



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

**DESARROLLO DE UN
ALGORITMO BASADO EN REDES
NEURONALES PARA LA
DETECCIÓN AUTOMÁTICA DEL
TIPO DE INSTRUMENTO
MUSICAL**

Alumno: Joaquín Gálvez Gómez

Tutor: Prof. D. Fernando Parra Rodríguez

Depto.: Ingeniería de Telecomunicación

Febrero, 2018

Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

Curso 2017-2018

DESARROLLO DE UN ALGORITMO BASADO EN REDES NEURONALES PARA LA DETECCIÓN AUTOMÁTICA DEL TIPO DE INSTRUMENTO MUSICAL

Alumno: Joaquín Gálvez Gómez

Tutor: Prof. D. Fernando Parra Rodríguez

Depto.: Ingeniería de Telecomunicación

Febrero, 2018

Visto bueno a la defensa del TFG

Firma del autor

A handwritten signature in black ink, consisting of a stylized 'J' followed by a horizontal line and a vertical stroke.

Firma del tutor

A handwritten signature in black ink, featuring a circular flourish around the name 'F. Parra'.

Febrero, 2018

Índice de contenido

CAPÍTULO 1. INTRODUCCIÓN AL PROYECTO	7
1.1 Resumen	7
1.2 Objetivos	8
1.3 Estado del arte	8
CAPÍTULO 2. FÍSICA DEL SONIDO Y TEORÍA MUSICAL	9
2.1 La onda de sonido.....	9
2.2 Psicoacústica	11
2.2.1 Sistema auditivo humano	11
2.3 Propiedades del sonido.....	12
2.3.1 Intensidad	13
2.3.1.1 Envolvente	14
2.3.2 Tono.....	14
2.3.2.1 Escala cromática.....	15
2.3.2.2 Tabla de frecuencias MIDI.....	16
2.3.3 Timbre	17
CAPÍTULO 3. CONCEPTOS RELEVANTES DE PROCESADO DIGITAL DE SEÑAL ...	19
3.1 Transformada discreta de Fourier (DFT).....	19
3.2 Autocorrelación	22
3.3 Enventanado	25
3.3.1 Longitud	25
3.3.2 Forma o suavizado	27
CAPÍTULO 4. CONCEPTOS RELEVANTES DE LA COMPUTACIÓN NEURONAL	29
4.1 Redes Neuronales Artificiales (ANN)	29
4.1.1 Marco biológico	29
4.1.2 Marco artificial	30
4.1.3 Funciones de activación.....	31
4.1.4 Topologías.....	33
4.1.5 Aprendizaje.....	34
4.1.5.1 Aprendizaje supervisado	34
4.1.6 Clasificación de las redes neuronales según la información de entrada y salida	35
CAPÍTULO 5. EXTRACCIÓN DE CARACTERÍSTICAS	36
5.1 Bloque de procesamiento.....	37

5.1.1 Implementación del método de la autocorrelación. Cálculo de la frecuencia fundamental.....	37
5.1.2 Estudio estadístico. Filtrado paso bajo.....	38
5.1.3 Densidad espectral de potencia. Cálculo de la potencia relativa de los armónicos.....	39
5.2 Bloque de red neuronal.....	41
5.2.1 Entrenamiento de la red	41
5.2.2 Prueba de la red. Resultados	44
5.2.2.1 Sobreaprendizaje (Overfitting).....	51
CAPÍTULO 6. CONCLUSIONES Y LÍNEAS DE FUTURO.....	53
6.1 Conclusiones	53
6.2 Líneas de futuro.....	54
CAPÍTULO 7. BIBLIOGRAFÍA.....	54
CAPÍTULO 8. ANEXOS	56

Índice de figuras

Figura 1. 1: Diagrama de bloques de un sistema de reconocimiento de instrumentos	7
Figura 1. 3: Shazam.....	8
Figura 1. 4: Robot Nao.....	9
Figura 2. 1: Función seno.....	10
Figura 2. 2: Sistema auditivo humano.....	11
Figura 2. 3: Membrana basilar	12
Figura 2. 4: Envolvente de un sonido.....	14
Figura 2. 5: Escala cromática	15
Figura 2. 6: Representación en el tiempo de la señal musical Do (3) tocada por un violín	18
Figura 2. 7: Evolución de los principales armónicos pares. Do (3) Violín	18
Figura 2. 8: Evolución de los principales armónicos impares. Do (3) Violín	19
Figura 3. 1: Muestreo de una señal seno de frecuencia igual a 400 Hz y frecuencia de muestreo 8000 Hz	20
Figura 3. 2: Espectro de señal seno de frecuencia igual a 400 Hz muestreada a una frecuencia de muestreo de 8000 Hz.....	21
Figura 3. 3: Autocorrelación señal periódica	23
Figura 3. 4: Autocorrelación Do (2) Violonchelo	24
Figura 3. 5: Explicación gráfica de enventanado con solapamiento	26
Figura 3. 6: Tipos de ventanas de análisis.....	27
Figura 4. 1: Neurona biológica.....	29
Figura 4. 2: Neurona artificial.....	30
Figura 4. 3: Función de activación lineal	31
Figura 4. 4: Función de activación sigmoide logística.....	32
Figura 4. 5: Función de activación sigmoide tangente hiperbólica.....	32
Figura 4. 6: Topología multicapa con conexión feedforward	33
Figura 4. 7: Pesos según aprendizaje por corrección de error	34
Figura 5. 1: Violonchelo Do2 – La5 (.wav)	37
Figura 5. 2: Viola Do3 – Fa6 (.wav)	37
Figura 5. 3: Violín Sol3 – Mi7 (.wav).....	37
Figura 5. 4: Flauta travesera Do4 – Do7 (.wav).....	37
Figura 5. 5: Ejemplo visual de un filtrado de mediana	38
Figura 5. 6: Espectro nota Do (4) Flauta travesera.....	40
Figura 5. 7: Base de datos	42
Figura 5. 8: Contenido de datos de entrenamiento	42
Figura 5. 9: Ejecución de la función ‘leer.m’	43
Figura 5. 10: Datos de entrada (instrumentos.mat)	44
Figura 5. 11: Datos de salida deseados (target.mat).....	44
Figura 5. 12: Porcentajes de acierto en Prueba 1	46
Figura 5. 13: Ejecución ‘clasificación.m’	47

Figura 5. 14: Entrada analizada (Si (3) Viola)	47
Figura 5. 15: Salida de la red neuronal para un Si (3) Viola	48
Figura 5. 16: Espectro Si (3) Chelo Vs. Espectro Si (3) Viola.....	48
Figura 5. 17: Porcentaje de acierto en Prueba 2	49
Figura 5. 18: Visión gráfica del método Descenso de Gradiente	51
Figura 5. 19: Entrenamiento Prueba 3.....	52
Figura 5. 20: Porcentajes de acierto en prueba 3.....	53

Índice de tablas

Tabla 2. 1: Frecuencias MIDI	17
Tabla 3. 1: Resolución y grado de fugas para las distintas ventanas de análisis	28
Tabla 5. 1: Cabecera de la función ‘auto.m’	38
Tabla 5. 2: Cabecera de la función ‘filtro_mediana.m’	39
Tabla 5. 3: Cabecera de la función ‘armonicos.m’	41
Tabla 5. 4: Cabecera función ‘leer.m’	43
Tabla 5. 5: Creación red neuronal Prueba 1	45
Tabla 5. 6: Resultados de la prueba 1.....	46
Tabla 5. 7: Creación red neuronal Prueba 2	49
Tabla 5. 8: Resultados prueba 2	50

CAPÍTULO 1. INTRODUCCIÓN AL PROYECTO

1.1 Resumen

Este proyecto tiene como objetivo la implementación de un sistema capaz de reconocer instrumentos musicales de forma automática, utilizando como herramienta las Redes Neuronales Artificiales.

El algoritmo que llevará a cabo la clasificación instrumental constará de varias partes. En primer lugar, llevaremos a cabo un procesado de señal para extraer parámetros característicos de una serie de muestras instrumentales, concretamente provenientes de violonchelo, viola, violín y flauta travesera. Estas características extraídas formarán un conjunto de datos imprescindible para el sistema, pues extraeremos de manera numérica lo que para nosotros es el timbre de un instrumento, característica del sonido que como veremos más adelante, es la que permite a nuestro cerebro saber diferenciar entre diferentes fuentes sonoras.

El timbre de un instrumento está relacionado con la disposición de los armónicos de cada nota musical y cómo varían sus amplitudes en el tiempo, por lo que será clave calcular en la parte de procesado la frecuencia fundamental de cada nota y a partir de ella extraer la potencia relativa de dichos armónicos.

Será esta información valiosa para la red neuronal, que nos dirá qué instrumento suena en un determinado momento tras haber sido entrenada de manera supervisada.

La Figura 1.1 plasma lo explicado anteriormente, donde el bloque de extracción característica engloba la parte de procesado con su análisis estadístico correspondiente para descartar posibles datos atípicos encontrados, y el bloque de decisión será correspondiente al entrenamiento y simulación de la red neuronal para la obtención del instrumento en concreto.

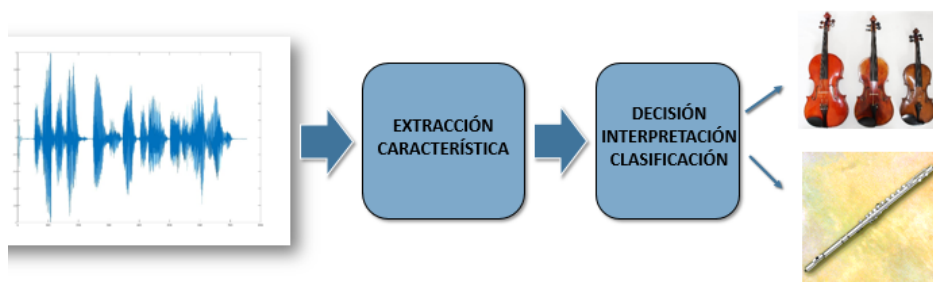


Figura 1. 1: Diagrama de bloques de un sistema de reconocimiento de instrumentos

1.2 Objetivos

Los objetivos de este proyecto son los siguientes:

- Comprender y llevar a la práctica un procesamiento de audio eficiente para la extracción de parámetros musicales que intervendrán en el algoritmo de estimación.
- Comprender la computación neuronal como tal, especialmente cuando es aplicada al reconocimiento de patrones y cuando aplicamos un entrenamiento supervisado.
- Estudiar el efecto en la precisión de la estimación de los parámetros extraídos en el procesamiento de señal
- Sacar las conclusiones oportunas sobre los resultados obtenidos para mejoras futuras.

1.3 Estado del arte

La oferta de software para reconocimiento y clasificación de sonidos es grande actualmente, y ha conseguido su mayor crecimiento en los últimos años. Combinar técnicas de procesamiento de audio con métodos de aprendizaje automático (*Machine Learning*) [1] es la clave para crear sistemas que reconozcan voz, música y cualquier sonido que nos rodea en el día a día.

En el ámbito musical destacamos los sistemas para la recuperación de información musical. Estos sistemas permiten obtener la información completa de un archivo de audio u obra musical a partir de un fragmento de ella. Es lo que hace la aplicación que hoy día lleva todo el mundo en su Smartphone, denominada *Shazam* [2], que reconoce cualquier canción con tan solo escuchar unos segundos.



Figura 1. 2: Shazam

Los sistemas para la transcripción musical también son destacados en este ámbito y que actualmente sustituyen a la manera tradicional de componer obras musicales, sabiendo escribir en notación musical lo que se escucha en tiempo real.

Respecto a voz es de importancia destacar la aplicación de estos sistemas a la robótica. Es hoy día lo más actual y lo que está revolucionando el mercado de la inteligencia artificial. La máquina o robot aprende escuchando, y es la persona humana que tiene el robot la que participa en primera persona en el aprendizaje, pues si el robot no reconoce algún patrón o sonido, seremos nosotros quien se lo hagamos saber, de esta manera la máquina aprende. Es lo que se conoce como *Machine Hearing* [3].

La Figura 1.3 muestra la forma del robot Nao [4], uno de los robots más caros actualmente en el mercado, rondando los 7000 euros y creado por Adebaran Robotics.



Figura 1. 3: Robot Nao

Muchas universidades del mundo ya disfrutan de él para proyectos de investigación. Entre otros robots del estilo se encuentra Aibo, previo a Nao y creado por Sony. Véase [5].

CAPÍTULO 2. FÍSICA DEL SONIDO Y TEORÍA MUSICAL

2.1 La onda de sonido

Un sonido, desde un punto de vista físico, es una onda. Concretamente denominada onda acústica, cuya velocidad es la velocidad característica del medio por el que se propaga.

La propagación del sonido se debe fundamental al medio, pues solo un medio material y además elástico experimentará cambios de presión debidos a dicha onda que permitirán su propagación. El que un medio sea material significa que está compuesto de

moléculas, y el que sea elástico significa que estas moléculas podrán recobrar su posición de equilibrio cuando se desplacen de ella por una determinada vibración [6].

Se dice que una onda es periódica cuando experimenta ciclos repetitivos para un periodo T de señal y que será la función seno, la forma más sencilla de expresar una excitación u onda de este tipo, a partir de la cual podrán surgir ondas más complejas.

Matemáticamente diremos que una señal es periódica si se cumple la siguiente ecuación:

$$a(t) = a(t + T) = a(t + nT) \quad (1)$$

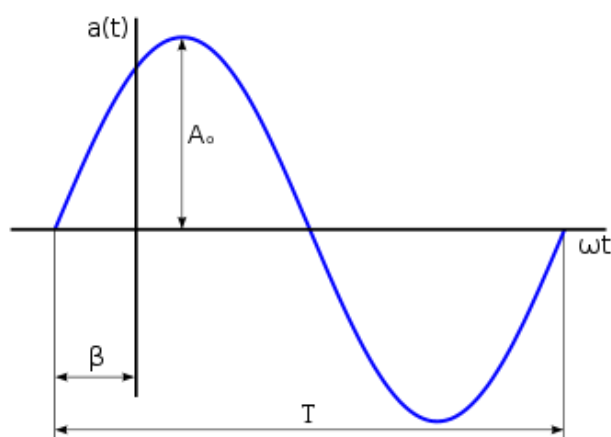


Figura 2. 1: Función seno

Como vemos, toda onda senoidal vendrá caracterizada por su amplitud A_0 , su posición de comienzo o fase β y su frecuencia f , la cual es la inversa del periodo T .

Conociendo la velocidad de propagación c de la onda y su frecuencia, podremos hallar fácilmente la longitud de onda, que viene a definirse como la distancia entre repeticiones de señal o la distancia entre dos picos de compresión o expansión de la onda [6], tal cual representamos matemáticamente con la siguiente ecuación:

$$\lambda = c/f \quad (2)$$

2.2 Psicoacústica

El cómo es percibido por el humano el fenómeno físico referenciado anteriormente, lo explica la ciencia denominada psicoacústica [7], que nos lleva a adentrarnos en el sistema auditivo humano para entender todo el proceso de audición, desde que la onda acústica penetra en nuestro oído, hasta que el cerebro procesa dicha información.

La complejidad de nuestro cerebro es mayor que la del cerebro animal o la del sistema nervioso de otro tipo de organismo, pues presenta unas características que lo individualizan plenamente (véase [18]). Entender el mecanismo de audición será de interés en un proyecto como este, donde uno de los objetivos será simular algo lo más parecido posible a un cerebro humano, capaz de llevar a cabo operaciones y aprender.

2.2.1 Sistema auditivo humano

Nuestro sistema auditivo se divide en tres partes fundamentales diferenciadas en la siguiente figura: Oído externo, oído medio y oído interno [7].

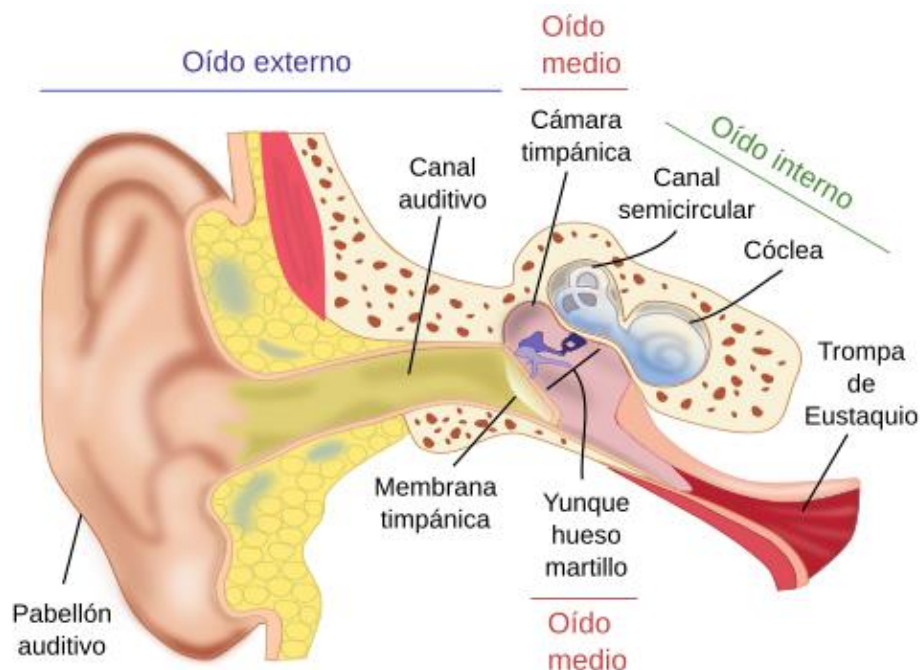


Figura 2. 2: Sistema auditivo humano

La captación de la onda acústica es realizada por la oreja, y es canalizada por el conducto auditivo hasta llegar al tímpano. En esta primera parte del sistema es donde se produce una amplificación de la señal que se transmite, gracias a una frecuencia de resonancia típica (4 KHz) para la cual se produce cierta ganancia [7].

Seguidamente, en el oído interno, compuesto por la cadena de huesecillos, es donde se realiza la adaptación de impedancias, necesaria por el cambio de medio (de aire a fluido).

Y finalmente, es en el oído interno donde se lleva a cabo la conversión mecánica – eléctrica para la transmisión de información al cerebro. En esta parte del sistema intervienen los canales semicirculares, la cóclea y el nervio auditivo.

De estas tres partes, es importante detenerse en la cóclea o caracol, donde se encuentra la membrana basilar, cuya fisiología, se plasma en la Figura 2.3.

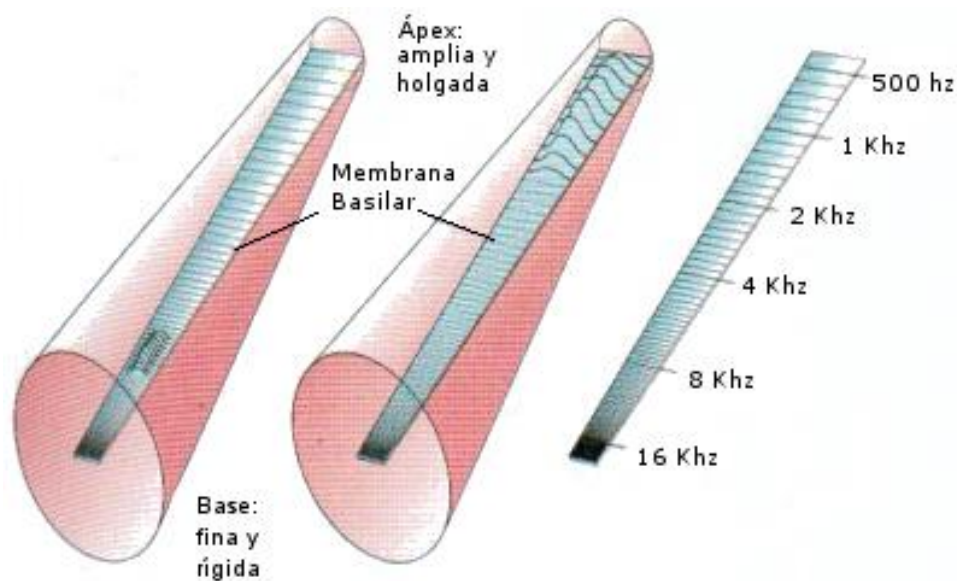


Figura 2. 3: Membrana basilar

Como vemos, en la parte más ancha de la membrana (ápex) predominan las bajas frecuencias, siendo en la parte más fina y estrecha de esta donde predominan las altas frecuencias [7]. Es hasta la mitad de dicha membrana donde se prolongan las medias y bajas frecuencias (hasta 4000 Hz aproximadamente) siendo en el último y pequeño tramo de esta donde predominan las altas (hasta 16000 Hz). Esto explica por tanto que, a cualquier variación de frecuencia dada en bajas frecuencias, seremos más sensibles.

2.3 Propiedades del sonido

En música, la forma de interpretar cualquier composición musical se plasma en lo que conocemos como partitura.

En la partitura se plasman tres de los cuatro atributos que definen de forma muy básica una nota musical: Intensidad sonora, tono y duración temporal. El timbre, que sería el cuarto atributo (y es el que define cómo suena subjetivamente el instrumento ejecutante) no se puede expresar.

2.3.1 Intensidad

La intensidad de un sonido es una magnitud física medible, definida como la cantidad de potencia acústica radiada por unidad de área por una determinada fuente sonora.

Siendo P la potencia media emitida por la fuente y r la distancia a la que se encuentra el oyente, la intensidad I del sonido vendrá determinada por la siguiente expresión [7].

$$I = \frac{P}{4\pi r^2} \quad (3)$$

El oído humano se adaptará a un intervalo de intensidades comprendido entre la intensidad del umbral absoluto o intensidad de referencia, igual a 10^{-12} W/m^2 y la intensidad intolerable de 1 W/m^2 .

En términos prácticos, nos solemos referir a la intensidad sonora con lo que llamamos nivel de intensidad (IL) [7], que es la forma de expresar en decibelios la intensidad de un sonido.

$$IL = 10\log\left(\frac{I}{I_0}\right) \quad (4)$$

Siendo I la intensidad del sonido, y la I_0 la intensidad de referencia. Según esta expresión, podemos decir que el umbral absoluto estará en los 0 decibelios de nivel de intensidad, mientras que el umbral intolerable será de 120 decibelios.

2.3.1.1 Envolvente

La variación temporal de la intensidad sonora determinará lo que conocemos como envolvente del sonido. Será un gráfico con el que apreciaremos cómo evoluciona la amplitud de cada nota musical.

En dicha envolvente diferenciaremos 3 partes, el ataque, el sostenimiento y el decaimiento.

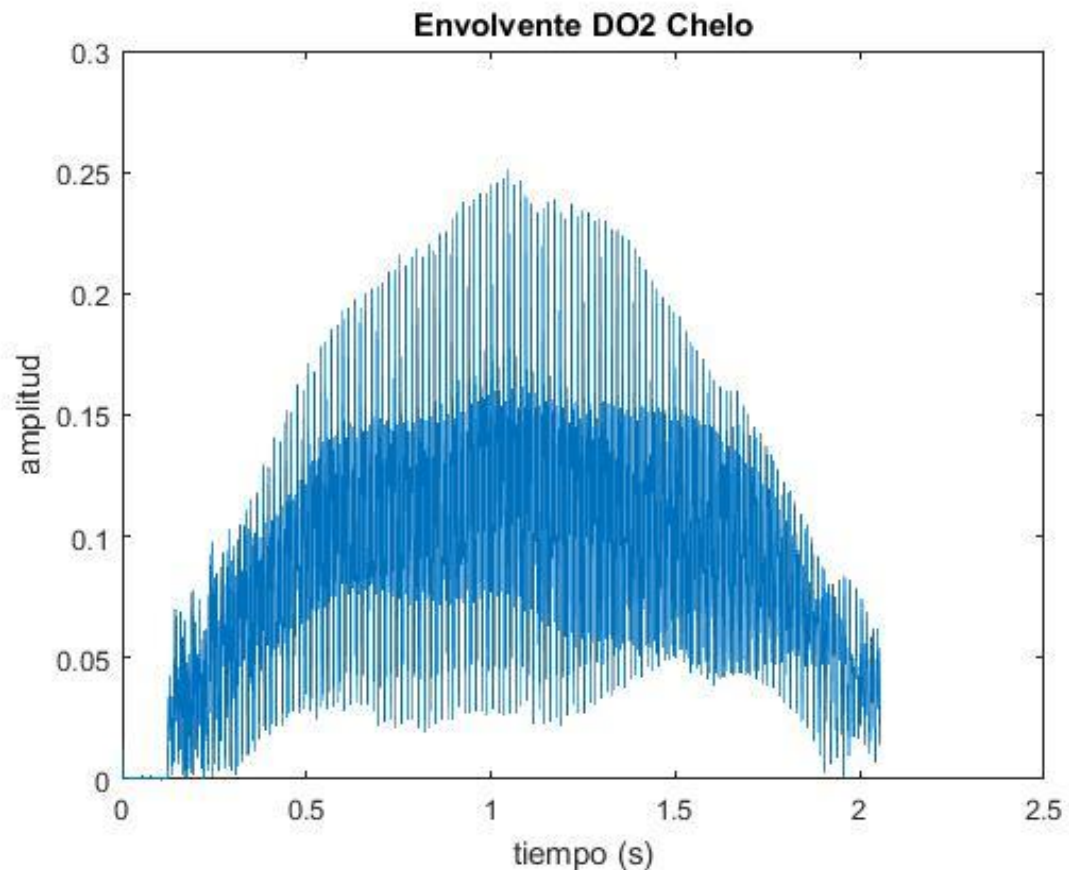


Figura 2. 4: Envolvente de un sonido

2.3.2 Tono

La percepción humana del tono o pitch está directamente relacionada con la frecuencia fundamental de una señal [7]. Las notas tocadas por instrumentos musicales son señales que a su vez son combinaciones de componentes sinusoidales (armónicos) de frecuencias f_0 , $2f_0$, $3f_0$, ... donde f_0 es la frecuencia fundamental de la nota en concreto, que domina en la percepción del tono.

Las frecuencias altas darán lugar a notas agudas, mientras que las frecuencias más bajas dan lugar a notas graves [7].

2.3.2.1 Escala cromática

En música, la escala cromática es una sucesión de sonidos de la que derivan los demás tipos de escalas, formada por 12 notas que dividen una octava [8]. La octava será el intervalo de frecuencia en hertzios comprendido entre una frecuencia y la de su valor doble. En la Figura 2.5 se representa esta escala, incluyendo también la nota con la que empieza la siguiente octava, f_{13} .

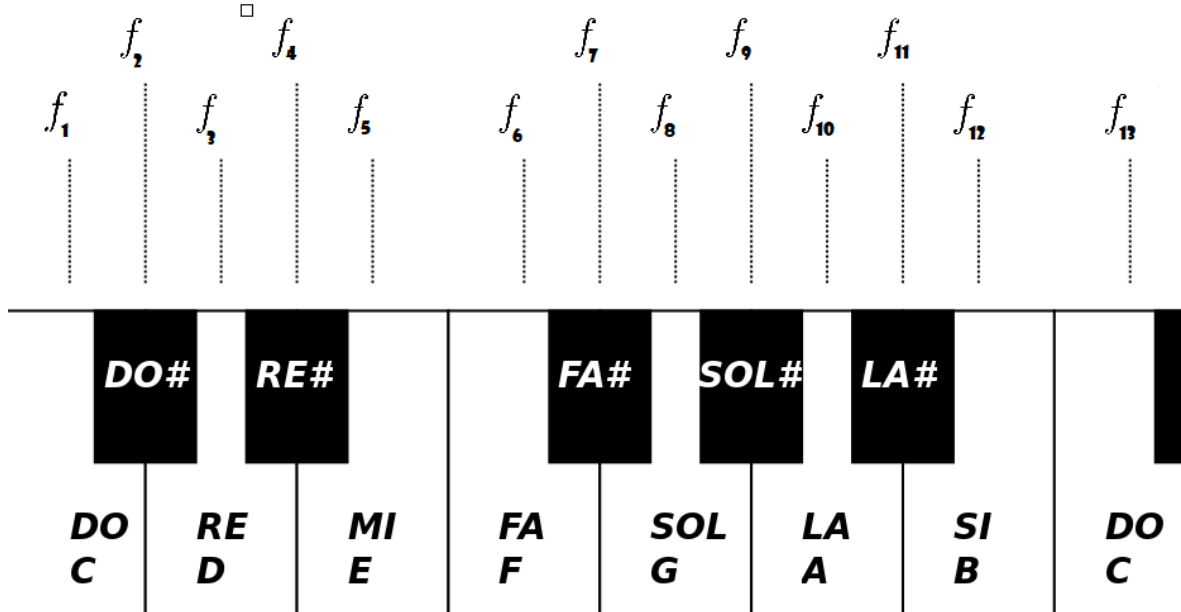


Figura 2. 5: Escala cromática

Un instrumento musical, podrá tocar todas estas notas para unas determinadas octavas. El conjunto de estas notas interpretadas dará nombre a la tesitura del instrumento. Concretamente en este proyecto, hemos partido analizando las tesituras de la familia de cuerda frotada (violín, viola y violonchelo), además de la de una flauta travesera.

Es importante saber que las frecuencias de las notas que componen la tesitura de un instrumento no siguen una progresión lineal, sino logarítmica, lo que significa que la relación entre una nota y su siguiente (1 semitono más alta) es constante (véase [8]).

$$\frac{f_2}{f_1} = K \rightarrow f_2 = K f_1 \quad (5)$$

$$\frac{f_3}{f_2} = K \rightarrow \frac{f_3}{Kf_1} \rightarrow \frac{f_3}{f_1} = K^2 \quad (6)$$

$$\frac{f_4}{f_3} = K \rightarrow \frac{f_4}{K^2 f_1} \rightarrow \frac{f_4}{f_1} = K^3 \quad (7)$$

Si siguiéramos este desarrollo, y sabiendo que la relación entre la nota de comienzo de la siguiente octava f_{13} y f_1 es del doble, deducimos que:

$$\frac{f_{13}}{f_1} = 2 = K^{12} = 1.059463 \quad (8)$$

Teniendo el valor de esta constante, deducimos la siguiente expresión con la que poder calcular el valor de frecuencia de una nota en concreto f , conociendo a priori el valor de otra f' y la distancia en semitonos d a la que se encuentran ambas. Véase [22].

$$f = f' K^d \quad (9)$$

2.3.2.2 Tabla de frecuencias MIDI

Nos podremos referir a las notas anteriores y a sus frecuencias en diversas octavas con su número de nota MIDI correspondiente, tal cual se representa en la Tabla 2.1. Generalmente se suele rellenar esta tabla hasta llegar a la nota MIDI 128, correspondiente a un Sol# en novena octava de frecuencia 13289,752 Hz.

Nota	Frecuencia	Nota MIDI
La 0	27,5	21
La # 0	29,135	22
Si 0	30,868	23

Do 1	32,703	24
Do# 1	34,648	25
Re 1	36,708	26
Re # 1	38,891	27
Mi 1	41,203	28
Fa 1	43,654	29
Fa # 1	46,249	30
Sol 1	48,999	31
Sol # 1	51,913	32
La 1	55	33
La # 1	58,27	34
Si 1	61,735	35
Do 2	65.406	36
Do# 2	69.296	37

Tabla 2. 1: Frecuencias MIDI

2.3.3 Timbre

El timbre es la característica subjetiva del sonido que permite a nuestro oído hacer una clasificación clara y concisa de los sonidos que se escuchan cuando son tonos de igual frecuencia y provenientes de diferentes instrumentos.

Esto se debe a su fuerte relación con las anteriores características explicadas, y a su dependencia con la fuente sonora y la forma de excitarla. El timbre de una nota determinada tocada por un violín será diferente al timbre de dicha nota tocada por una viola.

No es una característica que se pueda medir como por ejemplo la frecuencia de la señal que determina el tono, pero sí conceptual y de fácil descripción. El timbre es denominado en acústica como el color del sonido, la calidad de este, y vendrá determinado por la disposición de los armónicos o componentes sinusoidales que forman la señal y cómo varían en el tiempo.

Dicha variación se va a demostrar con el siguiente ejemplo: Análisis de la nota Do en cuarta octava tocado por una flauta travesera.

Como podemos ver a continuación en la Figura 2.6, por la forma de la onda, deducimos que las notas musicales no son una única señal periódica pura, sino que es la suma de varios sonidos o señales cuyas frecuencias son múltiplos enteros positivos de una frecuencia base o frecuencia fundamental, denominadas armónicos.

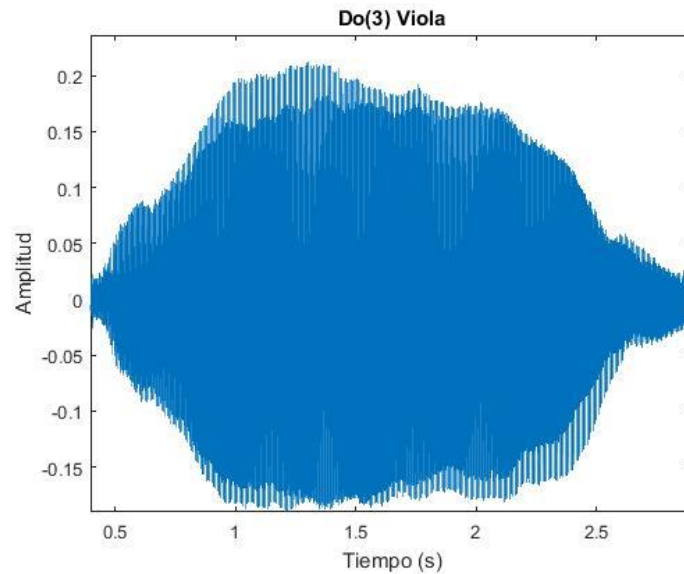


Figura 2. 6: Representación en el tiempo de la señal musical Do (3) tocada por un violín

Aun así, esta representación en el tiempo no nos da mucha más información sobre los armónicos que componen esta nota en concreto. Habría que llevar a cabo un análisis en frecuencia de dicha nota calculando la densidad espectral de potencia en pequeñas porciones temporales de esta (a la máxima resolución) y ver como varían sus amplitudes en el tiempo.

Esto se ha hecho en el software Matlab y se ha obtenido la siguiente variación en el tiempo para los principales armónicos pares e impares de la señal. Véase Figura 2.7 y Figura 2.8.

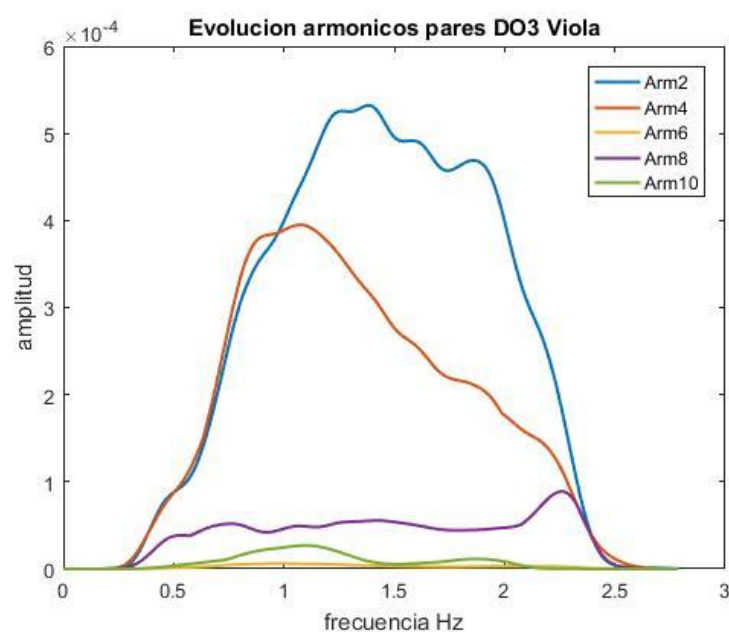


Figura 2. 7: Evolución de los principales armónicos pares. Do (3) Violín

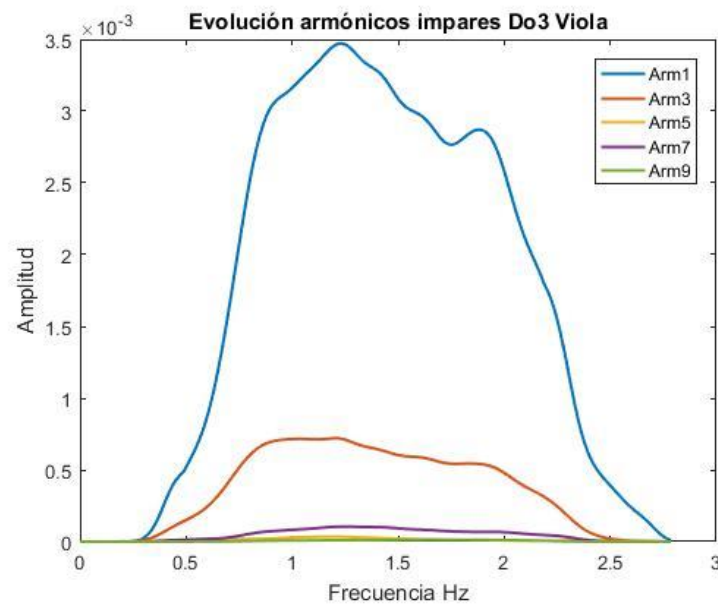


Figura 2. 8: Evolución de los principales armónicos impares. Do (3) Violín

Queda demostrado por tanto que la amplitud de los armónicos que componen una nota musical concreta varían en el tiempo, y que será un objetivo clave en este proyecto calcular dichas amplitudes, pues teniendo dicha información y pasándosela posteriormente a una red neuronal concreta, esta será capaz de reorganizarse y actuar de clasificador.

Para más información, podrá ver otro ejemplo como este en [9].

CAPÍTULO 3. CONCEPTOS RELEVANTES DE PROCESADO DIGITAL DE SEÑAL

3.1 Transformada discreta de Fourier (DFT)

La representación de toda señal en el tiempo nos proporciona una pobre información comparada con la que podríamos obtener si dicha señal es descrita en el dominio de la frecuencia o lo que es lo mismo, si obtuviéramos su espectro.

Para llevar a cabo esta representación, toda señal analógica previamente deberá ser registrada digitalmente mediante un proceso de muestreo, y así transformar cada una de las muestras de señal en el tiempo en componentes frecuenciales mediante la Transformada Discreta de Fourier (DFT).

El proceso de muestreo consiste en digitalizar la señal, de manera que el tiempo no transcurra de manera continua en ellas, si no en pequeños saltos equiespaciados determinados por la frecuencia a la que se han tomado las muestras, denominada frecuencia de muestreo f_s . La inversa de esta frecuencia será el periodo de muestreo T_s , que se define como el tiempo que transcurre entre dos muestras consecutivas de señal, y que deberá ser al menos el doble del periodo fundamental de señal, cumpliendo así el teorema de muestreo para evitar solapamientos o aliasing entre muestras frecuenciales, y permitir posteriormente la recuperación de la señal continua (véase [10]).

La Figura 3.1 representa un seno de 400 Hz, muestreado a una frecuencia de muestreo de 8000 Hz, esta nos indica el número de muestras que habrá en un tiempo de un segundo. En dicha imagen hemos representado cuatro milisegundos de señal, donde hay una cantidad de 33 muestras.

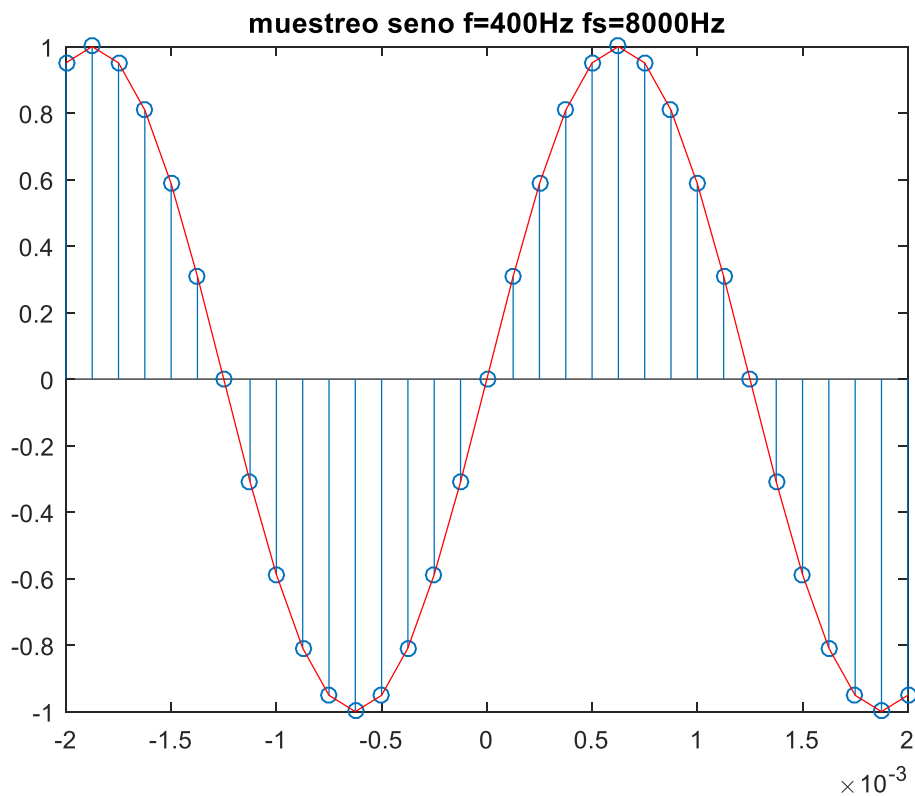


Figura 3. 1: Muestreo de una señal seno de frecuencia igual a 400 Hz y frecuencia de muestreo 8000 Hz

Si calculamos su espectro veremos un pico de amplitud en la frecuencia fundamental de dicha señal, tal y como vemos en la Figura 3.2.

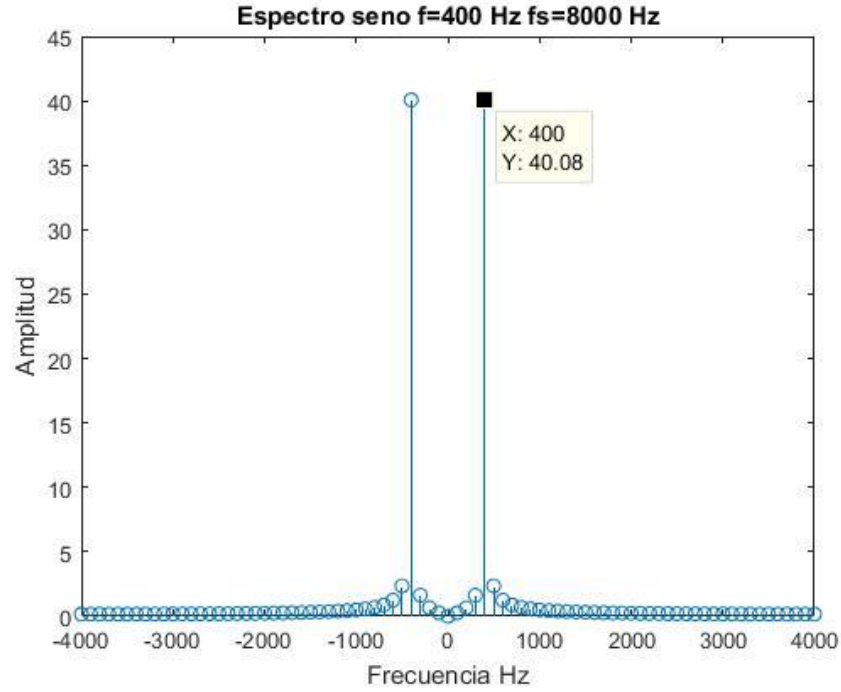


Figura 3. 2: Espectro de señal seno de frecuencia igual a 400 Hz muestreada a una frecuencia de muestreo de 8000 Hz

Afirmamos pues, que la información obtenida en el dominio de la frecuencia es mucho más valiosa que la que nos proporciona la señal representada en el tiempo.

La DFT es el algoritmo que nos permite obtener el espectro de una señal de acuerdo a la siguiente ecuación. Pues suponiendo que tenemos una señal muestreada en el tiempo $x[n]$ para $n \in [0, N - 1]$ siendo N el número de muestras de la señal, los coeficientes $x[k]$ del espectro serán:

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{N}}, \quad k = 0, \dots, N - 1 \quad (10)$$

La DFT se presenta en esta ecuación indicándonos que para N muestras de señal en el tiempo, se producen N coeficientes del espectro dando lugar a un muestreo en frecuencia dentro de un rango que va desde cero hasta la frecuencia de muestreo f_s . La distancia existente entre coeficientes contiguos se denomina resolución espectral, y vendrá determinada por el cociente de la frecuencia de muestreo y el número de coeficientes del espectro.

$$\Delta f = f_s/N \quad (11)$$

Visto cómo poder obtener el espectro de una señal a partir de su versión muestreada en el tiempo, puede surgirnos la curiosidad de si el proceso puede realizarse a la inversa, es decir, si a partir del espectro podemos obtener la señal digital. La respuesta es sí, y gracias a la DFT en su versión inversa o DFT^{-1} .

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[k] e^{j \frac{2\pi kn}{N}}, \quad k = 0, \dots, N-1 \quad (12)$$

3.2 Autocorrelación

La autocorrelación en teoría de la señal es una función que define cuánto se parece una señal a otra igual a ella, pero desplazada en el tiempo [10]. Siendo $x[n]$ nuestra señal temporal muestreada, y k el número de muestras que desplazamos la señal, la autocorrelación $R_x[k]$ se calculará de acuerdo a la siguiente ecuación:

$$R_x[k] = \sum_{n=-\infty}^{\infty} x[n]x[n+k] \quad (13)$$

Dicha función presenta algunas propiedades a tener en cuenta, como son (véase [10]):

- $R_x[k] = R_x[-k]$
- $R_x[k] \leq R_x[0]$
- $G_x[f] = TF\{R_x[k]\}$ $G_x[f] \rightarrow$ Densidad espectral de potencia
- $R_x[0] = \sum_{n=-\infty}^{\infty} x^2[n] \rightarrow$ Potencia de la señal

A la hora de extraer la frecuencia fundamental de las notas musicales analizadas en este proyecto se ha utilizado el cálculo de la autocorrelación.

Para una señal periódica la función de autocorrelación mostrará picos en los múltiplos del periodo fundamental de dicha señal, por lo que será de interés localizar el pico siguiente al situado en el origen. En la Figura 3.3 se muestra una sección de la función de autocorrelación de un seno de frecuencia 400 Hz con el que se han hecho las pruebas:

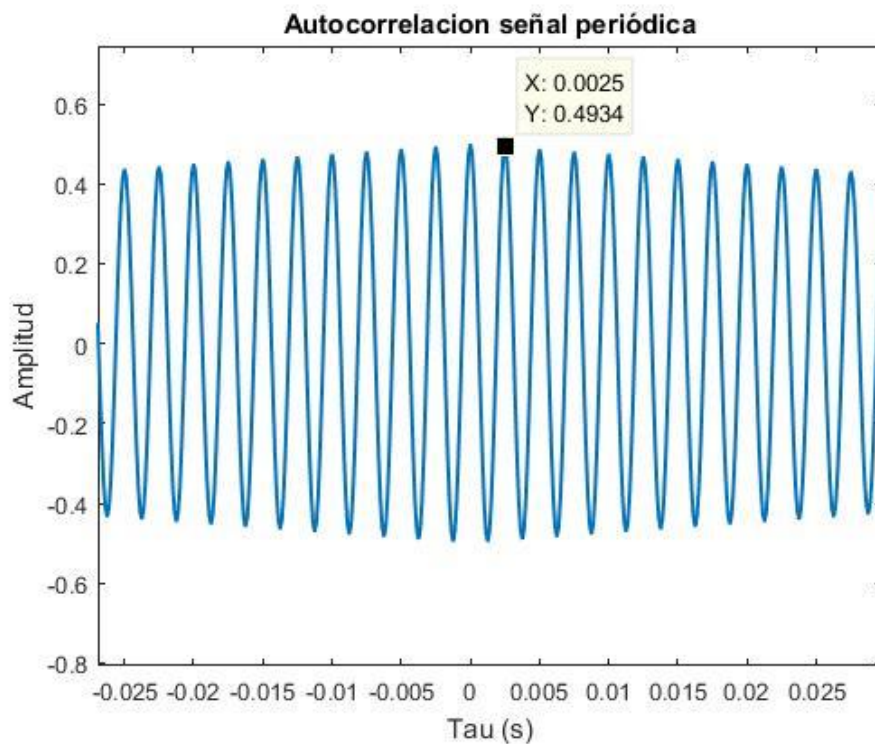


Figura 3. 3: Autocorrelación señal periódica

Como vemos $\tau = 0.0025$, por lo que la frecuencia $f = \frac{1}{\tau} = 400 \text{ Hz}$.

Nuestras señales musicales no son señales periódicas puras, por lo que su función de autocorrelación tendrá otra forma distinta, aun así, los picos de mayor amplitud seguirán estando en los múltiplos del periodo fundamental, y será de interés localizar el pico de mayor amplitud que no sea el situado en el origen.

En la Figura 3.4 se muestra la función de autocorrelacion de la nota Do (2) tocada por un violonchelo, señalándose en ella la posición del segundo pico mayor.

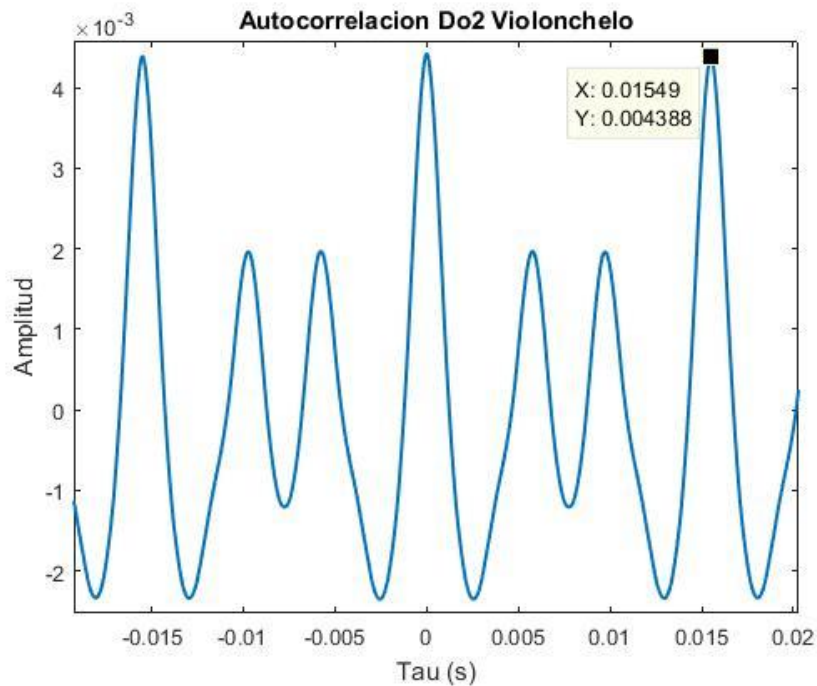


Figura 3. 4: Autocorrelación Do (2) Violonchelo

En este caso $\tau = 0.01549 \text{ s}$, por lo que la frecuencia $f = \frac{1}{\tau} = 64.5577 \text{ Hz}$. La tabla de frecuencias nos dice que un Do en segunda octava tiene una frecuencia teórica de 65.4 Hz, por lo que se podría decir que estamos en lo correcto.

Serán diversos los factores que intervendrán negativamente o positivamente en la medida de la frecuencia mediante la autocorrelación:

- Factores relacionados con la propia medida
 - Frecuencia de la nota (a frecuencias altas mayor imprecisión)
 - Número de muestras escogidas para la medida
- Factores relacionados con el propio instrumento:
 - Pericia del músico

- Forma de interpretar
- No linealidad de los instrumentos

3.3 Enventanado

La aplicación de la Transformada discreta de Fourier, en este caso concreto para señales sonoras, tendrá sentido si se aplica poco a poco en el tiempo, pues una señal no se presenta de igual manera en todos los instantes, sino que varía en cada uno de ellos, por lo que para saber cómo varían sus características en el tiempo será necesario aplicar la DFT a pequeñas porciones de esta.

Es lo que conocemos como enventanado o truncado de la señal, que nos permite llevar a cabo un procesamiento de señal variante en el tiempo. Matemáticamente, el efecto de ventana se plasma en la siguiente ecuación, donde nuestra señal discreta $x[n]$, suma de dos tonos de distintas frecuencias (f, f' respectivamente) es truncada por una función ventana $w[n]$ dando lugar a la señal enventanada $v[n]$ (véase [12]).

$$v[n] = A_1 w[n] \cos(2\pi f n) + A_2 w[n] \cos(2\pi f' n) \quad (14)$$

Si hiciéramos la transformada de Fourier de esta señal enventanada, obtendríamos el espectro de la señal ventana desplazado a las frecuencias f y f' .

La elección del tipo de ventana $w[n]$ y su tamaño no es tarea fácil, y dependerá fundamentalmente del tipo de señal con el que estemos tratando y el objetivo que queramos conseguir.

3.3.1 Longitud

Una ventana de análisis de longitud L en muestras será desplazada a lo largo de nuestra señal para ir cogiendo pequeñas porciones de esta y ser analizadas. Pues bien, el desplazamiento en muestras Q de dicha ventana se denominará resolución temporal y su duración Δt vendrá determinada por la siguiente ecuación, conociendo la frecuencia de muestreo f_s [23].

$$\Delta t = Q / f_s \quad (15)$$

Por otro lado, y a modo de recordatorio del capítulo anterior, la resolución espectral Δf dependerá de la longitud en muestras escogida para la ventana de análisis, siendo concretamente el cociente de la frecuencia de muestreo f_s y el número de muestras L .

Deducimos pues, que si nuestra ventana es desplazada tantas muestras como muestras tiene la ventana ($Q = L$), a mayor longitud de ventana, mayor resolución temporal y menor resolución en frecuencia. Ocurrirá lo contrario si la longitud es menor. Esto nos hace llegar a la conclusión de que no existe un método concreto que nos haga elegir eficazmente la longitud de una ventana para obtener a la vez buenos resultados en tiempo y en frecuencia. (Véase [23]).

Ahora bien, ¿y si desplazamos la ventana un número de muestras menor a la longitud de esta? Estaríamos en el caso donde ($Q < L$) y las ventanas de análisis se solaparían [23]. En este caso, mejoraríamos la resolución temporal sin necesidad de empeorar la resolución espectral, pues el número de desplazamientos en muestras es menor. Véase Figura 3.5.

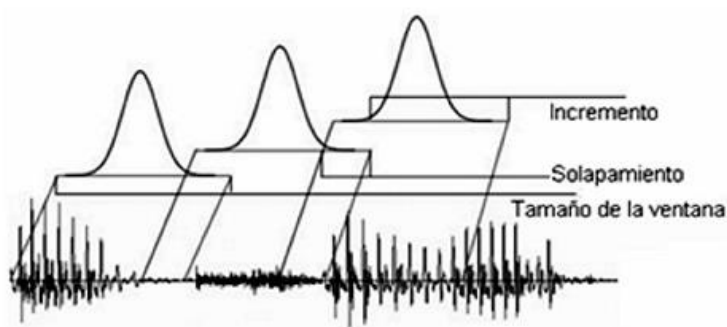


Figura 3. 5: Explicación gráfica de enventanado con solapamiento

En la parte de procesado de este proyecto, donde trabajo con las notas musicales provenientes de varios instrumentos, yo he llevado a cabo un enventanado de este tipo que ha mejorado mucho mis resultados. El objetivo principal de este bloque de procesado fue el de obtener la frecuencia fundamental o pitch de cada nota, y para el cual la ventana de análisis elegida fue de 2048 muestras que se solapaba un 50 % con su anterior (desplazábamos la ventana cada 1024 muestras).

Para este caso el tamaño de ventana fue fijo fuera cual fuera la nota musical tocada, pero la lógica de actuación fue distinta en la segunda parte de este bloque de procesado, donde tuve que calcular la potencia relativa de los diez primeros armónicos de la señal. Fijé un tamaño de ventana en función de la frecuencia fundamental teórica de la nota, calculada

anteriormente, y es que de esta manera mejoraban mucho mis resultados. La explicación de esta actuación es debida a que notas más graves (frecuencias bajas) varían más despacio en el tiempo, por lo que necesitaríamos una ventana de análisis mayor para ver su variación total. Por el contrario, las notas más agudas (frecuencias altas) tienen un efecto más acusado y varían más rápido en el tiempo, por lo que el tamaño de ventana debía ser menor. El porcentaje de solapamiento para este caso fue también del 50 %

Concretamente los tamaños fueron los siguientes:

- Para notas musicales de frecuencia fundamental menor a 500 Hz, el tamaño de ventana ha sido de $6 \cdot 2048$ muestras, (desplazábamos la ventana cada $3 \cdot 2048$ muestras).
- Para notas musicales de frecuencia fundamental mayor a 500 Hz, el tamaño de ventana ha sido de 2048 muestras, (desplazábamos la ventana cada 1024 muestras).

Como vemos, los valores elegidos para el tamaño de la ventana son potencia de dos, pues el cálculo de la DFT es especialmente eficaz para dichos valores (véase [11]).

3.3.2 Forma o suavizado

Existen diferentes tipos de ventanas diferenciados por su forma o suavizado. El tipo de ventana más simple es la ventana rectangular, definiéndose matemáticamente como 1 para un intervalo de $[0, L]$ y cero el resto.

Entre otras encontramos la ventana de hanning, hamming, blackman y bartlett o triangular cuyo suavizado se muestra en la gráfica de la Figura 3.6 a continuación.

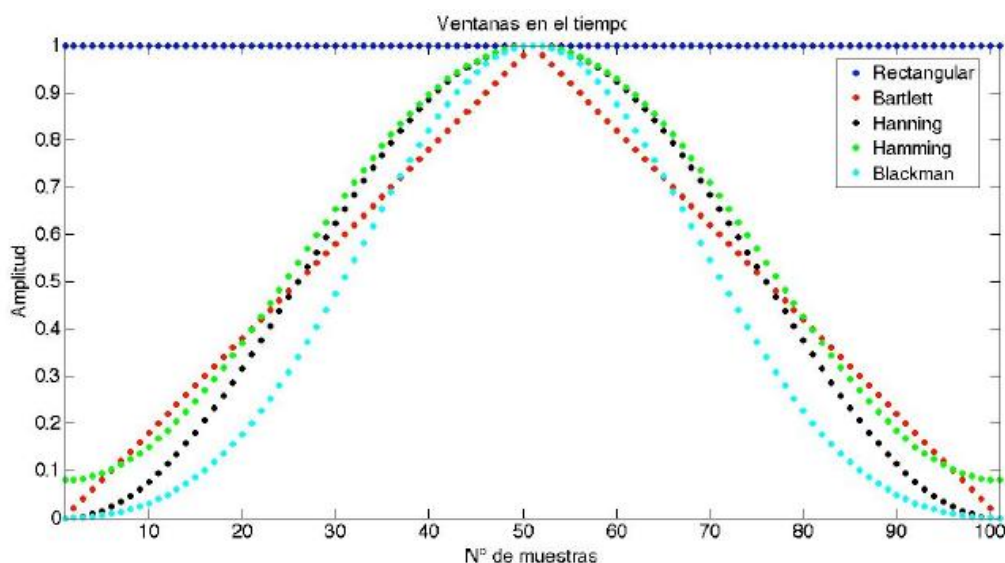


Figura 3. 6 [12]: Tipos de ventanas de análisis

Todos estos tipos de ventana introducirán distorsión al espectro, pero no todas de la misma manera e intensidad. Respecto a esto, la ventana rectangular es la más ineficiente, pues el hecho de multiplicar por cero el resto de señal que no entra en el intervalo de la ventana produce un paso a cero demasiado abrupto, por lo que se puede originar en el espectro frecuencias no deseadas. Esto se soluciona con la aplicación de otro tipo de ventana, cuyo suavizado no sea tan abrupto y sepa escalar de forma suave la amplitud de nuestra señal hasta que se haga cero por completo. En este proyecto se ha aplicado una ventana de hanning, cuya forma presenta dichas características, tal y como se representa en la figura de arriba.

En la Tabla 3.1 se representan los valores de los efectos típicos del enventanado, como son la resolución y el grado de fugas, para las distintas formas de ventana.

Tipo	A_{LP}	N_{LS} (dB)
Rectang.	2/L	13
Triang.	4/L	26
Hanning	4/L	32
Hamming	4/L	42

Tabla 3. 1 [12]: Resolución y grado de fugas para las distintas ventanas de análisis

La resolución está especialmente relacionada con el ancho del lóbulo principal de la TF de la ventana en concreto, siendo el grado de fugas la diferencia entre la amplitud del lóbulo principal y la amplitud del segundo mayor de la TF de la ventana, expresado en dB [12]. Como podemos ver, el ancho del lóbulo principal es menor o más estrecho en una ventana rectangular, por lo que este tipo de función será más adecuada para señales cuyos tonos sean simples, y así localizar eficazmente las frecuencias. Para señales sonoras complejas, como son las notas musicales de un instrumento, donde la energía no está localizada de manera clara en determinadas frecuencias, es mejor utilizar ventanas como la de hamming o hanning [23].

CAPÍTULO 4. CONCEPTOS RELEVANTES DE LA COMPUTACIÓN NEURONAL

4.1 Redes Neuronales Artificiales (ANN)

4.1.1 Marco biológico

Como su propio nombre indica, una red neuronal artificial es aquel sistema que se modela inspirándose de la estructura y funcionamiento de nuestro cerebro humano, con el fin de conseguir algo lo más parecido a este, capaz de realizar funciones complejas y aprender.

Empezando por una breve explicación sobre la composición biológica de nuestro cerebro, entenderemos mucho mejor el funcionamiento de las redes neuronales artificiales.

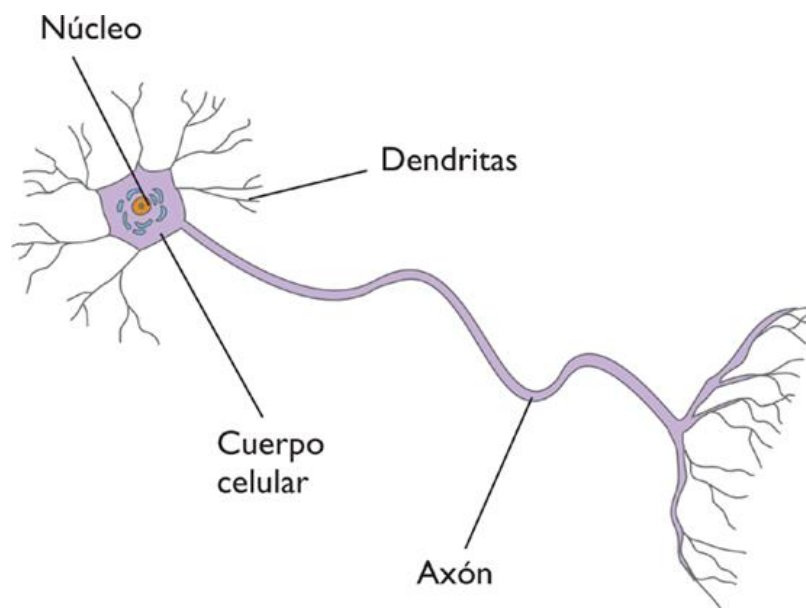


Figura 4. 1: Neurona biológica

Nuestro cerebro se compone de neuronas, célula representada en la Figura 4.1 cuyo cuerpo no excede de 5 micras de radio, caracterizado por unas pequeñas ramificaciones llamadas dendritas que hacen de entrada de la información, la cual es procesada por dicha célula y enviada al punto de interconexión con otra neurona (sinapsis) mediante el axón.

Las señales que intervienen en el proceso de comunicación entre neuronas en nuestro cerebro, pueden ser de naturaleza eléctrica, como por ejemplo la que genera la propia

neurona y que se transporta por el axón, y de naturaleza química, como las generadas en la sinapsis por sustancias transmisoras llamadas neurotransmisores [13].

El proceso de generación de señales por parte de las neuronas es complejo de explicar. Se podría decir que está especialmente relacionado con el nivel de iones de sodio y potasio concentrados en la membrana celular, pues dependiendo de dicha concentración se puede dar una diferencia de potencial que facilite el envío de información [13].

4.1.2 Marco artificial

En el ámbito artificial, las neuronas serán denominadas como nodos, y serán los dispositivos elementales del proceso. Habrá tres tipos de neuronas artificiales o nodos:

- Neuronas de capa de entrada: Las que son alimentadas con información externa
- Neuronas de capa oculta: procesan la información que les llega de las neuronas de entrada
- Neuronas de capa de salida: dan el resultado del proceso

Y cada una de estas unidades o nodos se caracterizarán por parámetros como el estado de activación, función de salida y señal de salida. Dichos parámetros están relacionados entre sí, pues la señal de salida de una determinada neurona artificial será resultado de transformar el estado actual de dicha neurona por medio de la función de salida de esta.

Si representásemos gráficamente todo lo anterior quedaría algo como lo representado en la Figura 4.2:

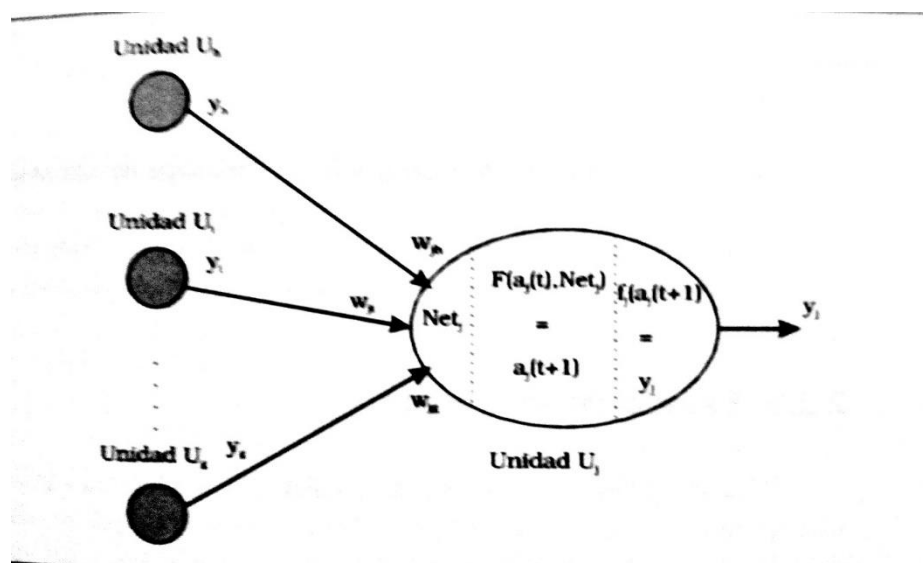


Figura 4. 2 [13]: Neurona artificial

De cada unidad U_i tendremos una señal de salida y_i , que combinadas con los pesos w_{ji} de cada canal unidireccional darán para el nodo U_j donde desembocan, una entrada total Net_j . Esta unidad vendrá caracterizada por una función de activación F que determinará el nuevo estado de activación $a_j(t + 1)$, que a su vez será transformado por la función f_j de salida de dicha unidad para producir una nueva salida y_j

$$Net_j = \sum_i y_i w_{ji} \quad (16)$$

4.1.3 Funciones de activación

Las neuronas que componen las capas ocultas y la capa de salida de una red neuronal vienen caracterizadas por una función de activación F , que podrá ser lineal o sigmoideal. De este último caso las habrá de dos tipos, logística (valores acotados entre 0 y 1) o tangente hiperbólica (valores acotados entre -1 y 1).

En las Figuras 4.3, 4.4 y 4.5 se muestran gráficamente la forma de estas funciones de activación:

$$y = x \quad (17)$$

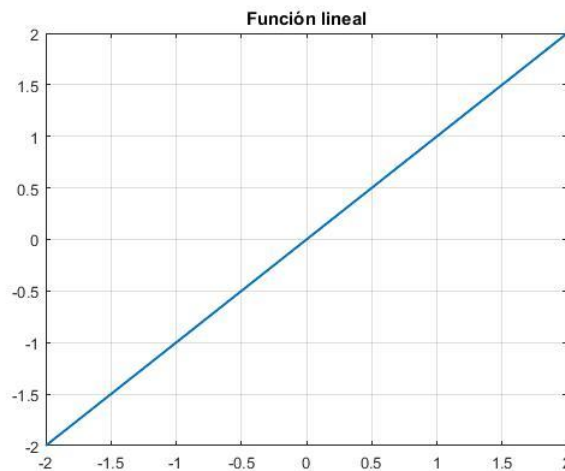


Figura 4. 3: Función de activación lineal

$$y = \frac{1}{1 + e^{-t}} \quad (18)$$

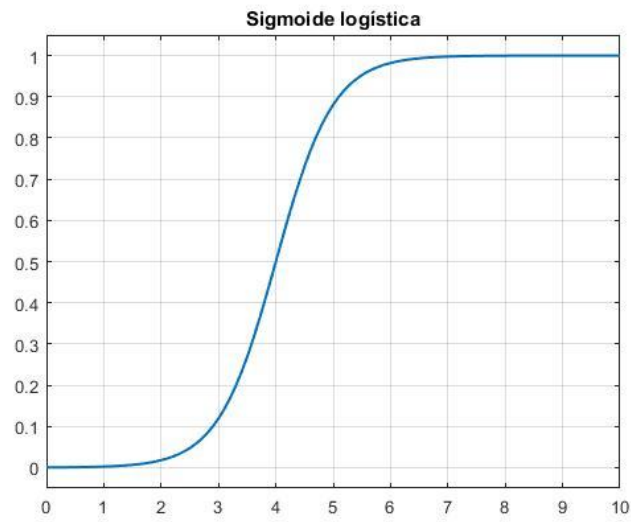


Figura 4. 4: Función de activación sigmoide logística

$$y = \frac{e^t - e^{-t}}{e^t + e^{-t}} \quad (19)$$

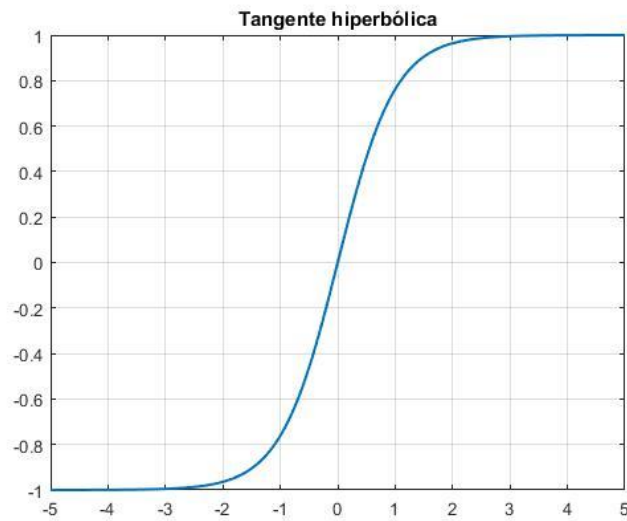


Figura 4. 5: Función de activación sigmoide tangente hiperbólica

4.1.4 Topologías

Hemos visto anteriormente que los nodos de una red neuronal pueden disponerse en capas, y que según la capa a la que pertenezcan llevarán a cabo una función u otra.

El cómo se dispongan y se organicen en cada una de estas capas e incluso el número de capas que tenga la red, dará nombre a una estructura o topología de red, que será clave para llevar a cabo un proceso. Diferenciaremos entre redes de una sola capa o redes monocapa, y redes de más de un nivel o multicapa.

Para una topología de un solo nivel, las conexiones entre neuronas podrán ser laterales o por otro lado propias, que significa que la salida de una neurona vuelve para ser posteriormente entrada.

Para una topología de varios niveles las conexiones serán hacia delante o feedforward, que viene a ser que las neuronas de una capa posterior reciben entradas de todas las neuronas de la capa anterior, o conexiones hacia atrás o feedback, que viene a ser lo contrario. La red neuronal utilizada en nuestro proyecto será de este tipo tal y como se muestra en la Figura 4.6.

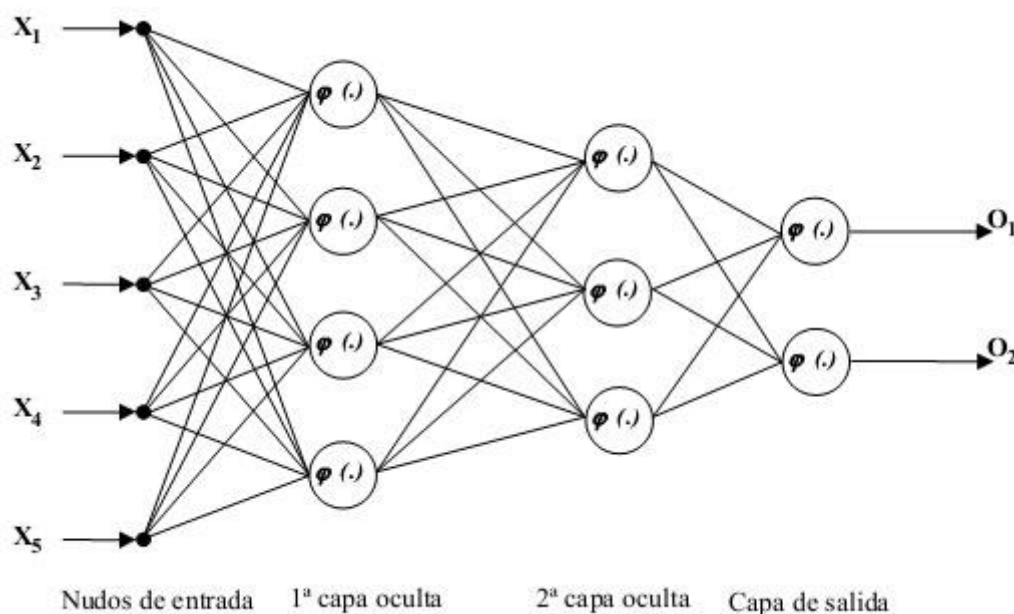


Figura 4. 6: Topología multicapa con conexión feedforward

4.1.5 Aprendizaje

Llevado al ámbito humano, aprender es saber de algo que antes desconocíamos, adquirirlo o fijarlo en nuestra memoria de forma eficiente para poder llevarlo a cabo bien siempre que se quiera, o simplemente recordarlo en menos tiempo que cuando no se sabía. En nuestra vida diaria, estamos continuamente aprendiendo cosas, y sobre todo a base de errores.

4.1.5.1 Aprendizaje supervisado

Para una red neuronal artificial, aprender consistirá en modificar los pesos de cada una de sus conexiones, según una regla de aprendizaje. Las hay de dos tipos:

- Reglas de aprendizaje supervisado → Redes neuronales de aprendizaje supervisado
- Reglas de aprendizaje no supervisado → Redes neuronales de aprendizaje no supervisado

Yo me centraré en explicar el primer tipo de redes neuronales, pues es el que se adapta a mi proyecto. La diferencia entre ambos grupos es clara, y es que mientras que en las de aprendizaje no supervisado no actúa ningún agente externo que controle ese aprendizaje, en las de supervisado sí, pues en estas se sabe a priori las salidas que deseamos obtener, y a partir de eso se inicia la modificación oportuna de pesos.

Habrán tres formas de llevar a cabo de manera supervisada un aprendizaje:

- Aprendizaje por corrección de error

Los pesos de las conexiones se modificarán según la diferencia entre los valores de entrada y salida. Véase Figura 4.7.

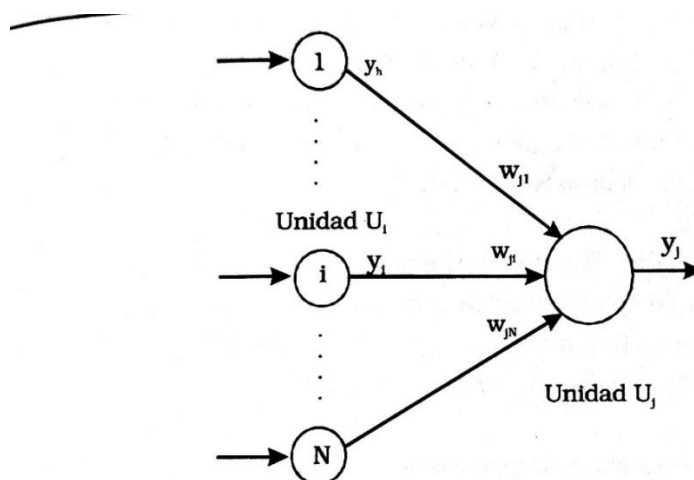


Figura 4. 7 [13]: Pesos según aprendizaje por corrección de error

Una expresión válida para este tipo de aprendizaje podría ser (véase [13]):

$$\Delta w_{ji} = \alpha y_i (d_j - y_j) \quad (20)$$

α : Factor de aprendizaje cuyo valor estará entre 0 y 1

y_i : Valor de salida para el nodo i

Δw_{ji} : Variación del peso (actual – anterior)

d_j : Valor de salida deseado para el nodo j

y_j : Valor de salida para el nodo j

Un ejemplo de algoritmo de aprendizaje de este tipo es el denominado como algoritmo de retropropagación de error (error backpropagation), utilizado en topologías multinivel con más de una capa oculta.

- Aprendizaje por refuerzo

Menos eficiente que la anterior y basada en la probabilidad. Consiste en no fijar de forma exacta la salida deseada, si no imponer una respuesta cercana a dicha meta, y que una señal de refuerzo (positiva o negativa) indique si la salida es cercana o lejana a lo que queremos, respectivamente [13].

- Aprendizaje estocástico

Para este tipo de aprendizaje se suele hablar en términos de energía para decir si la red aprendió o no. Se trata de un modelo probabilístico cuya función es cambiar de manera aleatoria los pesos de las conexiones hasta conseguir un estado de mínima energía, que nos diría que se ajusta lo más posible a nuestro objetivo [13].

4.1.6 Clasificación de las redes neuronales según la información de entrada y salida

La información representada como entrada o salida de una red neuronal puede ser analógica o, por otro lado, binaria. Según este tipo de representación las redes neuronales podrán clasificarse en tres grandes grupos (véase [13]):

- Redes neuronales continuas, cuya entrada y salida son analógicas
- Redes neuronales híbridas, con entrada analógica y salida binaria
- Redes neuronales discretas, entradas y salidas binarias

CAPÍTULO 5. EXTRACCIÓN DE CARACTERÍSTICAS

Después de los aspectos teóricos que han supuesto el capítulo 2, 3 y 4 de este proyecto, toca adentrarse un poco más en el sistema concreto que hemos implementado y ver de una manera más práctica estos conceptos.

Como bien se introdujo en el resumen del proyecto, el sistema que aquí implementamos consta fundamentalmente de dos bloques:

Un primer bloque de procesado que analiza una ventana temporal y extrae características espectrales de las notas de cada instrumento, como la frecuencia fundamental MIDI de la nota y la potencia relativa de los 10 primeros armónicos. Además, este bloque sabrá verificar también de manera estadística si los datos son congruentes o no, descartando los posibles datos atípicos que encontremos, pues a la red neuronal no se le pasarán datos incorrelados.

Finalmente, en el segundo bloque entrenaremos una red neuronal específica que será capaz de darnos el tipo de instrumento que toca a partir de las características anteriores. Esta red neuronal, como se puede suponer tras explicar lo anterior, constará de 11 entradas (frecuencia MIDI y potencia relativa de los 10 primeros armónicos) y 4 salidas que darán lugar a una sola binaria de cuatro bits. Esto significa que la red tendrá capacidad de hasta reconocer 16 instrumentos.

Nosotros hemos trabajado concretamente con 4 instrumentos, tres de ellos de la familia de cuerda frotada, como son el violonchelo, la viola y el violín. El cuarto instrumento analizado ha sido una flauta travesera.

Para ello se nos proporcionó desde un principio archivos de audio que contenían una gran cantidad de notas que podían ser tocadas por estos instrumentos. Dichos archivos fueron manipulados mediante el software Audacity para extraer cada nota por separado y analizarla posteriormente. En las Figuras 5.1, 5.2, 5.3 y 5.4 podemos ver dicho ejemplo.

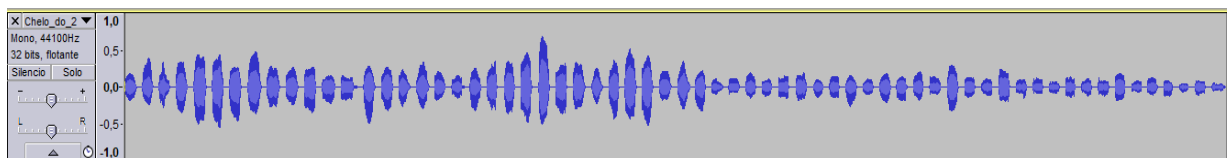


Figura 5. 1: Violonchelo Do2 – La5 (.wav)

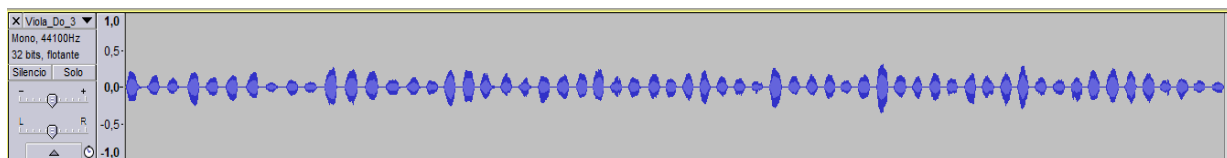


Figura 5. 2: Viola Do3 – Fa6 (.wav)

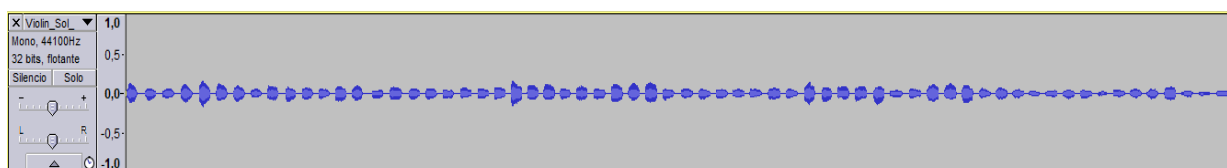


Figura 5. 3: Violín Sol3 – Mi7 (.wav)

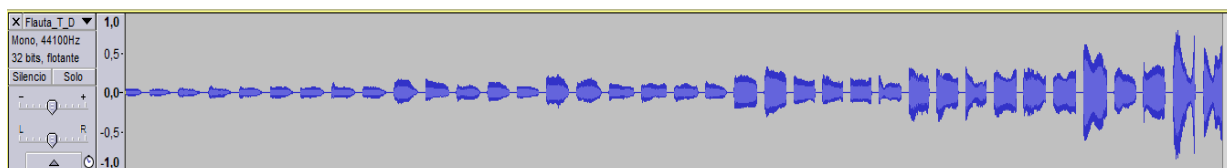


Figura 5. 4: Flauta travesera Do4 – Do7 (.wav)

En la carpeta de ANEXOS incluyo algunas de estas notas.

5.1 Bloque de procesado

5.1.1 Implementación del método de la autocorrelación. Cálculo de la frecuencia fundamental.

En primer lugar, a cada una de estas notas fue necesario extraerle su frecuencia fundamental, y tal y como se demostró gráficamente en el apartado 3.2, la autocorrelación nos podría resolver este trabajo. Consistía en detectar el pico de mayor amplitud en dicha función de autocorrelación que no fuera el situado en el origen. De esta manera obtendríamos el valor temporal τ , cuya inversa sería la frecuencia.

Se puede encontrar la implementación de este método en Matlab en la carpeta de ANEXOS. En la Tabla 5.1 se muestra tan solo la cabecera de la función que lo implementa y la explicación de esta:

```
function [ff]=auto(x,T,L,ventana,n_tramas,umbralCorte)
% Función que calcula la frecuencia fundamental mediante el método de
la autocorrelacion
%INPUTS:
    %x: señal de entrada (nota musical)
    %T: periodo de muestreo
    %L: tamaño de la ventana
    %ventana: tipo de ventana a utilizar
    %n_tramas: número de tramas en las que se dividirá la señal
    %umbralCorte: umbral de corte para la autocorrelacion
%OUTPUTS:
    %ff: vector con las frecuencias de cada trama
```

Tabla 5. 1: Cabecera de la función ‘auto.m’

5.1.2 Estudio estadístico. Filtrado paso bajo

La anterior función nos devolvía los valores de frecuencia obtenidos por cada trama analizada de la señal, por lo que seguidamente fue necesario hacer un estudio estadístico de las muestras que íbamos tomando. Por muestras nos referimos a la frecuencia central de la nota.

Calculamos en primer lugar la media y desviación típica de dicho vector de frecuencias, y en función de la desviación típica obtuvimos un valor de frecuencia medio:

- Si la desviación típica era igual o mayor a 0.5, el valor de frecuencia medio sería el de hacer la media de dicho vector de frecuencias suavizado mediante un filtro de mediana (filtro paso bajo). El filtrado de este tipo consistirá en asignar a cada muestra el valor de la mediana local para un conjunto de muestras vecinas, cambiando solo aquellos valores que no correspondan a dicha mediana [17] (valores incorrelados). De esta manera, los valores de frecuencia atípicos se ajustan adecuadamente. Véase ejemplo en Figura 5.5.

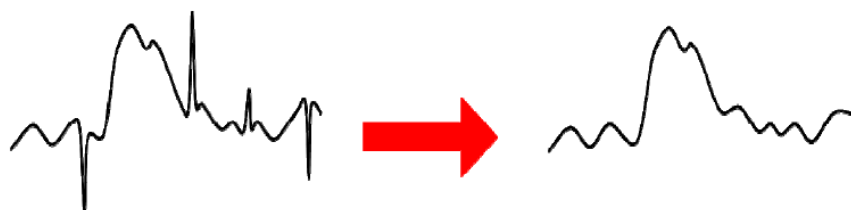


Figura 5. 5 [17]: Ejemplo visual de un filtrado de mediana

- Por el contrario, si la desviación típica era menor a 0.5, nos quedábamos con la media del vector obtenido. No será necesario el filtrado.

Una vez aplicada la lógica estadística, comparamos el valor medio de frecuencia obtenido con los valores de frecuencia teóricos conocidos y que forman la tabla de frecuencias MIDI. Nos quedamos con el valor teórico más cercano y que no excedía más de $\frac{1}{4}$ de semitono por arriba o por debajo del valor medio calculado. Las notas que no seguían este criterio eran descartadas, pues suponíamos que el intérprete desafinaba.

La Tabla 5.2 muestra la cabecera de la función en Matlab que aplica todo lo dicho anteriormente. Se podrá encontrar completamente esta función en la carpeta de ANEXOS.

```
function[new_media,new_des,frecuencia,midi]=filtro_mediana(ff,media,des,n_tramas)
%Función que calculará la frecuencia midi más cercana (no más de 1/4 de
%semitono por arriba o por abajo) a la frecuencia media, calculada a su
vez mediante un filtrado paso bajo (filtro de mediana).
%INPUTS:
% ff: frecuencias de trama
% media: media del vector de frecuencias
% des: desviación típica del vector de frecuencias
%OUTPUTS:
% new_media: nuevo valor de la media tras la mejora estadística
% new_des: nuevo valor de la desviación típica tras la mejora
estadística
% frecuencia: frecuencia teórica en Hz
% midi: número de frecuencia MIDI
```

Tabla 5. 2: Cabecera de la función ‘filtro_mediana.m’

5.1.3 Densidad espectral de potencia. Cálculo de la potencia relativa de los armónicos

Una vez que almacenamos el valor teórico de frecuencia, calculamos a partir de él la amplitud relativa de los armónicos que componen la nota musical en cuestión. Recordamos que los armónicos de la señal se sitúan en el espectro en frecuencias múltiplos de la frecuencia fundamental f_0 .

Aun así, no todo es tan sencillo, pues la relación de armónicos para las notas que componen la tesitura de un instrumento no es especialmente lineal.

Podemos ver cierta no linealidad en el siguiente experimento, donde gráficamente en la Figura 5.6, mediante el espectro de la señal musical Do (4) proveniente de una flauta travesera, vemos que las frecuencias de cada armónico no se sitúan en los múltiplos exactos de la frecuencia fundamental (experimento basado en ejemplos similares explicados en [14] y [15]).

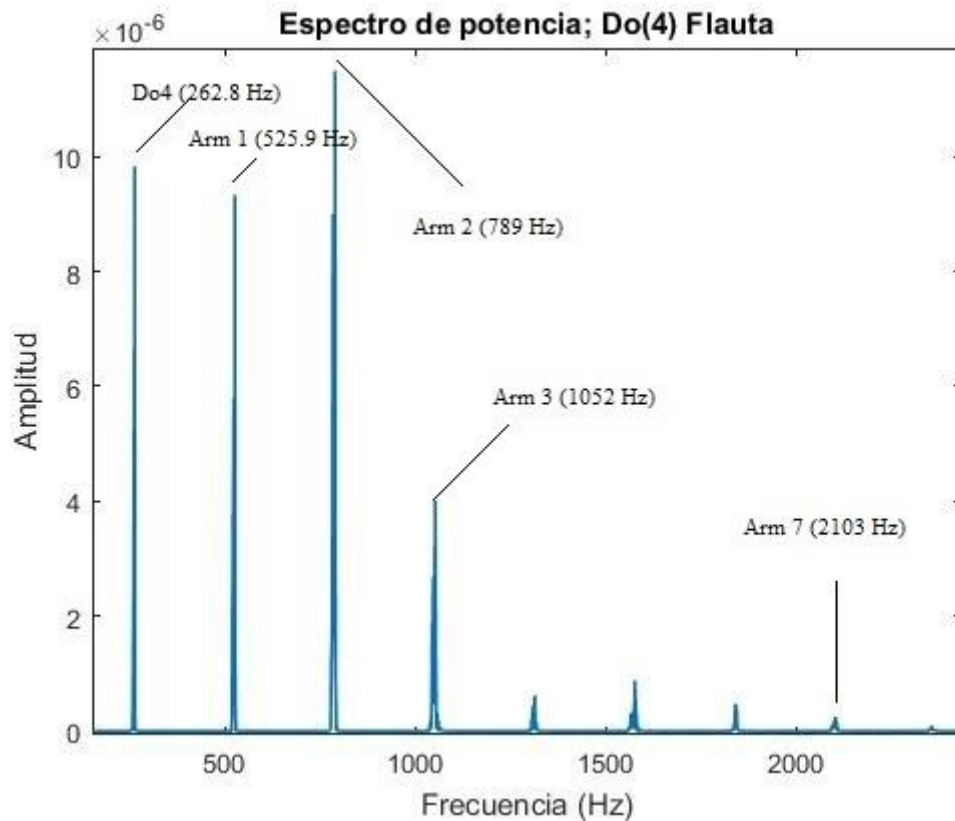


Figura 5. 6: Espectro nota Do (4) Flauta travesera

Según el cálculo de la densidad espectral de potencia de la nota Do (4) proveniente de una flauta travesera, la frecuencia fundamental es de 262.8 Hz. Las frecuencias de los armónicos deberían situarse en múltiplos exactos de dicha frecuencia, pero vemos que no ocurre exactamente así. Aun así, los errores dados son prácticamente despreciables.

- Para el armónico 1: $525.6/525.9 \rightarrow \text{Error} \approx 0.05 \%$
- Para el armónico 2: $788.4/789 \rightarrow \text{Error} \approx 0.07 \%$
- Para el armónico 3: $1051.2/1052 \rightarrow \text{Error} \approx 0.076 \%$

Esta no linealidad, denominada inarmonicidad (véase [8]) se debe a factores como el propio instrumento, la forma de tocarlo y de la nota en concreto. El instrumento que analizamos es un instrumento de viento, y hemos de decir que esta no linealidad es más propia en instrumentos de cuerda percutida, como el piano y la guitarra, ocasionando acordes que no suenan bien al oído de quien los escucha.

La Tabla 5.3 muestra la cabecera de la función en Matlab que nos ha permitido obtener las amplitudes de los armónicos de las notas musicales analizadas. La función completa podrá verse en la carpeta de ANEXOS.

```
function [R]=armonicos(x,fs,L,n_tramas,ventana,frecuencia)
% Función que calcula las amplitudes de los 10 primeros armónicos,
% normalizadas respecto a la amplitud del primer armónico
%INPUTS:
    %x: señal de entrada
    %L: longitud de la ventana
    %n_tramas: número de tramas
    %ventana: tipo de ventana
    %frecuencia: frecuencia teórica midi
%OUTPUTS:
    %R: matriz 10 x n_tramas que recoge las amplitudes de los
    armónicos en cada trama.
```

Tabla 5. 3: Cabecera de la función ‘armonicos.m’

El fundamento de esta función fue el de ir buscando el pico mayor o máximo de potencia dentro de un intervalo concreto. Este intervalo tendría su mínimo ‘low’ ligeramente por debajo de la frecuencia fundamental, y su máximo ‘high’ por encima de esta. Estos parámetros (low, high) se fueron incrementando según el armónico concreto que estábamos buscando.

Para no detectar máximos no deseados de la densidad espectral de potencia, previamente se fijó un umbral para eliminar parte de ellos y así obtener resultados fiables.

5.2 Bloque de red neuronal

Después del bloque de procesado tocó ver si realmente lo que habíamos extraído de cada señal musical era válido para que una red neuronal supiera darnos una respuesta acertada.

5.2.1 Entrenamiento de la red

Para ello, partimos de una base de datos provenientes del conjunto de notas analizadas, donde el 80 % de estos datos se destinaron a entrenar la red y el 20 % restante para hacer las pruebas de validación o test.

Unidad USB (F:) > BaseDatos_resultados			
Nombre	Fecha de modifica...	Tipo	Tamaño
ENTRENAMIENTO	23/10/2017 14:46	Carpeta de archivos	
TEST	04/11/2017 10:54	Carpeta de archivos	

Figura 5. 7: Base de datos

	89	90	91	92	93	94	95	96	97	98	99	100
1	76	76	76	76	76	76	76	76	76	76	76	76
2	1	1	1	1	1	1	1	1	1	1	1	1
3	0.6337	0.6499	0.6612	0.6531	0.6470	0.6378	0.6456	0.6369	0.6329	0.6261	0.6288	0.6213
4	0.4965	0.5349	0.4934	0.5320	0.5698	0.5721	0.5386	0.5249	0.5104	0.4812	0.5136	0.4954
5	0.1479	0.1559	0.1269	0.1504	0.1625	0.1593	0.1457	0.1432	0.1310	0.1258	0.1500	0.1470
6	0.0248	0.0231	0.0202	0.0236	0.0253	0.0240	0.0227	0.0215	0.0190	0.0168	0.0202	0.0180
7	0.0035	0.0043	0.0017	0.0027	0.0043	0.0054	0.0047	0.0051	0.0032	0.0020	0.0024	0.0015
8	0.0025	0.0019	0.0027	0.0026	0.0020	0.0020	0.0019	0.0021	0.0024	0.0028	0.0028	0.0027
9	0.0071	0.0063	0.0066	0.0067	0.0065	0.0061	0.0063	0.0061	0.0061	0.0067	0.0062	0.0056
10	6.3291e-04	6.9878e-04	6.7429e-04	5.1436e-04	6.4233e-04	6.4821e-04	7.3638e-04	7.9221e-04	8.9805e-04	9.1182e-04	7.1903e-04	7.6340e-04
11	3.9067e-05	6.2441e-05	3.6057e-05	8.2555e-05	1.3052e-04	1.6654e-04	1.5120e-04	1.2418e-04	1.3284e-04	1.0921e-04	7.0221e-05	6.3148e-05
12												

Figura 5. 8: Contenido de datos de entrenamiento

Las Figuras 5.7 y 5.8 muestran la estructura y el contenido de la base de datos creada para la prueba del sistema. Las dos carpetas mostradas en la Figura 5.7 contienen archivos (.mat) donde cada uno contiene características de una nota concreta de un determinado instrumento. En la Figura 5.8 se muestra el contenido de uno de estos archivos (.mat), concretamente de la nota MI (5) (MIDI = 76) proveniente de un violín. Las columnas representan las tramas en las que se ha dividido la señal para hacer un estudio de las características que extraemos a lo largo del tiempo, y las filas los parámetros que extraemos. La primera fila contiene la frecuencia MIDI, y el resto las potencias relativas de los diez primeros armónicos.

Para el entrenamiento de la red se cargaron a la vez todos los archivos (.mat) guardados en la carpeta 'ENTRENAMIENTO' y se concatenaron en una única matriz de entrada denominada 'instrumentos'. Además, ya que el entrenamiento fue llevado a cabo de forma supervisada, se creó una matriz 'target' con los datos de salida deseados para cada trama o columna de la matriz 'instrumentos'. Recordamos que nuestra red tendría 4 salidas

(1 binaria de 4 bits), de manera que cada instrumento se codificaba a la salida de la red con la siguiente combinación binaria:

- Violonchelo: ‘0000’
- Viola: ‘0001’
- Violín: ‘0010’
- Flauta travesera: ‘0011’

La función en Matlab que llevó a cabo la carga de todos los datos y la creación de los targets o datos de salida deseados se ha llamado ‘leer.m’ y cuya cabecera se presenta a continuación en la Tabla 5.4. La función completa está en la carpeta de ANEXOS. En la Figura 5.9 podemos el resultado de la ejecución.

```
function [instrumentos,instrumentos_bin,target]=leer()
% Función que almacenta en una sola matriz los datos característicos
% extraídos en el preprocesado para el entrenamiento de la red neuronal
%OUTPUTS:
%instrumentos: matriz en la que se concatenan la frecuencia y la
%amplitud de los 10 primeros armonicos de cada nota de cada
%instrumento
%target: matriz con salidas deseadas
```

Tabla 5. 4: Cabecera función ‘leer.m’

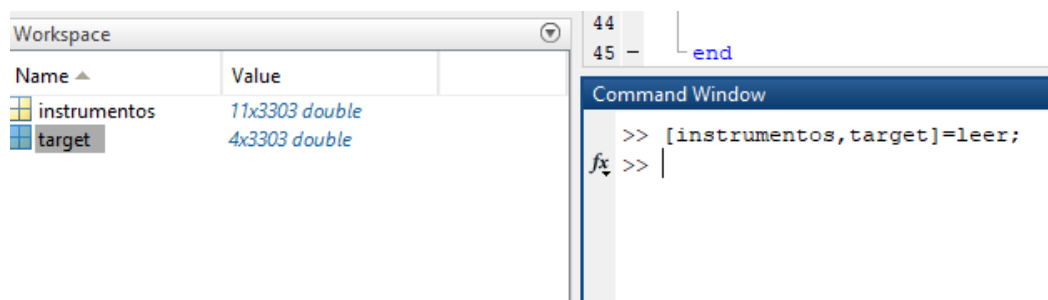


Figura 5. 9: Ejecución de la función ‘leer.m’

En las Figuras 5.10 y 5.11 se han capturado parte de los datos que contienen las variables ‘instrumentos’ y ‘target’ anteriores:

	526	527	528	529	530	531	532	533	534	535	536	537	538
1	53	53	53	53	53	53	55	55	55	55	55	55	55
2	1	1	1	1	1	1	0.0736	0.0806	0.0518	0.0549	0.0544	0.0553	
3	0.0697	0.0670	0.0760	0.1074	0.0759	0.0500	1	1	1	1	1	1	1
4	0.0191	0.0176	0.0187	0.0253	0.0167	0.0284	0.2702	0.1893	0.1750	0.1550	0.2120	0.2072	0.2297
5	0.0020	0.0019	0.0019	0.0031	0.0062	0.0104	0.1160	0.1565	0.1157	0.1128	0.1502	0.1641	0.2425
6	2.2888e-04	2.3079e-04	2.7144e-04	7.1027e-04	0.0012	0.0012	0.2869	0.2324	0.1494	0.0992	0.1464	0.1099	0.1625
7	9.2084e-04	8.7813e-04	9.0468e-04	0.0011	0.0018	0.0036	0.3434	0.2071	0.1346	0.1101	0.1087	0.1154	0.1752
8	5.7823e-05	5.8233e-05	1.2513e-04	4.1968e-04	0.0014	0.0054	0.0274	0.0337	0.0254	0.0160	0.0282	0.0184	0.0350
9	8.4885e-04	5.7724e-04	3.5190e-04	3.8951e-04	5.5566e-04	0.0013	0.0248	0.0292	0.0090	0.0051	0.0037	0.0029	0.0036
10	7.9599e-05	9.6428e-05	1.4967e-04	1.1314e-04	5.8124e-04	0.0010	0.0216	0.0091	0.0058	0.0126	0.0214	0.0182	0.0103
11	1.6998e-06	5.0912e-07	6.5422e-07	5.2284e-06	2.8630e-05	5.5792e-05	0.0687	0.0537	0.0408	0.0423	0.0569	0.0664	0.1307

Figura 5. 10: Datos de entrada (instrumentos.mat)

	527	528	529	530	531	532	533	534	535	536	537	538	539
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	1	1	1	1	1	1
5													
6													
7													
8													
9													
10													
11													
12													

Figura 5. 11: Datos de salida deseados (target.mat)

Como bien se ha explicado anteriormente y podemos ver, ‘instrumentos’ es una matriz que contiene las frecuencias y las potencias relativas de los armónicos de las notas analizadas de todos los instrumentos, es decir, que en esa misma variable hay datos de violonchelo, viola, violín y flauta travesera. Y la variable ‘target’ asignará a cada una de las tramas un código binario según el instrumento que sea. Esto se puede apreciar en dichas figuras, pues si nos damos cuenta, a partir de la trama 532 empiezan datos de otro instrumento, concretamente de viola (0001) tal y como se codifica en ‘target’.

5.2.2 Prueba de la red. Resultados

Llegados hasta aquí, solo faltaba crear la red, entrenarla y mostrar cuánto de eficaz podría ser identificando un instrumento u otro. Para ello hicimos diferentes pruebas con

diferentes modelos de red distintos. La eficacia de la que hablamos se definía como el tanto por ciento de acierto que lograba la red para un determinado instrumento. Si recordamos lo mostrado en la Figura 5.8, la información extraída que caracterizaba una determinada nota musical tocada por un instrumento, se organizaba en forma de matriz, pues bien, si dicha matriz hiciera de entrada de la red neuronal entrenada, la red devolverá una salida en forma de matriz también, cuyo número de columnas (tramas) será el mismo que las columnas de la matriz de entrada, donde cada una representará una combinación binaria según el instrumento que sea. Para el cálculo del porcentaje iremos contando las tramas acertadas comparando cada trama con la salida deseada, que será la combinación ('0000', '0001', '0010' o '0011') según el instrumento, y dividir dicho resultado entre las tramas totales. Esto se ha hecho con la función 'dameporcentaje.m' que podrá ser encontrada en la carpeta de ANEXOS.

Prueba 1: En un primer intento se partió de una base de datos de 101 notas diferentes. 31 provenientes de notas tocadas por un violonchelo, 22 de viola, 21 de violín y 27 de flauta travesera. Del total, 77 fueron destinadas al entrenamiento, y el resto para la validación. La red neuronal elegida fue una red multicapa con conexiones hacia adelante (feed-forward), 2 capas ocultas de 150 neuronas cada una. La función de activación para las neuronas de cada capa fue 'logsig' (sigmoide logística), y fue entrenada mediante el algoritmo 'trainrp' (backpropagation). Véase Tabla 5.5.

```
clear all;close all;clc
[instrumentos,instrumentos_bin,target]=leer; % Leemos parámetros de
entrenamiento
red=newff(minmax(instrumentos),[150 150
4],{'logsig','logsig','logsig'},'trainrp'); % Creamos red

red.trainParam.epochs=5000; % Máximo de iteraciones
red.trainParam.min_grad=1e-12; % Mínimo gradiente
view(red); % Visualizamos la red
[red tr]=train(red,instrumentos,target); % Entrenamos la red

save('red_150_trainrp.mat','red'); % Salvamos la red creada
```

Tabla 5. 5: Creación red neuronal Prueba 1

La función 'dameporcentaje.m' nos devuelve los siguientes porcentajes de acierto por instrumento como podemos observar en la Figura 5.12:

```

>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 1
El porcentaje de acierto para CHELO es del 96.9697 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 2
El porcentaje de acierto para VIOLA es del 63.1579 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 3
El porcentaje de acierto para VIOLIN es del 62.9032 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 4
El porcentaje de acierto para FLAUTA es del 93.1408 %
>> |

```

Figura 5. 12: Porcentajes de acierto en Prueba 1

En la Tabla 5.6 se han recogido los anteriores resultados en función de las características de la red utilizada. Como vemos, esta discrimina casi a la perfección entre el violonchelo y la flauta travesera, no siendo igual de eficaz para la viola y el violín, aunque tampoco son datos extremadamente malos.

	VIOLONCHELO	VIOLA	VIOLÍN	FLAUTA TRAVESERA
TAMAÑO BASE DE DATOS: 103 NOTAS ANALIZADAS	96,97%	63,16%	62,90%	93,14%
Nº DE CAPAS OCULTAS: 2				
Nº NEURONAS/CAPA: 150				
ENTRENAMIENTO: 'TRAINRP'				
F(x): 'logsig'				
EPOCH: 1025				

Tabla 5. 6: Resultados de la prueba 1

Para saber un poco más sobre por qué esta primera red no distingue bien entre estos dos instrumentos se creó un pequeño código en Matlab ('clasificacion.m') que permitía obtener la salida de nuestra red neuronal para una determinada nota. Esto nos permitió analizar las notas de viola y violín una a una, para ver qué notas no son bien reconocidas, y comparar los espectros de potencia, pues veremos que, para notas de distintos instrumentos con espectros muy parecidos, la red neuronal falla.

Las Figuras 5.13, 5.14 y 5.15 muestran como para una determinada nota de viola ('viola_test3.mat') la red neuronal nos devuelve como salida ('a.mat') un violonchelo.

```

1  %clear all;close all;clc;
2
3  load('red_150_trainrp.mat'); % Cargamos red neuronal entrenada
4  load('viola_test3.mat'); % Cargamos nota
5  a=sim(red,RR); % Salida de la red
6  instrumento=[];
7
8  %%
9  for i=1:4
10     salida_i=mean(a(i,:));
11     instrumento=[instrumento,salida_i];
12     instrumento=round(instrumento);
13     salida_i=0;
14 end
15
16 chelo=[0 0 0 0];
17 viola=[0 0 0 1];
18 violin=[0 0 1 0];
19 flauta=[0 0 1 1];

```

Command Window

```

>> clasificacion
suena un VIOLONCHELO
fx >>

```

Figura 5. 13: Ejecución 'clasificación.m'

4	5	6	7	8	9	10	11	12	13	14
59	59	59	59	59	59	59	59	59	59	
1	1	1	1	1	1	1	1	1	1	
0.3030	0.3091	0.2982	0.3121	0.2907	0.2799	0.2971	0.2601	0.2448	0.1854	
0.4147	0.4393	0.4045	0.4305	0.4202	0.3954	0.3853	0.3938	0.3972	0.5301	
0.0287	0.0224	0.0220	0.0222	0.0246	0.0233	0.0251	0.0227	0.0229	0.0219	
0.0983	0.1176	0.0948	0.1103	0.1020	0.0984	0.0985	0.0963	0.0846	0.0634	
0.0016	0.0022	0.0022	0.0026	0.0016	0.0014	0.0011	9.0197e-04	0.0011	0.0020	
0.0047	0.0050	0.0028	0.0023	0.0032	0.0040	0.0037	0.0057	0.0040	0.0012	6.26e-04
0.0047	0.0062	0.0042	0.0051	0.0047	0.0052	0.0051	0.0052	0.0071	0.0088	
8.8299e-04	8.2162e-04	0.0011	0.0012	0.0011	9.4064e-04	7.9167e-04	0.0011	0.0013	0.0018	
5.0602e-04	5.5775e-04	2.9104e-04	2.1171e-04	0.0014	0.0020	0.0024	0.0026	0.0022	0.0013	

Figura 5. 14: Entrada analizada (Si (3) Viola)

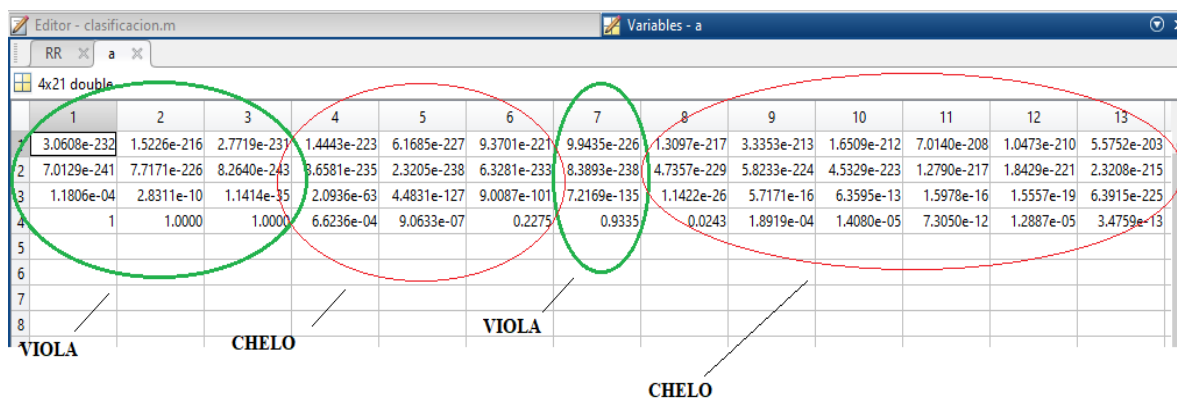


Figura 5. 15: Salida de la red neuronal para un Si (3) Viola

Como podemos ver en la Figura 5.15 son mayores las tramas identificadas como violonchelo que como viola, por lo que se ha calculado el espectro de la nota en concreto que está dando el problema para ambos instrumentos, y se han comparado. No son iguales, pero si presentan algunas similitudes, por las cuales tal vez, nuestra red no esté discriminando correctamente.

Como podemos ver en la Figura 5.16, ambos espectros presentan la mayor potencia en la frecuencia fundamental de la nota, el armónico 4 presenta potencias muy parecidas en ambos espectros y es a partir de este armónico donde la potencia empieza a decrecer.

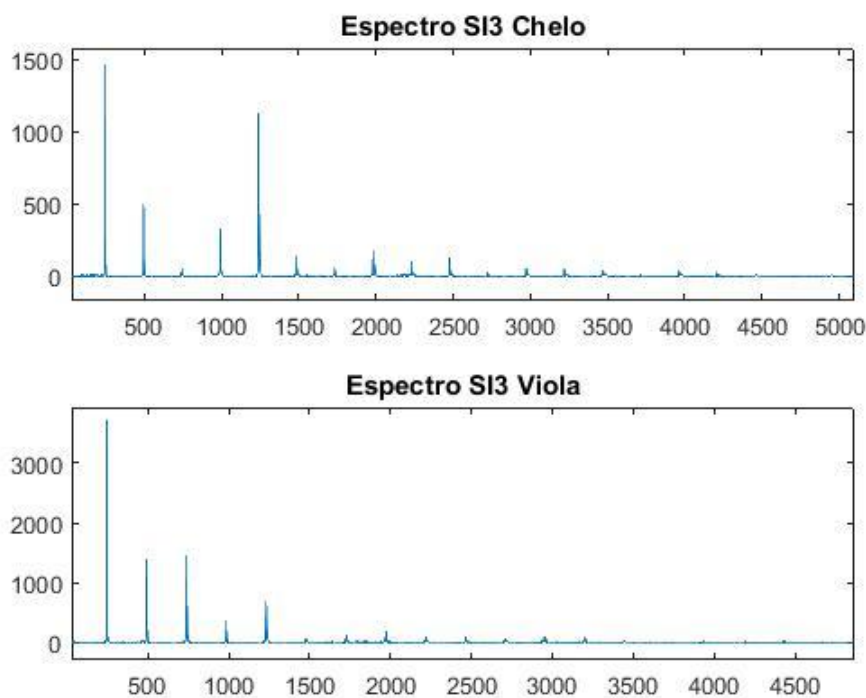


Figura 5. 16: Espectro Si (3) Chelo Vs. Espectro Si (3) Viola

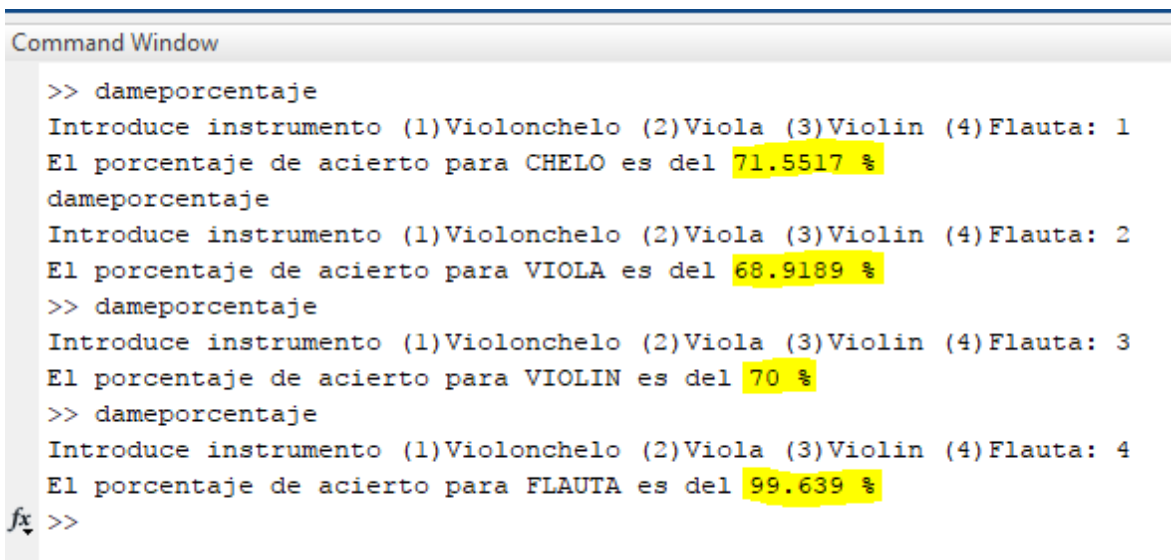
Prueba 2: Se aumentó el número de notas analizadas para viola y violín pues muchas fueron descartadas en el bloque de procesado por sonar desafinadas, teniendo un total de 108 notas. 32 de violonchelo, 25 de viola, 24 de violín y 27 de flauta travesera. Esta vez se entrenó la red con 86 notas, dejando el resto para test. El aumento de datos para el entrenamiento no fue excesivamente grande, por eso se optó por cambiar otros parámetros característicos de la red neuronal, como el método de entrenamiento. Al igual que la prueba anterior la red neuronal utilizada tenía 2 capas ocultas, también con 150 neuronas cada una, se optó por el descenso de gradiente ('traingd') como método de entrenamiento y como función de activación 'logsig'. Véase Tabla 5.7 y Figura 5.17 como ejemplo. La red ya entrenada junto con los datos reservados para TEST se incluye en la carpeta de ANEXOS.

```
clear all;close all;clc
[instrumentos,instrumentos_bin,target]=leer; % Leemos parámetros de
entrenamiento
red=newff(minmax(instrumentos),[150 150
4],{'logsig','logsig','logsig'},'traingd'); % Creamos red

red.trainParam.epochs=5000; % Máximo de iteraciones
red.trainParam.min_grad=1e-9; % Mínimo gradiente
view(red); % Visualizamos la red
[red tr]=train(red,instrumentos,target); % Entrenamos la red

save('150traingd1.mat','red'); % Salvamos la red creada
```

Tabla 5. 7: Creación red neuronal Prueba 2



```
Command Window

>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 1
El porcentaje de acierto para CHELO es del 71.5517 %
dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 2
El porcentaje de acierto para VIOLA es del 68.9189 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 3
El porcentaje de acierto para VIOLIN es del 70 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 4
El porcentaje de acierto para FLAUTA es del 99.639 %
fx >>
```

Figura 5. 17: Porcentaje de acierto en Prueba 2

Como vemos Figura 5.17 de resultados, el porcentaje de acierto para viola y violín mejora considerablemente respecto a los resultados de la prueba 1. Además, si ya eran buenos los resultados para la flauta travesera en la anterior prueba, ahora son mejores. No podemos decir lo mismo para el violonchelo, que baja su porcentaje de acierto aproximadamente un 25 %. En la Tabla 5.8 recogemos estos datos en función de las características de la red neuronal utilizada:

	VIOLONCHELO	VIOLA	VIOLÍN	FLAUTA TRAVESERA
TAMAÑO BASE DE DATOS: 108 NOTAS ANALIZADAS	71,55%	68,92%	70,00%	99,64%
Nº DE CAPAS OCULTAS: 2				
Nº NEURONAS/CAPA: 150				
ENTRENAMIENTO: 'TRAINGD'				
F(x): 'logsig'				
EPOCH: 5000				

Tabla 5. 8: Resultados prueba 2

Como podemos apreciar, el número de iteraciones que necesita la red para entrenar y dar posteriormente estos resultados es casi cinco veces mayor a las que necesitó la primera red utilizada. Esto tiene su explicación, y es debida al método de entrenamiento concreto.

El método de descenso de gradiente, aplicado al entrenamiento de redes neuronales se basa en el cálculo del gradiente de la función de error $f(x)$, para así obtener la dirección a la cual los valores de dicha función disminuyen considerablemente (véase [20]).

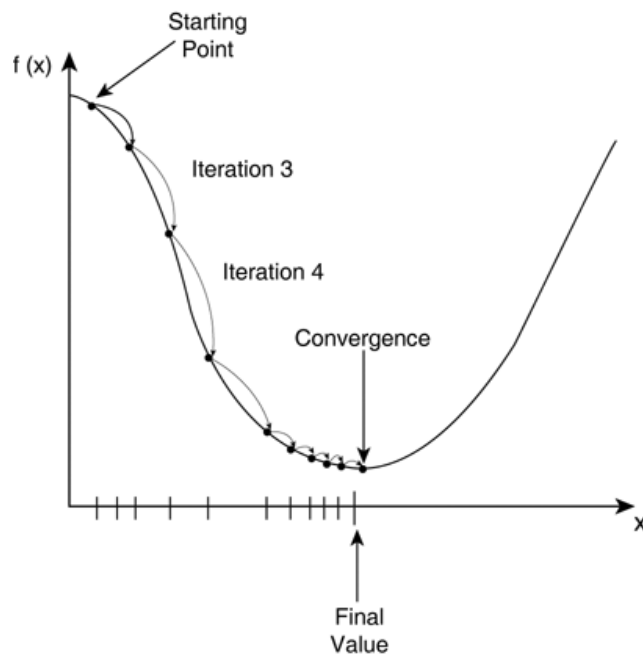


Figura 5. 18 [20]: Visión gráfica del método Descenso de Gradiente

Se dice que este método requiere más iteraciones para funciones de error complejas, donde el valle que forma es largo y estrecho [20]. Es lo que debe estar pasando en nuestro caso, donde el principal problema que encuentra nuestra red es saber diferenciar entre instrumentos de la misma familia (violonchelo, viola y violín). Es aquí donde la función de error, dada por los pesos sinápticos de nuestra red, debe estar adoptando una estructura de este tipo, y el proceso de entrenamiento se hace más lento.

5.2.2.1 Sobreaprendizaje (Overfitting)

Con el fin de obtener resultados semejantes a los de la prueba 2, pero consiguiéndolos con un menor número de iteraciones, probamos a aumentar el número de neuronas por capa y utilizar un método de entrenamiento más rápido como es el backpropagation ('trainrp'), pero los resultados fueron demasiado desfavorables.

Si desconocemos a priori los efectos indeseados que se dan al trabajar con redes neuronales, resultará extraño pensar que los resultados puedan empeorar cuando aumentemos el número de neuronas, aun así, eso pasa y fue el motivo por el que en segundas pruebas obtuviéramos tan pobres resultados.

Uno de los efectos que perjudican la decisión de una red neuronal es el overfitting o sobreaprendizaje, efecto por el que la red neuronal particulariza y no sabe generalizar [21], dando respuestas incorrectas. Nuestra red debe saber diferenciar entre notas musicales con

las cuales no ha sido entrenada pues, aunque no haya sabido de ellas previamente en el entrenamiento, si sabrá de otras procedentes del mismo instrumento las cuales tendrán ese ‘algo’ característico que sepa conocer cualquier sonido.

El aumento de neuronas es la principal causa de este efecto y se refleja muy bien en resultados obtenidos con redes que superaban las 150 neuronas por capa oculta. En las Figuras 5.19 y 5.20 se muestran los obtenidos para una red con 220 neuronas por capa oculta.

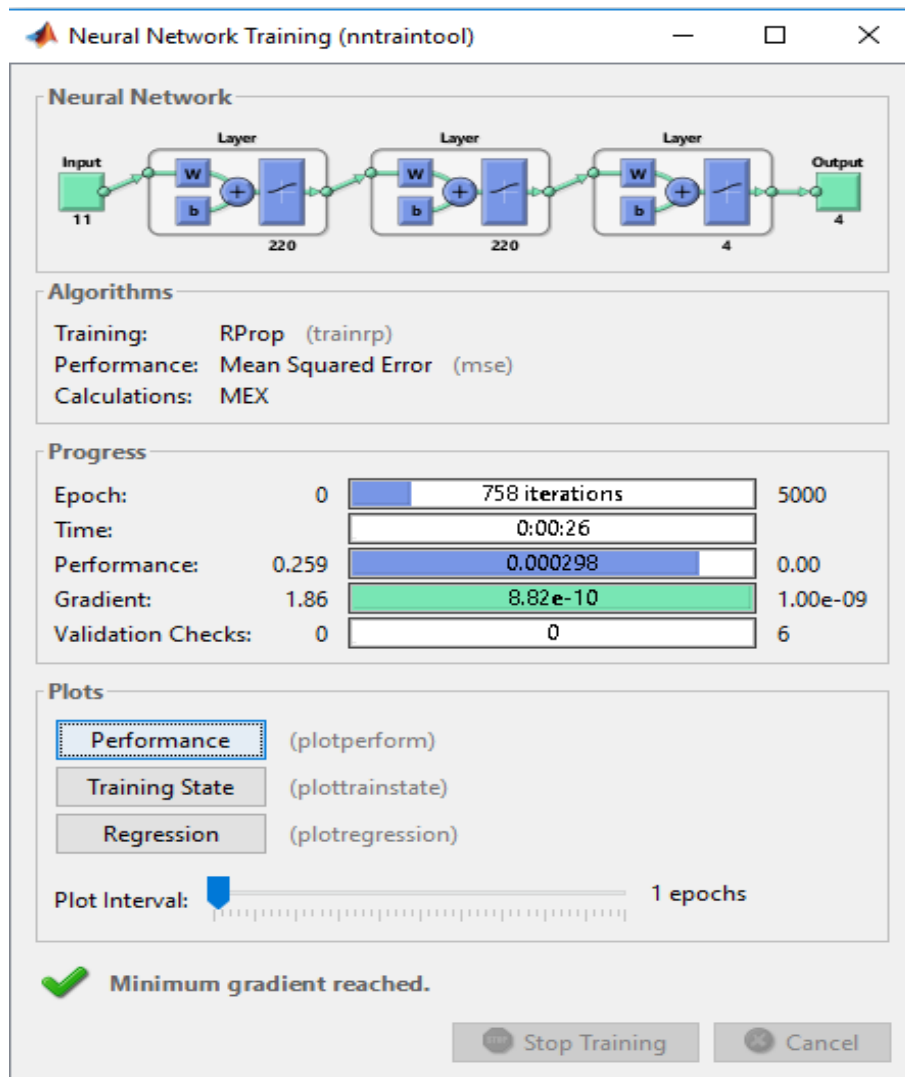


Figura 5. 19: Entrenamiento Prueba 3

```
Command Window
>> save('red_2_220_rp_logsig.mat','red');
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 1
El porcentaje de acierto para CHELO es del 42.2414 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 2
El porcentaje de acierto para VIOLA es del 78.3784 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 3
El porcentaje de acierto para VIOLIN es del 42.5 %
>> dameporcentaje
Introduce instrumento (1)Violonchelo (2)Viola (3)Violin (4)Flauta: 4
El porcentaje de acierto para FLAUTA es del 44.0433 %
```

Figura 5. 20: Porcentajes de acierto en prueba 3

CAPÍTULO 6. CONCLUSIONES Y LÍNEAS DE FUTURO

6.1 Conclusiones

Tras completar los objetivos planteados en un principio para este proyecto, es momento de mencionar las conclusiones a las que hemos llegado:

- La clasificación automática de sonidos es hoy en día objetivo de multitud de Apps, siendo las de temática musical las más demandas, como son los sistemas de transcripción musical y afinadores de instrumentos. Además, ya hay y con los cuales se investiga, robots capaces de reconocer voz.
- Es posible crear estos sistemas, combinando técnicas de procesamiento de audio con herramientas de aprendizaje automático, tales como las Redes Neuronales Artificiales.
- El bloque de procesamiento o bloque de extracción característica es clave. Un mal paso aquí empeoraría el acierto de la red neuronal.
- El bloque de clasificación también muestra sus dificultades e incógnitas, que se van resolviendo a base de continuas pruebas: La cantidad de datos para el entrenamiento es clave, a mayor número de datos, la red responde de manera más acertada. Un mayor número de neuronas no implica mejores resultados, sino todo lo contrario debido al *Overfitting*. Además, no existe un método de entrenamiento óptimo, todo dependerá del caso concreto en el que estemos trabajando.

- En nuestro caso, se ha demostrado que la mayor dificultad que encuentra la red neuronal es la diferenciación entre instrumentos de la misma familia (violonchelo, viola y violín), que se solventa mejor con el método de descenso de gradiente.
- La flauta travesera es bien reconocida para cualquier método de entrenamiento al ser el único instrumento de familia diferente.

6.2 Líneas de futuro

El ámbito para el que hemos trabajado en este proyecto es muy extenso y con multitud de aplicaciones. En este apartado se mencionan algunas de las mejoras y nuevas funcionalidades que podrían añadirse al proyecto para trabajos futuros:

- En primer lugar, trabajar con muchas más muestras instrumentales: Familias completas de instrumentos. Esto daría lugar a una investigación mucho más completa sobre la eficiencia de las Redes Neuronales como clasificadores de audio.
- Probar con algoritmos basados en técnicas diferentes a las Redes Neuronales, y probar de igual manera la eficiencia a la hora de discriminar entre instrumentos.
- Procesado de audio más eficiente: Ventanas de análisis con longitud variable en función de la nota analizada a la hora de aplicar el método de la autocorrelación para el cálculo de la frecuencia fundamental.
- Realizar una aplicación móvil o web.
- Incorporar el sistema a un determinado hardware: ¿Qué ocurriría si las mesas de mezclas digitales actuales llevaran incorporado un sistema como este? Tal vez podríamos llevar a cabo la ecualización del instrumento según el tipo.

CAPÍTULO 7. BIBLIOGRAFÍA

- [1] S.B. KOTSIANTIS, I.D. ZAHARAKIS, P.E. PINTELAS. Artificial Intelligence Review. *Machine Learning: A review of classification and combining techniques*, 2006, nº 3, vol 26, 159-190.
- [2] SHAZAM. *Company*. Disponible en: <https://www.shazam.com/es/company>
- [3] R. LYON. IEEE Signal Processing Magazine. *Machine hearing: An emerging field*Signal, 2010, nº 5, vol. 27, 131-139.

- [4] SOFT BANK ROBOTICS. *Robots: Nao*. Disponible en: <https://www.ald.softbankrobotics.com/en/robots/nao>
- [5] SONY. *Aibo*. Disponible en: <http://www.sony-aibo.com/>
- [6] PÉREZ DE PRADO, ROCÍO. Contenido de la asignatura Fundamentos de Ingeniería Acústica. *Introducción a la Ingeniería acústica: Bloque I*.
- [7] PÉREZ DE PRADO, ROCÍO. Contenido de la asignatura Fundamentos de Ingeniería Acústica. *Psicoacústica: Bloque II*.
- [8] PARRA RODRÍGUEZ, FERNANDO. Teoría de la música. *El origen de la escala natural II*, 2013. Disponible en: <http://www.artificialgenetics.es/index.php/teoria-de-la-musica/25-el-origen-de-la-escala-musical-ii>
- [9] PARRA RODRÍGUEZ, FERNANDO. Síntesis musical. *Análisis de la nota Do (4) en un oboe*, 2013. Disponible en: <http://www.artificialgenetics.es/index.php/sintesis-musical/35-nota-do4-en-un-oboe>
- [10] MARTÍNEZ MUÑOZ, DAMIÁN et al, *Transmisión Digital*. Jaén: Servicios de publicaciones, 2009.
- [11] PARRA RODRÍGUEZ, FERNANDO. Procesado de la señal musical. *Efecto del número de muestras en la transformada discreta de Fourier*, 2015. Disponible en: <http://www.artificialgenetics.es/index.php/procesado-de-la-senal-musical/20-efecto-del-numero-de-muestras-en-la-transformada-discreta-de-fourier>
- [12] CAÑADAS, FRANCISCO JESÚS. Contenido de la asignatura Técnicas de codificación y transmisión. *Aplicaciones de la transformada discreta de Fourier (DFT): Bloque III*.
- [13] HILERA GONZÁLEZ, JOSÉ RAMÓN. *Redes neuronales artificiales: fundamentos, modelos y aplicaciones*. Madrid: Ra-ma, D.L. 1995.
- [14] PARRA RODRÍGUEZ, FERNANDO. Procesado de la señal musical. *Análisis espectral para el violín*, 2015. Disponible en: <http://www.artificialgenetics.es/index.php/procesado-de-la-senal-musical/18-analisis-espectral-para-el-violin>
- [15] PARRA RODRÍGUEZ, FERNANDO. Procesado de la señal musical. *Análisis espectral para el violonchelo*, 2015. Disponible en: <http://www.artificialgenetics.es/index.php/procesado-de-la-senal-musical/17-analisis-espectral-para-el-violonchelo>
- [16] IAN MACLOUGHLIN. *Applied Speech and audio processing: with Matlab examples*. Cambridge: Cambridge University, 2009.

- [17] MÁRQUEZ, JORGE. Procesamiento y análisis de señales e imágenes. *Mediana de un conjunto de muestras y filtro de mediana*. Disponible en: http://www.academicos.ccadet.unam.mx/jorge.marquez/cursos/imagenes_neurobio med/Mediana_filtro.pdf
- [18] M.A. ROSALES-REYNOSO, C.I. JUÁREZ-VÁZQUEZ, P. BARROS-NÚÑEZ. Neurología. *Evolución y genómica del cerebro humano*, 2015, Elsevier, in press.
- [19] MARK HUDSON BEALE, MARTIN T. HAGAN, HOWARD B. DEMUTH. *Neural Network Toolbox – MATLAB 2016b*. Natick: The MathWorks, 2016.
- [20] CAPARRINI SANCHO, FERNANDO. Entrenamiento de Redes Neuronales. *Mejorando el gradiente descendiente*, 2017. Disponible en: <http://www.cs.us.es/~fsancho/?e=165>
- [21] ESCOM IPN. *Efectos no deseados en el entrenamiento de Redes Neuronales*. México: Escom IPN, 2012. Disponible en: <https://es.slideshare.net/mentelibre/redes-neuronales-entrenamiento-efectos-no-deseados>
- [22] MONTERO GABARRÓ, JAVIER. Polifacetismo y creatividad. *Cálculo de la frecuencia de nuestras notas musicales*, 2012. Disponible en: <http://elclubdelautodidacta.es/wp/2012/08/calculo-de-la-frecuencia-de-nuestras-notas-musicales/>
- [23] SEGURA SORGORB, MIGUEL. Trabajo fin de grado. *Estimación automática de acordes para uso en transcripción musical a partir de audio*, 2015, Universidad de Alicante.

CAPÍTULO 8. ANEXOS

En este apartado se explica el contenido de la carpeta comprimida “ANEXOS” que se adjuntará junto con la memoria en la entrega de esta.

En primer lugar, esta carpeta incluye los códigos de las funciones y script en MATLAB que han permitido el estudio y desarrollo del proyecto que hemos implementado:

Funciones y script en MATLAB (archivos .m):

- Función “auto.m” implementa el método de la autocorrelación para el cálculo de la frecuencia fundamental de las notas musicales.

- Función “filtro_mediana.m” realiza un filtrado paso bajo para el cálculo de un valor medio de frecuencia a partir del cual se calcula la frecuencia MIDI teórica de la señal en concreto.
- Función “armónicos.m” realiza la búsqueda de la amplitud de los 10 primeros armónicos de la señal, llevando a cabo el cálculo de la densidad espectral de potencia.
- Script “principal.m” ejecuta las anteriores funciones para una determinada nota musical. Almacena en una matriz las variaciones de amplitud de los 10 primeros armónicos en el tiempo, junto con la frecuencia teórica MIDI calculada tras el filtrado.
- Función “leer.m” concatena en una única matriz los datos analizados para cada instrumento, que será la matriz de entrada para el entrenamiento de la red neuronal. A cada entrada le asigna la salida deseada o target.
- Función “RedNeuronal.m” lleva a cabo el entrenamiento supervisado de la red neuronal.
- Función “dameporcentaje.m” devuelve el porcentaje de acierto por parte de la red para cada instrumento.
- Script “clasificación.m” nos permitió a modo de ayuda localizar aquellas notas que no eran bien distinguidas.

Muestras instrumentales (archivos . wav):

Por otro lado, se han añadido a esta carpeta algunas de las notas musicales (archivos .wav) que en el bloque de procesamiento fueron analizadas para la extracción de características.

Red Neuronal ‘150_traingd1.mat’ y datos para TEST (archivos .mat):

Finalmente se añadió también la Red Neuronal entrenada con el método de descenso de gradiente (Prueba 2) con la que obtuvimos mejores resultados (‘150_traingd.mat’). Se incluyen también los datos reservados para TEST que se utilizaron para probar dicha Red Neuronal.