourier Transform) Transformada de Fourier en Tiempo Corto

STOI (Short-Time Objective Intelligibility) Inteligibilidad Objetivo en Tiempo Corto

TF Tiempo-Frecuencia

WER (Word Error Rate) Tasa de Error de Reconocimiento de Palabras