



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Clasificador de música en subgéneros personalizados

PROYECTO FINAL DE MÁSTER

Máster en Inteligencia Artificial,
Reconocimiento de Formas e Imagen Digital

Autor: Carlos Ortiz Almendros

Director: Roberto Paredes Palacios

13 de julio de 2015

Resumen

Este proyecto de investigación se enmarca en el campo del aprendizaje automático sobre datos de audio. A partir de una colección de música, se pretende realizar la clasificación de todas las canciones de dicha biblioteca utilizando como referencia un número concreto de canciones que el usuario discrimina en función al estilo musical personal que él determina y al cual pertenecen. El objetivo de este aprendizaje automático es poder aplicar el clasificador a colecciones de gran tamaño de forma que el usuario pueda filtrar según sus gustos personales las canciones de dicha colección.

Un ejemplo de aplicación comercial de este sistema pueden ser tiendas online de música o colecciones privadas de profesionales del mundo de la música, ya que agilizan el proceso de exploración de nuevas canciones publicadas por artistas o simplemente la organización de bibliotecas privadas de una forma automática o semi-automática, por que puede ser la base de un sistema guiado de clasificación.

A lo largo de este proyecto se expondrán todas las fases de la investigación que han sido desarrolladas, así como los resultados obtenidos y las técnicas aplicadas.

Palabras clave: clasificación, música, música electrónica, estilos, subgéneros, BPM, muestras, red neuronal, PCA, corpus

Índice general

Lista de figuras	3
1. Preámbulo	4
2. Estado del arte	5
3. Objetivo	7
4. Conjunto de datos	8
4.1. Características del género electrónico	10
4.2. Preparación del audio	11
5. Preproceso	13
5.1. PCA	13
5.2. Barajado de datos	14
5.3. Particionado del corpus	14
6. Entrenamiento del sistema	15
6.1. Redes Neuronales	15
6.2. DeepLearnToolbox for MatLab	16
6.3. Configuración	17
7. Evaluación	18
7.1. Experimentación	18
7.2. Comparativa	19
8. Conclusiones	20
9. Futuros trabajos	21

Índice de figuras

4.1. Distribución del corpus en géneros.	9
4.2. Onda de audio típica del género electrónico.	10
4.3. Ejemplo de comando <i>FFmpeg</i>	12
5.1. Compresión de segmentos mediante PCA.	14
6.1. Ejemplo de red neuronal que clasifica una muestra en 3 posibles clases a partir de un vector de 5 características. [7]	16
6.2. Configuración, entrenamiento y evaluación con <i>DeepLearnToolbox</i>	17
7.1. Ejemplo de la evolución del entrenamiento de una ejecución. .	18

Capítulo 1

Preámbulo

La motivación para realizar esta investigación surge de la necesidad de un sistema que sea capaz de reconocer los gustos musicales de los usuarios y pueda recomendarles canciones de forma personalizada. Esta necesidad es muy evidente en un caso concreto alrededor del cual se va a realizar la investigación, este caso es el de la música electrónica.

Tras este estilo musical existe un amplio sinfín de subgéneros y tendencias musicales, incluso muchos de estos se solapan entre sí, por lo que existe ambigüedad cuando se intenta diferenciar las canciones por género. Los profesionales de este estilo, como son los productores, diskjockeys e incluso aficionados, exploran a diario plataformas online especializadas en música electrónica en busca de nuevas aportaciones a sus colecciones musicales. Esta tarea de exploración resulta bastante pesada ya que para encontrar música del agrado del usuario es necesario escuchar muchas canciones, de las cuales la gran mayoría pueden no ser de su interés.

Por ello es interesante la idea de poder dar libertad a los usuarios para determinar sus preferencias musicales, definiendo sus propios géneros musicales utilizando determinadas canciones como ejemplo, a partir de los cuales el sistema aprenderá y podrá ofrecerle al usuario canciones muy aproximadas a lo que busca.

Capítulo 2

Estado del arte

En la literatura el tema de la clasificación de música por género no es algo nuevo, existen diversas investigaciones en esta línea, no de una forma continua en el tiempo, pero cada 3 o 4 años aparece un nuevo artículo en este campo.

Uno de los primeros trabajos de clasificación de música en géneros utilizando aprendizaje automático es el trabajo de Tzanetakis, Essl & Cook [4], estos autores proponen el uso de audio real como entrada a sistemas de reconocimiento de formas, utilizando como técnicas de extracción de características Discrete Wavelet Transform (DWT), un método eficiente para la descomposición en octavas de la señal de audio. Posteriormente aplican una rectificación completa de onda, filtro de paso bajo, submuestreado, normalización y Fast Fourier Transformation (FFT) para la autocorrelación. Como se puede observar el preproceso es una parte fundamental en esta investigación, por ello se aplican filtros diferentes en función del género que se va a clasificar. El artículo de Li, Ogihara & Li [8] apuesta por un nuevo método de extracción de características, DWCHs (Daubechies Wavelet Coefficients Histograms). Este método captura información tanto global como local de la señal de la música mediante el computo de histogramas basados en coeficientes de Daubechies.

Desde 2003 hasta 2009 se produce un vacío en cuanto a investigación acerca de clasificación de música por géneros, de hecho es posible ver artículos como el de McKay & Fujinaga [3] en los que se empieza a plantear la problemática de la clasificación por género, exponiendo como argumento los solapamientos que existen entre muchos géneros y los confusos patrones que aplican los expertos a la hora de etiquetar audio en un determinado género.

Sin embargo en 2009 aparece una nueva serie de artículos como el de Mei-Lan

Chu [2], el cual propone métodos de extracción de características independientes de los géneros a tratar. Son dos las características que se extraen, por una parte el timbre del audio mediante cepstrales y energía del audio, y por otro lado el ritmo del sonido, aplicando algoritmos de detección de beats y melodía. Otro enfoque a la extracción de características es el que proponen Salamon, Rocha & Gómez [5], ya que utilizan características de la melodía a alto nivel extraídas directamente de la señal de audio polifónico. Estas características se extraen mediante una búsqueda automática del contorno del tono que predomina la melodía.

De este repaso del estado del arte es posible destacar la importancia que tiene en todos los proyectos existentes hasta la fecha la importancia que tiene el preproceso necesario para la extracción de características y la necesidad de un etiquetado por parte de expertos de todos los audios con el fin de conseguir sistemas que generalicen bien la clasificación de música por género. Por lo tanto, las principales aportaciones de este proyecto de investigación a la comunidad científica es crear un sistema que no necesite de preprocesos de filtrado del audio, consiguiendo de esta forma aplicar a cualquier género musical la clasificación.

Otro aspecto innovador es el etiquetado por parte del usuario, quizás esta sea la diferencia más notable con toda la literatura existente, ya que no se pretende realizar un clasificador que generalice para estilos musicales definidos por expertos, si no que el objetivo es conseguir adaptar la clasificación al etiquetado del audio por parte del usuario en función a sus distinción objetiva entre géneros musicales. Además este proyecto se centra en la aplicación directa sobre un determinado estilo musical en constante expansión, ya que diariamente se realizan nuevas publicaciones en plataformas tan importantes como *BeatPort* o *Soundcloud* entre otras. Además la idea de que sea el usuario el que determine los géneros según sus preferencias es un aspecto novedoso y muy interesante desde el punto de vista comercial de recomendación de productos.

Capítulo 3

Objetivo

La finalidad de este proyecto de investigación es estudiar el comportamiento de un sistema de aprendizaje automático ante una colección de música, con el objetivo de intentar que el sistema aprenda a identificar los matices que diferencia un género musical de otro. Todo ello con el añadido de que los géneros musicales no son estáticos, si no que son prefijados por el usuario de la biblioteca musical.

Para no limitar los posibles géneros que el usuario pueda definir, el audio se procesará en bruto, sin ningún tipo de filtrado ni algoritmo aplicado. Por este motivo hay que tener en cuenta que el objetivo no es tratar de conseguir un sistema que clasifique lo mejor posible, si no, comprobar qué se puede conseguir entrenando el sistema a partir del audio y analizar las posibles mejoras al sistema.

Capítulo 4

Conjunto de datos

Se ha utilizado como conjunto de datos la biblioteca personal de música electrónica del autor del proyecto [6], compuesta por 1035 canciones de una duración media de 6 minutos cada una y en el formato comprimido MP3. El 90% de estas canciones se han utilizado en el entrenamiento y el 10% restante se han reservado para la evaluación. Se han distinguido 10 géneros diferentes a juicio del autor del proyecto, hay que tener en cuenta que puede haber ambigüedad en algunas canciones en las que el género no este bien etiquetado, pero este factor es un componente más a asumir si se quiere evaluar el sistema con un caso real.

- *Deep Ambient*: Bombos no muy definidos, gran cantidad agudos de claps y hithats, voces, melodías de piano e instrumentos de viento.
- *Deep Bass*: Canciones donde el principal componete es una melodía de bajos potentes y graves, con claps muy marcados.
- *Deep Jump*: Destacan por bombos y hit hats muy definidos con estructuras de repetición alternas, no siempre cada 4 tiempos.
- *Deep Minimal*: Gran cantidad de bajos continuos de fondo y percusiones muy características, junto con redobles de claps.
- *Deep Tech*: Principalmente se basan en bombos muy marcados junto con bajos y algun sample corto como voces o efectos.
- *Deep Terrace*: Sonidos de xilófonos, marimbas o instrumentos muy re-

conocibles, con muchos efectos de cascada de beats (aceleración progresiva de los golpes que culmina en un corte seguido del ritmo original de la canción).

- *Deep Vocal*: El principal componente de estas canciones en una vocal, bien continua o repetida a modo de bucle.
- *Plastikman*: Sonidos puros de sintetizadores con gran calado electrónico. Este género es propio de un solo artista, algo que puede ser interesante para encontrar artistas con sonidos similares.
- *Techno Bass*: Canciones muy repetitivas fundamentalmente de bombos y platos.
- *Techno Minimal*: Mismos aspectos que el Deep Minimal pero más acen- tuados y con un tempo más rápido.

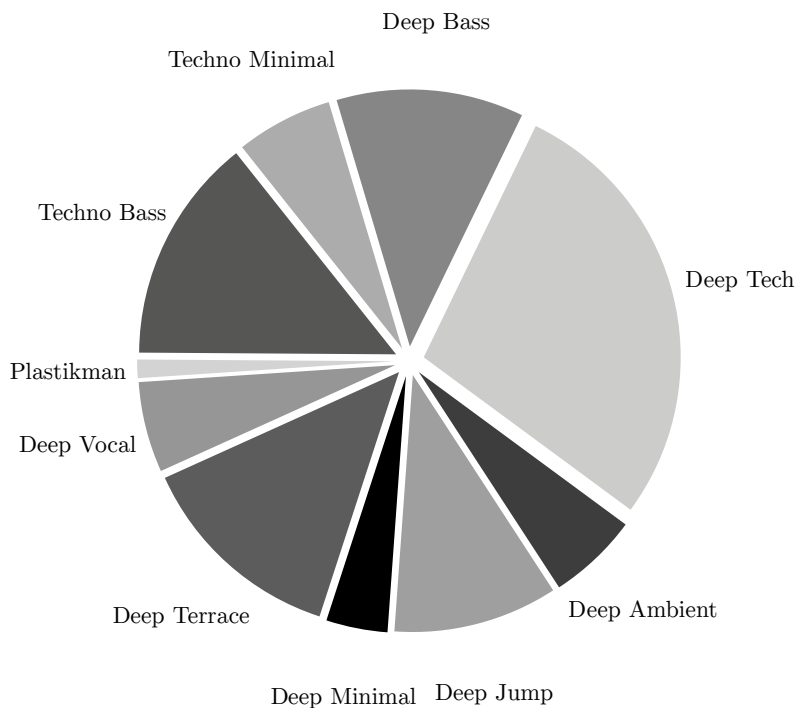


Figura 4.1: Distribución del corpus en géneros.

Las canciones no tienen un BPM (*Beats Por Minuto*) común, lo cual no es relevante puesto que lo que se pretende es analizar los componentes del sonido, no su estructura a lo largo del tiempo. Este aspecto hace que sea posible realizar una compresión de los datos tal y como se expone en el capítulo 5.1. El etiquetado de las canciones se ha realizado de forma manual, distribuyendo las canciones de un mismo género en un mismo directorio. Este proceso sería semi-automático en aplicaciones como tiendas online, ya que según el comportamiento que mostraría el usuario al recomendarle canciones de un mismo género a priori, aceptándolas o rechazándolas, el sistema etiquetaría automáticamente las canciones recomendadas.

Uno de los principales hándicaps de este proyecto es la importante influencia que tiene esta discriminación de géneros por parte del usuario, ya que cuanto mayor similitud o solapamiento exista entre estilos, será más complicado clasificar cada canción en su género correcto. Otro factor a tener en cuenta a la hora de valorar los resultados respecto del corpus, es la cantidad de géneros que el usuario diferencia y el número de canciones que etiqueta en cada uno.

Tras esta introducción a los datos de audio con los que se va a trabajar, es momento de profundizar en las características de los mismos y estudiar el procesado que es necesario para adaptar los datos al sistema de aprendizaje automático.

4.1. Características del género electrónico

La música electrónica se caracteriza por el uso repetitivo de instrumentos y samples (*pequeños cortes de audio*), por lo que el audio tiene una estructura muy definida ya que cada sonido es repetido en una escala temporal de múltiplos de 4. Este hecho hace que una pequeña muestra del audio sea lo suficientemente característica como para poder realizar la discriminación entre muestras.

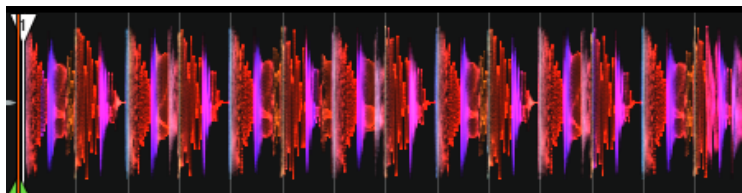


Figura 4.2: Onda de audio típica del género electrónico.

Así como una banda de rock tiene siempre presentes ciertos elementos como una batería, bajos, guitarras y teclados, el género electrónico tiene elementos acústicos comunes a cualquiera de sus subgéneros, siendo cada subgénero diferenciado de otro por la mayor presencia de determinados elementos o por el timbre característico de alguno de ellos. Los principales elementos de una canción electrónica son:

- Claps
- Hihats
- Timbales
- Bombos
- Bajos
- Sintetizadores
- Teclados
- Platos
- Samples

El usuario identifica consciente o inconscientemente los elementos de las canciones y puede enmarcar la canción en el género que él crea. Por ejemplo, para un determinado subgénero puede ser descriptivo la aparición de ciertos bajos característicos o un determinado patrón de repetición en los bombos. La idea es que el sistema sea capaz de encontrar estos matices que el usuario hace de forma instintiva y natural.

4.2. Preparación del audio

Se parte de una biblioteca que contiene todas las canciones clasificadas por género en su directorio correspondiente, se ejecuta un script que recorre cada canción de cada carpeta, posteriormente se procesa el audio con la herramienta *FFmpeg* y se asigna a la canción la etiqueta del género de su directorio.

FFmpeg es un herramienta multiplataforma que permite grabar, convertir o retransmitir tanto audio como vídeo. En este proyecto se ha utilizado con la finalidad de segmentar las canciones de la biblioteca a partir de la cual se va a generar el conjunto de datos del sistema. La segmentación del audio se realiza fijando un punto inicial y la duración del segmento. Además se pueden incorporar diversas opciones que se aplicarán sobre el audio de salida.

```
ffmpeg -i song.mp3 -ss 150 -t 10 -aq 1 -ac 1 -ar 44100 cut.wav (4.1)
```

Figura 4.3: Ejemplo de comando *FFmpeg*

- *ss*: inicio
- *t*: duración
- *aq*: calidad de audio (*1=estandar*)
- *ac*: canales de audio (*1=mono*)
- *ar*: frecuencia de muestreo

Se ha realizado el corte del audio a partir del minuto 2:30 de cada canción, ya que la mayoría de las canciones hacen una incorporación progresiva de los instrumentos y es a partir de estos instantes cuando se reproduce el núcleo de la canción. Se exportan 10 segundos de grabación, desde el minuto 2:30 hasta el minuto 2:40, esta longitud de grabación es suficiente ya que al ser un tipo de música de estructuras repetitivas, basta con un pequeño segmento de audio en este intervalo como muestra de cada canción. Otro aspecto a tener en cuenta es el archivo de salida, se codifica en formato de onda *WAV*, en modo mono para evitar tener un vector de características de doble tamaño.

Capítulo 5

Preproceso

En la etapa de preproceso se parte de los segmentos de audio de 10 segundos representados en forma de vector, donde cada celda es la energía del sonido en ese instante determinado. Estos valores son números reales que oscilan entre -1 y 1, y dado que la frecuencia de muestreo es de 44100Hz, por cada segundo de audio hay 44100 valores de energía. Si multiplicamos la cantidad de valores de cada segundo por los 10s de cada segmento, se eleva a 441000 la dimensión del vector de características de cada audio. Esta dimensión es demasiado elevada para ser introducidos los datos en una red neuronal, por lo que en primer lugar se comprime el vector utilizando la técnica de PCA (*Análisis de Componentes Principales*), después se realiza un barajado de los datos para evitar estructuras por defecto y se particiona el corpus para extraer los datos que pertenecerán a la evaluación y al entrenamiento.

5.1. PCA

PCA es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos. Intuitivamente la técnica sirve para hallar las causas de la variabilidad de un conjunto de datos y ordenarlas por importancia. Técnicamente, PCA busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. El funcionamiento de esta técnica se basa en el cálculo de la descomposición en autovalores de la matriz de covarianza, normalmente tras centrar los datos en la media de cada atributo.

Se ha optado por utilizar PCA como método de compresión de datos por su alto nivel de compresión de matrices sin pérdida de información gracias

al uso de autovectores y autovalores. Además al ser todos los segmentos de audio uniformes, misma duración y frecuencia de muestreo, aplicar PCA a todos ellos mantiene esta uniformidad y por lo tanto la compresión afecta a todos los datos por igual.

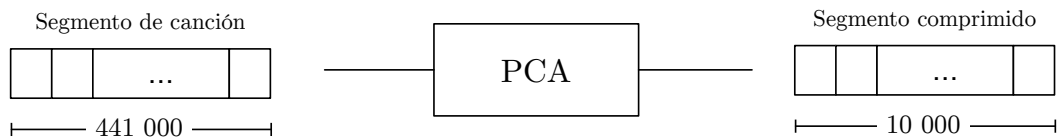


Figura 5.1: Compresión de segmentos mediante PCA.

5.2. Barajado de datos

Aplicar un barajado aleatorio a todo el conjunto de datos es aconsejable ya que es una forma simple de simular la aleatoriedad con la que llegan las nuevas muestras a clasificar en un sistema real. De esta forma se asegura que el posterior particionado del corpus sea representativo del conjunto total de datos, un claro ejemplo de romper estructuras previas es el almacenamiento secuencial por directorio de los audios, ya que todos los audios de la misma clase están contiguos en el conjunto total, si no se realizara el barajado, podría darse el caso de seleccionar para la evaluación datos de una sola clase.

5.3. Particionado del corpus

El proceso de particionado del corpus es sencillo, basta con elegir la proporción de datos que pasarán a formar parte de los datos de evaluación. En este caso el 10 % del conjunto total se reserva para la evaluación, teniendo en cuenta que el corpus cuenta con algo más de 1000 muestras, más o menos, 100 muestras se apartarían para test y las 900 restantes para entrenamiento.

Capítulo 6

Entrenamiento del sistema

Tras la construcción de un conjunto de datos adecuado al propósito del proyecto, es momento de comenzar con el entrenamiento del sistema, en primer lugar se expone la técnica de aprendizaje seleccionada así como la herramienta que la implementa, la cual ha servido como plataforma para la evaluación de resultados.

6.1. Redes Neuronales

Se ha optado por una red neuronal como método de aprendizaje. Una red neuronal consiste en una combinación de funciones log-lineales discriminantes agrupadas en capas, la ponderación automática mediante backtracking de los pesos entre capas hace que el nivel de abstracción se alto y por tanto la estrategia para abordar el aprendizaje de datos se reduzca a una buena configuración de la red.

Conceptualmente cada neurona representará una porción característica de un determinado género preestablecido y los pesos de las relaciones entre neuronas de capas diferentes harán referencia a la proporción de influencia de dicha porción característica en el género de la muestra que entra en la red. El aprendizaje se realiza mediante un proceso iterativo de backtracking por el cual la red modificará los pesos entre neuronas con el fin de clasificar de la mejor maneras todas las muestras de entrenamiento.

Un aspecto crucial para obtener buenos resultados mediante este sistema es una configuración inicial adecuada, en el capítulo 6.3 se entrará con detalle en cada parámetro ajustado.

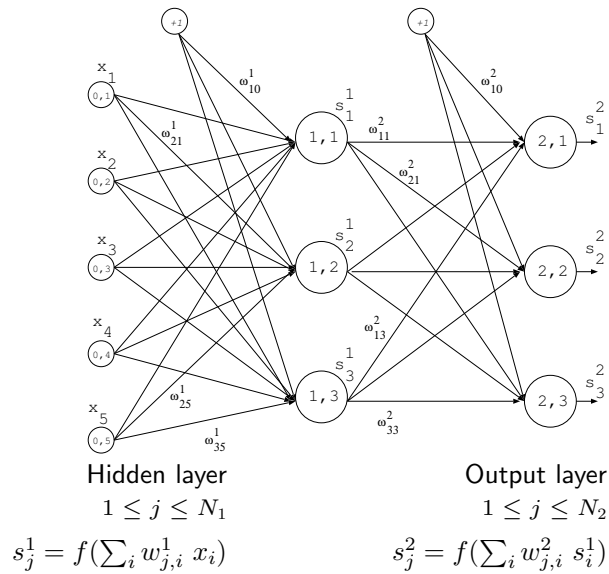


Figura 6.1: Ejemplo de red neuronal que clasifica una muestra en 3 posibles clases a partir de un vector de 5 características. [7]

6.2. DeepLearnToolbox for MatLab

DeepLearnToolbox [1] son una serie de scripts desarrollados para MatLab a modo de librería de los cuales su principal característica es que simplifican a unas pocas líneas los procesos de construcción y configuración de una red neuronal. A pesar de que este toolbox implementa redes neuronales más complejas como Deep Belief Networks (DBN) o redes neuronales convolucionales, se ha optado por utilizar una red neuronal básica y se ha apostado por la búsqueda de una configuración adecuada con la que puedan obtenerse buenos resultados.

```
nn = nnsetup([10000 75 8]);
```

```
nn.dropoutFraction = 0.5;
nn.weightPenaltyL2 = 1e-4;
nn.output = \'softmax\';
nn.learningRate = 0.001;
```

```
opts.numepochs = 100;
opts.batchsize = 5;
```

```
opts.plot = 0;

[nn, L] = nntrain(nn, train_x, train_y, opts);
[er, bad] = nntest(nn, test_x, test_y);
```

Figura 6.2: Configuración, entrenamiento y evaluación con *DeepLearnToolbox*.

6.3. Configuración

La parte importante del proceso de aprendizaje con este toolbox es la configuración de parámetros, para ello hay que saber el impacto que tiene cada parámetro con el resultado del aprendizaje.

- *output*: Determina el nivel de agresividad con el cual se activa la neurona de salida. Una función aconsejable es softmax, ya que suaviza la activación.
- *learning rate*: Establece la velocidad con la que el sistema aprende de los datos. Valores altos provocarán máximos locales, mientras que con valores muy bajos serán necesarias más iteraciones si se desea encontrar un máximo global.
- *numepochs*: Cantidad de iteraciones de backtracking, a mayor número de iteraciones más se aprende el sistema del conjunto de entrenamiento, hay que controlar el *overfitting*.
- *batchsize*: Número muestras que se utilizan para el mini-batch, pequeñas iteraciones por cada iteración de batch.

Capítulo 7

Evaluación

7.1. Experimentación

La experimentación del sistema se ha realizado mediante un script en el cual se han programando diferentes ejecuciones. En cada una de ellas se cargaban diferentes configuraciones de los parámetros vistos en el capítulo 6.3 para posteriormente entrenar y evaluar la red neuronal con cada configuración. Tras cada una de estas ejecuciones se han almacenado los parámetros utilizados y la tasa de error de evaluación obtenida. Posteriormente se han analizado los datos obtenidos de los cuales se ha extraído la comparativa de la siguiente sección.

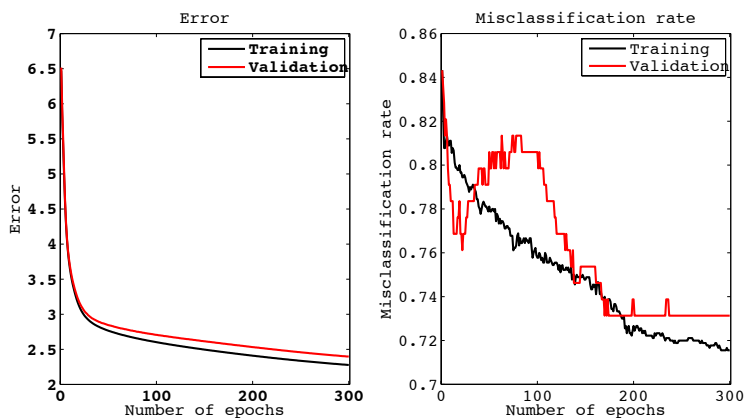


Figura 7.1: Ejemplo de la evolución del entrenamiento de una ejecución.

7.2. Comparativa

La comparativa se ha basado en los mejores resultados obtenidos en las baterías de pruebas programadas, tomando como referencia sus parámetros, se han realizado 5 ejecuciones con la configuración de cada mejor resultado reprocesando el barajado y particionado del corpus, para así manejar la tasa de error media.

red neuronal	learning rate	numepochs	batchsize	error
1 capa 125 neuronas	0.001	150	80	0.8090
1 capa 85 neuronas	0.001	300	20	0.7910
1 capa 75 neuronas	0.001	300	20	0.7141
1 capa 40 neuronas	0.001	400	20	0.7090

Cuadro 7.1: Comparativa de ejecuciones.

Las tasas de error obtenidas oscilan entre 0.9 y 0.7, por lo que el margen de mejora es muy justo. Como se puede observar, los mejores resultados se han obtenido con una tasa de aprendizaje muy baja, un minibach con una porción grande de muestras y muchas iteraciones, de esta forma los pesos de la red se adaptan muy lentamente al conjunto de entrenamiento y no se produce el peligroso overfitting. La mejor tasa de error media que se ha obtenido es de 0.7030, esto quiere decir que cuando se va a clasificar una muestra hay una probabilidad de acierto de la clase en un 30 % de los casos.

Capítulo 8

Conclusiones

De este proyecto de investigación es posible sacar varias conclusiones interesantes, a pesar de no haber conseguido un clasificador con una tasa de acierto a priori buena; hay que tener en cuenta las condiciones de las que se partía y cual era el objetivo del trabajo.

En primer lugar hay que destacar que un método aleatorio, en el caso concreto estudiado, tiene una probabilidad de acierto del 10% ya que hay 10 clases, por lo tanto conseguir que el sistema tenga una tasa de acierto 3 veces mayor a un método aleatorio es un buen dato. También hay que recordar que los datos de entrada al sistema son datos de audio en bruto comprimidos sin ningún tipo de filtro.

Otro aspecto a tener en cuenta es la heterogeneidad del conjunto de datos, para el objetivo que se pretendía alcanzar no era realista tratar con una colección de audio homogénea, donde cada clase tenga el mismo número de muestras y donde las clases estén bien definidas. Este último aspecto es a priori el que compromete la tasa de acierto del sistema, por ello en la siguiente sección se proponen una serie de indicaciones a seguir en futuros trabajos en esta línea de investigación.

Capítulo 9

Futuros trabajos

A partir del desarrollo de esta investigación, las técnicas aplicadas y los resultados obtenidos, es posible determinar un punto de partida desde el cual continuar la investigación en busca de mejorar los resultados.

Un aspecto bastante interesante a desarrollar es consultar para cada muestra clasificada automáticamente, la primera y la segunda clase más probable, comparar la diferencia de probabilidad y determinar un umbral que ayude a dar por válida la segunda clase si su probabilidad es muy cercana a la primera. Esta mejora se centra en resolver casos de solapamiento entre clases.

Otra cuestión a investigar es el sistema de aprendizaje automático a utilizar, probar otros tipos de redes neuronales como redes convolucionales aplicadas directamente sin PCA, técnicas de algoritmos genéticos para configurar automáticamente la iniciación de la red neuronal u otras técnicas de inteligencia artificial con el objetivo de adecuar el clasificador a los datos.

Bibliografía

- [1] Rasmus Berg. Deeplearntoolbox. <https://github.com/rasmusbergpalm/DeepLearnToolbox>, 2011.
- [2] Mei-Lan Chu. Automatic music genre classification. *Graduate Institute of Biomedical Electronics and Bioinformatic*, 2009.
- [3] Ichiro Fujinaga Cory McKay. Musical genre classification: Is it worth pursuing and how can it be improved? *International Symposium/Conference on Music Information Retrieval*, 2006.
- [4] Perry Cook George Tzanetakis, Georg Essl. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5), 2002.
- [5] Emilia Gómez Justin Salamon, Bruno Rocha. Musical genre classification using melody features extracted from polyphonic music signals. *Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain*, 2009.
- [6] Carlos Ortiz. Clasificador de música en subgéneros personalizados (dataset). <http://personales.alumno.upv.es/caroral1/dataset.zip>, 2015.
- [7] Roberto Paredes. Chapter 2. multilayer perceptron (i). <http://users.dsic.upv.es/rparedes/teaching/RNA/t2-1rna.pdf>, 2013.
- [8] Qi Li Tao Li, Mitsunori Ogihara. A comparative study on content-based music genre classification. *SIGIR'03*, 2003.