Redes neuronales dinámicas aplicadas a la recomendación musical optimizada

Laura Elena Gómez Sánchez, Humberto Sossa Azuela, Ricardo Barrón,
Francisco Cuevas y Julio F. Jimenez Vielma

***Resumen*—En este trabajo se presenta un método basado en la operación de las llamadas redes neuronales dinámicas (RND), para la recomendación musical optimizada. Las redes son entrenadas con las señales de cada melodía, y no con descriptores tradicionales. La propuesta fue probada con una base de datos compuesta por 1,000 melodías, a diferentes frecuencias de muestreo.**

***Palabras clave*—Recuperación de información musical, red neuronal dinámica, descriptor musical.**

Dynamic Neural Networks Applied to Optimized Music Recommendation

Abstract—A method based on the operation of so called dynamic neural networks (DNN) for music recommendation is described. DNNs are trained with the signals of each melody and not with traditional descriptors. The method has been tested with a database composed of 1.200 melodies, at different sampling frequencies.

***Index Terms*—Music information retrieval, dynamic neural networks, musical descriptor.**

I. Introducción

La Recuperación de Información Musical (MIR por sus siglas en inglés) ha sido definida por Stephen Downie como la "investigación multidisciplinaria que se esfuerza por desarrollar sistemas innovadores de búsqueda basados en el contenido, interfaces novedosas, y mecanismos para que el vasto mundo de la música esté al alcance de todos". Dado al gran interés en esta área de investigación, y a los costos elevados de las bases de datos de música, la mayoría de los investigadores se han visto en la necesidad de crear y utilizar sus propias bases de datos que no se encuentran en la literatura, esto debido a los derechos de autor.

El área de recuperación de información musical se organiza de acuerdo a los casos según cada tipo de consulta, de acuerdo a la forma de comparar la entrada con la salida. Las consultas y la salida puede ser información textual (metadatos), fragmentos de música, grabaciones, partituras o características de la música. Ésta se divide en tres áreas principales de estudio:

- **Análisis simbólico**. Se refiere a la recuperación de información musical a través de partituras digitales [2], [3], [4], [5] y [6].
- **Metadatos**. Tiene que ver con la recuperación de información musical usando metadatos [7], [8], [9] y [10].
- **Análisis de señales acústicas**. Tiene que ver con la recuperación de información musical mediante señales sonoras musicales [11], [12], [13], [14], [15], [16] y [17].

El contorno melódico se utiliza para representar las melodías, característica principal que se utiliza en [18] y [19], la melodía se transforma en una secuencia U, D, R que representan la nota superior, inferior o igual a la nota anterior, respectivamente. Este enfoque simplifica demasiado la melodía que no puede discriminar correctamente entre otras melodías, sobre todo cuando se tiene una gran cantidad de datos. En [20] no solo usa el contorno melódico, también agrega el uso del intervalo del tono y el ritmo. Posteriormente en [21] se introducen cuatro tipos básicos de segmento (A, B, C, D) para el modelo del contorno musical. En ese mismo año en [22] se utiliza un nuevo indicador entre la consulta y las canciones que se proponen.

Varias técnicas de búsqueda basadas en metadatos se han realizado con el modelo de espacio vectorial, el modelo Booleano, indexación, invertir el archivo de índice, medida del coseno, etc. [23], [24] y [25]. En el área de recuperación de información, existen técnicas de indexación y agrupamiento aplicadas al manejo de recuperación musical [26].

CompariSong, en primer lugar, convierte el archivo de audio en segmentos, 10 segmentos por segundo se extraen en base a la frecuencia fundamental y la energía, la serie de tiempo se convierte en letras, para encontrar la correspondencia entre la consulta y la base de datos se utiliza la distancia Levenshtein [27].

Manuscrito recibido el 28 de mayo de 2012; aceptado para la publicación el 5 de junio del 2012.

Laura Elena Gómez Sánchez, Humberto Sossa Azuela, Ricardo Barrón y Julio F. Jimenez Vielma pertenecen al Centro de Investigación en Computación-IPN, Unidad Profesional Adolfo-López Mateos, Av. Juan de Dios Batiz s/n y M. Othon de Mendizábal, Zacatenco, México, DF. 07738, México (e-mail: lenis45@hotmail.com, hsossa@cic.ipn. mx, rbarron@cic.ipn.mx, jfvielma@cio.mx).

Francisco Cuevas pertenece al Centro de Investigaciones en Óptica A.C. Loma del Bosque #115, Col. Lomas del Campestre C.P. 37150, León, Gto. México (e-mail: fjcuevas@cio.mx).

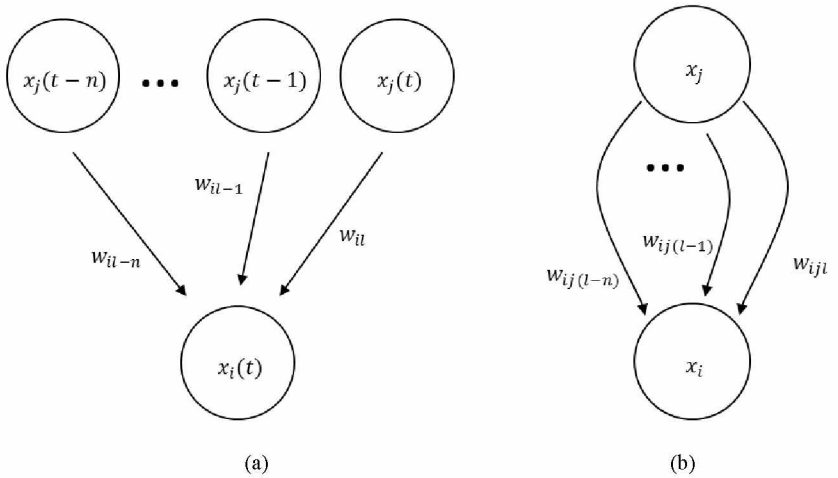

Fig. 1. Conexión de dos neuronas con retardo temporal

La implementación de redes neuronales, tales como las redes de Hopfield, Jordan y Elman, han sido ampliamente utilizadas, al igual que las técnicas de minería de datos dado que el manejo de metadatos es menor.

Existe un sistema híbrido de búsqueda por metadatos y consulta por tarareo, en el cual se realiza un tipo de filtro a través de una búsqueda racional de metadatos y la segunda mediante el tarareo para realizar un control de archivos por consulta.

Midomi es un sitio web comercial desarrollado por la corporación **Melodis** en 2006 [29]. Midomi trabaja en consultas mediante tarareo, canto y silbido, también proporciona una búsqueda avanzada sobre el género y lenguaje, tiene una base de datos musical de más de dos millones de melodías.

SoundHound es otro proyecto de la misma corporación especialmente diseñado para teléfonos móviles e iPods. Funciona igual que Midomi, por medio de tarareo, canto y silbido. Actualmente está diseñado para iPhone, iPod touch, iPod y teléfonos móviles android [30].

Musipedia es un proyecto de código abierto haciendo hincapié en la recuperación de melodías de manera muy diversa [31]. Este sistema proporciona facilidad de búsqueda de melodías en tres diferentes categorías, mediante transcripción melódica, transcripción del contorno melódico y la base rítmica. Soporta consultas por tarareo, silbido, canto, contornos y por golpeteo.

En este artículo se describe un método para la recomendación musical basada en el uso de redes neuronales dinámicas (RND). El método trabaja con diferentes frecuencias de muestreo en formato WAV. La base de datos cuenta con dos conjuntos de melodías, uno para entrenamiento y el otro para las consultas. Una vez que el conjunto de melodías es usado para entrenar las RND, se utilizan los pesos sinápticos de las redes como descriptores para la recomendación musical.

El resto del trabajo se organiza como sigue. En la sección 2 se describe como las redes neuronales de retardo temporal (TDNN por sus siglas en inglés) pueden ser usadas para la recomendación musical. En la sección 3 se detalla el trabajo realizado. En la sección 4 se muestran los experimentos y resultados obtenidos, respectivamente. Finalmente, en la sección 5 se dan las conclusiones de este trabajo.

I. Recuperación de Información Musical Usando TDNN

A. Redes neuronales de retardo temporal (TDNN)

La arquitectura TDNN fue desarrollada en [32]. Esta arquitectura se diseñó originalmente para el procesamiento de patrones de secuencias de voz en series de tiempo con desplazamientos.

Cuando se usan redes multicapa para el tratamiento de secuencias, se suele aplicar una idea muy simple: la entrada de la red se compone no sólo del valor de la secuencia en un determinado instante, sino por valores en instantes anteriores. Es como alimentar la red con una ventana temporal.

La idea de introducir el estado de una variable en diversos instantes en la red no sólo se puede aplicar a la entrada, sino también a las activaciones de las neuronas. Una red donde las activaciones de algunas neuronas son simplemente una copia de las activaciones de otra en instantes anteriores de la denominada Red Neuronal con Retardo Temporal o Time-Delay Neural Network (TDNN) [33] y [34].

Las neuronas con las que se trabaja en redes multicapa con retardo temporal responden a la ecuación:

$$x_i = f_i\left(\sum_j w_{ij}, x_j\right). \quad (1)$$

Como se observa, no existe una dependencia temporal, y la propagación o cálculo de las activaciones se realiza desde la capa superior a la inferior como en cualquier red multicapa.

En estas redes un paso de tiempo hay que entenderlo como iteración. La conexión entre la neurona *j* y la neurona *i*, con retardos temporales, se realizará como:

$$x_i = f_i\left(\sum_l w_{il}, x_l\right), \tag{2}$$

$$x_l = x_j(t - t_l), \tag{3}$$

donde t significa iteración y *tl* es el retardo temporal. Las neuronas *xl* son simplemente copias de la activación de *xj* en instantes o iteraciones anteriores. Se puede dar otra interpretación que consiste en asignar a los pesos distintas velocidades de conexión, siendo unos más lentos que otros, con lo cual en vez de tener una capa con varias neuronas conteniendo copias de las activaciones de la *j* tendríamos sólo la neurona *j* pero conectada a la *i* con varios pesos de distinta velocidad. La ecuación anterior se transformaría en:

$$x_i = f_i(\sum_j \sum_k w_{ijk}, x_j). \tag{4}$$

donde w_{ijk} correspondería al peso que conecta la neurona *j* con la *i* con retardo o velocidad *k*. En la figura 1 se visualiza la conexión entre dos neuronas para las dos interpretaciones. Esta última interpretación tiene un a gran importancia ya que es bien sabido que existen retrasos temporales significativos en los axones y sinapsis de las redes de neuronas biológicas.

B. Algoritmo Levenberg-Marquardt

Este algoritmo fue diseñado para encontrar las raíces de funciones formadas por la suma de los cuadrados de funciones no lineales, siendo el aprendizaje de redes neuronales, una aplicación especial de este algoritmo. El algoritmo de Levenberg Marquardt es una variación del método de iterativo de Newton para encontrar las raíces de una función.

El algoritmo de Levenberg Marquardt puede aplicarse en cualquier problema donde se necesite encontrar los valores de las raíces de una función; en el caso de las redes neuronales artificiales, la función es el error cuadrático medio de las salidas de la red y las raíces de esta función son los valores correctos de los pesos sinápticos.

En la ecuación (5), se presenta como se localiza un valor mínimo ($x_{\min}$) de una función de una variable $f(x)$, utilizando la primera y segunda derivada de acuerdo al método de Newton:

$$x_{\min}(t+1) = x_{\min}(t) - \frac{f'(x_{\min}(t))}{f''(x_{\min}(t))}. \tag{5}$$

Con base en esta ecuación se puede inferir la ecuación (6), donde se minimice el error global E_p en el espacio de los pesos sinápticos representado por la matriz W:

$$W(t+1) = W(t) - \frac{E_p'}{E_p''}. \tag{6}$$

La segunda derivada del error global (E_p'') corresponde a la matriz Hessiana H y la primera derivada (E_p') la conocemos como el vector gradiente G. El vector gradiente y la matriz Hessiana de la función de error los podemos calcular utilizando la regla de la cadena. Así, el vector gradiente se compone por las derivadas parciales del error con respecto a cada uno de los pesos w_i de la red, el elemento (i,j) de la matriz Hessiana se calcula con las segundas derivadas parciales del error con respecto a los pesos w_i y w_j.

Debido a la carga computacional que implica calcular de manera exacta la matriz H, se hace una estimación de la misma [35]. Debido a esto, en (6) se introduce un mecanismo de control para evitar los problemas que se puedan tener en la actualización de pesos de la red, dando origen a (7):

$$W(t+1) = W(t) - (H + \lambda I)^{-1} G. \tag{7}$$

El mecanismo de control para garantizar la convergencia del algoritmo consiste en introducir un factor λI. En primer lugar se prueba la ecuación del método de Newton. Si al evaluarla, el algoritmo no converge (el valor del error comienza a crecer), se elimina este valor y se incrementa el valor de λ en (7), con el fin de minimizar el efecto de la matriz H en la actualización de pesos. Si λ es muy grande, el efecto de la matriz H prácticamente desaparece y la actualización de pesos se hace esencialmente con el algoritmo de gradiente descendente. Si el algoritmo tiene una clara tendencia hacia la convergencia se disminuye el valor de λ con el fin de aumentar el efecto de la matriz H. De esta manera se garantiza que el algoritmo se comporta con un predominio del Método de Newton.

El método Levenberg Marquardt mezcla sutilmente el método de Newton y el método Gradiente Descendente en una única ecuación para estimar la actualización de los pesos de la red neuronal.

C. Audio digital

El audio digital es la representación de señales sonoras mediante un conjunto de datos binarios. Un sistema completo de audio digital comienza habitualmente con un transceptor (micrófono) que convierte la onda de presión que representa el sonido a una señal eléctrica analógica.

Esta señal analógica atraviesa un sistema de procesado analógico de señal, en el que se puede realizar limitaciones en frecuencia, ecualización, amplificación y otros procesos como el de compansión. La ecualización tiene como objetivo contrarrestar la particular respuesta en frecuencia del transceptor utilizado de forma que la señal analógica se asemeje mucho más a la señal audio originario.

Tras el procesado analógico la señal se muestrea, se cuantifica y se codifica. El muestreo toma un número discreto de valores de la señal analógica por segundo (tasa de muestreo) y la cuantificación asigna valores analógicos discretos a esas muestras, lo que supone una pérdida de

información (la señal ya no es la misma que la original). La codificación asigna una secuencia de bits a cada valor analógico discreto. La longitud de la secuencia de bits es función del número de niveles analógicos empleados en la cuantificación. La tasa de muestreo y el número de bits por muestra son dos de los parámetros fundamentales a elegir cuando se quiere procesar digitalmente una determinada señal de audio.

Los formatos de audio digital tratan de representar ese conjunto de muestras digitales (o una modificación) de las mismas de forma eficiente, tal que se optimice en función de la aplicación, o bien el volumen de los datos a almacenar o bien la capacidad de procesamiento necesaria para obtener las muestras de partida.

En este sentido hay un formato de audio muy extendido que no se considera de audio digital: el formato MIDI. MIDI no parte de muestras digitales del sonido, sino que almacena la descripción musical del sonido, siendo una representación de la partitura de los mismos.

El sistema de audio digital suele terminar con el proceso inverso al descrito. De la representación digital almacenada se obtienen el conjunto de muestras que representan. Estas muestras pasan por un proceso de conversión digital analógica proporcionando una señal analógica que tras un procesado (filtrado, amplificación, ecualización, etc.) inciden sobre el transceptor de salida (altavoz) que convierte la señal eléctrica a una onda de presión que representa el sonido.

Los parámetros básicos para describir la secuencia de muestras que representa el sonido son:

- El número de canales: 1 para mono, 2 para estéreo, 4 para el sonido cuadrafónico, etc.
- Frecuencia de muestreo: El número de muestras tomadas por Segundo en cada canal.
- Número de bits por muestra: Habitualmente 8 ó 16 bits.

Como regla general, las muestras de audio multicanal suelen organizarse en tramas. Una trama es una secuencia de tantas muestras como canales, correspondiendo cada una a un canal. En este sentido el número de muestras por segundo coincide con el número de tramas por segundo. En estéreo, el canal izquierdo suele ser el primero.

La calidad del audio digital depende fuertemente de los parámetros con los que esa señal de sonido ha sido adquirida, pero no son los únicos parámetros importantes para determinar la calidad. Una forma de estimar la calidad del sonido digital es analizar la señal diferencia entre el sonido original y el sonido reproducido a partir de su representación digital.

De acuerdo a lo anterior se puede hablar de una relación señal a ruido. Para los sistemas de audio que realicen compresiones digitales tipo *lossless*, esta medida va a estar determinada por el número de bits por muestra y la tasa de muestreo.

El número de bits por muestra determina un número de niveles de cuantificación y éstos una relación señal a ruido de pico de portadora que depende de forma cuadrática del número de bits por muestra para el caso de la cuantificación uniforme. La tasa de muestreo establece una cota superior para las componentes espectrales que pueden representarse, pudiendo aparecer distorsión lineal en la señal de salida y *aliasing* (o solapamiento de espectros) si el filtrado de la señal no es el adecuado. Para los sistemas digitales con otro tipo de compresión la relación señal a ruido puede indicar valores muy pequeños aunque las señales sean idénticas para el oído humano.

La frecuencia de muestreo es un número que indica la cantidad de muestras que se toman en determinado intervalo de tiempo, la resolución o profundidad del sonido es un número que indica cuántos bits (dígitos binarios, ceros y unos) se utilizan para representar cada muestra. Tanto la frecuencia como la resolución están directamente relacionadas con la calidad del sonido digital almacenado. Mientras mayores sean estos indicadores, más parecida será la calidad del sonido digitalizado con respecto al real. El estándar definido cuando se crearon los discos compactos de audio especifica que el sonido digita almacenado en ellos debe poseer una frecuencia de 44 100 KHz y 16 bits estéreo. Esto significa que se deben tomar unas 44 100 muestras por segundo, cada una se representará con 16 bits, y en dos canales independientes (sonido estéreo).

II. Trabajo realizado en MIR

De acuerdo a la literatura el análisis espectral de la señal en el dominio de la frecuencia ha brindado mejores resultados que las técnicas enfocadas al análisis de la misma en el dominio del tiempo. Algunas de las características como el tono, duración, ritmo, correlación cruzada, FFT entre otros están ampliamente ligados a la firma digital. Sin embargo, utilizar directamente la información contenida en la melodía sin hacer uso de firmas digitales o de funciones tradicionales, se evita el pre procesamiento de la información que estas requieren. En este trabajo se utiliza la melodía original para realizar la recuperación de información musical. No se ha realizado ningún cambio o tratamiento previo a las melodías para tratar de adaptarlas a un modelo tradicional, se introducen directamente a la TDNN, dicho procesamiento se muestra en la figura 2.

Tenemos una base de datos de melodías de los Beatles en formato WAV, se cuenta con un conjunto de entrenamiento y otro de prueba. Cada melodía es entrenada en una red neuronal de retardo temporal (TDNN_(i), donde i es la melodía, al terminar el entrenamiento se obtiene una matriz de pesos $WNN\_(i)$.

De cada melodía que se almacena en la base de datos, se obtiene un vector de datos que puede ser de longitud variable. El número de retardos es igual al número de neuronas en la

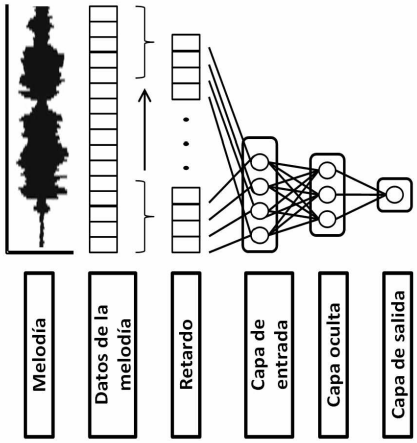

Fig. 2. Estructura de una Red Neuronal con Retardo Temporal

capa de entrada para el primer bloque de datos, en otras palabras se hace un ventaneo del vector. Cada una de las ventanas es la siguiente entrada de la red. Dicho proceso se aplica en toda la melodía.

La matriz obtenida es nuestro descriptor musical, descartando por completo cualquier descriptor tradicional. Esto se puede observar en la figura 3.

Para la recuperación de una melodía se introduce un segmento consulta, dicho segmento entra a una red neuronal de retardo temporal con los pesos sinápticos previamente entrenados, obteniendo los errores de recuperación por cada red (Re_(i)).

Este error se genera a partir de la comparación del segmento consulta en relación con la señal estimada o la predicción de la señal obtenida desde la red. Estos errores son almacenados en un vector, finalmente se aplica $\arg\min()$ retornando un índice (n^*), indicando que red neuronal tuvo el menor error. Este procedimiento se ilustra en la figura 4.

El error de recuperación está dado por:

$$\text{Re\_}i = \frac{\sum_{j=1}^{N}\left(x_j - y_j\right)^2}{w}. \qquad (8)$$

donde x_i son las matrices de pesos previamente entrenados, y_i es el segmento a reconocer, y w es el número de ventanas en las que el segmento fue dividido.

III. Experimentos y resultados

La topología de la TDNN seleccionada consistió de tres capas: una de entrada (número de neuronas es igual al número de retardos de la red), una oculta y una capa de salida (que corresponde a la predicción, ya que es una neurona). Este modelo de red fue programado en Matlab y los datos de entrada no han sido normalizados.

El algoritmo utilizado para el entrenamiento es el backpropagation. Este procedimiento ajusta iterativamente todos los pesos de la red con el fin de disminuir el error obtenido en la unidad de salida, utilizando el algoritmo Levengerg-Marquardt como función de activación.

Los pesos de conexión están inicializados aleatoriamente a [0:3]. Por razones de la velocidad de convergencia de todas las muestras entrenadas que se presentan una vez que los pesos se actualizan.

Se usan archivos en formato WAV de 16 bits (en modo estéreo, para entrenamiento o recuperación solo se utiliza un canal para evitar el *over-processing*, recordando que la información de un canal es copia fiel del otro. La base de datos para esta experimentación es de 1000 melodías (Beatles y Elvis Presley).

La configuración usada para el entrenamiento de las melodías fue de 5 y 10 neuronas en la capa oculta (debido a que en pruebas anteriores son las que mejores resultados han dado) y las iteraciones fueron de 15, 25 y 35, con un retardo de 10, recordando que el tamaño del retardo es igual al número de neuronas en la capa de entrada, obteniendo una predicción de datos en la salida. Al terminar el entrenamiento de cada melodía se obtiene su matriz de pesos, las cuales se utilizarán como descriptores.

Se realizaron pruebas con diferentes tipos de frecuencia de muestreo en formato WAV, utilizando la base de datos mencionada anteriormente. Para estos experimentos se cuenta con un conjunto de entrenamiento y un conjunto diferente para las consultas, cabe destacar que algunas melodías cuentan con 2 o 3 versiones diferentes, lo que permite analizar el desempeño de nuestra propuesta. La tabla I resume las características principales de los datos utilizados.

Como se puede observar la ventana de consulta tiene diferentes valores, ya que a menor frecuencia se cuenta con menos información para realizar un reconocimiento con un segmento pequeño de consulta. Las tablas II y III muestran el rendimiento obtenido en cada configuración utilizada. La tabla II muestra el tamaño necesario de la ventana consulta para una recuperación perfecta. La tabla III muestra el tamaño mínimo para obtener una recomendación de 10 melodías para el usuario. El porcentaje de recuperación por recomendación de melodías puede observarse en la Tabla IV.

IV. Conclusiones

En este trabajo se describió como las llamadas redes neuronales con retrasos pueden ser usadas para describir el contenido musical de melodías para su posterior recuperación, basándose en partes de dichas melodías. Mediante las TDNN se logró resolver el problema planteado sin necesidad de realizar un pre-procesamiento, evitando así el utilizar algún descriptor tradicional o firma digital de la melodía.

A diferencia de otras técnicas de MIR, la melodía original se puede considerar como una serie temporal que se introduce directamente en la TDNN, la salida de la red codifica una descripción de la melodía en la matriz de pesos. Por ejemplo, si se entrena una TDNN con una melodía de 7 938 000 de

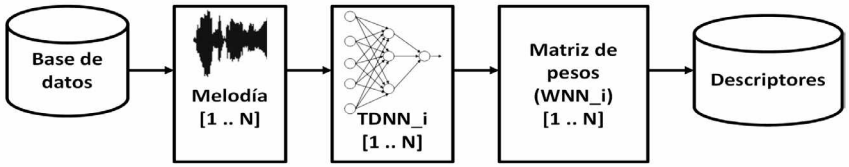

Fig. 3. Estructura de entrenamiento de las melodías con TDNN

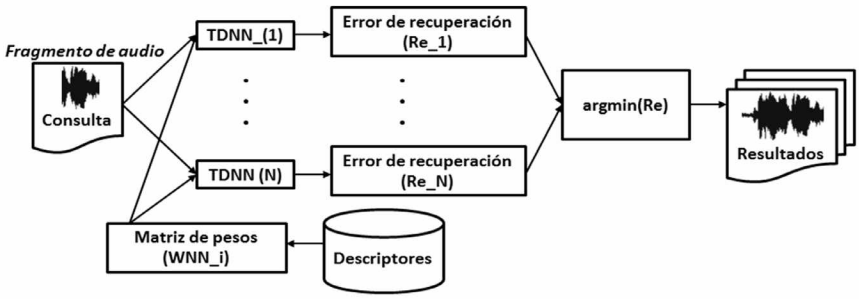

Fig. 4. Procedimiento de recuperación de una melodía usando el modelo propuesto

TABLA I
RASGOS CONSIDERADOS PARA EL AUDIO DIGITAL (WAV)

Tasa de muestreo (KHz)	Rango de la ventana de consulta (Datos)
22 050	25 000 – 60 000
24 000	20 000 – 54 000
32 000	15 000 – 45 000
44 100	10 000 – 40 000

TABLA II
TABLA DE RECUPERACIÓN PERFECTA

Tasa de muestreo (KHz):		22 050	24 000	32 000	44 100
Neuronas en la capa oculta	Iteraciones	Rango de la ventana de consulta (Datos)			
5	15	58 000	54 000	32 000	32 000
	25	41 000	40 000	35 000	29 000
	35	48 000	43 000	38 000	18 000
10	15	59 000	41 000	35 000	40 000
	25	43 000	57 000	39 000	31 000
	35	57 000	46 000	45 000	25 000

TABLA III
TABLA DE RECOMENDACIÓN DE 10 MELODÍAS

Tasa de muestreo (KHz):		22 050	24 000	32 000	44 100
Neuronas en la capa oculta	Iteraciones	Rango de la ventana de consulta (Datos)			
5	15	42 000	38 000	27 000	25 000
	25	31 000	29 000	24 000	21 000
	35	35 000	31 000	31 000	15 000
10	15	44 000	36 000	25 000	32 000
	25	36 000	52 000	28 000	24 000
	35	51 000	42 000	37 000	19 000

TABLA IV
TABLA DE PROCENTAJES DE RECUPERACIÓN POR RECOMENDACIÓN

Tasa de muestreo (KHz):		22 050	24 000	32 000	44 100
Neuronas en la capa oculta	Iteraciones	Porcentajes %			
5	15	72	74	92	89
	25	79	79	85	92
	35	75	75	83	94
10	15	80	80	84	82
	25	77	77	87	86
	35	74	74	81	87

cuadros (aproximadamente 3 minutos de una melodía) con calidad de audio a 44 100 KHz para una recuperación perfecta, solo se necesitan máximo 40 000 cuadros, lo cual refleja menos del 1% del total de la melodía.

Con los resultados obtenidos en esta experimentación se ha observado que el sistema funciona muy bien incluso al trabajar con diferentes frecuencias de muestreo, los mejores porcentajes se obtuvieron con frecuencias de 32 000 y 44 100 KHz, debido a que se tiene una mejor calidad de audio, sin embargo las frecuencias restantes logran realizar una buena recomendación musical. En experimentaciones previas se usaba el conjunto de entrenamiento tanto para el entrenamiento como para la consulta.

En el presente trabajo se analizó el desempeño del método propuesto al usar versiones diferentes de las melodías aprendidas para la recuperación.

REFERENCIAS

[1] F. Wiering, "Can humans benefit from music information retrieval?," AMR'06, *Proc. of the 4th international conference on Adaptive Multimedia Retrieval: User, Context and Feedback*, Springer, 2007, p. 82–94.

[2] K. Lemstrom, G.A. Wiggins and D. Meredith, "A three-layer approach for music retrieval in large databases," *2nd International Symposium on Music Information Retrieval, Bloomington*, USA, 2001, p. 13–14.

[3] H. Hoashi and K. Matsumoto, "Personalization of user profiles for content-based music retrieval based on relevance feedback," *Proceeding of the eleventh ACM international conference on Multimedia*, New York, USA, 2003, p. 110–119.

[4] H. Zhuge, "An inexact model matching approach and its applications," *Journal of Systems and Software*, 67 (3), 2003, p. 201–212.

[5] E. Hwang and S. Rho, "FMF (fast melody finder): A web-based music retrieval system", *Lecture Notes in Computer Science* vol. 277. Springer, 2004, p. 179–192.

[6] E. Hwang and S. Rho, FMF: "Query adaptive melody retrieval system," *Journal of Systems and Software*, 79 (1), 2006, p. 43–56.

[7] N. Ali, M. Mshtaq, "Hybrid query by humming and metadata search system (HQMS) analysis over diverse features," *International Journal of Advanced Computer Science and Applications*, Vol. 2, No. 9, 2011, p. 58.

[8] P. W. Bertin-Mahieux, B. Whitman, P. Lamere, "The Million Song Dataset," *12th Conference of International Society for Music Information Retrieval (ISMIR 2011)*, 2011.

[9] C. McKay, D. Bainbridge, D., "A Musical Web Mining and Audio Feature Extraction Extension to the Greenstone Digital Library Software," *12th Conference of International Society for Music Information Retrieval (ISMIR 2011)*, 2011.

[10] M. Weigl And C. Guastavino, "User Studies in the Music Information Retrieval Literature," *12th Conference of International Society for Music Information Retrieval (ISMIR 2011)*, 2011.

[11] M. Ryynanen, A. Klapuri, "Transcription of the singing melody in polyphonic music", *ISMIR 2006*.

[12] F. Ren, D. B. Bracewell, "Advanced information retrieval," *Journal Electronic Notes in Theoretical Computer Science (ENTCS)* 225, 2009, p. 303–317.

[13] K. Dressler, "Audio melody extraction", late breaking at ISMIR 2010, *Proceedings International Society for Music Information Retrieval Conference (ISMIR 2010)*, Utrecht, Netherlands, 2010.

[14] P. V. Kranenburg, J. Garbers, A. Volk, F. Wiering, L. P. Grijp, R. C. Veltkamp, "Collaboration perspectives for folk song research and music information retrieval: The indispensable role of computational musicology," *Journal of Interdisciplinary Music Studies*, 4 (1), 2010, p. 17–43.

[15] J. Salamon, E. Gómez, "Melody extraction from polyphonic music audio," *Music Information Retrieval Evaluation eXchange (MIREX)*, Utrecht, The Netherlands, 2010.

[16] J. Salamon, E. Gómez, "Melody Extraction from Polyphonic Music: MIREX 2011", in *Music Information Retrieval Evaluation eXchange (MIREX) 2011*, extended abstract, Miami, USA, 2011.

[17] J. Salamon, E. Gómez, "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech and Language Processing*, 20(6), 2012, p. 1759–1770.

[18] A. Ghias, "Query By Humming-Musical Information Retrieval in an Audio Database," *Proc. of ACM Multimedia 95*, 1995, p 231–236.

[19] S. Blackburn, D. De Roure, "A Tool for Content Based Navigation of Music," *Proc. ACM Multimedia 98*, 1998, p. 361–368.

[20] R. J. McNab, "Towards the Digital Music Library: Tune Retrieval from Acoustic Input," *Proc. of Digital Libraries*, 1996, p. 11–18.

[21] A. L. P. Chen, M. Chang, J. Chen, "Query by Music Segments: An Efficient Approach for Song Retrieval," *Proc. of IEEE International Conference on Multimedia and Expo*, 2000.

[22] C. Francu and C.G. Nevill-Manning, "Distance metrics and indexing strategies for a digital library of popular music." *2000 IEEE International Conference on Multimedia and Expo, 2000, ICME 2000*, Vol. 2, IEEE, 2000, p. 889–892.

[23] N.C. Maddage, H. Li, and M.S. Kankanhalli, "Music structure based vector space retrieval," *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2006, p. 67–74.

[24] N. Kosugi, Y. Nishihara, S. Kon'ya, M. Yamamuro, and K. Kushima, "Music retrieval by humming-using similarity retrieval over high dimensional feature vector space," *1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, IEEE, 1999, p. 404–407.

[25] L. Chen and B.G. Hu, "An implementation of web based query by humming system. *2007 IEEE International Conference on Multimedia and Expo*, IEEE, 2007, p. 1467–1470.

[26] L. Lu, H. You, and H.J. Zhang, "A new approach to query by humming in music retrieval," *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2001.

[27] G. Dzhambazov, "Comparisong: Audio comparison engine," *International Book Series* Number 11, 2009, p 33.

[28] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: current directions and future challenges," *Proceedings of the IEEE*, 96(4), 2008, p. 668–696.

[29] Midomi, http://www.midomi.com.

[30] Soundhound. http://www.soundhound.com.

[31] Musipedia. musipedia. URL: http://www.musipedia.org/.

[32] A. Weibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, "Phenomena Recognition Using Time-delay Neural Networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3), 1989, 328–339.

[33] J. B. Hampshire & A. H. Waibel, "A Novel Objetive Function for Improved Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Transactions on Neural Networks*, 1, 1990, p. 216–228.

[34] K. Lang and G. Hinton, "A Time-Delay Neural Network Arquitecture for Speech Recognition," Carnegie Mellon University, *Tech. Reprt. CMU-CS-88-152*, 1988.

[35] T. Masters, "Advanced Algorithms for Neural Network: A C++ Sourcebook," John Wiley & Sons Inc, 1995.