Clasificación de Música por Género utilizando Redes Neuronales Artificiales

Elkin García[1], Guillermo Pacheco[1] y Germán Mancera[2]

[1] Universidad de los Andes Carrera 1 N° 18A 10 Bogotá, Colombia
{elkin-ga, jor-pach}@uniandes.edu.co
[2] 140 Evans Hall University of Delaware Newark, DE 19711 USA
gmancera@ece.udel.edu

Resumen — En este proyecto se realiza un estudio de la información contenida en archivos digitales de música con el fin de determinar parámetros representativos de un género musical para ser usados como entradas a un sistema clasificador implementado con Redes Neuronales Artificiales. La extracción de parámetros se realiza haciendo uso de análisis en el dominio del tiempo, de la frecuencia, y utilizando la Transformada Wavelet. Se han implementado dos topologías diferentes de Red Neuronal: Perceptrón Multinivel y Red de Funciones de Base Radial (RBF), adicionalmente se implementa un clasificador utilizando el algoritmo Adaboost sobre un clasificador débil RBF. Los resultados que han sido obtenidos son comparados en cuanto porcentaje de clasificación correcta y tamaño de la red. De la misma manera, se ha estudiado la repercusión que tiene la aplicación de AdaBoost sobre este clasificador débil.

1 INTRODUCCIÓN

Bases de datos de música en formato digital cuyo tamaño era difícil imaginar hace unos años se comparten hoy en día con gran facilidad a través de las redes de computadores, creando la necesidad de disponer de métodos eficientes que permitan la búsqueda y organización de dichas bases. En los últimos años, y debido al creciente fenómeno de distribución de música a través de Internet, ha surgido la necesidad de crear sistemas capaces de clasificar música de manera automática. Sin embargo, son muchos los estudios que se han venido realizando con el fin de establecer cuáles son aquellas características que constituyen un estilo musical, qué parámetros musicales son relevantes para hacer la clasificación y cuáles son las técnicas de aprendizaje de máquina más efectivas para procesar tal información **[1],[2],[3].**

Los estudios que se han realizado al respecto son pocos. Seth Golub **[2]** realizó uno en la cual probó tres tipos diferentes de clasificadores aplicados a dos géneros similares. Paul Scott **[3]** elaboró un sistema clasificador usando un perceptrón multinivel para 4 géneros bastante diferentes. Numerosos investigadores han realizado trabajos relacionados con la identificación de características relevantes a partir de la música. Por ejemplo, Foote **[4]** empleó técnicas espectrales para distinguir entre voz y música con un alto grado de exactitud, mientras que Soltau **[5]** entrenó

una red neuronal auto–asociativa usando la técnica de Foote para los géneros rock, pop, tecno y clásico. Su razón de clasificación correcta fue similar a la obtenida por Golub. Este proyecto está enfocado hacia la clasificación de canciones en tres géneros bastantes populares a nivel colombiano y con características muy similares (Merengue, Salsa y Vallenato) con el fin de probar la robustez del sistema.

Este artículo comienza con una descripción de todos los desarrollos hechos durante la realización del proyecto en cuanto a extracción de características y algoritmos de entrenamiento. Se presentan las especificaciones del sistema en lo referente a la base de datos y finalmente se exponen los resultados obtenidos haciendo énfasis en las diferencias que se obtienen mediante la utilización del Perceptrón Multinivel, la Red de Funciones de Base Radial y AdaBoost.

2 DESARROLLOS

El sistema es capaz de clasificar canciones en formato MP3 y WAV, aunque realiza una conversión de MP3 a WAV mediante la utilización de una aplicación desarrollada por el grupo de investigación LAME **[12].** La base de datos consta de 500 canciones, 90% para entrenamiento y 10% para evaluación.

Las muestras que son tomadas de las canciones en formato WAV tienen una duración de 1.4861 segundos a partir de canciones cuya tasa de muestreo es 44.1 kHz es decir 2^{16} muestras (canción monofónica). Los parámetros extraídos pueden ser divididos en tres grupos: El primero de ellos se realiza por medio de un análisis en tiempo **[6]** que hace uso de la correlación. El segundo grupo se basa en un análisis en frecuencia que utiliza la Transformada de Fourier y se aplica a diferentes intervalos de frecuencia **[2],[3],[7]**. El último grupo se obtiene a partir de la extracción de dos aportes rítmicos independientes haciendo uso de la Transformada Wavelet y teniendo como función madre el sombrero mejicano definido por:

$$\psi(t) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)(1-x^2)\exp(-x^2/2) \quad . \qquad \textbf{(1)}$$

Estos parámetros surgen de la descomposición de la señal en 2 componentes ortogonales, denominadas patrones rítmicos. (Fig 1.)

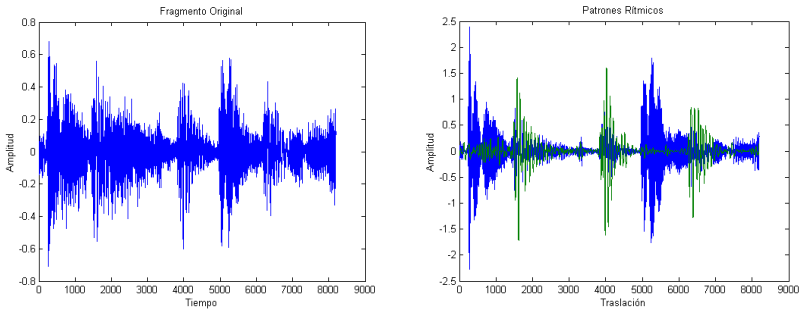

Fig. 1. Fragmento original antes de procesar *(Izq.)* y patrones rítmicos extraídos *(Der)*

Se extraen un total de 33 parámetros, 5 basado en análisis en tiempo, 5 basados en análisis en frecuencia y 23 a partir de la transformada Wavelet.
Mediante la utilización de la Transformación de Karhunen – Loéve **[8]** se busca descartar el contenido poco informativo que es usado como entrada a la Red Neuronal Artificial. Esto permite además reducir la complejidad del modelo ya que se reduce la dimensionalidad del espacio de entrada, dando como resultado, la selección de las 20 combinaciones lineales más representativas de parámetros iniciales.

2.1. Perceptrón multinivel y Red de funciones de base radial (RBF)

El perceptrón multinivel de una capa escondida se entrena utilizando propagación inversa del error (Backpropagation) a partir del siguiente pseudo–código:

Escoja el número de neuronas de la capa oculta.

Inicialice los pesos W.
Repita
Repita para cada uno de los datos X_i

Calcule la salida para la primera capa $G_j = f\left(\sum_{i=1}^{N} X_i W_{hji} + W_{hjB}\right)$

Calcule la salida total $F_j = f\left(\sum_{i=1}^{K} G_i W_{oji} + W_{ojB}\right)$

Fin.
Repita para cada uno de las salidas F_j *para todo j desde 1 hasta M*

Calcule el error como $\varepsilon_{oj} = (_j - F_j]F_j(- F_j)$

Fin

Hasta condición de terminación

La función de activación utilizada para todas las neuronas es la función sigmoide y se emplea la inicialización de pesos de Nguyen – Widrow **[9]** para optimizar la convergencia del entrenamiento. La salida de la red entrenada se calcula utilizando la distancia mínima euclidiana entre *F* y los vectores correspondientes a las posibles etiquetas T_p, donde p es el número de etiquetas posibles.

Por otra parte para RBF el pseudo–código del algoritmo de entrenamiento implementado es el siguiente:

Escoja el número de neuronas de la capa escondida.
Inicialice ω_i *(Centros de las Funciones de Base Radial)*
Repita
Repita para cada uno de los datos x_i

Asigne x_i *a* θ_i *(cluster) tal que* $\|x_i - \omega_i\|^2$ *sea la mínima entre las posibles.*

Fin.

Repita para cada uno de los clusters θ_i

$$\omega_i = \frac{1}{\|\theta_j\|} \sum_{j \in \theta_j} x_j$$

Fin.

Hasta que ninguno de los x_i *cambie de cluster.*

Encuentre la varianza de los datos mediante $\sigma_i^2 = \frac{1}{|\theta_j| - 1} \sum_{x \in \theta_j} x - \omega_i \ ^T \ x - \omega_i$

Encuentre los pesos de la capa de salida mediante $W^T = \Phi^\dagger T$

Con el objetivo de encontrar el orden del modelo adecuado para las redes del perceptrón multinivel y RBF se utiliza el proceso de 10 – Fold Cross Validation **[10]** para redes entre 2 y 20 neuronas.

Adicionalmente se introducen conocimiento previo pues es un factor fundamental en cualquier proceso de clasificación y se emplea con el fin de obtener desempeños superiores a aquellos que obtienen los clasificadores mediante solo el proceso de entrenamiento. Dado que las canciones de los géneros estudiados presentan irregularidades que se deben a arreglos musicales propios de cada intérprete, traducidas en espacios de silencio, improvisaciones, solos, cambios de velocidad entre otros y que la extracción de características se hace a partir de un fragmento muy corto, se compensa esta característica incorporando un sistema de votación 2 de 3, en el cual se evalúa la red en 3 fragmentos diferentes de la canción, reduciendo así la probabilidad de clasificación errónea.

2.3 AdaBoost.

AdaBoost se emplea como algoritmo de aprendizaje con el objetivo de disminuir el error de evaluación a medida que la red se sigue entrenando, incluso después de que el error de entrenamiento ha alcanzado el valor de cero **[11]**. Se escoge como clasificador débil una red de funciones de base radial de 11 neuronas puesto que presenta una buena relación entre el tiempo de entrenamiento y el porcentaje correcto de clasificación. El pseudo – código de Adaboost es el siguiente:

Se hace uniforme $D_t(i) = \frac{1}{m}$

Para t desde 1 hasta T

Se ejecuta el clasificador débil con entradas X, Y, D

Se calcula $\alpha_t = \frac{1}{2}\left(\log\left(\frac{1-\varepsilon_t}{\varepsilon_t} \right) \right) \in R$

Se actualiza $D_{t+1}(i) = \frac{D_t(i) e^{(-\alpha_t \beta_t(i))}}{Z_t}$ *donde* $\beta(i)$ *es un factor tal que*

$$\beta(i) = \begin{cases} -1 & si\ H_t(i) = Y(i) \\ 1 & si\ H_t(i) \neq Y(i) \end{cases}$$

Fin

Se calcula la hipótesis final $H(i) = p \quad tal\ que \sum_{t=1}^{T} \alpha_t g_t(i)$ *es máxima*

Puede notarse que en la primera iteración el algoritmo tenía igual probabilidad de equivocarse con cualquier canción, por lo cual la distribución inicial asigna igual probabilidad de escogencia a todas las muestras. Esta probabilidad es modificada de acuerdo al resultado obtenido por la red al momento de clasificar una muestra: Si ésta es clasificada correctamente su probabilidad disminuye; si no, su probabilidad aumenta. A partir de esta distribución se obtiene un conjunto de muestras bootstrap; lo que se desea con este tipo de conjunto, es que cada clasificador débil se entrene con un conjunto de muestras aleatorio, pero haciendo mayor énfasis en las canciones que no han sido clasificadas correctamente con el fin de llevar a cero el error de entrenamiento.

El coeficiente a_i entrega un grado de credibilidad para cada uno de los clasificadores débiles y además se utiliza para calcular la distribución de probabilidad, mientras que ε_i es el error en la iteración i. A partir de la suma de hipótesis parciales se genera la anterior hipótesis final, con la cual se calcula el error de entrenamiento y la distribución de márgenes para cada una de las clases, en diferentes épocas de entrenamiento.

3 RESULTADOS

El error promedio de clasificación, empleando 10 – Fold Cross Validation para el perceptrón multinivel en el conjunto de validación es bastante similar para los diferentes ordenes de la red, y no se presentan mejoras significativas al aumentar el número de neuronas de la capa oculta (Fig. 2.). El porcentaje promedio de clasificación correcta tiene su máximo para la red con 6 neuronas en la capa escondida, con un porcentaje de acierto del 72.25 %.

Por otra parte, utilizando la misma metodología para RBF se observa que el porcentaje promedio de clasificación aumenta a medida que se agregan neuronas a la capa escondida. El porcentaje promedio de clasificación correcta tiene su máximo para una red con 19 neuronas en la capa escondida, con un porcentaje promedio de acierto del 76.89 %.

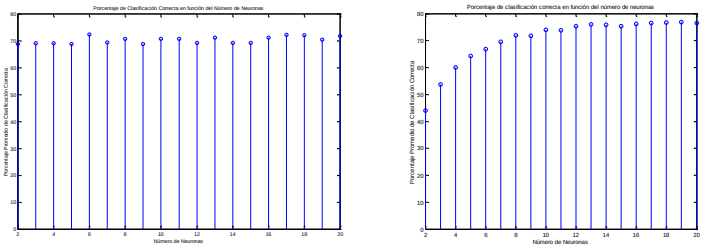

Fig. 2. Porcentaje Promedio de Clasificación Correcta obtenido empleando 10 – Fold Cross Validation en función del número de neuronas para perceptrón multinivel *(Izq.)* y RBF *(Der.).*

Para el clasificador Adaboost, mientras el error de entrenamiento baja hasta la iteración 10000 a un valor mínimo de 0.0022, el error de evaluación total también disminuye para los tres géneros (Fig. 3.), esta disminución también se corrobora por medio de los márgenes de confidencia (Fig. 4.). Un resumen de resultados para el clasificador Adaboost implementado se presenta en la Tabla 1

Tabla 1. Porcentajes de clasificación correcta utilizando Adaboost.

	Total	Merengue	Salsa	Vallenato
Porcentaje de Clasificación correcto (%)	80	70.59	88.24	81.25

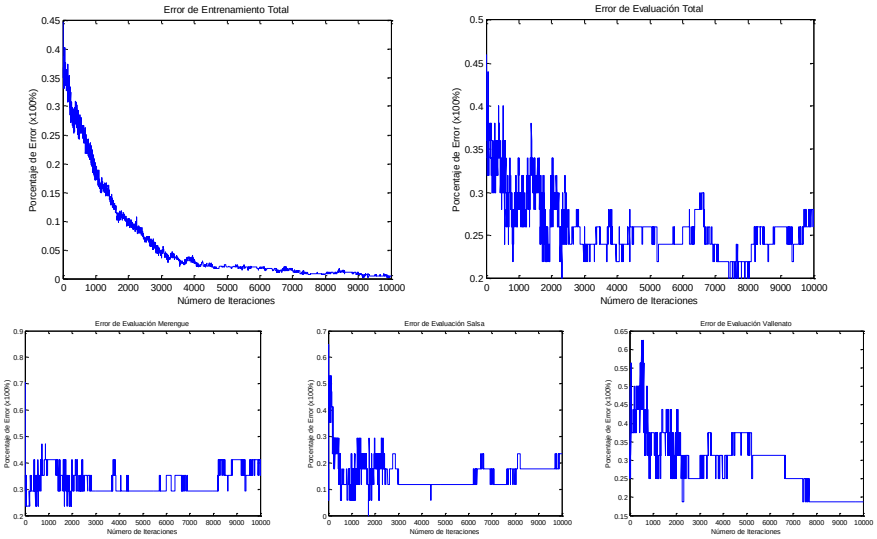

Fig. 3. Error de entrenamiento total *(Sup. Izq.)*. Error de evaluación a) Total *(Sup. Der.)*, b)Merengue *(Inf. Izq.)*, c)Salsa *(Inf. Cent.)* y d)Vallenato *(Inf. Der.)*.

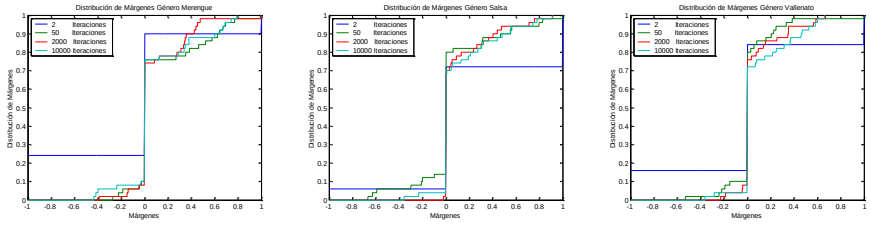

Fig. 4. Distribución de márgenes de evaluación para a) Merengue *(Izq.)* b) Salsa *(Cent.)* c) Vallenato *(Der.)*.

La Tabla 2 presenta un resumen detallado de todas las características obtenidas para las redes de Propagación Inversa y de Funciones de Base Radial.

Para la implementación de Adaboost, se utilizó una red de funciones de base radial con 11 neuronas en la capa escondida debido a que el tiempo requerido para su entrenamiento es bajo comparado con los demás tamaños de red. Por otra parte, fue posible cerciorarse de la forma en la cual se mejora el desempeño de un clasificador débil mediante la observación del porcentaje obtenido al realizar las 10000 iteraciones.

Tabla 2. Comparación de los resultados obtenidos para las topologías Red de Propagación Inversa y Red de Funciones de Base Radial.

	RED DE PROPAGACIÓN INVERSA	RED DE FUNCIONES DE BASE RADIAL
Tiempo total empleado en la búsqueda de la Red Óptima.	101.8357 h	20.84 h
Núm. de Neuronas Óptimo para la Capa Oculta.	6	16
Tiempo total empleado en el entrenamiento de la Red Óptima	173.8836 min	215.56 min
Núm. de Iteraciones realizadas en la Validación Cruzada.	200	2500
% Promedio de Clasificación en la Validación Cruzada	70.67	76.22
Máximo % de Clasificación en la Validación Cruzada	75.56	80
Porcentaje Correcto de Evaluación	76	80
Tiempo escogido para realizar la Evaluación	60 s	60 s
Porcentaje Correcto de Evaluación empleando Votación 2 de 3	78	84
Tiempos escogidos para realizar la evaluación con Votación	60, 75 y 90 s	45, 60 y 75 s

Adicionalmente al hecho que los errores de entrenamiento y evaluación disminuyan conforme aumenta el número de iteraciones, se observa que el porcentaje de acierto obtenido en la evaluación, utilizando Adaboost, es superior a este mismo porcentaje para la mejor red que se obtiene usando 10 – Fold Cross Validation, como se muestra en la Tabla 3.

Tabla 3. Resultado de la aplicación de Adaboost a un clasificador débil.

	Red de Funciones de Base Radial de 11 neuronas en la Capa Escondida	Clasificador Adaboost con 10000 iteraciones
% de acierto en la Evaluación	70	80

4.5 CONCLUSIONES

El estudio inicial realizado, acerca de la obtención y extracción de parámetros relevantes, a partir de un archivo de música en formato digital, muestra que las características utilizadas son apropiadas y permiten diferenciar entre los tres tipos de géneros que forman parte de este estudio. Los resultados arrojados por los clasificadores implementados corroboran la afirmación anterior, ya que de no ser así, la categorización de canciones no hubiese mostrado los resultados satisfactorios que fueron obtenidos.

Los parámetros extraídos a partir de la Transformada Wavelet tienen una estrecha relación con la forma en la cual el ser humano realiza el proceso de distinción de música por género. Esto es, que se centran en la identificación de un patrón rítmico básico presente en la música.

Es de destacar, que los parámetros que son extraídos buscando un patrón rítmico, son producto de la experimentación y la realización de numerosas pruebas, en las cuales se buscó representar esta característica. Esto constituye un aporte importante al campo de la Clasificación de Señales de Audio.

El proyecto obtuvo como resultado dos tipos básicos de clasificadores, que fueron producto de sucesivos estudios, en los cuales se optimizaron las respectivas arquitecturas de red, buscando aquellas para las cuales se presentaba el mejor desempeño, medido en términos de la clasificación correcta de canciones que no fueron usadas durante la fase de entrenamiento.

Las mejoras introducidas durante la fase de desarrollo fueron las siguientes:

- Extracción de Características Relevantes a partir de la transformación del algoritmo de Karhunen – Loéve.
- 10 – Fold Cross Validation, con el fin de determinar el orden apropiado del modelo.
- Incorporación de Conocimiento previo para inferir el intervalo de tiempo adecuado para tomar la muestra de la canción a clasificar.
- Votación 2 de 3, con el fin de disminuir la probabilidad de error debida a la escogencia de un intervalo de tiempo no representativo de la estructura musical.

La última y más relevante mejora implementada fue el uso de Adaboost aplicado al problema específico propio de este trabajo. Se demostró que su utilización eleva el porcentaje de acierto en la clasificación para un clasificador débil con 3 salidas.

Los resultados producidos en este proyecto son bastante buenos en comparación con los que han sido obtenidos por otros investigadores que han trabajado en el mismo campo, destacando la similaridad de los géneros utilizados. Por ejemplo Seth Golub **[2]** obtuvo un porcentaje de clasificación correcto del 77% utilizando redes de propagación inversa para dos géneros muy similares, mientras que Paul Scott **[3]** logró un porcentaje de clasificación de 94.8% utilizando como entradas, géneros muy diferentes como Rock, Clásica, Soul y Country. Otros trabajos como el de Soltau **[5]** obtuvieron 81.9% para Rock, Pop, Tecno y Clásica, mientras Tzanetakis **[7]** utilizó seis géneros y obtuvo un porcentaje de clasificación de 60%. Estos porcentajes son inferiores para aquellos estudios que utilizaron ritmos similares y comparables con los estudios que usaron ritmos diferentes. A pesar de la similitud existente en los géneros utilizados en este proyecto, se realizó un proceso de clasificación que contribuye al

desarrollo de la redes neuronales artificiales aplicadas no solo a la clasificación de música sino a la clasificación de señales de audio.

REFERENCIAS

[1] Gerhard, David. Ph.D. Depth Paper: Audio Signal Classification. School of Computing Science. Simon Fraser University. 2000.

[2] Golub, Seth. Classifying Recorded Music. MSc in Artificial Intelligence. Division of Informatics. University of Edinburgh. 2000.

[3] Scott, Paul. Music Classification using Neural Networks. EE373B Proyect. Stanford University. 2001.

[4] Foote, J.. A similarity measure for automatic audio classification. Technical report, Institute of Systems Science, National University of Singapure. 1997

[5] Soltau, H., Schultz, T., Westphal, M., y Waibel, A.. Recognition of music types. Interactive System Laboratory. 1998.

[6] Desain, P. Autocorrelation and the study of musical expression. Centre of knowledge technology , Utrecht School of arts. 1990

[7] Tzanetakis, George; Essl, Georg; Cook, Perry. Automatic Musical Genre Classification of Audio Signals. Computer Science Department. Princeton University. 2001

[8] Fukunaga, K. Introdutcion to Statistical Pattern Recognition. 2da Edición. SanDiego Academic Press. 1990

[9] Matlab Help. The MathWorks, Inc. Versión 6.5.

[10] Bishop, Christopher. Neural Networks for Pattern Recognition. New York. Oxford University Press Inc. 1995.

[11] R.E. Schapire. Theoretical views of boosting. In Computational Learning Theory: Fourth European Conference, EuroCOLT'99, 1999.

[12] www.lame.org Grupo de investigación dedicado a desarrollar algoritmos para descomprimir MP3.