Universidad Zaragoza
1542

Trabajo Fin de Máster

Segmentación automática de audio con modelos basados en redes neuronales para entornos broadcast

Automatic audio segmentation based on neural networks models in broadcast environments

Autor

Pablo Gimeno Jordán

Directores

Alfonso Ortega Giménez
Ignacio Viñals Bailo

Escuela de Ingenieria y Arquitectura
2018

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Pablo Gimeno Jordán,

con nº de DNI 72997678W en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster, (Título del Trabajo) Segmentación automática de audio con modelos basados en redes neuronales para entornos broadcast,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 20 de Junio de 2018

Fdo: Pablo Gimeno Jordán

A mis padres

Agradecimientos

Han sido muchas las personas que a lo largo de la realización de este Trabajo Fin de Máster me han dedicado una parte de su valioso tiempo. De lo contrario, el trabajo habría resultado mucho más complejo. Sirvan estas breves líneas como muestra de mi agradecimiento.

En primer lugar me gustaría dar las gracias a mis directores, Alfonso Ortega e Ignacio Viñals, por sus explicaciones, consejos y correcciones, que han hecho posible la redacción de este Trabajo Fin de Máster. Ha sido un placer trabajar con vosotros y quiero agradeceros la confianza que habéis depositado en mi.

De igual forma, hago extensivo este agradecimiento a todos los miembros de ViVoLab, grupo donde se ha desarrollado este Trabajo Fin de Máster, y a mis compañeros de laboratorio. Su disposición ha sido completa durante los meses en los que se ha desarrollado este trabajo y han sido otro punto de apoyo importante para su realización.

A mi familia, y en especial a mis padres, por su comprensión y apoyo constante durante todos estos años. Sin vosotros no habría podido llegar hasta donde estoy ahora.

Un recordatorio también para todos los compañeros y amigos que he conocido durante estos 6 años de etapa universitaria a los que pone fin este Trabajo Fin de Máster, con los que he compartido grandes momentos y que han hecho este camino mucho más asequible.

A todos, muchas gracias.

Pablo Gimeno Jordán
20 de Junio de 2018

SEGMENTACIÓN AUTOMÁTICA DE AUDIO CON MODELOS BASADOS EN REDES NEURONALES PARA ENTORNOS BROADCAST

RESUMEN

Debido al aumento de generación de contenido multimedia los sistemas que permiten extraer información de forma automática de este tipo de señales se están volviendo cada vez más importantes. Un ejemplo de estos son los sistemas de segmentación automática de audio, sobre los que se centra este Trabajo Fin de Máster. El objetivo de un sistema de segmentación de audio es obtener una indexación a bajo nivel para poder separar entornos acústicos distintos en la señal de audio. En concreto, en este Trabajo Fin de Máster se pretende separar aquellos fragmentos que contengan voz, música, ruido o una combinación de estos.

El acercamiento que se propone a la tarea de segmentación toma como núcleo del sistema el aprendizaje supervisado mediante redes neuronales. De las diferentes arquitecturas neuronales disponibles, el sistema implementado esta basado en Redes Neuronales Recurrentes por su capacidad para el modelado de secuencias temporales. Tras una serie de experimentos iniciales donde se ajustaron los parámetros principales que rigen la arquitectura neuronal, se realizó una exploración de las posibilidades que brindaba el espacio de características de entrada. Por un lado, se aumentó la resolución del análisis frecuencial lo que se tradujo en una mejora relativa del 5,42 % del error obtenido. Por otro lado, con el objetivo de aumentar la discriminación en las clases que contienen música, se introdujeron las características *chroma* obteniendo una mejora relativa del error del 6,04 %. Teniendo en cuenta la alta correlación entre muestras adyacentes en la señal de audio se evaluaron diferentes técnicas de refuerzo del contexto a corto plazo en la clasificación como el apilado temporal o el uso de capas convolucionales, lo que se tradujo en una mejora relativa del 2,63 %.

Finalmente, sobre una de las mejores configuraciones obtenidas, se realizaron una serie de experimentos para caracterizar el sistema de resegmentación propuesto, basado en Modelos Ocultos de Markov y con el objetivo de refinar la salida de la red neuronal. Con este bloque se consiguió reducir considerablemente el error en la segmentación, obteniendo el mejor resultado de este Trabajo Fin de Máster y resultando en una mejora relativa cercana al 12 %.

A la vista de los resultados obtenidos en este Trabajo Fin de Máster, se ha conseguido obtener un sistema de segmentación automático con resultados competitivos, llegando a mejorar ligeramente los mejores resultados de la literatura hasta la fecha.

Índice general

Índice de figuras

Índice de tablas

Listado de acrónimos

Adam	*Adaptative Moment Estimation*
BIC	*Bayesian Information Criterion* (Criterio de Información Bayesiana)
BLSTM	*Bidirectional Long Short Term Memory*
CMVN	*Cepstral Mean & Variance Normalization*
CPU	*Central Processing Unit* (Unidad de Procesado Central)
DCT	*Discrete Cosine Transform* (Transformada de Coseno Discreta)
EM	*Expectation-Maximization*
GLR	*Generalized Likelihood Ratio*
GMM	*Gaussian Mixture Model* (Modelo de Mezcla de Gaussianas)
GPU	*Graphics Processing Unit* (Unidad de Procesado Gráfico)
HMM	*Hidden Markov Model* (Modelo Oculto de Markov)
I3A	Instituto de Investigación en Ingeniería de Aragón
JFA	*Joint Factor Analysis*
LSTM	*Long Short Term Memory*
MFCC	*Mel Frequency Cepstrum Coefficients* (Coeficientes Mel-Cepstrum)
ML	*Maximum Likelihood* (Máxima Verosimilitud)
MLP	*Multilayer Perceptron* (Perceptrón Multicapa)
NIST	*National Institute of Standards and Technology*
PLP	*Perceptual Linear Prediction*
RNN	*Recurrent Neural Network* (Red Neuronal Recurrente)
RTTH	Red Temática de Tecnologías del Habla
SVM	*Support Vector Machines*
openSMILE	*Open Speech & Music Interpretation by Large Space Extraction*
TFM	Trabajo Fin de Máster
VAD	*Voice Activity Detector* (Detector de Actividad Vocal)
ViVoLab	*Voice Input Voice Output Laboratory*

Capítulo 1

Introducción

En este primer capítulo se ofrece una visión general del trabajo desarrollado, introduciendo sus principales conceptos y presentando sus objetivos. Así mismo se contextualizará dentro del entorno en el que se ha realizado, y se definirá la metodología de trabajo seguida. Finalmente, se detallará la estructura de este documento en que se plasma el trabajo llevado a cabo.

1.1. Contexto y trabajo previo

La evolución tecnológica acontecida en los últimos años, así como todos los avances en la sociedad de la información, han contribuido a la generación masiva de recursos multimedia, y a su mayor disponibilidad para el público general. Estos repositorios multimedia han sido bautizados como "*the biggest big data*" y representan una gran mayoría del volumen de tráfico en Internet. Por ejemplo, la plataforma de video Youtube, registra 300 horas de video subidas por minuto y cuenta con más de mil millones de usuarios que cada día visualizan unas mil millones de horas de vídeo [1]. Este incremento de volumen acaba resultando un problema, ya que los sistemas actuales no están preparados para la indexación eficiente de una cantidad de datos tan elevada. Sumado a la problemática del aumento del volumen de datos multimedia debemos añadir el hecho de que cada vez es mayor la cantidad de información diferente que se desea extraer de una señal multimedia. Intrínsecamente, la señal multimedia es muy rica en información y en los últimos años se han desarrollado nuevas técnicas que permiten extraer más información de interés en este tipo de señales. Si nos centramos en los dos flujos principales de la señal multimedia, audio y vídeo, estos son algunos ejemplos de extracción de información que podemos citar:

- **Vídeo**: reconocimiento facial [2], detección de objetos en escenas [3], extracción de escenas principales [4], detección de movimiento, análisis de sentimientos, etc, ...
- **Audio**: transcripción, reconocimiento del locutor [5], detección actividad vocal, detección de fragmentos musicales, detección de patologías [6], etc, ...

Las tareas de transcripción e indexación manual de este tipo de datos resultan especialmente arduas, pudiendo llegar a costar varias veces la duración del documento original. Es por esto que podemos entender la importancia del desarrollo de sistemas automáticos que faciliten esta tarea.

El punto de partida de este Trabajo Fin de Máster (TFM) será la segmentación automática de audio, que pretende obtener una indexación a bajo nivel para poder separar entornos acústicos distintos en la señal de audio. En concreto, se pretende clasificar o agrupar aquellos

segmentos que contengan voz, música y ruido, o combinaciones de los anteriores. El acercamiento a la tarea de segmentación propuesto en este TFM toma como núcleo principal el aprendizaje supervisado mediante redes neuronales, permitiendo automatizar la obtención de esta información en la mayor medida posible.

El trabajo realizado en esta memoria se enmarca dentro del grupo de investigación de tecnologías del habla Voice Input Voice Output Laboratory (ViVoLab) [7], perteneciente al Instituto de Investigación en Ingeniería de Aragón (I3A) de la Universidad de Zaragoza. Una de sus lineas de investigación actuales se centra en la segmentación y clasificación de documentos audiovisuales, tema sobre el que se vertebra este TFM.

1.2. Objetivos y alcance

El principal objetivo de este TFM es el desarrollo y evaluación de diferentes sistemas de clasificación basados en redes neuronales que permitan obtener una segmentación precisa de una señal de audio. Se trata de generar una serie de etiquetas que agrupen el audio en entornos acústicos homogéneos (determinar en qué fragmentos del audio hay voz, en cuales música, o en cuales ruido). El interés de esta tarea es elevado ya que, por un lado, suele utilizarse como primera fase de preprocesado en sistemas más complejos como pueden ser sistemas de identificación del locutor o sistemas de reconocimiento automático del habla. Por otro lado, este tipo de sistemas pueden facilitar la búsqueda e indexación de contenido en grandes repositorios multimedia.

Existen diferentes tipos de arquitecturas neuronales, desde las tradicionales redes *feed-forward* como el perceptrón multicapa, pasando por arquitecturas basadas en redes convolucionales, hasta redes recurrentes para procesar secuencias temporales. El problema común a todas estas arquitecturas es que, a menudo, resulta complicado predecir su comportamiento a priori. Por esto se hace necesario una evaluación de sus prestaciones para comprobar cuál de todas ellas funciona mejor en la tarea de segmentación, y bajo qué condiciones lo hace.

Además, dada la naturaleza de la salida de las redes neuronales, se pretende comprobar la capacidad de un sistema de postprocesado basado en modelos estadísticos generativos para refinar de alguna manera las fronteras de decisión generadas por la arquitectura neuronal.

1.3. Metodología de trabajo

La gran mayoría de herramientas utilizadas para el desarrollo de este TFM se encuentran disponibles a modo de libreria de libre distribución. El auge de disciplinas como el aprendizaje automático y el reconocimiento de patrones ha venido en parte motivado por la nueva filosofia de *software* libre y recursos compartidos. Muchos investigadores en este ambito comparten sus conocimientos, e incluso sus herramientas, de forma pública con la comunidad científica, lo que hace que la cantidad de opciones disponibles sea muy grande. Por tanto, como elemento principal del sistema se encontrará una de estas librerías que permitirá la evaluación de las distintas arquitecturas neuronales (*TensorFlow*[8], *PyTorch*[9], *Theano*[10], *Caffe*[11]). Se utilizarán también algunas herramientas previamente desarrolladas por el grupo de investigación para la producción de modelos estadísticos generativos. En siguientes secciones se llevará a cabo una explicación detallada de todas las herramientas que intervienen en el sistema de segmentación.

Se partirá de una serie de sistemas y modelos ya desarrollados previamente por el grupo de investigación en el ámbito de la detección de actividad vocal (VAD por sus siglas en inglés, *Voice Activity Detector*) con redes neuronales [12]. En esta serie de experimentos se trabajó con una variante de segmentación de audio que buscaba obtener una clasificación binaria de la señal de audio en fragmentos de voz y fragmentos que no sean voz. En este TFM se pretende mejorar y extender este sistema a una segmentación que sea capaz de distinguir varias clases: voz, música, ruido o una combinación de estas.

El grupo de investigación dispone de un *cluster* de computación para el desarrollo de sus tareas de investigación. Este *cluster* cuenta con recursos de CPU (*Central Processing Unit*) de altas prestaciones y GPU (*Graphics Processing Unit*) para cálculo intensivo que permiten acelerar notablemente la ejecución de algoritmos de *Machine Learning* y resulta muy útil a la hora de realizar tareas complejas y paralelizar trabajos. Todas las herramientas citadas anteriormente se utilizaron con este *cluster* como soporte para su ejecución.

Teniendo en cuenta todo lo expuesto, se desarrolló un plan de trabajo donde se explican las diferentes fases necesarias para la consecución de los objetivos expuestos anteriormente:

1. Familiarización con el entorno de trabajo y las herramientas a utilizar
2. Estudio del estado del arte
3. Desarrollo de alternativas y realización de experimentos
4. Interpretación de los resultados experimentales y evaluación crítica de las alternativas: a la vista de los resultados obtenidos será necesario realizar un análisis detallado de las causas y consecuencias de los mismos.
5. Elaboración de la memoria del trabajo: esta fase y la anterior se solapan en el tiempo. Una vez finalizada la parte experimental, es necesario plasmar por escrito los resultados obtenidos y las conclusiones extraídas al analizarlos.

1.4. Organización de la memoria

Además de este primer capítulo donde se introduce el trabajo realizado, este documento consta de 5 capítulos más, que son descritos a continuación:

- **Capítulo 2 - Estudio del estado del arte**: en este capítulo se presentan y analizan los conceptos teóricos de las tecnologías utilizadas, llevando a cabo un repaso de las soluciones propuestas hasta la fecha.
- **Capítulo 3 - Tarea de segmentación Albayzín 2010**: en este capítulo se introduce la tarea de segmentación propuesta para la evaluación Albayzín-2010 sobre la que se sustenta este trabajo, detallando la base de datos a utilizar y las métricas de evaluación
- **Capítulo 4 - Descripción del sistema de segmentación**: en este capítulo se detalla el sistema de segmentación implementado en este trabajo, presentando una descripción general a nivel de diagrama de bloque para posteriormente describir cada uno de sus componentes.
- **Capítulo 5 - Evaluación experimental**: este capítulo presenta la evaluación experimental del sistema de segmentación. Se describen los experimentos realizados así como las diferentes estrategias abordadas, para finalmente analizar e interpretar los resultados.

- **Capítulo 6 - Conclusiones y lineas futuras**: este capítulo recoge las conclusiones extraídas tras la realización de este trabajo y describe brevemente las posibles líneas futuras a seguir.

Capítulo 2

Estudio del estado del arte

Este capítulo presenta el actual estado del arte en el ámbito de la segmentación de audio. Para esto, es necesario introducir las características de audio que se utilizarán, así como una descripción general de las alternativas existentes para llevar a cabo la segmentación de audio. Dado que la aproximación llevada a cabo en este TFM se basa en redes neuronales, se describirán también los fundamentos teóricos de este tipo de sistemas. Finalmente, se expondrán también las bases teóricas de las Modelos Ocultos de Markov.

2.1. Extracción de características

La extracción de características es el conjunto de transformaciones que se aplican a unos datos con el objetivo de maximizar sus propiedades discriminativas adaptándolos al formato más adecuado dada la tarea a realizar. La señal de audio es una señal generalmente no estacionaria. Es por esto que todas las alternativas para la extracción de características en el ámbito de las señales de audio tienen en común el uso de análisis localizado en ventanas temporales pequeñas (entre 25 y 50 ms), que permitan asumir un comportamiento casi estacionario, tal y como se expresa en la ecuación (2.1)

$$Q_n = \sum_{k=-\infty}^{\infty} T(x[n]w[n-k]) \tag{2.1}$$

donde $x[n]$ es la señal de audio a analizar, $w[n]$ es la ventana de K muestras que se aplica a la señal y $T(*)$ es la transformación realizada. Algunas de las transformaciones más habituales son:

- Transformada de Fourier
- Bancos de filtros Mel
- Coeficientes de Mel-Cepstrum (MFCC o *Mel Frequency Cepstrum Coefficients*)[13]
- Perceptual Lineal Prediction(PLP)[14]

Nos centramos aquí en los MFCC y los bancos de filtros Mel, ya que se han convertido en algunas de las opciones más habituales en los sistemas actuales debido a los buenos resultados que permiten obtener en diferentes tareas.

2.1.1. Coeficientes de Mel-Cepstrum y bancos de filtros Mel

Tanto los coeficientes Mel-Cepstrum como los bancos de filtros Mel vienen derivados del análisis cepstral, por lo tanto introduciremos en primer lugar el concepto de cepstrum y análisis cepstral. Este tipo de procesados se utilizan habitualmente en el análisis de la señal de voz, modelada como una excitación $e[n]$ que se convoluciona con un filtro $h[n]$ (tracto vocal). Una de las propiedades más interesantes del análisis cepstral es el hecho de que transforme las convoluciones en el dominio temporal en sumas en el dominio cepstral, tal y como se muestra en la ecuación (2.2):

$$x_1[n] * x_2[n] \xrightarrow{\text{cepstrum}} \hat{x}_1[n] + \hat{x}_2[n] \tag{2.2}$$

De esta forma, la señal de voz obtenida a través de la expresión $v[n] = e[n] * h[n]$, se transformará en una suma en el dominio cepstral, facilitando así su manipulación. Sea $x[n]$ la señal de audio a analizar, podemos definir el cepstrum complejo de $x[n]$ según la ecuación (2.3)

$$\hat{x}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln(X(e^{j\omega n}))e^{j\omega n} d\omega \tag{2.3}$$

donde $X(e^{j\omega n})$ es la transformada de Fourier de la señal $x[n]$. Por tanto, podemos decir que el cepstrum complejo de la señal $x[n]$ se define como la transformada de Fourier inversa del logaritmo neperiano de la transformada de Fourier de $x[n]$. En las señales de audio, la información relevante se encuentra normalmente en el modulo de su transformada de Fourier, por lo que podemos prescindir de la información de fase. Para esto se define en (2.4) lo que se conoce como el cepstrum real:

$$c[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |X(e^{j\omega n})|e^{j\omega n} d\omega \tag{2.4}$$

Así, decimos que el cepstrum real es la transformada de Fourier inversa del logaritmo neperiano del módulo de la transformada de Fourier de la señal.

Los coeficientes Mel-Cepstrum tienen su base en la escala de frecuencias Mel, cuya representación es lineal hasta los 1000 Hz y logarítmica de ahí en adelante. Esto se sustenta en el modelo de percepción auditiva del ser humano, cuya discriminación en frecuencia es mayor para frecuencias bajas que para frecuencias altas [15]. Fueron planteados en los años 80 [13] y actualmente son la representación más utilizada en sistemas de reconocimiento automático del habla o de diarización del locutor. La conversión de la escala frecuencial lineal que se obtiene a la salida de la transformada de Fourier a la escala Mel se lleva a cabo en base a un banco de filtros. con menor ancho de banda en frecuencias bajas y mayor ancho de banda en frecuencias altas. En la Figura 2.1 se puede observar gráficamente una representación espectral de este banco de filtros. Para obtener la salida del banco de filtros se hace pasar el módulo de la trans-

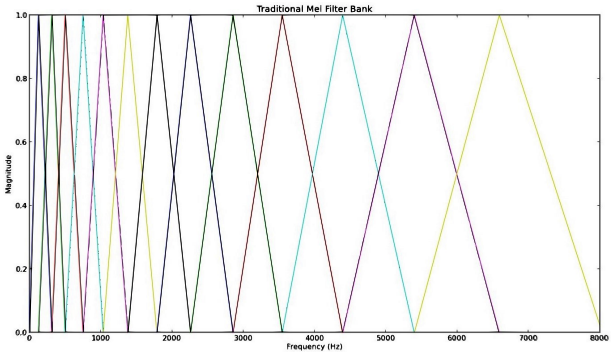


Figura 2.1: Banco de filtros para conversión a escala Mel

formada de Fourier por cada uno de los filtros y se promedia la energia a la salida de cada uno de ellos. Finalmente, como último bloque se aplica la transformada de coseno discreta (DCT o *Discrete Cosine Transform*) en lugar de la transformada inversa de Fourier. Se utiliza la DCT debido a sus propiedades de compresión y, sobre todo, de decorrelación de la información.

Teniendo en cuenta estas explicaciones, se muestra en la Figura 2.2 el diagrama de bloques a implementar para la extracción de las características MFCC.

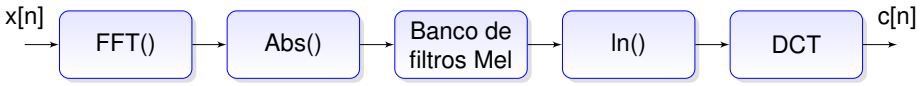


Figura 2.2: Diagrama de bloques para la extracción de características MFCC

Con el objetivo de tener en cuenta las variaciones temporales que se puedan producir en la señal, se suelen tomar también la primera y segunda derivada de los parámetros cepstrales para la representación dinámica, aportando más información a largo plazo. Del mismo modo, también resulta útil la información sobre la energía de la señal.

Siguiendo el mismo esquema de extracción pero tomando la salida un paso antes de aplicar la DCT podemos obtener otro tipo de características basadas también en la escala perceptual Mel, son las que se conocen como banco de filtros Mel. Al no aplicar la DCT, la principal diferencia con los MFCC será que los parámetros obtenidos estarán correlados entre si. El diagrama de bloques asociado a este tipo de características se puede observar en la Figura 2.3.

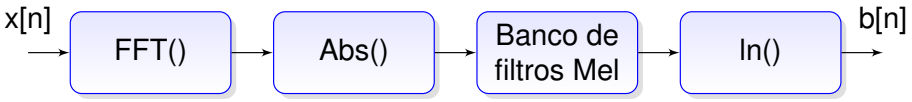


Figura 2.3: Diagrama de bloques para la extracción de características de banco de filtros Mel

2.2. Segmentación de audio automática

El concepto de segmentación de audio es muy amplio y, en general, puede incluir diferentes tipos de aplicaciones. La segmentación automática de audio busca dividir una señal de audio en diferentes clases de tal forma que cada una de ellas contenga información de una tipología acústica específica. Esta idea se puede ver representada de forma esquemática en la Figura 2.4. La principal diferencia entre los distintos tipos de sistemas de segmentación de audio radica en las clases acústicas a distinguir. Algunas de estas tareas han tomado nombre propio dada su relevancia, como por ejemplo:

- Detector de actividad vocal (VAD): clasificación binaria cuyo objetivo es distinguir las fragmentos que contienen voz y los que no la contienen.
- Diarización del locutor: se pretende segmentar la señal acústica de tal forma que se puedan separar los diferentes locutores presentes en el audio.
- Separación de escenarios genéricos: clasificación multiclase donde se busca separar la señal de audio en distintos escenarios genéricos como puede ser voz, ruido de fondo, música, o una combinación de estos.

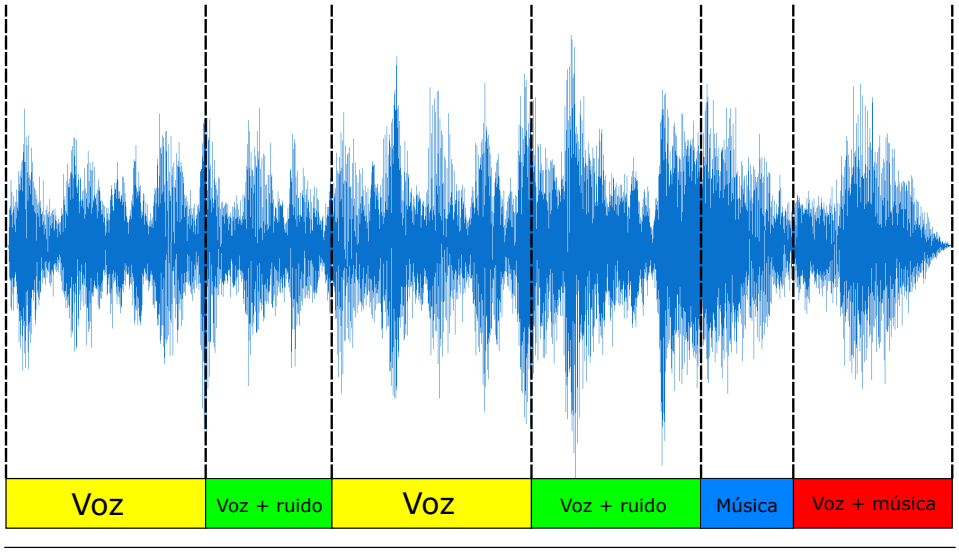


Figura 2.4: Representación esquemática de un sistema de segmentación de audio

Existen diferentes paradigmas para la segmentación automática de audio. Aquí introducimos dos que han sido ampliamente utilizados: la segmentación basada en distancia y la segmentación basada en modelos estadísticos.

2.2.1. Segmentación de audio basada en distancia

Los algoritmos de segmentación de audio basados en distancia calculan una métrica entre dos fragmentos de audio con el objetivo de determinar si existe un cambio de clase acústica entre ellos. Este tipo de técnicas aíslan segmentos con una clase común a lo largo de su duración sin etiquetarlos de forma explícita. Supongamos dos segmentos de audio i y j con sendas secuencias de características X_i, X_j de duración N_i y N_j respectivamente. Supongamos también el segmento que se genera al concatenar X_i y X_j, $X_{ij} = \{X_i \cup X_j\}$. Teniendo en cuenta estas definiciones podemos emitir dos hipótesis diferentes:

- **Hipótesis $\mathbf{H_0}$**: o hipótesis nula, que afirma que ambos segmentos de audio pertenecen a la misma clase acústica.
- **Hipótesis $\mathbf{H_1}$**: o hipótesis alternativa, que afirma que existe un cambio de clase acústica entre los segmentos i y j.

El objetivo de estos métodos es encontrar una distancia, $D(i,j)$, entre los segmentos i y j para determinar qué hipótesis es correcta, H_0 o H_1. Habitualmente la métrica obtenida se compara con un umbral ϵ para seleccionar una de estas dos hipótesis, tal y como se muestra en la ecuación (2.5).

$$D(i,j) \underset{H_0}{\overset{H_1}{\gtrless}} \epsilon \tag{2.5}$$

Aplicar esta estrategia tal cual sobre fragmentos adyacentes de audio podría provocar una estimación ruidosa de los cambios de clase acústica, más sensible a potenciales anomalías locales de la señal de audio. Es por esto que, para mejorar la precisión, la distancia se calcula habitualmente usando una ventana de N tramas, en lugar de únicamente entre dos tramas adyacentes. Cabe destacar que aplicando este tipo de técnicas se obtiene una serie de segmentos de audio que no pertenecen a ninguna clase acústica concreta. A partir de aquí habría que aplicar algún tipo de algoritmo de clasificación para asignar una clase a cada segmento. Es

por esto que este tipo de sistemas se suelen denominar como algoritmos de segmentación y clasificación.

2.2.2. Métricas de distancia

En la literatura sobre segmentación de audio se han propuesto multitud de criterios de distancia. Presentamos aquí dos de los mas habitualmente utilizados:

- ***Bayesian Information Criterion* (BIC)**: El Criterio de Información Bayesiana (BIC por sus siglas en inglés) [16] es una métrica que se utiliza para determinar el grado de ajuste de un modelo dado un conjunto de datos teniendo en cuenta la complejidad de dicho modelo en base a su número de parámetros libres. Dado un conjunto X de N muestras obtenidas de un proceso aleatorio y un modelo θ que describe estos datos, BIC se puede definir matemáticamente según la ecuación (2.6).

$$\mathrm{BIC}(X|\theta) = \log(\mathcal{L}(X|\theta)) - \frac{1}{2}\lambda \#(\theta) \log N \tag{2.6}$$

  El primer término de la ecuación, $\log(\mathcal{L}(X|\theta))$, representa el logaritmo de la verosimilitud de los datos dado el modelo θ, mientras que el segundo término es el encargado de penalizar los posibles modelos candidatos en función de su complejidad. $\#(\theta)$ representa el número de parámetros libres del modelo θ y λ es un peso ajustable para la penalización.

  BIC se puede utilizar para evaluar si existe un cambio en la clase acústica entre dos segmentos i y j. Para esto es necesario calcular dos valores de BIC: un BIC para la hipótesis H_0, asumiendo que los datos de los dos segmentos, $X_{ij} = \{X_i \cup X_j\}$, pueden ser descritos por un único modelo θ_{ij}. Otro BIC se calcula para la hipótesis H_1, asumiendo que los datos de cada segmento son explicados por modelos diferentes, θ_i y θ_j, respectivamente. Para obtener una métrica de distancia a partir de ambos valores, se calcula la diferencia entre ambos, obteniendo lo que se conoce como ΔBIC [17], tal y como se expresa en la ecuación (2.7).

$$\Delta\mathrm{BIC}(X) = \mathrm{BIC}(X|H_1) - \mathrm{BIC}(X|H_0) \tag{2.7}$$

- ***Generalized Likelihood Ratio* (GLR)** [18]: dadas dos secuencias de audio X_i, X_j y la secuencia obtenida al concatenar ambas $X_{ij} = \{X_i \cup X_j\}$, el GLR se puede calcular como el ratio de verosimilitudes entre la hipótesis H_0, que asume que ambos segmentos pertenecen a la misma clase acústica, y la hipótesis H_1, donde se asume que los segmentos pertenecen a clases acústicas distintas. En la ecuación (2.8) se puede observar el cálculo del GLR

$$\mathrm{GLR}(X) = \frac{\mathcal{L}(X_{ij}|H_0)}{\mathcal{L}(X_{ij}|H_1)} = \frac{\mathcal{L}(X_{ij}|\theta_{ij})}{\mathcal{L}(X_i|\theta_i)\mathcal{L}(X_j|\theta_j)} \tag{2.8}$$

  donde $\mathcal{L}(X|\theta)$ representa la verosimilitud de los datos X dado un modelo θ.

2.2.3. Segmentación de audio basada en modelos

A diferencia de los sistemas de segmentación basados en distancia donde solo se detectan las fronteras entre segmentos acústicos, en los sistemas de segmentación basados en modelos cada trama de audio se clasifica como perteneciente a una clase acústica concreta en función de un modelo, por eso se suelen denominar como sistemas de segmentación por clasificación. Dicho modelo hace uso de un conjunto de información a priori para generar una representación de cada clase acústica.

Cabe destacar que los sistemas basados en modelos tienen dos fases de funcionamiento muy bien diferenciadas. Por un lado, en la fase de entrenamiento el sistema ajusta sus parámetros para describir la distribución a priori de los datos de entrenamiento que se le proporcionan. Una vez finalizada la fase de entrenamiento, el sistema puede operar en modo evaluación. En esta fase se presentan nuevos vectores de características al sistema que, con la información obtenida en el entrenamiento, obtiene una estimación de cual es la clase acústica más probable. A continuación citamos algunas de las técnicas de modelado estadístico más habituales para la segmentación de audio:

- ***Gaussian Mixture Model* (GMM)**: Un modelo de mezcla de gaussianas (GMM por sus siglas en inglés) representa una distribución de probabilidad obtenida a través de la suma ponderada de varias distribuciones gaussianas. Se trata de una forma eficiente de modelar distribuciones multimodales, que son el tipo de distribuciones habituales en el ámbito de señales de voz o segmentación de audio. Para un vector de características x, la distribución del modelo GMM se define según la ecuación (2.9)

$$P(x|\lambda) = \sum_{k=1}^{K} \omega_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{2.9}$$

  donde K es número de gaussianas en la mezcla y $\lambda = \{\omega, \mu, \Sigma\}$ el conjunto de parámetros del modelo: ω_k, μ_k y Σ_k son, respectivamente, el peso, la media y la matriz de covarianza de la componente k de la mezcla. Además los pesos deben satisfacer las siguientes condiciones: $0 \leq \omega_k \leq 1$ y $\sum_{k=1}^{K} \omega_k = 1$.

  La técnica más habitual para obtener los parámetros de un GMM es mediante estimación de máxima verosimilitud (ML). Con esta técnica se busca obtener el conjunto de parámetros λ del modelo que maximicen la log-verosimilitud dado un conjunto de datos. La derivación de la solución exacta es compleja, por lo que habitualmente se usa una técnica iterativa conocida como algoritmo EM [19] (*Expectation-Maximization*) para obtener una aproximación.

- **Modelos Ocultos de Markov (HMM)**: una cadena de Markov modela una secuencia de estados cuya probabilidad de encontrarse en un estado depende únicamente del estado inmediatamente anterior. Los Modelos Ocultos de Markov (HMM por sus siglas en inglés, *Hidden Markov Model*) van un paso mas allá para modelar secuencias estocásticas como cadenas de Markov. Los estados del proceso no son directamente observables, se dice entonces que están ocultos. Dicha secuencia es observable solo a través de los procesos estocásticos que se definen en cada estado. Una explicación con mayor detalle sobre los HMM se puede encontrar en la sección 2.4.

- ***Joint Factor Analysis* (JFA)** [20]: JFA es un modelo generativo basado en modelos GMM que tiene en cuenta diferentes fuentes de variabilidad. A diferencia del modelo GMM estándar, donde cada una de las medias de las gaussianas se adapta de forma independiente, en el modelo JFA existen una serie de variables ocultas que atan las medias de las gaussianas a la hora de su adaptación para todas las tramas de un segmento [21]. Las medias de todos los componentes de la GMM se concatenan para construir un supervector de alta dimensionalidad que se puede modelar como la suma de varios factores. Dicha descomposición para el supervector medio de un segmento de audio $\mathbf{M}_{is}$ perteneciente a la clase i y afectado por unas condiciones de canal s se puede ver en la ecuación (2.10)

$$\mathbf{M}_{is} = \mathbf{m} + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_s + \mathbf{D}\mathbf{z} \tag{2.10}$$

donde $\mathbf{m}$ es el supervector de medias del modelo GMM, $\mathbf{V}\mathbf{y}_i$ es el término que modela la variabilidad interclase a través de la matriz de bajo rango $\mathbf{V}$ y el vector $\mathbf{y}_i$, variable oculta asociada a la identidad de la clase i. $\mathbf{U}\mathbf{x}_s$ es el término que modela la variabilidad asociada al canal mediante la matriz de bajo rango $\mathbf{U}$ y el vector $\mathbf{x}_s$, variable oculta asociada a unas condiciones de canal s. Finalmente el término $\mathbf{D}\mathbf{z}$ modela la variabilidad residual que no ha podido ser modelada por el término de clase y el término de canal a través de una matriz diagonal $\mathbf{D}$ y el vector de variabilidad residual $\mathbf{z}$

Existen además variaciones sobre esta idea que modelan unicamente una fuente de variabilidad como puede ser *Eigenchannel-MAP*, que tiene en cuenta unicamente la variabilidad del canal [22]. Este tipo de modelos y otras variantes de JFA se han utilizado con resultados satisfactorios para segmentación de audio en entornos *broadcast* [23].

- **Redes neuronales**: sistemas computacionales compuestos por la interconexión de varias unidades elementales denominadas neuronas que se basan en el paradigma del aprendizaje supervisado. Dado que este TFM se centrará en este tipo de modelos, en la sección 2.3 se lleva a cabo un análisis más detallado sobre los fundamentos teóricos de las redes neuronales.

- **Otros**: otros algoritmos y técnicas de reconocimiento de patrones y *Machine Learning* se han aplicado a la segmentación de audio como por ejemplo *Support Vector Machines*(SVM) [24], arboles de decisión [25], o lógica difusa [26].

2.3. Redes neuronales

Las redes neuronales presentan un paradigma de computación basado en el aprendizaje de fronteras de decisión no lineales adaptando sus parámetros en base a una serie de ejemplos. No se trata de un concepto novedoso ya que los primeros modelos datan de los años 40 [27]. Durante los años 70 y 80 se desarrollaron la mayoria de algoritmos y técnicas relacionadas con este campo pero, sin embargo, no ha sido hasta en torno al año 2010 donde los avances tecnológicos, sobre todo en el campo del *hardware*, y la mayor disponibilidad de datos para su entrenamiento, han permitido el último auge de este tipo de modelos. Las redes neuronales han probado su eficacia en varios campos del reconocimiento de patrones como puede ser el reconocimiento automático del habla [28], o la visión por computador [29].

2.3.1. Redes *feed-forward*

Se conocen como redes neuronales *feed-forward* a aquellas arquitecturas neuronales que no tienen ningún bucle de realimentación entre su salida y su entrada. Al no existir esta realimentación se puede decir que la información fluye en un único sentido. También se suelen conocer como perceptrones o MLP (*Multilayer Perceptron*) en caso de constar de varias capas.

Este tipo de redes neuronales se compone de un número N de unidades básicas de procesado, habitualmente conocidas como neuronas. La neurona consta de n entradas y una salida y tal y como se puede ver en la Figura 2.5. A continuación se describen sus elementos principales:

- **Entradas**: que se representan mediante x_j, y pueden ser tanto entradas de datos externos como las salidas resultantes de otras neuronas.

- **Pesos**: representados por w_j, son los coeficientes por los que se multiplica cada una de las entradas x_j de la neurona.

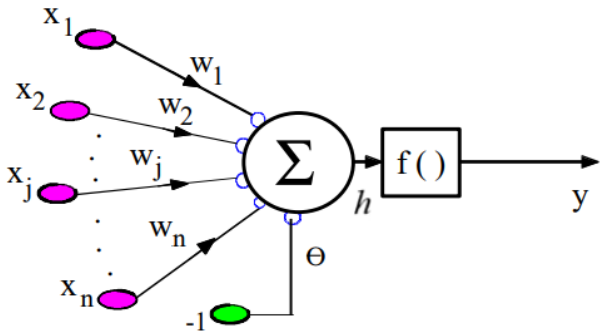


Figura 2.5: Esquema de una neurona básica

- ***Bias***: representado por θ, es el término constante de la combinación lineal que se aplica sobre las entradas.
- **Función de activación**: indicada por $f(*)$, función aplicada sobre la suma ponderada de las entradas de la neurona, que habitualmente tiene un comportamiento no lineal. Existen diferentes tipos de funciones de activación, dos de las más ampliamente utilizadas son la función sigmoide y la tangente hiperbólica, cuyas expresiones quedan definidas en las ecuaciones (2.11) y (2.12) respectivamente.

$$f(h) = \sigma(h) = \frac{1}{1 + e^{-h}} \tag{2.11}$$

$$f(h) = \tanh(h) = \frac{e^{h} - e^{-h}}{e^{h} + e^{-h}} \tag{2.12}$$

- **Salida**: representada por y, es el resultado final de las operaciones realizadas en la neurona. Para el cálculo de esta salida debemos distinguir dos pasos. Inicialmente se realiza una combinación lineal de las entradas x_j ponderada por los pesos w_j para obtener el termino h, tal y como se expresa en la ecuación (2.13).

$$h = \sum_{j=1}^{n} w_j x_j - \theta \tag{2.13}$$

  Finalmente, al termino h se le aplica la función de activación para obtener la salida y, según la ecuación (2.14).

$$y = f(h) = f\left(\sum_{j=1}^{n} w_j x_j - \theta\right) \tag{2.14}$$

La unión de múltiples neuronas forma lo que se conoce como red neuronal. La estructura habitual es formar capas, de tal forma que N neuronas trabajen en paralelo calculando sus salidas. Así, estas salidas servirán de entradas a la capa siguiente, con una interconexión como la que se muestra en la Figura 2.6, donde la salida de una neurona en la capa i sirve de entrada a todas las neuronas de la capa $i + 1$. La Figura 2.6 sirve también para ilustrar los diferentes tipos de capa que forman una red neuronal. En azul vemos las neuronas pertenecientes a la capa de entrada, mientras que en rojo vemos las neuronas de la capa de salida. En verde se representan las neuronas de la capa oculta. Este tipo de capa puede existir o no, y en caso de existir puede haber una o más capas ocultas. El número de este tipo de capas define la complejidad de la red neuronal, a la vez que su capacidad de modelado.

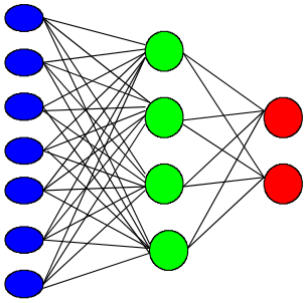


Figura 2.6: Ejemplo de una red neuronal con una única capa oculta

Otro elemento importante a citar en el contexto de las redes neuronales es la función de coste o función de error, $J(w)$, que sirve para medir la diferencia entre la salida de la red y la salida deseada. Con esta referencia de error, y a través del algoritmo *Backpropagation* [30], se puede obtener como influye cada uno de los parámetros de la red en la función de coste mediante sus derivadas parciales respecto de cada uno de los parámetros. Esta información que nos proporciona el algoritmo *Backpropagation* es utilizada para ajustar el modelo actualizando de forma iterativa los parámetros entrenables, pesos y *biases*, en la dirección de máxima pendiente del gradiente de la función de error asociada a la red neuronal. La ecuación que define la actualización de los pesos en un algoritmo de descenso por gradiente viene expresada en la ecuación (2.15).

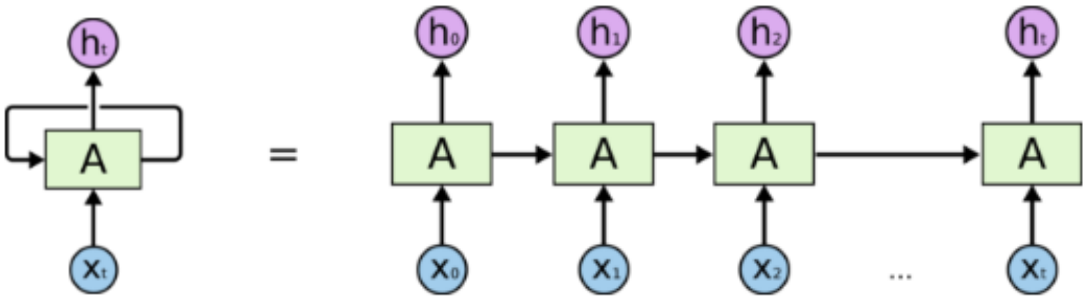
$$w_{n+1} = w_n - f(\alpha, \nabla_w J(w)) \tag{2.15}$$

donde w_n son los parámetros en el instante actual, y el término $f(\alpha, \nabla_w J(w))$ representa el término de actualización de los parámetros. Como se puede observar se obtiene a través de dos valores: α es un hiperparámetro conocido como tasa de aprendizaje o *learning rate* que afecta a la convergencia del algoritmo, y $\nabla_w J(w)$ es el gradiente de la función de coste con respecto de los parámetros w. Según el algoritmo de optimización utilizado se modificará el comportamiento de la función $f(*)$ que se aplica a estos dos valores.

2.3.2. Redes neuronales recurrentes (RNNs)

Uno de los principales inconvenientes que presentan las redes neuronales *feed-forward* a la hora de trabajar con secuencias es que no guardan ningún tipo de información sobre eventos secuencialmente distantes, es decir, cada ejemplo que se les presenta es tratado de forma completamente independiente al resto. Las redes recurrentes, denominadas habitualmente por sus siglas en inglés como RNN (*Recurrent Neural Networks*), son capaces de modelar dependencias temporales en secuencias de datos introduciendo bucles de realimentación entre la entrada y la salida de la red, tal y como se observa en la Figura 2.7a. Otra forma de entender este tipo de redes es pensar en ellas como múltiples copias de la misma red, cada una de ellas pasando un mensaje a la siguiente. Esta idea se ilustra en la Figura 2.7b. Con este esquema, este tipo de redes se pueden ver como cadenas de varias redes *feed forward* unidas. En esta idea se basa uno de los algoritmos a los que se recurre para su entrenamiento, el conocido como *Backpropagation through time* [31].

A pesar de que, teóricamente, las redes recurrentes son capaces de modelar dependencias temporales de larga duración, en la práctica se ha visto que esto no es así ya que presentan una serie de problemas asociados a los cálculos de gradiente [32]. Es por esto que el éxito de las redes recurrentes se debe, en gran parte, a la aparición de las LSTM (*Long Short Term Memory*) [33],



(a) Representación como realimentación entrada-salida

(b) Representación como cadena de redes *feed-forward*

Figura 2.7: Esquema general de una red recurrente

una modificación sobre la versión básica de red recurrente basada en la incorporación de una nueva estructura denominada celda de memoria.

2.3.3. Redes LSTM (Long Short Term Memory)

Las redes LSTM son un tipo de redes recurrentes capaces de asimilar dependencias de larga duración, es decir, aquellas dependencias donde el instante temporal en el que se utiliza la información aprendida dista del momento en el que se aprendió. Este nuevo modelo conlleva un aumento de la complejidad ya que, mientras que en una RNN estándar la salida se calcula aplicando una única capa neuronal a la entrada, las redes LSTM añaden una serie de pasos intermedios agregados en lo que se conoce como celda de memoria. Esta celda se puede observar en la Figura 2.8.

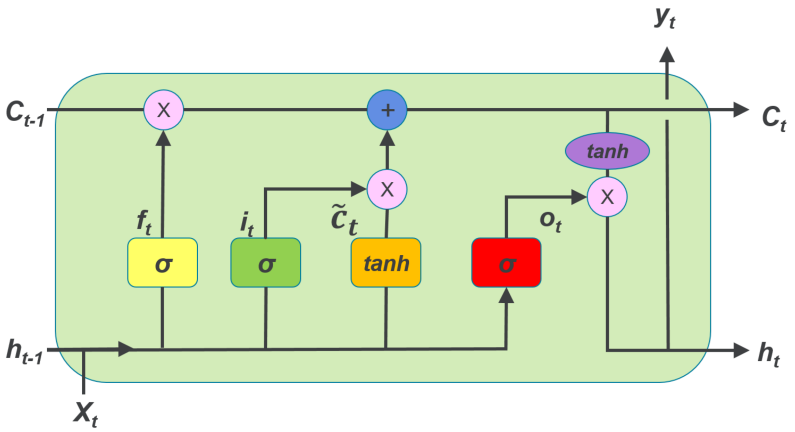


Figura 2.8: Esquema de una celda de memoria LSTM

El elemento clave de las redes LSTM es el uso del estado de la celda (C_t), un elemento de memoria que se propaga a lo largo del tiempo. La LSTM tiene la habilidad de quitar o añadir información a este estado de la celda con un tipo de estructuras que se denominan puertas (*gate* en inglés). Estas estructuras se componen de una capa neuronal con activación sigmoidea, cuya salida esta entre 0 y 1. Esta salida representará la cantidad de información que se permite pasar al estado de la celda. Una celda de memoria LSTM se constituye de tres puertas:

- **puerta de olvido** (f_t): combinando la información de la entrada actual, x_t, y la salida anterior, h_{t-1}, decide que cantidad de la información anterior, C_{t-1}, se ha de olvidar.

- **puerta de entrada** (i_t): se encarga de actualizar el estado de la celda, C_t, decidiendo qué cantidad de la entrada actual, x_t, se incorpora a la memoria de la celda.

- **puerta de salida** (o_t): su función es determinar la contribución a la salida de la celda, que se basará en una versión filtrada del estado de la celda actual, C_t, teniendo en cuenta la información de la entrada.

Cabe destacar también una evolución de las redes LSTM, las redes BLSTM (*Bidirectional Long-Short Term Memory*). Como su propio nombre indica, la principal ventaja que presentan es que son capaces de trabajar en dos direcciones. Se forman al combinar dos redes LSTM recorriendo la secuencia en sentidos opuestos. Esto hace que sean capaces de aprender dependencias tanto causales como anticausales, a costa de elevar la complejidad del modelo y de no poder operar en tiempo real.

2.3.4. Redes neuronales convolucionales

La base teórica de las redes neuronales convolucionales [34] es similar a las de las redes *feed-forward* en lo que respecta a que constan de una serie de parámetros entrenables y realizan un procesado de sus entradas para acabar aplicando una función de activación. Su principal diferencia radica en que cada neurona trabaja únicamente con la información de dependencias de alcance limitado, permitiendo así realizar un análisis localizado de la entrada.

Al trabajar de forma localizada, las redes neuronales convolucionales son capaces de modelar de forma consecutiva pequeñas piezas de información que después se combinan en capas más profundas de la red. Las capas finales llevan a cabo la predicción final como suma ponderada de las entradas. Para construir este tipo de redes habitualmente se intercalan capas convolucionales y capas de reducción o *pooling* tal y como se puede observar en la Figura 2.9 Las capas de convolución constan de una serie de filtros entrenables que operan sobre la en-

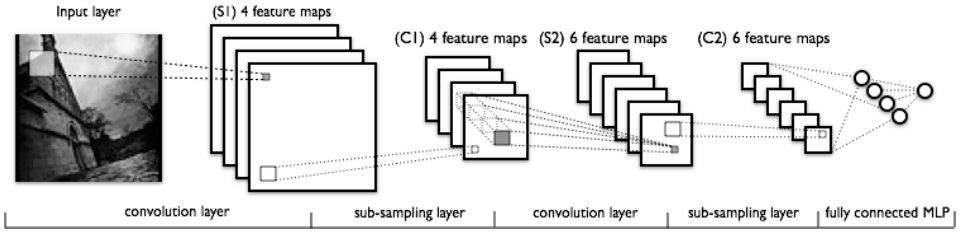


Figura 2.9: Ejemplo de arquitectura de una red neuronal convolucional

trada y extraen un mapa de características de esta. La capa de *pooling* se utiliza para reducir las dimensiones espaciales de la entrada, lo que conlleva una menor carga computacional y reduce el sobreajuste del modelo. En la Figura 2.10 se presenta una posible implementación de la capa de *pooling* habitualmente conocido como *max-pooling* donde se toma el máximo de un bloque de NxN de la imagen.

Las arquitecturas convolucionales fueron propuestas inicialmente a finales de la decada de los 90 y han supuesto una revolución en varios campos de la visión por computador como el reconocimiento de imágenes o la detección de objetos en escenas.

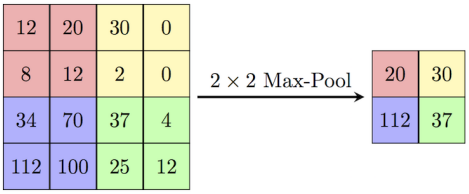


Figura 2.10: Ejemplo de aplicación de una capa de *max-pooling*

2.4. Modelos Ocultos de Markov (HMM)

Los Modelos Ocultos de Markov (HMM por sus siglas en inglés, *Hidden Markov Model*) son una herramienta de modelado estadístico para secuencias temporales de información. Se trata de un modelo probabilístico, es decir, dada una secuencia de entrada calculan un conjunto de etiquetas que mejor se ajustan al modelo.

Sea $\mathbf{Z} = [Z_1, Z_2, ..., Z_n]$ una secuencia de variables aleatorias, a partir del teorema de Bayes podemos obtener la probabilidad conjunta de observar la secuencia $\mathbf{Z}$ según la ecuación (2.16),

$$P(Z_1, Z_2, ...Z_n) = P(Z_1) \prod_{i=2}^{n} P(Z_i | Z_1^{i-1}) \tag{2.16}$$

donde $Z_1^{i-1} = [Z_1, Z_2, ...Z_{i-1}]$. Las variables aleatorias Z formarán una cadena de Markov de orden 1 si satisfacen la condición expresada en la ecuación (2.17):

$$P(Z_i | Z_1^{i-1}) = P(Z_i | Z_{i-1}) \tag{2.17}$$

La ecuación (2.17) es conocida como la Hipótesis de Markov, según la cuál la probabilidad condicional de una variable aleatoria en un instante depende sólo de la variable aleatoria en el instante anterior. Así, podemos rescribir la ecuación (2.16) de la siguiente forma:

$$P(Z_1, Z_2, ...Z_n) = P(Z_1) \prod_{i=2}^{n} P(Z_i | Z_{i-1}) \tag{2.18}$$

Si descartamos el índice de tiempos i podemos modelar eventos estacionarios como:

$$P(Z_i = s | Z_{i-1} = s') = P(s|s') \tag{2.19}$$

Asociando un estado s a Z_i, la cadena de Markov se puede representar mediante una máquina de estados finitos estocástica, con transiciones entre estados correspondientes a la probabilidad $P(s|s')$. Teniendo esto en cuenta, de la ecuación (2.18) podemos inferir que la probabilidad de que la cadena de Markov esté en un estado específico en un instante determinado depende únicamente del estado en el instante anterior.

Dada una cadena de Markov de N estados y sea s_t el estado de la cadena de Markov en el instante t, los parámetros que definen esta cadena vienen expresados en las ecuaciones (2.20) y (2.21):

$$a_{ij} = P(s_t = j | s_{t-1} = i) \quad 1 \leq i, j \leq N \tag{2.20}$$

$$\pi_i = P(s_1 = i) \quad 1 \leq i \leq N \tag{2.21}$$

donde a_{ij} es la probabilidad de transición entre el estado i y el estado j, y π_i la probabilidad de que la cadena de Markov comience en el estado i. Dichas probabilidades de transición y de

inicialización verifican las siguientes restricciones:

$$\sum_{j=1}^{N} a_{ij} = 1 \quad 1 \leq i \leq N \tag{2.22}$$

$$\sum_{j=1}^{N} \pi_j = 1 \tag{2.23}$$

Con esta descripción podemos denominar esta cadena de Markov como Modelo Observable de Markov, ya que la salida del proceso es el conjunto de estados en cada instante de tiempo t, donde cada estado se corresponde con un evento que podemos observar Z_i, por lo que hay una correspondencia uno a uno entre la secuencia de eventos observables y la secuencia de estados de la cadena de Markov.

Ahora supongamos una cadena de Markov donde la observación ya no es un valor determinista sino una variable aleatoria X, generada según una distribución de probabilidad asociada a cada estado. Ya no podemos encontrar una correspondencia unívoca entre la secuencia de salida y la secuencia de estados. Por eso se dice que la secuencia de estados está oculta. Es por esto que se les denomina *Hidden Markov Models* o HMM. Formalmente, un HMM queda definido por los siguientes parámetros:

- $S = [1, 2, ..., N]$: conjunto de estados que caracterizan el modelo.
- $A = \{a_{ij}\}$: probabilidad de transición entre estados, donde a_{ij} representa la probabilidad de transitar del estado i al estado j.
- $B = \{b_i\}$: conjunto de distribuciones de probabilidad, donde b_i es la distribución de probabilidad de la observación en el estado i.
- $\pi = \{\pi_i\}$: probabilidad de estado inicial, donde π_i es la probabilidad de estar en el estado i en el instante $t = 1$.

Por tanto, un HMM queda caracterizado por dos matrices de probabilidad, A y π, y un conjunto de distribuciones de probabilidad B. Es necesaria también una constante N que caracteriza el número de estados. La notación que se utilizará para referirnos a un HMM es la especificada en (2.24)

$$\Phi = (A, B, \pi) \tag{2.24}$$

Para la obtención de este modelo se han tenido en cuenta dos asunciones:

1. La hipótesis de Markov para las cadenas de Markov:

   $$P(s_t|s_1^{t-1}) = P(s_t|s_{t-1}) \tag{2.25}$$

   donde s_1^{t-1} representa la secuencia de estados $[s_1, s_2, ...s_{t-1}]$. Esta asunción implica que la probabilidad de encontrarse en el estado s en un instante de tiempo t sólo depende del estado en el que nos encontrábamos en el instante $t - 1$.

2. Independencia de las observaciones de salida anteriores:

   $$P(X_t|X_1^{t-1}, s_1^t) = P(X_t|s_t) \tag{2.26}$$

   donde X_1^{t-1} es la secuencia de salida $[X_1, X_2, ..., X_{t-1}]$. Esto significa que la probabilidad de emitir un determinado símbolo en un instante de tiempo t depende únicamente del estado s_t y es independiente de las observaciones anteriores.

A pesar de que estas dos propiedades implican una simplificación en el modelado de secuencias, en la práctica hacen que trabajar con este tipo de modelos sea asequible, reduciendo en gran medida el número de parámetros que se necesitan estimar. Un ejemplo de aplicación donde los HMM se han utilizado consiguiendo buenos resultados es el reconocimiento automático del habla [35]. Se introdujeron en la década de los 80, y se han convertido en la técnica más ampliamente utilizada en este campo hasta la actualidad, donde se empiezan a incorporar las redes neuronales en coexistencia con los HMM.

Capítulo 3

Tarea de segmentación Albayzín 2010

En este capítulo se describe de forma detallada la tarea de segmentación de audio propuesta para la evaluación Albayzín-2010 [36], sobre la que se sustenta este trabajo. Se introducirá la base de datos utilizada, así como las distintas clases acústicas a analizar y las métricas utilizadas para evaluar los resultados

3.1. Introdución

Las campañas de evaluación Albayzín son un conjunto de evaluaciones tecnológicas internacionales organizadas por la Red Temática de Tecnologías del Habla (RTTH) [37] con el objetivo de promover la investigación en diferentes temáticas relacionadas con las tecnologías del habla como puede ser reconocimiento automático del habla, diarización del locutor o identificación de idioma. En 2010 se introdujo por primera vez la tarea de segmentación automática de audio en estas evaluaciones. El objetivo de esta tarea es clasificar cada trama de audio como perteneciente a una única clase acústica de bajo nivel como pueden ser música, voz, ruido, o una combinación de estas.

3.2. Base de datos

Los datos con los que se trabaja en esta evaluación forman parte de un conjunto de programas de informativos emitidos por el canal de televisión catalán 3/24 TV en 2009. Los datos fueron recopilados por el *Language and Speech Technologies and Applications* (TALP) *Research Center* de la Universidad Politécnica de Cataluña y puestos a disposición de los participantes para la evaluación. A pesar de que todo el audio pertenece al dominio *broadcast*, la variabilidad que podemos encontrar en este tipo de contenidos suele ser bastante alta. Por ejemplo en un programa de informativos, si nos centramos en la voz, podemos encontrar desde voces grabadas en estudio con alta calidad, hasta conexiones telefónicas pasando por audio recogido en exteriores

La base de datos completa consta de unas 87 horas de audio muestreado a 16KHz que ha sido anotado manualmente (24 sesiones de unas 4 horas de duración cada una). Para llevar a cabo la evaluación se dividió la base de datos en dos subconjuntos: *train*, utilizado para entrenar cualquier modelo estadístico con el que se quiera afrontar la tarea, y *test* que servirá para evaluar los resultados obtenidos. En la Tabla 3.1 se resume la cantidad de horas de cada uno de estos subconjuntos

Subconjunto	Duración
train	58 horas (16 sesiones)
test	29 horas (8 sesiones)

Tabla 3.1: Duración de los subconjuntos *train* y *test* de la base de datos

La anotación manual de la base de datos se llevó a cabo en dos fases. En la primera fase se clasificó el audio con respecto a los sonidos de fondo (voz, música, ruido), condiciones de canal (estudio, telefónico, exterior), y locutores. En la segunda fase se generaron transcripciones literales del audio además de anotaciones de diferentes eventos acústicos (respiraciones, risas, articulaciones, pausas). Para la tarea de segmentación solo se utilizará la información de la primera fase de anotación. Teniendo en cuenta esta información, se definen 5 clases acústicas diferentes que se describen en la Tabla 3.2, y que serán las clases con las que trabajará en la evaluación.

Clase	Descripción
Voz [sp]	Voz en ambiente de estudio de un micrófono cercano
Música [mu]	Música entendida en un sentido general
Voz sobre música [sm]	Superposición de las clases de voz y música
Voz sobre ruido [sn]	Voz que no está grabada en condiciones de estudio o se encuentra solapada con algún tipo de ruido. Se incluye también en esta clase los momentos en los que varias voces se solapan.
Otros [ot]	Esta clase cataloga cualquier señal de audio que no se corresponde con ninguna de las otras cuatro clases acústicas

Tabla 3.2: Descripción de las cinco clases acústicas definidas para la tarea

La distribución de las clases acústicas en la base de datos se puede apreciar en la Figura 3.1. Podemos observar que existe un claro desbalanceo en la distribución de las clases ya que la mayoría de los datos contienen voz (92 % si sumamos voz, voz sobre música y voz sobre ruido), mientras que las clases que contienen música están representadas en un porcentaje muy bajo (20 % si sumamos música y voz sobre música).

El canal 3/24 TV emite su contenido principalmente en idioma catalán, es por esto que el 87 % de los fragmentos de voz están en catalán, mientras que el 13 % restante están en español. Finalmente, respecto a la distribución por género, un 63 % de los fragmentos de voz son de locutores varones frente a un 37 % que pertenecen a locutores mujeres.

3.3. Métrica de error y evaluación

La métrica de error propuesta para la evaluación se define como el error relativo promediado sobre todas las clases acústicas definidas. Se presenta en la ecuación (3.1):

$$\text{Error} = \frac{1}{N_c} \sum_{i=1}^{N_c} \left(\frac{dur(\text{FN}_i) + dur(\text{FP}_i)}{dur(\text{ref}_i)} \right) \tag{3.1}$$

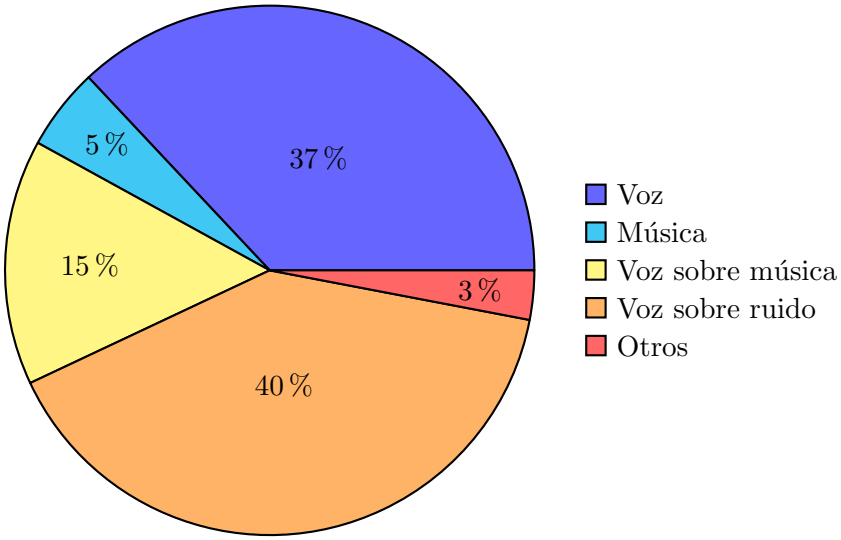


Figura 3.1: Distribución porcentual de las clases en la base de datos

donde $dur(\text{FN}_i)$ es la duración total de los errores de falso negativo para clase acústica i (predecir una clase distinta de la clase i, cuando realmente se trata de la clase i), $dur(\text{FP}_i)$ es la duración total de los errores de falso positivo para clase acústica i (predecir la clase i, cuando realmente se trata de una clase distinta de la clase i) y $dur(\text{ref}_i)$ es la duración total de clase acústica i según la referencia. N_c representa el número de clases acústicas, que para el cálculo de métricas es únicamente 4 ya que la clase "Otros" no se evalúa.

Un segmento de audio clasificado incorrectamente computará como un error de falso positivo para una clase acústica y un error de falso negativo para otra. Dado que la distribución de las clases está desbalanceada, los errores de cada clase acústica vendrán ponderados de forma distinta dependiendo de la duración total de la clase en la base de datos. Al tomar el promedio de todos los errores relativos de cada clase se estimula que los participantes no se centren únicamente en clasificar las clases mejor representadas.

A la hora de llevar a cabo la evaluación se excluyen las zonas donde la referencia presenta un cambio en la clase acústica con un margen de ±1s. Una representación esquematizada del método de evaluación se puede observar en la Figura 3.2. Esto se hace para tener en cuenta

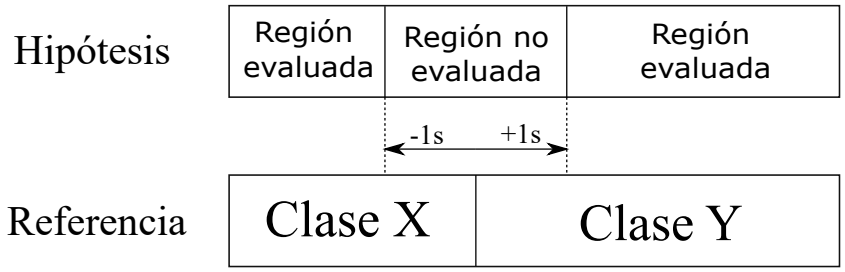


Figura 3.2: Representación esquemática de las regiones excluidas en la evaluación

posibles inconsistencias en la anotación humana de la base de datos y la incertidumbre sobre el punto de inicio o final de cada clase acústica.

Se introduce también otra métrica de error infuenciada por las evaluaciones de diarización

del locutor llevadas a cabo por NIST (*National Institute of Standards and Technology*) conocida como *Segmentation Error Rate* (SER) [38]. Esta métrica se define como el ratio entre la cantidad total de audio mal etiquetado y la cantidad total de audio evaluado. Los resultados finales que se obtengan en este TFM se presentarán teniendo en cuenta las dos métricas expuestas.

3.4. Resultados previos

En esta sección se presentan varios resultados obtenidos hasta el momento por diferentes sistemas de segmentación utilizando la base de datos Albayzín 2010. En la Tabla 3.3 se muestran los 3 mejores resultados de la evaluación original y el sistema propuesto por ViVoLab, y, de forma posterior a la evaluación, se compara con varios sistemas que han conseguido obtener buenos resultados.

Sistemas evaluación Albayzín'10			
Sistema	**Tipo de sistema**	**Características**	**Error**
Ganador evaluación Albayzín'10 (UPM/UC3M) [39]	HMM/GMM jerarquico	MFCC, *chroma* y entropía espectral	**30,20 %**
2o puesto evaluación Albayzín'10 (UVigo-GTM) [40]	ΔBIC + *clustering* aglomerativo	MFCC, 1a y 2a derivada	33,15 %
3er puesto evaluación Albayzín'10 (AVTS-UAM) [41]	HMM/GMM	MFCC, *Shifted Delta Coefficients*(SDC)	36,15 %
Sistema ViVoLab evaluación Albayzín'10 [42]	BIC + arboles de decisión + HMM/GMM	MFCC, 1a y 2a derivada	44,87 %
Mejores resultados posteriores a la evaluación			
Sistema	**Tipo de sistema**	**Características**	**Error**
Baseline publicación EURASIP (D.Castan et al) [23]	FA con matriz de compensación por clase	MFCC, 1a y 2a derivada	34,80 %
Mejor resultado publicación EURASIP (D.Castan et al) [23]	FA con matriz de compensación por clase + *Backend* gaussiano	MFCC, 1a y 2a derivada	**23.80 %**

Tabla 3.3: Resultados obtenidos con diferentes sistemas de segmentación en la base de datos Albayzín 2010 en términos de la métrica definida para la evaluación

Capítulo 4

Descripción del sistema de segmentación

En este capítulo se detalla el sistema de segmentación de audio desarrollado en este trabajo, presentando inicialmente una descripción general a nivel de diagrama de bloques, para posteriormente describir cada uno de sus componentes de forma pormenorizada.

4.1. Descripción general del sistema de segmentación

En términos generales, podemos describir el sistema de segmentación como un módulo que, partiendo de una señal de audio, genera una serie de etiquetas de segmentación que asignan a cada instante del audio una clase acústica de un conjunto predefinido.

Para llevar a cabo la tarea de segmentación, se propone un diagrama de bloques como el detallado en la Figura 4.1, en el que destacan tres bloques principales: un primer bloque de extracción de características del audio, la red neuronal que se encarga de clasificar el audio en las clases acústicas correspondientes, y un último bloque de resegmentación donde utilizaremos HMMs para refinar las fronteras de segmentación proporcionadas por la red neuronal.

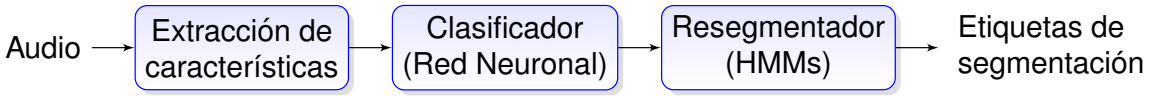


Figura 4.1: Diagrama de bloques del sistema de segmentación propuesto

A continuación se detallan cada uno de los tres bloques principales que componen el sistema de segmentación:

4.1.1. Extracción de características

Se trata del bloque de entrada al sistema de segmentación. Tal y como se ha explicado en la sección 2.1, la extracción de características se encarga de adaptar los datos al formato óptimo dada la tarea a realizar. Este bloque recibe como entrada las muestras de la señal de audio y obtiene una representación alternativa que permite maximizar sus propiedades discriminativas. En nuestro caso, a partir de la experiencia previa en el grupo de investigación en tareas de segmentación, las características con las que se trabajará serán los **bancos de filtros Mel**, expuestos en la sección 2.1.1. La parametrización utilizada usa ventanas de 25ms, con avance de ventana de 10ms.

La dimensionalidad de salida de este bloque es ajustable mediante el número de bandas en las que se analiza el espectro. Dado el conocimiento obtenido en otro conjunto previo de experimentos [12] el numero de bandas se fijará inicialmente en 32, pero se evaluará la posibilidad de incrementar este número en experimentos posteriores. En todos los casos, sumaremos un término extra para representar la energía de la señal en la ventana de análisis.

Además, con el objetivo de que la red neuronal trabaje con unos datos normalizados, se aplica lo que se conoce como CMVN (*Cepstral Mean & Variance Normalization*) a cada una de las 24 sesiones que componen la base de datos. Esta técnica permite obtener unas características con media nula y varianza unidad mediante la ecuación (4.1)

$$X_{\text{CMVN}} = \frac{X - \overline{X}}{\text{Var}(X)} \tag{4.1}$$

donde X es el vector de características original, $\overline{X}$ es el vector medio de características obtenido promediando todos las tramas temporales de una sesión, y $\text{Var}(X)$ es la varianza de los vectores de características de una sesión. Así, obtenemos X_{CMVN} para cada sesión de la base de datos y será con esta versión de la información con la que trabajaremos en las etapas posteriores.

4.1.2. Red neuronal

Es el elemento principal del sistema de segmentación. Dado un vector de características de entrada, realiza un procesado con el objetivo de clasificarlo como perteneciente a una de las clases acústicas definidas. Genera una etiqueta de salida por cada entrada, por tanto, ya que los vectores de características se generan cada 10 ms, tendremos una etiqueta de segmentación cada 10 ms.

Las arquitecturas neuronales que se evaluarán en este TFM tienen en común el uso de capas LSTM, en concreto nos centraremos en el uso de capas BLSTM dado que presentan una mejora de prestaciones frente a las capas LSTM simples. Como ya se ha explicado en la sección 2.3.3 este tipo de RNN son capaces de capturar dependencias de larga duración en secuencias temporales. Al usar este tipo de capas existen dos parámetros que debemos tener en cuenta:

- **Número de capas BLSTM**: define cuantas capas BLSTM se apilarán en la arquitectura neuronal. En los experimentos evaluaremos el uso de 1, 2 ó 3 capas para comprobar sus prestaciones.
- **Número de neuronas para las puertas**: las tres puertas que componen una celda de memoria LSTM trabajan con una capa lineal cuyo número de neuronas se puede configurar.

Además de las capas BLSTM, la arquitectura neuronal propuesta inicialmente constará de una capa *feed-forward* a la salida de la BLSTM que se encargará de reducir la dimensionalidad hasta 5 neuronas, cada una asociada a una clase acústica, y realizar el paso final en la clasificación. En la Figura 4.2 podemos observar un esquema de la red neuronal propuesta inicialmente para la tarea de segmentación. Con esta estructura, podemos relacionar cada clase acústica con la activación de una de las 5 neuronas de salida. Así, la etiqueta predicha por la red se obtendrá como el indice de aquella neurona que maximice su valor a la salida.

Para la fase de entrenamiento de la red se utilizará el subconjunto de datos *train* de la base de datos, compuesto por 58 horas de audio. No obstante, se reserva un 15 % de estos datos para validación del entrenamiento, obteniendo así un computo total de unas 49 horas de audio

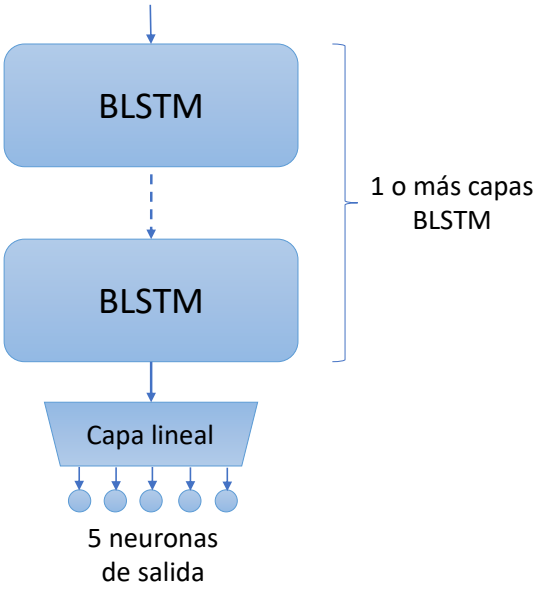


Figura 4.2: Esquema de la red neuronal propuesta inicialmente para la tarea de segmentación

para entrenamiento de la red y 9 horas para validación del entrenamiento. Para mejorar la capacidad de generalización del modelo el orden en que se presentan los datos es aleatorizado en cada iteración del proceso de entrenamiento.

Respecto al algoritmo de actualización de los pesos se ha optado por utilizar una versión evolucionada del descenso por gradiente estocástico conocida como *Adam* (*Adaptative Moment Estimation*) [43]. Este método calcula tasas de aprendizaje adaptativas para cada uno de los diferentes parámetros a partir de estadísticas de primer y segundo orden de los gradientes. Resultados empíricos han demostrado que *Adam* funciona bien en la práctica y que presenta una convergencia más rápida que otros métodos de optimización por gradiente [44].

Finalmente, mencionar que la implementación y evaluación de todas las arquitecturas neuronales desarrolladas en este TFM se ha llevado a cabo mediante la libreria *PyTorch* [9], con el lenguaje de programación *Python* como soporte.

4.1.3. Resegmentador

El sistema compuesto por el bloque de extracción de características y la red neuronal es capaz de emitir una hipótesis y obtener una serie de etiquetas de segmentación, por lo que se puede decir que ya es funcional. No obstante, dada la naturaleza de la salida de la red neuronal que puede presentar transiciones abruptas y falta de inercia en las decisiones, incorporamos este tercer bloque con el objetivo de aumentar la precisión en la tarea de segmentación y refinar las fronteras de decisión generadas por la red neuronal.

La resegmentación que proponemos esta basada en los Modelos Ocultos de Markov expuestos en la sección 2.4 y recibe como entrada, por un lado el conjunto de etiquetas de segmentación generadas por la red neuronal, y por otro lado la secuencia de vectores correspondiente a los valores de las 5 neuronas de salida de la red neuronal. Estos vectores, que contienen toda la información procesada por la red, son sobre los que se ha realizado la decisión para obtener las etiquetas ya que pueden interpretarse como una pseudoverisimilitud de las clases acústi-

cas. La salida de este bloque será, por tanto, un nuevo conjunto de etiquetas de segmentación generadas desplazando las fronteras de las iniciales.

El HMM propuesto se compone de 5 estados, uno por cada clase acústica. La probabilidad de emisión en cada estado se modela según una distribución gaussiana multivariante de dimensión 5 con su correspondiente vector de medias y matriz de covarianza completa, μ_i y Σ_i, respectivamente. En la Figura 4.3 se puede observar una representación esquemática del modelo implementado. Tal y como se puede apreciar, el modelo permite la transición desde un estado a cualquier otro estado.

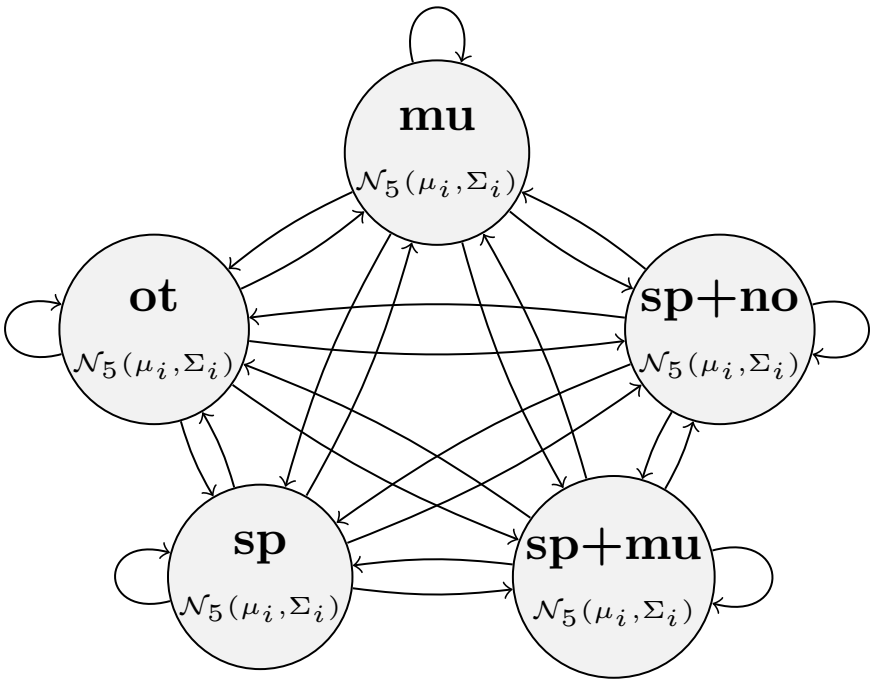


Figura 4.3: Representación esquemática del HMM de 5 estados propuesto para obtener la resegmentación de las etiquetas

Capítulo 5

Evaluación experimental del sistema

En este capítulo se presenta la evaluación experimental del sistema de segmentación desarrollado durante este TFM. Se describen todos los experimentos llevados a cabo y las diferentes estrategias y configuraciones implementadas, presentando así los resultados obtenidos además de su posterior análisis.

5.1. Experimentación inicial

En las fases iniciales de experimentación nos centramos en el núcleo del sistema de segmentación: la red neuronal, con el objetivo de evaluar diferentes arquitecturas neuronales y comprobar cuales son las mejores características de entrada para la red. Como punto de partida de la experimentación se pretende observar cómo influyen el número de capas BLSTM y el número de neuronas en el rendimiento del sistema de segmentación. Para ello, se entrena una red para cada configuración deseada según lo expuesto en la sección 4.1.2 y usando como entrada a la red bancos de filtros Mel de 32 componentes. Finalmente, se evalúa el error obtenido a la salida de la red neuronal sobre la partición de *test* de la base de datos. De experimentaciones previas a este TFM [12] se sabe que los valores razonables para el número de neuronas se encuentran entre 32 y 512. Teniendo esto en cuenta, aquí se evalúan 64, 128 y 256 neuronas. Cabe destacar que aumentar mucho más el número de neuronas resulta en problemas para gestionar el modelo en memoria, además de aumentar considerablemente la carga computacional. Respecto al número de capas BLSTM, se evalúan los casos donde hay una única capa, y los casos donde se apilan 2 y 3 capas BLSTM. Dado el limitado conjunto de datos de los que se disponen para el entrenamiento, aumentar cualquiera de los dos parámetros a barrer por encima de estos valores para intentar mejorar la capacidad de modelado podría resultar en una estimación incorrecta de los parámetros de la red neuronal.

Los resultados obtenidos para este experimento se pueden observar en la Tabla 5.1 donde se presenta el error total obtenido y el desglose del error por clase acústica para cada configuración evaluada. En términos generales se puede apreciar que a medida que aumenta el número de neuronas el error total disminuye notablemente. Si nos centramos en el número de capas BLSTM, la tendencia también es decreciente: un mayor número de capas se materializa en un menor error de segmentación en la mayoría de los casos, pero en este caso la disminución no es tan notable como para el número de neuronas. A la vista de que los resultados obtenidos para 256 neuronas son mejores para cualquier número de capas BLSTM, fijaremos el número de neuronas en 256 para el resto de experimentos por la mejora de prestaciones que supone.

Nº Neuronas	**Nº Capas BLSTM**	**Error(%)**				
		Total	**mu**	**sp**	**sp+mu**	**sp+no**
64	1	41,36	32,66	42,34	45,04	45,38
	2	41,39	34,01	40,68	45,97	44,91
	3	33,14	20,77	36,22	36,21	39,36
128	1	37,06	26,70	36,27	44,12	41,14
	2	32,06	22,96	32,60	35,69	37,01
	3	36,13	21,31	35,36	47,42	40,41
256	1	32,34	22,15	33,77	35,78	37,65
	2	31,00	21,00	32,24	34,30	36,44
	3	**30,35**	**22,58**	**30,55**	**32,79**	**35,49**

Tabla 5.1: Error de la red neuronal en la partición de *test* para diferentes configuraciones de Nº de neuronas y Nº de capas BLSTM

Teniendo en cuenta la Tabla 5.1 parece que la mejor combinación de parámetros se da al combinar 256 neuronas y 3 capas, sin embargo aquí se observa una única realización del entrenamiento. La inicialización aleatoria de los parámetros de la red hace que varias realizaciones diferentes obtengan resultados de segmentación diferentes debido al hecho de que los algoritmos de optimización por gradiente no garantizan su convergencia a mínimos globales. Por esto, podemos decir que un único entrenamiento no es suficientemente representativo de las prestaciones de la red. Para poder evaluar adecuadamente nuestro sistema, en los experimentos posteriores se realizan 5 entrenamientos diferentes para cada configuración a evaluar y se promediará el error obtenido. En la Tabla 5.2 se presentan los resultados de error promediados sobre 5 iteraciones para una configuración de 256 neuronas y diferente número de capas BLSTM. Observando el error promedio obtenido se puede concluir que existe una diferencia

Nº Neuronas	**Nº Capas BLSTM**	**Error promedio ± σ, N=5 (%)**				
		Total	**mu**	**sp**	**sp+mu**	**sp+no**
256	1	33,07±0,92	23,25	33,32	37,81	37,70
	2	30,53±0,99	19,41	32,40	34,20	36,10
	3	**29,57±0,47**	**20,14**	**30,66**	**32,38**	**35,09**

Tabla 5.2: Error de la red neuronal en la partición de *test* promediado sobre 5 entrenamientos diferentes para 256 neuronas y diferentes número de capas BLSTM

significativa al incrementar el número de capas BLSTM de 1 a 2, disminuyendo de un 33,37 % a un 30,53 % (mejora relativa del 8,51 %). Sin embargo la mejora que se obtiene al aumentar el número de capas de 2 a 3 es menor, con una mejora relativa del 3,24 %. Al realizar varias iteraciones del entrenamiento resulta interesante también comprobar la dispersión de los valores de error obtenidos. Para ello se presenta en la Figura 5.1 el diagrama de caja de los resultados obtenidos para una configuración de 256 neuronas y diferente número de capas BLSTM, donde en linea roja podemos observar la mediana y el intervalo azul representa el primer y el tercer cuartil. Las configuraciones de 1 y 2 capas presentan una mayor varianza, del 0,92 % y el 0,99 % respectivamente, mientras que la configuración de 3 capas reduce su varianza hasta un 0,47 %. Se comprueba que la diferencia entre utilizar 1 y 2 capas es estadísticamente significativa. Por otro lado, si nos centramos en la diferencia entre 2 y 3 capas, se aprecia que la mediana de la configuración con 3 capas esta contenida en el intervalo de varianza de la configuración de 2 capas por lo que su mejora resulta estadísticamente menos significativa.

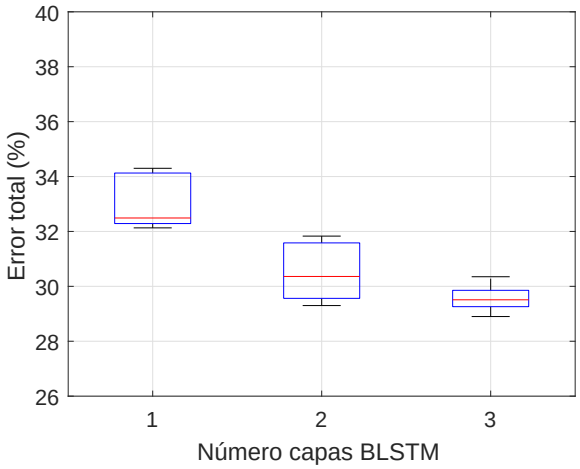


Figura 5.1: Diagrama de caja del error total obtenido para la partición de *test* en 5 entrenamientos diferentes con 256 neuronas y diferentes número de capas BLSTM

A la vista de esta información, y sabiendo que utilizar 3 capas supone un aumento de la carga computacional del modelo y de su número de parámetros libres, en próximos experimentos nos centraremos en los modelos de 1 y 2 capas BLSTM para comprobar las prestaciones de las siguientes estrategias de segmentación.

5.2. Exploración del espacio de características de entrada

Una vez comprobada la influencia del número de neuronas y del número de capas BLSTM en el error obtenido por la red neuronal se pretende comprobar la capacidad de modelado de la red aumentado la dimensionalidad de la entrada. Tal y como se explicó en la sección 4.1.1, el número de bandas en las que se analiza el espectro de la señal en los bancos de filtros Mel es un valor ajustable. En la fase inicial de experimentación se fijó en 32 bandas, pero en esta nueva fase experimental se explorará el espacio de características de entrada, aumentado este número de bandas con el objetivo de encontrar el valor óptimo para el sistema de segmentación. El procedimiento a seguir será el mismo que en el experimento anterior: se entrenan 5 redes diferentes y se evalúa cada una de ellas en la partición de *test* de la base de datos. En la Figura 5.2 se presentan los diagramas de caja obtenidos para el error total del modelo de 1 y 2 capas, utilizando 32, 64, 80 y 96 bandas de análisis en los bancos de filtros Mel.

Se comprueba que existe una tendencia decreciente del error promedio total a medida que se aumenta el número de bandas. Esto es coherente ya que estamos aumentando la resolución frecuencial en el análisis y, por tanto, aumenta la información disponible en las características de entrada. Sin embargo, aumentar demasiado el número de bandas puede hacer que el error aumente, como ocurre con el caso de 96 bandas que tiene unas medidas promedio de error similares a las mostradas por la configuración de 64 bandas. Un aumento de la dimensionalidad de entrada se traduce directamente en un aumento de los parámetros del modelo, que si se aumentan demasiado puede provocar que algunos de ellos no se estimen adecuadamente haciendo que aumente el error del sistema. Respecto a la diferencia entre 1 y 2 capas BLSTM podemos afirmar que la tendencia observada en el experimento anterior es consistente, ya que el error que se obtiene con 2 capas BLSTM es siempre inferior al obtenido con una única capa. El mejor resultado del experimento se obtiene para la configuración de 2 capas BLSTM y 80 bandas frecuenciales.

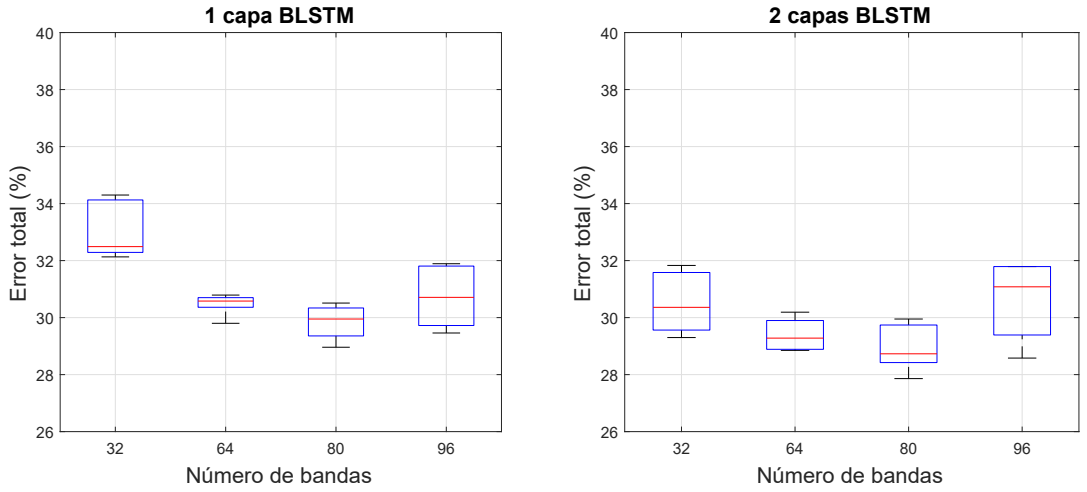


Figura 5.2: Diagrama de caja del error total obtenido para la partición de *test* en 5 entrenamientos diferentes con 256 neuronas, 1 y 2 capas BLSTM y diferente número de bandas

Los información mostrada en los diagramas de caja de la Figura 5.2 se complementa con los resultados mostrados en la Tabla 5.3, donde se desglosa el error total promedio obtenido para cada una de las clases acústicas a evaluar. Si comparamos la mejor configuración de 2

Nº bandas	**Nº Capas BLSTM**	**Error promedio ± σ, N=5 (%)**				
		Total	**mu**	**sp**	**sp+mu**	**sp+no**
32	1	33,07±0,92	23,25	33,32	37,81	37,70
	2	30,53±0,99	19,41	32,40	34,20	36,10
64	1	30,48±0,35	19,48	32,95	33,38	36,31
	2	29,40±0,52	18,11	31,98	32,54	34,99
80	1	29,84±0,56	19,06	32,07	32,94	35,29
	2	**28,96±0,35**	**19,05**	**31,44**	**30,92**	**34,45**
96	1	30,73±0,99	19,95	33,79	32,99	36,19
	2	30,58±1,27	21,00	32,71	33,34	35,27

Tabla 5.3: Error de la red neuronal en la partición de *test* promediado sobre 5 entrenamientos diferentes para 256 neuronas, 1 y 2 capas BLSTM y diferente número de bandas

capas BLSTM (80 bandas) con la peor (32 bandas) se observa que todas las clases acústicas disminuyen su error, aunque no todas lo hacen en la misma medida. Mientras que el error de la clase "Voz+Música" disminuye de un 34,20 % a un 30,92 % al aumentar el número de bandas (mejora relativa del 10,60 %), el error de la clase "Voz" disminuye de un 32,40 % a un 31,44 % (mejora relativa del 3,05 %). Esta diferencia se debe, en parte, a que al aumentar el número de bandas se aumenta la resolución frecuencial, especialmente en las zonas de alta frecuencia debido al comportamiento no lineal de los bancos de filtros Mel. Estas componentes de alta frecuencia son las que resultan mas discriminativas para poder diferenciar la música de la voz o del solape de ambas.

En general, se ha comprobado que aumentar la resolución frecuencial es beneficioso para la precisión del sistema de segmentación, consiguiendo reducir el error de clasificación promedio de un 30,53 % a un 28,96 %, lo que supone una mejora relativa del 5,42 % modificando únicamente el número de parámetros de la capa de entrada a la red.

Finalmente para cuantificar los errores se presenta en la Figura 5.3 la matriz de confusión normalizada obtenida para la mejor red neuronal evaluada con la configuración de 2 capas BLSTM y 80 bandas. Cada una de las filas se asocia a una de las etiquetas de la referencia mientras que cada columna con una clase predicha por el sistema. De esta forma podemos ver fácilmente cuales son las clases que más se confunden. El término de error más grande

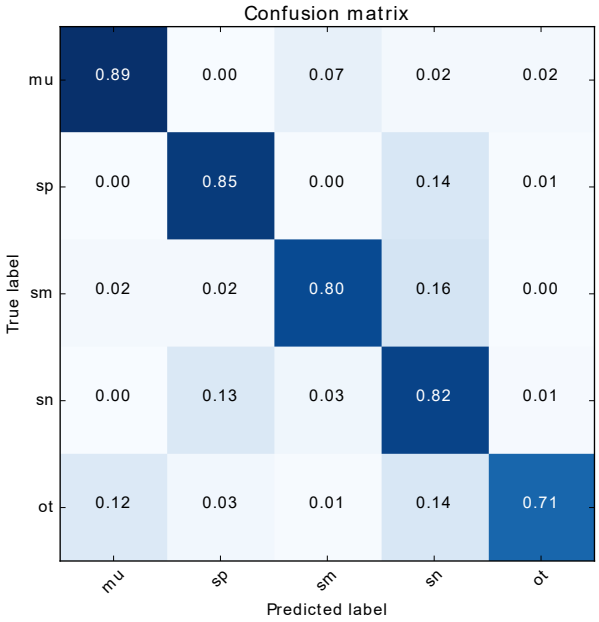


Figura 5.3: Matriz de confusión obtenida para la mejor red neuronal evaluada con 2 capas BLSTM, 256 neuronas y 80 bandas

que se puede ver es el que se corresponde con las tramas que realmente pertenecen a la clase "Voz+Música" y son etiquetadas como "Voz+Ruido", con un 16 % del total de las tramas pertenecientes a la clase "Voz+Música". Además resulta interesante comprobar que esto no ocurre al revés, ya que solo un 3 % de las tramas pertenecientes a la clase "Voz+Ruido" se clasifican erróneamente como "Voz+Música". Algo similar ocurre entre la clase "Voz" y "Voz+Ruido", donde un 14 % de las tramas realmente pertenecientes a la clase "Voz" se clasifican de forma errónea como ruidosas. En los próximos experimentos se tratará de corregir estos comportamientos incorporando nuevas estrategias sobre el sistema de segmentación.

5.3. Características *chroma*

Con el objetivo de corregir las confusiones del sistema para las clases que contienen música y las que no, se propone la incorporación de un nuevo conjunto de características que sean capaces de capturar la estructura particular de los fragmentos musicales: las características *chroma*. Estas características miden la distribución de energía de una señal a lo largo de una serie de conjuntos de tonalidades predefinidos. Son habitualmente utilizadas en el análisis y procesado de señales musicales ya que son capaces de modelar las características melódicas y armónicas de la música a la vez que son robustas a variaciones en el timbre y la instrumentación.

El fundamento teórico de estas características se basa en que dos sonidos cuya frecuencia difiere únicamente en una octava se perciben de forma similar por el ser humano. Basándonos en esta observación podemos separar cada sonido en dos componentes: por un lado la altura tonal y por otro lado la componente cromática o *chroma* [45]. Si consideramos como componentes *chroma* las 12 notas musicales usadas en la música occidental según su notación

anglosajona (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) se obtiene una representación en forma de hélice como la que se muestra en la Figura 5.4. Los coeficientes *chroma* aglomeran la

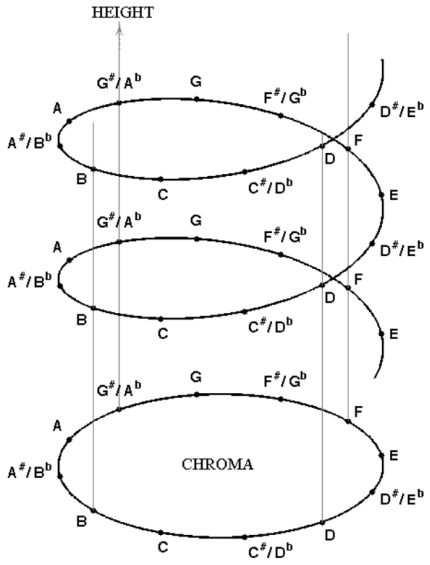


Figura 5.4: Representación altura tonal-*chroma* en el modelo de hélice frecuencial

información correspondiente a una componente *chroma* a lo largo de todas las alturas tonales para una ventana temporal especifica. Aplicando esto sobre todo el audio a analizar se obtiene una secuencia de coeficientes que expresa como se reparte la energía de la señal a través de las 12 componentes *chroma* definidas. En la Figura 5.5 se presenta, a modo de ejemplo, los coeficientes *chroma* asociados a la escala de C mayor interpretada por un piano. Se puede

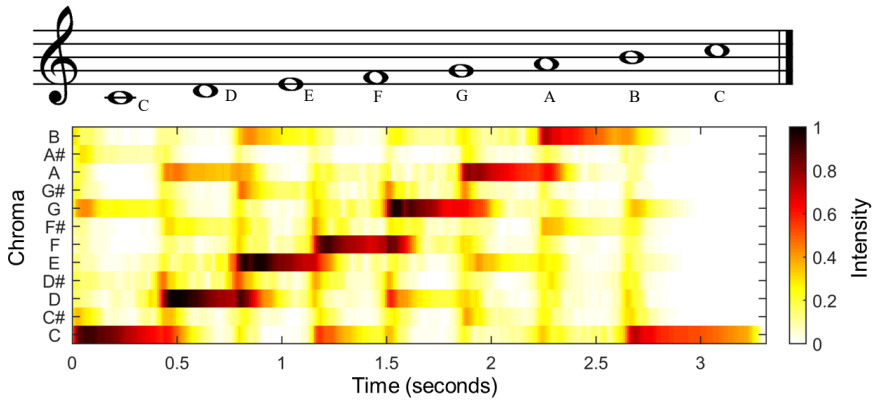


Figura 5.5: Coeficientes *chroma* para la escala de C mayor interpretada a piano

observar que tanto la primera nota C como la última, separadas por una octava, activan de forma similar la componente *chroma* asociada a dicha nota.

Por sus propiedades, las características *chroma* se han utilizado en varias aplicaciones de extracción de información musical como pueden ser el reconocimiento de acordes [46] o la sincronización y alineamiento de secuencias musicales [47]. La extracción de estas características se ha realizado mediante el software *openSMILE* (*Speech & Music Interpretation by Large-Space Extraction*) [48].

Estas nuevas características se concatenarán a las bancos de filtros Mel utilizados en experimentos anteriores generando un supervector que combina la información de ambas características y que se utilizará como nueva entrada a la red neuronal. Así, se realiza un nuevo conjunto de experimentos para evaluar las prestaciones de esta estrategia considerando unas características de entrada de 32 bandas+*chroma*, 64 bandas+*chroma*, 80 bandas+*chroma* y 96 bandas+*chroma*, y siguiendo la metodología expuesta en los experimentos anteriores. En la Tabla 5.4 se detallan los resultados obtenidos en términos de error promedio desglosado por cada clase acústica. Si comparamos estos resultados con los obtenidos en la Tabla 5.3 se

Configuración	**Nº Capas BLSTM**	**Error promedio ± σ, N=5 (%)**				
		Total	**mu**	**sp**	**sp+mu**	**sp+no**
32+*chroma*	1	30,30±1,09	21,60	32,79	31,48	35,34
	2	28,98±0,40	19,63	32,98	29,22	34,07
64+*chroma*	1	29,37±1,02	20,26	32,34	30,36	34,38
	2	28,02±1,45	19,00	30,60	29,13	33,34
80+*chroma*	1	29,73±0,54	19,45	33,35	31,26	34,88
	2	**27,31±0,57**	**18,16**	**30,82**	**27,21**	**32,95**
96+*chroma*	1	29,25±0,86	19,36	32,89	30,38	34,36
	2	28,11±0,74	19,55	31,54	27,29	33,18

Tabla 5.4: Error de la red neuronal en la partición de *test* promediado sobre 5 entrenamientos diferentes para 256 neuronas, 1 y 2 capas BLSTM y diferente número de bandas con coeficientes *chroma*

puede observar que el error total obtenido disminuye al incluir las características *chroma* para todas las configuraciones evaluadas. De forma similar al experimento anterior, se presenta

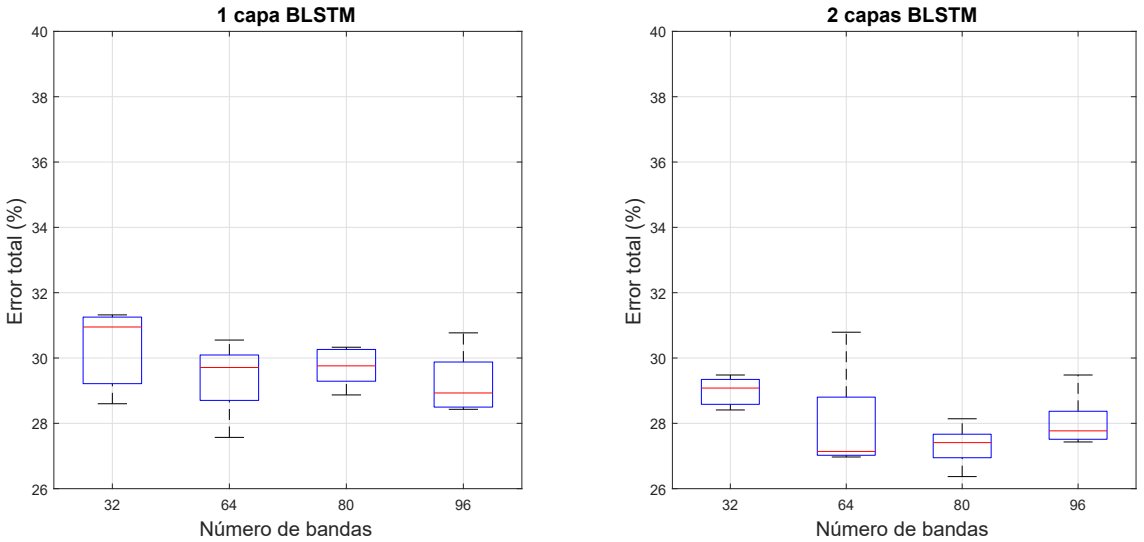


Figura 5.6: Diagrama de caja del error total obtenido para la partición de *test* en 5 entrenamientos diferentes con 256 neuronas, 1 y 2 capas BLSTM y diferente número de bandas con coeficientes *chroma*

en la Figura 5.6 el diagrama de caja del error total para las diferentes configuraciones. Las tendencias que se pueden observar son similares a las de experimentos anteriores: existe una mejora relevante entre utilizar 1 y 2 capas BLSTM, y el error disminuye conforme aumentamos el número de bandas de análisis alcanzando su mínimo en 80 bandas y empeorando cuando se aumenta a 96 bandas.

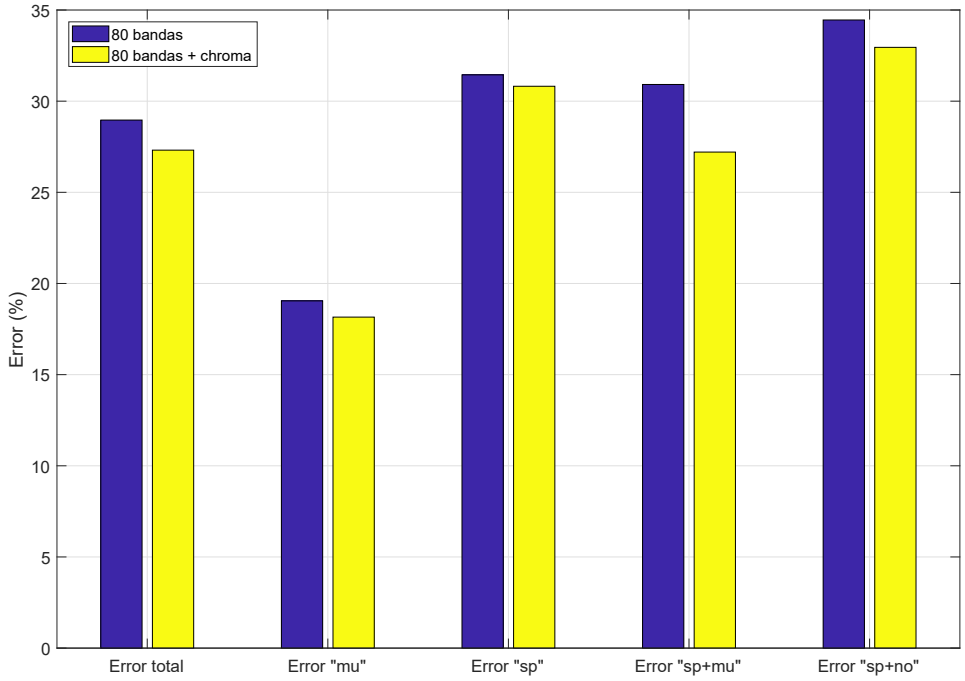


Figura 5.7: Comparativa de error promedio desglosada por clases entre la mejor configuración obtenida para las características sin *chroma* y con *chroma*

En la Figura 5.7 se comparan los resultados desglosados por clase acústica para la mejor configuración obtenida sin coeficientes *chroma* (80 bandas, 2 capas BLSTM) y la mejor configuración obtenida con coeficientes *chroma* (80 bandas, 2 capas BLSTM) con el objetivo de reflejar la influencia de estas nuevas características en el error de la red neuronal. Se observa que el error disminuye para todas las clases al incluir las características *chroma*, por lo que podemos concluir que el uso de estas características resulta apropiado para el sistema de segmentación. La clase "Voz+Música" es la que presenta una mayor mejoría, disminuyendo su error promedio de un 30,92 % a un 27,21 % (mejora relativa del 13,63 %), mientras que el error total del sistema del segmentación disminuye de un 28,96 % a un **27,31** % (mejora relativa total del 6,04 %), siendo el mejor resultado obtenido hasta el momento en este TFM.

Por último, en la Figura 5.8 se presenta la matriz de confusión obtenida para la mejor configuración evaluada al incluir las características *chroma* con el objetivo de poder comparar como se ha modificado la confusión de las clases respecto de la Figura 5.3, obtenida antes de incluir estas nuevas características. Se puede observar que el término de confusión de "Voz+Música" con "Voz+Ruido" se ha podido reducir de un 16 % a un 13 %, a la vez que el término de confusión de "Voz" con "Voz+Música" disminuye de un 14 % a un 11 %. Además el 12 % de las tramas de la clase "Otros" que eran erróneamente clasificados como "Música" se ve reducido hasta un 6 %. Por tanto, podemos afirmar que la incorporación de las características *chroma* resulta adecuada para poder separar aquellas clases que contengan música.

Una vez analizados los resultados obtenidos con las características *chroma* y dada la mejora de prestaciones que suponen, en el resto de experimentos a realizar se utilizará como entrada al sistema de segmentación la combinación de los bancos de filtros Mel de 80 bandas y las características *chroma*.

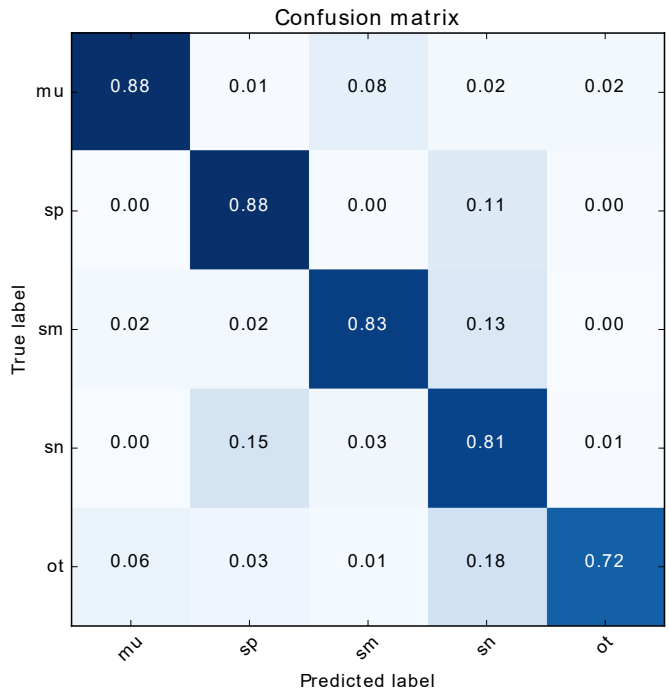


Figura 5.8: Matriz de confusión obtenida para la mejor red neuronal evaluada con 2 capas BLSTM, 256 neuronas, 80 bandas y coeficientes *chroma*

5.4. Capas ocultas y refuerzo del contexto a corto plazo

En este punto, ya se ha experimentado con varias posibilidades que ofrecen las características de entrada para el sistema de segmentación, evaluando la posibilidad de aumentar la resolución frecuencial e incorporando un nuevo tipo de características para incrementar las capacidades discriminativas del modelo. Se propone ahora modificar la arquitectura neuronal propuesta inicialmente con el objetivo de intentar mejorar la precisión de modelo. Por un lado se evaluará la opción de añadir varias capas ocultas tras las capas BLSTM. Por otro lado se realizará una serie de experimentos para dotar a la última capa de la red, encargada de realizar la clasificación final, de información de las tramas adyacentes y así reforzar el contexto a corto plazo en la decisión final. Teniendo en cuenta que en todos los experimentos anteriores la arquitectura compuesta por 2 capas BLSTM se comporta mejor que la formada por una única capa, en este nueva serie de experimentos se evaluará solo la opción de 2 capas BLSTM.

5.4.1. Incorporación de capas ocultas

La primera opción considerada para aumentar la capacidad de modelado de la red neuronal es aumentar el número de capas ocultas. En los experimentos previos la salida de la última capa BLSTM era clasificada a través de una única capa lineal para obtener la salida de la red. Se plantea en este experimento añadir dos capas ocultas con activación no lineal según una función ReLu que procesen la información de las capas BLSTM antes de que se realice la clasificación final.

En la Tabla 5.5 se muestran los resultados obtenidos siguiendo la misma metodología que la utilizada durante el resto de experimentos. Se puede comprobar que, respecto de la mejor configuración obtenida en los experimentos con las características *chroma*, los resultados sufren una ligera degradación ($27{,}31 \pm 0{,}57$ frente a $27{,}67 \pm 1{,}09$). El error promedio aumenta y, además, la varianza se duplica. Esto nos indica que este tipo de configuraciones por si solas

Configuración	Nº Capas BLSTM	Error promedio ± σ, N=5 (%)				
		Total	mu	sp	sp+mu	sp+no
2 capas ocultas	2	27,67±1,09	18,65	31,05	28,20	32,72

Tabla 5.5: Error de la red neuronal en la partición de *test* promediado sobre 5 entrenamientos diferentes para 256 neuronas, 2 capas BLSTM y 2 capas ocultas

no resultan adecuadas, ya que se aumenta el número de parámetros libres. Al disponer de una cantidad limitada de datos para el entrenamiento de la red neuronal, muchos de estos parámetros no quedarán bien estimados, provocando así una perdida de prestaciones. En la próxima serie de experimentos se intentará combinar esta estrategia con otras que aporten información contextual a la red.

5.4.2. *Stacking* o apilado temporal

Las señales de audio suelen tener una alta correlación temporal, es decir, dada una clase acústica en un instante temporal t es probable que instantes temporales cercanos pertenezcan a la misma clase. Las redes BLSTM ya son capaces de capturar relaciones contextuales a corto, medio y largo plazo. En esta serie de experimentos se propone, además, reforzar la información de contexto a corto plazo en las últimas capas de la red neuronal para que se pueda tener en cuenta en la clasificación final.

Una forma simple de generar un vector de características que tenga información de contexto es apilar las tramas adyacentes para generar un supevector que contenga la trama del instante actual t, las k tramas adyacentes por la izquierda $(t-1, t-2, ..., t-k)$ y las k tramas adyacentes por la derecha $(t+1, t+2, ..., t+k)$, donde k será el contexto temporal que queremos tener en cuenta. Esta técnica se ejemplifica en la Figura 5.9 para un contexto $k = 1$. En términos generales, si tenemos un vector de características de dimensión D, al aplicar el apilado temporal con un contexto k obtendremos un nuevo vector de dimensión $(2k+1)D$.

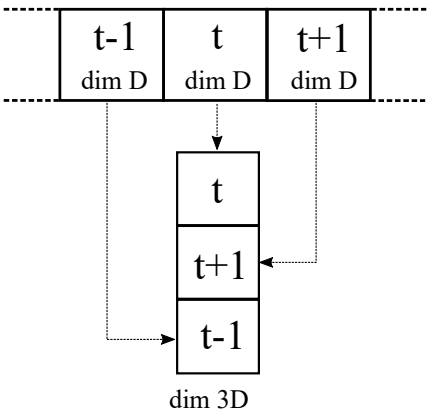
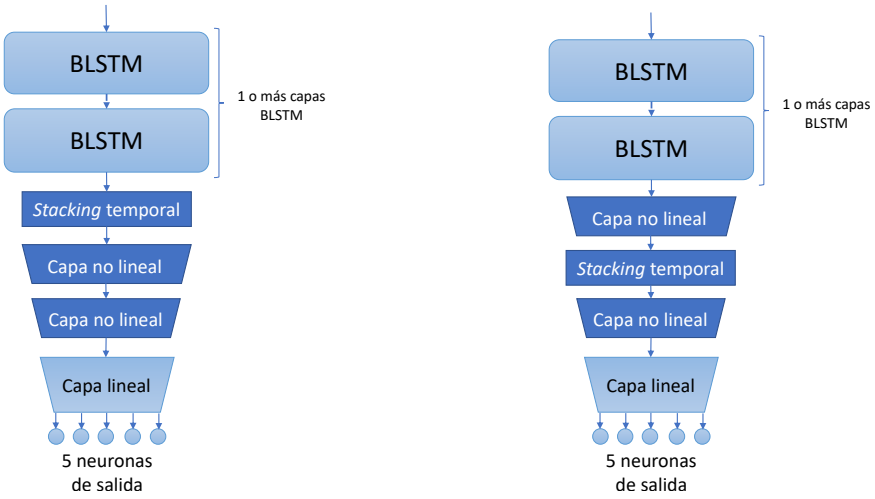


Figura 5.9: Esquema de la generación de características con contexto

Tomando como base este concepto e incorporando además la estrategia de las capas ocultas del experimento anterior, se implementan dos arquitecturas neuronales diferentes que se muestran en la Figura 5.10. Por un lado, en la Figura 5.10a se muestra una arquitectura donde el apilado temporal se aplica sobre las características de salida de la última capa BLSTM, mientras que en la Figura 5.10b se muestra otra arquitectura alternativa donde se aplica el apilado temporal sobre una versión comprimida de la información de salida de la última capa BLSTM, que ha pasado a través de una capa no lineal con activación ReLu. Esta arquitectura

tiene como objetivo evitar el aumento excesivo de la dimensionalidad trabajando con una versión previamente comprimida de la información de salida de la capa BLSTM. Los resultados



(a) Apilado temporal a la salida de la última capa BLSTM

(b) Apilado temporal tras compresión de dimensionalidad mediante capa no lineal

Figura 5.10: Esquema de las arquitecturas neuronales con apilado temporal implementadas

obtenidos para ambas arquitecturas evaluando diferentes tamaños de contexto se presentan en la Tabla 5.6. Estos resultados se complementan a su vez con el diagrama de caja mostrado en la Figura 5.11 para los 5 entrenamientos diferentes realizados para cada configuración. Respecto

Configuración	**Contexto**	**Error promedio ± σ, N=5 (%)**				
		Total	**mu**	**sp**	**sp+mu**	**sp+no**
Apilado salida BLSTM	$k = 1$	**26,61±0,92**	**17,78**	**29,61**	**27,01**	**32,03**
	$k = 2$	27,94±0,85	18,97	30,20	29,45	33,12
Apilado salida capa no lineal	$k = 1$	27,33±1,02	18,60	31,60	27,22	32,44
	$k = 2$	27,21±0,57	17,32	31,01	27,81	32,70
	$k = 3$	27,42±1,60	18,07	29,84	29,31	32,45

Tabla 5.6: Error de la red neuronal en la partición de *test* promediado sobre 5 entrenamientos diferentes para 256 neuronas, 2 capas BLSTM y diferentes configuraciones de apilado temporal

a la configuración donde se aplica el apilado a la salida de la capa BLSTM se puede comprobar que existe una diferencia significativa entre utilizar un contexto $k = 1$ y $k = 2$. Esto es debido a que, al trabajar con las características de salida de la capa BLSTM que tienen una alta dimensionalidad ($D = 256$), la dimensión de salida del apilado aumenta muy rápidamente si aumentamos el contexto ($D = 768$ para $k = 1$ frente a $D = 1280$ para $k = 2$). Este aumento de dimensionalidad se traduce directamente en un aumento del número de parámetro del modelo. Cabe destacar también que esta configuración consigue el mejor resultado promedio hasta el momento con un contexto $k = 1$, consiguiendo una mejora relativa del 2,63 % respecto del mejor resultado obtenido al incluir las características *chroma* (26,61 % frente a 27,31 %).

Si nos centramos en la configuración donde se introduce el apilado tras aplicar una capa no lineal, se puede ver que los resultados promedios obtenidos para todos los contextos evaluados son similares, es decir, no tienen una fuerte dependencia del contexto como ocurría en el caso anterior. La red neuronal dispondrá de una menor cantidad de información en bruto

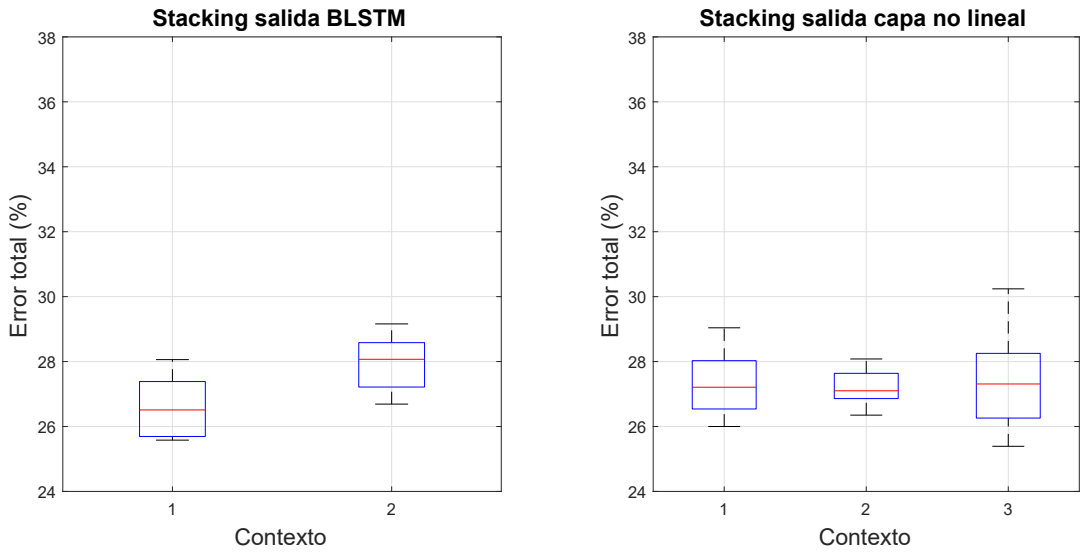


Figura 5.11: Diagrama de caja del error total obtenido para la partición de *test* en 5 entrenamientos diferentes con 256 neuronas, 2 capas BLSTM y diferentes configuraciones de apilado temporal

para generar el contexto, lo que podría provocar este comportamiento insensible al aumento del contexto. Además, las prestaciones obtenidas con esta nueva configuración se encuentran en valores muy cercanos a los obtenidos en los experimentos realizados al introducir las características *chroma*, resultando en mejoras prácticamente nulas.

En resumen, se ha comprobado que aplicar el apilado temporal directamente sobre la salida de la última capa BLSTM resulta beneficioso para las prestaciones del sistema. Se prueba así también que las técnicas que refuerzan el contexto a corto plazo en la clasificación final resultan en un aumento de la precisión. En la siguiente serie de experimentos se evaluará otra técnica para aportar contexto a corto plazo alternativa basada en capas convolucionales.

5.4.3. Capas convolucionales

La segunda estrategia propuesta para incorporar información de contexto a corto plazo en la clasificación es modificar las capas *feed-forward* tradicionales que se han venido usando en todos los experimentos anteriores por capas convolucionales. Tal y como se ha explicado en la sección 2.3.4, este tipo de capas constan de una serie de filtros entrenables que procesan la información en función de dependencias de alcance limitado. Este alcance se puede controlar mediante el tamaño de los filtros que componen la capa.

Como únicamente estamos interesados en las relaciones locales que existen a lo largo de la dimensión temporal recurriremos a capas convolucionales compuestas por filtros unidimensionales, a diferencia de los filtros matriciales utilizados en el ámbito de procesado de imagen, cuyo objetivo es capturar la correlación que existe en píxeles adyacentes de una imagen. Se implementarán entonces dos arquitecturas diferentes: una primera en la que únicamente se sustituirá una de las capas *feed-forward* tras la BLSTM por una capa convolucional 1D, y otra en la que todas las capas tras la BLSTM serán sustituidas por capas convolucionales 1D. A pesar de que la implementación difiere, esta última estrategia es equivalente al apilado temporal tras la capa BLSTM, tanto en parámetros libres como en tamaño del contexto, por lo que esperamos que los resultados sean similares. Aplicar un filtro de tamaño $N = 3$ es equivalente a tener en

cuenta la muestra actual, la anterior y la siguiente. Algo similar ocurre con un filtro de tamaño $N = 5$ que tiene en cuenta las dos muestras anteriores, la actual y las dos siguientes. En la Tabla 5.7 se presentan los resultados obtenidos para este nuevo conjunto de experimentos, cuya información se complementa con la Figura 5.12 donde podemos ver el diagrama de caja del error obtenido para 5 entrenamientos diferentes. Respecto a la configuración donde se utilizan

Configuración	**Tamaño filtros**	**Error promedio ± σ, N=5 (%)**				
		Total	**mu**	**sp**	**sp+mu**	**sp+no**
1 capa conv1D + 2 capas *feed-forward*	$N = 3$	27,11±0,66	18,85	29,92	27,39	32,30
3 capas conv1D	$N = 3$	**26,77±0,54**	**17,45**	**30,58**	**26,73**	**32,32**
	$N = 5$	27,90±0,84	18,09	30,88	29,67	32,97

Tabla 5.7: Error de la red neuronal en la partición de *test* promediado sobre 5 entrenamientos diferentes para 256 neuronas, 2 capas BLSTM y diferentes configuraciones de capas convolucionales

3 capas convolucionales podemos constatar un comportamiento similar a la configuración de apilado tras la capa BLSTM: de la misma forma se comporta peor al aumentar el contexto a tener en cuenta y arroja unos resultados promedio muy similares. Sin embargo la dispersión de los resultados mejora en esta implementación. Respecto a la configuración donde solo se

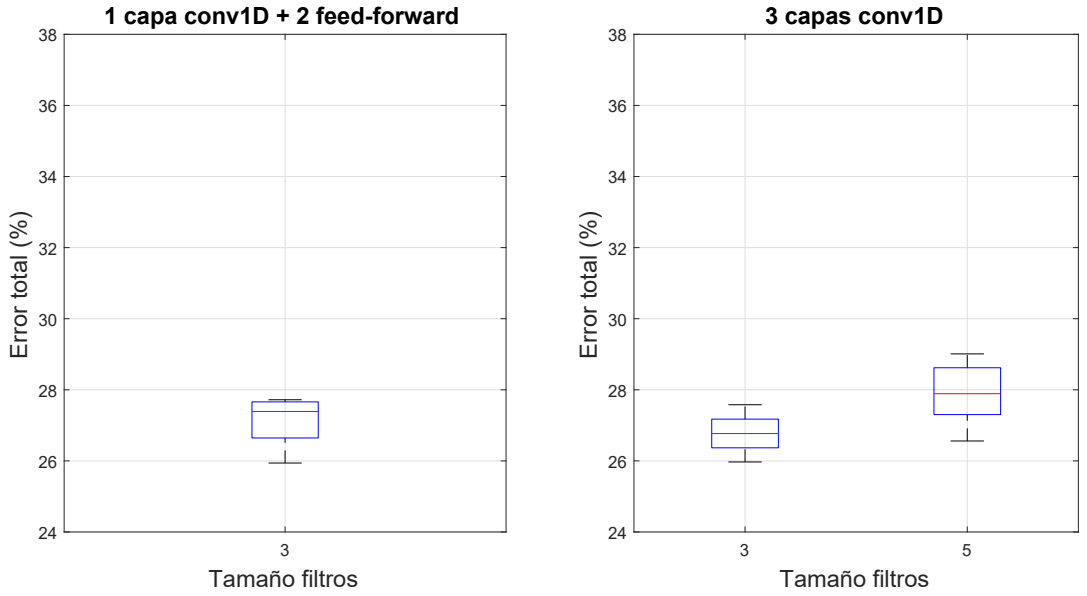


Figura 5.12: Diagrama de caja del error total obtenido para la partición de *test* en 5 entrenamientos diferentes con 256 neuronas, 2 capas BLSTM y diferentes configuraciones de capas convolucionales

utiliza una única capa convolucional, esta obtiene peores resultados. Esto se debe a que ve un contexto más reducido que la otra configuración, donde al apilar varias capas convolucionales hacemos que el contexto aumente progresivamente. Con todos estos resultados en mente podemos concluir que, tanto las estrategias de apilado como las capas convolucionales ofrecen resultados muy similares que mejoran respecto de los experimentos de la sección 5.3, lo que constata la mejora de prestaciones que supone el uso de técnicas de refuerzo del contexto a corto plazo para la tarea de segmentación.

5.5. Resegmentación con HMMs

En todos los experimentos anteriores se han evaluado diferentes estrategias y combinaciones de parámetros para maximizar las prestaciones de la red neuronal en términos de precisión en la tarea de segmentación. Aplicando diferentes características de entrada y diferentes arquitecturas se ha conseguido disminuir el error promedio de la red neuronal desde un 29,57 %, usando únicamente 32 bandas frecuenciales, hasta un 26,61 %, aumentando el número de bandas e incorporando las características *chroma* y diversas técnicas de refuerzo del contexto a corto plazo. Todo esto se traduce en una mejora relativa del 11,12 %. Sin embargo, aún no se ha tenido en cuenta la última fase de sistema: el resegmentador. En esta sección se realizarán una serie de experimentos para caracterizar experimentalmente este bloque buscando su configuración óptima de parámetros.

Para conseguir aumentar la inercia del sistema en la generación del etiquetado de segmentación se incluye en el modelo HMM propuesto el uso de estados atados (*tied states*) [49]. En este tipo de configuraciones cada uno de los estados del HMM representados en la Figura 4.3 esta compuesto a su vez por un número N de subestados que comparten la misma distribución de probabilidad. Una vez se alcanza el primer subestado, el modelo fuerza la transición en una topología de izquierda a derecha por todos los subestados hasta alcanzar el último, momento en el que se puede transitar a otro estado. Esta topología se muestra de forma esquemática en

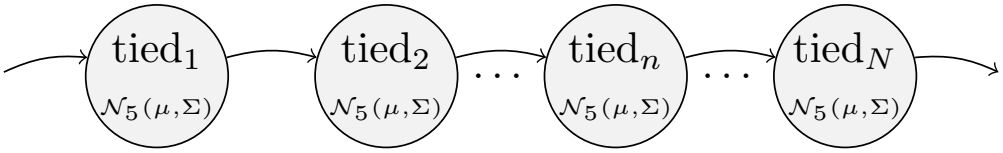


Figura 5.13: Representación esquemática de una topologia izquierda-derecha basada en N estados atados

la Figura 5.13. De esta forma, configurando el número de estados atados podemos modificar la duración mínima de las etiquetas antes de una transición entre clases.

La red neuronal proporciona una salida cada 10ms, lo cual puede resultar en estimaciones ruidosas de las transiciones y cambios abruptos en el etiquetado. Para reducir la resolución temporal y suavizar la salida de la red neuronal se propone realizar un submuestreo de la información antes de proporcionársela al resegmentador. En el caso de las etiquetas de entrada tomaremos solo una de cada L, mientras que en el caso de las pseudoverosimilitudes de la red neuronal aplicaremos previamente un filtro promediador de orden L mediante un filtrado *forward-backward* [50] para evitar distorsionar las componentes de fase de la señal, tomando una de cada L muestras a la salida.

Supongamos la secuencia de vectores $X = \{x_0, x_1, ..., x_t, ..., x_N\}$ que representa las pseudoverosimilitudes de la red neuronal para cada instante de tiempo t, y la secuencia de etiquetas generada por la red neuronal $Y = \{y_0, y_1, ..., y_t, ..., y_N\}$ que asigna una clase acústica c a cada instante de tiempo t. El vector de medias y la matriz de covarianza de la distribución gaussiana multivariante asociada al estado c se calculará mediante los estimadores de máxima verosimilitud según las ecuaciones (5.1) y (5.2) respectivamente,

$$\hat{\mu}_c = \frac{1}{N_c} \sum\nolimits_{\{t|y_t=c\}} x_t \tag{5.1}$$

$$\hat{\Sigma}_c = \frac{1}{N_c} \sum_{\{t|y_t=c\}} (x_t - \hat{\mu}_c)(x_t - \hat{\mu}_c)^T \tag{5.2}$$

donde N_c representa el número de tramas de X etiquetadas como pertenecientes a la clase acústica c. De este modo no es necesaria ninguna información adicional más que la proporcionada por la red neuronal para generar las distribuciones estadísticas del modelo HMM. Esta estimación se realiza a nivel de fichero, obteniendo así cada fichero de la base de datos sus propios vectores de medias y sus propias matrices de covarianzas.

Para caracterizar experimentalmente el bloque de resegmentación se llevó a cabo un experimento barriendo los dos parámetros explicados anteriormente: el factor de submuestreo y el número de subestados por estado. Se tomará como entrada del resegmentador los resultados del experimento con 3 capas convolucionales de la sección 5.4.3, ya que fueron uno de los mejores resultados obtenidos en este TFM, presentando además una menor varianza. Los resultados obtenidos al aplicar la resegmentación se recogen en la Figura 5.14, donde se puede observar la mejora relativa del error promedio respecto de la salida de la red neuronal en función de L y el número de estados atados. El mejor resultado obtenido en este experimento consigue una

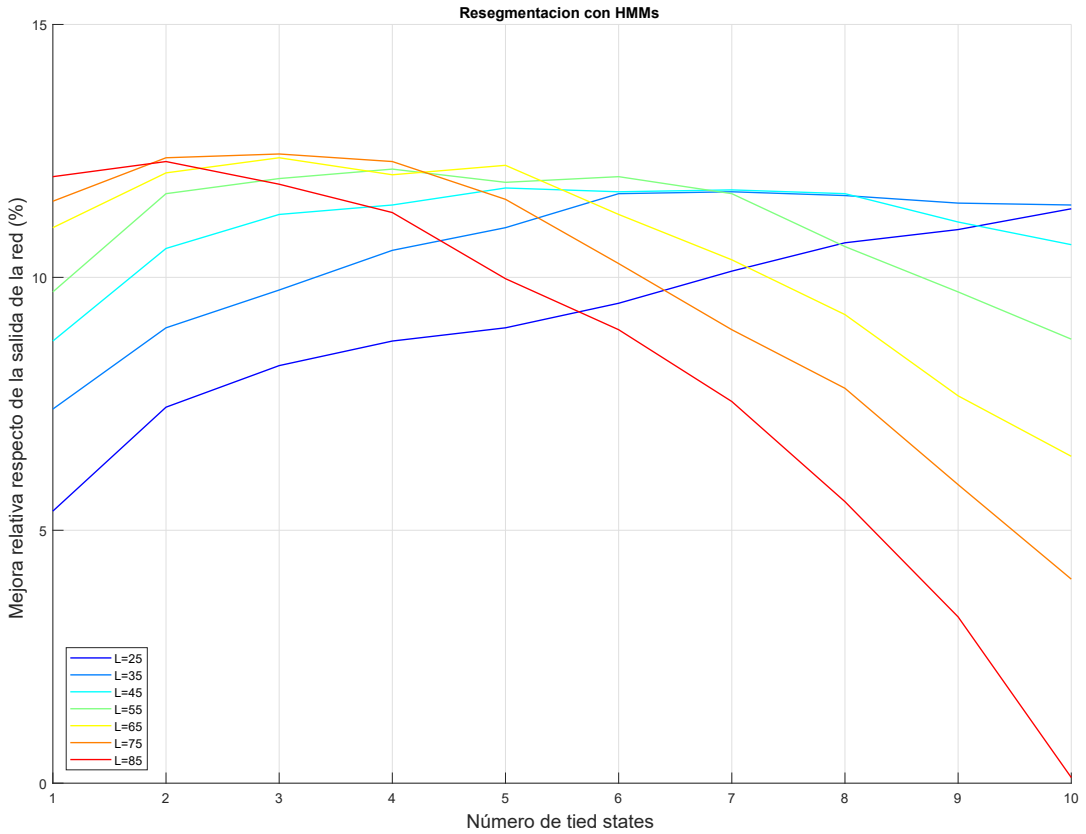


Figura 5.14: Mejora relativa del error promedio respecto de la salida de la red neuronal aplicando la resegmentación con modelos HMM en función del factor de submuestreo L y el número de *tied states*

mejora relativa del 12,44 % con una configuracion de $L = 75$ y 3 subestados atados. Respecto a las tendencias que podemos observar, para valores bajos de L como 25 y 35 aumentar el número de subestados hace que el error disminuya hasta alcanzar un valor mínimo. A medida que aumentamos el valor del factor de submuestreo L, la tendencia que se observa para los subestados se revierte, pasando a aumentar el error a medida que se incrementan el número de *tied states*. Esto se debe a que estamos aumentando de forma drástica la inercia en la emisión

de etiquetas, lo que puede reducir la precisión del etiquetado de salida. Este es el caso, por ejemplo, de la configuración con $L = 85$, donde vemos que la mejora relativa obtenida decae muy rápidamente cuando incrementamos los subestados.

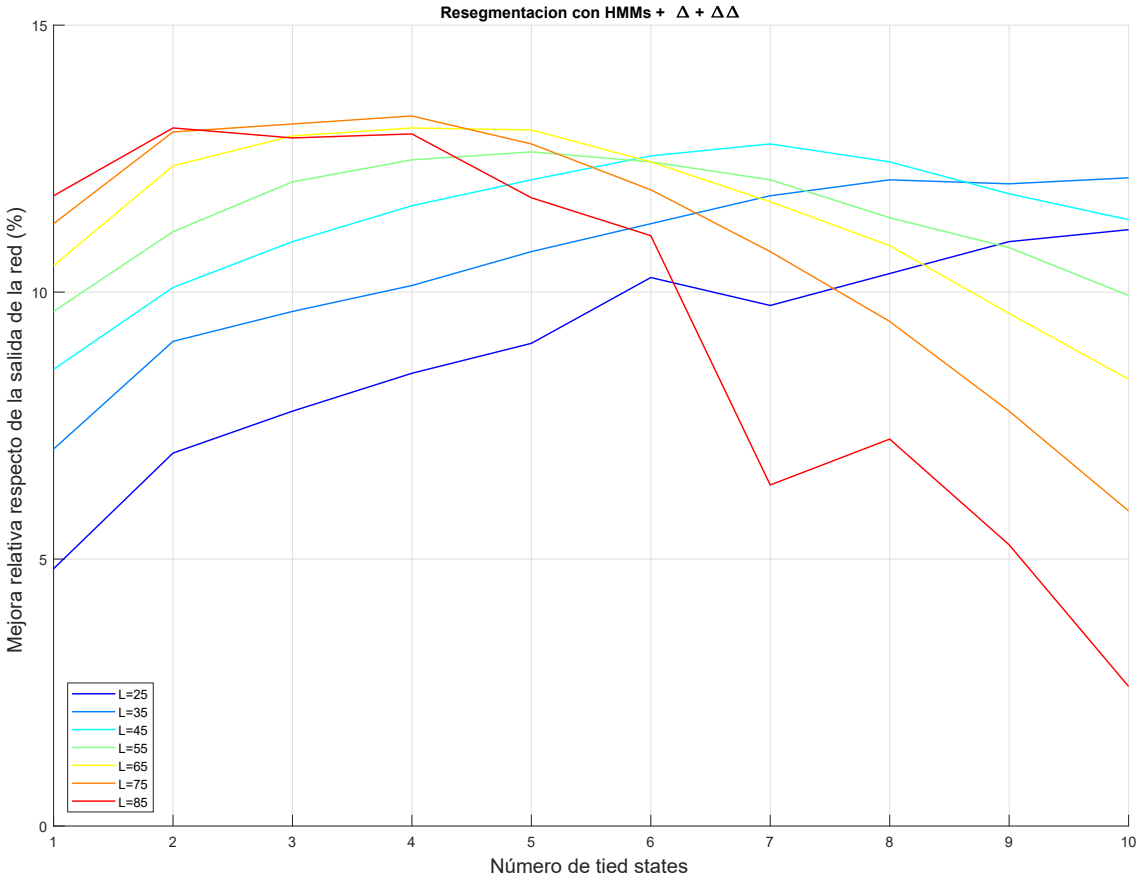


Figura 5.15: Mejora relativa del error promedio respecto de la salida de la red neuronal aplicando la resegmentación con modelos HMM incorporando 1a y 2a derivada en función del factor de submuestreo L y el número de *tied states*

Comprobado el buen funcionamiento del modelo HMM para la resegmentación de las etiquetas se propone una segunda iteración del modelo que incorpora también la información de la primera y segunda derivada de las pseudoverosimilitudes de salida de la red neuronal para ayudar al modelo a entender la dinámica de los datos. Dichas derivadas se obtienen aplicando dos filtros paso alto de orden 9 sobre la salida de la red neuronal para después concatenarse a los datos originales. Con este sistema modificado se lleva a cabo un experimento similar al realizado anteriormente. Los resultados obtenidos se pueden ver en la Figura 5.15, donde se presenta la mejora relativa del error promedio respecto de la salida de la red neuronal en función del factor de submuestreo L y el número de *tied states*. En general los resultados obtenidos son muy similares a los del experimento anterior, pudiendo observar unas tendencias muy similares. En este caso el mejor resultado evaluado obtiene una mejora relativa del 13,30 % respecto de la salida de la red neuronal para una configuración de $L = 75$ y 4 *tied states*, frente al 12,44 % obtenido en el experimento anterior con unos parámetros similares. Esto supone una ligera mejora de las prestaciones al incorporar la información de la primera y segunda derivada en el modelo HMM.

Con el objetivo de ilustrar como influye la inercia impuesta por el modelo HMM en la precisión del etiquetado de salida se presenta en la Figura 5.16 el diagrama de dispersión de

la mejora relativa del error promedio frente al tiempo mínimo de permanencia en una clase acústica forzada por el modelo. Este tiempo en segundos se calcula como el producto de 3 términos según la ecuación (5.3),

$$T_{min} = T_s L N_{ts} \tag{5.3}$$

donde T_s representa el periodo de muestreo del etiquetado de entrada al bloque de resegmentación, que en nuestro caso será siempre de 10ms, L es el factor de submuestreo aplicado, y N_{ts} es el número de subestados atados configurado en el modelo. Se puede apreciar que aquellas

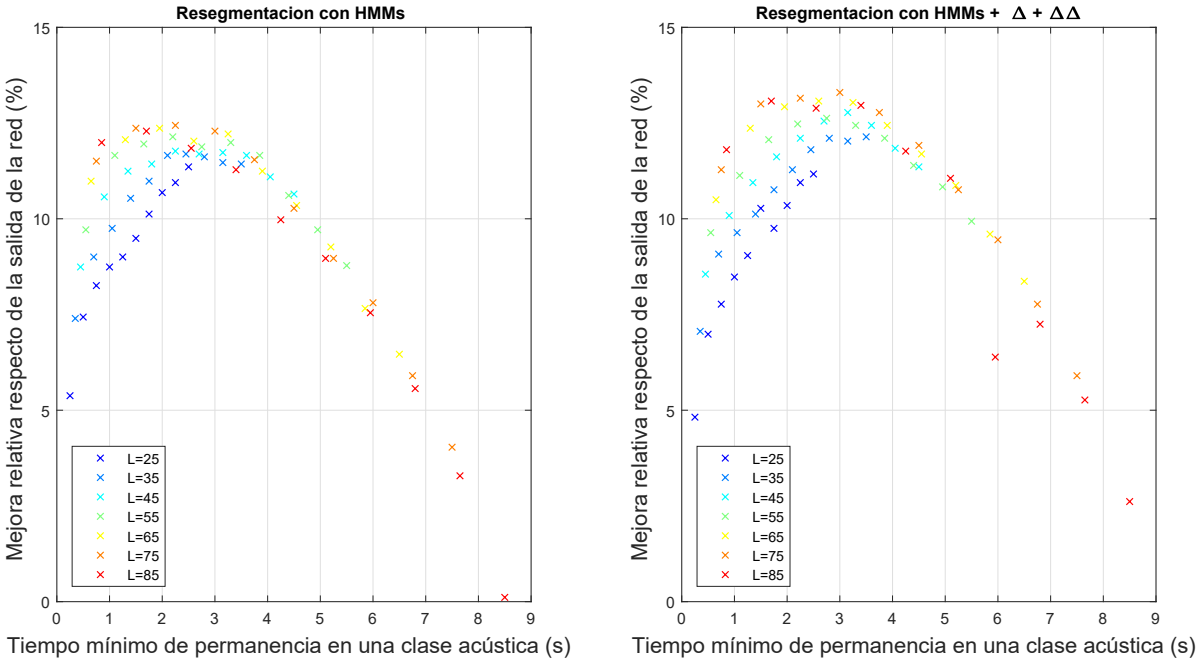


Figura 5.16: Mejora relativa del error promedio respecto de la salida de la red neuronal frente al tiempo mínimo de permanencia en una clase acústica para el modelo HMM sin derivadas (izda) y con derivadas (dcha)

configuraciones que obtienen una mayor mejora relativa respecto de la salida de la red neuronal se encuentran agrupadas entre 1 y 3 segundos de tiempo de permanencia mínimo en una clase acústica. Esto tiene sentido si tenemos en cuenta que el tiempo excluido en las fronteras entre clases a la hora de la evaluación es de 2 segundos (1 segundo a izquierda y derecha de la frontera), por lo que se sitúa en el mismo orden de magnitud. De hecho, las mejores configuraciones evaluadas tienen un tiempo mínimo de permanencia de 2,25 segundos y 3 segundos, para el modelo sin derivadas y con derivadas respectivamente. Se puede comprobar también, tal y como se ha comentado anteriormente, que los resultados obtenidos al incluir la información de la primera y segunda derivadas son ligeramente mejores.

A modo de resumen de esta serie de experimentos, en la Tabla 5.8 se recoge el error promedio obtenido en las dos mejores configuraciones evaluadas aplicando los modelos HMM. La configuración que incorpora la primera y segunda derivada es la que consigue el mejor resultado obtenido durante este TFM alcanzando un error promedio del 23,21 %, seguida por debajo del modelo HMM sin derivadas que alcanza un error promedio del 23,44 %. Cabe destacar también un aumento de la varianza del error respecto de la salida de la red neuronal debido a una fuerte dependencia en el error obtenido a la salida de la red neuronal para que el modelo HMM mejore las prestaciones del sistema.

Configuración	Parámetros	Error promedio ± σ, N=5 (%)				
		Total	**mu**	**sp**	**sp+mu**	**sp+no**
HMM	$L = 75$, 3 *tied states*	23,44±1,34	15,41	27,61	21,55	29,20
HMM + 1a y 2a derivada	$L = 75$, 4 *tied states*	**23,21±1,56**	**16,06**	**27,53**	**20,34**	**28,91**

Tabla 5.8: Error tras la resegmentación en la partición de *test* promediado sobre 5 entrenamientos diferentes para las dos mejores configuraciones de parámetros

Con esta serie de experimentos se ha podido comprobar la mejora de prestaciones que puede suponer aplicar un modelo HMM a la salida de la red neuronal para imponer ciertas condiciones de inercia sobre las etiquetas, proporcionando una mejora relativa del 13,30 %, obteniendo así el mejor resultado de este TFM.

5.6. Discusión de resultados

Para poner fin a la evaluación experimental del sistema de segmentación se pretende realizar un análisis de los resultados obtenidos. Para ello nos centraremos en el mejor resultado obtenido en este TFM, evaluado mediante la aplicación del bloque de resegmentación incorporando la información de las derivadas, para realizar una serie de discusiones sobre sus prestaciones.

Aunque durante el desarrollo de esta memoria se hayan comparado las prestaciones del error en términos de la métrica utilizada en la evaluación Albayzín 2010, y expuesta en la ecuación 3.1, adicionalmente se introdujo la métrica SER propuesta en las evaluaciones NIST. Esta métrica se computa como el ratio entre la duración total del audio etiquetado de forma incorrecta y la duración total del audio a evaluar. La principal diferencia con la métrica de la evaluación Albayzín es que esta nueva métrica si que tiene en cuenta la proporción relativa de cada clase respecto del total de los datos. La configuración de HMM con primera y segunda derivada que se muestra en la Tabla 5.8 obtiene una métrica NIST de **14,48 %±1,10 %**. A la vista de este resultado, en términos generales, se puede decir que el sistema de segmentación es capaz de clasificar correctamente 8,5 segundos de audio de cada 10 segundos procesados.

De forma similar a como se ha hecho en experimentos previos, con el objetivo de cuantificar cuales son los tipos de errores más habituales en la clasificación se presenta en la Figura 5.17 la matriz de confusión de la mejor configuración obtenida a la salida del resegmentador. Vemos que el término de error más significativo se obtiene para las tramas clasificadas como "Voz+Ruido" pero que realmente pertenecen a la clase "Voz", con un 12 % de las tramas de la clase "Voz". De la misma forma ocurre en el caso contrario ya que el 11 % de las tramas de la clase "Voz+Ruido" son clasificadas erróneamente como "Voz". Cabe destacar que la clase "Voz+Ruido" es quizás la que tiene una definición más amplia ya que engloba toda la voz que no este grabada en condiciones de estudio, e incluso el solape de dos o más locutores, por lo que resulta razonable que el sistema pueda llegar a confundir estas dos clases. Además, se puede ver que el 10 % de las tramas pertenecientes a la clase "Voz+Música" son clasificadas erróneamente como "Voz+Ruido", algo que no pasa en el sentido contrario. La clase "Voz+Ruido" representa un 40 % del total de la base de datos, frente a la clase "Voz+Música" que representa un 15 %, por lo que puede resultar factible que el modelo tienda a clasificar con mayor probabilidad un segmento como "Voz+Ruido" ya que su probabilidad a priori es mayor. Resulta interesante comprobar también que la clase "Música", a pesar de ser una de las menos

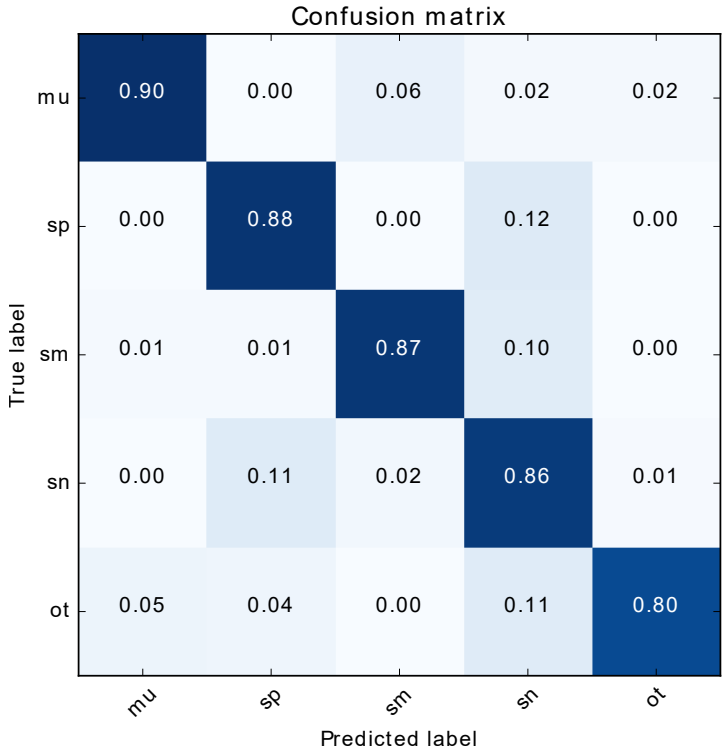


Figura 5.17: Matriz de confusión a la salida del resegmentador para la mejor configuración de parámetros evaluada (con derivadas, $L = 75$ y 4 subestados)

representadas en la base de datos (5 % del total), es la que mejores resultados obtiene, cediendo únicamente un 10 % de sus tramas a otras clases. Un 6 % de este error viene provocado por la confusión con la clase "Voz+Música", lo que es razonable ya que ambas estarán correladas al contener música. Esto viene a demostrar la capacidad de las características *chroma* para capturar las estructuras musicales ayudando al sistema a discriminar correctamente que es música y que no. Por último, la clase "Otros" es la que peor prestaciones obtiene con solo un 80 % de sus tramas clasificadas correctamente. Esta clase constituye únicamente el 3 % de los datos disponible. Aunque es similar al porcentaje de datos que presenta la clase "Música" (5 % del total de la base de datos), en este caso puede resultar más complicado de distinguir ya que engloba cualquier señal que se encuentre fuera de las otras cuatro clases, resultando en una definición poco específica.

El sistema de segmentación desarrollado en este TFM es un clasificador multiclase de 5 clases acústicas. Si agrupamos la información de varias de estas clases se puede generar un sistema de segmentación binario que nos de una clasificación con un menor nivel de detalle de la señal. Por un lado, podemos agrupar las clases que contienen voz ("Voz+Ruido", "Voz+Música" y "Voz") frente a las que no ("Música","Otros") para obtener un sistema que sea capaz de detectar aquellos fragmentos que contengan voz. En este caso nuestro sistema obtiene una precisión del 90,16 %, por lo que el sistema de segmentación implementado en este TFM podría llegarse a utilizar en el ámbito de la detección de actividad vocal con buenas prestaciones. De forma similar, si agrupamos aquellas clases que contienen música ("Música" y "Voz+Música") frente a las que no ("Voz", "Voz+Ruido" y "Otros") para obtener un sistema capaz de detectar música, se obtiene una precisión del 98,46 %, por lo que también se podría obtener un sistema con buenas prestaciones en esta tarea. En la Tabla 5.9 se muestran las prestaciones de estos sistemas que agrupan las clases acústicas en aquellas que contienen voz y aquellas que

contienen música en términos de error de falsa alarma y error de pérdida. En el caso de una

Sistema	**Error falsa alarma**	**Error pérdida**
Detector voz	9,84 %	0,82 %
Detector música	1,52 %	9,82 %

Tabla 5.9: Error de falsa alarma y error de pérdida para los sistemas de detección de voz y música derivados del sistema de segmentación automático multiclase

clasificación binaria, el error de falsa alarma se calcula como el ratio entre el número de falsos positivos y el número total de casos negativos, es decir, si lo aplicamos a nuestro problema sería el porcentaje de veces que una trama es erróneamente clasificada como voz cuando realmente no contiene voz. Por otro lado, el error de pérdida se calcula como el ratio entre el número de falsos negativos y el número total de casos positivos, esto aplicado a nuestro problema representa el porcentaje de tramas que no son clasificadas como voz cuando realmente si son voz . Teniendo esto en cuenta, podemos decir que nuestro sistema solo perdería 0,08 segundos de voz de cada 10 segundos procesados e introduciría 0,9 segundos de audio etiquetado erróneamente como voz por cada 10 segundos procesados. Respecto a la música, perdería unos 0,9 segundos de música por cada 10 segundos procesados, e introduciría en torno a 0,1 segundos de audio etiquetado incorrectamente como música por cada 10 segundos procesados.

Capítulo 6

Conclusiones y lineas futuras de trabajo

Este capítulo presenta una breve descripción y una valoración crítica del conjunto del trabajo realizado, además de proponer posibles lineas futuras.

6.1. Conclusiones

La generación masiva de contenidos audiovisuales ha hecho que la indexación y obtención de información de este tipo de contenidos sea cada vez más relevante. Un sistema como el propuesto en este TFM facilitaría la tarea de transcripción e indexación, convirtiéndola en un proceso automático. Tomando las redes neuronales como núcleo del sistema, y contextualizado dentro de la evaluación Albayzín-2010, el objetivo principal de este TFM es el desarrollo y evaluación de diferentes arquitecturas neuronales que permitan obtener una segmentación precisa de una señal de audio en aquellos fragmentos que contengan voz, música, ruido o una combinación de estos.

Para llevar a cabo este objetivo se partió de un sistema inicial basado en capas BLSTM cuyos parámetros de funcionamiento (Número de neuronas y número de capas BLSTM) se optimizaron mediante una primera serie de experimentos, obteniendo así un error promedio del 29,57 %. A continuación, se experimentó con los vectores de características de entrada al sistema, aumentando la resolución frecuencial en primer lugar, e incorporando las características *chroma* para reducir la confusión en las clases que contienen música. Con estas dos estrategias se logró reducir el error promedio hasta un 27,31 %. Seguidamente, dada la alta correlación temporal entre instantes cercanos en señales de audio, se introdujeron diferentes estrategias para reforzar el contexto a corto plazo en la clasificación. Mediante apilado temporal o el uso de capas convolucionales, se consiguió reducir aún más el error de segmentación, llegando a valores entre el 26,61 % y el 26,77 %. Finalmente se evaluó la incorporación de un bloque de resegmentación basado en HMMs para imponer cierto nivel de inercia sobre las etiquetas de la red neuronal. A través de un modelo basado en subestados atados se obtuvo un mejora relativa del 13,30 % respecto de la mejor estrategia evaluada con la red neuronal, alcanzando el mejor resultado de este TFM: un error promedio del 23,21 %.

En la Tabla 6.1 se compara este resultado con el mejor resultado obtenido en la literatura con esta base de datos. Los resultados obtenidos en términos promedio son muy similares, llegando a obtener un error total ligeramente inferior tanto en la métrica de la evaluación como en la métrica NIST. Sin embargo, si tenemos en cuenta el mejor de los 5 resultados evaluados para esta configuración se llega a reducir el error de forma significativa por debajo

de los mejores resultados de la literatura, obteniendo una mejora relativa del 11,16 % en la métrica de la evaluación, y una mejora del 11,44 % en la métrica NIST.

Sistema	**Descripción**	**Error (%)**					
		Total	**mu**	**sp**	**sp+mu**	**sp+no**	**NIST**
Mejor resultado literatura [23]	FA compensado por clase + HMMs	23,80	18,80	23,70	23,60	29,10	14,70
Mejor resultado TFM	RNN+ HMMs	**21,41**	**13,69**	**24,65**	**20,28**	**26,44**	**13,19**
Mejor resultado promedio TFM	RNN+ HMMs	23,21	16,06	27,53	20,34	28,91	14,48

Tabla 6.1: Comparativa de error para la métrica de la evaluación y la métrica NIST del mejor resultado de este TFM y el mejor resultado de la literatura

Ha quedado probada la mejora de prestaciones que supone el uso de las características *chroma* a la hora de reconocer aquellas clases que contienen música, resultando a su vez en una mejora de las prestaciones globales. También se ha demostrado el beneficio que implica el uso de técnicas de refuerzo del contexto a corto plazo en la clasificación. Por último se ha probado que el uso combinado de las redes neuronales con modelos estadísticos generativos como los HMMs permite obtener mejoras muy significativas en el sistema de segmentación.

Por lo tanto, **el objetivo que se pretendía alcanzar con la realización de este Trabajo Fin de Máster ha sido plenamente satisfecho**. Se ha conseguido obtener un sistema de segmentación automático con unos resultados competitivos, demostrando la capacidad conjunta de las redes neuronales y los modelos HMM en esta tarea.

6.2. Lineas futuras de trabajo

A la vista de los resultados del sistema de segmentación expuestos en esta memoria, cabe la posibilidad de aprovechar parte del trabajo realizado en este TFM como punto de partida en futuros proyectos siguiendo nuevas líneas de actuación.

Se propone inicialmente el estudio de arquitecturas alternativas que usen una combinación de clasificadores binarios en lugar de clasificadores multiclase. Por ejemplo, 3 clasificadores binarios: uno especializado en distinguir voz, otro especializado en distinguir música y otro especializado en distinguir ruido. La información de estos tres se combinaría a posteriori para generar el etiquetado. Esta arquitectura simplificaría la tarea de clasificación de la red neuronal reduciéndola a una decisión binaria que trabaja con clases disjuntas.

A pesar de que las características basadas en los bancos de filtros Mel y sus variantes se han convertido en las más populares para su uso en redes neuronales. Este tipo de procesados basados en medidas perceptuales pueden no ser la representación frecuencial más apropiada

para según qué tipo de tarea. Siguiendo el ejemplo de varios trabajos en este ámbito [51] [52] se propone utilizar varias capas convolucionales a modo de extractores de características entrenables que sean capaces de adaptarse a la tarea de segmentación. Con esta modificación se podría usar como entrada la transformada de Fourier de la señal de audio o incluso, tal y como propone la literatura, la forma de onda tal cual.

La secuencialidad que introduce el uso de capas BLSTM en el modelo acaba siendo su cuello de botella en términos de tiempo de computo. Eliminar por completo las capas BLSTM de la arquitectura neuronal permitiría un mayor grado de paralelismo, a la vez que reduciría en gran medida la carga computacional. Varios trabajos de investigación han demostrado ya la eficacia de las redes convolucionales en el modelado de secuencias temporales a través de lo que se conoce como *Temporal Convolutional Networks* [53] por lo que esta podría ser una opción interesante para sustituir a las capas BLSTM.

Como último punto a mejorar respecto de la red neuronal, se podría aumentar la cantidad de datos de entrenamiento disponibles utilizando técnicas de *data augmentation* aplicadas al ámbito de las señales de audio [54]. Además, existe la posibilidad de evaluar el sistema de segmentación desarrollado en otras bases de datos diferentes para comprobar la capacidad de generalización del modelo.

En cuanto al bloque de resegmentación, se plantea estimar las distribuciones de probabilidad del modelo HMM utilizando el etiquetado de referencia de la partición de *train* de la base de datos. Se plantea también la posibilidad de combinar la información de esta nueva propuesta con la que se ha venido utilizando en este TFM para adaptar las medias de cada distribución.

Bibliografía

[1] Youtube prensa, youtube en cifras. `https://www.youtube.com/yt/about/press/`, 2018 (Acceso: 2 Mayo 2018). 1

[2] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011. 1

[3] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*, 2017. 1

[4] Partha Pratim Mohanta, Sanjoy Kumar Saha, and Bhabatosh Chanda. A model-based shot boundary detection technique using frame transition parameters. *IEEE Transactions on multimedia*, 14(1):223–233, 2012. 1

[5] Douglas A Reynolds. An overview of automatic speaker recognition technology. In *Acoustics, speech, and signal processing (ICASSP), 2002 IEEE international conference on*, volume 4, pages IV–4072. IEEE, 2002. 1

[6] Richard J Moran, Richard B Reilly, Philip de Chazal, and Peter D Lacy. Telephony-based voice pathology assessment using automated speech analysis. *IEEE Transactions on Biomedical Engineering*, 53(3):468–477, 2006. 1

[7] Vivolab, sitio web: Pagina principal. `http://vivolab.unizar.es/`, 2015 (Acceso: 20 Abril 2018). 2

[8] Martín Abadi, Ashish Agarwal, Paul Barham, and Eugene Brevdo et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. Software available from `http://tensorflow.org`. 2

[9] Adam Paszke, Gregory Chanan, Zeming Lin, Sam Gross, Edward Yang, Luca Antiga, and Zachary Devito. Automatic differentiation in PyTorch. *Advances in Neural Information Processing Systems 30*, pages 1–4, 2017. 2, 25

[10] The Theano Development Team et al. *Theano: A Python framework for fast computation of mathematical expressions*. pages 1–19, 2016. `http://arxiv.org/abs/1605.02688`. 2

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. *Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093*, 2014. 2

[12] Diego Álvarez Gutiérrez and Ignacio Viñals Bailo. *Detector de actividad vocal para Diarización mediante redes neuronales en entornos Broadcast*. 2017. Trabajo Fin de grado, accesible en `https://deposita.unizar.es/record/32674?ln=es`. 3, 24, 27

[13] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Readings in speech recognition*, pages 65–74. Elsevier, 1990. 5, 6

[14] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990. 5

[15] Stanley Smith Stevens, John Volkmann, and Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937. 6

[16] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978. 9

[17] Scott Chen, Ponani Gopalakrishnan, et al. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. DARPA broadcast news transcription and understanding workshop*, volume 8, pages 127–132, 1998. 9

[18] Perrine Delacourt and Christian J Wellekens. Distbic: A speaker-based segmentation for audio data indexing. *Speech communication*, 32(1-2):111–126, 2000. 9

[19] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. 10

[20] Patrick Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal,(Report) CRIM-06/08-13*, 14:28–29, 2005. 10

[21] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000. 10

[22] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1435–1447, 2007. 11

[23] Diego Castán, Alfonso Ortega, Antonio Miguel, and Eduardo Lleida. Audio segmentation-by-classification approach based on factor analysis in broadcast news domain. *EURASIP Journal on Audio, Speech, and Music Processing*, 2014(1):34, 2014. 11, 22, 48

[24] Gaël Richard, Mathieu Ramona, and Slim Essid. Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–461, 2007. 11

[25] Yorgos Patsis and Werner Verhelst. A speech/music/silence/garbage/classifier for searching and indexing broadcast news material. In *Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on*, pages 585–589, 2008. 11

[26] Mingchun Liu, Chunru Wan, and Lipo Wang. Content-based audio classification and retrieval using a fuzzy logic system: towards multimedia search engines. *Soft Computing*, 6(5):357–364, 2002. 11

[27] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 11

[28] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. 11

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 11

[30] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 13

[31] Mc Mozer. A focused back-propagation algorithm for temporal pattern recognition. *Complex systems*, 3(4):349–381, 1989. 13

[32] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 13

[33] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 13

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 15

[35] Lalit R Bahl, Peter F Brown, Peter V de Souza, and Robert L Mercer. Speech recognition with continuous-parameter hidden markov models. In *Readings in Speech Recognition*, pages 332–339. Elsevier, 1990. 18

[36] Taras Butko and Climent Nadeu. Audio segmentation of broadcast news in the Albayzin-2010 evaluation: overview, results, and discussion. *EURASIP Journal on Audio, Speech, and Music Processing*, 2011(1):1, 2011. 19

[37] Red temática de tecnologías del habla, página principal. `www.rthabla.es`, 2014 (Acceso: 20 Mayo 2018). 19

[38] NIST. The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan, (Melbourne 28-29 May 2009). 22

[39] Ascensión Gallardo Antolín and Rubén San Segundo Hernández. UPM-UC3M system for music and speech segmentation. 2010. 22

[40] Laura Docio-Fernandez, Paula Lopez-Otero, and Carmen Garcia-Mateo. The UVigo-GTM speaker diarization system for the Albayzin'10 evaluation. *Proc. FALA*, 2010. 22

[41] Javier Franco-Pedroso, Ignacio Lopez-Moreno, Doroteo T Toledano, and Joaquin Gonzalez-Rodriguez. ATV-UAM system description for the audio segmentation and speaker diarization Albayzin 2010 evaluation. In *FALA VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop*, pages 415–418. Citeseer, 2010. 22

[42] Diego Castán, Alfonso Ortega, Carlos Vaquero, Antonio Miguel, and Eduardo Lleida. VIVOLAB-UZ audio segmentation system for Albayzin evaluation 2010. In *FALA VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop*, pages 437–440. Citeseer, 2010. 22

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 25

[44] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 25

[45] Roger N Shepard. Circularity in judgments of relative pitch. *The Journal of the Acoustical Society of America*, 36(12):2346–2353, 1964. 31

[46] Taemin Cho and Juan P Bello. On the relative importance of individual components of chord recognition systems. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(2):477–492, 2014. 32

[47] Sebastian Ewert, Meinard Muller, and Peter Grosche. High resolution audio synchronization using chroma onset features. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1869–1872. IEEE, 2009. 32

[48] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent developments in openSMILE, the Munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838. ACM, 2013. 32

[49] Steve J Young. The general use of tying in phoneme-based HMM speech recognisers. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 569–572. IEEE, 1992. 40

[50] Fredrik Gustafsson. Determining the initial states in forward-backward filtering. *IEEE Transactions on Signal Processing*, 44(4):988–992, 1996. 40

[51] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. 49

[52] Dimitri Palaz, Ronan Collobert, and Mathew Magimai Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. *arXiv preprint arXiv:1304.1018*, 2013. 49

[53] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 49

[54] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017. 49