Reconocimiento de entornos acústicos para dispositivos móviles

Universidad Politécnica de Valencia

Autor: Adriana Hernández Pérez (adherpe@gmail.com)

Director: José Javier López Monfort

Abstract

Muchos dispositivos móviles pueden actuar de manera sensitiva respecto al entorno en el que se encuentren, pero aun así depende del humano, que utilice dicho dispositivo, la adaptación. Sería preferible que fueran capaces de adaptarse al entorno en el que se encuentren sin intervención humana. En el proceso independiente de adaptación del dispositivo, la definición de las características adecuadas para los sonidos ambientales es el problema más importante en los sistemas de reconocimiento ambiental. Así como en problemas de reconocimiento de patrones de voz, extraer el conjunto de características que definen el entorno es la llave para obtener una precisión efectiva.

En trabajos anteriores, se ha mostrado la precisión de muchas técnicas de extracción de características, adecuadas para señales de voz y música, aplicadas al análisis de sonidos ambiente, asumiendo el buen funcionamiento con sonidos no estructurados. Así mismo, hay estudios en los que se ha propuesto la utilización de otro tipo de métodos de extracción de características para sonidos no estructurados como son los sonidos ambiente o del entorno.

En el presente proyecto, tras revisar el trabajo relacionado en el área de reconocimiento de entornos acústicos, se presentan los resultados obtenidos de intentar clasificar las características obtenidas mediante las técnicas anteriormente propuestas, haciendo uso de redes neuronales, motivado por la observación de que los sistemas de codificación biológica parecen estar altamente relacionados por el entorno en el que nos encontremos. Además, se ha intentado proponer nuevas técnicas de extracción de características en el dominio frecuencial de la señal, movido por la observación de que las señales pertenecientes a distintos entornos tienen un comportamiento diferente entre sí en energía, con puntos comunes entre las señales pertenecientes a los mismos entornos.

Índice general

Glosario

MFCC : Mel-Frecuency Cepstral Coefficients

MP : Matching Pursuit

ANN/RNA : Artificial Neural Networks/Redes Neuronales Artificiales

GMM : Gaussian Mixture Model

PCA : Principal components analysis

KNN : K-Nearest Neighbors

ΔMFCC : Delta Mel-Frecuency Cepstral Coefficients

LPC : Linear Predictive Coding

ΔLPC : Delta Linear Predictive Coding

LPCC : Linear Prediction Cepstral Coefficients

HCC : Homomorfic Cepstral Coefficients

STFT : Short-Time Fourier Transform

FWT : Fast (discrete) Wavelet Transform

CWT : Continuous Wavelet Transform

DTW : Dynamic Time Warping

LVQ : Learning Vector Quantization

FT : Extracción Frecuencial

LTS : Long-term Statistics

DFT : Transformada Discreta de Fourier

FFT : Transformada Rápida de Fourier

DCT : Transformada Discreta Coseno

MSE : Minimal Square Error

Capítulo 1
Introducción

El reconocimiento del entorno a partir de sonidos es un problema básico en el procesamiento de la señal. Tiene importantes aplicaciones en dispositivos móviles, sistemas de navegación e incluso, como se demuestra en [5], pueden ser interesantes en aplicaciones de agenda personal.

El sonido del entorno define la localización del dispositivo en entornos con ciertas características acústicas en común tal cómo una calle, un restaurante o una oficina. Una señal de audio contiene una cantidad de información significativa, que permite definir el entorno, dando una mayor compresión del mismo que la que obtenemos de la información visual. Un ser humano es capaz de reconocer el entorno simplemente mediante el sonido percibido aunque el individuo se encuentre en un lugar completamente nuevo.

Se podría estudiar la manera perfecta de capturar la descripción completa de una escena mediante la fusión del audio e información visual, lo cual puede ser ventajoso como método de frontera en el reconocimiento de entornos acústicos. Pero no es el objeto de este proyecto.

En este proyecto, para conseguir la una clasificación óptima del entorno, se centra en la señal de audio. Se pretende ahondar un poco más en el estudio de sistemas capaces de analizar el entorno en el que se encuentran y adaptarse a los mismos con el fin de disminuir en la medida de lo posible la interacción con el usuario. Para ello, se experimenta en diversas direcciones tratando de extraer el conjunto de características que definen el entorno de manera óptima a partir de muestras de audio.

Una señal de audio puede obtenerse en cualquier momento durante el funcionamiento de cualquier dispositivo independientemente de las condiciones externas tales como falta de luz o visibilidad. Es una señal de un ancho de banda pequeño relativamente y por tanto tiene unos requerimientos de procesamiento y almacenamiento pequeños y por otro lado se pueden encontrar dispositivos de grabación de audio baratos, robustos y de buena calidad. Lo más importante es que la señal de audio no se ve prácticamente afectada por cambios espontáneos de aspecto, posición, iluminación, etc. Además el entorno siempre deja evidencias acústicas identificables, tanto es así

que en algunas ocasiones una persona hablando a través de un dispositivo móvil es capaz de identificar dónde se encuentra su interlocutor.

Se han usado muchos tipos de características para definir las señales de audio. La mayoría de los estudios existentes se basan en el uso de MFCC. La extracción de los MFCC está basada en el funcionamiento del sistema auditivo humano y se ha demostrado su buen funcionamiento en el reconocimiento de sistemas estructurados tales como voz o música, pero la precisión de los sistemas basados en la extracción de éste tipo de características se degrada en presencia de ruido. [1][2]

En el caso de sonidos ambiente, se trata con sonidos no estructurados, que típicamente son clasificables como ruido ambiente, que contienen una gran variedad de diferentes sonidos y que en ocasiones pueden venir caracterizados por picos espectrales estrechos, tales como chirpings de insectos. Basándose en este concepto, se han llevado a cabo estudios de reconocimiento de entornos acústicos haciendo uso de la técnica de MP. Las características extraídas mediante esta técnica, son más robustas respecto al ruido de fondo. Se ha demostrado, como se verá en la siguiente sección, que este tipo de características pueden usarse para apoyar otro tipo de características de audio para obtener precisiones relativamente altas en el reconocimiento del entorno.

En el presente trabajo, se estudian todas las soluciones propuestas en trabajos anteriores con la diferencia de que se pretende clasificar las características extraídas de la señal por las técnicas propuestas mediante el uso de Redes Neuronales Artificiales. El uso de ANN viene motivado por la observación de que los sistemas perceptuales biológicos muestran estructuras de filtros similares a aquellas derivadas de procedimientos de teoría de la información para la codificación óptima, que buscan maximizar la independencia estadística de los componentes del filtro. Esto quiere decir, que un codificador expuesto únicamente a sonidos pertenecientes a un entorno concreto, mostrará estructuras características a dicho entorno. [1][15]

En el presente texto se investigan diferentes técnicas de extracción de características en ocho entornos diferentes que se clasifican mediante ANN. Se demuestra así mismo que las técnicas de extracción de características usadas normalmente no siempre funcionan bien para sonidos no estructurados así como tampoco se puede confiar plenamente en el buen funcionamiento de técnicas propuestas para la extracción de características en señales de sonido ambiente. Ahondando un poco más en el problema, se propone una técnica de extracción de características centrada en el espectro de la señal y las características espectrales de la señal por bandas de frecuencia. Se muestra finalmente que tampoco este tipo de extracción presenta una precisión por encima de las demás técnicas propuestas, aunque sí contribuye a la mejora de la precisión en combinación con otras técnicas vistas.

1.1. Estructura del proyecto

En lo sucesivo se trata de manera más profunda el problema que nos ocupa y se introduce el contexto teórico. En el capítulo 2, Antecedentes, se trata con detalle el trabajo previo relacionado marcando un punto de partida en la investigación. Se intenta explicar el punto actual en el que se encuentra el estudio del reconocimiento de entornos por medio de sonidos ambientales al mismo tiempo que se han sentado las bases del presente proyecto.

El capítulo 3, Análisis de las señales, se mete de lleno en el análisis teórico del problema. En él, el lector puede encontrar el desarrollo mediante métodos numéricos de los dos algoritmos bajo prueba de extracción de características de la señal, así como el funcionamiento de las RNA.

En el capítulo 4, Experimentos, se describe la preparación de los tres experimentos llevados a cabo en el proyecto de manera más práctica, es decir, sin entrar en tantos detalles teóricos, los pasos seguidos para la preparación de las señales de audio, extracción de las características de las señales y la creación, entrenamiento y validación de dichas características con las tres RNA correspondientes a cada tipo de características.

Los resultados de los experimentos descritos en el capítulo anterior se tratan en el capítulo 5, Resultados, en el que además se compara la precisión de la clasificación mediante RNA con la del cerebro humano mediante un experimento similar llevado a cabo con un grupo de personas.

Por último, encontramos un resumen de las conclusiones en el capítulo 6, Conclusiones. En este capítulo se hace una pequeña descripción y evaluación personal del desarrollo del proyecto, problemas encontrados, errores cometidos y posibles líneas de investigación futuras.

Capítulo 2
Antecedentes

Una posible manera de mostrar los antecedentes es explicar un poco el "estado del arte". Como ya se ha mencionado en secciones anteriores, el principal problema que se presenta en el reconocimiento de entornos acústicos es la extracción de las características adecuadas que lo definen de la señal de audio. En estudios previos se han propuesto métodos innovadores que tratan de encontrar la mejor solución a dicho problema, con la conclusión de que lo que puede valer para el reconocimiento del habla no siempre funciona bien para sonidos desestructurados así como las técnicas propuestas para sonidos no estructurados tampoco han resultado la solución perfecta al problema.

A continuación se explica un poco más en detalle estudios previos en los que se apoya para realizar el presente proyecto, sus resultados y conclusiones.

2.1. Classifying user environment for mobile applications using linear autoencoding of ambient audio[1]

Cuyo objetivo es la clasificación del entorno en los teléfonos móviles. Se pretende que el teléfono móvil sea capaz de cambiar entre modos dependiendo del entorno en el que se encuentre el usuario. Presenta un nuevo método para clasificar los entornos basándose en señales acústicas. Hace uso del método de auto codificación de las redes neuronales (ANN) y del modelo de mezcla gaussiano (GMM), motivado por la observación de que los sistemas de codificación biológicos están muy influenciados por las estadísticas del entorno. En los experimentos llevados a cabo se demuestra que el método de auto codificación supera al GMM estándar en un conjunto representativo de muestras y que la combinación lineal de ambos (método de codificación y GMM) produce mejores resultados que cada uno por separado.

Los estudios previos que se mencionan son dos principalmente: **Clarkson & Pentland** que estudia el conocimiento del entorno usando señales de audio, video y otros flujos sensoriales en el contexto de un sistema diseñado para encontrar patrones de la vida cotidiana del usuario desde datos obtenidos por medio de sensores a partir de señales y **Ellis & Lee** que presentan un sistema de archivos personal que usa una técnica de agrupamiento espectral sin supervisión para analizar el audio del entorno.

Como base teórica cabe destacar el empleo de señales de audio para la clasificación del entorno por ser señales de bajo ancho de banda con requisitos de procesado y almacenamiento pequeños y además los sensores de audio de buena calidad son relativamente robustos y baratos. Más importante, la señal de audio no se ve afectada por cambios potencialmente aleatorios en aspecto, iluminación, posición, etc. y por otro lado, el entorno siempre deja evidencias claras en estas señales. Para la extracción de características hacen uso de los MFCC's, que explicaré con detalle en secciones posteriores aplicando PCA para la reducción de la dimensionalidad de las características extraídas. Para la clasificación de los resultados hacen una comparación de dos tipos, las redes neuronales feed forward que pueden tener más de una capa oculta. Intentan reconstruir la capa de entrada en la capa de salida por tanto la salida es del mismo tamaño que la entrada. Son redes neuronales supervisadas que se forman usando un método de gradiente descendente, tal como la propagación hacia atrás. La capa escondida, es de menos tamaño que la entrada, por tanto, la dimensionalidad de los datos de entrada se ve reducida a un espacio de menos dimensión en esta capa. La salida de la capa escondida se reconstruye luego en la capa de salida. Si se usa un n° mayor de capas escondidas, se puede reducir a un espacio de menores dimensiones datos que a la entrada presentan gran dimensionalidad. El problema es que entrenar una red con muchas capas escondidas resulta tedioso y puede no dar buenos resultados por la propagación de un error a lo largo de las capas. Este método, puede resultar en un mapeo no lineal de la capa de entrada a la de salida. El método de auto codificación no es capaz de reducir la dimensionalidad de los datos en el mismo grado que PCA. La otra técnica de clasificación que se compara es GMM**,** modelo bien maduro para realizar "clustering" que también se usa para la estimación de densidad.

El experimento se lleva a cabo en 11 entornos diferentes, outdoor e indoor, con diferentes escenarios para cada ambiente, i.e: diferentes grabaciones para cada entorno (apartamento, pasillo de oficina, ascensor, clase, oficina, exterior, exterior con lluvia, restaurante, teatro y vehículo). Grabadora digital pequeña y micrófono estéreo Sony ECM-717. Grabaciones resampleadas a 16KHz y 16bit mono. Para asegurar el dominio del entorno se calcula la potencia media de cada grabación y se seleccionan aquellos segmentos que son más silenciosos que la media. Un 80%de los datos recogidos se usan como datos de entrenamiento y el 20% restante se usan como test. Se extraen 64 MFCC´s + Centroide espectral a una tasa de 100

frames por Segundo. Se normalizan los MFCC´s y los centroides (media nula y varianza unidad) a lo largo de la secuencia de entrenamiento y se combinan en un vector. Se comprime con PCA este vector a 35 dimensiones conservando el 75% de la varianza. Las características de PCA se escalan cada una a varianza unidad produciendo un conjunto de datos esféricos. Se usan Siete configuraciones de entrenamiento diferentes para testear el efecto de variar el n° de parámetros (2, 4, 8, 12, 16, 20, 24 unidades escondidas o gaussianas). Cada autoencoder se inicializa con pesos entre -0,05 y 0,05 durante 100 iteraciones usando un algoritmo de propagación hacia atrás con tasa de aprendizaje adaptativa inicializada a 0,05 y un impulse de 0,045. Cada GMM se entrena con las mismas características durante 20 iteraciones usando el algoritmo neural gas con una temperatura de inicio de 0,5 y tasa de enfriamiento de 0,01.

Para cada segmento de prueba se calculan os resultados con GMM y ANN. Según la hipótesis, el entorno será aquel en el que el modelo produzca mejores resultados. En general el modelo de ANN supera a GMM con el mismo n° de parámetros para un modelo con menos de 20 componentes. Después de hacer el experimento, se dan cuenta de que en algunos escenarios tales como el restaurante o el teatro, se producen algunas incongruencias y deciden realizar de nuevo el experimento pero combinando ambos métodos en uno híbrido. Éste, produce unos resultados mejores que los de cada modelo por separado aunque no pueden compararlo con un modelo humano real ya que no se ha realizado estas pruebas, pero comparándolo con algunas parecidas que sí se han realizado se puede entrever que el modelo híbrido produce buenos resultados, el error para 16 unidades/gaussianas es de un 19,95% por ejemplo, lo cual es un 2,32% mejor que el mejor modelo ANN y un 2,64% mejor que el mejor de los GMM. Se demuestra que las ANN superan a GMM en la clasificación de entornos, validando así la hipótesis inicial de que los procedimientos basados en una codificación óptima son eficientes para esta tarea. Una combinación lineal de ambos modelos resulta en una reducción considerable de las tasas de error. En trabajos futuros, intentarán mejorar la precisión del sistema, cobertura y utilidad.

Class	ANN		GMM	
	Precisión	Recall	Precisión	Recall
Apartament	87,7%	89,44%	81%	90%
Corridor	74,19%	82,14%	92,59%	89,28%
Elevator	68,90%	95,34%	65,34%	96,51%
Lecture	80,88%	77,45%	77,88%	78,38%
Meeting Room	86,40%	80,90%	96,56%	75,45%
Office	93,86%	93,29%	96,27%	94,51%
Outdoor	78,01%	55,59%	86,14%	53,35%
Raining	69,69%	87,34%	75,27%	86,70%
Restaurant	43,47%	64,51%	32,58%	93,54%
Theater	71,99%	90%	93,40%	84,99%
Vehicle	76,05%	68,06%	75,37%	63,02%

Tabla 1. Precisión por clase en [1]

2.2.Environmental sound recognition using MP-Based features[2][3]

Se marca como objetivo el reconocimiento del entorno acústico centrándose particularmente en la extracción de características usando la técnica de Matching Pursuit, forma de extraer características que describan el sonido donde otras técnicas fallan. Es más robusta respecto al ruido de fondo. Provee de una representación aproximada y reduce la energía residual con los mínimos átomos posibles. Se estudia una variedad de características de audio y se provee de una evaluación empírica en 14 tipos diferentes de entornos acústicos. Se intenta probar que MFCC, que es la técnica de extracción de características más comúnmente usada, no funciona siempre bien ya que es una técnica usada para extracción de características del habla, y puede ser complementada mediante MP para obtener una mayor precisión. El objetivo es obtener el mínimo número de bases necesarias para representar una señal. Los métodos propuestos son por ejemplo el método de las ventanas, basis pursuit, etc. En este caso se hace uso de un diccionario que descompone la señal seleccionando en el mismo el mejor conjunto de bases. MP usa un diccionario y provee de una forma de seleccionar un n° pequeño de bases que producen características significativas además de una representación flexible. Se basa en la elección del diccionario (conjunto de bases para obtener la combinación lineal que produce una representación aproximada de la señal). Los diccionarios más frecuentes son, en frecuencia el diccionario de Fourier (para datos de alta frecuencia), en tiempo el diccionario de Haar (para señales más estables y de baja frecuencia) y el diccionario que finalmente usan de tiempo-frecuencia es el de Gabor (diccionario más completo ya que describe el comportamiento de la señal en tiempo y frecuencia por tanto presenta las ventajas de los dos diccionarios anteriores ya que caracteriza la señal tanto en tiempo como en frecuencia, permitiendo una representación general mayor. Gabor resulta en el mínimo error de reconstrucción de todos los diccionarios probados).

Las características se buscan robustas, estables, físicamente interpretables, etc. MP consigue estos requerimientos ya que es potencialmente invariante al ruido de fondo y por tanto puede capturar características donde otros métodos fallan. El proceso de extracción es el siguiente: Para cada ventana de muestreo se descompone cada segmento usando MP. Se para tras obtener n átomos. Al descomponer después cada átomo con sus parámetros originales, se obtiene la frecuencia, escala y posiciones de traslación de cada átomo. Se acumulan todos los átomos de cada ventana de muestreo y se hace la media y la varianza correspondiente a cada parámetro por separado. Se prueba para ver que n° de átomos presenta mejor precisión, obteniendo que para n=5 se consigue una mayor precisión de reconstrucción. La idea es que se eligen los picos más prominentes en la ventana, que son las características de Gabor con una escala de coordenadas fija y sin información adicional. La información más importante que describe la señal se puede

encontrar en las bases con la mayor energía. El proceso por el que MP selecciona estas bases es exactamente en el orden en el cual elimina la mayor energía residual. Esto quiere decir que incluso los primeros átomos que encuentra el método MP contienen gran cantidad de información convirtiéndolos en características significativas.

Para el experimento se usan dos métodos de clasificación, KNN y GMM para evaluar el funcionamiento de estas características extraídas con los métodos MP, MFCC y MP+MFCC para 14 escenarios diferentes. Las muestras se toman de la manera más limpia posible para que el ruido de fondo no influya de manera negativa y la señal del ambiente a reconocer esté lo más limpia posible. Clips de 1 a 3 minutos, pre procesados y divididos en segmentos de 4 segundos y re muestreados a 22KHz. Canal mono a 16 bit por muestra. Las características se calculan de una ventana rectangular de 256 ptos. (11,6 mseg) con un 50% de superposición. Cada segmento de 4 segundos constituye una muestra para entrenamiento o prueba. Se escogen 100 segmentos de 4 segundos para definir cada clase. Todos los datos se normalizan con media nula y varianza unidad. Para KNN se usa una distancia euclídea como la distancia de medida entre vecinos para obtener los resultados. Para GMM, se calibra el número de mezclas por clase para producir unos mejores resultados. Se examinan las características de MP y otros como MFCC, ΔMFCC, LPC, ΔLPC, LPCC), ... Catorce entornos: Interior de restaurante, jardín de juegos, calle con tráfico y gente, tren pasando, interior de vehículos en marcha, interior de casinos, calle con sirena de policía, calle con sirena de ambulancia, naturaleza de día, naturaleza de noche, olas del océano, agua corriente/rio, ducha/lluvia y relámpagos. Se usan fuentes diferentes para entrenamiento o prueba. Por la limitación de los datos se requiere que cada emplazamiento contenga al menos cuatro fuentes de grabaciones diferentes, usando 3 para entrenamiento y 1 para prueba. Se hacen cuatro validaciones cruzadas para el experimento.

Los resultados presentados son que MP produce un 72,5% de media de aciertos, poco más que MFCC con un 70,9%. Al realizar el mismo experimento usando una combinación de todas las características juntas resulta en una precisión del 55,2%, más pobre que 12MFCC sólo. Al combinar MP y MFCC concatenando los dos vectores de características se consigue una precisión del 83,9% en la discriminación de catorce escenarios.

Clases	MP	MFCC	MP+MFCC
Naturaleza (día)	48%	45%	90%
Vehículo (mov)	82%	92%	100%
Restaurante	35%	47%	65%
Casino	60%	0	52%
Naturaleza (noche)	100%	0	100%
Calle w/policía	52%	75%	100%
Patio de colegio	60%	70	95%
Calle w/tráfico	50%	55%	70%
Lloviendo	52%	85%	90%
Agua corriente	45%	42%	50%
Tronando	35%	5%	42%
Tren pasando	65%	0	62%

Olas	90%	50%	100%
Calle w/ambulancia	58%	0	52%

Tabla 2. Resultado de la clasificación en [2]

2.3. Comparison of Techniques for environmental sound recognition[4]

Hace una comparativa de las principales técnicas de extracción de características así como de las principales técnicas de clasificación de características, para lo cual, evalúa dichas técnicas de extracción de características con cada una de las técnicas de clasificación.

Las técnicas de extracción de características que se evalúan las separan en dos grandes grupos, aquellas que realizan la extracción de características de forma estacionaria y aquellas que la realizan de forma no estacionaria, es decir, técnicas estacionarias de extracción de características y técnicas no estacionarias de extracción de características. Las técnicas estacionarias de extracción de características que se evalúan y el entorno en el que se usan comúnmente son, extracción frecuencial (música y habla), HCC (música y habla) y MFCC (música y habla). Por otro lado, las técnicas no estacionarias de extracción de características que se evalúan son, STFT, FWT y CWT.

Las técnicas de clasificación usadas en el estudio comparativo y que son las que se usan comúnmente para el reconocimiento de la voz/habla son, DTW, LVQ, ANN, Long-term statistics. Además de dichas técnicas se evalúa GMM, técnica usada comúnmente en el reconocimiento de instrumentos musicales. A diferencia que en el trabajo que nos ocupa, en el experimento llevado a cabo en éste que describo se intentan reconocer sonidos concretos tales como el sonido de unas llaves, pasos que se acercan o un cristal al romperse.

De los resultados, dado que no se evalúan entornos acústicos sino sonidos concretos, sólo se puede obtener una idea de la manera de llevar a cabo y evaluar los experimentos que se desarrollan a continuación. Además, dado que en el presente trabajo me centro en la clasificación de las características extraídas mediante ANN, se puede concluir que la parte más interesante es aquella en la que se comparan las diferentes técnicas mediante la clasificación con ANN.

Los resultados obtenidos en este estudio podrían resumirse en la siguiente tabla:

	DTW[15]	LVQ[16]	ANN	GMM	LTS[18]
FT[17]	66%	50%	0	21%	29%
MFCC	70%	37,5%	4%	46%	-
HCC[11]	29%	12,5%	0	12%	-
STFT[12]	58%	0	0	46%	-
FWT[13]	12%	12,5%	0	25%	-
CWT[14]	70%	54%	41%	21%	-

Tabla 3. Resultado de la comparación en [4]

Por tanto y como conclusión del presente capítulo, decir que los antecedentes arriba descritos son aquellos en los que se ha basado el trabajo desarrollado a continuación, en el que se ha hecho una nueva y más actual comparativa de las técnicas de extracción de características para sonidos ambiente no estructurados a las que se les ha aplicado siempre la técnica de clasificación mediante ANN. Se ha intentado además investigar el funcionamiento de dicha técnica de clasificación con otros métodos de extracción de características basados en la energía de la señal. Todo esto se desarrolla de manera más detallada en posteriores capítulos.

Capítulo 3
Análisis de las señales

El objetivo del presente proyecto es el reconocimiento de entornos a partir de señales de audio recogidas mediante la grabadora de los dispositivos móviles con el fin de conseguir una adaptación perfecta del dispositivo sin mediación del usuario.

Para conseguir dicho objetivo, se han estudiado diferentes métodos de extracción de características de la señal de audio y se ha aplicado una clasificación de dichas características mediante ANN.

Por tanto, se puede dividir éste análisis en dos grupos que se describen teóricamente en detalle a continuación, la extracción de características, dónde se proporciona una base matemática para cada técnica de extracción que se ha llevado a cabo y la clasificación mediante ANN, en el que se ahonda un poco en la descripción de las Redes Neuronales.

3.1. Extracción de características.

En este punto se analizan matemáticamente las técnicas de extracción de características, elegidas en base a los antecedentes y a las pruebas realizadas, que mejores resultaros han producido, trabajando con ANN en la fase de clasificación, en la caracterización de los ocho entornos a tener en cuenta. Dichas técnicas de extracción de características que se han aplicado explicadas de manera matemática son las siguientes:

- ***MFCC (Mel Frecuency Cepstral Coefficients):*** El análisis de la señal mediante los coeficientes cepstral se ha utilizado comúnmente para reconocimiento de voz con buenos resultados. Al aplicar esta técnica, se trasforma las muestras de la señal de voz a un conjunto de coeficientes que representan eficientemente las propiedades espectrales y concentraciones de Energía de la Señal de Voz, tratando de emular el tipo de procesamiento que realiza nuestro sistema auditivo. Al tener en cuenta las características del oído, se trata de asemejar el sistema

al reconocimiento hecho por una persona. Este análisis se basa en el uso de la escala de frecuencia Mel, la cual es un espaciamiento lineal de la frecuencia por debajo de los 1000Hz y un espaciamiento logaritmo por arriba de los 1000Hz. El esquema para la obtención de los coeficientes cepstrales es como sigue:

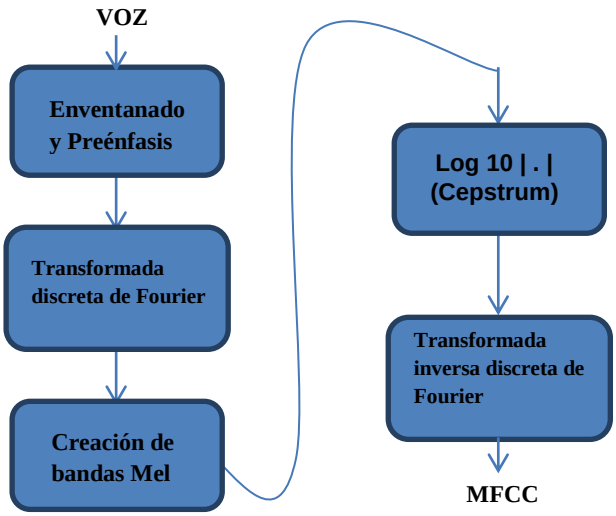


Fig. 1 Esquema de parametrización para la obtención de MFCC

Enventanado y Preénfasis: Produce el efecto de convolucionar el espectro de la señal muestreado con el espectro de la ventana, produciendo una distorsión de la Transformada de Fourier de la señal original. Por ello, conviene elegir un tipo de ventana que produzca la menos distorsión posible.

Si se define una ventana como $w(n),\ 0 \leq n \leq N-1$, donde N es el número de muestras de cada trama, entonces el resultado del enventanado de la trama $x_i(n)$ es:

$$y_i(n) = x_i(n)\, w(n),\ 0 \leq n \leq N-1$$

Ec.1 Enventanado de la trama $x_i(n)$

Típicamente se usa la ventana de Hamming la cual tiene la siguiente formulación y forma:

$$w(n) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi n}{N-1}\right), \mathbf{0} \leq n \leq N-1$$

Ec.2 Formulación de la ventana de Hamming

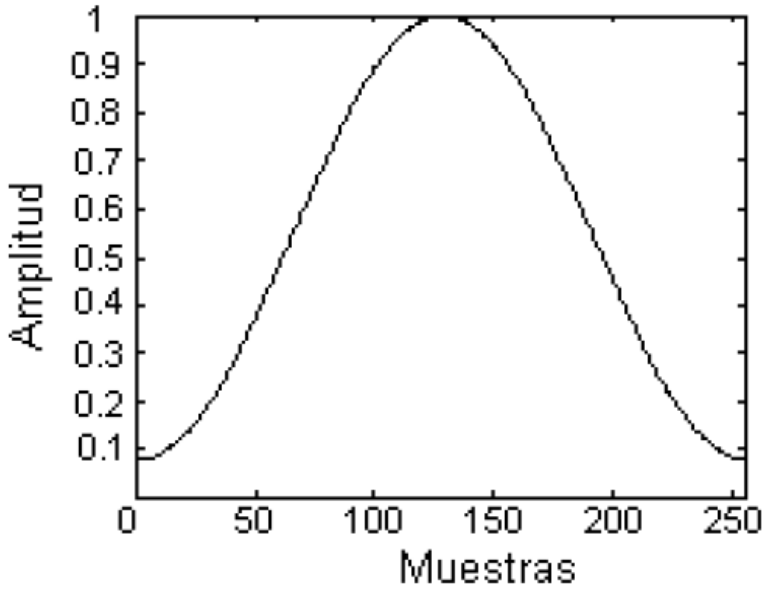


Fig. 2 Ventana de Hamming

El preénfasis es necesario debido a la atenuación de 6dB/octava que se produce conforme va aumentando la frecuencia. Es un filtrado cuya función es incrementar la relevancia de las componentes de alta frecuencia. Este proceso se puede implementar mediante un filtrado digital paso alto, diseñado mediante la siguiente ecuación en diferencias:

$$y[n] = x[n]-a \cdot x[n-1],\ a \in [0,1]$$

Ec.3 Ecuación en diferencias del filtro de preénfasis

Transformada Discreta de Fourier: En esta etapa se convierte cada trama de N muestras del dominio del tiempo al de la frecuencia mediante la Transformada Discreta de Fourier, definida por:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi k n/N},\ \ k = 0,1,2,\dots N-1$$

Ec.4 Transformada Discreta de Fourier

La implementación directa de esta ecuación es muy ineficiente, especialmente cuando la secuencia de longitud N es muy larga ya que para obtener un conjunto completo de coeficientes DFTse necesitan N^2 multiplicaciones complejas y N(N-1) sumas complejas, lo cual es un inconveniente. Por tanto se recurre a la Transformada Rápida de Fourier (FFT), algoritmo que permite implementar de manera rápida y eficiente la DFT[19].

En la elaboración de la FFT se explotan las propiedades de simetría y periodicidad del factor de fase w_N, basándose en el uso de la estrategia "divide y vencerás". Se divide la DFT de N muestras en DFTs más pequeñas, dónde N se representa como el producto de dos enteros L y M:

$$N=L\cdot M \qquad \text{Ec.5}$$

Así, la secuencia x(n) se almacena en una matriz bidimensional indexada por l y m, de manera que los índices n y k de la Ec.4 se reescriben según:

$$n = M\cdot l + m \qquad 0 \le l \le L-1,\ 0 \le m \le M-1 \qquad \text{Ec.6}$$

$$k = p + L\cdot q \qquad 0 \le p \le L-1,\ 0 \le q \le M-1 \qquad \text{Ec.7}$$

A continuación se divide la secuencia x(n) en M pequeñas secuencias de longitud L, se toman M pequeñas DFTs de L puntos y se combinan en una DFT más grande. Las secuencias x(n) y X(k) se pueden escribir como matrices x(l,m) y X(p,q), respectivamente, pudiendo reescribir la Ec. 4 como sigue:

$$X(p,q) = \sum_{m=0}^{M-1} \left\{ W_N^{mp} \left[\sum_{l=0}^{L-1} x(l,m) W_N^{lp} \right] \right\} W_M^{mq}$$

DFT M puntos — DFT L puntos

Ec.8 Transformada Rápida de Fourier

Ecuación que se desarrolla en un cálculo de tres pasos:

1. Calcular las DFTs de L puntos:

$$F(p,m) = \sum_{l=0}^{L-1} x(l,m) W_L^{lp}, \qquad 0 \le p \le L-1$$

Ec.9 DFT de L puntos

para cada una de las filas $m = 0,1,\dots,M-1$

2. Se calcula la nueva matriz rectangular G(p,m) definida como:

$$G(p,m) = W_N^{pm} F(p,m), \qquad 0 \le m \le M-1, 0 \le p \le L-1 \qquad \text{Ec.10}$$

3. Finalmente se calculan las DFTs de M puntos:

$$X(p,q) = \sum_{m=0}^{M-1} G(p,m) W_M^{mq}, \qquad 0 \le q \le M-1$$

Ec.11 DFT de M puntos

para cada columna $p = 0,1,\dots,L-1$ de la matriz G(p,m)

El número total de operaciones complejas para esta aproximación es :

$$C_{N0} = ML^2 + L + LM^2 < N^2$$ Ec.12 Num. de op. complejas de la FFT

Creación de las bandas de Mel: Por definición, un sonido de 1kHz , con 40 dB por encima del umbral de percepción, tiene un tono de 1000 Mels. Para computar las unidades Mels del sonido a una frecuencia f (Hz) se usa la siguiente fórmula:

$$Mel(f) = 2595 \cdot log_{10}\left(1 + \frac{f}{700}\right)$$ Ec.13 Escala Mel

La figura, muestra una gráfica de la escala Mel respecto a la frecuencia. Se aprecia la tendencia lineal de la escala Mel por debajo de los 1000Hz y el espaciamiento logarítmico por encima de los 1000Hz.

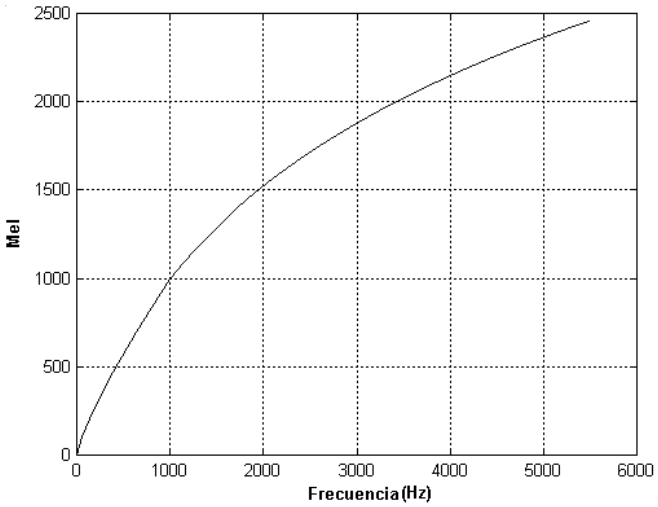


Fig. 3 Mel vs Frecuencia

Una manera de aproximarse a este espectro subjetivo es usar un banco de filtros muchos más estrechos y linealmente espaciados hasta los 1000Hz y más amplios y logarítmicamente espaciados a partir de dicha frecuencia. De este modo, se da más importancia a la información contenida en las bajas frecuencias, en concordancia con el comportamiento del oído humano.

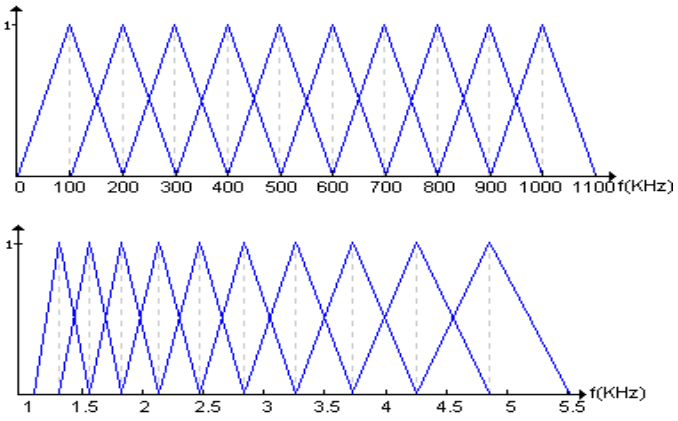


Fig. 4 Banco de filtros escala de Mel

Los filtros son aplicados directamente en el dominio de la frecuencia y su respuesta está dada por la siguiente ecuación:

$$\omega(f) = \begin{cases} \dfrac{2f}{BW-1}, & 0 \le f \le \dfrac{BW-1}{2}\ (Hz) \\ 2 - \dfrac{2f}{BW-1}, & \dfrac{BW-1}{2} \le f \le BW-1\ (Hz) \end{cases}$$

Ec.14 Filtros Escala de Mel

Seguidamente, se calcula la energía de la señal en cada una de las bandas de frecuencia en las que el banco de filtros divide el espectro, definida por la siguiente fórmula:

$$E(i) = \int_{F1}^{F2} |X(F)|^2 dF\,, \qquad i = 1,2,\dots,L$$

Ec.15 Energía por banco de frecuencias

F1: Frecuencia de inicio del filtro

F2: Frecuencia de fin del filtro

$|X(F)|^2$: Densidad espectral de energía

L: Número de filtros

Cepstrum: El cepstrum se define como la transformada inversa del logaritmo del módulo de la transformada de Fourier de la señal.

$$c(n) = TF^{-1}[\log|X(w)|], \qquad X(w) = TF[x(n)]$$

Ec.16 Cepstrum

Este algoritmo está basado en el modelo del tracto vocal, en el cual la voz se genera por una excitación producida por dos fuentes, la cual pasa a través de un filtro, cuya respuesta en frecuencia modifica el espectro añadiendo información al sonido. Por tanto si denotamos con E(w) y H(w) las transformaciones de la excitación y el filtro respectivamente, se cumple:

$$X(w) = E(w) \cdot H(w)$$

Ec.17 Expresión DFT función de la excitación y filtro

Reemplazando esto en la Ec.13 se tiene:

$$c(t) = TF^{-1}[\log(|E| * |H|)] = TF^{-1}[\log(|E|)] + TF^{-1}[\log(|H|)] = c_e(n) + c_h(n)$$

Ec.18 Cepstrum en función de la excitación y filtro

$$c_e(n) = c(n) * w_{lp}(n)$$
$$c_h(n) = c(n) * w_{hp}(n)$$

Ec.19 Cepstrums de la excitación y el filtro en función del cepstrum de la señal y las ventanas

$$w_{lp} = \begin{cases} 1, & |n| \leq N_1 \\ 0, & en\ otro\ caso \end{cases}$$

$$w_{hp} = \begin{cases} 0, & |n| \leq N_1 \\ 1, & |n| > N_1 \end{cases}$$

Ec.20 Ventanas paso alto y paso bajo

Es decir, el cepstrum de una señal es la suma del cepstrum de la excitación y del cepstrum del filtro, por tanto es posible separar las componentes espectrales de la señal pertenecientes a la excitación del tracto y del filtro usando ventanas paso bajo y paso alto respectivamente.

Transformada Inversa Discreta de Fourier: Ya que los coeficientes del espectro y su logaritmo son números reales, se pueden convertir al dominio del tiempo usando la Transformada Discreta Coseno (DCT[21]).

$$x[n] = \sqrt{\frac{2}{N-1}} \sum_{k=0}^{N-1} a[k] \cdot a[n] \cdot X^{C1}[k] \cos\left(\frac{\pi n k}{N-1}\right)$$

Ec.21 DCT

X^{C1}: Transformada de la secuencia x[n]

$$a[n] = \begin{cases} 2^{-1/2}, & n = 0, N-1 \\ 1, & otro\ caso \end{cases}$$

Ec.22 Ventanas paso alto y paso bajo

Finalmente el cálculo de los MFCC responde a la siguiente ecuación:

$$MFCC_j(i) = \sum_{k=1}^{NF} \log[E(j,k)] \cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{L}\right]; \quad i = 1,2,3,\ldots,p$$

Ec.23 MFCC

Donde:

k: Banda de frecuencias.

j: Trama en curso.

E(j,k): Energía de la banda k en la trama j.

NF: Número de bandas o Filtros

P: Número total de coeficientes MFCC.

- ***MP (Matching Pursuit):*** Recientemente las técnicas de selección adaptable se han vuelto muy populares dentro la comunidad científica encargada del estudio de señales, principalmente debido a su empleo en la búsqueda de representaciones de grandes clases de funciones. En una selección adaptable, el objetivo principal consiste en encontrar la representación de una función *f*, como una suma ponderada de elementos a partir de un diccionario redundante.

  Matching Pursuit es un algoritmo que escoge las formas de onda óptimas de un diccionario redundante, adaptadas también de forma óptima para descomponer una estructura de señal en una combinación no lineal de estas formas de onda. . Estos diccionarios redundantes se han empleado en diversas aplicaciones, por ser extremadamente flexibles. Inicialmente, S.Mallat propuso un algoritmo de selección adaptable de funciones llamado "Matching Pursuit", posteriormente, S.Jaggi propuso otro algoritmo llamado "High Resolution Pursuit".

  También se podría entender Matching Pursuit como un algoritmo que escoge la forma de onda (átomo) que más se aproxima a una parte local de la señal. Estos átomos se conforman a partir de la modulación, variación de la escala o traslación de una función tipo ventana. Estos átomos siguen una distribución de tipo Wiegner en tiempo-frecuencia. La suma de estas distribuciones individuales pueden proporcionar una buena representación de una señal en el plano tiempo-frecuencia.

  Diccionario: Un diccionario es una colección de formas de onda parametrizadas:

$$D = (g_\gamma : \gamma \in \Gamma) \qquad \text{Ec.24 Diccionario}$$

  Se puede definir a un diccionario **D** como una familia de átomos o vectores $(g_\gamma)_{\gamma \in \Gamma}$.

  Una familia genérica de átomos puede construirse a partir del escalado, traslación y modulación de una única función ventana g(t) Є $L^2(R)$, siendo g(t) real, diferenciable en todo $L^2(R)$. También se va a imponer que ||g||=1 , que la integral de g(t) no sea nula. A partir de esto, se define una función g(t), para una escala s >0, una frecuencia de modulación ξ y una traslación *u* tal que:

$$g_{\gamma(t)} = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t} \qquad \text{Ec.25 Átomo}$$

  Además definimos el índice γ=(s, u, ξ) como un elemento del espacio $\Gamma=R^+ \times R^2$. La familia $D=((g_\gamma(t)))_{\gamma \in \Gamma}$ es altamente redundante, donde sus propiedades han sido estudiadas.

  El comportamiento del algoritmo va a depender de los elementos que conforman el diccionario. Para poder representar de forma efectiva una función cualesquiera f(t), se ha de seleccionar un subconjunto de átomos $(g_{\gamma_n}(t))_{n \in N}$ de tal manera que:

$$f(t) = \sum_{n=-\infty}^{+\infty} a_n g_{\gamma_n}(t)$$

Ec.26 Representación de una función en función de los átomos de un diccionario

Dependiendo de qué átomos, g_{γ_n} (t) , se hayan escogido, los coeficientes de expansión a_n proporcionan información de ciertas propiedades de f(t).

Para señales altamente oscilatorias que incluyen gran variabilidad en su escala, no es posible definir, a priori, los límites tanto de escala como de modulación que definen los átomos tiempo-frecuencia g_{γ_n} (t) usados en la combinación.

De acuerdo al número de átomos contenidos en el diccionario, podemos encontrar diccionarios completos con N átomos, siendo N el número de átomos con el cual se puede decir que se caracteriza completamente a la señal, sobre completos con más de N átomos y subcompletos. También existen los diccionarios continuos, el cual contienen un número infinito de átomos. Aunque existe una gran cantidad de diccionarios propuestos, se van a exponer los más relevantes:

• Diccionarios triviales: En este tipo entran los diccionarios más simples. Aquí se encuentra el diccionario de Dirac, el cual es una colección de formas de onda que valen cero excepto en un único punto o el diccionario de Heaviside, que está conformado por un conjunto de ondas "escalón", las cuales saltan en un determinado punto cada vez. Para el diccionario de Heaviside los átomos de este diccionario ya no conforman una base ortogonal, pero se puede conseguir que para una señal exista una representación a partir de la combinación de los átomos.

• Diccionarios frecuenciales: En este grupo se encuentra el diccionario de Fourier, el cual es una colección de formas de onda sinusoides, indexadas a partir de $\gamma = (\omega, \nu)$, donde $\omega \in [0; 2\pi)$ se define como la frecuencia angular, y la variable $\nu \in \{0,1\}$ determina el tipo de fase, este diccionario conforma una base, ya que todos sus átomos son mutuamente ortogonales. Como ejemplo, un diccionario sobrecompleto de Fourier se obtiene a partir de sobremuestrear las frecuencias.

• Diccionarios tiempo-escala: En este conjunto un tipo de diccionario es el de Wavelets, este diccionario se compone de una colección wavelets trasladados y escalados.

La indexación se realiza a partir de $\gamma = (a, b, \nu)$, donde $a \in (0,\infty)$ is la variable de escala, $b \in [0; n]$ indica posición y $\nu \in \{0,1\}$ indica el género:

$$g_{(a,b,1)} = \varphi(a(t-b))\sqrt{a}\, g_{(a,b,0)} = \psi(a(t-b))\sqrt{a}$$ Ec.27 Diccionario Tiempo-Escala

Se obtiene entonces un diccionario con n átomos, a la vez que se genera una base ortonormal. Se pueden conseguir otros diccionarios tipo wavelets realizando variaciones a esta base. Las más destacadas son utilizando splines de wavelets definidas a partir de relaciones de filtrado a dos escalas. Esto conlleva a reglas de construcción más complicadas, pero mantiene la misma estructura de indexado, o, en otras palabras, se aplica de la misma manera en el algoritmo.

• Diccionarios Tiempo-Frecuencia: Una evolución del análisis de señales se centra en el estudio de los fenómenos en tiempo-frecuencia. Uno de los ejemplos más extendidos, es el diccionario de Gabor. Este trabaja con un indexado $\gamma = (\omega, \tau, \theta, \delta t)$, donde $\omega \in [0, \pi)$ determina la frecuencia, τ determina la posición, θ la fase y s la duración del átomo:

$$g_\gamma(t) = e^{\frac{(t-\tau)^0}{s^0}} \cdot \cos(\omega(t-\tau)+\theta)$$ Ec.28 Diccionario Tiempo-Frecuencia

Estos átomos concentran su energía en frecuencias cercanas a ω y tienen una extensión considerable en tiempo. Para generar diccionarios de Gabor discretos para análisis tiempo frecuencia, se generan una $\omega_k = k\Delta\omega$ y una $\tau_l = l\Delta\tau$, y una $\theta \in \{0, \pi/2\}$, donde $\Delta\tau$ y $\Delta\omega$ son lo suficientemente pequeños como para considerar los diccionarios completos. Este va a ser el diccionario con el que se va a realizar el análisis de las señales en este proyecto fin de carrera. Se va a generar a además una variación de s, $s_u = u\Delta s$, que nos permitirá analizar la señal incluso para diferentes escalas.

• Otros diccionarios : Siempre es posible juntar diccionarios para crear mega diccionarios, ya que la redundancia en estos no es perjudicial para el algoritmo.

Una vez introducido de forma general el algoritmo y posibles diccionarios implementados en este, se va a analizar de forma más extensa el comportamiento del Matching Pursuit utilizando el diccionario de Gabor.

Algoritmo: Sea un $f \in L^2(R)$ (f, señal a analizar). Se busca una combinación lineal de f sobre un conjunto de átomos escogidos de D, de tal manera que se consigue la máxima similitud con la estructura interna de f. Esto se puede conseguir a partir de la aproximación recurrente de f con las proyecciones ortogonales de los elementos en D. Entonces, como primer paso, f puede ser descompuesta en:

$$f = < f, g_{\gamma o} > g_{\gamma o} + Rf$$ Ec.29 Descomposición de f

$$< f, g_{\gamma o} > = \int_{-\infty}^{\infty} f(t) g_{\gamma o} dt$$ Ec.30 Proyecciones ortogonales

R*f* es el vector residuo obtenido tras la aproximación de *f* en la dirección de g_{y0}. Dado que g_{y0} es ortogonal a R*f*:

$$\|f\|^2 = < f, f > = | < f, g_{\gamma o} > |^2 + \|Rf\|^2$$ Ec.31 Aproximación de *f*

Para poder minimizar el residuo $\|Rf\|$, se debe escoger un $g_{y0} \in D$ tal que $|<f, g_{y0}>|$ sea máximo. Es decir, se ha de encontrar un átomo g_{y0} tal que cumpla la condición:

$$| < f, g_{\gamma o} > | \geq | < f, g_{\gamma o} > |, para\ todo\ \gamma$$ Ec.32 Condición

El algoritmo Matching Pursuit subdescompone el residuo R*f* , proyectando este sobre un vector o átomo en D, que se asemeja lo más posible a R*f*, de la misma manera que se hizo con *f*. Este proceso se repite con cada nuevo residuo obtenido, hasta que se cumpla una cierta condición establecida, como, por ejemplo, llegar a un número N de átomos o un valor concreto de la relación $\|R^n f\| / \|f\|^2$.

En definitiva, el Matching Pursuit se implementa de la siguiente manera:

1. Se define $R_0 f = f$.

2. Suponiendo que se calcula el residuo de orden n: $R^n f$, con $n \geq 0$, se tiene que elegir un elemento $g_\gamma \in D$ que se asemeje lo máximo posible al residuo $R^n f$, este g_γ ha de cumplir la siguiente condición:

$$| < R^n f, g_{\gamma^n} > | = sup_{\gamma \in \Gamma} | < R^n f, g_\gamma > |$$ Ec.33 Condición

$<R^n f, g_\gamma>$ es una función de correlación que mide la similitud entre sus factores.

3. Entonces, el residuo $R^n f$ se descompone en:

$$R^n f = < R^n f, g_{\gamma^n} > g_{\gamma^n} + R^{n+1} f$$ Ec.34 Descomposición del residuo

En esta igualdad se define el residuo para n+1.

Esta ya probado que para $n \to \infty$, $\|R^n f\| \to 0$.

En esta situación *f* vale: $\sum_{n=0}^{N-1} < R^n f, g_{\gamma^n} >$ Ec.35

3.2. Clasificación mediante RNA.

A continuación se trata a fondo las RNAs, detallando su funcionamiento y explicando teóricamente la manera de usarlas según las funciones en las que se apoyan las RNAs para simular el funcionamiento de las mismas. Como introducción se comenta un poco la historia de las mismas y el principio en el que se basan, dando así una razón suficiente para su uso.

- ***ANN (Artificial Neural Network),*** Los primeros modelos de redes neuronales datan de 1943 por los neurólogos McCulloch y Pitts. Años más tarde, en 1949, Donald Hebb desarrolló sus ideas sobre el aprendizaje neuronal, quedando reflejado en la "regla de Hebb". En 1958, Rosemblatt desarrolló el perceptrón simple, la primera red neuronal como tal y en 1960, Widrow y Hoff desarrollaron el ADALINE, que fue la primera aplicación industrial real. Minsky y Papert, en 1969, publicaron un análisis matemático que explicaba las deficiencias de las redes neuronales. Debido a esto y a la muerte de Ronsenblatt en 1971, los fondos para investigación en redes neuronales se terminaron. En 1986, se demostró que las redes neuronales podían ser usadas para resolver problemas prácticos y fue entonces que se empezó a investigar considerablemente sobre redes neuronales.

Las **redes de neuronas artificiales** (denominadas habitualmente como **RNA** o en inglés **ANN**) son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. Es frecuente referirse a ellas como **redes neuronales**.

El cerebro humano contiene aproximadamente 100,000 millones de neuronas. Cada neurona está conectada aproximadamente a otras 1000 neuronas, excepto en la corteza cerebral donde la densidad neuronal es mucho mayor. Las redes neuronales son una implementación muy sencilla de un comportamiento local observado en nuestros cerebros. El cerebro está compuesto de neuronas, las cuales son elementos individuales de procesamiento. La información viaja entre las neuronas, y basado en la estructura y ganancia de los conectores neuronales, la red se comporta de forma diferente.

La unidad básica de una RNA es la neurona de una sola entrada, como se muestra en la figura:

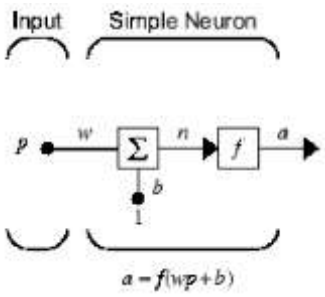


Fig. 5 Neurona simple

El funcionamiento de la neurona es tal que la entrada *p*, se multiplica por el peso *w*. En segundo lugar, la entrada pesada *wp* y la entrada escalar *b* (bias) se suman formando la entrada de red *n* (el bias traslada la función *f* hacia la izquierda una cantidad *b*. El bias es como un peso excepto que tiene un valor constante 1).Finalmente a la entrada de red *n* se le aplica la función de transferencia *f* para generar la salida escalar *a*.

Los parámetros *w* y *b* son ambos parámetros escalares ajustables. La idea central de las RNA es que dichos parámetros se ajusten de tal forma que la red muestre el comportamiento deseado según el tipo de problema a resolver.

Función de Transferencia: Los dos tipos de funciones de transferencia más usadas en RNA se muestran a continuación. En primer lugar, la función de transferencia lineal, se usa normalmente en la capa de salida de una RNA multicapa usadas como aproximadores.

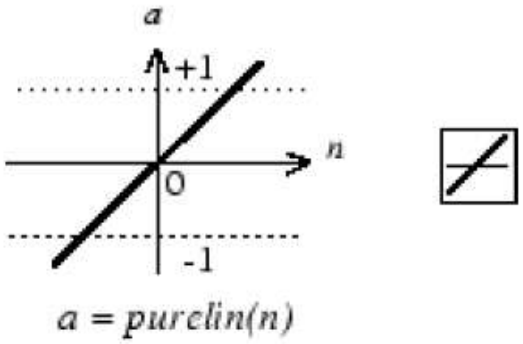


Fig. 6 Función de transferencia lineal

En segundo lugar, la función sigmoidea, cuya entrada puede ser cualquier valor entre más y menos infinito y produce una salida entre 0 y 1 en el caso de la sigmoidea logarítmica o entre -1 y 1 en el caso de la sigmoidea tangente hiperbólica.

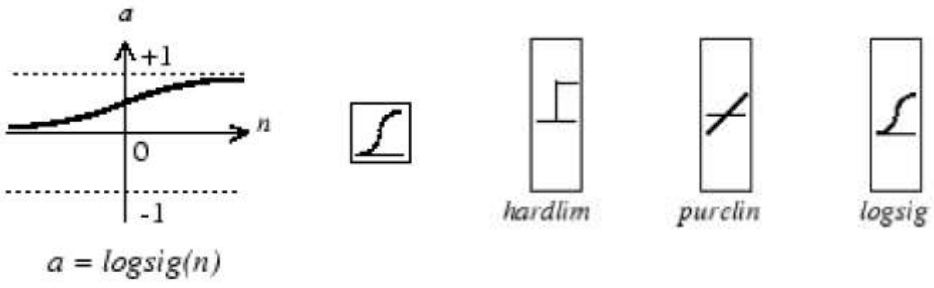


Fig. 7 Función de transferencia sigmoidea

Una función sigmoidea es una función continua de valores reales, con dominio en los reales, cuya derivada es siempre positiva y cuyo rango está restringido.

Vector de entrada: Se puede extender el concepto de neurona simple para el manejo de vectores de entrada. En una neurona con un vector de entrada (perceptrón), los elementos del vector se multiplican por los pesos y los productos que se obtienen se suman junto con el bias formando la entrada de red *n*, que es el argumento de la función de transferencia *f*.

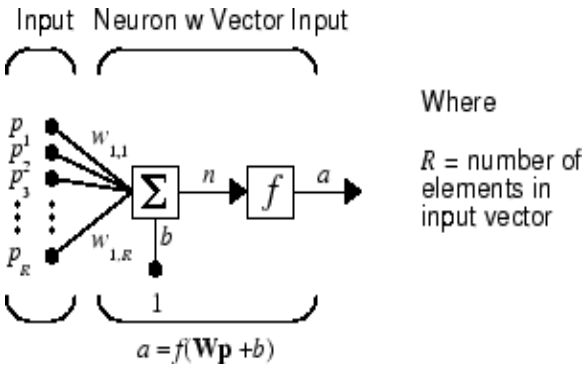


Fig. 8 Perceptrón

Un perceptrón usa entrenamiento supervisado. Si durante el entrenamiento la salida no es correcta, los pesos se reajustan de acuerdo a la siguiente fórmula:

$$w_{new} = w_{old} + \alpha(deseada - salida) \cdot entrada, \qquad \alpha \text{ es la tasa de aprendizaje}$$

Ec.36 Entrenamiento supervisado

Arquitecturas de las RNA: Dos o más neuronas de las descritas anteriormente pueden combinarse formando una capa neuronal y la combinación de dos o más capas forma propiamente una **Red Neuronal Artificial**.

- Capa de neuronas: Una RNA de una sola capa con R elementos de entrada y S neuronas es como sigue:

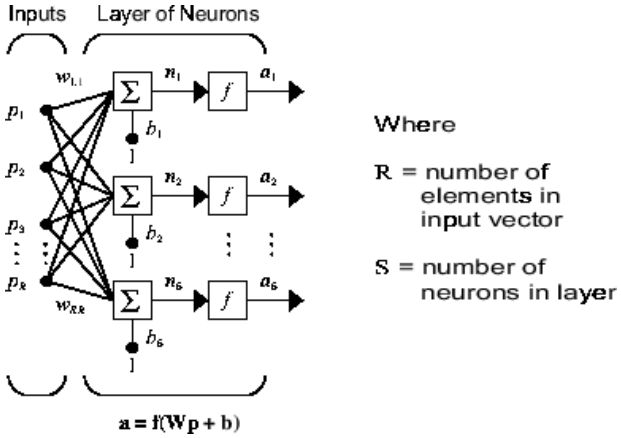


Fig. 9 Capa de múltiples neuronas

En este tipo de redes, cada elemento del vector de entrada p está conectado a todas las neuronas de la capa mediante la matriz de pesos **W**. Cada neurona *i-ésima* posee la correspondiente función que suma el bias y los productos del vector de entrada y los pesos, creando así su propia salida escalar *n(i).* Finalmente, las salidas de la capa neuronal forman el

vector columna *a*. Notar que el número de elementos de entrada no tiene por qué ser el mismo que el número de neuronas de la misma.

Se puede crear una RNA de una sola capa con diferentes funciones de transferencia juntando en paralelo dos o más RNA con un número determinado de neuronas cada una y diferentes funciones de transferencia cada RNA. En este caso, las diferentes redes en paralelo tendrán las mismas entradas y cada una creará algunas salidas del total.

La matriz de pesos a través de la cual el vector de entrada penetra en la red se expresa como sigue:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

Los índices de las filas en la matriz indican la neurona destino del peso correspondiente, los índices de las columnas indican el elemento de entrada que multiplica a dicho peso.

▪ <u>RNA de múltiples capas:</u> Así mismo, una RNA puede tener múltiples capas. En ese caso, cada capa tiene una matriz de pesos **W**, un vector de bias **b** y un vector de salida **a**. Por tanto, para distinguir entre las la matriz de pesos, bias y salida de una capa a otra el número de la capa se indica en forma de superíndice.

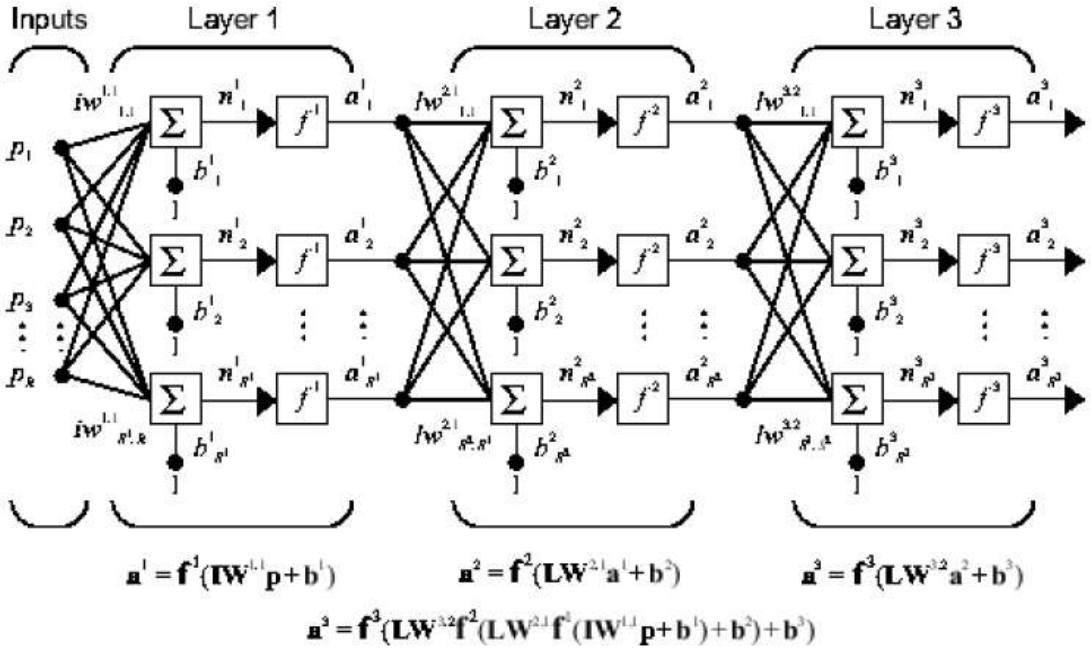


Fig. 10 Red de múltiples capas

Las RNA multicapa son bastante poderosas. De hecho, una RNA con dos capas en la que la primera capa tiene una función de transferencia de tipo sigmoidea y la segunda capa de tipo lineal, puede ser entrenada para aproximar relativamente bien cualquier tipo de función arbitraria (con un número finito de discontinuidades).

Pesos de la RNA: Los pesos en una RNA son el factor más importante para determinar la función de la RNA. Entrenar la RNA es el acto de presentar a la RNA las muestras y modificar los pesos para aproximar lo mejor posible la función deseada.

Hay dos tipos principales de entrenamiento, el **entrenamiento supervisado** que provee a la RNA de las entradas y las salidas deseadas, y el **entrenamiento no supervisado** que provee a la RNA únicamente de las entradas. En el **entrenamiento supervisado**, los pesos se modifican para reducir la diferencia entre la salida actual y la salida deseada, mientras que en el entrenamiento no supervisado, la RNA ajusta sus pesos de forma que las entradas similares causen salidas similares, la RNA identifica los patrones y las diferencias en las entradas sin ninguna asistencia externa.

Otro parámetro importante que forma parte del entrenamiento son los **epoch.** Un **epoch** es una iteración en la que se provee a la RNA de una entrada y se actualizan los pesos. Típicamente se requieren muchos **epochs** para entrenar una RNA.

Tipos de RNA: El tipo de RNA más común son las Redes Multicapa Feedforward (Multilayer Feedforward Networks) que son una extensión del perceptrón con múltiples capas "escondidas" entre las capas de entrada y salida. La función de activación es normalmente la función sigmoidea. En estas RNAs la información fluye en un solo sentido, es decir la salida de una capa actúa como entrada de la capa siguiente. El método más común para obtener los pesos es **backpropagation** (propagación hacia atrás) que es una forma de entrenamiento supervisado. El algoritmo de propagación hacia atrás básico trata de minimizar el error de la RNA usando las derivadas de la función de error. La medida del error más usada es MSE (mínimos cuadrados),

$$E = (deseada - salida)^2$$ Ec.37 Medida del error mse

El cálculo de las derivadas parciales se realiza según:

- **Neuronas de salida**, sea: $\delta_j = f'(red_j)(deseada_j - salida_j)$

$$\frac{\partial E}{\partial w_{ji}} = -salida_i \delta_j, \qquad \begin{matrix} j{:}neurona\ de\ salida \\ i{:}neurona\ en\ la\ última\ capa\ escondida \end{matrix}$$

Ec.38 Cálculo de las derivadas en la capa de salida

- **Neuronas escondidas**, sea: $\delta_j = f'(red_j)\sum(\delta_k w_{kj})$

$$\frac{\partial E}{\partial w_{ji}} = -salida_i \delta_j, \qquad \begin{matrix} j{:}neurona\ escondida \\ i{:}neurona\ capa\ anterior \\ k{:}neurona\ pr\acute{o}xima\ capa \end{matrix}$$

Ec.39 Cálculo de las derivadas en la capa escondida

Dicho cálculo se realiza hacia atrás, de ahí el nombre. Estas derivadas apuntan en la dirección del máximo incremento de la función de error. Una tasa de aprendizaje pequeña (paso pequeño) en la dirección contraria resulta en un decremento máximo de la función de error local:

$$w_{new} = w_{old} - \frac{\alpha \partial E}{\partial w_{old}}, \qquad \alpha\ es\ la\ tasa\ de\ aprendizaje$$

Ec.40 Cálculo de los pesos

La tasa de aprendizaje es importante ya que una tasa demasiado pequeña hace que la función converja muy lentamente y una tasa demasiado grande puede hacer que la función no converja nunca.

Capítulo 4

Experimento

En este capítulo, se trata a fondo el experimento llevado a cabo en el presente proyecto, se exponen todas las herramientas usadas así como los procedimientos llevados a cabo para alcanzar la solución final.

Por tanto, se divide éste capítulo en varios subapartados que permitan explicar mejor tanto el experimento como el proceso seguido para realizarlo. Los subapartados se organizan siguiendo un orden lógico de acontecimientos para poder realizar el experimento. Dicho orden lógico es por tanto realizar la recolección de datos, es decir, realizar las grabaciones necesarias para poder empezar con el experimento y trabajar un poco los datos para tenerlos organizados de manera que luego sea más fácil trabajar con ellos.

Tras realizar dichas grabaciones que se presentarán en detalle a continuación, se hace la extracción de características. Este subapartado se centra en tres tipos de características que se pueden extraer de la señal cuya base teórica ha sido explicada en el apartado anterior, y son MFCC, FBE y MP.

El último subapartado del presente capítulo se centra en RNA. Por tanto, se explica el procedimiento seguido para la elección del tipo de RNA que mejor se adapta a mi problema, siendo éste un problema de clasificación, la creación de la RNA en MatLab, el entrenamiento y validación de la RNA.

4.1. Grabación de la base de datos de sonidos ambientales.

Se pretende evaluar la precisión en el reconocimiento de entornos acústicos de las características extraídas aplicadas a la clasificación mediante RNA de fragmentos de 10 segundos de audio pertenecientes a 8 escenarios diferentes.

Dado que el presente proyecto está orientado al reconocimiento de entornos acústicos para aplicaciones móviles, como indica el título, las grabaciones llevadas a cabo, aunque de la mejor calidad posible, han sido tomadas haciendo uso de la grabadora de un Smartphone. La aplicación con la que se han hecho dichas grabaciones ha sido ***Pocket WavePad Sound Editor***, herramienta de libre distribución que permite configurar el formato de grabación de audio, la tasa de muestreo y el tamaño de la muestra en bits. En este caso, dichos parámetros de grabación han sido configurados previamente a la realización de las grabaciones de manera que se ha grabado en un formato WAV/PCM, a una tasa de muestreo de 44,1kHz y con un tamaño de muestra de 16 bits, dicha configuración se muestra en la siguiente figura.

Fig. 112 Configuración de la grabación de sonidos

Los entornos en los que se han realizado las grabaciones se han elegido para representar una variedad de ambientes tanto exteriores como interiores. Así mismo para cada entorno se han grabado varias escenas correspondientes a diferentes emplazamientos y diferentes horas del día, es decir, aunque dos escenas pertenezcan al mismo emplazamiento, éstas corresponden a diferentes momentos del día lo que hace que la colección de audio cubra al máximo el espectro de posibilidades dentro de los ambientes elegidos.

Finalmente, aunque de media se han hecho grabaciones de 2 minutos por escena, éstas se han dividido en fragmentos de audio de 10 segundos que son los fragmentos con los que se ha trabajado en los apartados posteriores. Todo esto, entornos, escenas por entorno, tiempo total de grabación por entorno y número de fragmentos de audio de 10 segundos obtenidos finalmente de cada entorno se muestra en la tabla a continuación.

ENTORNOS	ESCENARIOS	TIEMPO	FRAGMENTOS/ESCENA	Ent	Val
Apartamento (4637seg)	Düsseldorfstr., 18 CO	4’24’’/3’47’’/2’23’’/6’40’’	(1034seg) 103 fragmentos	77	26
	Bornstr., 123 DO	5’38’’/4’01’’/3’37’’/1’04’’	(860seg) 86 fragmentos	64	22
	Palmstraße, 5 ER	3’11’’/5’45’’/2’38’’/7’34’’	(1148seg) 114 fragmentos	85	29
	Klönnenstr., 64 DO	4’05’’/6’37’’/9’57’’/5’56’’	(1595seg) 159 fragmentos	119	40
Calle (10077seg)	Bornstr. DO	1’16’’/4’23’’/5’12’’/5’45’’	(996seg) 99 fragmentos	74	25
	Leopoldstr. DO	1’54’’/4’52’’/3’58’’/3’01’’	(825seg) 82 fragmentos	61	21
	Klönnenstr. DO	5’46’’/1’59’’/1’14’’/1’24’’	(623seg) 62 fragmentos	46	16
	Kaiserstr. DO	2’42’’/1’13’’/6’19’’/1’39’’	(713seg) 71 fragmentos	53	18
	Hansastr. DO	1’38’’/1’33’’/4’15’’/1’16’’	(522seg) 52 fragmentos	39	13
	Borsigplatz DO	3’03’’/1’09’’/1’05’’/1’24’’	(401seg) 40 fragmentos	30	10
	Märkischerstr. DO	3’02’’/3’06’’/2’05’’/2’10’’	(623seg) 62 fragmentos	46	16
	Burgholzstr. DO	2’02’’/3’26’’/1’39’’/3’20’’	(627seg) 62 fragmentos	46	16
	Pslmastr. ER	2’39’’/2’04’’/4’55’’/3’19’’	(777seg) 77 fragmentos	57	20
	Schlossplatz ER	3’10’’/2’59’’/2’47’’/1’03’’	(599seg) 59 fragmentos	44	15
	An der Drehbank GE	1’52’’/2’20’’/2’42’’/2’32’’	(566seg) 56 fragmentos	42	14
	Düsseldorferstr. CO	3’26’’/2’36’’/1’19’’/1’32’’	(533seg) 53 fragmentos	39	14
	Karl Kroz Straße ST	5’17’’/6’57’’/3’11’’/1’15’’	(1000seg) 100 fragmentos	75	25
	Hauptstr. DO	3’19’’/4’16’’/3’24’’/3’15’’	(854seg) 85 fragmentos	63	22
	Steinstr. DO	2’17’’/2’34’’/2’57’’/1’03’’	(531seg) 53 fragmentos	39	14
Estación (7150,41seg)	Dortmund Hbf	1’55’’/4’23’’/2’47’’/1’08’’	(613seg) 61 fragmentos	45	16
	Colonia Hbf	2’41’’/4’26’’/7’12’’/1’59’’	(978seg) 97 fragmentos	72	25
	Witten Hbf	1’59’’/2’17’’/2’21’’/3’16’’	(593seg) 59 fragmentos	44	15
	Hagen Hbf	2’32’’/3’17’’/7’31’’/3’05’’	(985seg) 98 fragmentos	73	25
	Essen Hbf	3’02’’/3’23’’/1’07’’/4’19’’	(711seg) 71 fragmentos	53	18
	Gevelsberg Hbf	4’56’’/2’32’’/6’54’’/1’45’’	(967seg) 96 fragmentos	72	24
	Witten Annen-Nord	53,41’’/6’40’’/2’18’’/3’37’’	(772,41seg) 77 fragmentos	57	20
	Ennepetal-Gevelsberg	5’34’’/2’19’’/1’57’’/1’29’’	(679seg) 67 fragmentos	50	17
	Gevelsberg Kipp	1’28’’/1’02’’/1’37’’/1’23’’	(330seg) 33 fragmentos	24	9
	Wetter Ruhr	2’09’’/2’01’’/2’10’’/1’46’’	(486seg) 48 fragmentos	36	12
Gimnasio (4054seg)	FitX Dortmund	3’42’’/1’32’’/2’08’’/3’05’’	(627seg) 62 fragmentos	46	16
	Mc Fit Dortmund	3’58’’/2’05’’/4’11’’/1’04’’	(678seg) 67 fragmentos	50	17
	FitX Colonia	9’10’’/3’34’’/4’56’’/1’22’’	(1142seg) 114 fragmentos	85	29
	FitX MÖ	3’43’’/3’01’’/2’23’’/2’44’’	(711seg) 71 fragmentos	53	18
	McFit Colonia	3’09’’/5’02’’/3’18’’/3’27’’	(896seg) 89 fragmentos	66	23
Oficina (7712,72seg)	AVU Dr. Peter R.	1’05’’/4’33’’/40,5’’/1’06’’	(444,5seg) 44 fragmentos	33	11
	AVU Dr. Ralph Holt.	15,42’’/11,6’’/42,3’’/2’05’’	(194,32seg) 19 fragmentos	15	4
	AVU Dr. Gregor Nac.	1’30’’/4’22’’/3’/3’18’’	(730seg) 73 fragmentos	55	18
	Gas Natural Pilar G.J.	1’33’’/2’34’’/1’53’’/25,2’’	(385,2seg) 38 fragmentos	10	8
	Dept. EN AVU	3’53’’/2’58’’/1’13’’/45,3’’	(529,3seg) 52 fragmentos	39	13
	AVU Adriana Hdz Pz	1’14’’/32’4’’/54’15’’/2’26’’	(5435seg) 543 fragmentos	407	136

Restaurante (8175,52seg)	AVU Cantina	1’51’’/54,52’’/9’38’’/1’21’’	(824,52seg) 82 fragmentos	61	21
	RWE Cantina	5’28’’/1’22’’/1’32’’/7’43’’	(965seg) 96 fragmentos	72	24
	Burger King	5’01’’/3’34’’/4’05’’/6’50’’	(1170seg) 117 fragmentos	87	30
	Mc Donalds	15’42’’/8’27’’/2’35’’/1’27’’	(1560seg) 156 fragmentos	117	39
	Steinbach Biergarten	9’58’’/5’57’’/3’08’’/5’46’’	(1489seg) 148 fragmentos	111	37
	Peter´s Brauerei	1’36’’/9’16’’/4’43’’/4’38’’	(1213seg) 121 fragmentos	90	31
	Pide Lahmakun	4’06’’/3’05’’/2’39’’/3’53’’	(823seg) 82 fragmentos	61	21
Supermercado (6470seg)	Kaufland DO	3’31’’/1’33’’/3’25’’/5’20’’	(829seg) 82 fragmentos	61	21
	Kaufland CO	3’18’’/1’38’’/9’13’’/1’37’’	(946seg) 94 fragmentos	70	24
	Kaufland ER	1’15’’/5’46’’/5’17’’/1’04’’	(802seg) 82 fragmentos	61	21
	Lidl Bornstr. DO	2’18’’/1’18’’/2’15’’/1’21’’	(432seg) 43 fragmentos	32	11
	Lidl ER	3’12’’/1’08’’/1’36’’/1’45’’	(461seg) 46 fragmentos	34	12
	Lidl ES Hbf	3’’/1’04’’/2’13’’/1’40’’	(477seg) 47 fragmentos	35	12
	Aldi WEZ DO	1’53’’/1’48’’/1’07’’/2’53’’	(461seg) 46 fragmentos	34	12
	Aldi ER	1’47’’/2’19’’/2’50’’/1’32’’	(508seg) 50 fragmentos	37	13
	Netto DO	5’23’’/3’44’’/1’57’’/2’36’’	(820seg) 82 fragmentos	61	21
	Rewe Bornstr. DO	1’21’’/1’08’’/1’34’’/2’56’’	(419seg) 41 fragmentos	30	11
	Rewe DO Hbf	1’23’’/1’05’’/1’30’’/1’17’’	(315seg) 31 fragmentos	23	8
Tren (5019seg)	S-Bahn S5	3’21’’/1’59’’/1’14’’/1’24’’	(478seg) 47 fragmentos	35	12
	S-Bahn S8	1’04’’/4’30’’/1’36’’/1’16’’	(506seg) 50 fragmentos	37	13
	U-Bahn 46	3’03’’/1’09’’/1’05’’/1’54’’/2’	(551seg) 55 fragmentos	41	14
	Regional R4	1’24’’/2’53’’/3’52’’/3’02’’	(671seg) 67 fragmentos	50	17
	Regional R7	3’06’’/1’45’’/2’05’’/2’07’’	(543seg) 54 fragmentos	40	14
	Regional R8	1’45’’/3’18’’/2’24’’/1’24’’	(531seg) 53 fragmentos	39	14
	Regional R16	1’20’’/1’46’’/1’25’’/2’03’’	(394seg) 39 fragmentos	29	10
	ICE	3’04’’/3’15’’/2’52’’/1’07’’	(618seg) 61 fragmentos	45	16
	IC	5’48’’/1’53’’/2’47’’/1’39’’	(727seg) 72 fragmentos	54	18
Total	**67**	**53295,65’’ (14h 48’ 16’’)**	**5298 fragmentos**	**3931**	**1367**

Tabla 4. Entornos / Escenas / Fragmentos de audio

Llegados a este punto, existen 14h 48’ 16’’ de grabaciones en total, 5298 fragmentos de 10 segundos pertenecientes a 8 entornos diferentes, de los cuales uso 300 fragmentos de cada entorno para entrenamiento de la RNA y 100 fragmentos de cada entorno para validación, es decir, en total 2400 fragmentos para entrenamiento (8 para crear la red y 2392 para entrenarla) y 800 fragmentos para validar la RNA.

En este punto del apartado de experimentos no se hace distinción en las señales debido al tipo de características ya que las señales son las originales obtenidas de las grabaciones realizadas y por tanto dicha distinción no tendría sentido en este punto.

Por tanto se concluye este subapartado habiendo obtenido los fragmentos de señal de audio de 10 segundos preparados para realizar la extracción de características y el entrenamiento y validación de las RNA.

4.2. Extracción de características de las señales.

Hasta el momento, se ha realizado la recolección y preparación de los datos para llevar a cabo el experimento. Se podría decir que es a partir de éste punto dónde empieza el experimento como tal ya que es a partir de aquí cuando se empieza a trabajar con MatLab para, mediante procedimientos numéricos, extraer la máxima información posible de los fragmentos de señal de audio.

Como se ha explicado en la primera parte del capítulo anterior, se van a extraer 3 tipos de características mediante dos algoritmos distintos de extracción de características, MFCC y MP. Del algoritmo de extracción de los coeficientes cepstral, además de dichos coeficientes se extraerá también la energía media por banco de frecuencia en la escala de Mel. Por tanto, de dicho algoritmo se extraen 64 coeficientes cepstral y 20 coeficientes de energía media perteneciente cada uno a un banco de frecuencia de los 20 bancos de frecuencia en la escala de Mel en los que se ha dividido el espectro de frecuencias audibles por el oido humano (20 – 20000 Hz).

Del algoritmo de Matching Pursuit, se obtienen 16 coeficientes que representan la señal como suma ponderada del diccionario de Gabor, creado también en MatLab.

Por tanto, se puede dividir este punto en dos nuevos subapartados. En el primero de ellos explica el procedimiento funciones desarrolladas en MatLab para extraer los 64 coeficientes cepstral y los 20 coeficientes de energía mediante el algoritmo MFCC. En el segundo punto se hace lo mismo pero para el caso de los 16 coeficientes MP, explicando previamente el proceso de generación del diccionario de Gabor.

Finalmente, dado que en las pruebas previas a la realización del proyecto en las que se han evaluado la precisión de distintas técnicas de extracción de características usando como clasificador RNA, siendo las que mejores resultado han producido las que se presentan aquí, se crea el conjunto de características que se va a utilizar en cada uno de los tres casos mediante la concatenación de las mismas, así como los distintos sets de creación (CreaSet), entrenamiento (TrainSet) y validación (ValidSet) que se han usado más adelante para crear, entrenar y validar las RNA.

- ***MFCC, Mel Frequency Cepstral Coefficients.***

Para la extracción de los coeficientes cepstral así como para los coeficientes de energía media por banco de frecuencia en la escala de Mel, se usa la función ***mfcc.m*** del Toolbox de

libre distribución ***mfcc*** de MatLab que directamente extrae dichos coeficientes a partir del vector de la señal (s), la frecuencia de muestreo (fs), duración de la ventana de análisis temporal (Tw), tiempo de superposición entre ventanas (Ts), coeficiente de preénfasis (alpha), el tipo de ventana o la función del tipo de ventana que se vaya a usar (ventana), rango de frecuencias (R), número de bancos de filtros que se usaran en la escala de Mel lo que constituirá también el número de coeficientes de energía media que se usan (M), número de coeficientes cepstral a extraer (C) y el parámetro de estiramiento o liftering (L).

Dicha función realiza la extracción de los coeficientes cepstral siguiendo el esquema a continuación, que es el mismo que el mostrado en el capítulo 3 cuando se explica teóricamente el algoritmo.

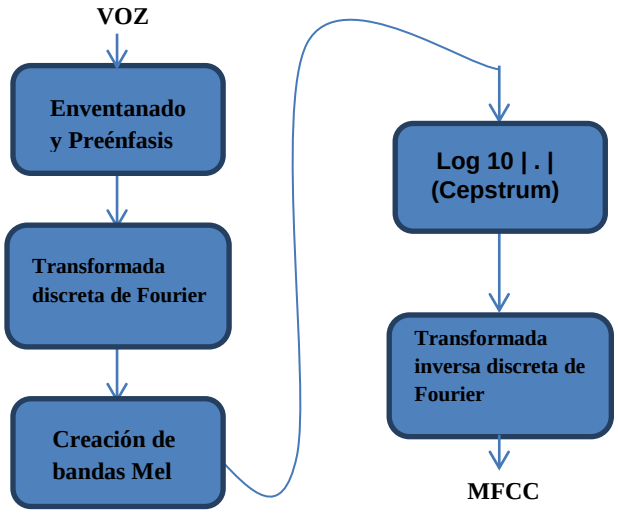


Fig. 13 Esquema de parametrización para la obtención de MFCC

La llamada a esta función y la extracción de los coeficientes cepstral así como los coeficientes de energía en MatLab es la siguiente:

>> [CC, FBE, frames] = mfcc (s, fs, Tw, Ts, alpha, ventana, R, M, C, L)

s: Vector columna que representa un fragmento de 10 segundos de audio

s = wavread (señal, fs) → s [44100 x 1]

fs: Frecuencia de muestreo, 44100Hz

Tw: Ventana de análisis temporal, Tw = 35ms

Ts: Tiempo de superposición, Ts = 15ms

Alpha: Coeficiente de preénfasis, alpha = 0,98 (Valor usado comúnmente en el reconocimiento del habla)

Ventana: Tipo de ventana para el enventanado. En este caso Hamming

hamming = @(N) (0.54-0.46*cos(2*pi*[0:N-1].'/(N-1)))

R: Rango de frecuencias, R = [20, 20000] Hz

M: Número de bancos de filtros en la escala de Mel, M = 20 (valor que define el número de coeficientes FBE[22])

C: Número de coeficientes cepstral, C = 64

L: Parámetro de liftering, típicamente L = 22

El resultado de dicha extracción, como se puede observar en la llamada a la función, son dos matrices de características, la primera de ellas MFCC de tamaño 64x2857 que contiene los vectores de características, correspondientes a cada ventana en la que se divide la señal, como columnas. La segunda matriz, FBE de tamaño 20x2857, contiene los vectores de características de energía media por banco de frecuencias en la escala de Mel, correspondientes a cada ventana temporal en la que se divide cada fragmento de señal, como columnas.

Dado que para el entrenamiento de la RNA se necesita un único vector de características que defina la señal, es necesario trabajar un poco estas matrices para obtener los vectores de características que interesan. Por tanto, basándose en la experiencia del experimento llevado a cabo en [1], y se reduce la dimensionalidad usando el algoritmo PCA. Para ello, se agrupa todas las columnas en un único vector de características. Se comprime luego usando PCA a 64 dimensiones en el caso de la matriz MFCC y a 20 dimensiones en el caso de la matriz FBE, preservando el 75% de la varianza.

Así, se obtienen dos vectores de características por cada fragmento de señal de 10 segundos, 64MFCC y 20FBE, que son dos de los tipos de características con los cuales se evalúa la precisión de las RNA.

- ***MP, Matching Pursuit.***

Como ya se ha comentado anteriormente, MP es una técnica de selección adaptable en la que se pretende representar una función como suma ponderada de los elementos de un diccionario óptimo. También se han comentado algunos de los diferentes diccionarios que pueden usarse, siendo el diccionario en tiempo – frecuencia de Gabor el elegido en el presente proyecto.

En MatLab, la creación de dicho diccionario es como sigue:

```
[rg, sigmas, gabors] = Gabor (M, N)

rg = ( ( - ( M/2 ) ) : ( ( M/2 ) - 1 ) )’

sigmas = exp (log (2 ) : ( log ( 200 ) - log( 2 ) ) / ( N-1 ) :log ( 200 ) )-1

gabors = exp(-.5*(rg. ^ 2 ) * sigmas. ^ (-2 ) ).* cos ( rg*sigmas. ^ (-1)*2*pi*2 )
```

Siendo M el número de muestras de la señal a la que se le va a aplicar el algoritmo MP y N el número de bases que queremos que compongan el diccionario, que será igual al número de coeficientes MP que queremos extraer, por tanto en este experimento, M=44100 y N=16.

Como en el caso de MFCC, para la extracción de los coeficientes MP se ha usado la función ***matchPurs.m*** contenida en el Toolbox de libre distribución de MatLab ***matchPurs*** a la que se le pasa como argumento la señal (s) y el diccionario creado (dic o gabors) y devuelve la proyección del residuo en cada elemento del diccionario (S), el residuo (R), los índices de los vectores de los elementos del diccionario usados (index) y los coeficientes MP (MP).

```
>> [S, R, MP, index] = matchPurs (s, dic)
```

El resultado de la extracción, como se puede observar en la llamada a la función, además de los vectores S y R que corresponden respectivamente a la proyección del residuo en cada elemento del diccionario dado y al residuo, se obtiene el vector de coeficientes Matching Pursuit MP de tamaño 1x16.

En este caso no es necesario reducir la dimensionalidad del resultado obtenido del algoritmo MP ya que es un único vector de características que define la señal. Así, se ha obtenido un vector de características por cada fragmento de señal de 10 segundos, 16MP, el cual para cada fragmento, se concatena con el vector 64MFCC obtenido de la extracción anterior para formar el tercer tipo de características para evaluar con la RNA, 64MFCC+16MP.

4.3. Redes Neuronales.

Este punto, como se ha dicho anteriormente, se centra en la creación de las RNA. En el capítulo 2, antecedentes, se puede comprobar que para dos de los tipos de características bajo estudio en este proyecto ya se han realizado experimentos, los expuestos en el susodicho capítulo. En este caso, la diferencia estriba en el uso de las RNA como clasificadores.

Ya que se trata con tres tipos de características diferentes, no se puede usar la misma RNA para los tres tipos sino que hay que crear una RNA adecuada a cada tipo de características. Esto quiere decir que aunque se usa el mismo tipo de RNA en los tres casos, ya que de acuerdo al problema que nos ocupa siendo este de clasificación uso el tipo de RNA que mejor se adapta a ello, para cada tipo de características varían el número de neuronas de la capa oculta.

En el capítulo 3 se explica el fundamento y funcionamiento teórico de las RNA, así como la motivación para su estudio y su uso. En este apartado se explica ampliamente la creación, entrenamiento y validación de cada una de las tres RNA en MatLab. Al final del presente punto se tendrá también una idea general del funcionamiento y manejo del Toolbox ***Neural Networks*** de MatLab.

Para empezar a trabajar con RNA hay varios aspectos a tener en cuenta en la elección de la RNA óptima para la resolución del problema, como es el tipo de RNA que se va a usar ya que MatLab ofrece una amplia variedad de posibilidades. En este caso concreto se busca una RNA óptima para problemas de clasificación. La clasificación es el proceso de identificación de un dato dentro de un conjunto posible de resultados. Una red puede ser entrenada para identificar distintos tipos de señales y esa es la meta del presente trabajo.

Para ello, se elige una RNA **feed-forward backpropagation**, ya que de acuerdo con la documentación del Toolbox de MatLab y la bibliografía consultada es el tipo óptimo para el presente problema. En el capítulo 3, donde se trata de los fundamentos teóricos de las RNA ya se dio una pequeña pista del tipo de RNA usada y por tanto queda explicado en dicho capítulo el contenido teórico de este tipo de redes. Aun así, se hace un breve recordatorio diciendo que las redes **feed-forward** son una extensión del perceptrón básico con varias capas, en este caso la capa de entrada, un capa escondida y la capa de salida, la función de activación es una sigmoidea y la información fluye en un solo sentido. Respecto al algoritmo **backpropagation**, es el método más común para obtener los pesos de la red, es un tipo de entrenamiento supervisado.

Los pasos seguidos para el diseño y uso de las RNA han sido:

a) Crear el conjunto de datos de creación de cada RNA (CreaSet).

b) Crear el conjunto de datos de entrenamiento (TrainSet).

c) Crear el conjunto de datos de validación (ValidSet).

d) Crear las RNA (usar el conjunto de datos de creación).

e) Entrenar las redes (usar el conjunto de datos de entrenamiento).

f) Validar la red para ver si ha aprendido y generalizado (usar el conjunto de datos de validación).

g) Uso de las RNA aplicando datos nuevos, posiblemente diferentes a los datos de creación entrenamiento y validación.

a) Los puntos **a)**, **b)** y **c)** se pueden condensar en un mismo punto. A partir de todos los datos que tenemos, es decir las características pertenecientes a cada fragmento de 10 segundos de señal, y basándose en lo leído en los antecedentes y bibliografía, se ha usado ¾ partes de todo el conjunto de datos para cada tipo de extracción de características como conjunto de datos de entrenamiento y la ¼ parte restante como conjunto de datos de validación. Un punto importante para realizar esta separación ha sido coger todo el conjunto de muestras conocidas y separarlo en dos grupos independientes (ortogonales). La manera de conseguir esto es usando grabaciones de escenas diferentes de un mismo entorno para entrenamiento y validación. Por tanto, durante la obtención de las grabaciones primero se han grabado las que se usan para entrenamiento y una vez creado este conjunto de datos, en escenas totalmente diferentes, horarios diferentes, etc. se han grabado las que forman parte del conjunto de datos de validación. Para elegir un buen conjunto de entrenamiento hay que tener en cuenta ciertos aspectos como que las muestras sean una buena representación de la población general, que contengan miembros de todas las clases y que las muestras en cada clase contengan un amplio rango de variaciones o efectos de ruido, condiciones que se han tenido en cuenta a la vista de la tabla en la que se muestran los tipos, escenarios y horarios de las grabaciones realizadas. Estas mismas consideraciones han sido tomadas para la creación del conjunto de validación.

Como conjunto de datos de creación, se ha cogido una muestra representativa de cada escenario del conjunto de datos de entrenamiento. Por tanto, de los **2400** conjuntos de características (300 pertenecientes a cada entorno) usados para entrenamiento he separado **8** conjuntos en una matriz llamada **CreaSet** que se ha usado para la creación de la RNA, agrupando los restantes **2392** en una matriz llamada **TrainSet** usada para el entrenamiento de la RNA. Este proceso es el mismo para cada tipo de características extraídas, es decir para 64MFCC, 20FBE y 64MFCC+16MP.

El tamaño del conjunto de datos de entrenamiento está relacionado con el número de neuronas de la capa oculta así como con los pesos que hay que calcular para ajustar la RNA, por tanto hay que tener en cuenta que un conjunto de datos de entrenamiento demasiado

pequeño podría no ser válido para resolver el sistema de ecuaciones que tiene como incógnitas los pesos de la RNA.

La ¼ parte restante, es decir **400** conjuntos de características (100 por entorno) se agrupan en una matriz llamada **ValidSet** que uso para la validación de la RNA. Por supuesto este proceso también se realiza por triplicado.

Así mismo, para cada una de las matrices de características creadas anteriormente se crea una matriz de clases (**CreaT**, **TrainT** y **ValidT**) que indica a qué clase pertenece cada conjunto de características que constituye cada una de las columnas de las matrices creadas anteriormente. Esto sirve para realizar el entrenamiento supervisado así como para comprobar la validación de los datos clasificados por la RNA.

De acuerdo a esto se tiene:

Matrices	Tamaño de la matriz 64MFCC	Tamaño de la matriz 20FBE	Tamaño de la matriz 64MFCC+16MP
CreaSet = []	64 x 8	20 x 8	80 x 8
CreaT = []	8 x 8	8 x 8	8 x 8
TrainSet = []	64 x 2392	20 x 2392	80 x 2392
TrainT = []	8 x 2392	8 x 2392	8 x 2392
ValidSet = []	64 x 800	20 x 800	80 x 800
ValidT = []	8 x 800	8 x 800	8 x 800

Tabla 5. Tamaño de las matrices

Un punto importante en RNA el procesamiento de los datos de entrada y salida y sobre todo los de entrada, lo que se conoce con el nombre de pre procesamiento. Esto consiste en adaptar los datos de entrada a la RNA de acuerdo a sus necesidades. En este caso se ha implementado mediante la función ***mapminmax*** que transforma los datos de entrada tal que los valores caigan dentro del intervalo [-1, 1].

d) Las RNA se han creado usando los comandos específicos del Toolbox ***Neural Networks*** de MatLab, por supuesto teniendo en cuenta lo mencionado anteriormente respecto al número de capas de la red y el número de neuronas de cada capa escondida. Para la mayoría de problemas, una capa escondida es suficiente, por tanto es lo que se ha usado. El número de neuronas de esta capa escondida es muy importante ya que demasiado pocas hacen que la RNA no sea capaz de aprender los detalles y demasiadas hacen que aprenda detalles insignificantes, por tanto se ha empezado con un número pequeño de neuronas y se ha ido aumentándolo hasta obtener unos resultados satisfactorios tal y como indica la bibliografía consultada.

Las redes se crean en MatLab mediante la orden que se muestra a continuación donde el comando newff crea una red ***Feedforward backpropagation***, el primer parámetro de entrada es el nombre de la variable donde se crea la red, el segundo parámetro es la matriz que contiene los datos para crear la red y como este algoritmo de aprendizaje realiza el entrenamiento supervisado, se le pasa tanto para la creación de la red como para su entrenamiento la matriz CreaT (o TrainT en el caso de entrenamiento) como salidas deseadas.

```
red = newff (red1, CreaSet, CreaT)
```

Por tanto cada una de las redes neuronales creadas está compuesta como se muestra a continuación.

```
>> red.net4
ans =
    Neural Network object:
    architecture:
        numInputs: 1
        numLayers: 2
      biasConnect: [1; 1]
     inputConnect: [1; 0]
     layerConnect: [0 0; 1 0]
    outputConnect: [0 1]
       numOutputs: 1  (read-only)
    numInputDelays: 0  (read-only)
    numLayerDelays: 0  (read-only)
    subobject structures:
            inputs: {1x1 cell} of inputs
            layers: {2x1 cell} of layers
           outputs: {1x2 cell} containing 1 output
            biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1
input weight
      layerWeights: {2x2 cell} containing 1
layer weight
    functions:
          adaptFcn: 'trains'
         divideFcn: 'dividerand'
       gradientFcn: 'gdefaults'
```

```
>> red.net4.layers{1}
ans =
       dimensions: 11
      distanceFcn: ''
        distances: []
         initFcn: 'initnw'
             name: 'Hidden Layer'
     netInputFcn: 'netsum'
    netInputParam: [1x1 struct]
        positions: [0 1 2 3 4 5 6 7 8 9 10]
             size: 11
     topologyFcn: 'hextop'
     transferFcn: 'tansig'
    transferParam: [1x1 struct]
         userdata: [1x1 struct]

>> red.net4.layerweights{2,1}
ans =
         delays: 0
        initFcn: ''
          learn: 1
       learnFcn: 'learngdm'
     learnParam: [1x1 struct]
           size: [1 11]
       userdata: [1x1 struct]
       weightFcn: 'dotprod'
```

```
        initFcn: 'initlay'
      performFcn: 'mse'
        plotFcns:
{'plotperform','plottrainstate','plotregression'}
        trainFcn: 'trainlm'
    parameters:
      adaptParam: .passes
     divideParam: .trainRatio, .valRatio
```

```
weightParam: [1x1 struct]
```

El número de neuronas de la capa escondida son 7 para la Red1 (64MFCC), 9 para la Red2 (20FBE) y 6 para la Red3 (64MFCC+20FBE). Este se ha obtenido mediante un procedimiento de prueba y error, se ha probado con diferente número de neuronas en la capa escondida de cada red eligiendo la que mejor resultados producía.

e) Una vez creada la red, haciendo uso del conjunto de datos de entrenamiento, se entrena mediante los comandos específicos del Toolbox de MatLab para este fin.

```
red = train (red, TrainSet, TrainT)
```

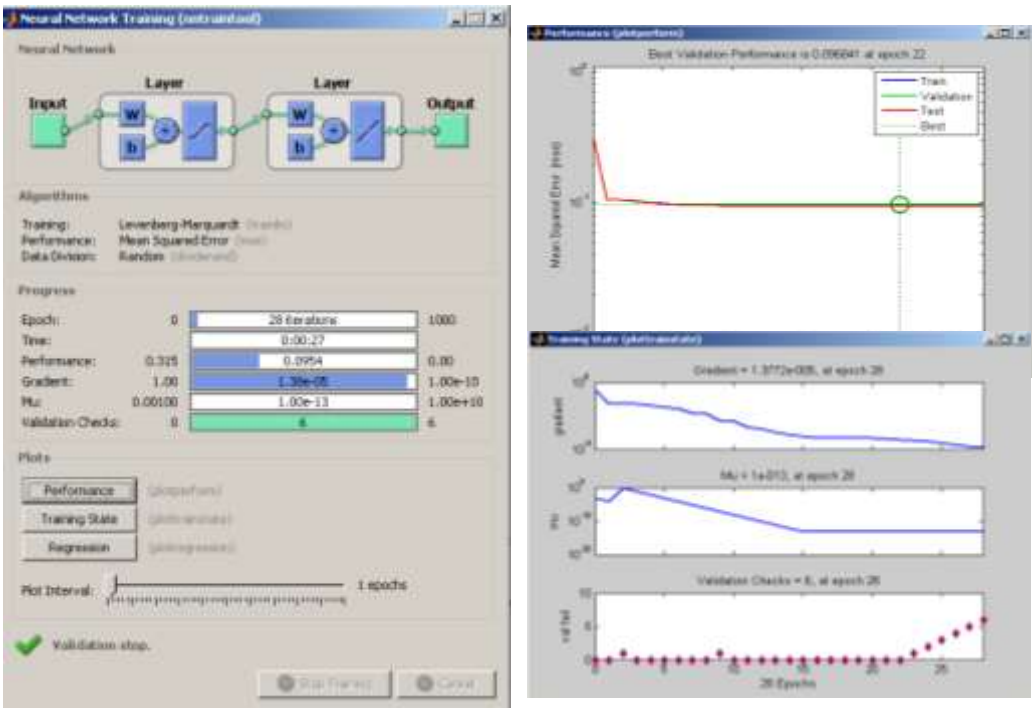


Fig. 14 Captura de la precisión y estado del entrenamiento

Ahora como en el caso anterior, se hace uso de la matriz de clases ya que el algoritmo backpropagation es un algoritmo de entrenamiento supervisado y por tanto necesita conocer las salidas deseadas para realizar el aprendizaje.

f) Validar la red es simplemente simular el conjunto de datos ValidSet con la RNA creada anteriormente. En realidad en lo presente, hay que validar los tres conjuntos de datos diferentes con la RNA correspondiente a cada uno. Esta validación da una idea de la calidad de la red. Un error común es validar la red usando las mismas muestras que se han usado para entrenar la RNA, error porque la red está optimizada para dichas muestras y obviamente la precisión es muy buena, por otro lado no da ninguna indicación de lo buena que es la red o de lo bien que ésta es capaz de clasificar las entradas que no estaban en el conjunto de entrenamiento.

Se pueden usar varias formas de medir la precisión de una RNA basándose en los resultados del conjunto de validación, en este proyecto se ha usado MSE (mean square error). Otro método alternativo para estimar la tasa de error de la RNA es resamplear. La idea básica es iterar el proceso de entrenamiento y validación múltiples veces, en este caso no se ha usado dicho método sino MSE.

La simulación del conjunto de datos de validación con la RNA en MatLab se hace de la siguiente manera:

```
OUT = sim (red, ValidSet)
```

Como en el caso del pre procesamiento de los datos, aunque no tan importante, podría ser interesante realizar un pos procesamiento a los datos de salida. Esto se realiza mediante la función inversa a la usada en el pre procesamiento, ***mapminmax***, adaptando así los datos de salida a la forma en la que se desea observar.

OUT es una matriz del mismo tamaño y forma que ValidT. En el caso óptimo, OUT debería ser completamente igual a ValidT. Cada columna de la matriz OUT representa la clase en la que la RNA ha clasificado cada una de las columnas de características de la señal que componen la matriz ValidSet.

4.4. Validación de los sonidos ambientales en personas

Por último cabe mencionar un pequeño experimento realizado con el fin de comparar los resultados obtenidos artificialmente mediante las técnicas descritas frente a la precisión del cerebro humano sin verse influido por las imágenes captadas.

Este experimento ha consistido en evaluar la precisión en la clasificación de los sonidos ambientales por las personas. El grupo de personas en las que se ha realizado el experimento estaba compuesto por 20 personas de ambos sexos y edades comprendidas entre los 18 y los 57

años, lo que permite tener una visión más general ya que la calidad de la audición de una persona varía, además de por las condiciones de vida de dicha persona, también por la edad. Además, aunque en menor medida el tipo de vida que haya llevado la persona influye, se ha cubierto un amplio espectro ya que las personas en las que se ha realizado el test pertenecen a entornos distintos cubriendo varios continentes, países y ciudades.

Cada una de las personas que componen el grupo de pruebas ha escuchado 40 fragmentos de 10 segundos de sonidos ambientales pertenecientes todos ellos a los fragmentos que se han utilizado para la validación de las RNA. En este conjunto de 40 fragmentos de sonidos ambientales, había 5 fragmentos de 10 segundos pertenecientes a cada entorno. Dichos sonidos han sido escuchados de manera aleatoria sin seguir ningún patrón, de tal forma que no se permita la identificación de los mismos de otra manera que no sea realmente el reconocimiento acústico.

Los resultados del experimento se muestran en el capítulo siguiente.

Capítulo 5
Resultados

Tras haber sido explicados en el capítulo anterior los pasos seguidos para extraer la información interesante de las señales de audio grabadas en los 8 entornos diferentes en los que se ha centrado, en este capítulo se exponen los resultados obtenidos de dicho análisis para cada tipo de extracción de características.

Si se examinan los resultados en el orden en que se han desarrollado los experimentos, primero debería centrarse en la recolección de los sonidos ambientales, evaluando luego la extracción de las características para por último observar la precisión de las RNA para cada tipo de extracción de características, proveyendo una evaluación empírica en 8 tipos de sonidos ambientales.

Para los experimentos se han grabado sonidos ambientales de una amplia variedad de localizaciones y escenas diferentes, por tanto, los sonidos tienen una muy alta variación, lo cual es muy interesante para evaluar los resultados.

Los resultados se han medido sobre 100 muestras diferentes por entorno. En las figuras siguientes se pueden observar los resultados en forma de matrices de confusión de cada tipo de extracción de características aplicada a la correspondiente RNA. Al final se ha incluido una gráfica comparativa resumen de la precisión de las tres técnicas en cada entorno y en general. Además se incluyen también los resultados del experimento llevado a cabo con personas.

- ***64 Coeficientes cepstral, 64MFCC***

Para estas características, habiendose validado el conjunto de datos contenido en la matriz ValidaSet con la Red1, se obtiene la matriz de confusión que se muestra a continuación. A la izquierda de la matriz se indican las entradas y arriba se indican las salidas, el número que aparece en cada cruce es el porcentaje de aciertos, es decir, el porcentaje de veces que la red clasifica la entrada de la izquierda como la salida de arriba. Como ya se ha dicho, se han usado

100 muestras diferentes por entorno de 8 entornos para la validación, por tanto 800 muestras en total.

	Apart.	Calle	Estac.	Gim.	Ofic.	Rest.	Super.	Tren
Apart.	68	1	0	0	24	2	2	3
Calle	1	94	0	0	0	4	1	0
Estac.	0	6	25	4	0	43	16	6
Gim.	0	0	48	3	0	1	48	0
Ofic.	3	1	0	1	75	13	4	3
Rest.	0	0	0	1	0	99	0	0
Super.	0	0	2	4	0	5	88	1
Tren	0	26	11	1	4	12	0	46

Tabla 6. Matriz de confusión 64MFCC

La precisión general de este método de características MFCC clasificadas mediante una RNA con una sola capa oculta y 7 neuronas en la capa oculta es de 62,25% de aciertos

Se observa en la matriz de confusión el buen funcionamiento de las características MFCC junto con la clasificación mediante RNA especialmente para los casos de Calle y de Restaurante. Que funcione tan bien en dichos casos puede ser debido a que, especialmente de las señales de audio de dichos entornos, hay mucha información útil que extraer como el ruido de motores, sirenas y gente hablando en el caso de la calle o en el caso del restaurante, el ruido metálico de los cubiertos y de los platos además de un murmullo constante de gente, en definitiva, sonidos muy característicos de este tipo de entornos, para los cuales se consigue muy buena precisión tanto en este experimento como en los restantes.

- ***20 Coeficientes de energía media por banco de frecuencias, 20FBE***

En este caso, como en el anterior, se ha obtenido la siguiente matriz de confusión de la validación de los datos contenidos en CreaSet2 con la Red2.

	Apart.	Calle	Estac.	Gim.	Ofic.	Rest.	Super.	Tren
Apart.	70	1	0	0	23	0	3	3
Calle	1	89	3	5	1	0	1	0
Estac.	0	3	9	27	20	27	4	10
Gim.	0	0	2	81	0	1	16	0
Ofic.	1	0	0	0	90	8	0	1
Rest.	0	0	3	1	4	89	2	1
Super.	1	1	2	2	2	1	89	2
Tren	0	22	9	8	0	0	2	59,2

Tabla 7. Matriz de confusión 20FBE

En este caso, se consigue una precisión general del 72,03%, algo mejor que la obtenida en el caso de los MFCC únicamente. Éstas características de energía se obtienen mediante la técnica de MFCC, pero se refieren a la energía media de la señal por banda de frecuencia de Mel, lo que en un principio era el objetivo del presente trabajo, i.e. extraer la información de la señal contenida en la energía por banda de frecuencias.

Aunque la precisión general no es mucho mejor que en el caso anterior, se puede observar que excepto en el caso de Estación, en el resto funciona de una manera bastante estable y con muy buenos resultados. El hecho de que los resultados no sean óptimos en el ambiente de las estaciones puede deberse a lo comentado en la introducción, la dificultad en la clasificación de sonidos ambientes estriba en la variabilidad de una señal respecto de otra perteneciendo ambas a un mismo entorno. El ejemplo más claro es que una estación de un pueblo pequeño con tan solo dos vías a las 6h de la mañana, no suena igual que una estación principal de una capital en pleno frenesí a las 14h de la tarde. Las señales de una y otra estación no tienen la misma energía en las mismas bandas de frecuencia y por tanto, aunque se ha intentado cubrir un espectro de señales bastante amplio, no funciona bien para este escenario, al igual que el resto de experimentos.

- ***64 Coeficientes cepstral concatenados con 16 coeficientes Matching Pursuit, 64MFCC+16MP***

Tras validar la matriz ValidaSet3, en la que cada columna corresponde al vector de características de este tipo de cada fragmento de 10 segundos de señal que se ha usado para validación de la Red3, se obtiene la siguiente matriz de confusión.

	Apart.	Calle	Estac.	Gim.	Ofic.	Rest.	Super.	Tren
Apart.	14	1	0	14	67	1	1	2
Calle	4	79	5	2	7	1	1	1
Estac.	1	0	19	10	13	38	9	10
Gim.	1	16	54	23	1	2	3	0
Ofic.	15	1	0	1	71	7	3	2
Rest.	0	0	5	6	1	85	2	1
Super.	0	8	3	2	2	3	81	1
Tren	0	18	10	0	1	10	0	61

Tabla 8. Matriz de confusión 64MFCC+16MP

En este tercer experimento numérico se ha concatenado las características extraídas mediante la técnica MFCC y las extraídas mediante MP. La idea ha surgido de los antecedentes, donde se comprueba que la concatenación de ambas características mejora la precisión de cada una de ellas por separado, la diferencia del experimento leído en los antecedentes difiere del actual en que en este caso dichas características se han aplicado a la clasificación mediante RNA.

La precisión general en este caso es del 54,125% algo peor que los dos casos anteriores, aunque siempre hay que tener en cuenta que se está tratando con la precisión general. Si se fija un poco más en detalle, por entornos, aunque un poco peor que en el caso anterior, presenta una precisión más o menos aceptable para los ambientes de Calle, Restaurante y Supermercado, entornos en los cuales ya se ha comentado que las señales contienen una gran cantidad de información útil.

En los tres casos se han usado 100 fragmentos de señal de audio de 10 segundos cada uno por entorno, extraído las correspondientes características y validado con cada una de las RNA creadas para cada tipo de extracción.

Respecto a los resultados del experimento llevado a cabo en personas, se ha creado la siguiente matriz de confusión en la que los resultados se muestran en tanto por ciento, por ello hay que tener en cuenta que al contrario que en la validación de las redes neuronales, en este experimento se han usado únicamente 40 fragmentos de señales frente a los 800 usados en el resto de experimentos por tanto para dar los resultados en tanto por ciento se han tenido que extrapolar.

	Apart.	Calle	Estac.	Gim.	Ofic.	Rest.	Super.	Tren
Apart.	75	0	0	0	25	0	0	0
Calle	0	83,3	0	0	0	0	0	16,7
Estac.	0	15,8	52,7	0	0	15,8	0	15,8
Gim.	2,5	0	20	55,5	0	0	22	0
Ofic.	13,2	0	0	10	66,6	0	0	10,2
Rest.	0	0	0	10	0	77,7	12,3	0
Super.	0	10,2	0	0	0	16,5	73,3	0
Tren	0	0	10,2	10,2	18,5	0	0	61,1

Tabla 9. Matriz de confusión del test en personas

La precisión general obtenida en este experimento ha sido de un 65,5% de aciertos, precisión que se puede comparar con el resto de experimentos en la tabla resumen que se muestra a continuación.

	64MFCC	20FBE	64MFCC+16MP	Personas
Apartamento	68	70	14	75
Calle	94	89	79	83,3
Estación	25	9	19	52,7
Gimnasio	3	81	23	55,5
Oficina	75	90	71	66,6
Restaurante	99	89	85	77,7
Supermercado	88	89	81	73,3
Tren	46	59,2	61	61,1
General	62,25	72,03	54,125	68,15

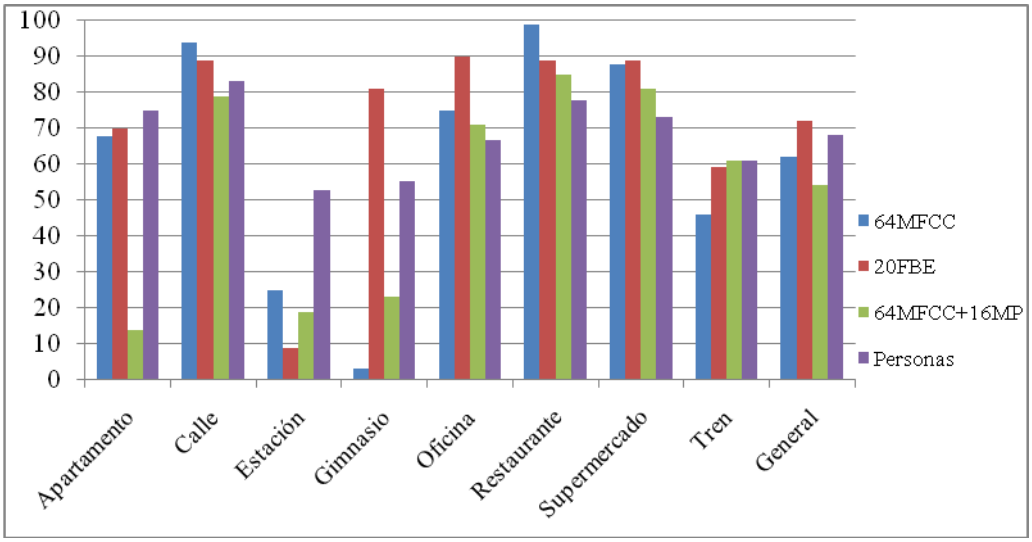


Tabla 10. Comparativa de resultados de los 4 experimentos

A la vista de las figuras mostradas, pudiendo observar en una misma tabla y gráfica los pertenecientes a todos los experimentos llevados a cabo, se obtienen unos resultados similares a los observados en los antecedentes, es decir, aunque la precisión es mejor en los experimentos del presente proyecto, lo que se puede achacar al uso de las RNA así como a que en los trabajos revisados se intenta clasificar un mayor número de entornos (a mayor número de entornos a clasificar peor es la precisión), en general se consiguen mejores resultados en los entornos en los que estos experimentos son comparables con los antecedentes.

Cabe destacar, las características de energía por banda de frecuencias en la escala de Mel, que aplicadas a RNA funcionan de una manera bastante buena en general y estable por otro lado, ya que la tasa de aciertos entre entornos es muy similar entre unos y otros.

Al compararlos con el experimento llevado a cabo en las personas, destaca la técnica de extracción de coeficientes cepstral de la que se extraen las energías medias por banda de frecuencias en la escala de Mel, características que dan muy buena precisión tanto por entornos, excepto en el caso de la estación, como general, superando incluso la tasa de aciertos observada en las personas.

Capítulo 6
Conclusiones

Tras describir los experimentos llevados a cabo y los resultados obtenidos de dichos experimentos, en este último capítulo se pretende hacer una valoración personal comentando el desarrollo del proyecto, dificultades encontradas y la resolución de las mismas. Así mismo, sentar las bases para futuros trabajos en este área con el fin de mejorar la calidad de los dispositivos que desde hace algún tiempo y cada vez más forman parte de nuestra vida diaria y por extensión nuestra calidad de vida.

Este proyecto presenta los resultados de un estudio comparativo de diversas técnicas de extracción de características de sonidos ambientales, por tanto no estructurados, aplicadas a la clasificación mediante RNA. Ahí estriba el primer y principal problema del proyecto, es decir, las técnicas de extracción de características de sonidos no estructurados ya que en el lenguaje común, se trata con ruido y se intenta extraer información de dicho ruido.

Como ya se ha expuesto a lo largo de la memoria, la tecnología de audio usada ha sido una grabadora de un Smartphone y una aplicación Freeware que supera en prestaciones al dispositivo, consiguiendo capturas ricas en información.

Como ya se ha expuesto a lo largo de la memoria, se ha recopilado una gran base de datos sonora de diferentes escenarios. Además, se ha utilizado como dispositivo grabador un teléfono móvil (Smartphone) para conseguir que dicha grabación sea lo más realista posible. Para ello, se usó una aplicación Freeware de grabación mucho más flexible y con más prestaciones que la que viene por defecto en el Sistema Operativo.

En el presente trabajo se trata sobre todo la clasificación mediante RNA lo que ha constituido una ventaja en los resultados ya que son una herramienta muy potente en este campo además de en muchos otros y se puede comprobar, que para un mismo entorno, se consigue mayor precisión al usar RNA con la misma técnica de extracción de características que en otros experimentos que no usan RNA, como por ejemplo, algunos de los expuestos en

el capítulo de antecedentes. Se puede deducir de lo anterior que el uso de RNA constituye un gran avance en la clasificación de señales.

Como meta, se ha intentado caracterizar este tipo de señales no estructuradas, en el dominio de la frecuencia encontrando una representación de energía ya que es lógico pensar a la vista del espectro que cada señal tiene unas características de energía en unas bandas determinadas pudiendo encontrar un punto común entre señales pertenecientes al mismo entorno. Esto ha constituido una de las dificultades ocurridas durante el desarrollo del proyecto. Al profundizar en la materia se ha podido comprobar que dicha lógica no es del todo cierta ya que el sonido ambiental de un mismo entorno en momentos o escenarios diferentes puede cambiar sobremanera, como es una estación de tren pequeña en un pueblo a las 6h de la mañana frente a una estación principal en una capital en pleno frenesí a las 14h, debiendo ser clasificadas ambas de estación, por volver al mismo ejemplo usado anteriormente. Aun así, se ha conseguido extraer un conjunto de características de energía de la señal que ofrecen buenos resultados y por tanto se puede pensar en las posibilidades de este tipo de técnica para un estudio futuro.

En el experimento llevado a cabo con personas se pone de manifiesto la necesidad de la mente humana de apoyarse en la imagen para el reconocimiento del entorno, ya que la precisión ocurrida en el experimento con personas no puede clasificarse de perfecta. Si se comparan los resultados de dicho experimento con los obtenidos artificialmente, mediante la extracción de características y la clasificación mediante RNA, queda demostrado el buen funcionamiento de las técnicas estudiadas ya que los resultados son muy próximos entre sí..

Por tanto, creo que en futuras investigaciones, habiendo demostrado el buen funcionamiento de las RNA, sería interesante encontrar un conjunto de características que defina de una forma óptima las señales no estructuradas como son las señales ambiente y de esta manera poder comprobar la precisión real en la aplicación de las RNA al entorno.

Investigando en la línea de la energía de la señal por banda de frecuencias, encontrar tal vez un estadístico de la señal que la defina completamente, mejorando los resultados ocurridos en el caso de la Estación, por ejemplo.

El punto final a este estudio sería pensar en la implantación de un sistema de reconocimiento de entornos acústicos en los dispositivos móviles.

Anexo A

Presentación

A continuación las transparencias de la presentación.

RECONOCIMIENTO DE ENTORNOS ACÚSTICOS PARA APLICACIONES MÓVILES

Autor: Adriana Hernández Pérez
Director: Jose Javier López Monfort

Reconocimiento de entornos

ÍNDICE

- Motivación
- Objetivos
- Antecedentes
- Definición del problema
- Extracción de características
- Clasificación mediante RNA
- Resultados
- Conclusiones y líneas futuras

Motivación

Motivación

- Diversas aplicaciones que mejoran nuestra calidad de vida
  - Sistemas de navegación
  - Vigilancia
  - **Dispositivos móviles**
- Nuestra actividad diaria se desarrolla en diversos ambientes acústicamente clasificables

Objetivos

Objetivos

- Encontrar un conjunto de **características óptimas** que definan la señal
- **Mejorar** la precisión de las **técnicas existentes**
- Aplicar el uso de **RNA** al reconocimiento
- **Mejorar la calidad** de los dispositivos móviles de uso diario

Antecedentes

- **Sonidos ambiente**
  - Lo que oimos constantemente (calles, restaurantes, estaciones de tren/metro,...)
  - Señales **no estructuradas**
    - Habla
    - Música
  - Dificultad en el reconocimiento del entorno
    - Lugares diferentes que suenan parecido
    - Sonidos diferentes que suenan parecido

Antecedentes

ANTECEDENTES (artículos consultados)	ENTORNOS	PRECISIÓN	
Classifying user environment	Apartamento	MFCC+ANN	87,7%
	Oficina	MFCC+ANN	93,9%
	Exterior	MFCC+ANN	78%
	Restaurante	MFCC+ANN	43,5%
Environmental sound recognition	Restaurante	MP	
		35%	
	Calle	MP	
		50%	
Comparison of techniques	Sonidos definidos (llaves, cristal roto...)	MFCC+ANN	4%

Definición del Problema

- Comparación de tres técnicas de extracción de características
- Clasificación mediante RNA
- Validación de los resultados en ocho entornos diferentes

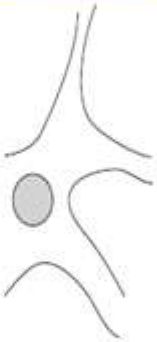
- Apartamento
- Calle
- Estación
- Gimnasio
- Oficina
- Restaurante
- Supermercado
- Tren

Definición del Problema

- Las técnicas típicas de extracción de características no funcionan bien para sonidos ambientales
- La técnica de extracción de características más usada es **MFCC**
  - MFCC → Sonidos estructurados (música, voz)
- Otras técnicas usadas
  - MP → Sonidos no estructurados

Definición del Problema

- Uso de **RNA**
  - Implementación **sencilla** de un comportamiento local observado en nuestros cerebros
  - El cerebro está compuesto de **neuronas** → Elementos individuales de procesamiento
  - Basandose en los eventos de los conectores neuronales la red se comporta de forma diferente



Características de las Señales

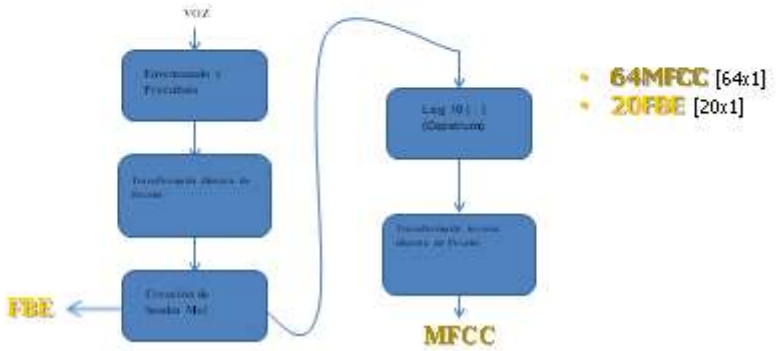
- **Grabación de sonidos ambientales**
  - Grabadora del smartphone + aplicación gratis de grabación (WAV/PCM 44,1kHz 16bit)
  - Señales de audio de **8 entornos**, **diferentes** escenas/entorno, muchas grabaciones por escena → Amplia variedad de señales para cubrir un espectro aplio.
  - Señales de **10 segundos** (Pruebas en personas)

Características de las Señales

- Extracción de 3 **tipos de características** distintas mediante 2 técnicas de extracción
  - 64MFCC → Mel Frequency Cepstral Coefficients
  - 20FBE → Mel Frequency Cepstral Coefficients
  - 64MFCC + 16MP → MFCC + Matching Pursuit

Características de las Señales

- Mel Frequency Cepstral Coefficients, **MFCC**
  - Funcionan muy bien para sonidos estructurados

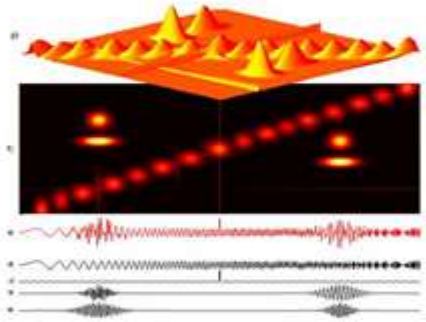


Características de las Señales

Matching Pursuit, MP

- Técnica de selección adaptable.
- Representación de la señal como suma ponderada de elementos de un diccionario
  - Diccionario tiempo-frecuencia de Gabor (16 bases)
- Concatenación

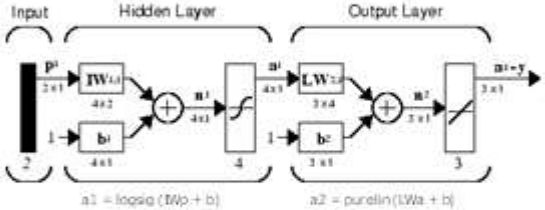
  64MFCC + 16MP [80x1]



Clasificación mediante RNA

- 64MFCC [64x1]
- 20FBE [20x1]
- 64MFCC+16MP [80x1]

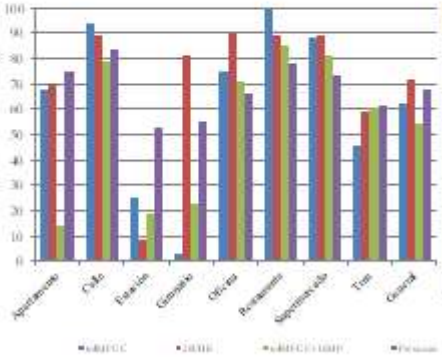
- CreaSet [nx8] + CreaT[8x8]
- TrainSet [nx2392] + TrainT [8x2392]
- ValidSet [nx800]+ ValidT[8x800]



Apt.	Calle.	Est.	Com.	Ofi.	Redt.	Super.	Trast
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1

Resultados

Comparación de resultados



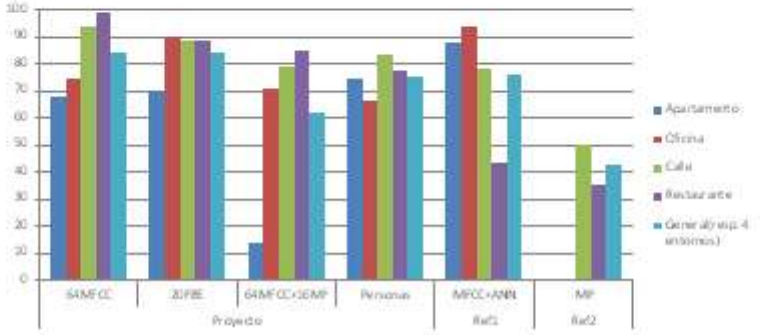
	64MFCC	20FBE	64MFCC+64MP	Personas
Apartamento	68	70	14	75
Calle	94	89	79	83,3
Estación	25	9	19	52,7
Gimnasio	3	81	23	55,5
Oficina	75	90	71	66,6
Restaurante	99	89	85	77,7
Supermercado	88	89	81	73,3
Tren	46	59,2	61	61,1
General	62,25	72,03	54,125	68,15

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Resultados

Resultados generales respecto a los antecedentes

- La combinación de dos técnicas diferentes de extracción de características usando como clasificador RNA
- Uso de los coeficientes de energía media en la banda de Mel como características de la señal



Conclusiones y líneas futuras

- Evaluación de las técnicas de extracción usando **RNA como clasificador**
- Uso de **RNA** para reconocimiento de entornos
- A la vista de los resultados, queda de manifiesto la dificultad del problema de clasificación ambiental
  - Mismo experimento → Resultados diferentes

Conclusiones y líneas futuras

- Se ha demostrado que las características obtenidas de las energías por banda de frecuencia de Mel consiguen buenos resultados
- Respecto a MFCC y la combinación MFCC+MP, se puede decir que donde MFCC falla MP lo complementa
- Los **resultados** generales de las 3 técnicas respecto al experimento en personas son **buenos**, existiendo una tendencia similar también por entornos

Conclusiones y líneas futuras

Lineas futuras

- Seguir investigando en la línea de la energía de la señal por banda de frecuencia
- Encontrar el conjunto de características óptimas que definan perfectamente a la señal ambiente que se intenta reconocer
- La meta es la implantación de este tipo de sistemas en los dispositivos móviles de modo que se mejore la calidad del servicio

FIN

Adriana Hernández Pérez
adherpe@gmail.com

Bibliografía

[1] Robert G.Malkin and Alex Waibel. ***Classifying user environment for mobile applications using linear autoencoding of ambient audio***.

[2] Selina Chu, Shrikanth Narayanan and C.-C. Jay Kuo. Signal and Image Processing Intitute and Viterbi School of Engineering. University of Southern California, Los Angeles. ***Environmental sound recognition using MP-Based features***.

[3] Selina Chu (Student Member, IEEE), Shrikanth Narayanan (Fellow, IEEE), and C.-C. Jay Kuo (Fellow, IEEE). ***Environmental sound recognition with Time-Frecuency audio features***. IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 17, NO. 6, AUGUST 2009

[4] Michael Cowling and Renate Sitte. School of Information Technology, Griffith University. Australia. ***Comparison of Techniques for environmental sound recognition***. May 2003.

[5] Daniel P.W.Ellis and Keansub Lee. LabROSA, Dept. of Elec. Eng. Columbia University. New York. ***Minimal Impact Audio-Based Personal Archives***.

[6] Pedro Vera Candeas. Dep. de teoría de la señal y comunicaciones. Escuela Politécnica. Universidad de Alcalá. Tesis doctoral. ***Desarrollo de técnicas de codificación de audio basadas en modelos de señal paramétricos***. 2006

[7] Juan Luis Navarro Mesa. Universidad de Las Palmas de Gran Canaria. Dep. de Señales y Comunicaciones. Tratamiento de la señal de audio. ***Procesador Acústico: El bloque de extracción de características.***

[8] Universidad de Zaragoza, 2010. ***Aplicación del algoritmo MP para la estimación del ritmo cardiaco en señales obtenidas mediante un sensor capacitivo.***

[9] Alěs Procházka and Jaomír Kukal. Institute of Chemical Technology. Dep. od Computing and Control Engineering. Prague. ***Wavelet Transform use for Feature Extraction ans EEG Signal segments classification.*** http://dsp.vscht.cz

[10] Mathworks Documentation Center. Matlab Functions in Auditory Toolbox by Malcolm Slaney (c) 1998 Interval Research Corporation

[11] Mathworks Documentation Center. Matlab Functions in Neural Network Toolbox. www.mathworks.es/es/help/nnet/functionlist.html

[12] Matlab Central. Matlab Newsreader. ***Neural Networks Configuration***

[13] Stackoverflow. ***How to feed with two or more inputs a Neural Network in MatLab.*** www.stackoverflow.com/questions

[14] Dr. Sergio Ledesma. Facultad de Ingeniería. Universidad de Guajamato. ***Las Redes Neuronales. Implementación y consideraciones prácticas.***

[15] Primoz Potocnik. University of Ljubljana. Faculty of Mechanical Engineering. LASIN-Laboratory of Synergetics. ***Neural Networks: Matlab examples. Neural Networks course (practical examples)***. 2012

[16] ***Using Neural Networks for Pattern Classification Problems***. www.csun.edu/skatz/nn_proj/percept_intro_nnproj_bw.pdf

[17] Vincent Cheung and Kevin Cannons. Signal & Data Compression Laboratory. University of Manitoba, Canada. ***An introduction to Neural Networks***

[18] Gonzalo Fernández de Córdoba Martos. Salamanca, Septiembre de 2007. ***Creación de Interfaces Gráficas de Usuario (GUI) con MatLab***.

[19] José Francisco Ecribano Molina. Universidad Carlos III. Ingeniería Industrial. Madrid, Octubre 2009. ***Desarrollo de una interfaz gráfica en MatLab para la aplicación de modelos de regresión local polinómica.***