

CAPITULO 4

RESULTADOS Y CONCLUSIONES

4.1 ENTRENAMIENTO REDES NEURONALES

El sistema S.R.P.M. cuenta con dos redes neuronales artificiales como parte de su módulo de reconocimiento. Cada clasificador neuronal cuenta con características distintas y su propósito es diferente. Los dos clasificadores neuronales serán denominados “red de notas” y “red de tonos” respectivamente de acuerdo con la función que tienen dentro de S.R.P.M.

Para que una red neuronal funcione, es decir sea capaz de reconocer patrones, primero debe entrenarse apropiadamente. Este proceso implica varias decisiones por parte de quien las diseña:

- **Determinar una condición de fin de entrenamiento.-** En el caso de ambas redes se determinó como condición de fin de entrenamiento un número determinado de barridas.
- **Determinar un valor como coeficiente de aprendizaje.-** Este valor se define arbitrariamente por el diseñador pero siempre se encuentra dentro del rango:

$$0 < \eta < 1 \quad (4.1)$$

Ambas redes se probaron con varios coeficientes de aprendizaje, sin embargo, los más representativos fueron los valores de 0.1 y 0.7 que son los reportados en la tabla 4.1.

- **Determinar el número de clases que conforman el conjunto de entrenamiento.-** Este parámetro varía de red a red. Para la red de notas se determinaron 14 clases. (Ver tabla 3.1) Por otro lado, para la red de tonos se definieron 17 clases. (Ver fig. 3.20) El tamaño del conjunto de entrenamiento determina el número de neuronas que tendrá la capa de salida de una red neuronal.
- **Definir la configuración de la red neuronal.-** Si se da por hecho que el número de neuronas que existen en la capa de entrada está predefinida por el tamaño del vector

de características y ese es un parámetro definido en el diseño del extractor de características, la configuración de una red neuronal implica dos cosas: la determinación del número de capas neuronales escondidas y el número de neuronas en cada una de esas capas. En ambas redes se decidió probar con una sola capa de neuronas escondidas aunque se varió el número de neuronas en la ya mencionada capa. De ésta manera se definió:

N = Número de neuronas en la capa de entrada

L = Número de neuronas en la capa escondida

M = Número de neuronas en la capa de salida

4.1.1 Entrenamiento red de notas

Esta red neuronal tiene como objetivo reconocer los caracteres musicales por su forma. Se diseñó una red con un solo nivel escondido, un conjunto de entrenamiento de 14 clases donde cada clase se conforma de 20 patrones diferentes.

La red de notas se entrenó con el algoritmo de retropropagación (Ver 2.7.5) obteniendo en cada proceso de entrenamiento un archivo de texto que reportó el porcentaje de reconocimiento obtenido por la red barrida a barrida así como el error total por barrida asociado con la red. (Ver fig. 4.1)

Barrida No.	Error	% Reconocimiento
14941	4.8109	42.86%
14942	3.9819	57.14%
14943	4.0813	71.43%
14944	4.4295	57.14%
14945	4.2044	57.14%
14946	4.4642	57.14%
14947	3.3588	71.43%
14948	3.2949	78.57%
14949	3.7339	71.43%
14950	4.5582	50.00%
14951	4.2286	57.14%
14952	3.4602	78.57%
14953	3.9091	64.29%
14954	3.9772	71.43%
14955	3.3797	78.57%
14956	3.9004	64.29%
14957	2.9067	78.57%
14958	2.9444	78.57%
14959	3.7917	64.29%
14960	4.1827	57.14%
14961	4.8095	42.86%
14962	3.9797	57.14%
14963	4.0756	71.43%
14964	4.4205	57.14%
14965	4.1998	57.14%
14966	4.4574	57.14%
14967	3.3540	71.43%
14968	3.2896	78.57%
14969	3.7295	71.43%
14970	4.5549	50.00%
14971	4.2247	57.14%
14972	3.4549	78.57%
14973	3.9048	64.29%

Figura 4.1 Archivo reporte entrenamiento

Los resultados obtenidos de las pruebas de entrenamiento aplicadas a la red de notas se resumen en la tabla 4.1.

NUM. EXP.	NÚMERO BARRIDAS (N)	COEF. APREND. (η)	CONFIG. RED NEURAL (N-L-M)	PORCENT. RECONOC. (%)	ERROR FINAL (E_T)	CONCLUSIÓN PESOS Y UMBRALES
1	15000	0.1	140-15-14	64.29 %	3.7838	APROBADOS
2	15000	0.7	140-15-14	78.57 %	1.5532	APROBADOS
3	20000	0.1	140-15-14	78.57 %	2.6789	APROBADOS
4	20000	0.7	140-15-14	100 %	0.0132	APROBADOS
5	23000	0.1	140-15-14	78.57 %	2.6425	APROBADOS
6	23000	0.7	140-15-14	100 %	0.0259	APROBADOS
7	15000	0.1	140-25-14	21.43 %	5.3182	RECHAZADOS
8	15000	0.7	140-25-14	92.86 %	0.5194	APROBADOS
9	20000	0.1	140-25-14	85.71 %	1.7341	APROBADOS
10	20000	0.7	140-25-14	7.14 %	9.0852	RECHAZADOS
11	23000	0.1	140-25-14	7.14 %	10.6343	RECHAZADOS
12	23000	0.7	140-25-14	100 %	0.0072	APROBADOS
13	15000	0.1	140-50-14	7.14 %	91.0000	RECHAZADOS
14	15000	0.7	140-50-14	7.14 %	91.0000	RECHAZADOS
15	23000	0.7	140-50-14	7.14 %	91.0000	RECHAZADOS

Tabla 4.1 Pruebas de entrenamiento red de notas

Las figuras 4.2(a) y 4.2(b) muestran un comparativo del comportamiento del error total y el porcentaje de reconocimiento alcanzado en cada barrida para los mejores pesos y umbrales obtenidos cuando se entrenó a la red de notas empleando 15 y 25 neuronas en la capa escondida respectivamente.

4.1.1.1 Problemas encontrados en entrenamiento red de notas

El entrenamiento de la red de notas presentó las siguientes eventualidades:

- Un número mayor de barridas.-** Se intentó entrenar a la red de notas con un número mayor a las 23000 barridas, sin embargo, el sistema produjo un error que se intuyó estar asociado con falta de memoria RAM. Esta aseveración se vió avalada ya que originalmente se estaba trabajando en una computadora Pentium 4 con 256 MB en RAM y ésta no permitía entrenamientos mayores a las 10000 barridas. Posteriormente se tomó la decisión de ampliar la memoria RAM de la máquina de desarrollo a 512 MB permitiendo así la ejecución de un número mayor de barridas (24461 barridas como máximo)

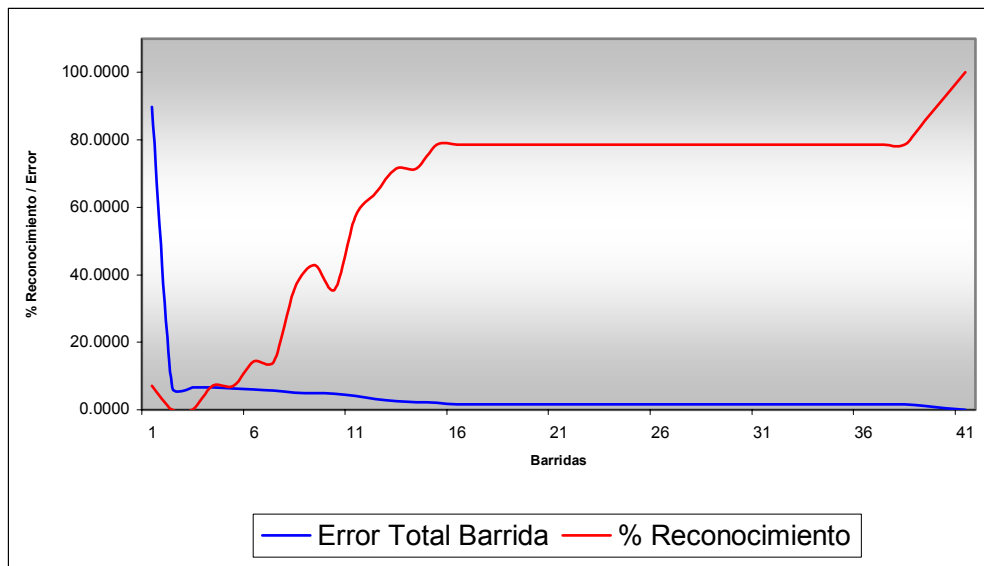


Figura 4.2(a) Comparativo error vs % reconocimiento experimento 4.

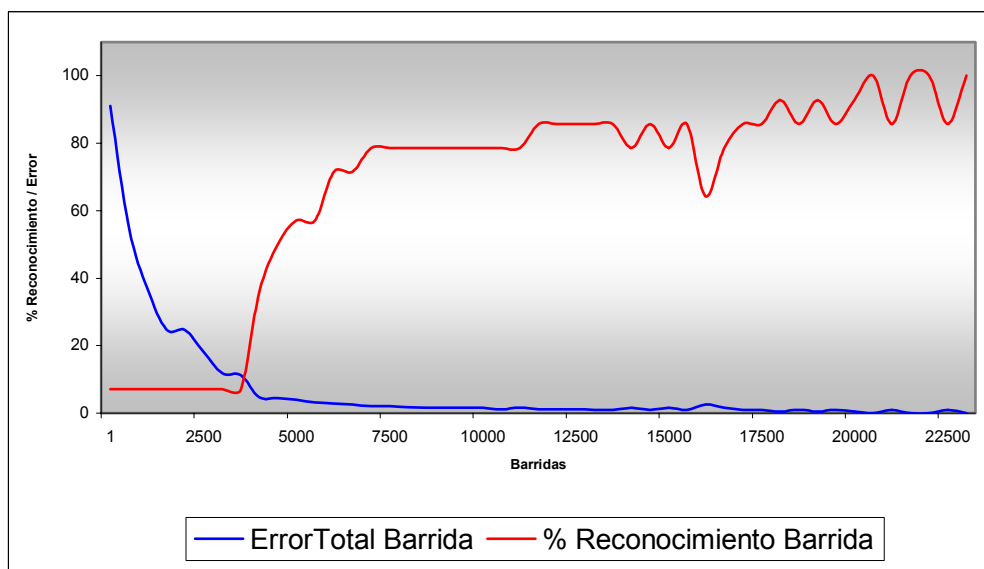


Figura 4.2(b) Comparativo error vs % reconocimiento experimento 12.

- **Tiempo de ejecución del entrenamiento.**- El tiempo de entrenamiento de la red de notas está en función directa con el número de barridas que se establece como condición de fin de entrenamiento. Para un entrenamiento de 23000 barridas el tiempo estimado de ejecución fue de 9 horas aproximadamente. Esto limitó enormemente la posibilidad de realizar un mayor número de experimentos para obtener conclusiones aún más sólidas sobre el proceso de entrenamiento

4.1.1.2 Trabajo a futuro para el entrenamiento de la red de notas

Queda como proyecto a futuro:

- Probar otras configuraciones neuronales además de las ya reportadas.
- Tratar de entrenar a la red con otros algoritmos de entrenamiento además de retropropagación.
- Probar otras arquitecturas neuronales distintas a las redes *feed-forward* como pueden ser las redes recurrentes.
- Disminuir el tiempo de ejecución del entrenamiento de la red así como lograr aumentar el número de barridas a ser ejecutadas.

4.1.2 Entrenamiento red de tonos

Esta red neuronal tiene como objetivo reconocer la posición que tienen en el pentagrama los caracteres musicales. Se diseñó una red con un solo nivel escondido, un conjunto de entrenamiento de 17 clases donde cada clase se conforma de 15 patrones diferentes.

Al igual que como sucedió con la red de notas, la red de tonos también se entrenó con el algoritmo de retropropagación. Los resultados obtenidos de las pruebas de entrenamiento aplicadas a la red de tonos se resumen en la tabla 4.2.

Las figuras 4.3(a) y 4.3(b) muestran un comparativo del comportamiento del error total y el porcentaje de reconocimiento alcanzado en cada barrida para los mejores pesos y umbrales obtenidos cuando se entrenó a la red de tonos empleando 25 y 35 neuronas en la capa escondida respectivamente.

NUM. EXP.	NÚMERO BARRIDAS (N)	COEF. APREND. (η)	CONFIG. RED NEURAL (N-L-M)	PORCENT. RECONOC. (%)	ERROR FINAL (E_T)	CONCLUSIÓN PESOS Y UMBRALES
1	10000	0.1	240-25-17	94.12 %	0.5258	ACEPTADOS
2	10000	0.7	240-25-17	100 %	0.0181	ACEPTADOS
3	12000	0.1	240-25-17	76.47 %	2.0054	ACEPTADOS
4	12000	0.7	240-25-17	88.24 %	1.0145	ACEPTADOS
5	10000	0.1	240-35-17	76.47 %	2.0142	ACEPTADOS
6	10000	0.7	240-35-17	94.12 %	0.5112	ACEPTADOS
7	12000	0.1	240-35-17	5.88 %	135.9999	RECHAZADOS
8	12000	0.7	240-35-17	5.88 %	135.9999	RECHAZADOS

Tabla 4.2 Pruebas de entrenamiento red de tonos

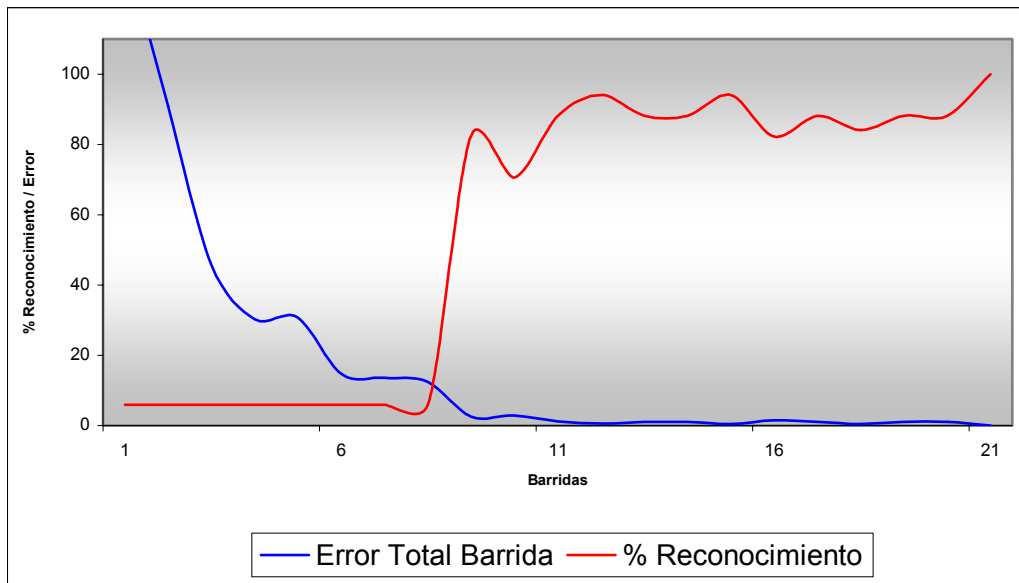


Figura 4.3(a) Comparativo error vs % reconocimiento experimento 2.

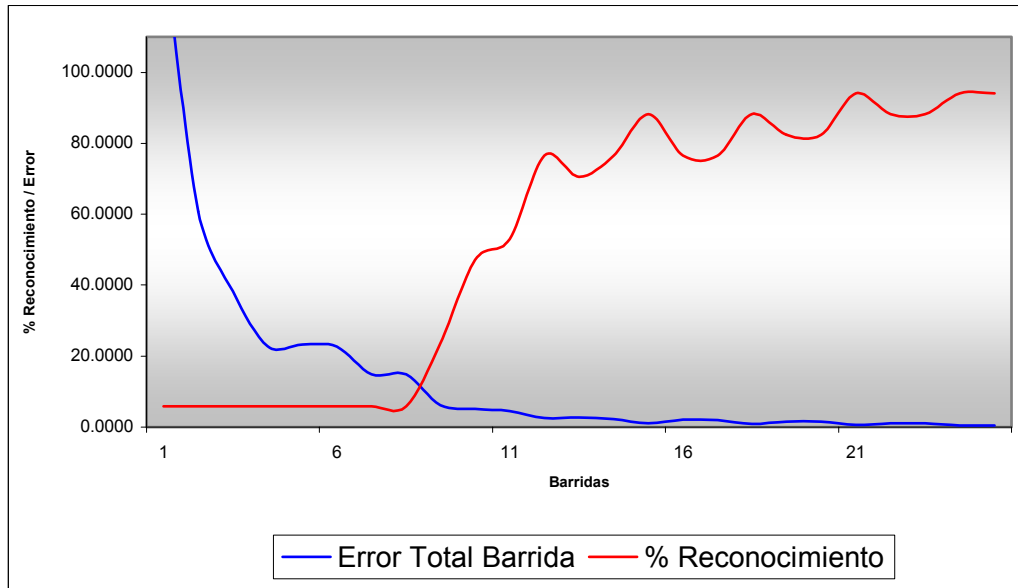


Figura 4.3(b) Comparativo error vs % reconocimiento experimento 6.

4.1.2.1 Problemas encontrados en entrenamiento red de tonos

El entrenamiento de la red de tonos presentó las siguientes eventualidades:

- **Un número mayor de barridas.-** Como sucedió en la red de notas, se intentó entrenar a la red de tonos con un número mayor a las 12000 barridas obteniendo el error anteriormente citado. El número máximo de barridas que permitió la red fue 12560. La red de tonos permite menos barridas que la red de notas porque su configuración y su conjunto de entrenamiento son mayores.
- **Tiempo de ejecución del entrenamiento.-** Por razones aún desconocidas, el tiempo de ejecución del entrenamiento de la red de tonos es menor que la red de notas; siendo este de 5 horas aproximadamente. De cualquier manera, éste tiempo también limitó la posibilidad de realizar un número mayor de experimentos.

4.1.2.2 Trabajo a futuro para el entrenamiento de la red de tonos

Queda como proyecto a futuro las mismas sugerencias citadas en la red de notas.

4.2 PRUEBAS CON PARTITURAS MUSICALES

Se probaron cinco partituras musicales diferentes:

- Partitura 1 = “Ma qué” *Anónimo*. Resolución= 200 ppp. (Ver fig. 4.4)
- Partitura 2 = “Carros de Fuego” *Vangelis*. Resolución= 300 ppp. (Ver fig. 4.5)
- Partitura 3 = “Tema de Nadia” *Barry de Vorzon*. Resolución=300 ppp. (Ver fig. 4.6)
- Partitura 4= “La Bella Durmiente” *Stephen Foster*. Resolución=300 ppp (Ver fig. 4.7)
- Partitura 5= “El Danubio Azul” *Johan Strauss*. Resolución=200 ppp (Ver fig. 4.8)



Figura 4.4 Partitura 1 “Ma Qué” [MONZON,98]

Cabe mencionar que todas las partituras, exceptuando la partitura 4, tuvieron que ser preprocesadas manualmente después de su transducción. Este preprocesamiento manual incluyó:

- Mejorar las partituras en brillo, contraste, orientación.
- Eliminar aquellos caracteres musicales que no se encontraban en el conjunto de entrenamiento de la red de notas. Tales como compás, arpegio, bemol, etc.
- Retocar en la medida de lo posible aquellas “imperfecciones” tales como sombras y omisiones producto de una transducción defectuosa.
- Guardarlas en disco con el formato de mapas de bits de 8 bits por píxel.

Figura 4.5 Partitura 2 “Carros de Fuego” [MONZON,98]

Figura 4.6 Partitura 3 “Tema Nadia” [MONZON,98]



Figura 4.7 Partitura 4 “Bella Durmiente” [MONZON,98



Figura 4.8 Partitura 5 “Danubio Azul (fragmento)” [MONZON,98

Las pruebas que se aplicaron a cada partitura incluyeron: segmentación, binarización, normalización, extracción de características y reconocimiento por medio de la red de notas y de tonos. La tabla 4.3 muestra los resultados de las pruebas hechas a las cinco partituras con la red de tonos.

ID PARTITURA	NUM. EXP. ENTRENAM.	CONFIG. RNA	NUM. CARACT. MUSICALES	PORCENTAJE GENERALIZACIÓN
Partitura 1	4	140-15-14	100	67 %
Partitura 2	4	140-15-14	83	76 %
Partitura 3	4	140-15-14	154	56 %
Partitura 4	4	140-15-14	320	58 %
Partitura 5	4	140-15-14	61	83 %
Partitura 1	12	140-25-14	100	70%
Partitura 2	12	140-25-14	83	68 %
Partitura 3	12	140-25-14	154	56 %
Partitura 4	12	140-25-14	320	62 %
Partitura 5	12	140-25-14	61	76 %

Tabla 4.3 Generalización red de notas

La tabla 4.4 muestra las pruebas realizadas a las tres partituras con la red de tonos.

La tabla 4.5 clasifica los errores encontrados a la hora de procesar cada partitura. La causa por la que un caracter musical no sea reconocido correctamente se contabilizará a solo una posible falla, ya que una falla en un proceso previo puede provocar una falla en un proceso posterior.

4.2.1 Problemas encontrados en el procesamiento de partituras

Entre los principales problemas encontrados en la construcción de S.R.P.M. se pueden citar:

ID PARTITURA	NUM. EXP. ENTRENAM.	CONFIG. RNA	NUM. CARACT. MUSICALES	PORCENTAJE GENERALIZACIÓN
Partitura 1	2	240-25-17	80	43 %
Partitura 2	2	240-25-17	82	56 %
Partitura 3	2	240-25-17	126	41 %
Partitura 4	2	240-25-17	210	58 %
Partitura 5	2	240-25-17	27	74 %
Partitura 1	6	240-35-17	121	55 %
Partitura 2	6	240-35-17	65	50 %
Partitura 3	6	240-35-17	126	38 %
Partitura 4	6	240-35-17	210	59 %
Partitura 5	6	240-35-17	27	81 %

Tabla 4.4 Generalización red de tonos

ID PARTITURA	CONFIG. RNA NOTAS	CONFIG. RNA TONOS	FALLOS SEGM. PARTIT.	FALLOS SEGM. ACORDE	FALLOS SEGM. CORCHEA
Partitura 1	140-15-14	240-25-17	0	12	3
Partitura 1	140-25-14	240-35-17	0	12	3
Partitura 2	140-15-14	240-25-17	1	4	0
Partitura 2	140-25-14	240-35-17	1	4	0
Partitura 3	140-15-14	240-25-17	21	8	0
Partitura 3	140-25-14	240-35-17	21	8	0
Partitura 4	140-15-14	240-25-17	0	4	0
Partitura 4	140-25-14	240-35-17	0	4	0
Partitura 5	140-15-14	240-25-17	0	0	0
Partitura 5	140-25-14	240-35-17	0	0	0

Tabla 4.5 Fallos en pruebas con partituras

- **Pentagramas con poca separación uno del otro.-** Este problema produce una segmentación errónea ya que aunque los caracteres musicales son detectados correctamente la segmentación produce caracteres musicales incompletos. Este error se verificó en el quinto pentagrama de la partitura 3 en donde se encuentran, en el pentagrama de clave de Fa, tonos musicales pertenecientes a las clases 16 y 17, que resultan ser los más altos de acuerdo a su posición que guardan en el pentagrama y en el pentagrama de clave de Sol, corcheas de dos y tres bastones cuya barra se dibujó hacia abajo provocando éste error. Ver figura 4.9.

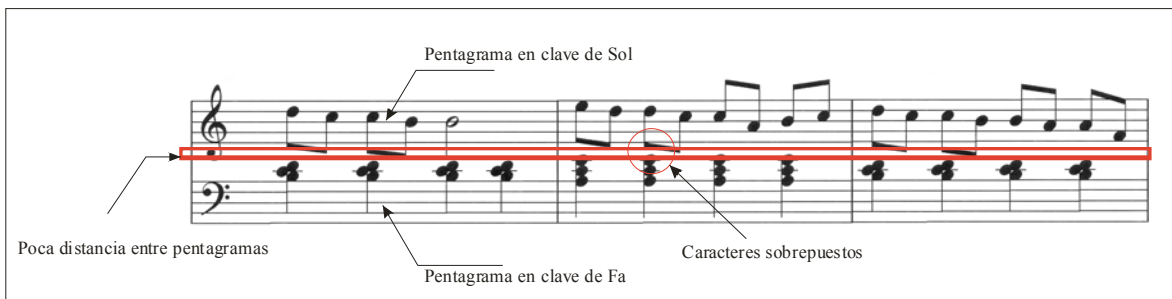


Figura 4.9 Poca distancia entre pentagramas

- **Caracteres musicales compuestos.-** Este problema se encontró durante la segmentación cuando un carácter musical se componía de dos o más figuras y éstas no se encontraban unidas entre sí, como en el caso de la clave de Fa, el carácter musical debe ser segmentado juntando las figuras que lo componen. La función AfinaCeros() (Ver Apéndice B) hace frente a este error (Ver fig. 4.10)

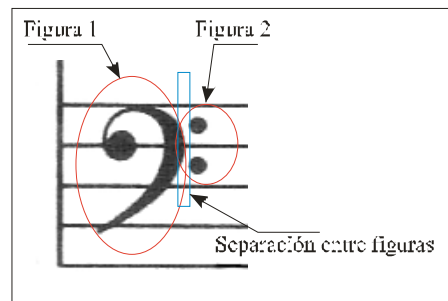


Figura 4.10 Caracteres musicales compuestos

- **Líneas del pentagrama.-** Las líneas de los pentagramas que componen a una partitura musical representaron otro problema ya que los algoritmos de segmentación que se codificaron fallaban al encontrarse que los objetos que componían la escena en 2-D no podían ser separados ya que detectaba todo el pentagrama como un solo objeto. Ver fig. 4.11. La función DetectaLíneasPentagrama() (Ver Apéndice B) da solución a este problema. Ver fig 4.12.
- **Corcheas incompletas.-** Como se explicó con anterioridad, SRPM necesita eliminar las líneas de los pentagramas para poder realizar el proceso de segmentación, sin embargo, existen ocasiones en las que algunas barras que unen los bastones de las corcheas provocan separación vertical entre un bastón y otro. Esto hace que SRPM detecte dos objetos en lugar de uno. Este error se intentó resolver también mediante la aplicación de la función AfinaCeros antes citada, sin embargo existieron casos en los que aún con la aplicación de dicha función el error persistía. Fue entonces cuando se decidió resolver este error retocando un poco más la barra de las corcheas problema. Este problema se verificó también para algunas redondas.

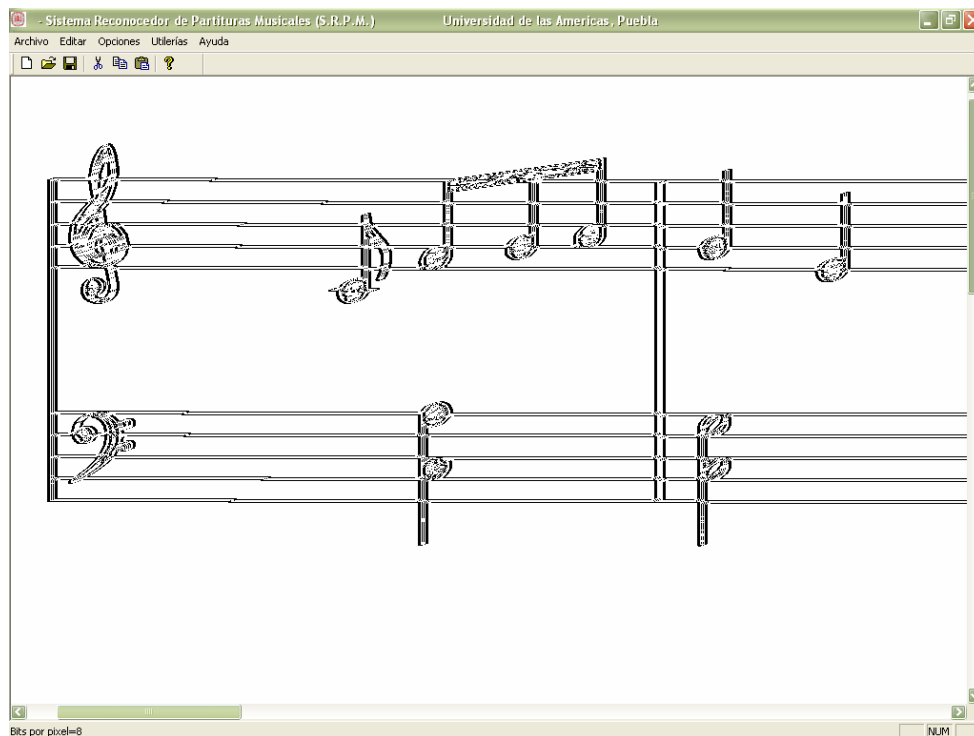


Figura 4.11 Error en segmentación por presencia de líneas de pentagrama.

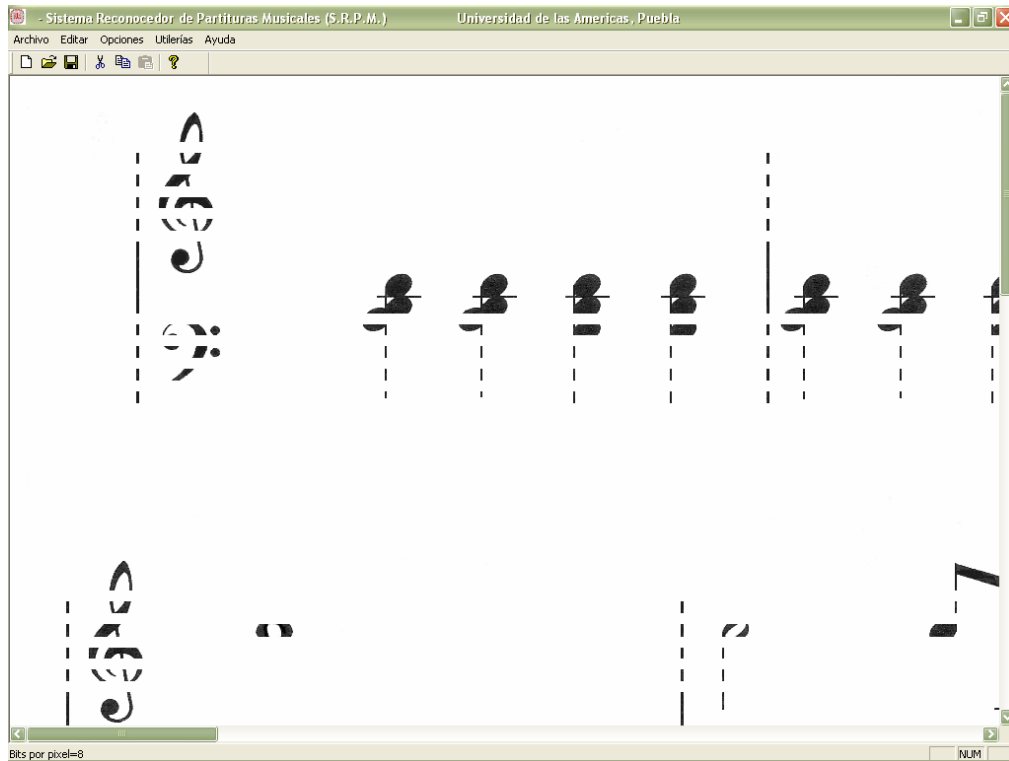


Figura 4.12 Partitura musical sin líneas de pentagrama

- **Error en el reconocimiento de los tonos de una corchea.-** Las corcheas presentaron un problema muy peculiar; la barra de la que parten los bastones causó mucho ruido a la red de tonos ya que constantemente la red confundía la ubicación de la bola creyendo que se encontraba donde estaba la barra. La fig. 4.13 muestra éste problema. Cabe mencionar que por falta de tiempo no se corrigió este error por lo que se sugiere como trabajo a futuro en este proyecto

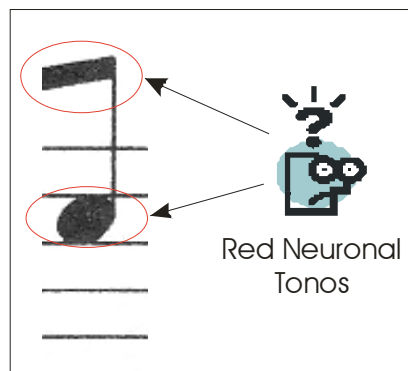


Figura 4.13 Error en el reconocimiento del tono de una corchea segmentad

- **Confusión entre líneas y acordes.**- Los vectores de características de las líneas de mediano grosor son muy poco ortogonales con respecto a los acordes de 4 y 3 tonos provocando que la red de notas constantemente reconozca equivocadamente a estos patrones. Como ya se mencionó anteriormente, los vectores de características de los patrones que conforman el conjunto de entrenamiento de la red de notas se obtiene mediante la técnica de histogramas horizontales; este método de extracción de características le otorga a la red de notas poco “margen de maniobra” para distinguir ciertos patrones de otros.

4.2.2 Trabajo a futuro par el sistema reconecedor de partituras musicales

Las posibilidades de mejora a este proyecto son muchas y muy variadas. A continuación se sugieren algunas ampliaciones al proyecto S.R.P.M. buscando alcanzar objetivos más ambiciosos que los planteados en esta ocasión:

- Construir el módulo de generación de código MIDI, ya que por falta de tiempo este módulo quedo pendiente de ser desarrollado.
- Ampliar el número de caracteres musicales que puedan ser reconocidos por la red de notas actualmente en 14 pero dejando todavía varios caracteres especiales por ser incluidos tales como, arpeggio, bemol, becuadrado, compás, natural y diferentes significados de las líneas.
- Mejorar el proceso de segmentación de corcheas y acordes buscando ayudar a la red de tonos a mejorar notablemente su capacidad de reconocimiento.
- Específicamente hablando de las líneas éstas tienen distintos significados en una partitura, uno de los más interesantes es el de repetición de un pedazo de melodía. Una ampliación al proyecto en este tenor sería de real interés
- Probar otros algoritmos de segmentación sobre todo en lo que se refiere a algoritmos cuya base es la unión y separación de regiones.
- Agregar un módulo de generación automática de archivos maestros empleados para poder ejecutar el módulo de reconocimiento. Esta tarea se efectúa manualmente en este momento.

- Añadir algún módulo de preprocesamiento automático de imágenes digitales ya que esta tarea consume mucho tiempo y es bastante desgastante.
- Trabajo con otros formatos de archivos gráficos adicionales a los mapas de bits (*.BMP).
- Probar métodos de extracción de características más robustos que los ya probados de histogramas horizontales y verticales buscando obtener vectores de características más ortogonales que faciliten a la red el correcto aprendizaje de los patrones.
- Generar vectores de características más grandes que los ya probados para ambas redes neuronales (actualmente en 240 neuronas para la red de tonos y 150 para la red de notas)

4.3 CONCLUSIONES

Al término del desarrollo de este proyecto se obtuvieron las siguientes conclusiones:

- El procesamiento de imágenes digitales es un campo muy amplio y complejo, la segmentación de imágenes digitales debe ser considerada como uno de los mayores retos dentro de este campo. Cada problema debe estudiarse por separado para poder elegir el mejor método segmentación automática.
- Cualquier procesamiento digital de imágenes es enormemente dependiente del ruido producido por el proceso de transducción. Es aquí donde toman una real importancia los métodos de preprocesamiento de imágenes tales como: corrección de brillo, contraste, etc.
- Las redes neuronales, como es lógico suponerlo, dependen enormemente de un buen proceso de entrenamiento, sin embargo, el éxito o fracaso de éste último depende de muchos más factores que el simple algoritmo aplicado. Una de las dependencias más fuertes que tiene es la calidad de los vectores de características sobre todo para patrones que presentan poca ortogonalidad. Esto hace suponer que el proceso de extracción de características no debe ser considerado como trivial así como tampoco lo es la definición del tamaño del vector producido.

- Las redes neuronales son una buena opción como módulos reconocedores de patrones, pero hay que estar conscientes que su desempeño puede ser impredecible ya que se ven afectadas por la definición de muchos parámetros como pueden ser el coeficiente de aprendizaje, la determinación de la condición de fin de entrenamiento y la determinación del número de niveles escondidos junto con la definición del número de neuronas asociadas a cada nivel escondido.
- El campo de la música es muy amplio y en algunas ocasiones parece demasiado complejo el desarrollo de un sistema que pudiese reconocer correctamente cualquier partitura ya que la cantidad de detalles que éstas pueden incluir son interminables.
- El tiempo de desarrollo del presente proyecto fue excesivo ya que el número de dificultades a las que se le tuvo que dar frente fueron muchas siendo algunas de ellas realmente complicado darles solución. Otro factor que contribuyó fue el que los alcances propuestos para este proyecto fueron muchos y demasiado ambiciosos. El desarrollo de este proyecto fue más complicado que el cursar las asignaturas de la maestría misma.
- El proceso de segmentación de acordes y corcheas debe ser mejorado sustancialmente ya que un buen proceso de segmentación fortalece la capacidad de reconocimiento de redes neuronales.
- Se observó que al aumentar el tamaño de las clases las redes mejoraron un poco su capacidad de reconocimiento aunque, obviamente, aumentaba el tiempo total de entrenamiento. El tamaño actual de las clases es de 40 patrones para cada clase que conforma el conjunto de entrenamiento de la red de notas y de 30 patrones para la red de tonos.
- Se observó que existía un mejor desempeño de ambas redes al mantener una sola capa escondida con no más de 25 neuronas. El coeficiente de aprendizaje que mejores resultados arrojó fue de 0.7 aunque con 0.1 el desempeño tampoco es malo. Se probaron otros coeficientes obteniendo siempre desempeños muy malos en cuanto a la disminución del error total por barrida.
- La red de notas utiliza entre 20,000 y 23,000 barridas para lograr reconocer al 100% del conjunto de entrenamiento reportando un error aceptable de menos de 0.1. Para

la red de notas se necesitaron entre 10,000 y 12,000 barridas para lograr los mismos resultados que la red de notas.

- El tamaño del vector de características se determinó a prueba y error modificándolo en repetidas ocasiones para lograr mejores desempeños de ambas redes quedando al final en 150 bits para patrones notas y 240 bits para patrones tonos. Se puede asumir que un mayor tamaño podría redituarse en un mejor desempeño de las redes ya que éstas obtendrían mayor información acerca de cada uno de los patrones que intentan aprenderse. Es muy claro que la memoria RAM de la computadora juega un papel preponderante en este tenor.
- La segmentación de partituras musicales fué el proceso más complicado de entender y codificar y sin embargo es el que mejor desempeño muestra. Se probaron varias partituras además de las reportadas en este documento teniendo muy buen desempeño. Cabe mencionar que todavía existen algunos factores importantes que determinan en buena parte el éxito de este proceso tales como: el que la partitura esté derecha o con muy poca inclinación desde su transducción, no tener una resolución mayor a los 8 bits por píxel así como no manejar resoluciones durante la transducción mayores a 350 ppp. Se observó que el mejor desempeño del módulo de segmentación era cuando se tenía una resolución de 300 ppp.
- El sistema solo manipula archivos gráficos cuyo formato sea el de mapas de bits (*.BMP), la posibilidad de aumentar a otro tipo de formatos se propone como mejora a este proyecto.
- Finalmente y como conclusión final se puede decir que existen muchas metas en la vida de un ser humano y quizá algunas de ellas tomen tintes de inalcanzables y solo mediante el sacrificio, empeño y esfuerzo sostenido se logran alcanzar constituyendo entonces los verdaderos éxitos.