UNIVERSIDAD DE ALCALÁ

MASTER EN DEEP LEARNING
2018/19

EXTRACCIÓN DE RITMOS DE AUDIOS

Francisco J Durá Galiana

MASTER EN DEEP LEARNING
2018/19

EXTRACCIÓN DE RITMOS DE AUDIOS

Francisco J Durá Galiana
franciscojosedg@gmail.com

Recientemente, y gracias a la explosión que están teniendo las redes profundas, se están utilizando redes neuronales para generar y clasificar música automáticamente. Muchas de las arquitecturas propuestas producen resultados interesantes, pero todavía hay mucho camino para que una máquina componga canciones al completo por sí sola. Un campo que tiene poca investigación desde la perspectiva de las redes neuronales es la exploración de los ritmos de la música, campo que se considera esencial. Por lo tanto, en este trabajo se presenta una arquitectura convolucional para predecir género y tempo como paso inicial a la predicción de ritmos completos.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Planteamiento del problema y revisión de la literatura

1.1. Planteamiento del problema

La rítmica es una de las partes más importantes de la música. Una misma pieza puede aportar una sensación completamente distinta a un oyente simplemente cambiando el tempo o el ritmo de ésta Elowsson (2018). Hasta el resurgimiento de los modelos de aprendizaje profundo los modelos de extracción de información musical (MIR por sus siglas inglesas) se han basado en transformaciones de las señales de audio; estas transformaciones se basan en procesos que requieren tener una gran longitud del audio o pre-procesamiento y creación manual de características para su modelado. Añadido a la complejidad del problema, el diseño de los procesos de extracción de características requiere un amplio conocimiento musical J Humphrey et al. (2012). El problema surge cuando se quiere inferir información de un audio con una longitud más corta (3 segundos por ejemplo), ya sea por disponibilidad o porque se quiera obtener la información más rápidamente sin necesidad de ser experto en características musicales.

En este trabajo se propone el siguiente problema: Predicción de ritmos dada una corta grabación para ayudar en la elección de una base rítmica. Actualmente lo que haría un músico (o productor) para solucionar esto es probar con varios ritmos utilizando conocimiento previo hasta que uno encaje, esto además requiere que se genere la base

rítmica. Sin embargo, un sistema automático podría rápidamente detectar un tempo y predecir un ritmo adecuado. Dados los requisitos del problema y la disponibilidad de recursos se simplificará el problema a una clasificación de género y predicción de tempo, ya que estas dos características son principales determinantes del ritmo de una canción.

En el siguiente apartado se va a presentar una revisión literaria del estado de la clasificación musical a través de aprendizaje profundo.

1.2. Revisión literaria

Los primeros trabajos de redes neuronales con música aparecieron en 1988 y generaron una época de composición musical algorítmica que duró hasta 2009 sobreviviendo el invierno de la inteligencia artificial (Pons 2018). Una de las propuestas que inició el movimiento fue una red profunda (dos capas ocultas y una capa de salida) en la que los autores defienden que este tipo de redes son capaces de generar resultados artísticos originales a partir de suficientes datos (Lewis 1988). Por otro lado Todd (1988) propuso una red recurrente (RNN) para generar música de manera secuencial. Esta tendencia a usar redes secuenciales es la que ha isnpirado el movimiento (Pons 2018) y los resultados han ido mejorado al hacer uso de células LSTM (Eck & Schmidhuber 2002).

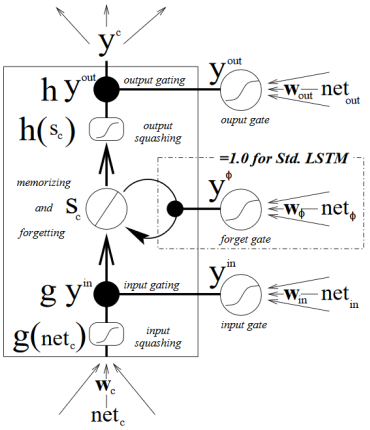

Figura 1.1: Diseño de una neurona LSTM de Gers et al. (2000), Gers & Schmidhuber (2001).

Eck & Schmidhuber (2002) comentan que, aunque las redes recurrentes (RNN) se hayan utilizado para generar música, éstas tienen un defecto y éste es que la música generada por RNN suele tener falta de coherencia global como la que suele estar presente en música real. Ésto se debe a que las redes neuronales recurrentes (RNN) suelen tener problemas de gradientes desvanecedoras. Una manera de hacer que una red temporal recuerde eventos pasados sin problemas de gradiente es utilizando neuronas LSTM (Figura 1.1). Eck & Schmidhuber (2002) proponen una red LSTM para generar música automáticamente, específicamente componen un Blues de 12 compases. En sus estudios descubren que una red secuencial basada en neuronas LSTM no se desvía de la estructura principal de la pieza y consigue componer blues con buen tiempo y estructura.

Recientemente ha habido un cambio importante en el campo de MIR con la introducción de los procesadores gráficos (GPU) y las redes convolucionales (CNN). En 2009 se presentó una investigación que muestra que las representaciones de características creadas por redes convolucionales sobrepasaba la efectividad de características usadas hasta el momento en procesamiento de audio (Lee et al. 2009). En su investigación, Lee et al. (2009) aplican estos resultados para el diseño de redes profundas *Deep Belief* (CDBN) y presentan resultados prometedores en el campo de clasificación musical (para etiquetado de género y artista).

Mientras que la extracción de información musical clásica requiere la construcción de características a partir de señales de audio o creación de meta-datos, el procesamiento de música con redes profundas se puede realizar con muchos tipos de datos. Entre ellos se encuentran: representaciones textuales de música escrita, representaciones visuales de música escrita, audios puros, audios pre-procesados y representaciones visuales de audios. En el campo del aprendizaje profundo y de cara a información extraída de audios reales se han utilizado audios y representaciones visuales de audios principalmente. Las representaciones de audios utilizadas son los espectrogramas de coefficientes *Cepstral* (MFCC). Éstos son mapas de intensidad de frecuencia con respecto del tiempo, normalmente representándose el tiempo en el eje 'x', la frecuencia en el eje 'y' y la intensidad en el eje 'z' (generalmente representada en color). Éste tipo de representaciones son las más usadas por modelos CNN.

Para comparar un modelo clásico con uno profundo, Costa et al. (2017) presentan un etiquetador de género comparando el uso de espectrogramas (con y sin cálculo

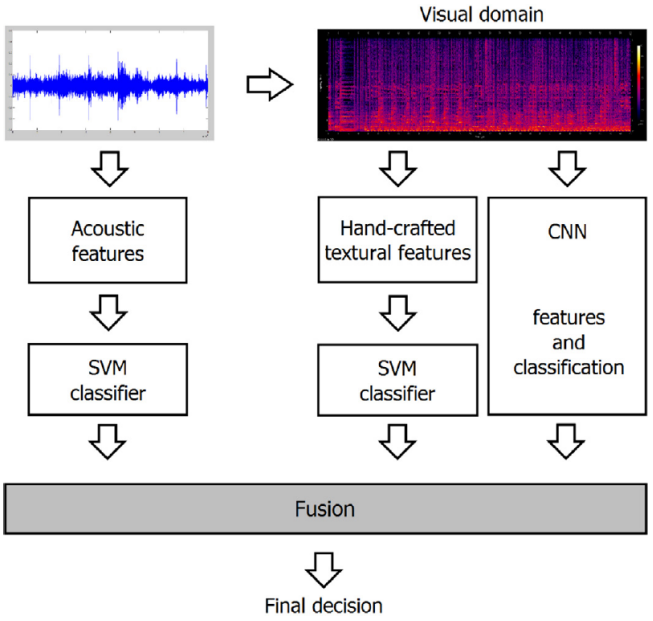

Figura 1.2: Modelo de clasificación utilizando ambos audio y representación de espectrograma por Costa et al. (2017).

manual de características) y audio sin procesar. Además proponen un modelo de conjunto que utilice los tres métodos para mejorar la precisión (Figura 1.2). Para sus predicciones utilizan el set de datos ISMIR (http://www.ismir.net/resources/datasets/), el cual será útil más adelante cuando se necesiten datos de entrenamiento.

En su investigación descubren que el clasificador entrenado en representaciones visuales (espectrogramas) tiene un mejor desempeño que clasificadores entrenados con representaciones sonoras y características manufacturadas.

Estos resultados son parcialmente afines a los resultados por Pons et al. (2017a) y Lee et al. (2009). Sin embargo, Pons et al. (2017a) concluyen que la diferencia entre los modelos basados en audio y los modelos basados en espectrogramas se diferencian principalmente en la cantidad de datos y la cantidad de pre-procesamiento de estos. Cuando hay pocos datos disponibles (100k canciones de entrenamiento) es necesario el pre-procesamiento de los datos para un buen funcionamiento de los modelos y en este caso las redes convolucionales destacan. Usando conocimientos musicales uno puede diseñar los filtros de las convoluciones para extraer la información deseada por lo tanto haciendo un modelo más específico y eficiente para la tarea planteada. Por otro lado,

cuando la cantidad de datos más grande (1M canciones de entrenamiento) un modelo basado en audio puro supera los modelos basados en espectrogramas.

Además de su investigación sobre la eficiencia de las redes convolucionales, Pons et al. (2017a) realizan una revisión de los modelos actuales y proponen una arquitectura típica para modelos de extracción de información musical (Figura 1.3).

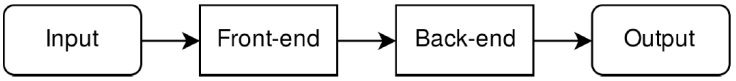

Figura 1.3: Arquitectura general de un modelo de extracción de información musical definida por Pons et al. (2017a).

Pons et al. (2017a) identifican que la arquitectura principal que se está utilizando para clasificación de audio está principalmente dividida en "Input", "Front-end", "Back-end" y "Output"(Figura 1.3) y estas se constituyen de lo siguiente:

- **Input:** estructura de datos que acepte el modelo (audio, espectrograma, metadatos).
- **Front-end:** extracción de características, generalmente a través de capas convolucionales.
- **Back-end:** Procesamiento de características (capas densas) que tienen que poder admitir entradas de tamaño variable en ciertas ocasiones. Lo último se puede conseguir mediante capas de pooling o capas recurrentes, por ejemplo LSTMs (neuronas de Long-Short Term Memory)
- **Output:** El output es la clasificación o etiquetado del modelo.

Retomando el tema de la preferencia del MIR por las CNN: recientemente se ha estado identificando que la razón principal por la que las redes convolucionales fallan al clasificar imágenes es porque éstas ven principalmente texturas donde el ser humano ve formas. Sin embargo, esta preferencia por las CNN de ver texturas en imágenes ha sido aprovechada en el campo de extracción de información musical para determinar el tipo de sonido (o tipo de instrumentos) en una pieza. Han et al. (2016) presentan una red convolucional (Figura 1.4) para clasificar el instrumento predominante en un audio.

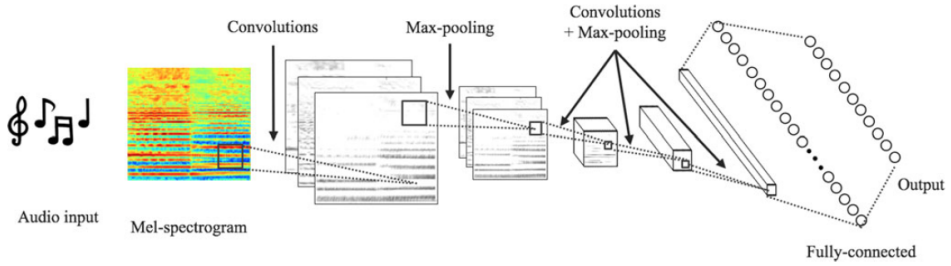

Figura 1.4: Red Convolucional para clasificación de instrumento por Han et al. (2016).

La clasificación de instrumentos con espectrogramas es posible ya que, para una misma nota, distintos instrumentos tienen distinto timbre. Ésto significa que las frecuencias resonantes dominantes serán distintas para cada instrumento y esto se verá como un patrón dentro del espectrograma; la red convolucional es capaz de extraer estas texturas o patrones gracias a los filtros aplicados. Han et al. (2016) defienden que sus experimentos consiguen una mejora del $\approx 20\,\%$ sobre algoritmos del estado del arte al comparar métricas F1 en este tipo de clasificadores. También concluyen que para el diseño de las capas convolucionales es necesario un conocimiento musical para saber que características se están extrayendo. Han et al. (2016) también encuentran que, al igual que en redes profundas, la primera capa de convoluciones está extrayendo características simples (en este caso líneas verticales y horizontales) y el resto de capas se activan con patrones más complejos.

Pons et al. (2017b) también investigan el uso de la extracción del timbre en espectrogramas con el fin de resolver tareas de clasificación/etiquetado de audio musical y de audio general. En su artículo investigan como el diseño de los filtros y capas convolucionales ayudan a diseñar una red para la tarea en mente. A consecuencia de esto proponen unas pautas para diseñar redes CNN para clasificación de música. Por ejemplo proponen que filtros pequeños son buenos para la eficiencia en general del modelo además de ser óptimos para activar frecuencias de bajo o bombo de la batería. También denotan que para capturar sonidos de percusión es necesario detectar formas llanas (horizontales) en las representaciones espectrales. ésta información es de gran utilidad para poder diseñar una red CNN que capture ritmos.

Invariabilidad	**Descripción**
Tempo	Invariable con respecto al tempo
Fase	Invariable con respecto a la fase del ritmo
Ritmo	Invariable con respecto al tempo y a la fase
Pulso	Invariable con respecto a un pulso predeterminado
"Cepstroid"	Invariable con respecto a la repetición dominante de la canción
Tiempo	Invariable con respecto al tiempo
Tono	Invariable con respecto a la frecuencia

Tabla 1.1: Invariabilidades a tener en cuenta para el procesamiento de ritmo en música, de Elowsson (2018)

Por otro lado, para la correcta extracción de ritmos de un audio, un sistema de extracción musical debe acertar el ritmo independientemente del tempo de éste, esto significa que no solo es necesario captar percusiones en el espectrograma si no que es necesario que infiera el tempo o contexto temporal. Elowsson (2018) presenta la idea de que las redes convolucionales pueden detectar distintos ritmos de un audio independientemente del tempo de dicho audio.

En su análisis, Elowsson (2018) propone que hay distintos tipos de variabilidades con respecto al tiempo en la música y un sistema de extracción de información de música debe tener una serie de invariabilidades con respecto a las características de una pieza. Los tipos de invariabilidades posibles se muestran en la tabla 1.1.

El ritmo musical y el tempo (o métrica) son especialmente importantes porque ofrecen un contexto en el que el oyente puede interpretar la estructura y melodías de una pieza. Elowsson (2018) pone como ejemplo una nota que ocurre a los 0.125 segundos del comienzo de un compás musical. En una pieza que tenga de tempo 120 pulsaciones por minuto esta nota estaría a una distancia de una semicorchea (1/16 pulsos) del inicio, en una pieza de 160 pulsaciones por minuto la nota estaría a una distancia de una corchea con puntillo (3/16 pulsos) del inicio.

Por otro lado, la estructura del ritmo en la música es importante y suele estar muy definida globalmente en una pieza. por ejemplo: la música rock popular suele tener $\frac{4}{4}$ (4 pulsos por compás), pero un vals tiene una métrica de $\frac{3}{4}$ (3 pulsos por compás). Esto significa que cambios melódicos importantes suelen ocurrir cada cuatro pulsos en el rock y cada 3 en un vals (Eck & Schmidhuber 2002)

Basado en esto la literatura muestra que las redes convolucionales han sido elegidas para el procesamiento de información musical gracias a sus propiedades de Invariabilidad-tiempo e invariabilidad-tono. Esto es gracias al uso de convoluciones idénticas en distintas zonas de una representación gráfica/linear de un clip de audio: Eventos repetidos temporalmente pueden ser captados por convoluciones a lo largo de la dimensión del tiempo, mientras que convoluciones con respecto a la frecuencia puede captar ritmos en distintas frecuencias o tonos (Elowsson 2018).

Elowsson (2018) encuentra que cambiando de los tipos de filtros usados en la última capa antes de las capas densas de la red se puede optimizar la red para distintas tareas. Entre ellas distinguen: género musical, correlación de patrones, complejidad rítmica, claridad rítmica, métrica, detección de infracciones de copyright y *swing* (estilo de ritmo típico de Jazz y Blues). Sin embargo, las investigaciones de este artículo se realizan sobre audios simplificados de ritmo, por lo que solo son representativos del potencial de la técnica y más trabajo es necesario en éste area.

Las redes convolucionales han demostrado una gran eficiencia prediciendo el género musical de audios, pero hacer predicciones de tempo, aún teóricamente posible segun los resultados de la literatura, ha demostrado ser algo más complicado. Previamente esto se hacía analizando trozos de un audio y superponiendo audios para calcular relaciones temporales (Jehan 2005, Klapuri et al. 2006) o bien haciendo cálculos avanzados sobre series temporales (Sheng Gao & Chin-Hui Lee 2004).

Un ejemplo a mitad de camino dónde se mezcla la extracción clásica de características y los algoritmos de aprendizaje automático lo dan Lazaro et al. (2017). En su investigación, Lazaro et al. (2017) utilizan extracción clásica de características de audio y aplican un algoritmo de aprendizaje automático basado en máquinas de vectores soporte (SVM). Para la extracción de características, usan tres tipos de extracción de características clásicas: centroide del espectro de audio (ASC), llanura del espectro de audio (FSC) y dispersión del espectro de audio (ASS). Estas tres características se sacan haciendo transformaciones de la señal usando principalmente transformaciones de Fourier y logaritmos para extraer frecuencias dominantes de la señal. Posteriormente se pasan las señales alteradas a una máquina de vector soporte como se muestra en el esquema (Figura 1.5).

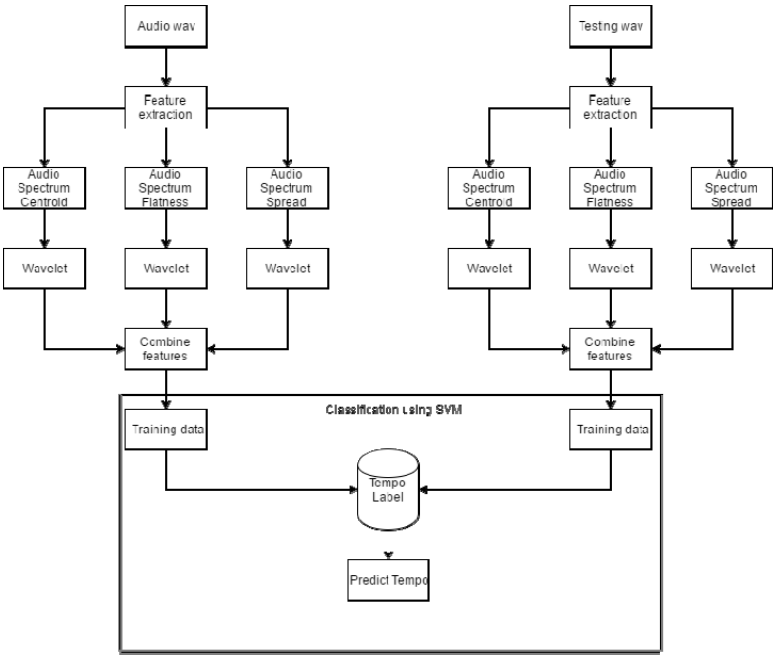

Figura 1.5: Arquitectura de un predictor de tempos por Lazaro et al. (2017).

Para las predicciones, Lazaro et al. transforman los tempos en 3 clases: lento, medio y rápido. Ésta es una simplificación significativa del problema, pero se debe de hacer debido a que el algoritmo usado no sería potente suficiente cómo para separar alrededor de 100 clases con los datos disponibles. Para separar las 3 clases utilizan una máquina de vector soporte (SVM) y consiguen una precisión del 80 %, éste es un buen resultado pero está todavía bastante alejado de conseguir una predicción del tempo actual precisa. Para realizar ésto necesitaríamos utilizar algoritmos de aprendizaje profundo.

J Humphrey et al. (2012) defienden que la creación de características es un proceso en cadena que podría ser simulado por una red profunda y encuentran paralelismos entre la forma que una red convolucional procesa información y la manera clásica de obtener características. Una gran ventaja sobre la extracción clásica que añaden a las ya discutidas es que al seleccionar características podríamos estar desechando otras importantes. Para evitar eso se necesitaría un conocimiento muy profundo del problema y dichas características serían muy especificas para el problema para el que se han diseñado.

Accuracy0	42,1 %
Accuracy1	69,3 %
Accuracy2	86,4 %

Tabla 1.2: Precisiones medias de la red profunda de Schreiber & Meinard (2018)

Aunque el problema de la predicción del tempo usando redes convolucionales necesita bastante trabajo, J Humphrey et al. (2012) concluyen es posible diseñar un modelo capaz de predecir tempo. No obstante, esto requiere una re-evaluación del problema, ya que éste se ha ido desarrollando para que tenga una solución algorítmica que no es fácilmente adaptable a una red profunda.

Recientemente, Schreiber & Meinard (2018) han demostrado que una implementación profunda directa al problema es posible y presentan una investigación donde utilizan una red profunda de capas convolucionales (CNN) para extraer tempo directamente del espectrograma de un audio sin procesar. La red tiene dos características destacables: los filtros que utilizan para procesar el audio son todos horizontales y utilizan módulos multi-filtro (Figura 1.6) en la parte central de la red. Los módulos multi-filtro están compuestos por convoluciones paralelas con diferentes características que son concatenadas posteriormente y pasadas por una convolución final a modo de unión de la capa. Todas las capas están activadas con la función Exponential Linear Unit (ELU).

Schreiber & Meinard (2018) prueban la red con varios sets de datos y definen tres tipos de precisión para evaluar los resultados: *Accuracy0* es la precisión de predecir el íntegro más cercano del tempo, *Accuracy1* es la precisión permitiendo una desviación del 4 % y *Accuracy2* es similar a la anterior pero también permitiendo errores por octava (los errores por octava es la predicción de tempos múltiplos del verdadero por 2 o 3). De media reportan los resultados mostrados en la Tabla 1.2.

Como se puede ver en la Tabla 1.2, la precisión estricta de tempo no es ideal, aunque aumentar una tolerancia del 4 % incrementa significativamente la precisión de la red. Los resultados también muestran que la predicción de tempos dobles o triples al actual es bastante común para este tipo de red. Schreiber & Meinard (2018) concluyen que los resultados podrían ser mejorados utilizando sets de datos más balanceados o bien mejorando la arquitectura de la red. Entre las propuestas sugieren filtros más cortos en las convoluciones y capas densas más grandes.

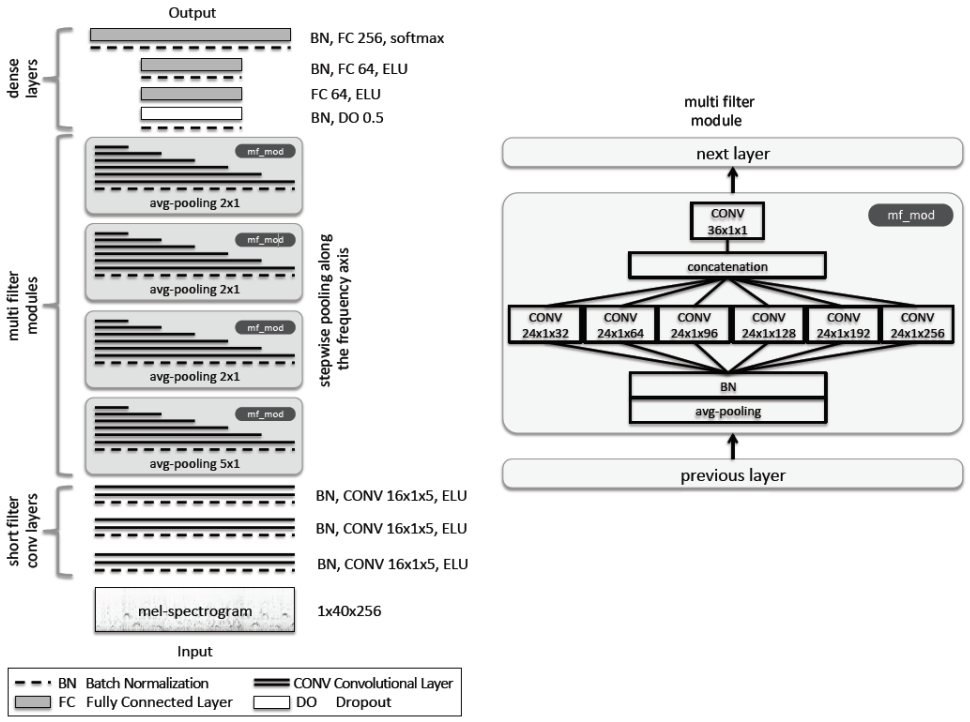

Figura 1.6: Arquitectura de un predictor de tempos con una red CNN profunda por Schreiber & Meinard (2018).

1.3. Propuesta de modelo

La revisión literaria del estado actual de la extracción de información musical (MIR) sugiere que es posible diseñar un modelo que sea capaz de predecir tanto género como tempo a partir de un audio convertido a espectrograma. Las redes convolucionales han sido demostradas como eficientes para el problema planteado, aunque las predicciones de tempo no son todavía completamente satisfactorias.

Usando la información recogida en esta sección se van a proponer unas cuantas arquitecturas de redes convolucionales para predecir género y tempo y se intentará mezclar estas arquitecturas en una sola red. La idea de estos experimentos es, demostrar la posibilidad de extraer ritmos de un audio con un modelo único utilizando dichas herramientas.

Capítulo 2

Descripción de la técnica y resultados

2.1. Introducción

Como ya se ha comentado, se va a estudiar y presentar el uso de arquitecturas convolucionales para la extracción de ritmos. Para ello se han desarrollado varias redes para extraer género y ritmo a partir de audios de música.

La arquitectura y el procesamiento de datos se han derivado inicialmente tomando como ejemplo el código de Guimares (2017), que presenta una CNN para clasificación de géneros utilizando la arquitectura VGG16.

A continuación se va a presentar los métodos de análisis y modelado desarrollados y utilizados para el problema presentado.

2.2. Datos

Los datos para el modelo deben de ser clips de audio musical, el cual es abundante hoy en día. Sin embargo, como con cualquier otro problema de aprendizaje profundo, lo difícil es encontrar datos etiquetados apropiadamente. Afortunadamente el estudio

Dataset	**Tempo**	**Género**	**nº de audios**	**Longitud media (s)**	**Longitud total**	**Formato**
ACM	Si	No	1410	38.6	15H 6m	.mp3
FMA	Si	Si	8000	30.0	66H 40m	.mp3
GTZAN	Si	Si	1000	30.0	8H 20m	.au
hainsworth	Si	Si	222	53.9	3H 20m	.wav
ISMIR2004	No	Si	729	197.9	40H 6m	.mp3

Tabla 2.1: Sets de datos encontrados

Dataset	**Tempo**	**Género**	**nº de audios**	**Longitud media (s)**	**Longitud total**	**Formato**
FMA	Si	Si	1294	30.0	10H 47m	.mp3
GTZAN	Si	Si	1000	30.0	8H 20m	.au
hainsworth	Si	Si	222	53.9	3H 20m	.wav

Tabla 2.2: Sets de datos disponibles para entrenamiento de la red

de la extracción de información musical (MIR) es una disciplina de bastante interés. En este caso, el interés es suficiente para existir una sociedad internacional con congresos anuales: la Sociedad Internacional para la Extracción de Información Musical (ISMIR). Gracias a esta organización existen varios sets de datos disponibles al público libremente.

2.2.1. Obtención y selección de los datos

De los sets de datos encontrados se han conseguido datos de 5 sets, su información resumida se puede ver en la Tabla 2.1.

Dado que la intención de este estudio es predecir género y tempo se descartan los sets que no tienen toda la información. Se podría utilizar estos datos incompletos para entrenar los pesos relevantes de las redes, pero nos vamos a centrar en datos completos en la fase de investigación. Al descartar los datos incompletos, se pierden los sets de datos ISMIR2004 y ACM además de gran parte de los datos del set FMA, resultando en los datos disponibles mostrados en la Tabla 2.2.

Una vez tenemos los datos, debemos validar la calidad de las etiquetas que tenemos. Por un lado, los tempos han sido obtenidos en dos tipos de formatos de las

fuentes encontradas: enteros y decimales. Para igualar los datos, los tempos se han convertido todos a números enteros. Al comparar los tres sets de datos podemos que tienen distribuciones de tempos similares (Figura 2.1) y con forma de distribución normal.

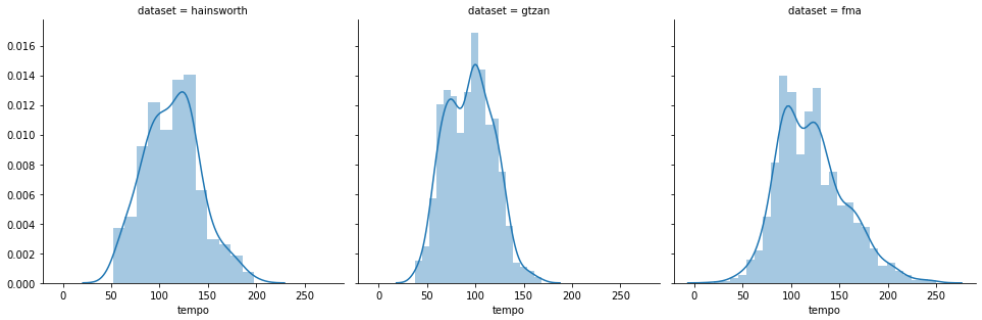

Figura 2.1: Distribución de tempos en los sets de datos encontrados.

Por otro lado, los géneros venían con distintos nombres o sub-etiquetas asociadas. Para simplificar el problema se han agrupado géneros similares y se han igualado nombres en todos los sets. Igualmente, el número de etiquetas no es el mismo en todos los sets, lo que hace tarea difícil unificarlos. Al comparar las distribuciones de géneros de cada fuente (Fig. 2.2), se ha comprobado que aún unificando etiquetas la calidad de éstas es muy distinta para cada set de datos.

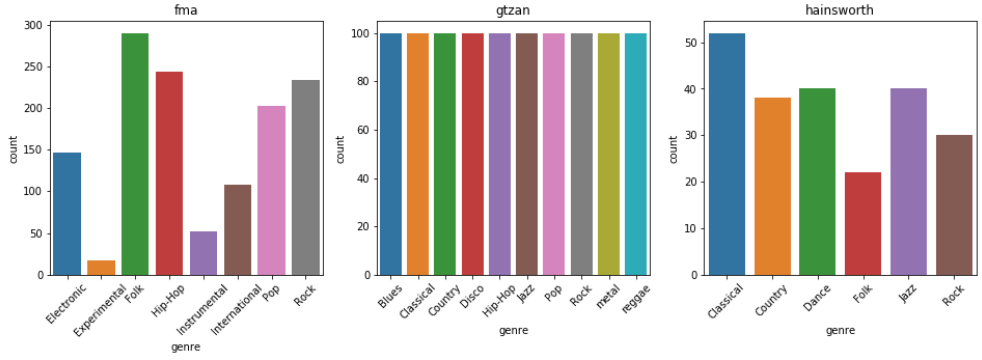

Figura 2.2: Distribución de género en los sets de datos encontrados.

Teniendo en cuenta número de casos y calidad de etiquetas se ha llegado a la conclusión de que el mejor set de datos es el set GTZAN y se utilizará éste exclusivamente en la investigación de la red y obtención de resultados.

2.2.2. Procesamiento de datos de entrada

Dado que vamos a utilizar convoluciones 2D en nuestra red debemos convertir nuestros audios a matrices (imágenes) de dos dimensiones. Como ya se ha comentado en la revisión literaria, el proceso más común es la conversión de audio a espectrograma; para esto utilizaremos la librería `librosa` de Python. Además, para igualar todas las entradas a la red se van a tomar cortes de 3 segundos de los audios (con un solape del 50 % entre ellos para no perder información en el corte).

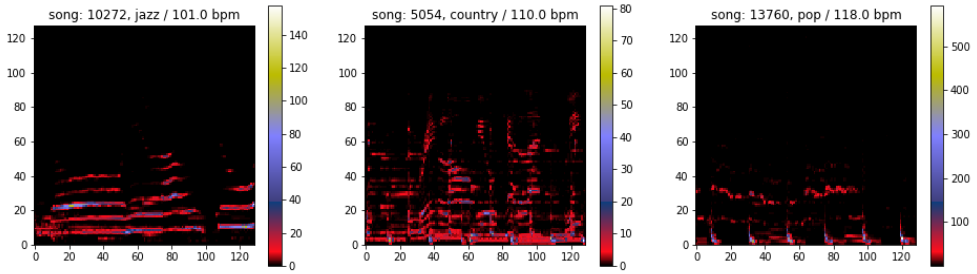

Figura 2.3: Espectrogramas de frecuencia.

Los resultados de la conversión a espectrograma de los audios se puede ver en la Figura 2.3. Aquí se muestran las frecuencias en el eje y, el tiempo del clip en el eje x y la densidad de la frecuencia en el eje z (representado con una escala de color). El uso de espectrogramas de éste modo es suficiente para entrenar la red, pero la representación de los datos en este momento es muy dispersa. Una forma de evitar esto es realizando una transformación de los datos usando las medias y desviaciones medias, pero como estamos tratando audios vamos a recurrir a una transformación logarítmica y representar los datos en decibelios (Figura 2.4).

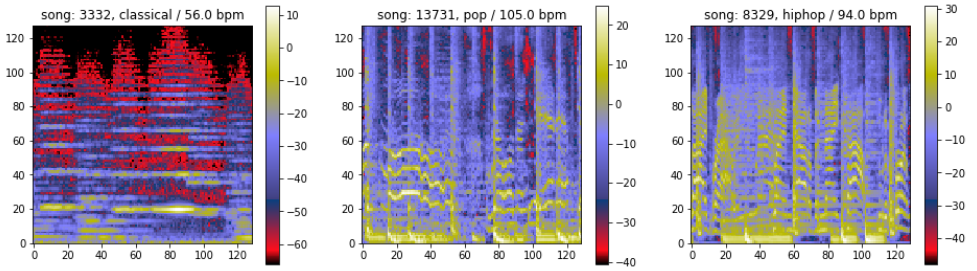

Figura 2.4: Espectrogramas de frecuencia en escala dB.

Género	**clase**	**Género**	**clase**
Metal	0	Country	5
Disco	1	Pop	6
Clásica	2	Blues	7
Hip-hop	3	Reggae	8
Jazz	4	Rock	9

Tabla 2.3: Asignación de clases de géneros

Como se puede ver en la Figura 2.4 las imágenes ahora parecen contener más información en todas las áreas de la imagen y, además, una representación en decibelios es más afín a lo que el oído humano interpreta y por lo tanto a como se han grabado estos audios. Se espera que esto ayude a los modelos de predicción, aunque se harán pruebas con los dos tipos de datos para validar ésto.

2.2.3. Procesamiento de datos de salida

En las arquitecturas que se van a proponer se van a predecir dos variables distintas y éstas requieren su trato individual.

El género se ha tratado como una variable cualitativa la cual tiene 10 clases, por lo que la variable se codificará en formato disperso o *one-hot*. Los índices del vector que se han asignado a cada género se pueden ver en la Tabla 2.3.

La variable tempo ha sido tratada de distintas maneras. Inicialmente se trató como una variable continua y se intentó predecir utilizando una activación lineal en la capa de salida pero esto no daba resultados muy prometedores. Como ya se ha visto en la Figura 2.1, los datos siguen una distribución normal y por este motivo no se esperaba una mejora haciendo una normalización de los datos. Sin embargo al aplicar una transformación de este estilo se consigue que la variable tenga valores más cercanos a 0 (o 1) y un rango menor. Por este motivo, se intentó hacer normalizaciones sobre el tempo (*min-max* y estándar), pero, finalmente, no se consiguió que mejoraran los resultados.

Llegado este punto se re-estimó el problema y se planteó la variable de tempo como una variable discreta o cualitativa, por lo que se separaron los tempos en clases categóricas. En concreto, se construyeron dos variables de tempo. En la primera variable

Tempos	**clase**	**Tempos**	**clase**
[38, 48]	0	(108, 118]	7
(48, 58]	1	(118, 128]	8
(58, 68]	2	(128, 138]	9
(68, 78]	3	(138, 148]	10
(78, 88]	4	(148, 158]	11
(88, 98]	5	(158, 168]	12
(98, 108]	6		

Tabla 2.4: Asignación de clases sobre la variable de tempo agrupado

de tempo se creó una clase por valor de tempo en el set de datos (un total de 131 clases). La segunda variable y la que acabó dando resultados prometedores, divide el tempo en grupos de 10 como se muestra en la Tabla 2.4.

2.2.4. Partición de los datos

La separación de los datos en los sets de entrenamiento y test se realizó utilizando la clase `train_test_split` del paquete `scikit-learn` de Python. Se configuró una partición del 80 % de entrenamiento y un 20 % de test, apuntando a una distribución plana de las variables a predecir. Durante el entrenamiento se utilizó un 25 % de los datos de entrenamiento como set de validación, por lo que la partición total de entrenamiento, validación y test es del 60 %, 20 %, y 20 % de los datos totales respectivamente. Las dimensiones totales se pueden ver en la Tabla 2.5.

Variable	**Entrenamiento**	**Validación**	**Test**
X (entrada)	(11400, 128, 129, 1)	(3800, 128, 129, 1)	(3800, 128, 129, 1)
género	(11400, 10)	(3800, 10)	(3800, 10)
tempo	(11400, 1)	(3800, 1)	(3800, 1)
tempo discreto	(11400, 131)	(3800, 131)	(3800, 131)
tempo agrupado	(11400, 13)	(3800, 13)	(3800, 13)

Tabla 2.5: Dimensiones de los datos utilizados para el entrenamiento de la red

En la Figura 2.5 se puede ver la distribución de las variables objetivo con el corte entrenamiento/test. En la primera gráfica (de izquierda a derecha) se ve la variable de género, que está divida a partes iguales en todas las clases. Las dos gráficas restantes pertenecen a cortes sobre la variable de tempo agrupado. La gráfica de en medio muestra

el número total de registros en cada clase de tempo y la gráfica de la derecha muestra las distribuciones de densidad de cada parte del corte. Ésta última hace que sea más visible que los cortes son de proporciones iguales en todas las clases.

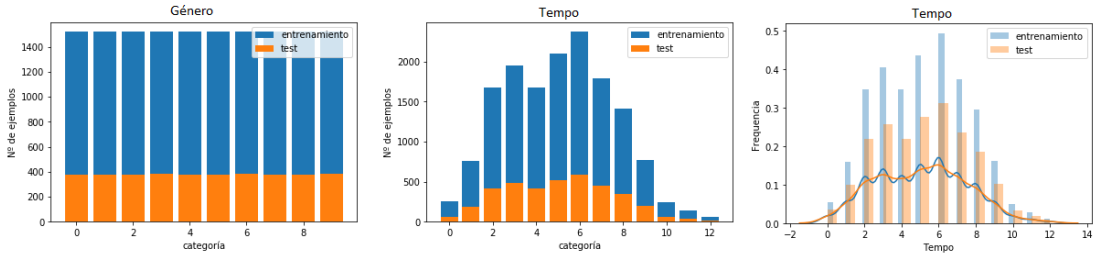

Figura 2.5: Distribución de las variables objetivo después del corte de entrenamiento y test.

2.3. Modelos y arquitecturas

El objetivo de este proyecto es diseñar una red que sea capaz de predecir género y tempo simultáneamente, para esto se debe de crear una red que tenga por lo menos dos ramas en la salida. Revisando la literatura se aprecia que los filtros verticales y horizontales procesados de forma separada y luego unidos mejoran la eficiencia de la extracción de información musical. Por lo que la red debería tener ramas paralelas interiormente además de la ramificación de salida (para las predicciones). Dada la complejidad del problema, se decidió diseñar la red por fases:

1. Diseño de una red CNN para predicción de género simple
2. Diseño de una red CNN para predicción de género con ramas paralelas de filtros verticales y horizontales
3. Diseño de una red CNN para la predicción de tempo
4. Diseño de una red CNN Para la predicción simultanea de género y tempo

Varias arquitecturas se probaron en cada fase y fueron modificadas según las pruebas eran efectivas o no, a continuación se describen las arquitecturas principales probadas.

2.3.1. Diseño de una red CNN simple para predicción de genero

La red que se utiliza aquí sirve como base para todas las demás, ésta es la arquitectura basada en el código de Guimares (2017). La red simple de predicción de género tiene la siguiente estructura:

- **Entrada:** Imágenes de tamaño 128x129x1 (solo un canal)
- **Bloque 1:**
  - Convolución 2D: 16 filtros 3x3, activación relu
  - Pooling 2D: Maxpooling 2x2
  - Dropout: dropout del 25 % de activaciones
- **Bloque 2:**
  - Convolución 2D: 32 filtros 3x3, activación relu
  - Pooling 2D: Maxpooling 2x2
  - Dropout: dropout del 25 % de activaciones
- **Bloque 3:**
  - Convolución 2D: 64 filtros 3x3, activación relu
  - Pooling 2D: Maxpooling 2x2
  - Dropout: dropout del 25 % de activaciones
- **Bloque 4:**
  - Convolución 2D: 128 filtros 3x3, activación relu
  - Pooling 2D: Maxpooling 2x2
  - Dropout: dropout del 25 % de activaciones
- **Bloque 5:**
  - Convolución 2D: 64 filtros 3x3, activación relu
  - Pooling 2D: Maxpooling 4x4
  - Dropout: dropout del 25 % de activaciones

- **Salida:** Capa densa con 10 neuronas (salidas), activación softmax (para multi-clasificación)

Esta red funciona satisfactoriamente bien, como se vera en la sección de resultados, y por esto se ha dejado como está. Las siguientes arquitecturas son las que han necesitado un poco más de uso de la intuición además de refinamiento por prueba y error, pero todas utilizan bloques similares en cada convolución de Convolución 2D → Pooling (máx.) → Dropout (25 %).

2.3.2. Diseño de una red CNN con ramas paralelas para predicción de genero

En esta arquitectura se divide la red en dos ramas con filtros de estructura horizontal o vertical como se muestra en la Figura 2.6.

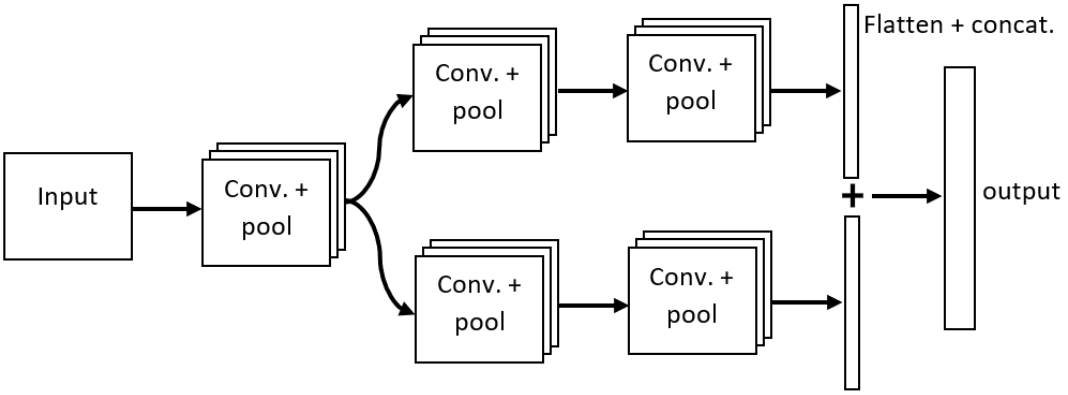

Figura 2.6: Esquema de arquitectura de una red CNN con ramas paralelas.

La red tiene las mismas entradas y salidas que la red simple, al igual que el primer bloque de filtros convolucionales, pooling y dropout situado justo después de la entrada (sección 2.3.1). Los bloques de las dos ramas paralelas son como se describen a continuación:

- **Rama 1:** Filtros verticales
  - **Bloque 1:**

    - Convolución 2D: 32 filtros 8x2, activación relu
    - Pooling 2D: Maxpooling 2x2
    - Dropout: dropout del 25 % de activaciones
  - **Bloque 2**
    - Convolución 2D: 64 filtros 8x2, activación relu
    - Pooling 2D: Maxpooling 4x4
    - Dropout: dropout del 25 % de activaciones

- **Rama 2:** Filtros horizontales
  - **Bloque 1:**
    - Convolución 2D: 32 filtros 2x8, activación relu
    - Pooling 2D: Maxpooling 2x2
    - Dropout: dropout del 25 % de activaciones
  - **Bloque 2:**
    - Convolución 2D: 64 filtros 2x8, activación relu
    - Pooling 2D: Maxpooling 4x4
    - Dropout: dropout del 25 % de activaciones

2.3.3. Diseño de una red CNN para predicción de tempo

Para el diseño de la red de predicción de tempo hubieron más iteraciones que para el resto. Inicialmente se hicieron pruebas con una red muy parecida a la red simple y con dos tipos de salidas: una salida de predicción numérica y otra de clasificación (Figura 2.7).

En la Figura 2.7 se ve las dos arquitecturas iniciales para la predicción de tempo. Al igual que en las arquitecturas anteriores todos los bloques contienen Maxpooling y dropout. La arquitectura de la parte superior pertenece a la predicción numérica de tempo, con varias capas densas al final. La arquitectura de la parte inferior de la imagen pertenece a una predicción por clases, dónde cada clase era un número entero de tempo entre 38 y 168. Estas arquitecturas resultaron ser muy poco eficientes y se recurrió a filtros verticales y horizontales más alargados, acercándose ligeramente a los filtros vistos en la literatura.

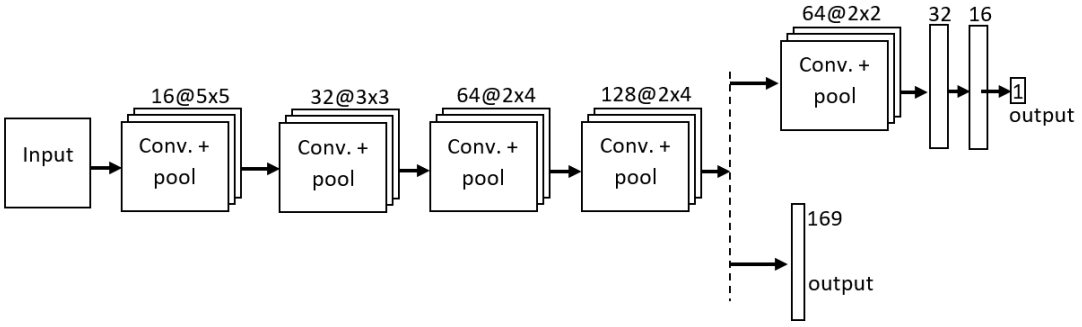

Figura 2.7: Esquema de arquitectura de una red CNN simple para predicción de tempo.

La siguiente tanda de arquitecturas para tempo se componía de estructuras similares a la mostrada en la Figura 2.7, pero con filtros horizontales y verticales de tamaño 8x2 y 2x7. La mayor estabilidad se obtuvo con una red de tres bloques de 8x2 (de 16, 32 y 32 filtros) seguida de dos bloques 2x7 (64 y 128 filtros) antes de los trozos finales para los dos tipos de red (numérica y clasificación). Aún siendo más estable y obteniendo menos sobreajuste, los resultados eran menos que satisfactorios. Llegados a este punto se decidió substituir la variable objetivo por el tempo agrupado (Tabla 2.4, sección 2.2.3).

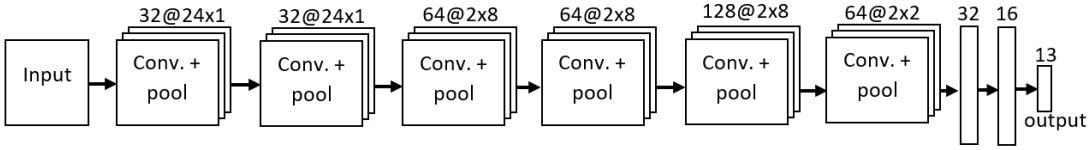

Figura 2.8: Esquema de arquitectura de una red CNN para predicción de tempo agrupado con filtros verticales y horizontales (versión H2V).

La figura 2.8 muestra una red de tempo con filtros horizontales y verticales más definidos. Se hicieron dos variedades de esta red: la primera con los filtros mostrados en el esquema (versión H2V) y la segunda (versión V2H) con las dimensiones de los filtros intercambiadas, poniendo primero filtros horizontales seguidos de filtros vérticales (1x24 y 8x2 respectivamente). Estas dos redes llegaban rápido a una precisión mejor que las anteriores pero se estancaban. No obstante, se pensó que al juntar los predictores de género y tempo en una misma red la retro-propagación proveniente del género ayudaría a encontrar mejores filtros generales, al ser más estable.

2.3.4. Diseño de una red CNN para predicción de género y tempo

Al igual que con los problemas individuales se han probado dos arquitecturas distintas para éste problema. Inicialmente se probó con una arquitectura derivada de la red simple utilizando filtros genéricos de 3x3. Como ya se ha comentado en la sección anterior, aunque el tempo todavía no obtenga buenos resultados, se espera que la red aprenda mejor gracias a la estabilidad de la predicción de género.

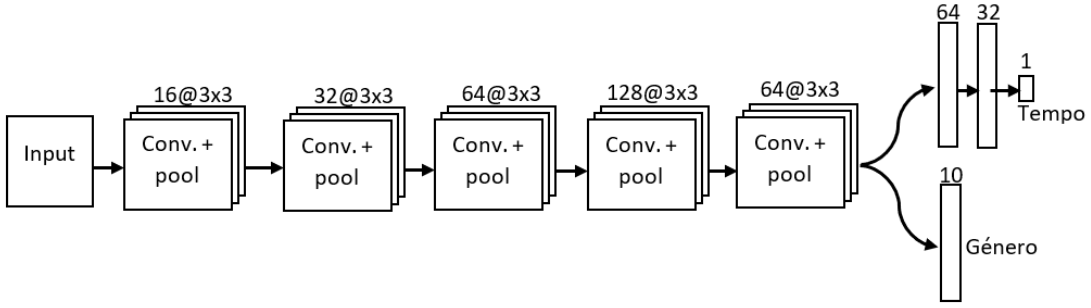

Figura 2.9: Esquema de arquitectura de una red CNN simple para predicción de tempo y género.

La Figura 2.9 muestra la estructura de la red que hace una predicción numérica del tempo y una clasificación de género. Nuevamente, al igual que todas las arquitecturas presentadas, cada bloque contiene una capa de Maxpooling y Dropout después de la convolución.

Finalmente, se juntaron todas las pruebas realizadas hasta el momento para hacer una propuesta de una red más estable capaz de predecir género y tempo. Para esta red se utilizó la variable de tempo agrupado teniendo dos clasificadores como salida de la red.

La Figura 2.10 muestra una red con ramas paralelas de filtros horizontales y verticales que se unen para predecir la clase de género. Para el tempo es más importante sacar resolución temporal de intensidades que resolución sobre las frecuencias y por esto está solo conectado a la rama horizontal. Por otro lado el género es dependiente de las dos ramas y, al compartir rama con la predicción de tempo, se prevé que la retro-propagación del género ayude a aprender al tempo sin resultar perjudicados significativamente los pesos para predicción de género.

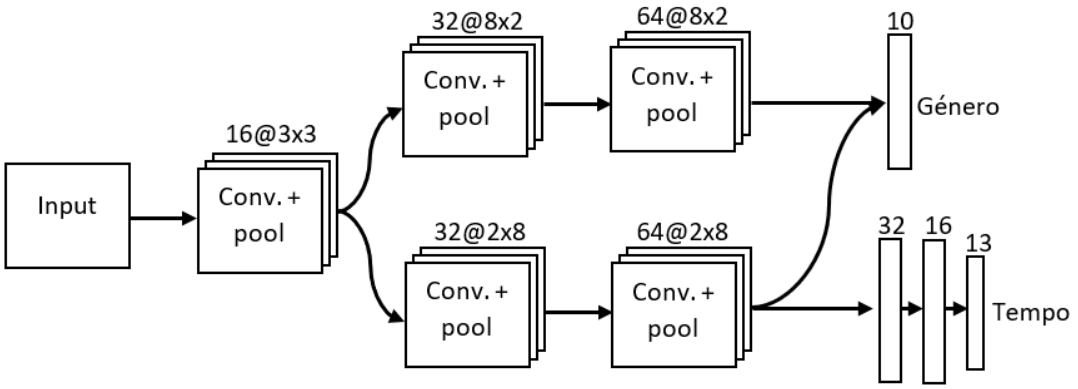

Figura 2.10: Esquema de arquitectura de una red CNN simple para predicción de tempo y género.

2.3.5. Algoritmos de aprendizaje y otros meta-parámetros

El énfasis de este estudio ha sido la arquitectura de la red, por lo que los meta-parámetros no arquitectónicos de la red no han sido explorados exhaustivamente.

Para el algoritmo optimizador se han explorado los algoritmos RMSprop y ADAM con varios valores de ratio de aprendizaje inicial entre 0.1 y 0.001. Dado que el valor por defecto de 0.001 en ambos métodos daba la mayor estabilidad del algoritmo se ha elegido éste para todas las arquitecturas. Para predicción de género ADAM funcionaba bien, pero para predicción de tempo la utilización de RMSprop obtiene mejores estimaciones iniciales y ofreciendo estabilidad al modelo. Se piensa que la inclusión de momento con ADAM creaba modelos que se salían de la zona de aprendizaje cuando se trataba de predecir tempo.

Para el problema actual, especialmente con la entrada en decibelios, los valores positivos/grandes son más importantes que los negativos/pequeños, por este motivo se realiza *maxpooling* después de las convoluciones siempre y se ha elegido la función de activación ReLU (Rectified Linear Unit). Para las predicciones multi-clase siempre se ha elegido una activación *softmax*, mientras que las predicciones de valores de tempo se han hecho utilizando una activación lineal en la última capa.

La función de perdida utilizada para las clasificaciones es la de Entropía cruzada (`categorical_crossentropy`) con una métrica de precisión (`accuracy`) para clases en

las predicciones cualitativas y el error medio cuadrado para las predicciones numéricas, con una métrica del R^2.

El valor del Dropout inicialmente estaba en torno al 25 %, pero debido a divergencias entre los valores de entrenamiento y validación, se amplió el ratio de Dropout al 50 % en varios modelos.

Finalmente, se ha utilizado el valor de coste en los datos de validación para evaluar los modelos, al igual que para implementar un *Early Stopping* y frenar el aprendizaje antes de que ocurra un sobre-ajuste o el modelo deje de aprender.

Los métodos importados para la construcción de las redes son del paquete Keras (utilizando el *backend* de TensorFlow). Se ha utilizado el modo funcional de Keras para un mayor control de la arquitectura de la red y para la monitorización de los resultados se ha utilizado el paquete `tensorboard` de TensorFlow.

2.4. Resultados

En los siguientes apartados se van a comentar los resultados de las arquitecturas investigadas en la sección anterior. Para cada arquitectura se hicieron cambios de metaparámetros hasta obtener resultados concluyentes utilizando los resultados de validación como objetivo.

Ninguno de los resultados obtenidos investigando la utilización del tempo como una variable continua fueron concluyentes y no entran en la discusión de resultados. No se encontró una arquitectura capaz de predecir esta variable resultando siempre en gradientes desvanecedoras y todas las predicciones de tempo al rededor de la media de la variable.

A continuación se presentará una discusión de los mejores resultados obtenidos en cada apartado, comparando las precisiones obtenidas en los datos de entrenamiento y test. Los resultados de entrenamiento de cada una de las redes discutidas se pueden ver en el apéndice A.

2.4.1. Predicción de género

Como ya se ha comentado en apartados anteriores el género ha resultado ser una variable muy estable de predecir, siempre tiende a una solución con una precisión bastante alta. Por lo que ha resultado fácil hacer pruebas con ésta variable. Es posible que este resultado se deba a que cada género tiene una textura característica en representaciones por espectograma, lo que lo hace fácil de clasificar para una CNN.

Una de las primeras pruebas que se realiza es la comparación de resultados cambiando la entrada entre un espectrograma directo o un espectrograma escalado en decibelios (Figuras 2.3 y 2.4 respectivamente).

	Entrenamiento	**Test**
Espectrograma simple	0.90	0.84
Espectrograma en dB	0.88	0.83

Tabla 2.6: Precisión de la red de Género simple con respecto al formato de entrada

En la Tabla 2.6 se muestra la comparación de precisiones para la red simple de predicción de género cambiando la entrada. Al contrario de lo que se pensaba al diseñar la red, la red distingue mejor el género en la imagen no tratada que en la imagen transformada a decibelios. Esto se debe a que al estandarizar las imágenes la red tiene más dificultades para distinguir imágenes parecidas, ya que todas ellas acaban con un aspecto (textura) similar.

Similarmente, se compara la arquitectura paralela (Figura 2.6) con la arquitectura simple para predicción de género. Como muestra la Tabla 2.7 Los resultados son ligeramente inferiores para la arquitectura paralela. Ésto es de esperar ya que en redes profundas la serielización es más potente que la paralelización. A pesar de esto, la red de arquitectura paralela sigue teniendo una precisión buena.

	Entrenamiento	**Test**
Red simple	0.90	0.84
Red paralela	0.88	0.82

Tabla 2.7: Comparación de precisión de las redes de Género simple y paralela.

2.4.2. Predicción de tempo

La red de predicción de tempo fue más complicada de construir para que tuviera resultados aceptables. La mayoría de arquitecturas daban precisiones (o valores de R^2 en el caso de las predicciones numéricas) muy malas o dejaban de aprender rápidamente topándose con gradientes desvanecedoras. Finalmente se hicieron dos variedades una red CNN muy profunda (mostrada en la Figura 2.8). La primera variedad tiene filtros 24x1 seguidos de filtros 2x8 (que llamaremos V2H) y en la segunda variedad se invierten las dimensiones con filtros de 1x24 seguidos de filtros 8x2 (que llamaremos H2V). En ambas redes se predice el tempo agrupado en 13 clases. Estas redes siguen teniendo un aprendizaje poco estable, pero llegan rápidamente a una solución aceptable.

	Entrenamiento	**Test**
Red H2V	0.57	0.57
Red V2H	0.56	0.50

Tabla 2.8: Comparación de precisión de las redes de predicción de tempo.

En la Tabla 2.8 se comparan las precisiones de estas dos últimas redes de predicción de tempo. Las dos tienen una precisión similar en entrenamiento con una gran diferencia: la red con filtros verticales seguidos de horizontales (V2H) presenta sobre-ajuste mientras que la otra no tiene ningún sobre-ajuste. Comparando los resultados con la literatura se llega a la conclusión de que, como los filtros verticales son más importantes para la predicción de tempo, el uso de filtros horizontales después de los verticales impide el entrenamiento de éstos. Debido a estos resultados se elige utilizar filtros principalmente verticales para la predicción de tempo en la red conjunta.

2.4.3. Predicción de género y tempo

Para la predicción de género y tempo se realizaron dos pruebas principales. La primera prueba se realizó con una arquitectura simple (Figura 2.9). Esta prueba se realizó antes de conseguir ningún resultado satisfactorio con la red de tempo y el objetivo era comprobar que, al juntar género y tempo, los pesos aprendidos para la variable de género ayudarían a predecir el tempo. El resultado muestra que el tempo mejora gracias al género como se esperaba, pero al mismo tiempo la precisión del género disminuía considerablemente (Tabla 2.9).

	Entrenamiento	**Test**
Género	0.35	0.41
Tempo (R^2)	0.03	0.004

Tabla 2.9: Precisión (R^2 en este caso de Tempo) de la red simple con predicciones de género y tempo.

Finalmente, utilizando todo lo aprendido en las pruebas anteriores, se diseñó y probó una red con ramas paralelas (Fig. 2.10) que predice género y tempo de manera satisfactoria. En esta red el género de nuevo pierde precisión a favor de una mejora en la precisión del tempo, pero esta vez ambas variables obtienen buenos resultados de precisión como puede observarse en la Tabla 2.10.

	Entrenamiento	**Test**
Género	0.82	0.78
Tempo	0.68	0.62

Tabla 2.10: Precisión de la red final con predicciones de género y tempo.

A modo de cerrar el círculo de pruebas, se repitió el entrenamiento de la red final utilizando los espectrogramas escalados a decibelios para ver si los resultados obtenidos con la red simple de género se replican con el modelo final.

	Entrenamiento	**Test**
Género	0.77	0.66
Tempo	0.67	0.65

Tabla 2.11: Precisión de la red final con predicciones de género y tempo utilizando espectrogramas escalados a decibelios.

Comparando las Tablas 2.10 y 2.11, se puede observar que la precisión del género sigue empeorando cuando se usan espectrogramas en decibelios. Sin embargo, los espectrogramas en decibelios parecen ayudar a la precisión del tempo. Se sospecha que la conversión a decibelios resalta los pulsos en un rango más amplio del espectrograma, lo que los hace más distinguibles para la predicción de tempo.

2.5. Conclusión

En este trabajo se ha investigado el estado de los algoritmos y modelos de extracción de información musical (MIR) en el campo del aprendizaje profundo. Así mismo, se han investigado parámetros y arquitecturas CNN para su uso en la predicción de características rítmicas de audios de música. Las características elegidas han sido género y tempo, dada su importancia para determinar el ritmo de una pieza musical.

Las investigaciones sobre género han demostrado que las CNN son excepcionales para la clasificación de ésta variable. Estos modelos son más efectivos cuando se utilizan espectrogramas no escalados ni estandarizados, aunque un escalado en decibelios no daña significativamente la precisión.

Las investigaciones sobre tempo han demostrado que, en efecto, la selección de filtros principalmente verticales o horizontales es muy importante de cara a la eficiencia del modelo. Por lo tanto se debe tener cuidado al elegir el orden de filtros horizontales y verticales. Por otro lado, la predicción de tempo se ha visto mejorada por la utilización de espectrogramas escalados con decibelios. Este fenómeno está causado por la ampliación de intensidades de pulso sobre todas las frecuencias al convertir el espectrograma a decibelios.

Finalmente, se ha demostrado que se puede construir un modelo que prediga simultáneamente género y tempo. En un modelo conjunto la variable más estable y fácilmente predecible por la red (el género) ha ayudado a entrenar los pesos para predecir una variable mucho más complicada (el tempo).

Estos resultados no son totalmente concluyentes, ya que todavía se pueden incluir mejoras al modelo introduciendo más capas en cada rama o continuando la investigación del efecto de la alteración de los espectrogramas sobre la precisión. También, se podría investigar el uso de otras activaciones o algoritmos de aprendizaje alternativos al ADAM o el RMSprop.

En conclusión, este estudio sugiere que la predicción de ritmos en la música es posible, ya que un modelo bastante modesto ha sido capaz de predecir género y tempo de un set de datos de sólo 8 horas de música.

Apéndice A

Resultados de entrenamiento

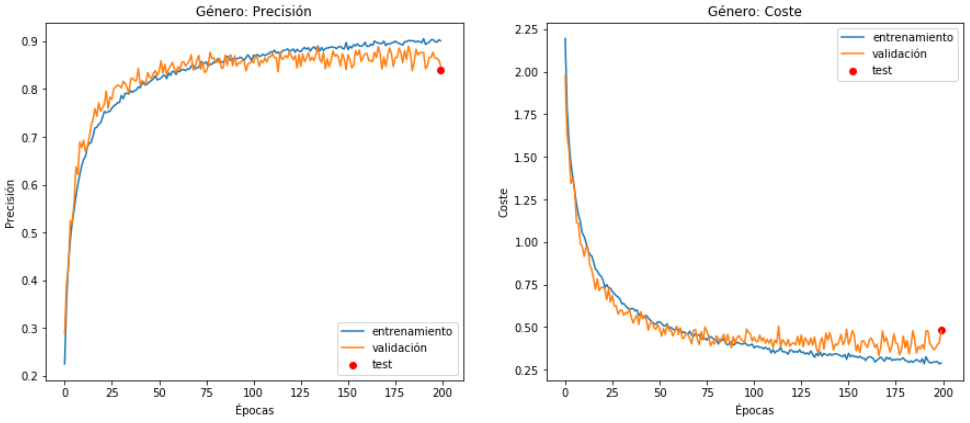

Figura A.1: Entrenamiento de la red simple para predicción de género.

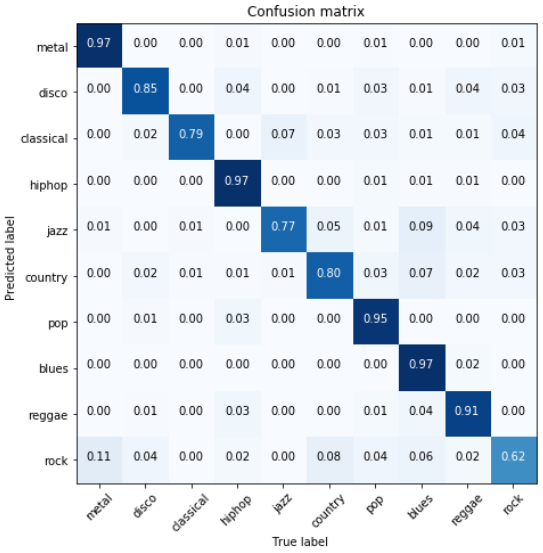

Figura A.2: Matriz de confusión de los resultados de la red simple para predicción de género.

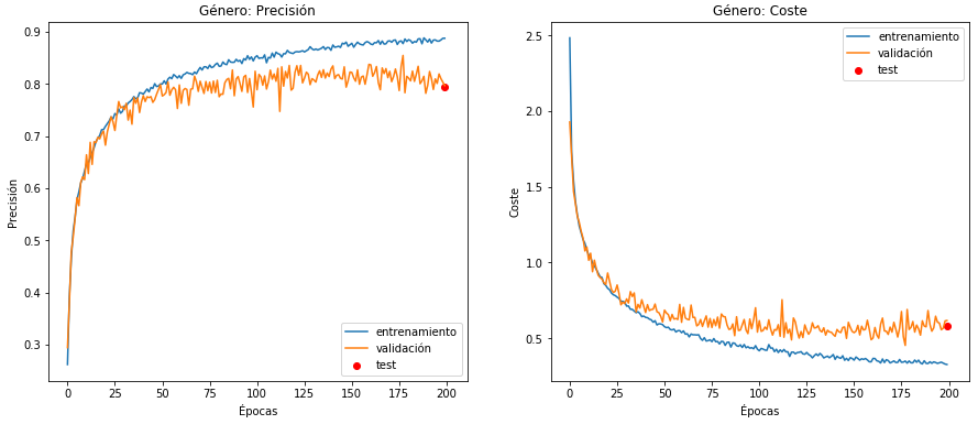

Figura A.3: Entrenamiento de la red paralela para predicción de género.

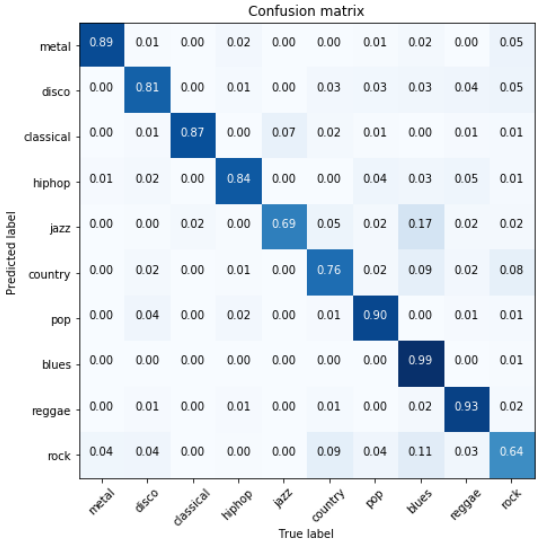

Figura A.4: Matriz de confusión de los resultados de la red paralela para predicción de género.

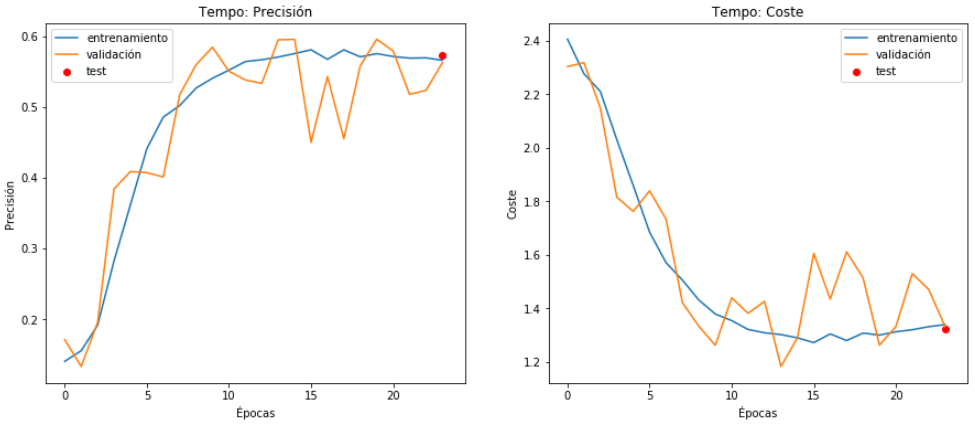

Figura A.5: Entrenamiento de la red H2V para predicción de tempo.

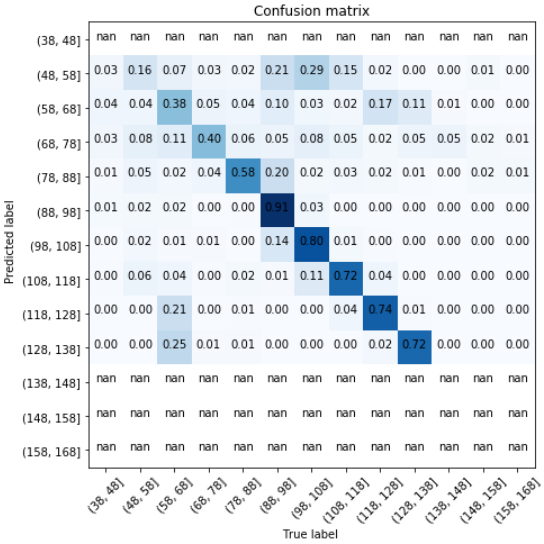

Figura A.6: Matriz de confusión de los resultados de la red H2V para predicción de tempo.

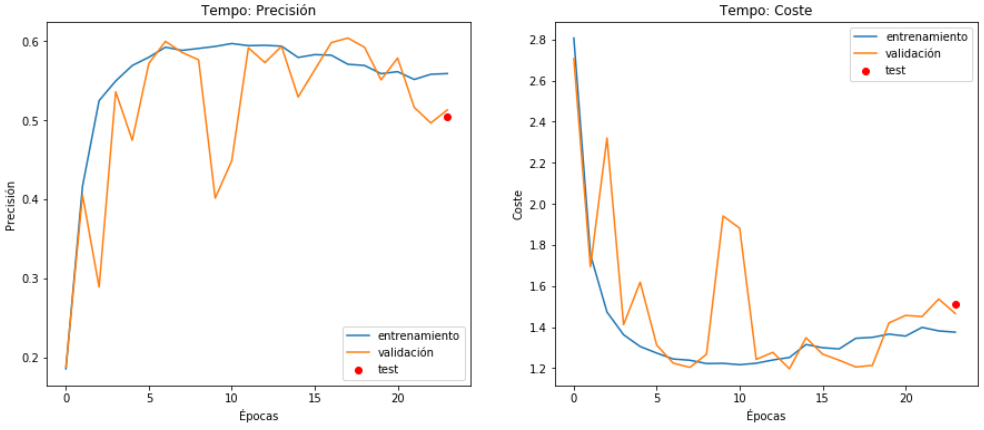

Figura A.7: Entrenamiento de la red V2H para predicción de tempo.

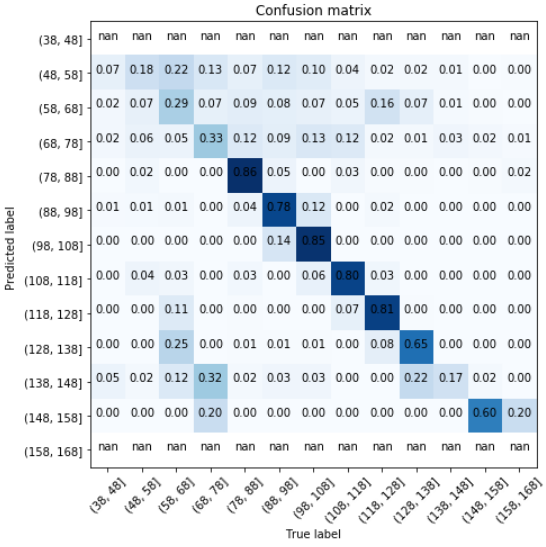

Figura A.8: Matriz de confusión de los resultados de la red V2H para predicción de tempo.

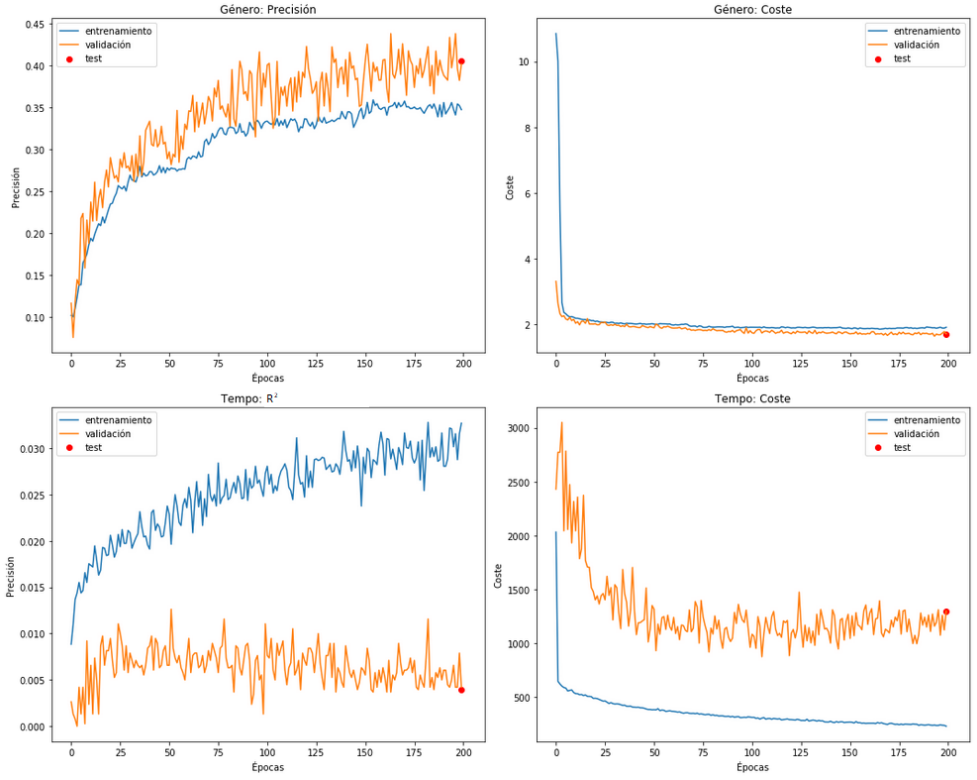

Figura A.9: Entrenamiento de la red simple para predicción de género y tempo.

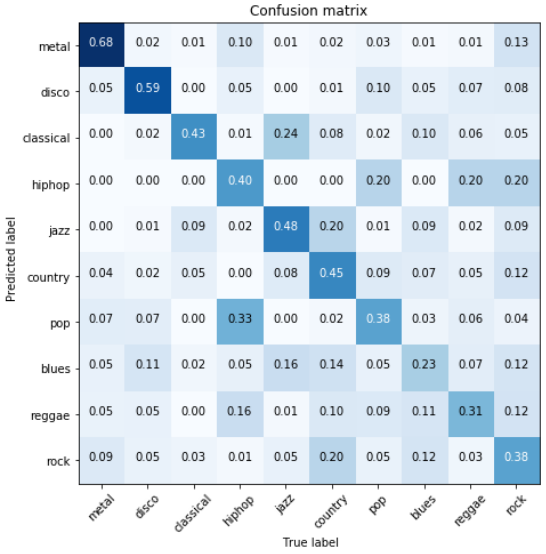

Figura A.10: Matriz de confusión de los resultados de género de la red simple para predicción de género y tempo.

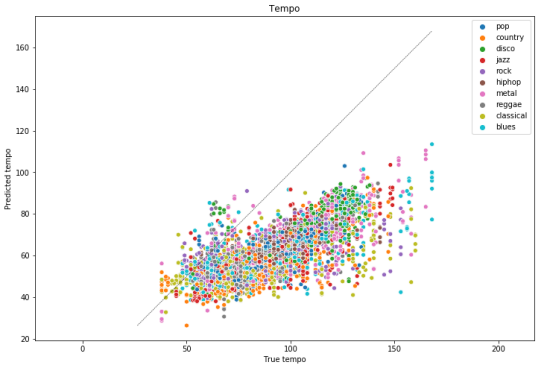

Figura A.11: Dispersión de resultados de tempo de la red simple para predicción de género y tempo.

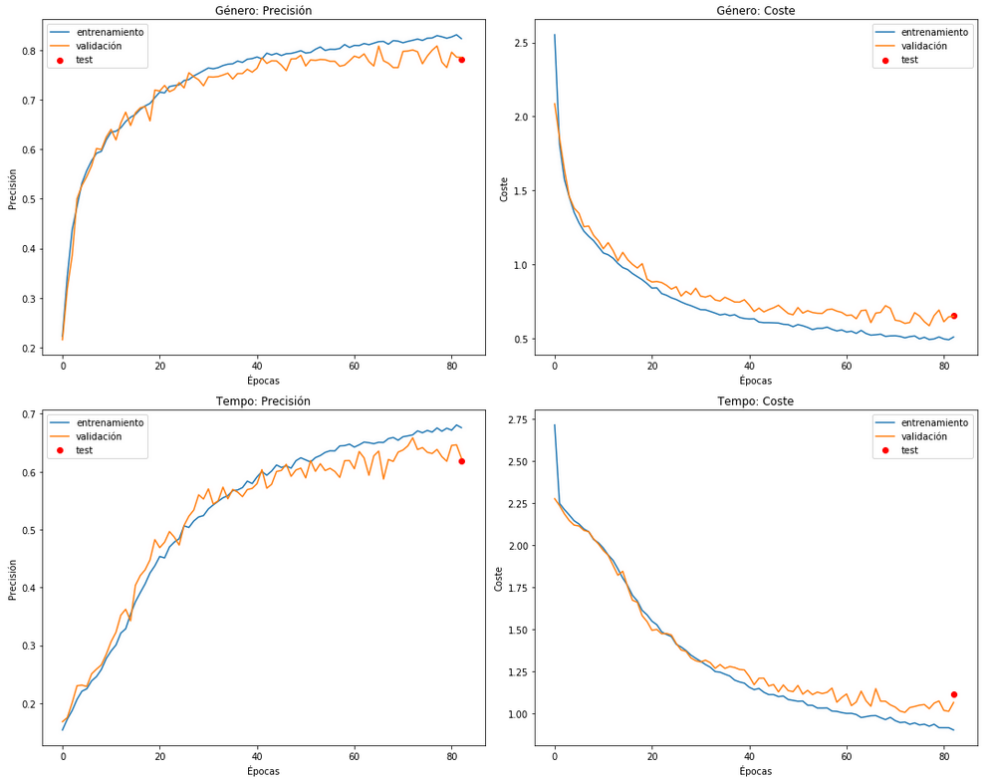

Figura A.12: Entrenamiento de la red final/paralela para predicción de género y tempo.

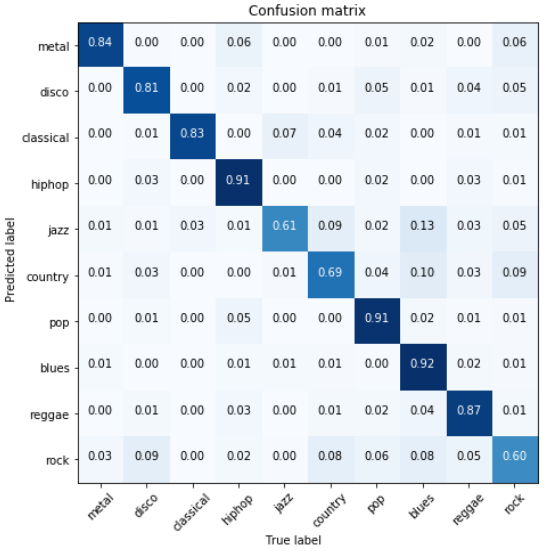

Figura A.13: Matriz de confusión de los resultados de género la red final/paralela para predicción de género y tempo.

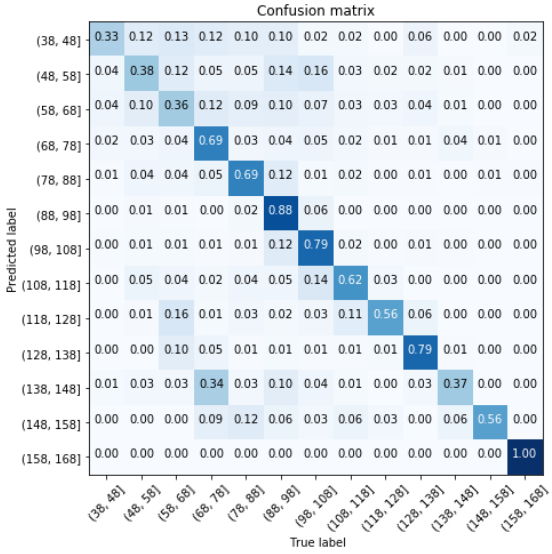

Figura A.14: Matriz de confusión de los resultados de tempo de la red final/-paralela para predicción de género y tempo.

Bibliografía

Costa, Y. M. G., de Oliveira, L. S. & Silla, C. (2017), 'An evaluation of convolutional neural networks for music classification using spectrograms', *Applied Soft Computing* **52**.

Eck, D. & Schmidhuber, J. (2002), Finding temporal structure in music: blues improvisation with lstm recurrent networks, *in* 'Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing', pp. 747–756.

Elowsson, A. (2018), 'Tempo-invariant processing of rhythm with convolutional neural networks', *CoRR* **abs/1804.08167**.
URL: *http://arxiv.org/abs/1804.08167*

Gers, F. A. & Schmidhuber, J. (2001), 'Lstm recurrent networks learn simple context free sensitive language', *IEEE Transactions on Neural Networks* **12**(6), 1333–1340.

Gers, F. A., Schmidhuber, J. & Cummins, F. A. (2000), 'Learning to forget: Continual prediction with lstm', *Neural Computation* **12**(10), 2451–2471.

Guimares, H. (2017), 'Music Genre classification on GTZAN dataset using CNNs '.
URL: *https://github.com/Hguimaraes/gtzan.keras*

Han, Y., Kim, J. & Lee, K. (2016), 'Deep convolutional neural networks for predominant instrument recognition in polyphonic music', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **25**(1), 208–221.

J Humphrey, E., Bello, J. & Lecun, Y. (2012), 'Moving beyond feature design: Deep architectures and automatic feature learning in music informatics', *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012* .

Jehan, T. (2005), Downbeat prediction by listening and learning, *in* 'IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.', pp. 267–270.

Klapuri, A. P., Eronen, A. J. & Astola, J. T. (2006), 'Analysis of the meter of acoustic musical signals', *IEEE Transactions on Audio, Speech, and Language Processing* **14**(1), 342–355.

Lazaro, A., Sarno, R., André, R. & Mahardika, M. N. (2017), 'Music tempo classification using audio spectrum centroid, audio spectrum flatness, and audio spectrum spread based on mpeg-7 audio features', *2017 3rd International Conference on Science in Information Technology (ICSITech)* pp. 41–46.

Lee, H., Pham, P., Largman, Y. & Ng, A. Y. (2009), Unsupervised feature learning for audio classification using convolutional deep belief networks, *in* Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams & A. Culotta, eds, 'Advances in Neural Information Processing Systems 22', Curran Associates, Inc., pp. 1096–1104.
URL: *http://papers.nips.cc/paper/3674-unsupervised-feature-learning-for-audio-classification-using-convolutional-deep-belief-networks.pdf*

Lewis (1988), Creation by refinement: a creativity paradigm for gradient descent learning networks, *in* 'IEEE 1988 International Conference on Neural Networks', pp. 229–233 vol.2.

Pons, J. (2018), 'Neural networks for music: A journey through its history', *Towards Data Science* .
URL: *https://towardsdatascience.com/neural-networks-for-music-a-journey-through-its-history-91f93c3459fb*

Pons, J., Nieto, O., Prockup, M., Schmidt, E. M., Ehmann, A. F. & Serra, X. (2017a), 'End-to-end learning for music audio tagging at scale', *CoRR* **abs/1711.02520**.
URL: *http://arxiv.org/abs/1711.02520, https://github.com/jordipons/music-audio-tagging-at-scale-models*

Pons, J., Slizovskaia, O., Gong, R., Gómez, E. & Serra, X. (2017b), 'Timbre analysis of music audio signals with convolutional neural networks', *CoRR* **abs/1703.06697**.
URL: *http://arxiv.org/abs/1703.06697*

Schreiber, H. & Meinard, M. (2018), A single-step approach to musical tempo estimation using a convolutional neural network, *in* 'Proceedings of the 19th International

Society for Music Information Retrieval Conference (ISMIR)', Paris, France, pp. 98–105.

Sheng Gao & Chin-Hui Lee (2004), An adaptive learning approach to music tempo and beat analysis, *in* '2004 IEEE International Conference on Acoustics, Speech, and Signal Processing', Vol. 4, pp. iv–iv.

Todd, P. M. (1988), A sequential network design for musical applications, *in* 'Connectionist Models Summer School', pp. 76–84.