Departamento de Teoría de la señal y comunicación

PROYECTO FIN DE CARRERA

RECONOCEDOR DE HABLA BASADO EN LA EXTRACCIÓN DE CARACTERÍSTICAS ARTICULATORIAS

Autor: Isabel Horrillo Peña
Tutor: Carmen Peláez Moreno

Leganés, abril de 2015

Título: Reconocedor de habla basado en la extracción de características articulatorias
Autor: Isabel Horrillo Peña
Director: Carmen Peláez Moreno

EL TRIBUNAL

Presidente: Fernando Díaz de María

Vocal: Mª Jesús Poza Lara

Secretaria: Ascensión Gallardo Antolín

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 29 de abril de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

En primer lugar, quiero dar las gracias a mi tutora, Carmen, por la entrega, ayuda y paciencia durante todo el PFC.

A Pátroci, por haber estado ahí todas y cada una de las veces que la he necesitado, no tengo palabras...

A Marc, por confiar en mi mucho más que yo y por ser esa lluvia de la que hablaba Cortázar...

Por supuesto a mis padres, porque es mucho lo que me enseñan cada día y porque son para mí un ejemplo a seguir en todos los sentidos. Gracias por haberme apoyado siempre, desde el principio hasta el final. También a mis hermanos, mis dos “seres” especiales.

Sin olvidar a mi *belle-famille* y a mis amiguis, de aquí y de allá, por haberme animado tanto.

Muchas gracias a todos, es un regalo teneros cerca.

Resumen

Los sistemas de reconocimiento automático de habla persiguen proporcionar un interfaz natural entre máquinas y humanos mediante el uso de la voz. En muchos casos, se adopta la estrategia de imitar en la medida de lo posible los mecanismos de comunicación entre humanos. La implementación del sistema es, pues, muy importante y debe tener en cuenta los diversos problemas a los que se enfrenta, como el ruido aditivo o la variabilidad del hablante.

El trabajo realizado en este PFC tiene como objetivo ensayar nuevas técnicas de extracción de características haciendo uso de información articulatoria, para averiguar si el sistema resultante tiene mejores prestaciones. Para llevar a cabo dicha tarea, utilizaremos la extracción de las características articulatorias de la voz, utilizando como clasificador un modelo híbrido con redes neuronales (perceptrones multicapa).

Para la extracción de las características se crearon 7 clasificadores (a los que luego se añadió un octavo) para cada uno de los 7 niveles articulatorios que definimos, donde cada uno de ellos tomará, a su vez, diferentes valores atendiendo a la naturaleza del sonido emitido. Se consideraron además las diferencias que existen entre un entorno ideal y uno real (añadiendo ruido aditivo), para evaluar la pérdida de prestaciones existente.

Los resultados obtenidos no sólo nos dan una visión general del sistema en cuanto al rendimiento global del mismo, sino que también nos muestran qué características de la voz son más robustas frente a alteraciones procedentes del ruido ambiente.

Abstract

The systems of automatic speech recognition aim to provide a natural interface between machines and human beings by the use of the voice. The strategy of imitating the mechanisms of communication between human beings is adopted -as far as possible- in many cases. The implementation of the system is very important and has to take into account the different problems that it faces, like ear noise or the variation of the speaker's voice.

The work carried out on this Final Year Project aims to test new feature extraction techniques by using articulatory information, and so resolves if the resulting system has the best performance. To do this, we will extract the articulatory characteristics of the voice, using, as a sorter, a hybrid model with neuronal networks (multilayer perceptrons).

For the extraction of the characteristics, 7 classifiers were created (then an eight one was added) for each of the 7 articulatory levels defined. Each of them will take different values relating to the nature of the sound issued. Also, the difference between an ideal surrounding and a real one (added noise) will be studied, in order to evaluate the losses of the existing benefits.

The results obtained will not only give us a general vision of the system's overall performance, but it will also show us which characteristics of the voice are more robust against changes in the transmission channel.

Índice general

Agradecimientos.....
Resumen.....
Abstract.....
Índice general.....
Índice de figuras.....
Índice de tablas.....

Capítulo 1. Introducción y objetivos.....1
1.1 Introducción.....1
1.2 Objetivos 2
1.3 Fases del desarrollo.....3
1.4 Estructura de la memoria.....3

Capítulo 2. Introducción al reconocimiento automático de habla.....5
2.1 Definición de un sistema ASR.....6

Capítulo 3. Extracción de características.....9
3.1 MFCC:Mel Frecuency Cepstral Coefficients.....10
3.2 PLP: Perceptual Linear Prediction.....12

Capítulo 4. Características articulatorias.....15
4.1 Fonética articulatoria 16
4.2 Niveles articulatorios.....18
4.3 Problemática del modelo.....25

Capítulo 5. Un Reconocedor Automático de Habla basado en la extracción de características articulatorias. Enfoque híbrido.....31
5.1 Modelos ocultos de Markov (HMM's).....31
5.2 Redes Neuronales Artificiales (ANN's).....33
5.3 Sistema de reconocimiento híbrido HMM/ANN.....35
5.4 Diseño de nuestro Reconocedor.....36

Capítulo 6. Pruebas y resultados.....40
6.1 Marco Experimental.....40
6.2 Base de datos 41
6.3 Herramientas para el análisis de resultados.....41
6.4 Pruebas y experimentos.....44
6.5 Resultados y análisis.....51

Capítulo 7. Conclusiones finales y líneas futuras.....67

Capítulo 8. Presupuesto.....69

ANEXO.....72
1). Sistema de pruebas ISOLET Testbed.....72
2). Script 1stream_concatenacion.csh.....74
3.) Etiquetas.m.....79
4.) Valores promedio de los experimentos.....81

GLOSARIO DE ACRÓNIMOS.....87

BIBLIOGRAFÍA Y REFERENCIAS.....89

Índice de figuras

Figura 1: Representación temporal de una señal de voz.....5
Figura 2: Esquema básico de un sistema ASR.....6
Figura 3: Etapas de un sistema de reconocimiento de la señal de voz [40].....8
Figura 4: Ilustración de la extracción de las características de una señal.....10
Figura 5: Esquema básico de la extracción de coeficientes MFCC [42].....11
Figura 6: Esquema básico de la extracción de coeficientes PLP [3].....12
Figura 7: Comparativa de la precisión de un sistema ASR para locutores independientes utilizando PLP de orden 5 y LPC de orden 14, [3].....14
Figura 8: Órganos y zonas del aparato fonador.....16
Figura 9: Órganos fonadores en las cavidades supraglóticas [25].....18
Figura 10: Niveles de AF's.....19
Figura 11: Modelo simplificado de una señal de habla con ruido aditivo.....25
Figura 12: Las líneas horizontales representan las posiciones de destino ideales para 1 articulador simple y 2 fonemas adyacentes. PB1, PB2 y PB3 son los límites ideales para el Fonema 1 y el Fonema 2.....26
Figura 13: Transición articulatoria ideal desde la posición destino del fonema 1 (T1) hasta la posición destino del fonema 2 (T2). PB representa la posición aproximada del límite entre fonemas.....26
Figura 14: 2 articuladores con distintos grados de inercia recorrerán la misma distancia en tiempos diferentes. El articulador A1 tiene menos inercia y recorrerá la distancia entre los límites de los fonemas T1 y T2 más rápidamente que el articulador A2.....27
Figura 15: Para el mismo tiempo el articulador A1, que tiene una menor inercia, se acerca más a la posición de destino del fonema T2.....27
Figura 16: IT se refiere a la posición de destino ideal, pero observamos cómo el articulador sólo alcanza la posición UT, que no se corresponde con la deseada.....27
Figura 17: A1, A2 y A3 representan a 3 articuladores solapándose entre los límites T1, T2 y T3 de 2 fonemas adyacentes.....28
Figura 18: Ritmo relativo en la producción articulatoria de la palabra "pan" en inglés...29

Figura 19: Ejemplo de HMM de 3 estados, siendo qi conjunto de estados, p(xn|qi) la densidad de emisión de probabilidad asociada a cada estado y p(qj|qi) la transición de probabilidad del estado qi al estado qj [7].....32
Figura 20: Estructura básica de una Red Neuronal [31].....33
Figura 21: Arquitectura de una red neuronal artificial (perceptrón de tres capas). Los pesos son los parámetros que deben ajustarse durante el proceso de entrenamiento para conseguir que la capa de salida de resultados coincida con los observados. El error entre lo observado y el resultado de la red se propaga hacia atrás (backpropagation) y debe ir disminuyendo en las sucesivas iteraciones en las que se presentan los valores de las variables predictoras. La complejidad de una red depende del número de nodos de su capa oculta [13].....34
Figura 22: Modelos de palabras creados por la sucesión de fonemas para el reconocimiento continuo de habla.....37
Figura 23: Sistema de 2 redes neuronales en serie a las que denominaremos Fase AF y Fase Conc, respectivamente.....39
Figura 24: Ejemplo de matriz de confusión. La diagonal en tonos más oscuros significa que el clasificador actúa con bastante precisión [39].....42
Figura 25: Diagramas de información extendida de la entropía relativa en una distribución bivariante, siendo a) diagrama convencional y b) diagrama representando las entropías por separado [38].....42
Figura 26: Esquema de las zonas interpretables de un triángulo entrópico [39].....43
Figura 27: Esquema del diseño del reconocedor.....46
Figura 28: Esquema del sistema de redes creado para nuestro ASR. WTS denota los parámetros de las redes neuronales (WeighTS o pesos) y ACT (activations) las salidas blandas de la redes neuronales o estimaciones de las probabilidades a posteriori de las clases de salida, dadas las entradas. NN son las redes neuronales del sistema de referencia cuyas entradas son los PLP y las salidas, las clases fonéticas. NN_AF son las redes neuronales cuyas entradas son PLP y las salidas las clases articulatorias y NN_ACT las que toman como entrada la concatenación de las activaciones de las NN_AF y devuelven las probabilidades a posteriori de las clases fonéticas.....48
Figura 29: Manner.....52
Figura 30: Place.....52
Figura 31: Voice.....53
Figura 32: High-low.....53
Figura 33: Front-back.....53
Figura 34: Round.....53
Figura 35: Static.....54
Figura 36: Nasality.....54
Figura 37: Manner.....55
Figura 38: Place.....55
Figura 39: Voice.....55
Figura 40: High-low.....55
Figura 41: Front-back.....56
Figura 42: Round.....56
Figura 43: Nasality.....56
Figura 44: Static.....56
Figura 45: Exp.2 (nhidden=100). Salidas del reconocedor.....59

Figura 46: Exp.1 (nhidden=800). Salidas del reconocedor.....59
Figura 47: Exp.1 (nhidden=800). Precisión.....60
Figura 48: Exp.2 (nhidden=100). Precisión.....60
Figura 49: Exp.1(nhidden=800).....60
Figura 50: Exp.2(nhidden=100).....60
Figura 51: Exp.2 (nhidden=100). Mapa de calor.....61
Figura 52: Exp.1 (nhidden=800). Mapa de calor.....61
Figura 53: Exp.4 (cw=3). Salidas del reconocedor.....62
Figura 54: Exp.5 (cw=7). Salidas del reconocedor. 62
Figura 55: Exp.5 (cw=7). Precisión.....62
Figura 56: Exp.4 (cw=3). Precisión.....62
Figura 57: Exp.6. Mapa de calor.....63
Figura 58: Exp.9. Mapa de calor.....63
Figura 59: NN "Nasality" activando el parámetro "Reject_last".....63
Figura 60: Nueva NN "Nasality", aislada de "Manner".....63
Figura 61: Exp.1(7NN's).Precisión.....64
Figura 62: Exp.11(8NN's con reject_last).Precisión.....64
Figura 63: Exp.1(7NN's).Fallo en la nasalidad.....64
Figura 64: Exp.11(8NN's).Fallo en la nasalidad.....64
Figura 65: Exp.7. Matriz de confusión.....65
Figura 66: Exp.7. Detalle del conjunto "e-set".....65
Figura 67: Experimento 16.....66

Índice de tablas

Tabla 1: Tabla de valores de AF's.....19
Tabla 2: Ejemplos de valores de tasa de detección correcta (corr) y precisión (acc) en un sistema AT CRF para los casos ideal y real.....28
Tabla 3: Distancias en frames entre los límites AF-fonema de algunos sonidos comparando un detector que utiliza MFCC y otro que utiliza información a largo plazo ("long term").....30
Tabla 4: Mapeo fonema-valor articulatorio [21].....38
Tabla 5: Tipo de entrenamiento y test para el experimento de referencia.....49
Tabla 6: Tipo de entrenamiento y test para cada salida del Reconocedor.....49
Tabla 7: Las 8 redes neuronales y sus clases de salida.....52
Tabla 8: Valores promedio de precisión en entrenamiento (train) y validación (cv) para los casos limpio y ruidoso. Se ha desechado el conjunto 1 (fold 1) de la validación cruzada para realizar los cálculos porque es el utilizado para ajustar algunos parámetros del modelo (tuning).....57
Tabla 9: Tiempo empleado en la realización de cada fase.....70
Tabla 10: Gastos en material.....70
Tabla 11: Gastos de personal.....71
Tabla 12: Costes directos del proyecto.....71

Capítulo 1. Introducción y objetivos

1.1 Introducción

El habla es el medio natural de comunicación entre personas, sin embargo, hasta el presente se puede afirmar que en su comunicación con las máquinas el hombre ha hecho uso exclusivo del lenguaje escrito. Resulta natural, por tanto, extender la capacidad de comunicación hombre-máquina al mensaje oral.

El reconocimiento automático de habla (en inglés, Automatic Speech Recognition -ASR-)[1] consiste, básicamente, en la transformación automática de habla en texto. En dicha tarea, existe una diferencia básica entre los humanos y los ordenadores. Se trata de la representación de la entrada que recibe cada uno; los humanos aprovechamos el conocimiento del contexto y somos capaces de utilizar toda la información que está presente en la señal acústica, mientras que los sistemas ASR sólo pueden utilizar la información codificada en las características acústicas. Idealmente, éstas poseen toda la información relevante para el reconocimiento pero, en realidad, éste no es siempre el caso. El conocimiento acerca de qué información de la señal de habla es detectada de forma más robusta por los oyentes, puede mejorar el rendimiento de la máquina de reconocimiento.

El reconocimiento del habla es aplicable a una gran variedad de situaciones donde se requiera una comunicación hombre-máquina. Algunas de las aplicaciones más comunes hoy en día serían por ejemplo el dictado automático (muy utilizado sobre todo en medicina y en el dictado de textos legales), el control por comandos, telefonía, sistemas portátiles o sistemas para personas discapacitadas.

1 Adoptamos la postura de reflejar los acrónimos en su nomenclatura inglesa puesto que es, en general, mucho más extendida

1.2 Objetivos

El objetivo del trabajo de investigación propuesto en este PFC será evaluar si representaciones alternativas basadas en la extracción de características articulatorias (que explicaremos más adelante) pueden contribuir a mejorar las prestaciones de un reconocedor automático de voz. Se pondrá especial énfasis en la creación de un sistema robusto, que al cambiar de entorno (del entorno del laboratorio al de la vida real), no sufra una pérdida llamativa de prestaciones. Es decir, que supere uno de los mayores problemas que existen hoy por hoy en el reconocimiento del habla, el problema de "mismatch"[26], provocado por distorsiones acústicas.

La robustez de los sistemas de reconocimiento de habla es, pues, la dirección más relevante en el desarrollo de sistemas ASR.

Los enfoques prevalecientes hasta ahora, están típicamente basados en la idea de que una palabra puede representarse como una secuencia individual de estados fonéticos. Sin embargo, la producción de una palabra implica el movimiento simultáneo de varias articulaciones como los labios o la lengua, que se mueven asíncronamente y que no siempre llegan a sus posiciones de destino. Al hablar no se hacen pausas entre los fonemas, pero los humanos los reconocemos sin dificultad, separándolos sin problemas dentro de cada palabra. La longitud de las palabras y la entonación son muy importantes a la hora de ayudarnos con todo esto. Necesitamos, por tanto, sistemas robustos que puedan afrontar estas dificultades.

Surge entonces la idea en la que nos basamos en este pfc, de representar palabras utilizando flujos múltiples de características fonéticas que caracterizan los aspectos más esenciales de las propiedades articulatorias de los sonidos del habla (sonoridad, nasalidad, punto y modo de articulación etc.). Son las **características articulatorias,** a las que a partir de ahora denominaremos **AF's** (-Articulatory Features-).

La idea subyacente es permitir asincronía en la decodificación de las distintas características articulatorias que sabemos que existe, debido al fenómeno de la coarticulación. De esta manera, las fronteras entre dos fonemas contiguos pueden darse en distintos instantes en los distintos flujos de características, resultando en un sistema más flexible y bajo nuestro punto de vista, más cercano a la realidad. Para la implemantación de este sistema hemos optado por hacer usos de los denominados híbridos entre redes neuronales (ANN –Artificial Neural Networks-) y modelos ocultos de Markov (HMM –Hidden Markov Models-) por brindarnos una capacidad de expresión más amplia y una arquitectura más flexible.

1.3 Fases del desarrollo

Las fases que se han seguido en la creación del proyecto han sido las siguientes:

- En primer lugar se ha llevado a cabo una fase previa de documentación para conocer el estado del arte en el ámbito de los reconocedores de habla. Diversos artículos y bibliografía han sido clave para conocer los estudios e investigaciones realizadas sobre el tema.

- A partir de este punto, se ha comenzado a trabajar con una base de datos con 7800 muestras vocales, emitidas por 150 hablantes. Tras la obtención de los vectores de parámetros y extracción de las características acústicas, se ha llevado a cabo el reconocimiento propiamente dicho para caracterizar el sistema de referencia.

- Se ha realizado el entrenamiento de un sistema híbrido de redes neuronales y modelos ocultos de markov, que hemos utilizado para crear nuestro sistema, ayudándonos de las herramientas SPRACH_Core y Quicknet. En él, se ha asociado a cada vector de características articulatorias, una unidad con significado lingüístico (28 fonemas en nuestro caso) y posteriormente se ha realizado un alineamiento de Viterbi.

- Se ha realizado el test para conocer en rendimiento final de nuestro reconocedor.

- Con la herramienta Matlab se han generado diferentes gráficas para poder visualizar los resultados de una manera sencilla.

- Por último, se ha llevado a cabo una etapa de experimentación, modificando el sistema mediante la variación de parámetros, para comprobar la reacción del mismo ante determinados cambios.

1.4 Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo.

Capítulo 1: Se presentan las motivaciones para las realización del PFC, así como los objetivos que se persiguen y las fases que se han seguido para su desarrollo.

Capítulo 2: Recoge una visión general sobre los sistemas de Reconocimiento Automático de Habla, se realiza una descripción de los mismos y de su funcionamiento.

Capítulo 3: Este capítulo se centra en la forma y en los métodos utilizados para llevar a cabo una de las partes más importantes dentro del sistema, la extracción de características.

Capítulo 4: Este apartado detalla en profundidad el método elegido para la implementación de nuestro reconocedor, las características articulatorias. Se realiza una descripción de la fonética articulatoria, los niveles articulatorios que utilizaremos en el sistema así como la problemática del modelo y sus posibles soluciones o mejoras.

Capítulo 5: Se desarrolla el modelo híbrido en el que nos basamos para la implementación del reconocedor, el diseño de nuestro sistema, partiendo de la explicación de los Modelos Ocultos de Markov y las Redes Neuronales Artificiales.

Capítulo 6: El capítulo comienza con un resumen del marco experimental en el que hemos trabajado, la descripción de la base de datos y las herramientas utilizadas para ayudarnos con el análisis de resultados.

Posteriormente se detallan las pruebas realizadas y los resultados obtenidos con los diferentes experimentos, comparándolos entre ellos.

Para finalizar con el capítulo se realiza el análisis de los resultados.

Capítulo 7: Exposición de las conclusiones finales y descripción de posibles líneas futuras de trabajo.

Capítulo 8: Para terminar se realiza un presupuesto donde se detallan las tareas llevadas a cabo y el coste económico que han supuesto.

Se incluye además un apartado de Anexos con información relevante a nivel más específico del sistema (scripts, programas, tablas, etc.).

Capítulo 2. Introducción al reconocimiento automático de habla

El habla constituye la forma más natural de comunicación entre las personas, de ahí el gran interés que tiene el desarrollo de sistemas informáticos capaces de procesar el habla y generarla de forma automática. Es por ello muy importante determinar cómo se produce y percibe la voz a la hora de realizar su tratamiento automático.

El objetivo de un ASR es conseguir que la comunicación humano-máquina sea lo más natural y fluida posible. Para ello debería ser lo más parecido al sistema de audición humano.

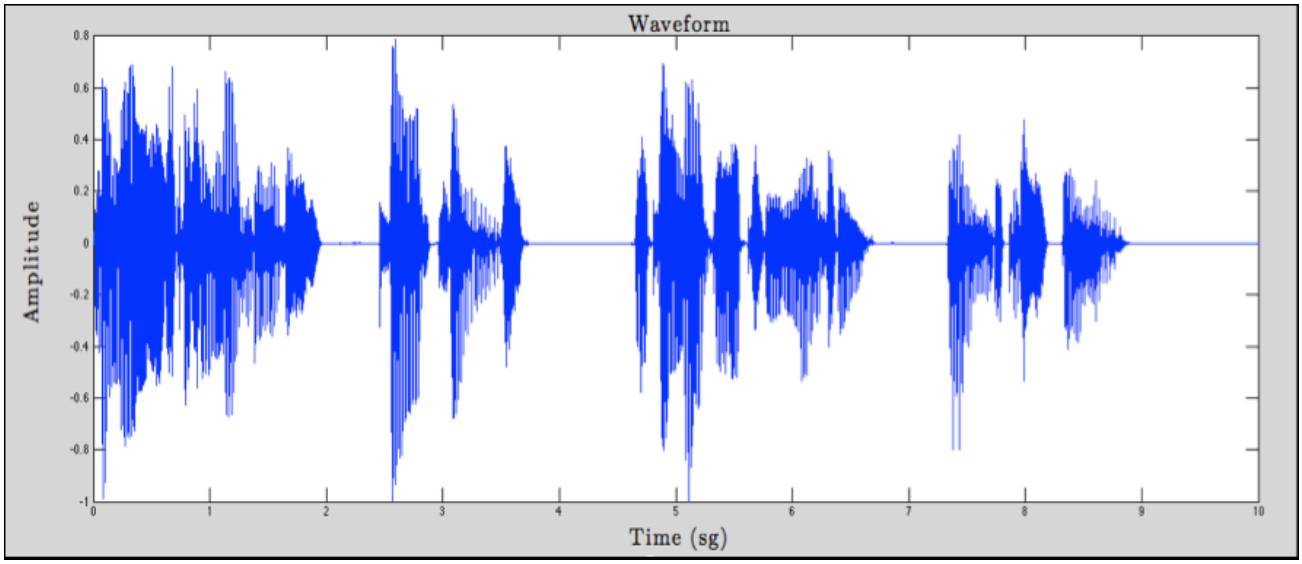


Figura 1: *Representación temporal de una señal de voz*

2.1 Definición de un sistema ASR

La **estructura básica** de un sistema de este tipo sería la siguiente:

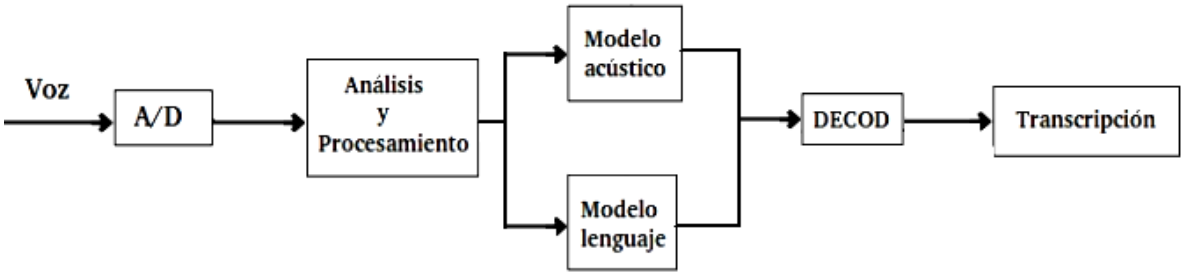


***Figura 2**: Esquema básico de un sistema ASR*

1. En primer lugar el sistema ASR **captura la voz** a través del equipamiento de transducción (micro, preamplificador, filtros...etc) y realiza la conversión A/D. Los datos observados con los que contamos en el PFC son los contenidos en la base de datos **Isolet,** la cual describiremos con detalle en el apartado 6.2.

2. En el bloque de **análisis y procesamiento**, la señal se convierte en una secuencia de vectores de parámetros y se extraen las características acústicas para diferenciar unos sonidos frente a otros. El número de parámetros debe ser reducido, puesto que la base de datos de entrenamiento siempre es limitada, por lo que cuantos más parámetros tenga la representación, menos fiables son los valores entrenados y, por otro lado, más costoso es el proceso de reconocimiento. Las técnicas utilizadas pueden ser **MFCC** (coeficientes que representan el habla) o **PLP** (predicción lineal perceptual), entre otras. Ambas serán explicadas en detalle en el capítulo 3.

3.1. En el **modelado acústico** se reciben como entradas los vectores de características generados en el bloque anterior. Se realiza un entrenamiento para asociar a cada vector una unidad con significado lingüístico, es decir, relaciona la secuencia de parámetros acústicos con la secuencia de palabras. Se pueden utilizar para ello diferentes tipos de modelos.

DTW: Dynamic Time Warping. Normalización temporal mediante propagación dinámica. Consiste en la comparación de los datos de entrenamiento con plantillas de referencia para obtener diferencias entre palabras. Dicha comparación se realiza calculando la distancia que existe entre ambos patrones. Será elegido como patrón reconocido aquel que guarde una menor distancia con el patrón de referencia. Se trata de un algoritmo sencillo de implementar pero muy dependiente del locutor, ya que no existen pronunciaciones idénticas. Además, es necesaria la utilización de vocabularios reducidos con el fin de poder obtener varias realizaciones de cada palabra y así paliar la mencionada variabilidad intralocutor [33][41].

HMM: Modelos ocultos de Markov. Reconocimiento de patrones con modelos estocáticos. Se trata de una máquina de estados finitos donde cada uno de ellos tiene asociada una distribución de probabilidad y en cada transición entre estados se produce un vector de parámetros. El objetivo del reconocedor es obtener la secuencia de estados más probable, dadas unas observaciones. Tienen alta tasa de reconocimiento en muchas circunstancias, pero siguen teniendo limitaciones importantes que no nos permiten aplicarlos en algunas situaciones del mundo real, donde esta tasa se ve fuertemente degradada.

NN: Redes neuronales. Son estructuras de procesamiento paralelo formadas por unidades simples (neuronas) conectadas entre sí. A finales de los años 80 y 90 surgieron como alternativa con el fin de superar las limitaciones de los HMM. Las NN utilizan unas técnicas discriminativas de clasificación que poseen ciertas propiedades que las hacen muy adecuadas para utilizarlas en los problemas de clasificación de patrones en ASR. A pesar de obtener un buen funcionamiento en problemas de clasificación estáticos, presentan notables limitaciones al tratar con la clasificación de señales de voz continua.

En este proyecto hemos optado por utilizar un sistema híbrido de NN y HMM que explicaremos con detalle en 5.3.

3.2. El **modelado del lenguaje** modela la probabilidad de aparición de una secuencia de palabras. Típicamente, calcula la probabilidad a priori de cada una de las palabras teniendo en cuenta cuáles son sus predecesoras.

4. En el bloque de **decodificación** se realiza propiamente el reconocimiento de voz. Se trata de encontrar la secuencia de palabras (W) pronunciadas con mayor probabilidad, dada una secuencia de vectores acústicos (X) parametrizados en el bloque anterior.

$$\hat{W} = \arg\max P(W)\, P(X|W) \qquad (1)$$

(1) se trata de la ecuación fundamental del ASR, siendo P(X|W) el modelo acústico y P(W) el modelo del lenguaje.

Esta tarea se realiza mediante el algoritmo de Viterbi, que obtendrá la secuencia de estados que ofrecen una mayor probabilidad, proporcionando, además de la secuencia de observación más probable, el camino de máxima verosimilitud. Dicho camino lo busca en una matriz donde se representan los estados de los HMM (eje de abcisas) frente a las tramas del habla (eje de ordenadas).

5. En el bloque de **transcripción** obtenemos los resultados finales del reconocimento en forma del texto esperado.

Una forma de entender todos los conceptos mencionados de manera conjunta sería mediante la siguiente figura, que esquematiza muy bien cada proceso.

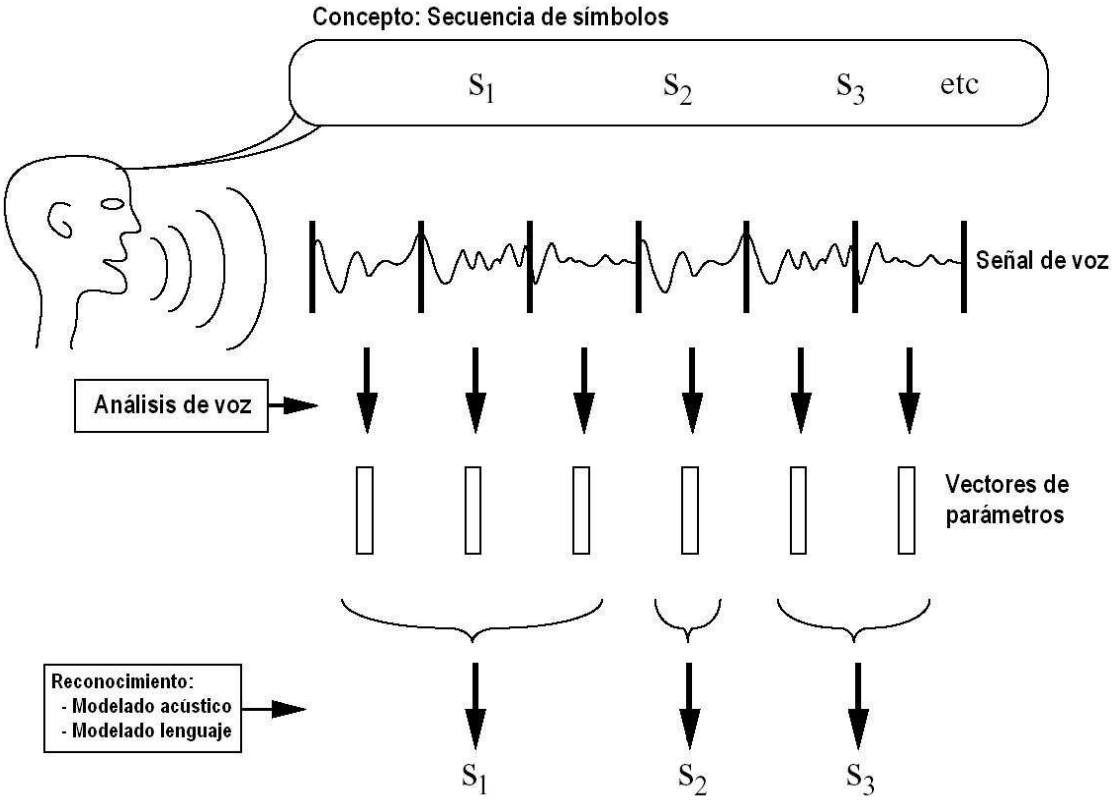


Figura 3: *Etapas de un sistema de reconocimiento de la señal de voz [40].*

Capítulo 3. Extracción de características

El gran inconveniente de los actuales sistemas ASR es su falta de robustez en condiciones acústicas adversas tales como el ruido de fondo o la variabilidad del canal. Una forma de solucionarlo y obtener una mayor robustez es explotar múltiples fuentes de información sobre la señal de habla en lugar de confiar sólo en una representación individual de la misma. Un ejemplo de estas fuentes múltiples de información son conjuntos de características acústicas extraídas mediante diferentes métodos: en nuestro caso serán las **características articulatorias.**

En un sistema estándar ASR existe un primer paso de prepocesado en el que se extraen características de la señal acústica con el objetivo de obtener una representación más compacta y más discriminativa de la señal de voz. Segmentos consecutivos de la señal de voz se convierten a secuencias temporales de vectores de parámetros. Este proceso se conoce con el nombre genérico de **parametrización o extracción de características.**

Su objetivo es la extracción de información relevante de la señal acústica, eliminando las redundancias, la información no relevante y la información asociada a las fuentes de variabilidad que tiene la misma, así como realzar aquellos aspectos de la señal que contribuyen significativamente a su identificación. La etapa de parametrización determinará en buena parte las prestaciones del sistema.

Resumiendo, la señal de habla se dividirá en segmentos para posteriormente extraer, de cada uno de ellos, un vector de características.

Para llevar a cabo la extracción de características o parametrización, primero se realiza la adquisición de la señal de voz. En nuestro PFC no será necesario este paso ya que contamos con la base de datos proporcionada, ISOLET.

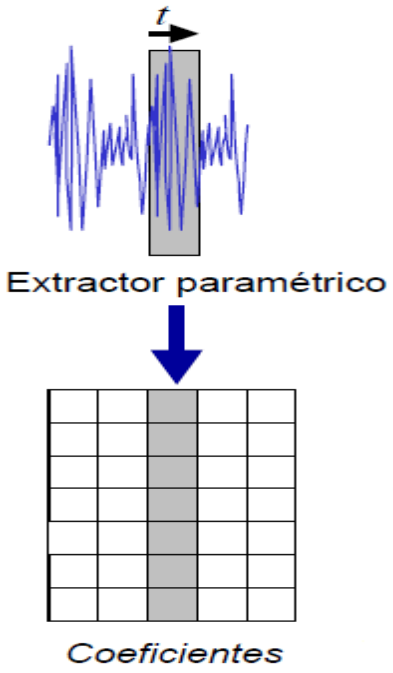


Figura 4: *Ilustración de la extracción de las características de una señal*

Existen numerosos tipos de técnicas de parametrización de la señal de voz. En este proyecto haremos mención a las 2 más extendidas, estas son: los coeficientes MFCC y los coeficientes PLP (éstos últimos serán los utilizados en nuestro PFC). Las dos técnicas utilizan el análisis localizado de la señal de voz, que consiste en enventanar la señal con una ventana normalmente de 25ms con un desplazamiento de 10ms, tratando estos fragmentos de la señal de manera independiente.

3.1 MFCC:Mel Frecuency Cepstral Coefficients

Se trata de la técnica más utilizada para la parametrización del habla de los sistemas automáticos de reconocimiento de voz, principalmente porque se adapta bien a las hipótesis utilizadas para estimar las distribuciones de estados en los HMM y, también, debido a la robustez de ruido que ofrece sobre otras técnicas alternativas de extracción de características. Este algoritmo está basado en la implementación de un banco de filtros, los cuales están espaciados unos de otros según una escala de frecuencia log-lineal.

El esquema básico para la extracción de los coeficientes es el siguiente:

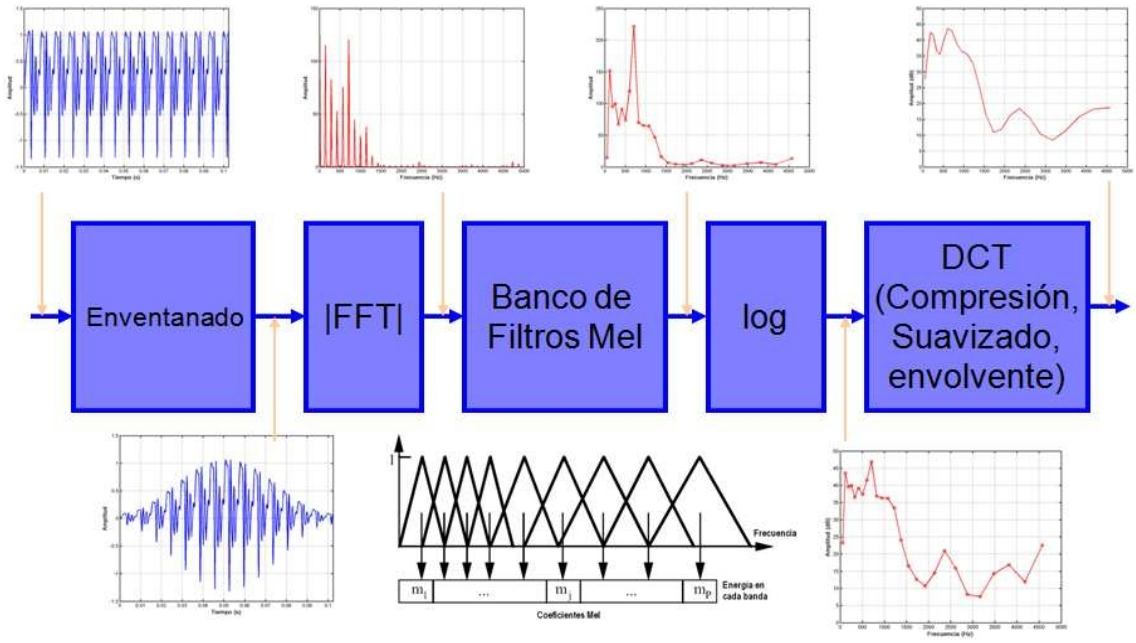


Figura 5: *Esquema básico de la extracción de coeficientes MFCC [42].*

En primer lugar, en el *enventanado* dividimos la señal en segmentos de pocos ms (solapados entre sí para para proporcionar transiciones suaves) para levar a cabo un análisis localizado. Posteriormente, se calcula la *FFT (Transformada de Fourier)* y se trabaja a partir de aquí con el módulo de la señal de voz, la cual se pasa por un *banco de filtros* triangulares (cuyos triángulos están espaciados de acuerdo con la escala de frecuencias MEL). A continuación se calcula la energía correspondiente en cada uno de los filtros así como su *logaritmo*, pasando por tanto al dominio de la potencia espectral logarítmica. El inconveniente de trabajar en este dominio es que los espectros de los filtros en las bandas adyacentes presentan un alto grado de correlación, para eliminarla se hace uso de la *Transformada Discreta del Coseno* (DCT), que convierte los coeficientes al dominio de la denominada "quefrencia" dando lugar, a los coeficientes MFCC.

En el caso de reconocimiento de voz se suelen calcular los 13 primeros coeficientes MFCC. Nos interesa captar los cambios temporales que se dan en el espectro, ya que juegan un papel muy importante en la percepción humana y la coarticulación. Por ello se utilizan además los coeficientes delta ya que capturan esa información, así como los coeficientes de aceleración. Estos últimos no son más que la primera y segunda derivada de los coeficientes MFCC respectivamente. Por lo tanto se suele trabajar con un vector de características de 39 dimensiones.

3.2 PLP: Perceptual Linear Prediction

El principio básico del análisis PLP es obtener mediante un modelo de todo polos una aproximación de la envolvente espectral de la voz. El modelo de todo polos lo obtenemos mediante el método de la autocorrelación y se utiliza la escala de frecuencias Bark.
El análisis PLP utiliza 3 conceptos de la psicofísica de la audición: la resolución espectral de banda crítica, las curvas isofónicas y la ley de potencia de intensidad a sonoridad, como se explica en [3], [36].
El orden del modelo todo polos variará según el grado de detalle que queramos que tenga la aproximación del espectro auditivo de la voz.

El esquema de extracción de los coeficientes PLP sería:

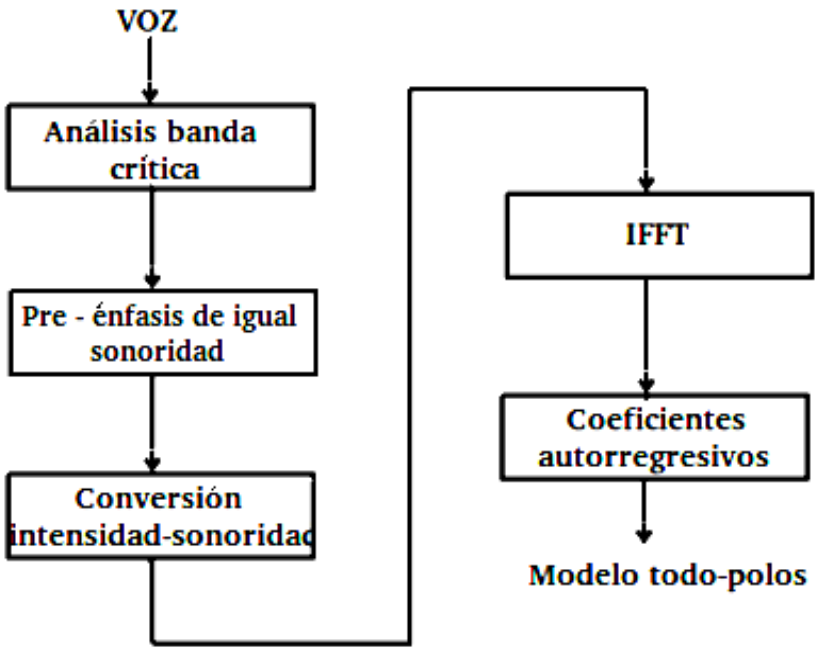


Figura 6: *Esquema básico de la extracción de coeficientes PLP [3].*

En primer lugar se realiza un *análisis en bandas críticas* enventanando la señal (normalmente con ventanas tipo Hamming de tamaño 20 ms) y transformando los segmentos de habla al dominio de la frecuencia calculando la FFT. A su vez, la

envolvente de la señal de voz se pasa por filtros de pendiente trapezoidal aplicados aproximadamente en intervalos de 1 Bark[2].

Posteriormente, el espectro es *pre-enfatizado* con el fin de aproximarlo a las desigualdades del oído humano con las distintas frecuencias. La *conversión intensidad-sonoridad* simula la relación no lineal entre la intensidad del sonido y el volumen percibido, reduciendo las variaciones espectrales en la amplitud de la banda crítica del espectro. Tras calcular la *Transformada Discreta Inversa de Fourier (IFFT)* se realiza el *suavizado espectral* en forma de modelo autorregresivo (es decir, realizando un análisis LPC -Linear Predition Coding-). Los coeficientes autorregresivos se transforman, finalmente, en variables cepstrales [35].

Los parámetros PLP están basados en los LPC (que realiza la estimación del modelo autorregresivo de todo polos del espectro de potencia), con la ventaja de que las técnicas PLP permiten la supresión de la información dependiente del hablante, eligiendo adecuadamente el orden del modelo en particular. Es por ello que los elegimos para implementar nuestro reconocedor. Según los experimentos de Flanagan (1955), se trata del modelo de orden 5 el que produce los mejores resultados en la identificación de información lingüística, [3].

El experimento detallado en [3] evalúa el uso de una parametrización PLP en un sistema ASR con locutores independientes, utilizando un reconocedor basado en distorsión de tiempo dinámico con multi-plantilla y 11 dígitos que constituyen la base de datos. La siguiente figura ilustra cómo la precisión del reconocimiento aumenta conforme lo hace el número de plantillas por palabra y como, a su vez, el sistema que utiliza PLP es mejor que el basado en LPC.

2 El espectro P(ω) se convierte a lo largo del eje de frecuencias ω en la frecuencia Bark Ω mediante la ecuación:

$$\Omega(\omega) = 6 \ln \{ \omega/1200\pi + [(\omega/1200\pi)^2 + 1]^{0,5}\} ,$$

reduciendo la sensibilidad espectral de la estimación espectral original, sobre todo en altas frecuencias.

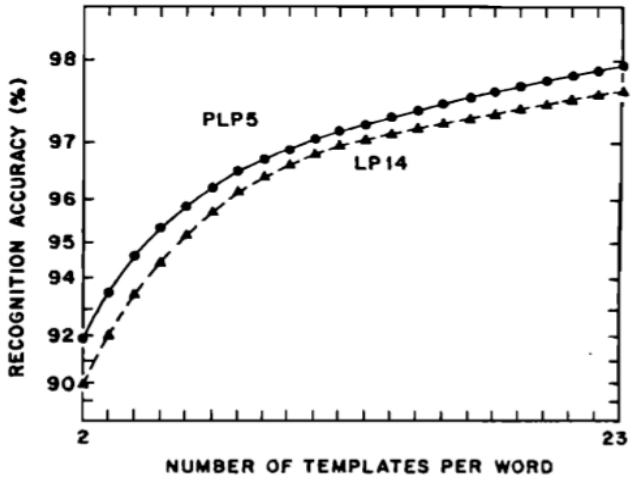


Figura 7: *Comparativa de la precisión de un sistema ASR para locutores independientes utilizando PLP de orden 5 y LPC de orden 14, [3].*

Los análisis PLP son computacionalmente eficientes y producen una baja dimensionalidad del habla, es decir, compactan la información.

Capítulo 4. Características articulatorias

Como hemos mencionado, una parte importante de los sistemas ASR es la extracción de características. En este PFC, adoptamos una arquitectura híbrida ANN/HMM para el modelado acústico y por lo tanto, una etapa importante del sistema es la formada por los clasificadores de las características de entrada en las unidades acústicas elegidas (en nuestro caso, alófono) para los que empleamos MLP (Multi-Layer Perceptrons).

Las **características articulatorias** describen propiedades en la producción del habla y se usan para representar las señales acústicas de una manera compacta. Se trata de clases abstractas que caracterizan los aspectos más esenciales de las propiedades articulatorias de los sonidos del habla (voz, nasalidad, etc...) [1].

Dichas características tienen una larga historia en cuanto a la creación de propuestas de técnicas para ASR debido, por ejemplo, a su capacidad para explicar los efectos de la coarticulación, la robustez de algunos de sus aspectos articulatorios que pueden detectarse mejor que otros o al hecho de que varios clasificadores (cada uno con un pequeño número de clases) pueden hacer mejor uso de los datos de entrenamiento que un gran clasificador individual [5] [6].

Los enfoques con AF's han tenido éxito en condiciones con ruido o en discursos hiperarticulados.

4.1 Fonética articulatoria

La fonética articulatoria es la rama que estudia los órganos que intervienen en la formación de los sonidos. Es pues, la vertiente de la fonética que nos interesa en el desarrollo de este PFC. A parte de la fonética articulatoria, existen también la **fonética acústica** que se ocupa de las ondas sonoras que se generan al hablar y la **fonética auditiva** que trata el punto de vista del oyente (receptor) [24].

La fonación es el trabajo muscular realizado para producir sonidos inteligibles. El objetivo final es la articulación de palabras a través de un proceso que, en rasgos generales, comienza con la generación de un flujo de aire en los pulmones, la modificación de ese flujo de aire en las cuerdas vocales, y su posterior perturbación por algunas constricciones de los órganos superiores. Así, en el proceso fonador intervienen distintos órganos a lo largo del llamado tracto vocal, que en nuestro caso asumiremos que se restringe a la zona comprendida entre las cuerdas vocales y las aberturas finales: los labios y las fosas nasales [22], [23].

Para describir este proceso de forma más detallada es necesario el conocimiento del aparato fonador humano, el cual nos permite producir sonidos mucho más desarrollados que el de los animales, por ejemplo. Esto es debido a que nuestra laringe ocupa una posición más baja, permitiendo así que las cavidades bucal y faríngea tengan un espacio más amplio para poder conformarse de una u otra forma para actuar como resonadores.

Los órganos fonadores se clasifican en 3 zonas: cavidad infraglótica, laringe o cavidad glótica y cavidad supraglótica, que se muestran detalladamente en la figura inferior.

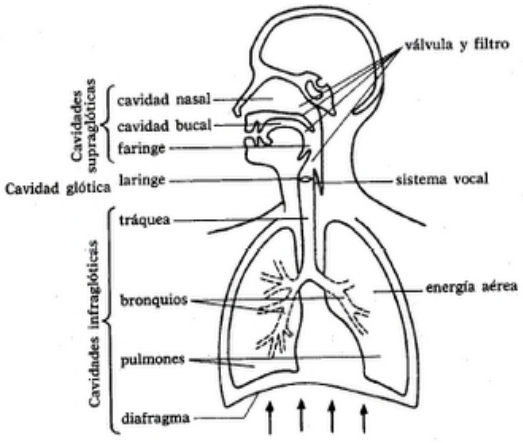


Figura 8: *Órganos y zonas del aparato fonador*

Cavidad infraglótica

Está formada por la tráquea, los bronquios, los pulmones y el diafragma. El diafragma comprime los pulmones llenos de aire para expulsarlo con la fuerza y el ritmo necesario para producir el sonido.

En esta cavidad tiene lugar el proceso de la inspiración y espiración. En el primero de ellos, el diafragma se contrae , las costillas se elevan y el volumen de la cavidad torácica aumenta, el aire exterior entra a los pulmones pasando por la nariz o la boca, la faringe y los bronquios. En el proceso de espiración, el diafragma se relaja y el aire se expulsa de los pulmones, produciéndose la fonación

Laringe o cavidad glótica

La laringe es un tubo construido por músculos y cartílagos (cricoides, tiroides y aritenoides) que contiene las cuerdas vocales y la glotis.

Las cuerdas vocales no son más que un músculo que forma un par de repliegues vocales en la línea media de la laringe. Dependiendo de si este músculo es grueso o delgado, el sonido emitido será grave o agudo respectivamente. Además, dependiendo de cómo sean las cuerdas vocales existirán diferentes aspectos en la fonación. Si las cuerdas son más largas y gruesas, la velocidad de vibración es más lenta, si las cuerdas son más cortas y delgadas mayor resulta la frecuencia. La velocidad de vibración media para voces masculinas es de 100/150 p/s y unos 200/300 p/s para voces femeninas.

La glotis es el espacio delimitado por las cuerdas vocales, puede estar abierta para respirar o puede cerrarse para hacer posible la fonación. Los sonidos en los que sólo actúa la glotis se denominan glotales.

Cavidades supraglóticas

Se trata de la zona del aparato fonador que más tiene que ver con nuestro sistema en particular ya que es la que contiene los órganos articuladores. El objetivo final de los órganos articuladores es la producción de diferentes grados de contracción en ciertos puntos del tracto vocal.

La cavidad supraglótica está formada por la cavidad nasal, la cavidad bucal y la faringe, que actúan como resonadores.

Se muestran en detalle mediante la siguiente figura.

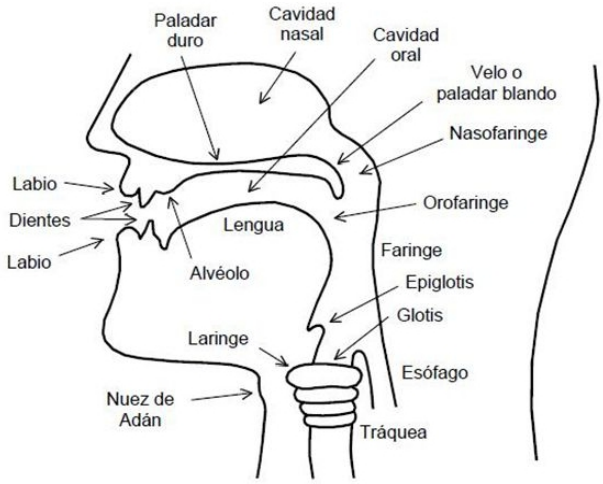


Figura 9: *Órganos fonadores en las cavidades supraglóticas [25].*

La cavidad bucal puede adaptar infinidad de formas y volúmenes diferentes gracias a los movimientos de la lengua, que se trata del órgano más importante y que ocupa la gran parte de dicha cavidad.

El paladar blando será quien determine si un sonido es o no nasal (dependiendo de si el aire pasa por la nariz o no, respectivamente).

Los labios, gracias a su gran movilidad, modifican el efecto de la cavidad bucal.

4.2 Niveles articulatorios

Nos centraremos en un enfoque para estudiar la gran variabilidad en la pronunciación del habla conversacional. En este PFC utilizaremos un conjunto de iteraciones de la base de datos utilizando 7 niveles o características articulatorias, que se muestran a continuación [5].

Téngase en cuenta que, aunque proporcionaremos también su denominación en español, hemos mantenido los nombres en inglés para no alterar el modelo y el mapeo en los que nos hemos basado (Wester (2003) [20] y Ladefoged (1982) [21], respectivamente), además de ser el inglés, el idioma de la base de datos utilizada.

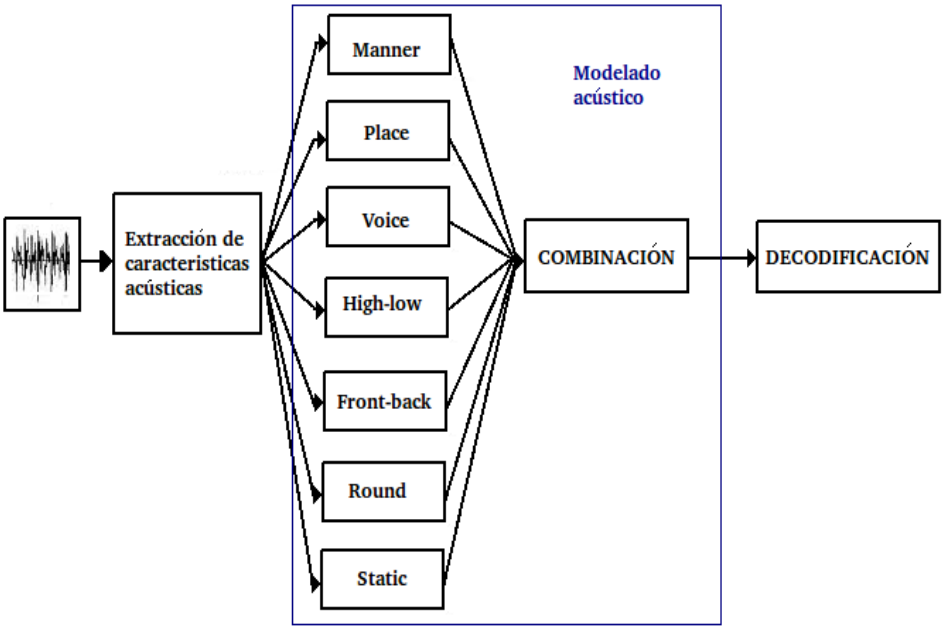


Figura 10: *Niveles de AF's*

Cada uno de estos niveles de características toma diferentes valores:

AF	VALOR
manner	approximant, retroflex, fricative, nasal, stop, vowel, silence
place	bilabial, labiodental, dental, alveolar, velar, nil, silence
voice	+voice, -voice, silence
high-low	high, mid, low, nil, silence
fr-back	front, central, back, nil, silence
round	+round, -round, nil, silence
static	static, dynamic, silence

Tabla 1: *Tabla de valores de AF's*

donde el valor positivo (+) indica la presencia de la característica citada y el valor negativo (-), la ausencia de la misma. Y siendo las traducciones en español de los valores, las siguientes: high - *alto*, mid - *medio*, low - *bajo*, nil - *nulo*, silence - *silencio*, front - *delantero*, central - *central*, back - *posterior*, static – *estático* y dynamic – *dinámico*.

Manner: *(Esp. Modo de articulación*). Configuración e interacción de los órganos articuladores (lengua, labios y paladar) en el discurso sonoro. El concepto de *manner* se utiliza sobre todo para consonantes, aunque también es importante para la identificación de vocales.

- Retroflex (*esp. Retrofleja*): Consonante coronaria donde la lengua tiene una forma plana, cóncava o incluso doblada y se articula entre la cresta alveolar y el paladar duro.
- Approximant (*esp. Aproximante*): donde hay muy poca obstrucción. Se trata de sonidos donde los articuladores se acercan entre ellos, pero no lo suficiente, ni con la suficiente precisión articulatoria para crear el flujo de aire turbulento. Están por lo tanto entre los sonidos fricativos (que producen flujo turbulento) y vocales (que no lo producen).
- Fricative (*esp. Fricativa*): Existe un flujo de aire turbulento y ruidoso en el lugar de la articulación. El aire pasa por un canal formado por 2 articuladores que se encuentran juntos, es por tanto, un espacio estrecho. Estos pueden ser, por ejemplo, el labio inferior contra los dientes superiores (/f/). Este flujo de aire turbulento se llama *fricacion*. La mayoría de los idiomas cuentan con fricativas, aunque hay algunos que tienen solo una /s/.
- Nasal (*esp. Nasal*): Existe obstrucción de la vía oral pero el aire pasa a través de la nariz. La forma y posición de la lengua determinarán la cavidad resonante que dará a las diferentes nasales sus sonidos característicos. Ejemplos en inglés serían por ejemplo /m/ o /n/. Casi todas las lenguas tienen nasales.
- Vowel (*esp. Vocal*): Se trata de un sonido pronunciado con el tracto vocal abierto de forma que no hay ningún aumento de la presión de aire en cualquier punto por encima de la glotis.
- Stop (*esp. Oclusiva*): oclusión o bloqueo del tracto vocal oral, sin flujo de aire nasal, con lo que el flujo de aire se detiene completamente. Si la consonante es la voz, la voz es el único sonido durante la oclusión; si no existe voz, *stop* corresponde a silencio total. Lo que oimos como /p/ o /k/ es el efecto que tiene el inicio de la oclusión en la vocal precedente, así como la explosión de liberación y su efecto sobre la vocal siguiente. Todos los idiomas lo tienen.

Place: (*esp. Punto de articulación de las consonantes*). Se trata del punto de contacto donde se produce una obstrucción en el tracto vocal entre un gesto articulatorio, un articulador activo (normalmente una parte de la lengua) y una ubicación (normalmente la parte del "cielo" de la boca). Es sencillo de ver en los labios pero dificil de escuchar. Es importante destacar que para un mismo punto de articulación puede haber diversos modos de articulación. Se trata de la característica más difícil de clasificar para los sistemas automáticos, debido sobre todo a la coarticulación, donde existen 2 puntos de articulación para una misma consonante (véase la sección 4.3). Los nombres coinciden para el castellano.

- Bilabial: La articulación se realiza uniendo ambos labios. Ej. /b/, /p/, /m/
- Labiodental: Articulación tocando el labio inferior con los dientes superiores. Ej. /f/, /v/
- Dental: Se pronuncian insertando la punta de la lengua entre los dientes. Ej. /t/, /d/, /n/
- Alveolar: Articulación con la lengua contra o cerca de la cresta alveolar superior, que se llama así porque contiene los alveolos de los dientes superiores. Ej. /n/, /s/, /z/
- Velar: Se producen con la parte posterior de la lengua (dorso) contra en el paladar blando o velo (parte trasera del techo de la boca). Ej. /k/, /g/

Voice: (*esp. Sonoridad o Carácter tonal (sonoro) o no tonal (sordo)*)**.** Indica si la vibración de las cuerdas vocales se produce con la articulación del segmento.

- Sonoros: Son aquellos sonidos que hacen vibrar las cuerdas vocales. Esta vibración es cuasi periódica y su espectro es muy rico en armónicos, que son múltiplos de la frecuencia de vibración de las cuerdas. A esta frecuencia de vibración de las cuerdas se le llama frecuencia fundamental. La frecuencia fundamental depende de la presión ejercida al pasar el aire por las cuerdas y de la tensión de éstas. En un hombre la frecuencia fundamental se encuentra en el rango 50-250 Hz, mientras en la mujer el rango es más amplio, encontrándose entre 100 y 500 Hz.

High - low: (*esp. Punto de articulación de las vocales*. Describe la posición vertical de la lengua en la boca). Difícil de clasificar debido a que cada persona tiene distinta forma para el tracto vocal, la articulación es fuertemente dependiente del hablante y las posiciones no son absolutas. Ej. High: /i/, /ɪ/, /u/, /ʊ/, Mid: /e/, /ɛ/, /ʌ/, /o/, /ɔ/, /ɔɪ/, Low: /æ/, /a/, /ɑ/, /aɪ/, /aʊ/.

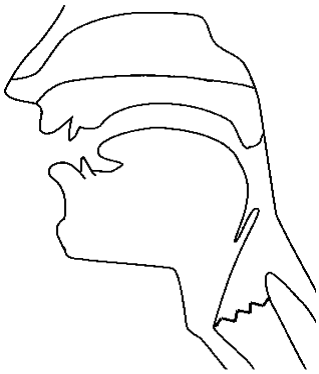
Front – back: (*esp. Punto de articulación de las vocales. Describe la posición horizontal de la lengua en la boca*). Difícil de clasificar debido a que cada persona tiene distinta forma para el tracto vocal, la articulación es fuertemente dependiente del hablante, las posiciones no son absolutas. Ej. Front: /i/, /ɪ/, /e/, /ɛ/, /æ/, Central: /a/, /aɪ/, /aʊ/, Back: /ʌ/, /ɑ/, /o/, /ɔ/, /u/, /ʊ/, /ɔɪ/ .

Round: (*esp. Posición de los labios*). Los labios pueden tener una forma redonda en la pronunciación (+) ej. /o/, /ɔ/, /u/, /ʊ/, /aʊ/, /ɔɪ/ o no tenerla (-) ej. /i/, /ɪ/, /e/, /ɛ/, /æ/, /a/, /ʌ/, /ɑ/, /aɪ/.

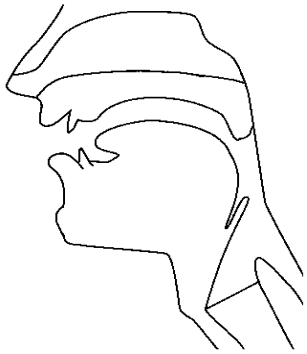
Static: (*esp. Indicación de la tasa de cambio acústico*). Se produce, por ejemplo, durante los diptongos, que serán clasificados como dinámicos (o "no estáticos").

Es probable que los diferentes aspectos de articulación muestren diferentes grados de robustez y no se deterioren bajo el mismo grado de condiciones acústicas adversas. Por ejemplo, *voice* se detecta de manera bastante robusta, sin embargo *place* tiende a ser menos robusta y más dependiente de las características del tracto vocal del hablante.
El módulo de combinación de la figura 9 puede, por ejemplo, utilizar valores de confianza como base para asignar pesos a las salidas de los sub-clasificadores. Por esas razones un enfoque de modelado acústico basado en la casificación en términos de AF's es, normalmente, más robusto en condiciones acústicas adversas.

Para detallar gráficamente algunos tipos de las citadas características articulatorias se muestran estos ejemplos [19]:

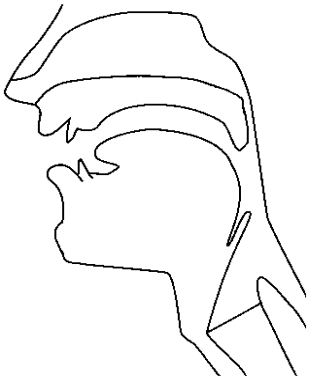


Sonora

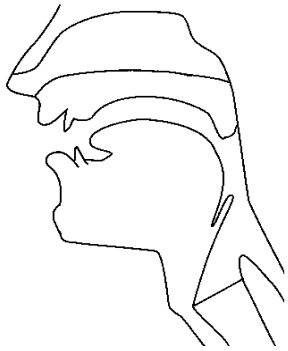


Sorda

Fijándonos en las cuerdas vocales, vemos como en el primer caso se genera una vibración de las mismas, mientras que en el segundo se mantienen bloqueadas, hecho que impide que se produzca sonido alguno.

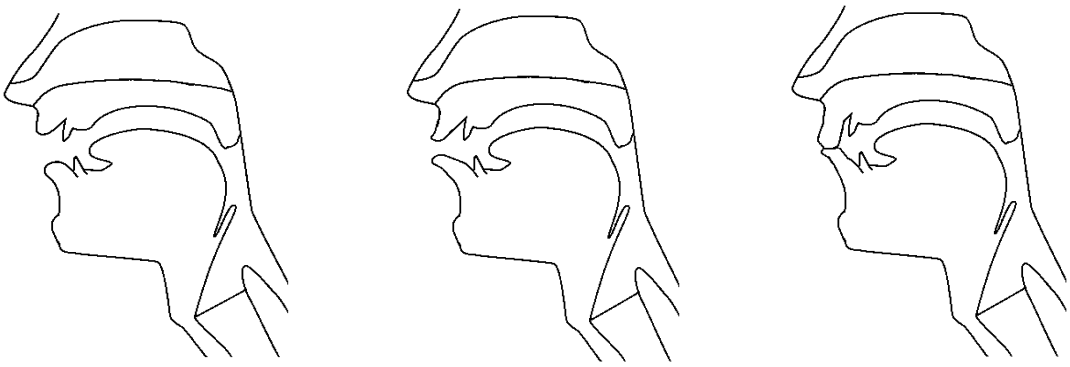


Nasal



Oral

Con estos 2 ejemplos se diferencia cláramente cómo para el caso *nasal* se permite que el flujo de aire suba hasta las fosas nasales para ser expulsado, mientras que en el caso *oral* este conducto permanece cerrado, permitiendo la salida del aire tan sólo por la boca.

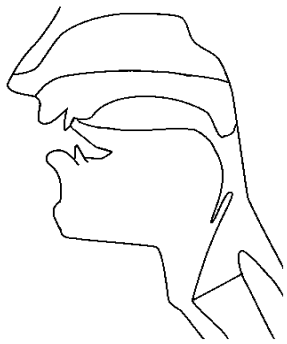


Labios extendidos Labios redondeados Labios en reposo

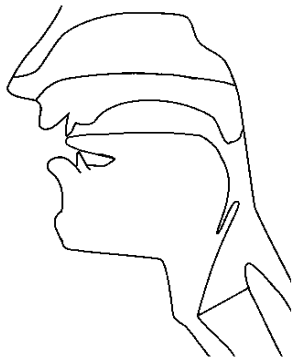
Observamos la característica a la que nos referimos como *round* fijándonos en la posición que adquieren los labios.

Punto art. bilabial – Modo art. fricativa Punto art. labiodental – Modo art. fricativa

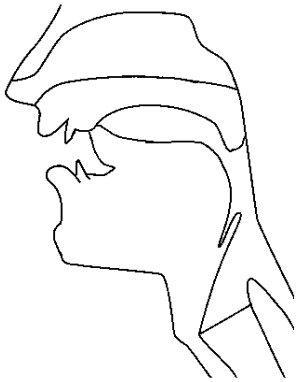
Comparando estas 2 nuevas imágenes diferenciamos, fijándonos en los labios y dientes, 2 puntos de articulación (*place)* para un mismo modo de articulación (*manner)*.



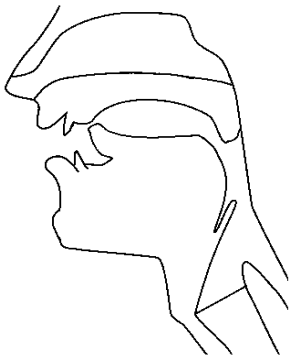
Punto articulación dental – lengua reposo



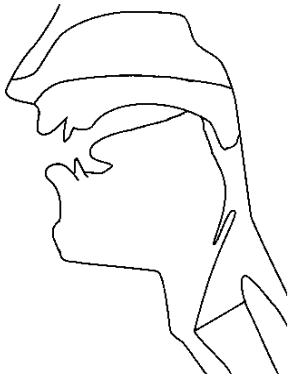
Punto articulación dental – lengua fricativa



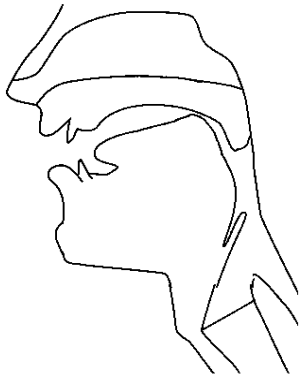
Punto articulación alveolar – lengua reposo



Punto articulación alveolar – lengua fricativa



Punto articulación velar – lengua reposo



Punto articulación velar – lengua fricativa

Otro caso sería, como observamos en la zona de la boca para las 6 imágenes superiores, mismo punto de articulación (*place*) con diferentes modos de articulación (*manner*).

4.3 Problemática del modelo

Incluso para sistemas de reconocimiento de habla de alta precisión en entornos controlados, su salida a escenarios reales supone una pérdida de prestaciones debido, fundamentalmente, a 2 grandes problemas: la distorsión de la señal de habla debido al ruido ambiental o al desajuste del canal y la variabilidad articulatoria del hablante.

1. La **distorsión de la señal** es generada principalmente por el ruido de fondo que se superpone a la señal de habla en cualquier discurso conversacional. Serían agentes distorsionadores, por ejemplo, el entorno acústico en el que se produce el habla, el ruido aditivo, las características de la sala o los efectos del canal de transmisión.

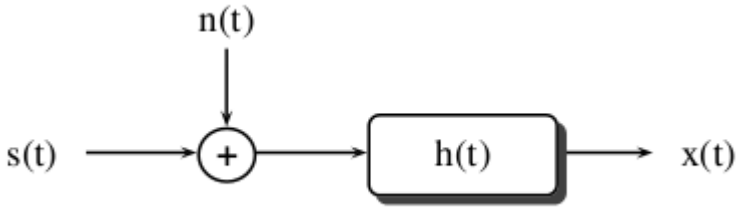


Figura 11: *Modelo simplificado de una señal de habla con ruido aditivo*

Hablamos de *mismatch* (esp. desajuste) cuando al comparar el habla generada en un entorno acústico específico, difiere de la misma señal grabada bajo condiciones acústicas distintas. Se trata de un problema muy típico en el uso de reconocedores, cuando el entrenamiento se realiza en un entorno en el que después no será utilizado.

2. Por otra parte, la **variabilidad articulatoria del hablante** genera, a su vez, varios problemas relacionados entre sí:

Antes de detallarlos es importante describir el funcionamiento de los **modelos articulatorios** [10], donde cada fonema tiene una “posición de destino” para cada articulador. El objetivo ideal sería alcanzar para cada fonema esta posición.

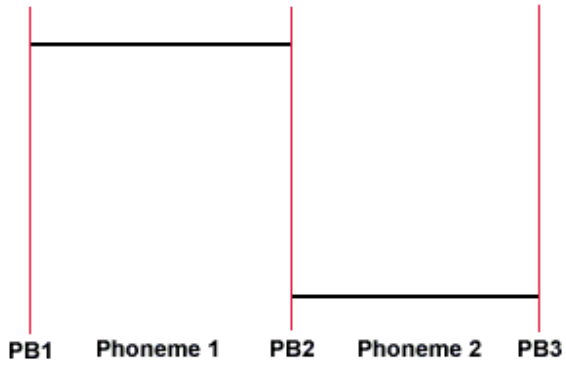


Figura 12: *Las líneas horizontales representan las posiciones de destino ideales para 1 articulador simple y 2 fonemas adyacentes. PB1, PB2 y PB3 son los límites ideales para el Fonema 1 y el Fonema 2.*

En un entorno real para habla continua esta representación no existe, excepto cuando los límites están caracterizados por pausas. Por lo tanto, es más acertado cuando hablamos de límites entre fonemas, referirnos a un punto aproximado de transición entre ellos, más concretamente entre sus posiciones de destino articulatorias. Estas transiciones son causadas por los movimientos de los articuladores al pasar de un fonema al siguiente.

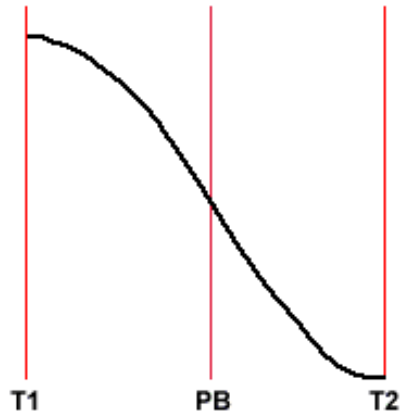


Figura 13: *Transición articulatoria ideal desde la posición destino del fonema 1 (T1) hasta la posición destino del fonema 2 (T2). PB representa la posición aproximada del límite entre fonemas.*

El punto de transición comparte características acústicas de ambos fonemas con cambios graduales, donde en ocasiones predomina el fonema 1 y en otras el fonema2.

Existe una tendencia de los articuladores a resistirse al movimiento o a cambios en su dirección de movimiento, es la **inercia articulatoria.** Por ejemplo, el "cuerpo" de la lengua es más pesado y se moverá más lentamente que la "punta" de la lengua.

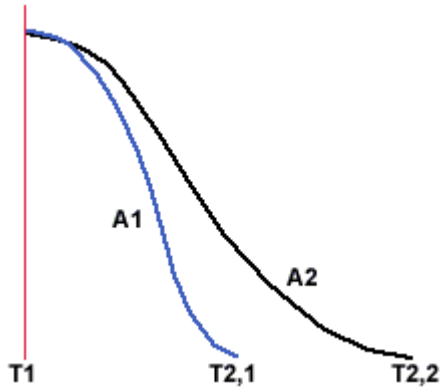


Figura 14: *2 articuladores con distintos grados de inercia recorrerán la misma distancia en tiempos diferentes. El articulador A1 tiene menos inercia y recorrerá la distancia entre los límites de los fonemas T1 y T2 más rápidamente que el articulador A2.*

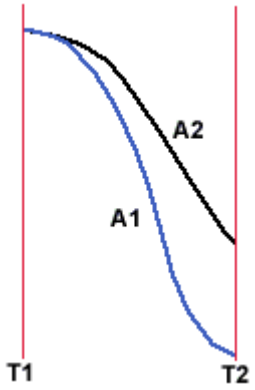


Figura 15: *Para el mismo tiempo el articulador A1, que tiene una menor inercia, se acerca más a la posición de destino del fonema T2.*

En ocasiones no existe tiempo suficiente para que un articulador llegue a la posición de destino de un fonema. Puede ocurrir por ejemplo porque tenga demasiada inercia para alcanzar dicha posición.

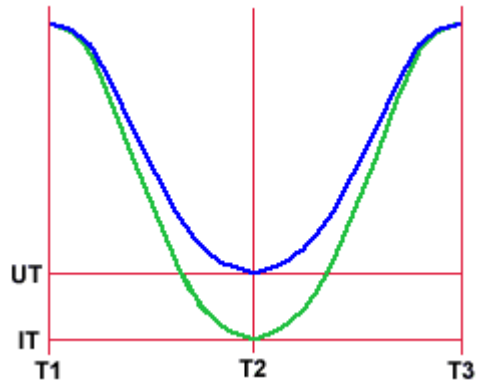


Figura 16: *IT se refiere a la posición de destino ideal, pero observamos cómo el articulador sólo alcanza la posición UT, que no se corresponde con la deseada.*

Existe también otro fenómeno muy importante que es el **solapamiento articulatorio,** en el que las trayectorias en el movimiento de los articuladores para llegar de un fonema al siguiente se solapan. Esto evidencia el hecho de que varios articuladores se mueven de forma simultánea.

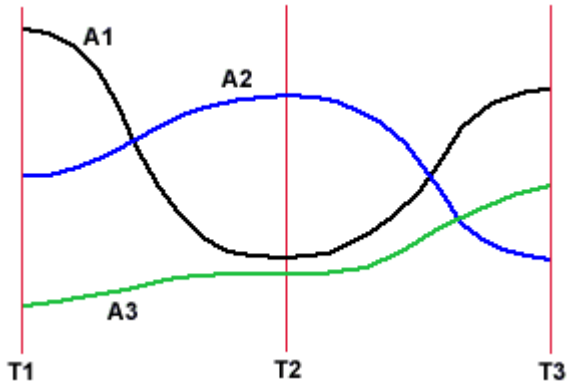


Figura 17: *A1, A2 y A3 representan a 3 articuladores solapándose entre los límites T1, T2 y T3 de 2 fonemas adyacentes*

Resultan ahora más evidentes los problemas que surgen con la variabilidad articulatoria del hablante:

Asincronía: Se trata de un fenómeno causado por las variaciones en la producción del habla natural, en el cual existe un desplazamiento entre el límite del fonema y los límites de sus características articulatorias asociadas. Por ejemplo, el punto de arranque de las AF's cambia para diferentes condiciones contextuales del habla.
Como consecuencia, el detector (que utiliza criterios asimilados con el entrenamiento) podría dar lugar a representaciones inexactas como salidas del reconocedor.

Son varios los estudios que se han realizado para comprobar estas alteraciones [15], dando como resultado la obtención de valores superiores en cuanto a detección y precisión para el caso ideal (sin errores del detector ni asincronía) que para el caso real de un sistema de detección en un discurso conversacional.

Test Data Type	System	Corr	Acc
Ideal (upper bound)	AT CRF	71.49	70.31
Detected (real case)	AT CRF	64.87	62.32

Tabla 2: *Ejemplos de valores de tasa de detección correcta (corr) y precisión (acc) en un sistema AT CRF para los casos ideal y real.*

Es sobre todo por este problema de asincronía por lo que un enfoque basado en AF's resulta bastante apropiado, ya que poseen una gran habilidad para describir información contextual así como para cambiar asíncronamente.

Coarticulación: Es la manera en que los movimientos de los diferentes articuladores afectan a otros, o lo que es lo mismo, la modificación de un sonido del habla debido a la anticipación o retraso de los sonidos adyacentes. Un ejemplo muy claro sería cómo el cambio en la AF "*place*" de las velares plosivas depende de la vocal que le sigue: /k/ en "*kitchen*" es diferente a /k/ en "*car*".
En la mayoría de las ocasiones se trata de modificaciones imperceptibles para el oído humano e irrelevantes para un sistema HSR (-Human Speech Recognition-), pero que tienen su efecto en el modelo estadístico del sonido en cuestión.

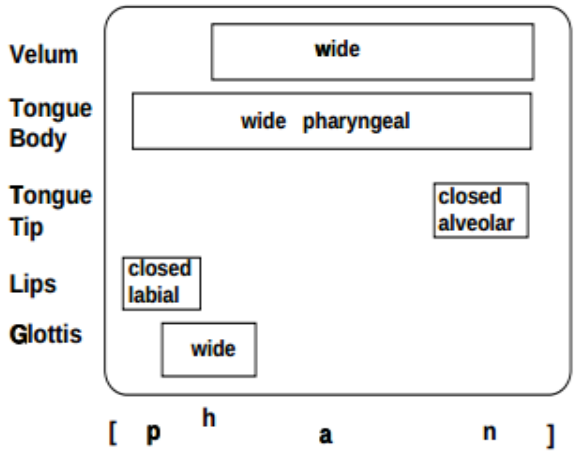


Figura 18: Ritmo relativo en la producción articulatoria de la palabra "pan" en inglés.

Debido a la inercia de los articuladores (explicada en lineas superiores), las constelaciones articulatorias no cambian tan rápidamente como se esperaría de un segmento al siguiente. Las propiedades espectrales de estos segmentos están afectadas por el tracto vocal del hablante [34].

Una consonante "coarticulada" [32] es un sonido en el que 2 consonantes individuales son pronunciadas al mismo tiempo. Este fenómeno de la coarticulación tiene, por tanto, mucho que ver con nuestra AF *place of articulation* ya que una misma consonante cuenta con 2 puntos de simultáneos de articulación. Éstos pueden ser de 2 tipos:

1. El mismo *modo de articulación (manner)*, por ejemplo la oclusiva labial-velar sorda /kp/ que se pronuncia simultáneamente con el velo del paladar /k/ y los labios /p/.

2. Diferente *manner* en el segundo punto de articulación, como por ejemplo la labialización en el sonido /$k^{w/}$ o la velarización en el sonido inglés /$I^{ɣ}$/.

Asimilación: Se trata de un fenómeno ligado a la coarticulación, en el que un segmento de un sonido del lenguaje se articula con rasgos fonéticos del segmento adyacente. La pronunciación del segmento se "acomoda" a la del siguiente en una misma palabra o en el límite entre 2.

Todos estos procesos provocados por la variabilidad articulatoria del hablante ocurren siempre, en todos los idiomas, para todas las secuencias de sonidos no separadas por pausas. El hablante no puede evitarlo, debe articular sonidos adyacentes de manera natural y comprensible, y para ello son necesarios dichos fenómenos. Una de las razones por las que acentuamos ciertas palabras es para incrementar la duración de algunos segmentos y así poder alcanzar las posiciones de destino deseadas en determinados fonemas. Lo único que se puede hacer pues, es intentar aminorar los efectos de dichos problemas en el reconocimiento.

Sabiendo que una de las mayores causas de la asincronía en las AF's es la variación contextual del habla natural, se puede permitir al detector que aprenda directamente de dichas variaciones. Para ello, se debe introducir información a largo plazo del habla al detector. Es conocido que este tipo de información ayuda al reconocimiento y la detección resulta más estable que usando sólo información local.

AFDT sys	a	A	E	h	H	i	I	N	S	u	U
MFCC	0.94	4.27	1.18	0.94	0.59	1.07	2.44	1.74	0.56	1.88	7.16
Long Term	0.75	3.57	0.94	0.77	0.38	0.91	2.36	1.15	0.43	1.32	4.55

Tabla 3: *Distancias en frames entre los límites AF-fonema de algunos sonidos comparando un detector que utiliza MFCC y otro que utiliza información a largo plazo ("long term")*.

Observamos como con el uso de esta técnica las distancias se reducen y, por consiguiente, se reducirá la asincronía [15].

Por otra parte, para intentar mejorar el problema de la coarticulación se utilizan técnicas como el reconocimiento con bandas anchas de frecuencias (en la parametrización) o la utilización de HMM's con mayor número de estados.

Un obstáculo bastante común es la falta de datos etiquetados con valores de AF's para contextos temporales largos. Esto puede solucionarse ponderando los valores en dicho contexto temporal o dividiendo el mismo en 2 partes para cada una de las cuales se utilizará un clasificador independiente para el entrenamiento. Se ha comprobado que las tasas de error mejoran hasta en un 23% [37].

Capítulo 5. Un Reconocedor Automático de Habla basado en la extracción de características articulatorias. Enfoque híbrido

En los últimos años se han desarrollado multitud de teorías y experimentos que han establecido la viabilidad de las Redes Neuronales (ANN's) para su uso en reconocimiento del habla. Se ha demostrado que dichas ANN's pueden utilizarse para aumentar la precisión de reconocedores cuya estructura está basada, esencialmente, en los Modelos Ocultos de Markov (HMM's). En particular, se ha demostrado que se pueden entrenar discriminativamente unas estructuras de ANN's bastante sencillas para estimar la emisión de probabilidades para un HMM.

Tras la observación de sistemas sencillos de reconocimiento de voz basados en este enfoque, se confirma que son tan precisos como los sistemas que utilizan HMM's, además de resultar más eficientes en cuanto a CPU y memoria.

Utilizaremos pues, en este PFC, las ANN's como **clasificadores** que tomarán como entradas los vectores de parámetros ya calculados y estudiaremos los resultados.

A continuación se detallan cada uno de los componentes del sistema híbrido así como su funcionamiento.

5.1 Modelos ocultos de Markov (HMM's)

Un modelo oculto de Markov es un autómata estocástico de estados finitos, construido sobre un conjunto estados también finitos $Q = \{q_1, \ldots., q_k\}$. Cada uno de ellos está asociado a una distribución de probabilidad específica. Dichos estados pueden ser emisores o no emisores, dependiendo de si

emiten función de probabilidad o no lo hacen. Los HMM's modelan la secuencia de vectores de caracteristicas como un proceso estacionario por tramos, en el que cada segmento será asociado a un estado del HMM.

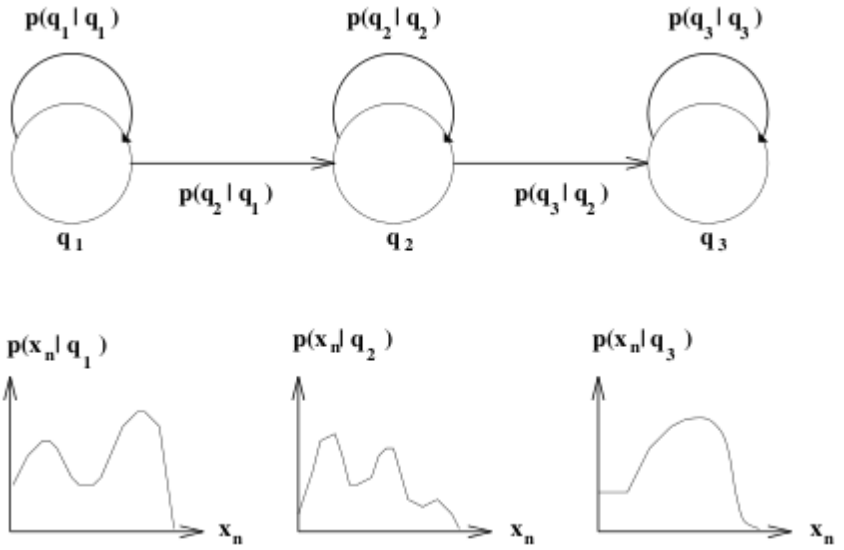


Figura 19: *Ejemplo de HMM de 3 estados, siendo q_i conjunto de estados, $p(x_n|q_i)$ la densidad de emisión de probabilidad asociada a cada estado y $p(q_j|q_i)$ la transición de probabilidad del estado q_i al estado q_j [7].*

El sistema cuenta con 2 procesos estocásticos concurrentes: la secuencia de estados y un conjunto de de emisiones. Al modelo de Markov se le denomina “oculto” porque el proceso estocástico subyacente no es directamente observable, pero afecta a la secuencia de observación de características acústicas.

Generalmente se adopta un esquema jerárquico para reducir el número de posibles modelos donde por ejemplo, una frase se modela como una secuencia de palabras. Para reducir aún más el número de parámetros (o hacer mejor uso del material de entrenamiento) y para evitar la necesidad de un nuevo entrenamiento cada vez que una palabra se añade al conjunto, los modelos de palabras están a menudo constituidos por una concatenación de unidades de sub-palabra. Las unidades de sub-palabra más utilizadas en sonidos del habla son los **fonemas**. Se trata de categorías sonoras que resultan suficientes para distinguir las diferentes palabras en un lenguaje. Cada fonema tendrá una distribución de salida. Un modelo oculto de Markov para una secuencia de fonemas se construye concatenando los modelos ocultos entrenados para cada fonema por separado. Un modelo de palabra consiste en la concatenación de fonemas y un modelo de frase consiste, a su vez, en la concatenación de modelos de palabras.

Al hablar, enlazamos fonemas, de forma que un fonema puede verse alterado por los fonemas que lo rodean (este es el efecto conocido como coarticulación). Por ello, habitualmente distinguimos tres partes en cada fonema. La primera y la tercera estarán condicionadas por los fonemas anterior y posterior, respectivamente. La segunda comprende la parte estable del fonema. Por lo tanto es

comprensible asociar a cada fonema un HMM de tres estados, aunque esto no descarta el uso de modelos con más de tres estados en casos concretos.

Los modelos ocultos de Markov usados en el reconocimiento de voz tienen dos características:

1) Si contamos con información actual, la historia anterior de la cadena no influye en la evolución futura de la misma (hipótesis de Markov de primer orden).
2) De manera similar, la observación actual de la cadena de estados, no depende de observaciones pasadas ni de la evolución futura si se ha especificado la última transición de la cadena (hipótesis de independencia de las salidas).

5.2 Redes Neuronales Artificiales (ANN's)

Una red neuronal artificial es un sistema de aprendizaje y procesamiento de información basado en la interconexión de neuronas entre sí, para la generación final de una salida. Esta formación está inspirada en las redes neuronales biológicas del cerebro humano y poseen por tanto características similares, como el aprendizaje a través de la experiencia, la aplicación de aprendizajes anteriores a situaciones actuales (con pequeñas variaciones) o la abstracción de las caracteristicas principales de una serie de datos.

La estructura principal de una red neuronal es:

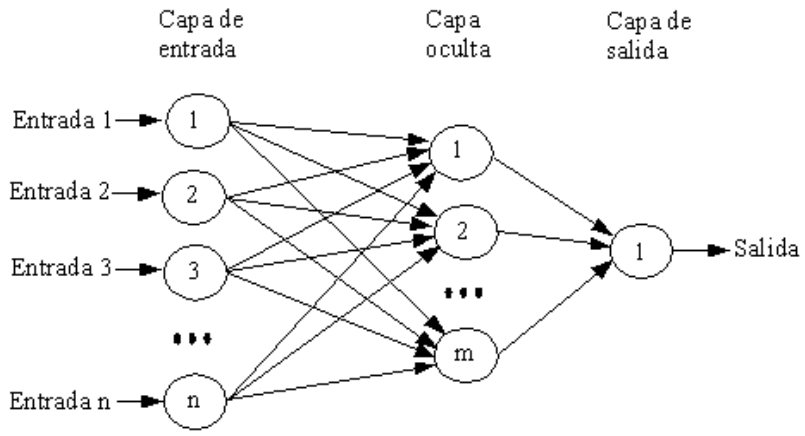


Figura 20: *Estructura básica de una Red Neuronal [31].*

donde cada capa está formada por elementos procesadores llamados *neuronas*. Una neurona es pues, una unidad procesadora simple que recibe y combina señales desde y hacia otras neuronas. Si la combinación de entradas es lo suficientemente fuerte, la salida se activa.
El interés de estas ANN's reside en la forma de conexión de las neuronas, éstas suelen organizarse en niveles o capas, estando 2 de ellas (entrada y salida) conectadas con el exterior y la/las que no lo están reciben el nombre de “capas ocultas”. Las capas de entrada y salida reciben y entregan, respectivamente, datos.

Las ANN's típicamente utilizadas para el reconocimiento de voz son los Perceptrones Multicapa (MLP's) que son redes multicapa que necesitan ser entrenadas y tienen alimentación “hacia delante”. Entre la capa de entrada y la de la salida, se añaden capas ocultas con capacidad de procesamiento pero sin conexión con el exterior. El algoritmo que se utiliza normalmente es Retropropagación (Back propagation), cuyo objetivo es obtener el valor de los pesos.

Los vectores de observación se insertan en la red a través de los nodos de la capa de entrada, pasan a la capa oculta y las combinaciones lineales entre los pesos y los vectores de entrada, se pasan por la función de activación. El resultado se obtiene en los nodos de la capa de salida, comparándolo con el resultado esperado. Con el error que se obtiene se realiza la “propagación hacia atrás”, modificando el valor de los pesos de la capa oculta hasta obtener un valor del error permitido.

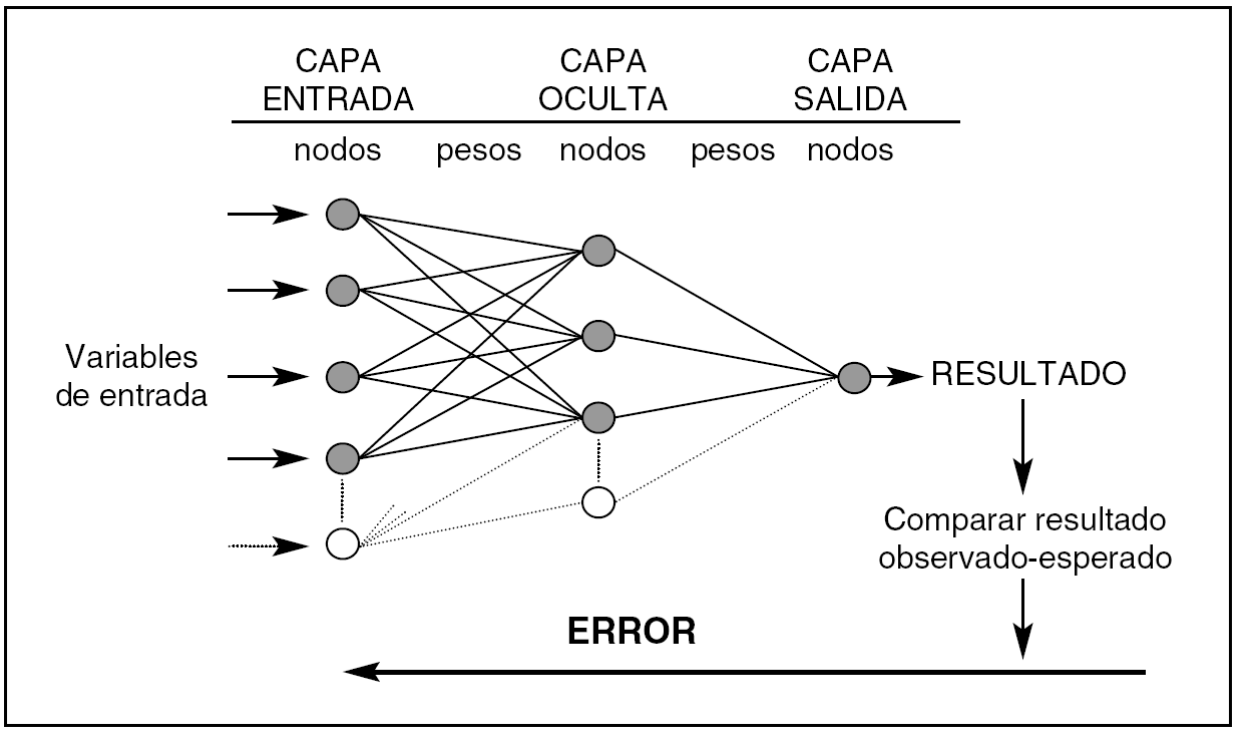


Figura 21: *Arquitectura de una red neuronal artificial (perceptrón de tres capas). Los pesos son los parámetros que deben ajustarse durante el proceso de entrenamiento para conseguir que la capa de salida de resultados coincida con los observados. El error entre lo observado y el resultado de la red se propaga hacia atrás (backpropagation) y debe ir disminuyendo en las sucesivas iteraciones en las que se presentan los valores de las variables predictoras. La complejidad de una red depende del número de nodos de su capa oculta [13].*

Ahora el MLP es ya un clasificador de caraterísticas. Los vectores de salida de la red son las probabilidades a posteriori de los fonemas que, a su vez, son las probabilidades de emisión de los HMM's.

5.3 Sistema de reconocimiento híbrido HMM/ANN

La razón principal por la cual las NN's no han obtenido éxito en reconocimineto de habla, es por su incapacidad para modelar la variabilidad del tiempo de las señales de voz de forma tan eficiente como los HMM, que permiten además construir fácilmente sistemas de miles de palabras.

Para superar todas estas dificultades comentadas, muchos investigadores han propuesto los llamados sistemas híbridos ANN-HMM. La idea básica es combinar en un solo sistema los HMM y las ANN para conseguir beneficiarnos de las propiedades de ambos : la habilidad de los HMM de modelar la variabilidad temporal de la señal de voz y la habilidad discriminativa de las ANN. Numerosos estudios demuestran que estos sistemas híbridos alcanzan unos resultados equivalentes o incluso mejores en algunas tareas que los sistemas HMM. También presentan un mejor comportamiento cuando la cantidad de datos de entrenamiento no es muy grande.

Para que las redes neuronales “aprendan” deben ser entrenadas mediante algoritmos diseñados para ello. En nuestro caso, entrenamos las redes con un conjunto de datos de entrenamiento etiquetados (all.align.ilab) utilizando un script “perl” (véase Anexo 2). Script 1stream_concatenacion.csh).

Inicialmente, las ANN se utilizaban en problemas sencillos de reconocimiento de habla, clasificando unidades como fonemas o palabras mediante mapeo temporal. Este tipo de clasificaciones para secuencias temporales no obtuvo éxito en el ámbito del reconocimiento continuo de habla. Sin embargo, su configuración como sistemas híbridos entre ANN y HMM sí que resulta ventajosa.

En particular, dadas las ecuaciones básicas de un HMM, las ANN se encargarían de calcular la $p(x_k|q_k)$, esto es, la probabilidad del vector de datos observado dado un estado de HMM. Afortunadamente, las ANN pueden calcular probabilidades e integrarse así en un enfoque basado en HMMs. Lo que se hace es entrenarlas para que produzcan las probabilidades a posteriori de un estado HMM, utilizando la regla de Bayes sobre las salidas de las redes, de la forma:

$$\frac{P(q_k|x_n,\theta)}{P(q_k)} = \frac{p(x_n|q_k,\theta)}{p(x_n|\theta)}$$

donde θ representa el conjunto de parámetros de ANN utilizado.

Las ventajas con las que cuentan las ANN's son las siguientes:

- Precisión del modelo: el cálculo de probabilidades con una ANN no necesita una hipótesis detallada sobre la forma de la distribución estadística a ser modelada, resultante de modelos acústicos más precisos.

- Sensibilidad al contexto: si varios vectores acústicos se utilizan como entrada de un MLP, la correlación de dichos vectores acústicos pueden tenerse en cuenta en la distribución de probabilidad. Esto proporciona un mecanismo sencillo para incorporar contexto acústico en la formulación estadística.
- Discriminación: una ANN puede soportar un entrenamiento discriminativo (de manera local y a nivel de trama)
- Uso más compacto de los parámetros, ya que todas las distribuciones se representan mediante el mismo conjunto (no como en el caso de los HMM en el que cada uno de ellos se entrena exclusivamente con el subconjunto de muestras de la clase que representa).
- Flexibilidad: permiten sencillas combinaciones de diversas características.
- El conocimiento de las probabilidades a posteriori permite una poda más eficiente para sistemas de reconocimiento de gran vocabulario.

5.4 Diseño de nuestro Reconocedor

La tarea principal que realizaremos será extraer las características articulatorias que hemos definido en la sección 4.2 a partir de los datos almacenados en una base de datos, para, una vez realizado el reconocimiento utilizando dichas características en lugar de los clásicos MFCC o PLP y obtenidos resultados, podamos valorar si se trata de un sistema que mejora las prestaciones.

Nuestro sistema cuenta con 2 tipos de datos, limpios y ruidosos que están almacenados en la base de datos con la que trabajaremos durante todo el PFC, véase la sección 6.2**.**

Para nuestro caso en particular nos basaremos en un sistema de referencia híbrido. Contaremos con una Red Neuronal Artificial tipo MLP que recibe como entradas coeficientes PLP de orden 12, cuenta con 800 unidades en la capa oculta y una salida con 28 fonemas (descritos en lineas inferiores).

Para entrenar nuestros MLP's utilizaremos la herramienta **QuickNet** (6.1). Cada modelo de palabra consiste en una sucesión de fonemas, cada uno de los cuales cuenta con una probabilidad de emisión calculada por la red MLP aplicado a las salidas de las redes neuronales. Se utiliza **Noway** (6.1) como decodificador de Viterbi.

Sobre este sistema de referencia será sobre el que iremos realizando toda serie de modificaciones hasta conseguir completar un sistema robusto que trabaje de forma eficiente.

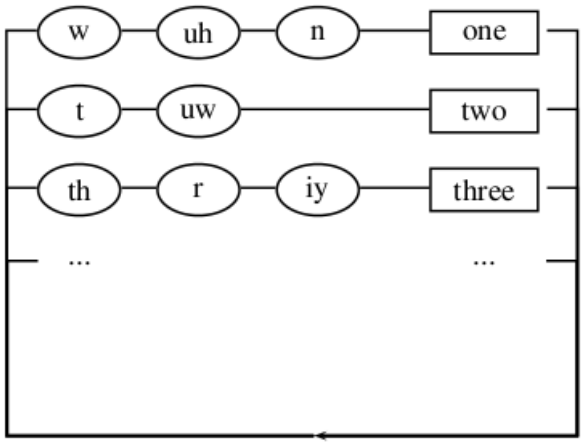


Figura 22: *Modelos de palabras creados por la sucesión de fonemas para el reconocimiento continuo de habla*

Para comenzar, hemos transformado nuestro sistema base de forma que trabaje con varias Redes Neuronales, una por cada característica articulatoria. Para ello definiremos un modelo de 7 AF's propuesto en Wester (2003) [20].

Lo primero será realizar un mapeo entre los fonemas y los valores de AF's, para lo que nos hemos basado en el modelo de Ladefoged (1982) [21] (nótese que la parte silenciosa de una oclusiva se mapea como “stop” y no como “silencio” para nuestros experimentos).

Tras el experimento de extracción de características, entrenamiento y testeo con 7 redes detallado en el apartado 6.4 (fase AF), continuaremos diseñando nuestro sistema añadiendo una segunda fase de redes neuronales en la que las salidas del sistema anterior de 7 redes serán las nuevas entradas (fase Conc).

Para ello se realizará la concatenación de los 7 vectores de salida que se habían generado, siendo la dimensión del nuevo vector de entrada la suma de los vectores de salida de las 7 NN anteriores. Este nuevo flujo de probabilidades que se obtiene es el que pasaré por el decodificador para el reconocimiento.

Ahora contamos, por tanto, con 2 sistemas de redes neuronales en serie, cuyo diagrama de bloques mostramos en la figura 23.

A continuación se muestra la tabla de mapeo:

Fonema/AF	***'manner'***	***'place'***	***'voice'***	***'high–low'***	***'fr–back'***	***'round'***	***'static'***
aa	vowel	nil	+voice	low	back	+round	static
ae	vowel	nil	+voice	low	front	-round	static
ah	vowel	nil	+voice	mid	central	-round	static
ax	vowel	nil	+voice	mid	central	-round	static
ay	vowel	nil	+voice	low	front	-round	dynamic
b	stop	bilabial	+voice	nil	nil	nil	dynamic
ch	fricative	alveolar	-voice	nil	nil	nil	dynamic
d	stop	alveolar	+voice	nil	nil	nil	dynamic
eh	vowel	nil	+voice	mid	front	-round	static
ey	vowel	nil	+voice	mid	front	-round	dynamic
f	fricative	labiodental	-voice	nil	nil	nil	static
iy	vowel	nil	+voice	high	front	-round	dynamic
jh	fricative	alveolar	+voice	nil	nil	nil	dynamic
k	stop	velar	-voice	nil	nil	nil	dynamic
l	approx.	alveolar	+voice	nil	nil	nil	dynamic
m	nasal	bilabial	+voice	nil	nil	nil	static
n	nasal	alveolar	+voice	nil	nil	nil	static
ow	vowel	nil	+voice	mid	back	+round	dynamic
p	stop	bilabial	-voice	nil	nil	nil	dynamic
r	retroflex	alveolar	+voice	nil	nil	nil	dynamic
s	fricative	alveolar	-voice	nil	nil	nil	static
t	stop	alveolar	-voice	nil	nil	nil	dynamic
uw	vowel	nil	+voice	high	back	+round	dynamic
v	fricative	labiodental	+voice	nil	nil	nil	static
w	approx.	velar	+voice	nil	nil	nil	dynamic
y	approx.	velar	+voice	nil	nil	nil	dynamic
z	fricative	alveolar	+voice	nil	nil	nil	static

Tabla 4: *Mapeo fonema-valor articulatorio [21].*

El diagrama de bloques del sistema en serie de redes neuronales es el siguiente:

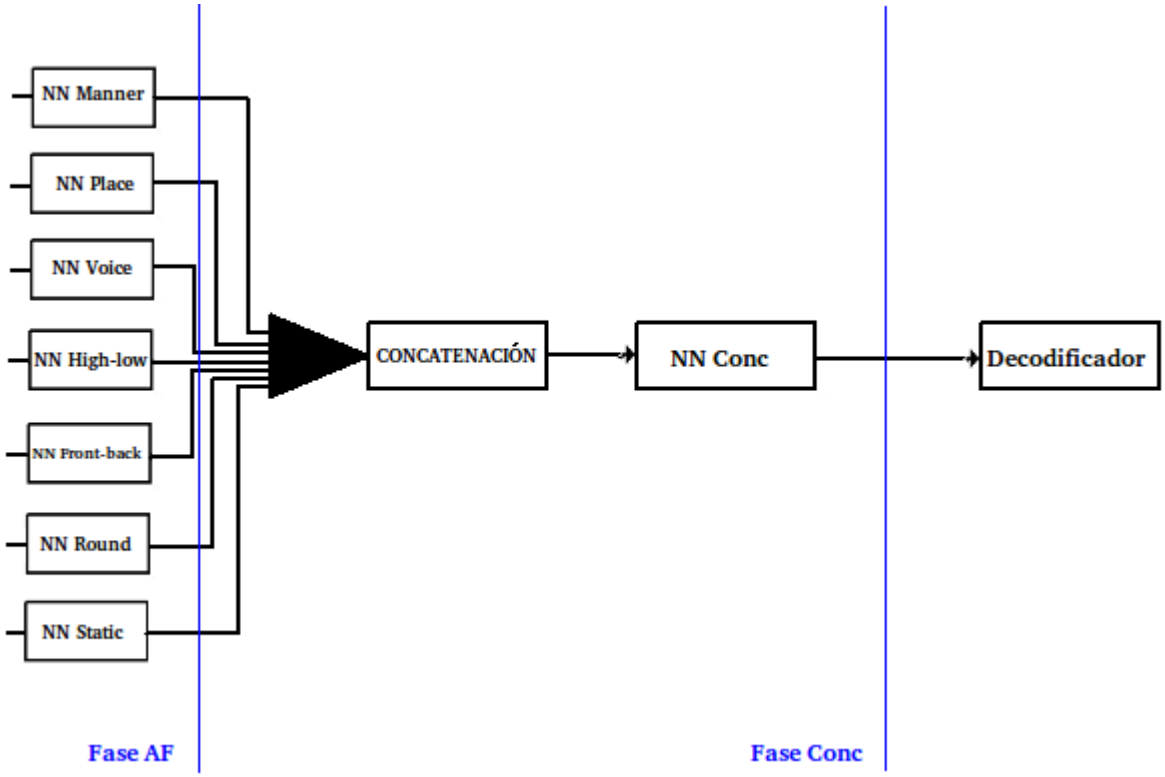


Figura 23: *Sistema de 2 redes neuronales en serie a las que denominaremos Fase AF y Fase Conc, respectivamente.*

Diferenciamos entre fase AF y fase Conc para poder referirnos a cada una de ellas de aquí en adelante de manera clara y sencilla.

Capítulo 6. Pruebas y resultados.

6.1 Marco Experimental

Para la ultilización de las herramientas con las que hemos trabajado en este PFC era necesario, en primer lugar, disponer de un sistema operativo Unix, nosotros hemos elegido la versión 11.4 de Ubuntu .

El método utilizado para la implementación del proyecto está basado en el sistema de pruebas “Isolet Testbed” (más información en el anexo 1). Sistema de pruebas ISOLET Testbed), trata de un modelo de implementación de un sistema ASR que constituirá el sistema de referencia de nuestro PFC.

Como complemento a este sistema de pruebas se ha utilizado el paquete de herramientas “SPRACHcore_27_2_09”, “Quicknet3” y las librerías “dpwelib”, que facilitam el uso de MLP's en sistemas de reconocimiento con patrones estadísticos. A su vez, el paquete “SPRACHcore_27_2_09” contiene las herramientas necesarias para la extracción de características , la introducción de parámetros en los perceptrones multicapa o la decodificación (feacalc, feacat, noway y pfile_utils).

El desarrollo de Isolet Testbed así como el paquete de herramientas SPRACHCore y Quicknet lo lleva a cabo el ICSI Speech Group [30].

6.2 Base de datos

La base de datos que recoge las muestras con las que trabajaremos en nuestro PFC es **Isolet** [27].

Se trata de una base de datos que recoge 7800 muestras vocales representadas por 150 hablantes que pronuncian las letras del alfabeto americano 2 veces.

Contamos con 2 versiones de la base de datos, limpia y ruidosa, que nos permitirán conocer las prestaciones de nuestro reconocedor en 2 tipos de entorno, el primero de ellos ideal y el segundo más cercano a un entorno real. Para conseguir la representación de un entorno lo más similar posible al real, la base de datos ha sido contaminada con diferentes tipos de ruido, obtenidos de la colección RSG-10 ([29]).

Para mejorar la significatividad estadística de los experimentos se implementa un procedimiento de LOO (Leave One Out) en el que los datos de ISOLET se dividen en 5 subconjuntos, cada uno con 1560 frases pronunciadas por 30 hablantes, 15 masculinos y 15 femeninos, de entre 14 y 72 años.

El agrupamiento para cada subconjunto es aleatorio y se almacenan en los archivos “clean.wav.files.rand” y “noisy.wav.files.rand” para facilitar que los experimentos sean replicables.

Los tipos de ruido utilizados para los subconjuntos “noisy” de Isolet fueron “Speech babble”, “Factory floor noise 2” y “Car interior noise”, aparte de algunos otros descritos en [28].

6.3 Herramientas para el análisis de resultados

En nuestro PFC utilizaremos como herramientas para el análisis de resultados las matrices de confusión y los triángulos entrópicos, ya que ambas resultan muy útiles en cuanto a la visualización de los mismos.

La **matriz de confusión** puede definirse como un registro total de las decisiones de un clasificador. En la tarea de clasificación, dados los conjuntos de entrada y salida $V_x = \{x_i\}$ y $V_y = \{y_j\}$ respectivamente, se presenta al clasificador un patrón de la clase de entrada x_i para obtener el identificador de la clase de salida y_j.
El clasificador se prueba sobre N experimentos para obtener una matriz N_{XY}, donde se representa el número de veces que ocurre que $X = x_i$, $Y = y_j$. Es ésta la matriz de confusión, también denominada tabla de contingencia [39].

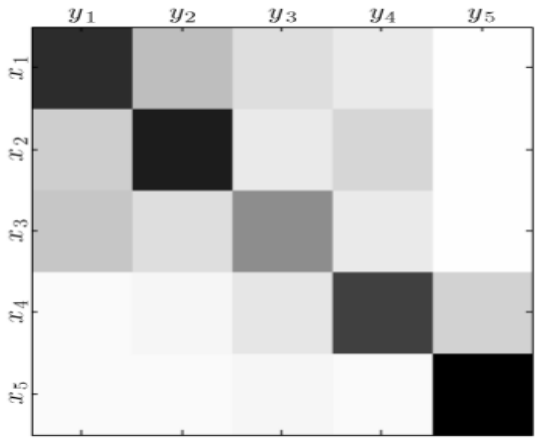


Figura 24: *Ejemplo de matriz de confusión. La diagonal en tonos más oscuros significa que el clasificador actúa con bastante precisión [39].*

Una medida a menudo utilizada para medir el rendimiento de un reconocedor es la *precisión (ing. accuracy)*, que se refiere a la proporción de veces que el clasificador toma la decisión correcta.

$$A(N_{XY}) \approx \Sigma_i N_{XY}(x_i, y_i) / N$$

Una mayor precisión en las matrices de confusión se traduce en un oscurecimiento en el tono de la misma en la representación de mapas de calor (heatmaps) de la figura 24. Es por eso que un buen resultado será aquel que muestre una diagonal en tonos más oscuros que el resto de la matriz, esto significará que ha sucedido muchas veces que $x_i = y_i$ y el reconocedor "ha acertado".

El **triángulo entrópico** es una herramienta que, a partir de los datos de entropía que generan las matrices de confusión, analiza el comportamiento de los clasificadores multiclase.
En primer lugar, se obtiene la ecuación de equilibrio de las entropías que contienen propiedades del clasificador, posteriormente, se normaliza dicha ecuación y se obtiene un símplex en un espacio entrópico tridimensional, a partir del cual se genera el triángulo entrópico o diagrama entrópico de Finetti.
Gráficamente puede verse más claro mediante los diagramas de entropía:

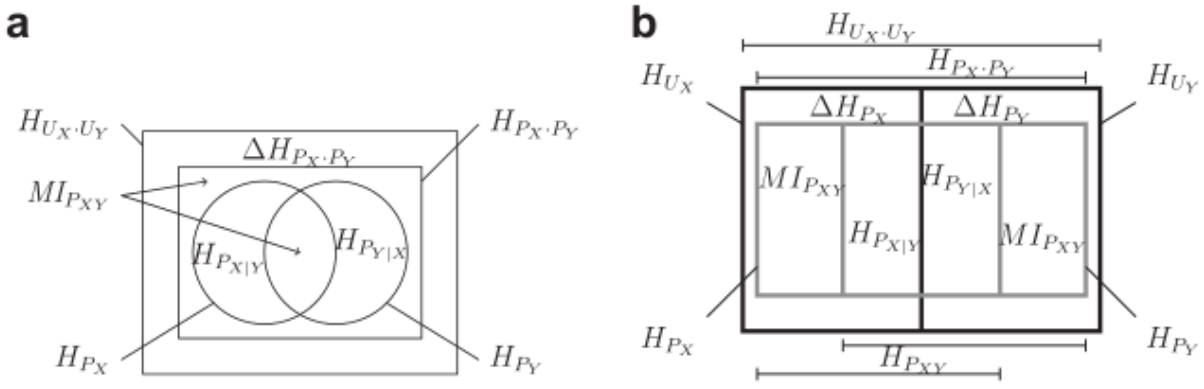


Figura 25: *Diagramas de información extendida de la entropía relativa en una distribución bivariante, siendo a) diagrama convencional y b) diagrama representando las entropías por separado [38].*

Donde para el primer diagrama (**a**) podemos distinguir cada entopía por separado H_{Px} y H_{py}, así como la intersección de ambas MI_{Pxy} (o Información mutua) y la entropía de las condicionales $H_{Px|y}$ y $H_{Py|x}$. Además $\Delta H_{Px \cdot Py}$ representa la diferencia entre las áreas de los rectángulos extreno ($H_{Ux \cdot Uy}$) e interno ($H_{Px \cdot Py}$), respectivamente.

Gracias al segundo diagrama (**b**) se pueden desarrollar las ecuaciones de equilibrio para cada variable por separado, de la forma:

$$H_{Ux} = \Delta H_{Px} + MI_{Pxy} + H_{px|y}$$
$$H_{Uy} = \Delta H_{Py} + MI_{Pxy} + H_{Py|x}$$

Ahora se pueden representar los datos de entrada y salida en el mismo triángulo entrópico teniendo en cuenta el siguiente esquema:

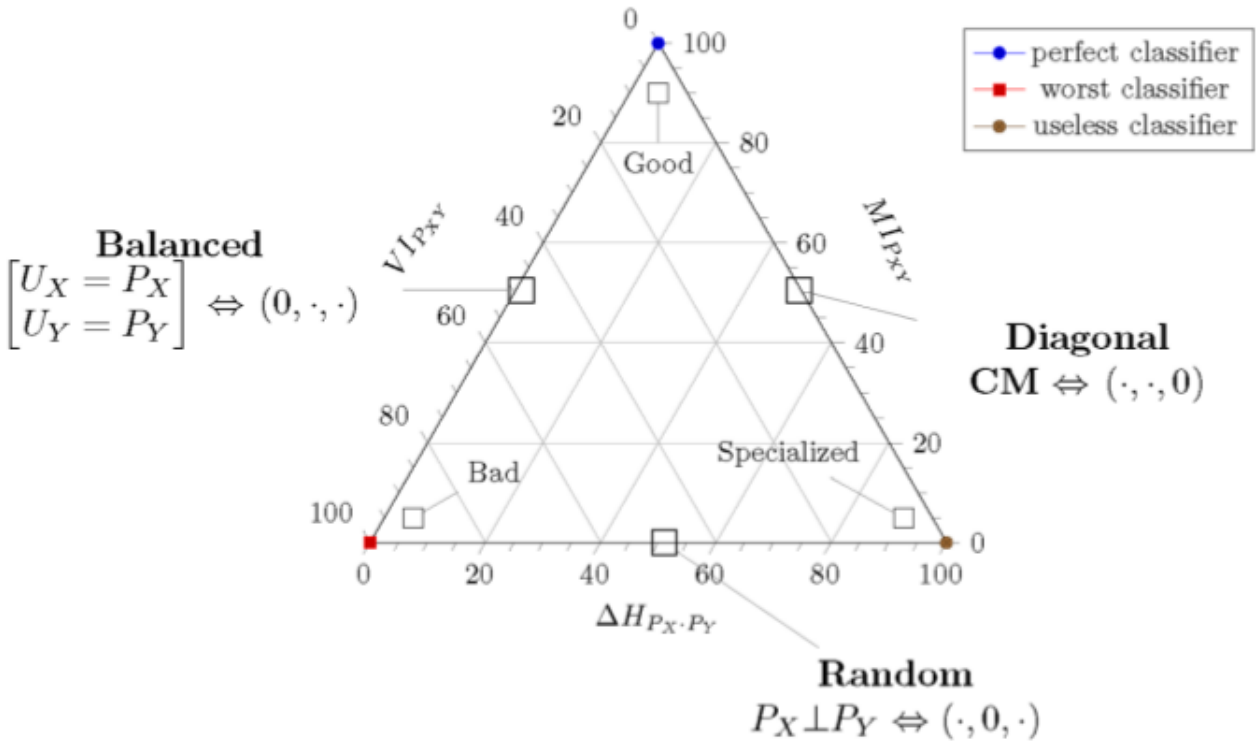


Figura 26: *Esquema de las zonas interpretables de un triángulo entrópico [39].*

La posición de los datos obtenidos en el experimento determinará la eficacia del reconocedor. Si los datos se encuentran en la zona superior y cerca del márgen izquierdo, estaremos ante unos buenos resultados. Si por el contrario los datos están más centrados en la zona del vértice inferior izquierdo los datos no serán muy satisfactorios. La zona del margen inferior derecho recoge los datos especializados y en el centro se acumularían datos clasificados aleatoriamente, algo que tampoco sería óptimo [38].

6.4 Pruebas y experimentos

Contamos en un principio con el script *1stream.csh*, sobre el que trabajaré durante todo el PFC y que terminará por convertirse en *1stream_concatenacion* (véase el Anexo 2). Script 1stream_concatenacion.csh). Realizamos en primer lugar algunas modificaciones para adaptarlo a mi estructura de directorios. Este script realiza todos los procesos de un sistema ASR (extracción de características mediante coeficientes PLP, entrenamiento con parámetros etiquetados, decodificación y posterior testeo de los resultados).

En un primer momento hemos ejecutado este **sistema de referencia** sin ningún tipo de modificación en el diseño, es decir, 2 NN's sencillas, una para cada tipo de datos, limpios y ruidosos que llamaremos “clean” y “noisy”. Una vez obtenido los resultados del mismo, que serán detallados más adelante, se ha comenzado con el diseño y obtención de resultados del sistema definido en 5.4, en el cual nos centraremos en este apartado.

Es importante aclarar en primer lugar, que en la base de datos Isolet con la que trabajamos en el PFC 6.2, las frases representan una sola palabra, que a su vez es la pronunciación de una letra, es decir, estamos llamando frase realmente a la letra pronunciada. Dependiendo de la duración de la frase, tienen más o menos tramas, por ejemplo, las duraciones en número de tramas para los siguientes ejemplos de la **e** y la **z** serían 73 en ambos casos y la secuencia de etiquetas se muestra a continuación, donde “0” codifica el silencio, “5” codifica /e/ y “3” codifica /z/:

Frase “e”:
Columns 1 through 15

0 0 0 0 0 5 5 5 5 5 5 5 5 5 5

Columns 16 through 30

5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

Columns 31 through 45

5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

Columns 46 through 60

5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

Columns 61 through 73

5 5 5 5 5 5 5 0 0 0 0 0 0

Frase “z”:
Columns 1 through 15

0 0 0 0 0 0 3 3 3 3 3 3 3 3 3

Columns 16 through 30

3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

Columns 31 through 45

3 5 5 5 5 5 5 5 5 5 5 5 5 5 5

Columns 46 through 60

5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

Columns 61 through 73

5 5 5 5 5 5 5 5 5 0 0 0 0

Además, todas las pronunciaciones de consonantes incluyen vocales, es por eso que hay muchas más vocales. Las listas de las frases están almacenadas en los ficheros “clean.wav.files.rand” y “noisy.wav.files.rand”.

De forma detallada, se han ejecutado los siguientes pasos (de la misma manera para los 2 tipos de datos, “clean” y “nosiy”):

El primero de ellos ha sido la **extracción de características** mediante coeficientes PLP de habla, con las herramientas “feacalc” y “feacat”, almacenándolas en formatos de características *pfile.* Posteriormente se ha realizado una normalización de los ficheros obtenidos.

A continuación hemos comenzado con los **entrenamientos de las redes,** en nuestro caso del sistema de 7 NN's definido en 5.4. En este punto comienzan las diferencias y modificaciones con respecto al sistema de referencia con el que contabamos en un primer momento.

Se crean 7 redes neuronales, 1 por cada característica articulatoria descrita en la figura 10 para lo cual, lo primero fue realizar un mapeo como se explica en la sección 5.4, en el cual se asignaron los valores de cada característica articulatoria para cada fonema. Creamos también etiquetas nuevas para cada nueva NN.

Para la generación de las etiquetas fue necesario leer el archivo *all.align.ascii* que contiene 3 columnas cuyos valores son, respectivamente, nº de frase - nº de trama - etiqueta. Realizamos un pequeño programa en Matlab (véase el anexo 3.) Etiquetas.m) con el que se generan las

etiquetas para las 7 NN**.** Además, para cada NN, el número de neuronas de la capa de salida deberá coincidir necesariamente con el nº de clases, es decir:

manner ==> 7 neuronas en la capa de salida
place ==> 7 neuronas
voice ==> 3 neuronas
high-low ==> 5 neuronas
front-back ==> 5 neuronas
round ==> 4 neuronas
static ==> 3 neuronas

Se realiza un entrenamiento para cada una de las 7 redes de, a su vez, cada uno de los 5 subconjuntos creados según lo explicado en 6.2, obteniendo los valores de precisión global, precisión de conjunto de validación cruzada, velocidad, etc. Los resultados se han almacenado en archivos de tipo *foldX.log***.**

Obtenemos 7 salidas, que son las probabilidades a posteriori de cada característica articulatoria.

El objetivo ahora será la creación de una nueva NN cuyas entradas sean estas 7 salidas concatenadas. La probabilidad a posteriori que nos devuelve esta segunda NN será la que pasemos al decodificador para el reconocimiento final.

En esta nueva NN retomamos las etiquetas originales *all.align.ilab* y utilizaremos como entrada la concatenación de los 7 vectores de salida de la red anterior.

Gráficamente se ve mucha más claro:

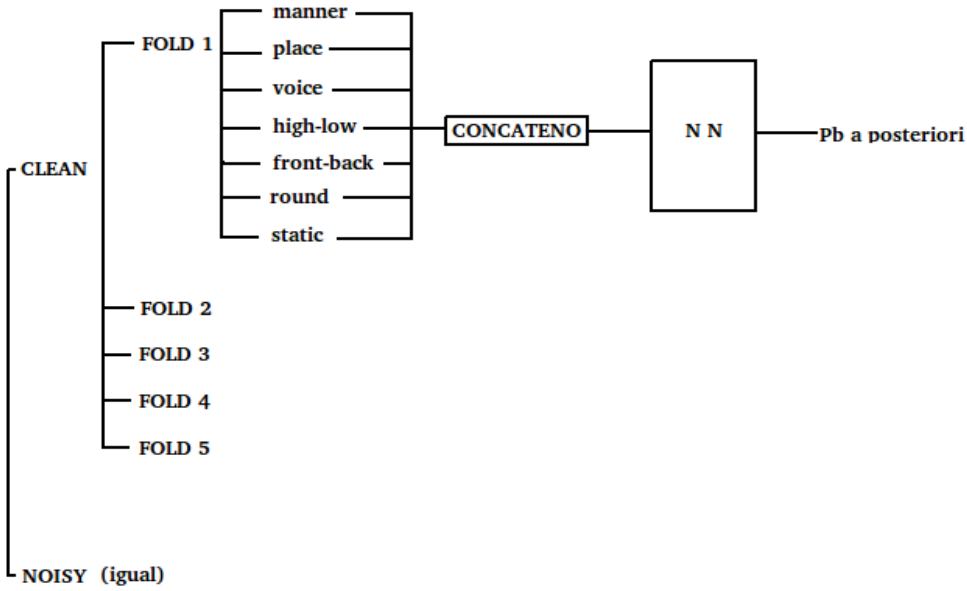


Figura 27: *Esquema del diseño del reconocedor*

El enfoque que daremos a esta nueva parte será el siguiente:

Teniendo en cuenta que cada “fold” es una partición del conjunto de datos Isolet, trataremos cada una de estas particiones como un experimento independiente.

Tendremos entonces 7NN por cada experimento, con lo cual:

(5 folds para CLEAN + 5 folds para NOISY) x 7NN = 70 NNs

es la cantidad de NN's que tengo antes de realizar la concatenación. Como concatenamos conjuntos de 7 salidas, obtendremos un total de 10 nuevas redes neuronales.

Para **concatenar** trabajamos con la herramienta “fwd”, modificando algunos parámetros para adaptarla a nuestras necesidades. La principal modificación ha sido el cambio de trabajar con feaType = plpD2 a trabajar con **art_feaType = manner, place, voice, high-low, front-back, round y static**. En nuestro caso, el fichero de entrada es el mismpo para las 7 NN.

Se han modificado también los archivos para que admitan números de neuronas en la capa de salida distintos para cada art_feaType (tipo de característica articulatoria), dependiendo de los valores que tome cada una de ellas.

Una vez concatenadas las salidas y generados los 10 archivos de salida pfile, concatenaremos los 5 correspondientes a “clean” y a “noisy”, para la generación de 2 únicos archivos *pfile* que actuarán como entradas a cada una de las 5 “folds”. Utilizaremos en este caso la herramienta “pfile_concat”.

A continuación se realiza la parte de **test**, que genera los archivos *.hyps* que contienen las letras reconocidas, *.score* donde se guardarán las matrices de confusión y *best tune.out* que me devuelve el valor de los parámetros que dan el mínimo WER (tasa de error de palabra).

De forma resumida, en el archivo *.hyps* se refleja lo que el reconocedor ha decidido que se dice en cada frase de la lista *clean.part1.wav.files.rand.* Este resultado habrá que compararlo con lo que verdaderamente se ha dicho. En algunos casos los resultados coinciden y en otros no.

Score es la herramienta que hace esta comparación. Para ello busca en el nombre del fichero de la lista, que es de la forma “fdcfo-Y1-t.sph” la letra que se ha pronunciado, siendo la transcripción para este caso, por ejemplo, la letra Y (antepenúltimo valor antes del sufijo .sph).

Tras realizar la comparación se generan las matrices de confusión con los resultados que determinan cual es la calidad de nuestro reconocedor. Al haber generado 2 redes en serie, contamos con 4 posibles salidas que definimos como:

Gráficamente resulta más sencillo de comprender:

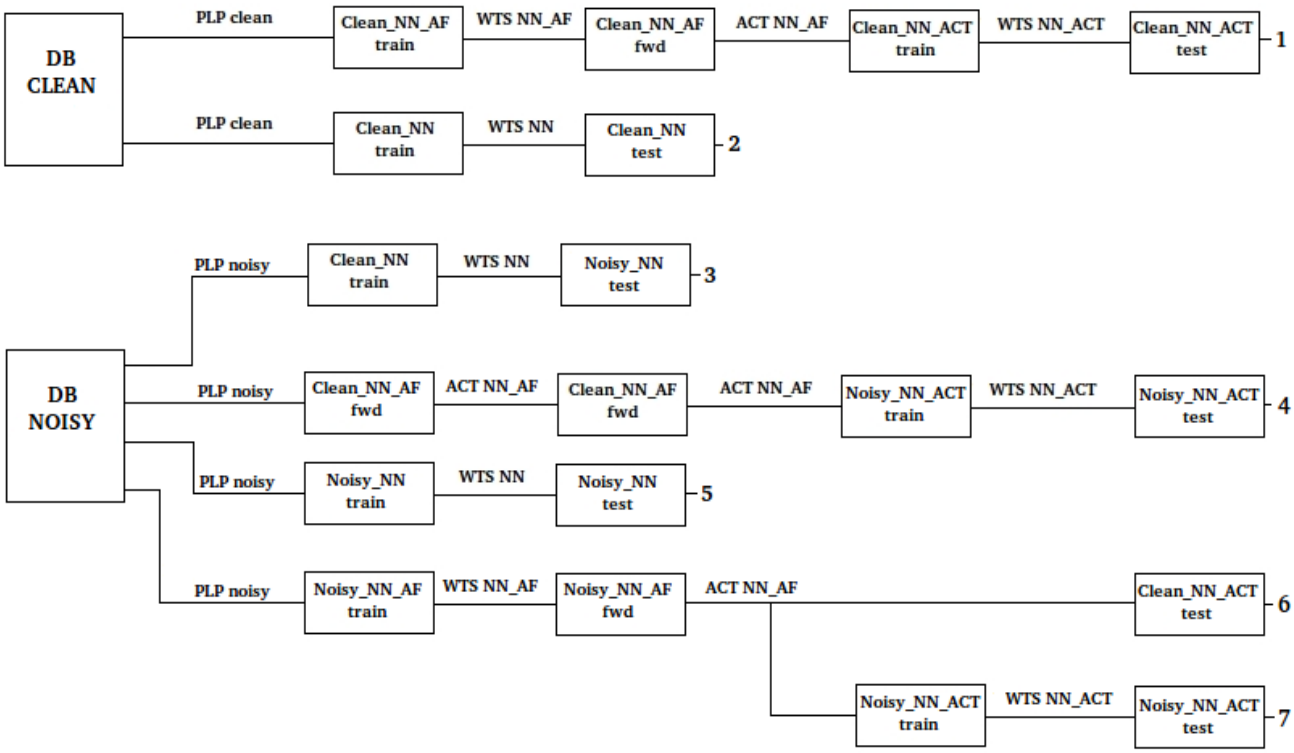


Figura 28: *Esquema del sistema de redes creado para nuestro ASR. WTS denota los parámetros de las redes neuronales (WeighTS o pesos) y ACT (activations) las salidas blandas de la redes neuronales o estimaciones de las probabilidades a posteriori de las clases de salida, dadas las entradas. NN son las redes neuronales del sistema de referencia cuyas entradas son los PLP y las salidas, las clases fonéticas. NN_AF son las redes neuronales cuyas entradas son PLP y las salidas las clases articulatorias y NN_ACT las que toman como entrada la concatenación de las activaciones de las NN_AF y devuelven las probabilidades a posteriori de las clases fonéticas.*

Siendo 2, 3 y 5 las salidas del experimento de Referencia (clean-clean, clean-noisy y noisy-noisy) y el resto 1, 4, 6 y 7 las salidas del nuevo sistema con 2 fases (AF y Conc) en serie (clean-cleanx10-clean-cleanx10, clean-cleanx10-clean-noisyx10, clean-cleanx10-noisy-noisyx10, noisy-noisyx10-noisy-noisyx10, respectivamente).

Es importante aclarar que en este gráfico, los bloques definidos como “ACT” se refieren a la fase Conc, es decir, aquella en la que cada entrada es la concatenación de las salidas de la fase anterior.

Para identificar los nombres de las salidas con el proceso realizado, incluimos las siguientes tablas:

Salida	Entrenamiento	Test
clean-clean	clean	clean
clean-noisy	clean	noisy
noisy-noisy	noisy	noisy

Tabla 5: *Tipo de entrenamiento y test para el experimento de referencia.*

Salida	Entrenamiento fase AF	Entrenamiento fase Conc	Test
clean-cleanx10-clean-cleanx10	clean	clean	clean
clean-cleanx10-clean-noisyx10	clean	clean	noisy
clean-cleanx10-noisy-noisyx10	clean	noisy	noisy
noisy-noisyx10-noisy-noisyx10	noisy	noisy	noisy

Tabla 6: *Tipo de entrenamiento y test para cada salida del Reconocedor.*

El experimento de referencia es bastante sencillo ya que se trata del sistema de 1 NN sometido a un entrenamiento y posterior test. Las salidas son, pues, directas (salidas 2, 3 y 5). Sin embargo, el resto de experimentos son algo más complicados debido al sistema diseñado de 2 fases en serie. En este caso nos encontramos ante un sistema en el que, al haber 2 fases de NN deben por tanto realizarse 2 entrenamientos (uno para cada fase). La tabla 5 describe para cada uno de los casos, qué se hace en cada fase del sistema (entrenamiento o test y tipo de cada uno).

El siguiente paso ha sido **automatizar los resultados** que obtenemos de las matrices de confusión y de los entrenamientos para poder comparar los experimentos entre ellos y visualizar, de manera más rápida las diferencias entre unos y otros. Nos serviremos también de las gráficas que obtenemos de las matrices de confusión para realizar las comparaciones.

A partir de aquí comenzaré a hacer diversos **experimentos** para obtener resultados definitivos de la precisión de mi reconocedor. Iré modificando los parámetros para sacar conclusiones finales. Los experimentos que se han llevado a cabo han sido los siguientes:

Experimento inicial: el experimento de referencia (Baseline), cuenta con 800 unidades en la capa oculta que reciben las entradas por medio de coeficientes PLP de orden 12. El enventanado se realiza con ventanas tipo Hamming de 25ms, desplazamiento de 10 ms y un contexto de 5 tramas. La salida cuenta con 28 fonemas.

En un principio, definimos un sistema de 7 Redes neuronales en el que hemos ido jugando con las neuronas de la capa oculta y con el contexto, así, hemos realizado diferentes experimentos en esta línea:

- Modificaciones en neuronas de la capa oculta de la NN:

Experimento 1: 7 NN con nhidden 800 y cw5
Experimento 2: 7 NN con nhidden 100 y cw5
Experimento 3: 7 NN con nhidden 450 y cw5

- modificaciones en el contexto:

Experimento 4: 7 NN con nhidden 800 y cw3
Experimento 5: 7 NN con nhidden 800 y cw7

donde nhidden es el parámetro que indica el número de neuronas de la capa oculta y cw el tamaño del contexto en número de tramas.

En un principio, se han utilizado los mismos valores de neuronas en la capa oculta para todas las redes dentro del experimento, sin embargo, a medida que se han ido viendo resultados, nos pareció que sería óptimo utilizar más o menos neuronas dependiendo de si se trata de una red más simple o más compleja. En nuestro caso, el primero de los sistemas de NN (Fase AF, con 7 redes simples) es más sencillo y por tanto necesitará un número bajo de neuronas, sin embargo el segundo de los sistemas (fase Conc) es más complejo y sería interesante utilizar un número más elevado de neuronas para procesarlo. Nos moveremos en este campo haciendo diferentes pruebas:

Experimento 6: 7 NN con nhidden_af 100, nhidden_conc 800 y cw5
Experimento 7: 7 NN con nhidden_af 100, nhidden_conc 1200 y cw5
Experimento 8: 7 NN con nhidden_af 450, nhidden_conc 800 y cw5
Experimento 9: 7 NN con nhidden_af 450, nhidden_conc 1200 y cw5

donde ahora nhidden_af se refiere al número de neuronas de la capa oculta de la fase AF y nhidden_conc al número de neuronas de la fase Conc.

A partir de estos nuevos experimentos, se observa que existe un problema que tiene que ver con la nasalidad. Diversos estudios han tratado este caso [18] y parece una buena opción aislar este valor de su red neuronal inicial (manner) y tratarlo como una red independiente.(nasality), contando por tanto ahora con 8 NNs.

Esta nueva red “Nasality” contará con los valores: 0 – silence
1 – nasal
2 – no nasal

Experimento 10: 8 NN con nhidden 800 y cw5

Existen algunas complicaciones ya que la nasalidad es una característica que puede co-existir con otras (cuando se producen nasalizaciones de otros fonemas) y no disponemos de etiquetas para identificar y entrenar este hecho. A la hora de describir esta nueva NN para nasalidad (a la que hemos denominado **nasality**), hemos tenido que decidir para todos los fonemas que tenemos si son nasales o no. Simplistamente sólo /m/ y /n/ deberían serlo, pero hay otros fonemas que podrían considerarse nasalizados, aunque no en todos los contextos. Para esto parece claro que m y n en inglés deberían ser "stops". Respecto al detector de nasalidad nos vamos a tener que conformar con marcar /n/ y /m/ como nasales y el resto como **no** nasales ya que no tenemos marcas de nasalización. Esto podría ser problemático porque las nasales son una porción muy pequeña del total y además existen fonemas que se pueden nasalizar (ejemplo:francés). Para solucionarlo, se realiza un segundo experimento de este tipo en el que añadiremos el parámetro de ***qnstrn,*** r*eject_last,* que elimina muestras del entrenamiento marcándolas como la etiqueta última. Así, las muestras que no sepamos si están o no nasalizadas (como las vocales, los silencios o los aproximantes) se sacan del entrenamiento, quedando éste más equilibrado.

Experimento 11: 8 NN con reject_last, nhidden 800 y cw5

Vistos estos resultados, se han realizado una segunda serie de experimentos utilizando los parámetros con los que se han obtenido las mejores prestaciones en los hasta ahora realizados.
Se ha comprobado por ejemplo, que las 8 NN son bastantes eficientes y que además, utilizar un número mayor de neuronas en la red compleja que en las sencillas también mejora los resultados, por lo que nos moveremos en esta dirección.

Experimento 12: 8 NN con nhidden_af 450, nhidden_conc 1200 y cw5
Experimento 13: 8 NN con rejec_last, nhidden_af 450, nhidden_conc 1200 y cw5
Experimento 14: 8 NN con nhidden_af 450, nhidden_conc 1500 y cw5
Experimento 15: 8 NN con nhidden_af 450, nhidden_conc 2000 y cw5
Experimento 16: 8 NN con nhidden_af 450, nhidden_conc 1500 y cw7

6.5 Resultados y análisis

En nuestro PFC utilizaremos como herramientas de análisis de resultados las matrices de confusión y los triángulos entrópicos, ya que ambas resultan muy gráficas en cuanto a la visualización de los mismos.

Para comenzar con el estudio de los resultados, analizaremos en primer lugar aquellos referentes a las características articulatorias, observando cada una de ellas por separado. Para ello, dado que tenemos una gran cantidad de experimentos, tomaremos como referencia aquellos con los que hemos obtenido un mejor y peor resultado, para visualizar mejor las diferencias entre ambos y poder así sacar mejores conclusiones.

Para poder identificarlos en las gráficas, deberemos tener en cuenta cómo hemos definido cada red neuronal en relación a los posibles valores o clases que la representan:

AF/Clase	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Manner	silence	approximant	retroflex	fricative	stop	vowel	-
Place	silence	bilabial	labiodental	dental	alveolar	velar	nil
Voice	silence	+voice	-voice	-	-	-	-
High-low	silence	high	mid	low	nil	-	-
Front-back	silence	front	central	back	nil	-	-
Round	silence	+round	-round	nil	-	-	-
Static	silence	static	dynamic	-	-	-	-
Nasality	silence	nasality	no nasality	-	-	-	-

***Tabla 7**: Las 8 redes neuronales y sus clases de salida.*

Tomando como referencia los datos obtenidos en uno de los experiementos que proporcionan un mejor resultado global, el **experimento 14** (8 NN con nhidden af 450, nhidden conc 1500 y cw5) tenemos para cada una de las 8 AF's las siguientes matrices de confusión:

a) Salidas de la NN articulatoria con datos **clean (limpios)**:

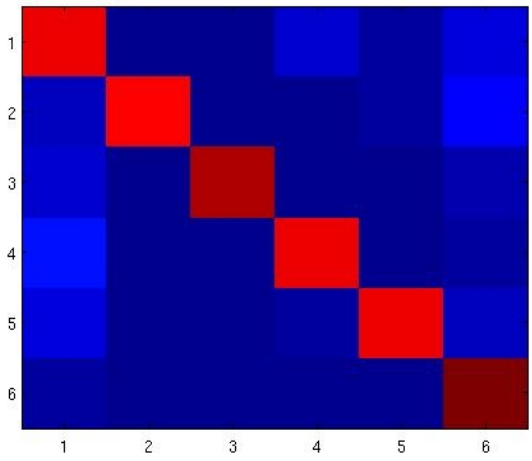


***Figura 29:** Manner*

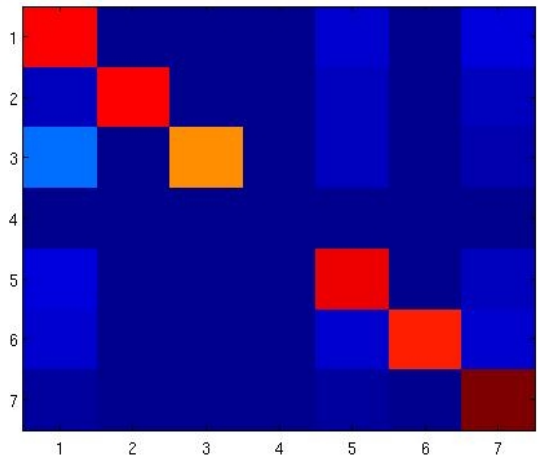


***Figura 30:** Place*

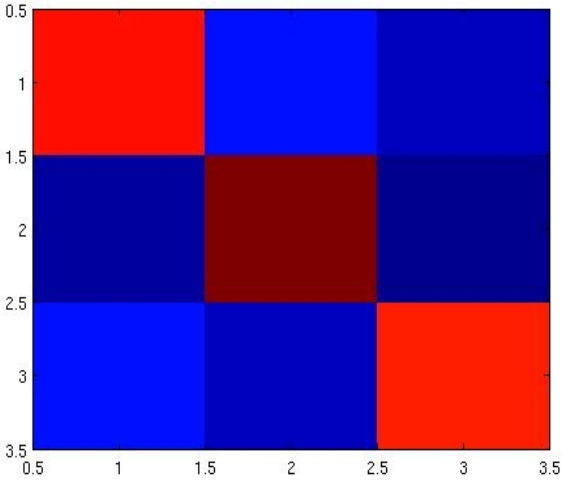


Figura 31: *Voice*

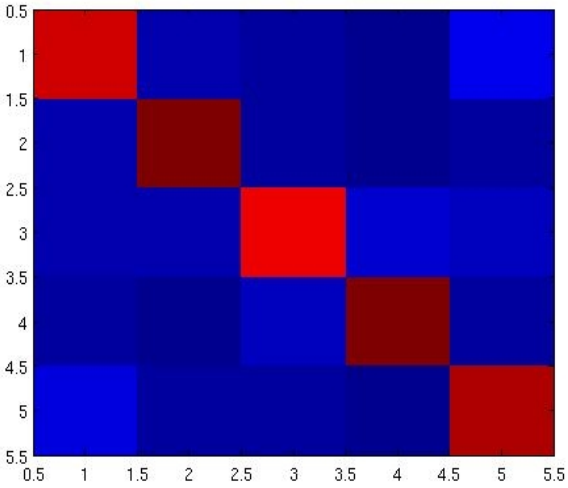


Figura 32: *High-low*

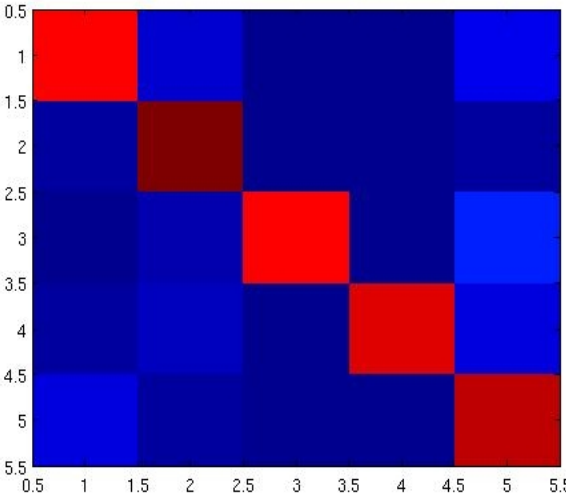


Figura 33: *Front-back*

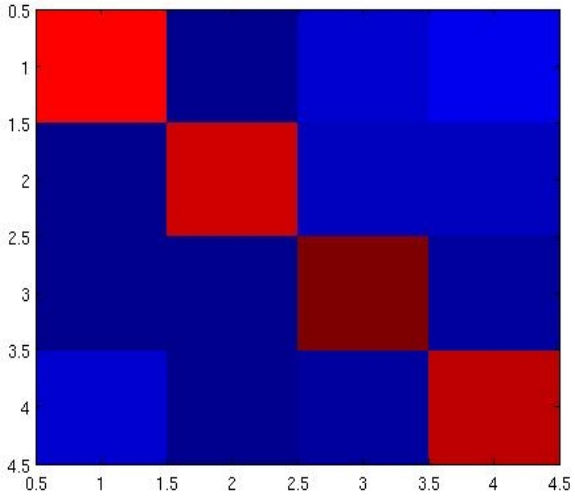


Figura 34: *Round*

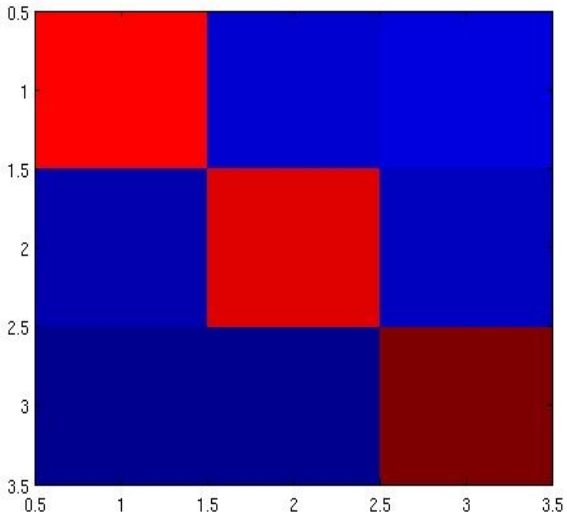


Figura 35: *Static*

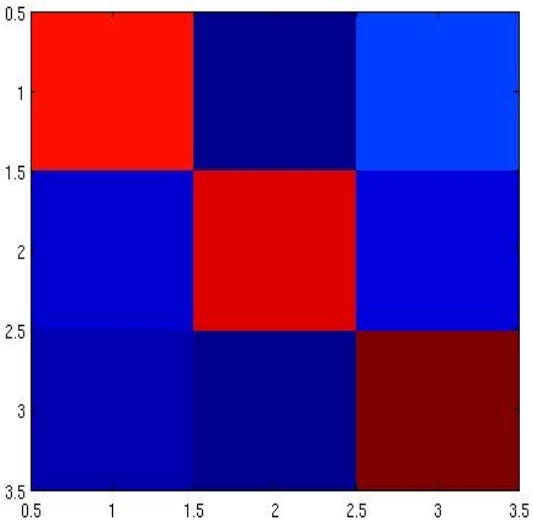


Figura 36: *Nasality*

Las matrices de confusión nos permiten conocer, valga la redundancia, las confusiones individuales entre las clases en un reconocedor. Para ello, debemos interpretarlas sabiendo que en este tipo de representación (mapa de calor), el nivel máximo de intensidad de color se corresponde con la máxima confusión. Una matriz de buenas prestaciones será entonces aquella en la que se observe, claramente, una diagonal de colores cálidos.

Analizándolas, vemos que se genera en casi todas ellas dicha diagonal, que se traduce como una precisión elevada. Esto es normal y esperable ya que se trata de datos limpios, tratados en un entorno ideal.

Resulta llamativa la gráfica de la red “Place”, que discrimina con muy baja precisión el valor “labiodental”, a parte de confundirlo con “silencio”. Podríamos, en estudios futuros, analizar en profundidad dicho valor de la característica, al igual que hemos hecho con “nasality” para averiguar cuál está siendo el motivo.

El valor dental aparece vacío ya que, como podemos comprobar en el mapeo, ninguno de los fonemas toma dicho valor.

b) Salidas de la NN articulatoria con datos **noisy (ruidosos)**:

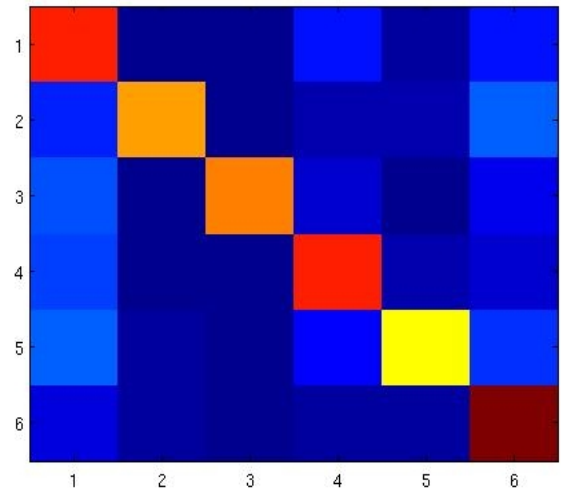


Figura 37: *Manner*

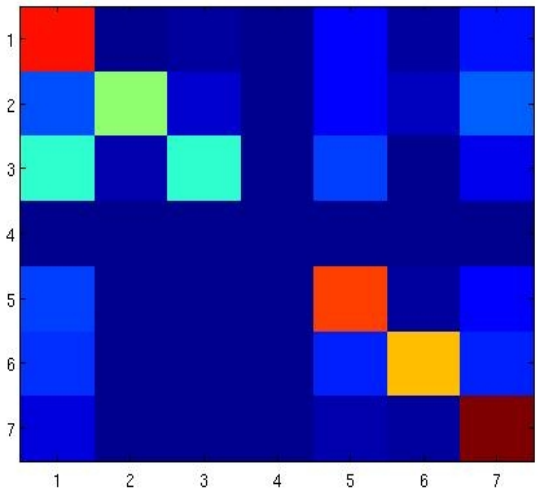


Figura 38: *Place*

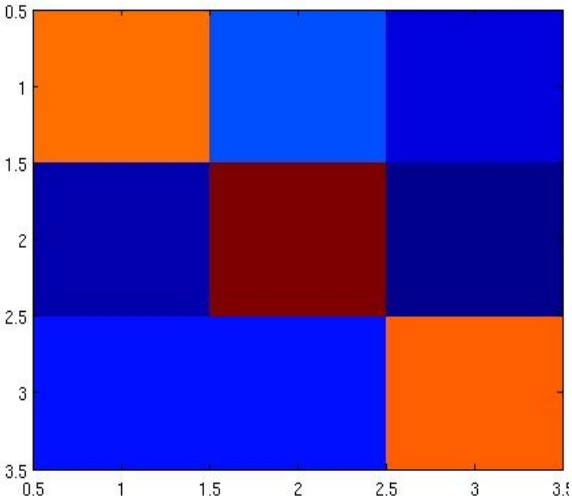


Figura 39: *Voice*

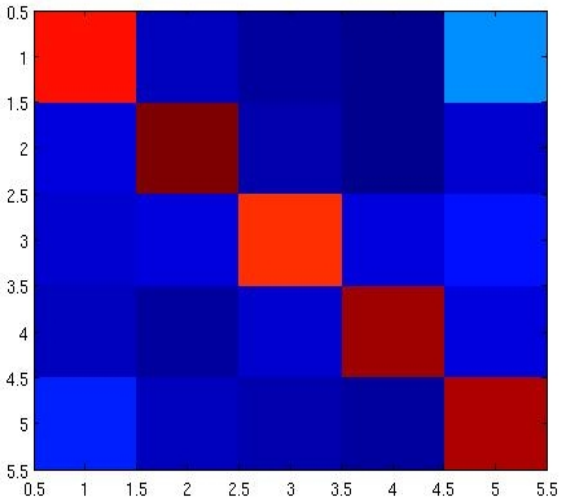


Figura 40: *High-low*

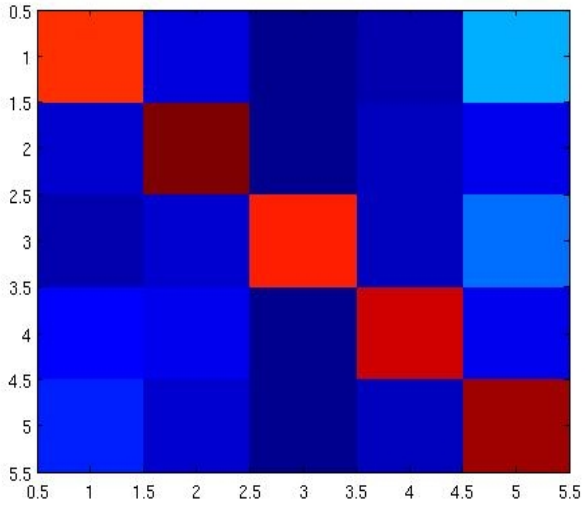


Figura 41: *Front-back*

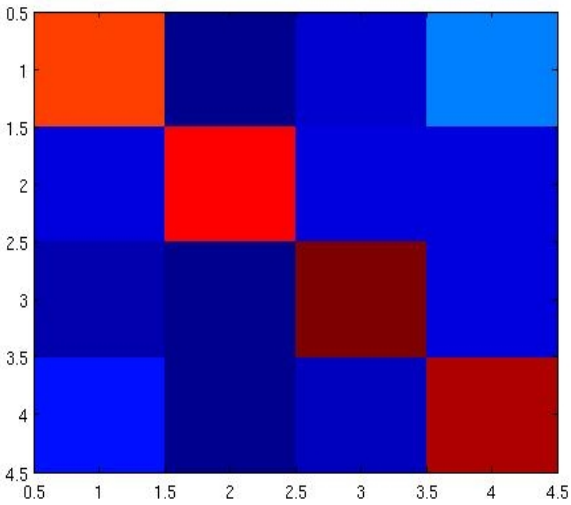


Figura 42: *Round*

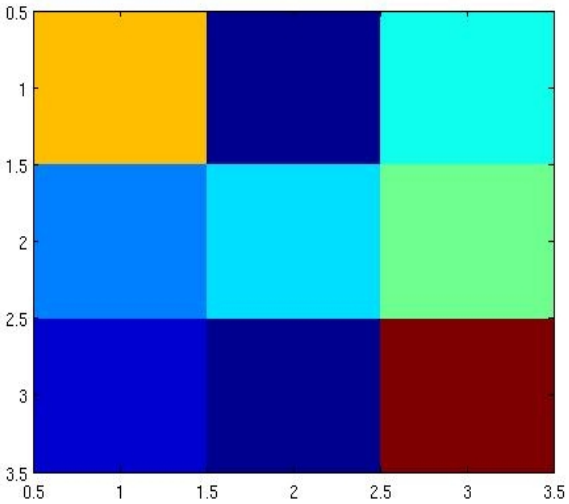


Figura 43: *Nasality*

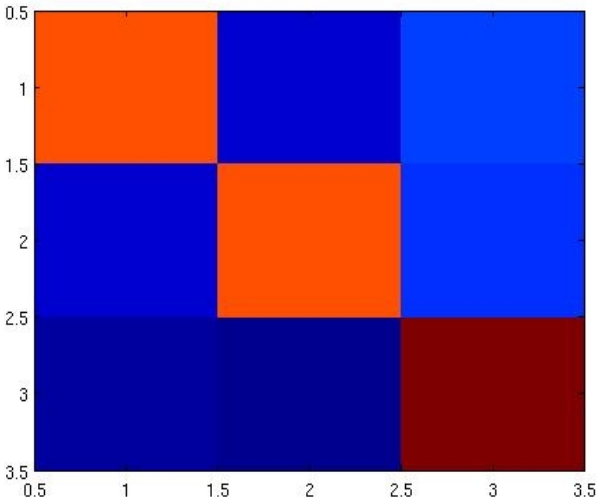


Figura 44: *Static*

De foma general, lo que buscábamos era, esencialmente, que los elementos de la diagonal fueran mayores que el resto y se puede observar que se ha conseguido. Existe una clara pérdida de prestaciones con respecto al tipo de datos “clean” debido a las distorsiones producidas por el ruido, pero aún así vemos que no ha sido muy agresiva en características como “high-low”, “front-back” o “round”. Sin embargo, en algunas otras se acusa más este empeoramiento, sobre todo en “place” y en “nasality”. Para la primera de ellas arrastra la confusión que ya sufría con datos “clean” con

respecto al valor labiodental, pero bastante más acusada en este caso, además de reducir considerablemante la precisión del valor “bilabial”. Para “nasality”, lo más llamativo es la bajada en la precisión de las nasales con respecto a los datos de “clean”. Para “manner” también se han reducido bastante las prestaciones aunque de manera más sutil con respecto a los datos limpios.

De forma más exacta, los valores promedio de precisión de cada una de las carcterísticas se muestran en la siguiente tabla siendo “cv” el conjunto de validación cruzada:

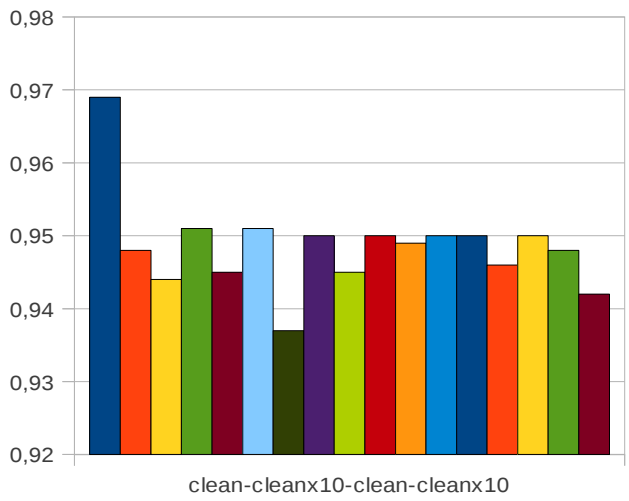
AF	CLEAN acc cv	CLEAN acc train	NOISY acc cv	NOISY acc train
Manner	90,23	92,67	80,66	84,58
Place	89,5	92,72	79,79	84
Voice	92,44	93,58	84,85	87,36
High-low	88,52	91,75	77,96	82,28
Front-back	89,97	92,73	79,7	84,25
Round	90	92,82	79,97	84,08
Static	90,79	93,24	83,02	86,42
Nasality	92,98	94,09	87,16	88,31

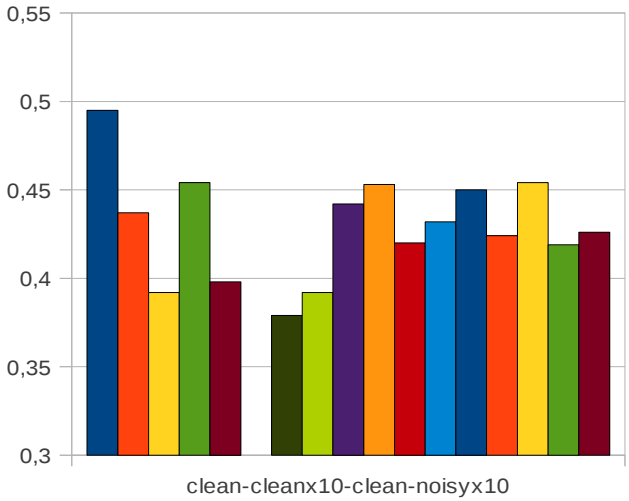
Tabla 8: *Valores promedio de precisión en entrenamiento (train) y validación (cv) para los casos limpio y ruidoso. Se ha desechado el conjunto 1 (fold 1) de la validación cruzada para realizar los cálculos porque es el utilizado para ajustar algunos parámetros del modelo (tuning).*

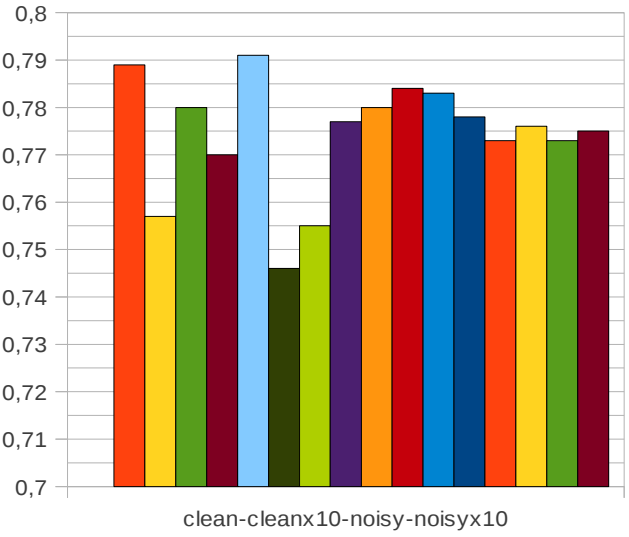
Cuando hablamos de precisión (“*accuracy*”, “*acc*”) en nuestros resultados, nos estamos refiriendo al cociente entre la suma de todos los casos de acierto (diagonal de la matriz de confusión resultante) y el número total de entradas.

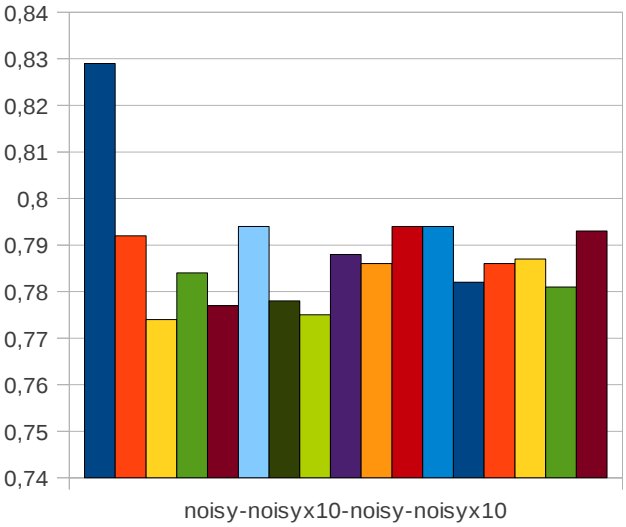
La característica articulatoria más precisa tanto en el conjunto de validación como en el conjunto de entrenamiento es “Nasality”, (cuyos datos hemos identificado de color verde) seguida de “Voice” (azul). Así mismo, los peores resultados, los obtenemos para “High-low” (datos en rojo). Para el mismo sistema pero con una ventana de contexto de 7, estos resultados mejoran más de medio punto.

Centrándonos ahora en el **sistema global**, podemos comprobar de manera general cómo han ido los experimentos y cuáles son los más satisfactorios observando la gráfica de los **promedios de precisión** final. Las tablas de datos obtenidos se pueden encontrar en el anexo 4.) Valores promedio de los experimentos.

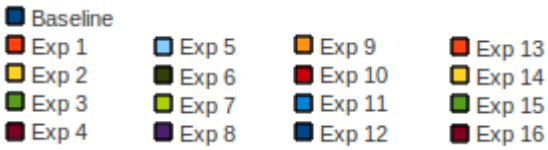








siendo:



Comprobamos que, finalmente, el mejor de los resultados ha sido el de referencia (con el que contábamos en un principio, Baseline), si bien, con los experimentos realizados vemos cómo mejoran o empeoran las prestaciones de nuestro reconocedor dependiendo de los valores de los parámetros. Además, dado que los resultados de las redes neuronales que clasifican las

características articulatorias es bueno, pensamos que se podría mejorar en un futuro actuando sobre la red de la concatenación.

A continuación pasamos a detallar los resultados más llamativos que se han observado.
Teniendo en cuenta para todos los experimentos la siguiente escala de color en los triángulos entrópicos de la salida del reconocedor:

donde en cada nombre distinguimos el tipo de entrenamiento y test del experimento, explicados en la tabla 6, el valor de cada parámetro de ajuste del decodificador de Viterbi a, d y p, explicados en el Anexo 1). Sistema de pruebas ISOLET Testbed, así como el conjunto de test utilizado (s1, s2, s3, s4 o s5, correspondientes a las 5 particiones de la base de datos que realizamos para la validación cruzada LOO).

Las primeras modificaciones para el **número de neuronas en la capa oculta** de la NN nos muestran cómo si disminuimos mucho el número, obtenemos los peores resultados (nhidden=100). Compramos las gráficas del sistema con 100 y con 800 neuronas y vemos que en el segundo, los resultados son ligeramente mejores.

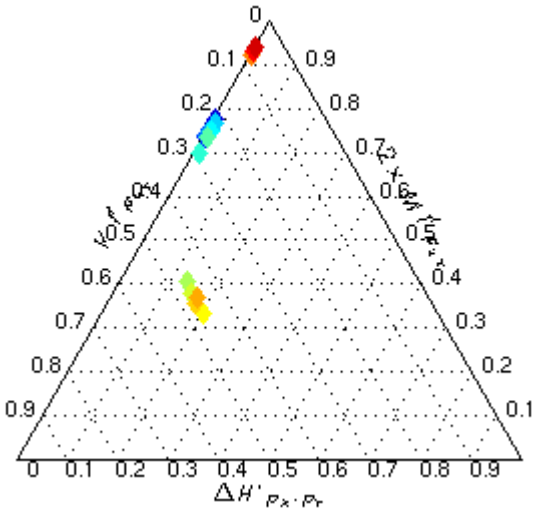


Figura 45: *Exp.2 (nhidden=100). Salidas del reconocedor.*

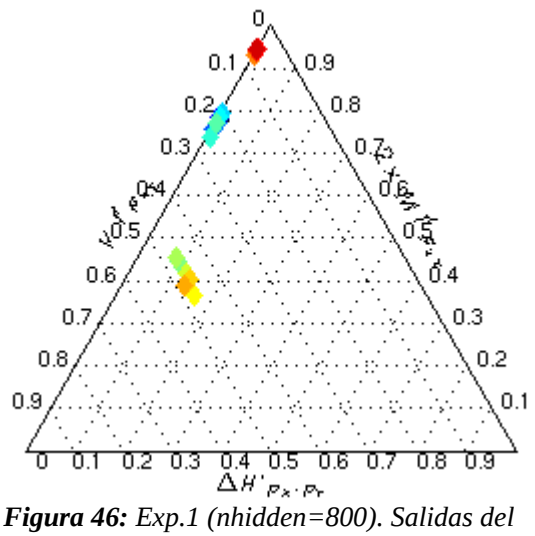


Figura 46: *Exp.1 (nhidden=800). Salidas del reconocedor.*

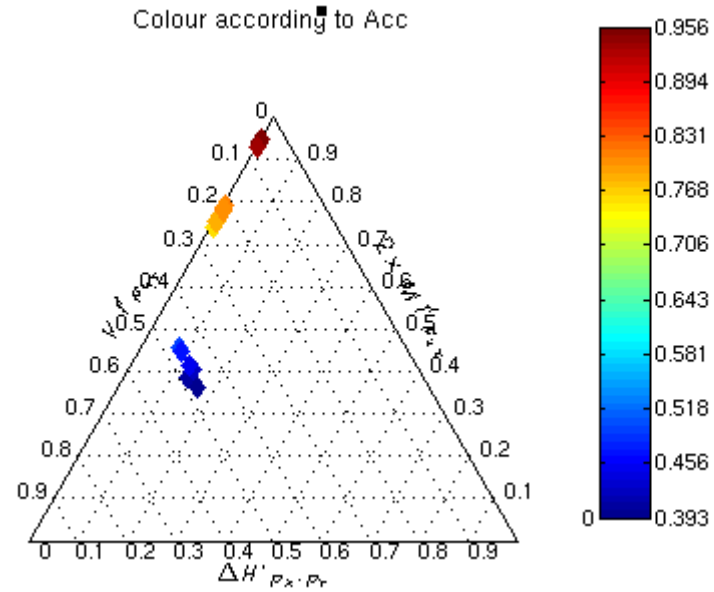


Figura 47: *Exp.1 (nhidden=800). Precisión.*

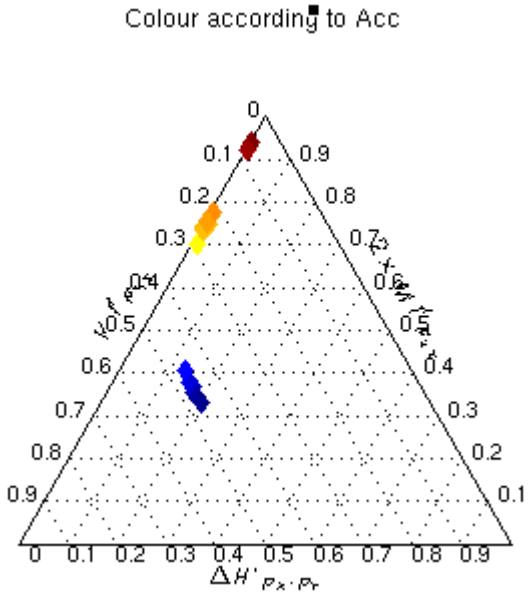
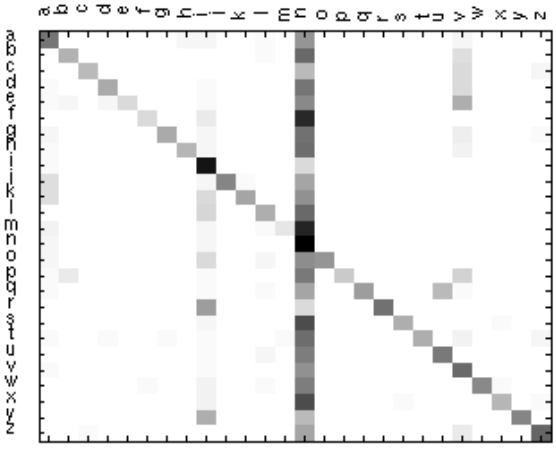
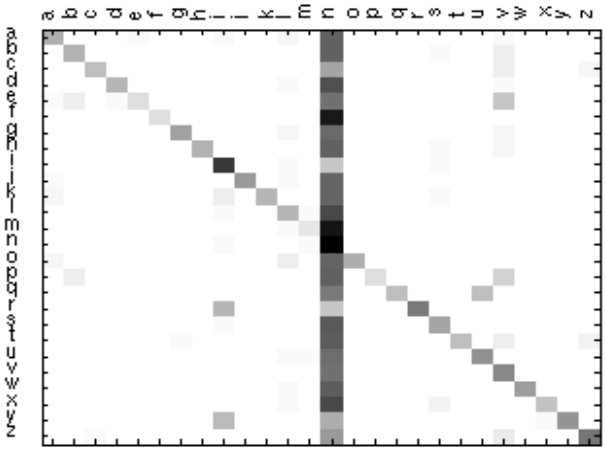


Figura 48: *Exp.2 (nhidden=100). Precisión.*



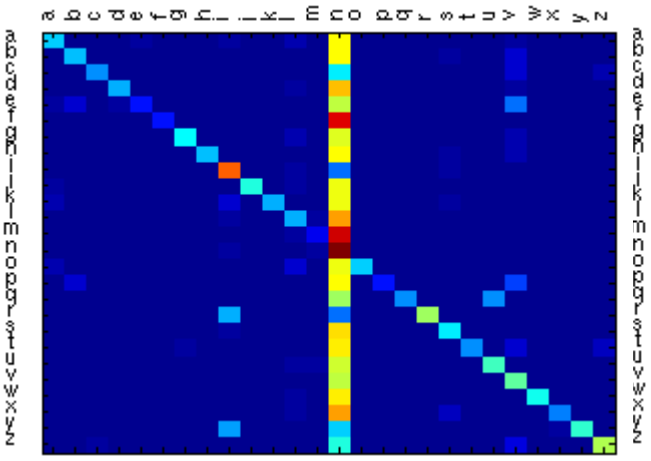
clean-cleanx10-clean-noisyx10

Figura 49:*Exp.1(nhidden=800).*



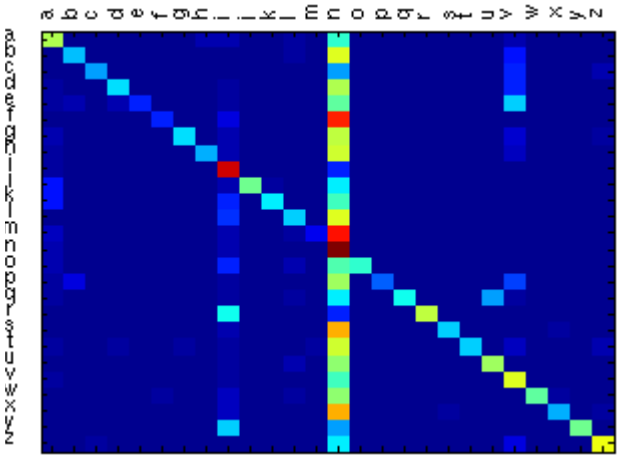
clean-cleanx10-clean-noisyx10

Figura 50: *Exp.2(nhidden=100).*



clean-cleanx10-clean-noisyx10

Figura 51: *Exp.2 (nhidden=100). Mapa de calor.*



clean-cleanx10-clean-noisyx10

Figura 52: *Exp.1 (nhidden=800). Mapa de calor.*

Con los mapas de calor resulta más fácil comparar ambos experimentos. Es importante tener en cuenta que lo que estamos mostrando son las confusiones entre las clases (es decir, las pronunciaciones de las letras del alfabeto) y no los 28 fonemas de salida de las redes neuronales de concatenación. Observamos que la diagonal en el exp.1 contiene colores más cálidos. Resulta también muy llamativo la columna de /n/, que pone de manifiesto un grave problema en el reconocimiento de la nasalidad.

Al probar **distintos valores de la ventana de contexto** también observamos un dato curioso y es que al utilizar una ventana de contexto de valor 7 (cw=7) el experimento clean-cleanx10-clean-noisyx10, no termina de ejecutarse enviando un mensaje de error de fallo del decodificador.

El fallo está siendo que el decodificador está encontrado resultados tan malos que para de buscar. Al comprobar los valores que están dando los ficheros .score y .hyps vemos que el último valor que da en .hyps es: *a0.7d0.2p0.8-s1.hyps*, que en muchos de los casos decide más de 1 letra (columna de la matriz de confusión de .score) y que en otros incluso me muestra el mesaje *No final hypothesis!*, que significa que deja de buscar porque los resultados que encuentra no son buenos.

Para solucionar este error elimino el experimento (clean-cleanx10-clean-noisyx10) del script matrices2.m para no tenerlo en cuenta a la hora de calcular los promedios. El resto de experimentos dan muy buenos resultados. Mostraremos gráficamente una comparativa entre los experimentos 4 y 5, con 3 y 7 ventanas de contexto, respectivamente, para visualizar mejor las diferencias entre ambos.

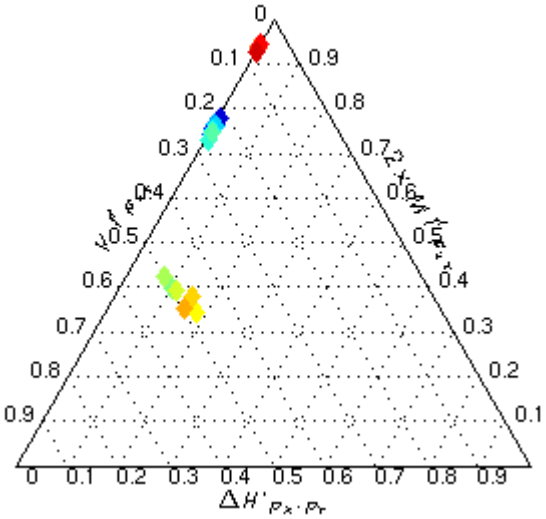


Figura 53: *Exp.4 (cw=3). Salidas del reconocedor.*

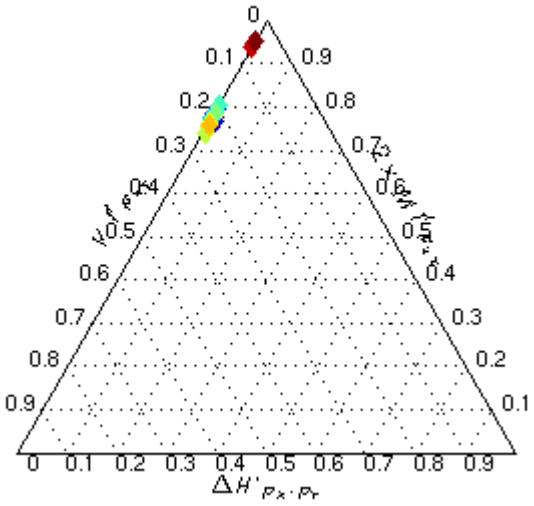


Figura 54: *Exp.5 (cw=7). Salidas del reconocedor.*

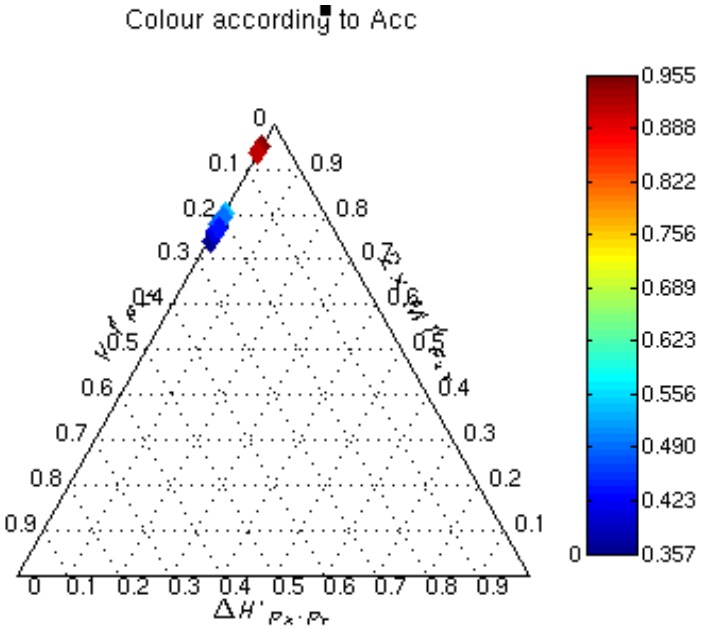


Figura 55: *Exp. 5 (cw=7). Precisión.*

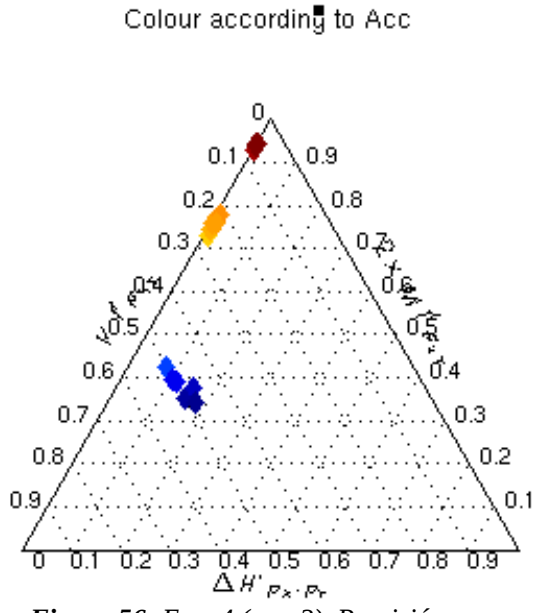


Figura 56: *Exp. 4 (cw=3). Precisión.*

Resulta evidente observando las gráficas cómo, tanto la precisión del sistema en general como la de cada salida del reconocedor por separado, mejora bastante al trabajar con 7 ventanas de contexto.

Utilizar un número diferente de neuronas en la capa oculta dependiendo de la complejidad de la fase de NN también ha aumentado la precisión del sistema, observando que un número de 450

neuronas en la fase AF y un número más elevado en la fase Conc, que es más compleja, resulta lo más apropiado para conseguir unos mejores resultados.

En esta ocasión la comparativa la haremos entre los experimentos 6 y 9, para la salida clean-cleanx10-noisy-noisyx10. El experimento 6, con 100 neuronas en la fase AF y 800 en la fase Conc, arroja los peores resultados de todos los experimentos que se han realizado, por lo que queda patente que elegir bien el número de neuronas en las capas ocultas que trabajarán en las NN's es muy importante. Teniendo en cuenta que nos movemos en márgenes muy pequeños, pues las precisiones son ya bastante elevadas, las diferencias no parecen muy significativas, pero ahí están.

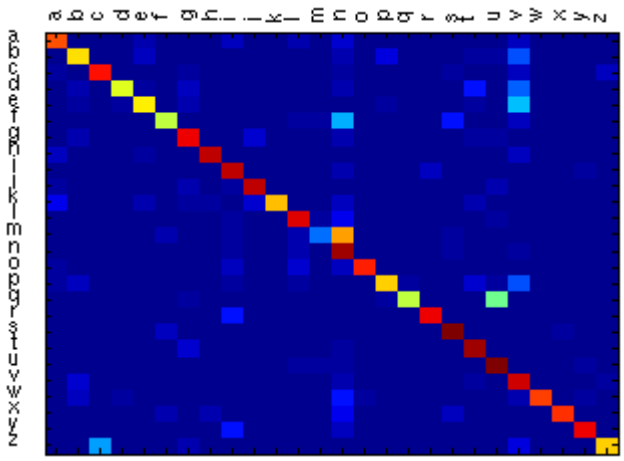


Figura 57: *Exp.6. Mapa de calor.*

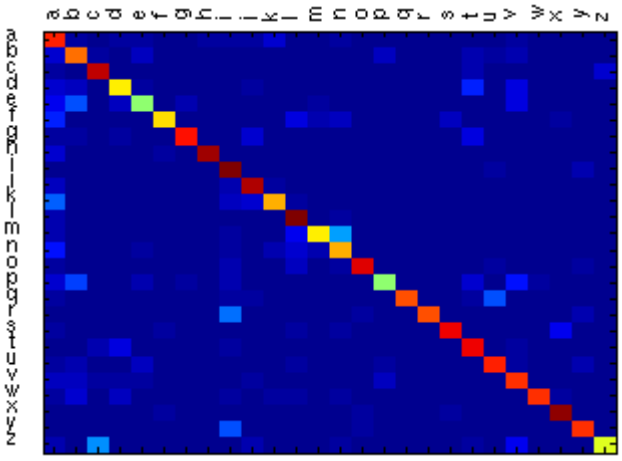


Figura 58: *Exp.9. Mapa de calor.*

Puede verse como para el exp.9, el problema de la nasalidad queda prácticamente solucionado.

La creación de la nueva red nasality y la consiguiente modificación de la red manner, genera una mejora de los resultados.
En primer lugar veremos cómo la activación del parámetro “Reject_last” mejora la precisión en esta nueva red, como ya se mencionó en 6.4, ya que cuando estaba inactivo, el número de nasales frente a no nasales para la red “Nasality” era muy pequeño.

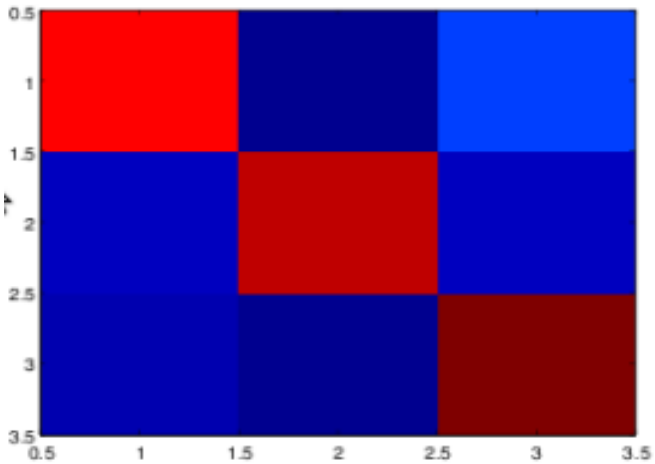


Figura 60: *Nueva NN "Nasality" aislada de "Manner"*

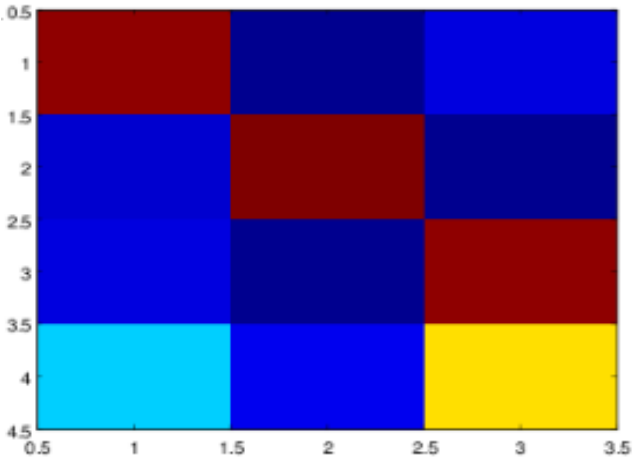


Figura 59: *NN "Nasality" con el parámetro Reject_last activado*

Vemos que la red mejora significativamente activando “Reject_last”. El cuadro amarillo se refiere a las vocales que tenían etiqueta 3 y que reject_last ha desechado. Sin embargo, en el sistema global no se observa una diferencia significativa de las prestaciones para ambos experimentos, como esperábamos en un principio.

Comparando el sistema de 8NN's con el inicial de 7NN's apenas observamos diferencias en los valores de precisión del sistema global:

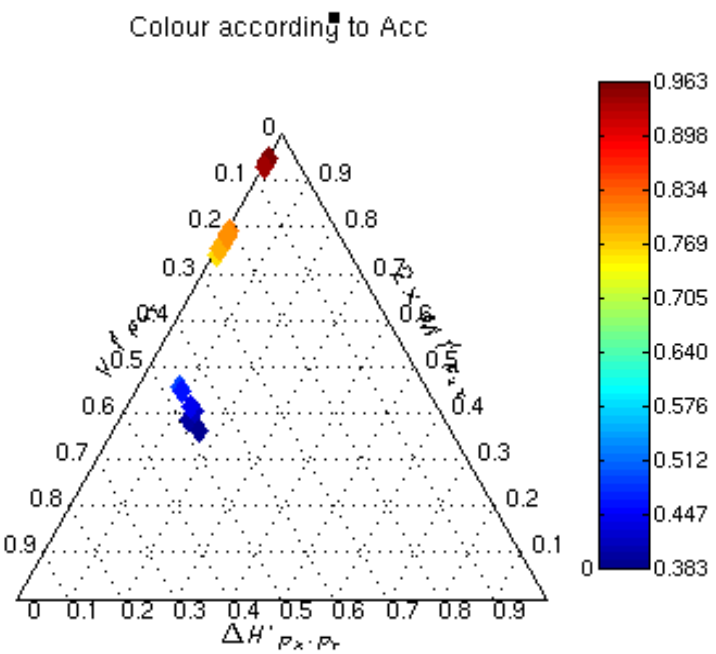


Figura 61: *Exp.1(7NN's).Precisión.*

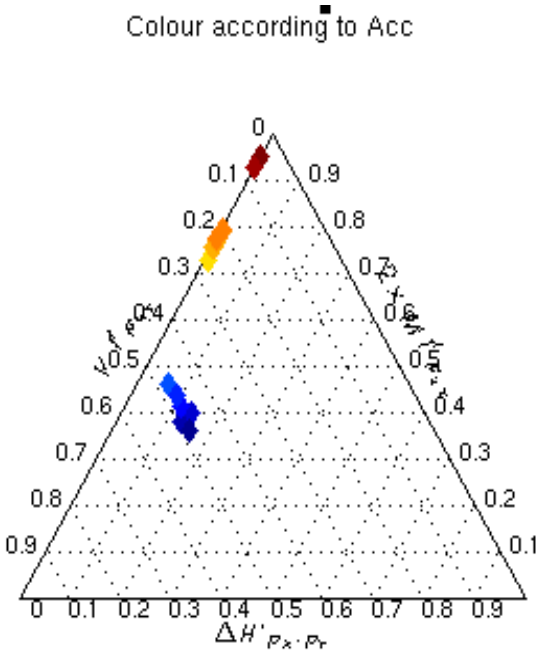


Figura 62: *Exp.11(8NN's con reject_last).Precisión.*

Centrándonos en las matrices de confusión:

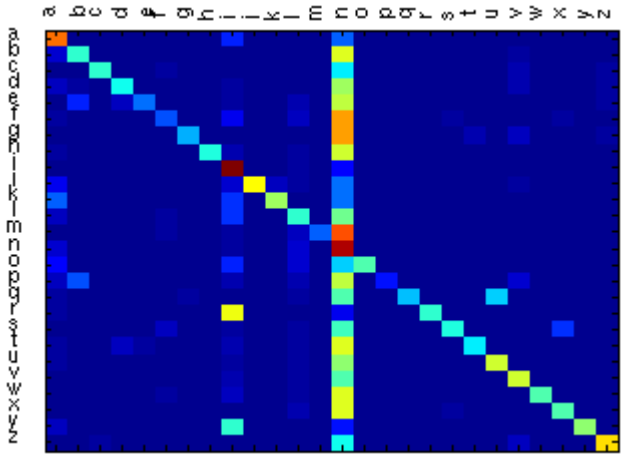


Figura 63: *Exp.1(7NN's).Fallo en la nasalidad.*

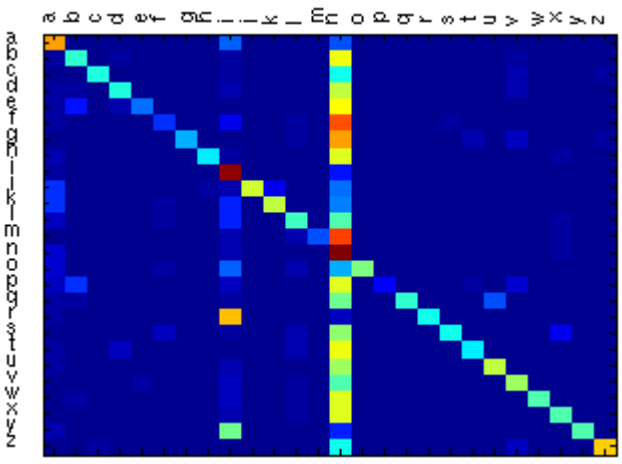


Figura 64: *Exp.11(8NN's).Fallo en la nasalidad.*

Tampoco se observan grandes diferencias entre los sistemas de 7 y 8 redes.

Otra observación que resulta llamativa al estudiar estos mapas de calor es la confusión que produce el fonema /i/. Una posible explicación de que esto suceda es que se trata de un fonema que está muy mal representado en la base de datos, ya que sólo aparece en /i/ y en /y/ (pronunciado “guay”), es decir, que contamos con muy pocos casos para poder estudiarlo. En principio, sería esperable que, con una base de datos de mayor tamaño esto no ocurriera. En cualquier caso, se trata de un asunto que habría que investigar en un futuro.

Otro de los rasgos característicos en muchos reconocedores de habla para lengua inglesa es la confusión que se produce a menudo con el “e-set”. Se trata del conjunto de sonidos caracterizados por contener “e”, siendo fácilmente confundibles entre sí. Notamos como en nuestro sistema también se produce:

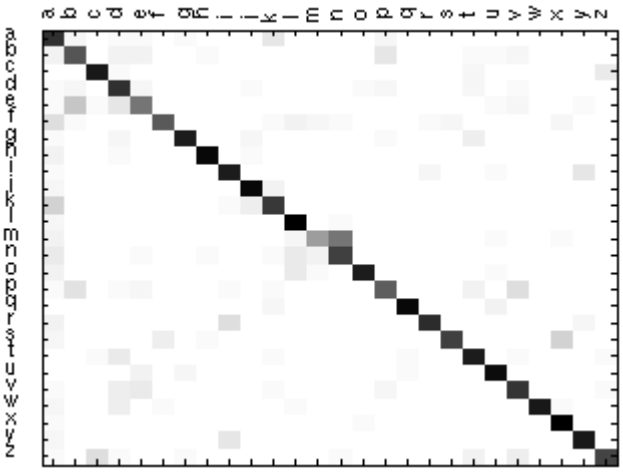


Figura 65: *Exp.7. Matriz de confusión.*

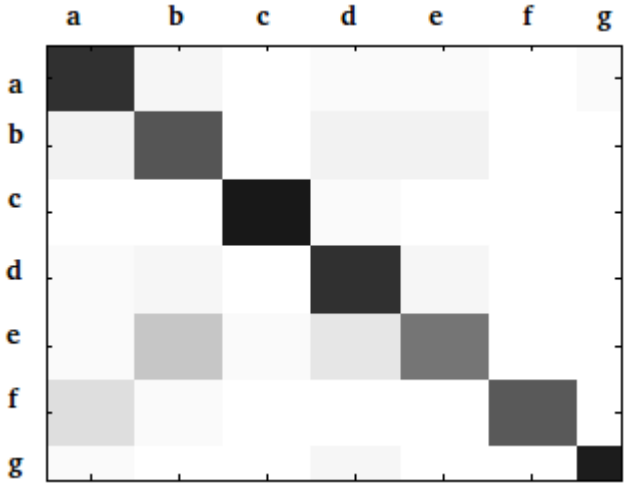


Figura 66: *Exp.7. Detalle del conjunto "e-set".*

Vemos como claramente se producen confusiones entre los sonidos del conjunto “e-set”.

Una observación curiosa en referencia al parámetro reject_last, es que al utilizarlo en el sistema con un mismo número de neuronas en la capa oculta en ambas fases, mejoran ligeramente los resultados de precisión globales para la mayoría de salidas del reconocedor, mientras que cuando aplicamos el parámetro en el sistema utilizando diferente número de neuronas en la capa oculta para cada fase, empeoran. En todo caso, dado que estas variaciones son muy pequeñas, habría que hacer más experimentos para determinar si esto es relevante.

A partir de aquí, para el resto de experimentos realizados comprobamos que la mejora sigue en la misma línea, un número más bajo de neuronas en la capa oculta de las redes más sencillas (fase AF) y un número más elevado en la red más compleja (fase Conc) hacen que la precisión aumente. Los mejores resultados los encontramos para un sistema de 8 redes neuronales con 450 neuronas en la fase AF y 1500 en la fase Conc, con una ventana de contexto de 5.
En esta ocasión,y para finalizar, la comparatoria la veremos sobre el mismo triángulo entrópico. Representaremos el experimento 16 junto con el sistema de referencia con el que contábamos en un principio (Baseline), que ha resultado ser, finalmente, el mejor de todos.

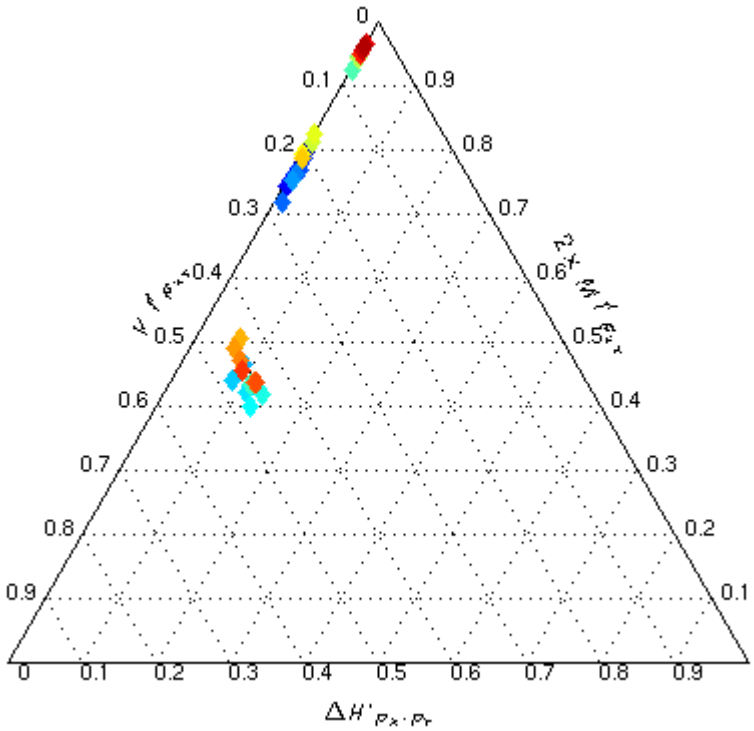
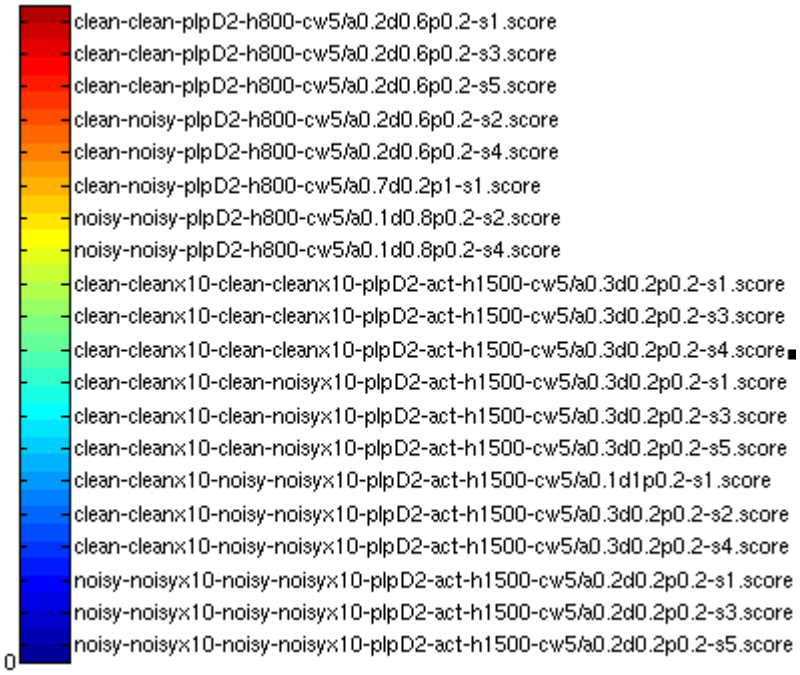


Figura 67: *Experimento 16*

siendo



Capítulo 7. Conclusiones finales y líneas futuras

Para hablar de conclusiones finales, debe tenerse en cuenta que el desarrollo del sistema no ha sido trivial, han surgido diversos contratiempos que han hecho que hayamos ido modificando la estructura del mismo conforme se ha ido avanzando. Se han creado varios programas de automatización o visualización de resultados, algunos de ellos mostrados en el apartado de Anexos.

La creación del sistema de 2 fases (AF y Conc) ha resultado bastante compleja desde el punto de vista de la validación cruzada, ya que al añadir una fase más, se han multiplicado los experimentos a realizar. Desde el punto de vista del estudio del sistema ha sido muy positivo, pues se ha podido comprobar cómo influye la variación de los valores de los parámetros que se han utilizado en la implementación, obteniendo mejores o peores resultados mediante la modificación de los mismos.

Además, se ha conseguido implementar un sistema diferente, abierto a futuras modificaciones, con una estructura de fácil comprensión.

Podemos destacar como **conclusiones finales**, que se trata de un sistema cuya primera parte (faseAF) resulta bastante satisfactoria, ya que se obtienen buenos resultados, sin embargo, el sistema global no alcanza los resultados esperados, es decir, no se ha conseguido superar los valores aportados por el sistema de referencia. Esto puede ser debido a que la segunda parte del sistema (fase Conc) no esté implementada de manera óptima.

Uno de los mejores resultados que se han obtenido ha sido el generado por un sistema de 8 NN, con 450 neuronas en la capa oculta de la red en la fase AF , 1500 en la red de la fase Conc y una ventana de contexto de valor 5. Esto confirma que cada una de las modificaciones que se han ido aportando al sistema inicial, han servido para mejorar los valores finales. Un claro ejemplo de ello es la transformación del sistema inicial de 7 redes neuronales por uno de 8, aislando el valor “nasality” de la red “manner” para convertirla en una NN independiente. Se ha comprobado que arroja el mejor de los resultados en cuanto a precisión, por encima de “voice” que era, en un principio, lo esperado.

Los diferentes enfoques que se han desarrollado demuestran que la línea de trabajo emprendida en este PFC es prometedora. Los problemas que han ido surgiendo, han supuesto una obligación de mejora constante del sistema, de esta forma se han obtenido datos que aportan cuestiones interesantes a tener en cuenta en proyectos o investigaciones futuras. Así, serían varias las posibles **líneas de trabajo a seguir en adelante:**

- Elección de un modelo óptimo para la fase Conc. La fusión de datos o combinación de indicadores es un tema en constante investigación que podría ayudarnos a mejorar las prestaciones de esta parte del sistema. Existen infinidad de modelos que podrían aplicarse, como la complementación de dichas NN con coeficientes MFCC o PLP, utilizar otro tipo de clasificadores, etc.

- El problema de la nasalidad. Se ha intentado solucionar aislando su característica articulatoria y se han obtenido mejoras en los resultados finales, aún así, se debería serguir investigando en esta línea para tratar de reducir al máximo los errores que siguen surgiendo.

- Mejora en el reconocimiento del fonema /i/. Como se ha comentado en el capítulo aterior, este problema podría estar asociado al tamaño de la base de datos, con lo que la solución podría ser la utilización de una base de datos de mayor tamaño.

- Estudio en profundidad de la confusión surgida en la NN “place” entre los valores “labiodental” y “silencio”.

- En trabajos futuros sería conveniente también aplicar sobre los resultados tests estadísticos para calcular los intervalos de confianza.

Capítulo 8. Presupuesto

Para este último apartado, realizaremos un presupuesto para conocer el coste económico del proyecto. En él detallaremos, para cada una de las tareas realizadas, la inversión de tiempo llevada a cabo así como el gasto en material y personal involucrado para, finalmente, calcular el coste total.

Nos centraremos en primer lugar en la **descripción de las fases llevadas a cabo** y el tiempo empleado para cada una de ellas:

1.) **Definición de los objetivos** del proyecto y **organización** de los pasos a seguir en el mismo. Primer esbozo de la estructura que será llevada a cabo.

2.) Búsqueda y **estudio de la documentación** existente en relación a los Reconocedores automáticos de habla, conocimiento del estado del arte e investigación sobre características articulatorias.

3.) **Instalación del software** necesario para la realización del proyecto (sistema operativo, herramientas para el manejo de reconocedores, base de datos, librerías, herramientas de programación...).

4.) **Adaptación y desarrollo del script** proporcionado teniendo en cuenta nuestro sistema en particular.

5.) **Creación de programas** cortos necesarios tanto para el desarrollo del sistema, como para la automatización de resultados y visualización de los mismos.

6.) **Experimentación** sobre el sistema ya desarrollado mediante la variación de parámetros sencillos.

7.) Análisis y **evaluación de resultados**, presentación de los mismos mediante la creación de tablas y gráficas.

8.) **Redacción de la memoria**.

De forma esquemática, el tiempo empleado en su realización ha sido:

TAREA	**TIEMPO EMPLEADO**
1. Definición de los objetivos y organización del PFC	10 horas
2. Estudio de la documentación	250 h.
3. Instalación del software necesario	50 h.
4. Adaptación y desarrollo del script	150 h.
5. Creación de programas cortos	50 h.
6. Experimentación	200 h.
7. Análisis y evaluación de resultados	70 h.
8. Redacción de la memoria	200 h.
TOTAL	**980 horas**

Tabla 9: *Tiempo empleado en la realización de cada fase.*

En segundo lugar se llevará a cabo la evaluación del **coste económico**:

1.) Coste del material/equipamiento:

MATERIAL	**COSTE**
1. Ordenador Hp Intel Pentium Inside	500 €
2. Software Paquete Office 2007	100 €
Licencia Matlab para uso comercial	1900 €
3. Material de oficina	100 €
TOTAL	**2600 euros**

Tabla 10: *Gastos en material*

2.) Coste de los honorarios al personal (costes directos):

Nombre	Categoría	Dedicación %	Tiempo dedicado	Precio/hora	Coste
Isabel Horrillo	Ing.técnica	100	980 h.	30	29400 €
Cármen Peláez	Directora proyecto	20	196 h.	50	9800 €
					39200 euros

Tabla 11: *Gastos de personal*

Por tanto, el coste total del proyecto será:

CONCEPTO	COSTE
Coste material/equipamiento	2600 €
Honorarios personal	39200 €
I.V.A. 21%	8778 €
TOTAL	**50578 euros**

Tabla 12: *Costes directos del proyecto*

El presupuesto total de este proyecto asciende a la cantidad de CINCUENTA MIL QUINIENTOS SETENTA Y OCHO EUROS.

Leganés a 20 de abril de 2015

La ingeniera proyectista

Fdo. Isabel Horrillo Peña.

ANEXO

En este anexo describimos el sistema de pruebas ISOLET con el que hemos trabajado así como los scripts y programas más importantes desarrollados a lo largo del proyecto.

1). Sistema de pruebas ISOLET Testbed

Siguiendo el procedimiento LOO, la base de datos se divide en 5 partes, 4 de las cuales utiliza para entrenamiento y la restante para test. Dependiendo de qué parte utilicemos para test, nombraremos las particiones como:

"s1" => la 1ª parte se utiliza para test y las 2ª,3ª,4ª y 5ª se utilizan para entrenamiento.
"s2" => la 2ª parte se utiliza para test y las 1ª,3ª,4ª y 5ª se utilizan para entrenamiento.
"s3" => la 3ª parte se utiliza para test y las 1ª,2ª,4ª y 5ª se utilizan para entrenamiento.
"s4" => la 4ª parte se utiliza para test y las 1ª,2ª,3ª y 5ª se utilizan para entrenamiento.
"s5" => la 5ª parte se utiliza para test y las 1ª,2ª,3ª y 4ª se utilizan para entrenamiento.

Se utiliza s1 como un conjunto de puesta a punto para optimizar los valores de los parámetros del decodificador que realiza con la herramienta "tune". Esta puesta a punto se puede repetir cuando hay un cambio en el tipo de datos (clean o noisy), el vector de características, el número de unidades ocultas, el tamaño de la ventana de contexto...etc

El reconocedor utiliza un enfoque híbrido que usa MLPs como estimador de las probabilidades de emisión del modelo acústico (NN). Estos MLPs usan una normalización *qqnorm* y entrenamiento de los pesos *qnstrn*, que están en el paquete "**quicknet"**. El motivo de utilizar MLPs es porque normalmente dan mejores resultados que los GMM (Gaussian mixture model) ya que se basan en un sencillo enfoque multi-flujo en el que un MLP se usa para cada flujo por separado y luego se

combinan de manera probabilística a nivel de frame. Para la decodificación se utiliza el decodificador “Noway”.

Cálculo de características:

En las redes neuronales, los formatos de características *pfile* almacenan dichas características para varias iteraciones. La herramienta “feacalc” realiza el cálculo de un pfile de características PLP. “feacat” también puede generar *pfiles*.
Las iteraciones para “clean” o “noisy” pueden colocarse en *pfiles* en el mismo orden en el que aparecen en las listas *clean.wav.files.rand*. Es necesario que el orden de las iteraciones en *pfile* coincida con el orden de la etiqueta frame-level *all.align.ilab*. También es necesario que coincidan el nº de frames en ambos casos. Si varían en alguna trama, el comando *-deslenfile* puede solucionarlo.

Se deben proporcionar los archivos de estadísticas para normalización global (1 por cada sX) para acompañar a cada *pfile* y almacenarlos en el mismo directorio pero con el sufijo *.norms* en lugar de *.pfile*. Para crear estos archivos se usa la herramienta "norms". Se creará un *.norms* por cada sX y contendrá estadísticas calculadas sobre los datos de entrenamiento.

Puesta a punto de los parámetros del decodificador:

Es recomendable utilizar “s1” para proporcionar un conjunto de puesta a punto que optimice los valores de los parámetros. Una vez optimizados estos parámetros, se pueden utilizar las otras particiones como datos de test ordinarios La herramienta “tool” se puede usar para testear un rango de posibles valores para esos parámetros utilizando “s1”; nos devolverá aquellos valores que nos dén el mejor rendimiento.

Ejemplo de reconocimiento para un sistema de un flujo:

El tamaño de la capa de entrada de un MLP se calcula con el nº de características x el nº de frames de la ventana de contexto. Los datos se especifican en el nombre al hacer la invocación, x ejem: “clean-plpD2-h1600- cw5” siendo *clean* el tipo de dato, *plpD2* el tipo de característica, *h1600* el nº de capas ocultas y *cw5* el tamaño de la ventana de contexto.

La herramienta "tune" experimenta con distintos valores de parámetros del decodificador y crea el fichero *best-tune.out* que especifica los valores de dichos parámetros que dan el mínimo **WER** (Word Error Rate). Si existen varios conjuntos que proporcionan el mínimo WER, nos informa de todos ellos. Los parámetros que se especifican son:

a => acoustic scale (escala acústica)
d => duration scale (escala temporal)
p => phone detection penalty (detección de fonema)

Utilizaremos después estos parámetros para obtener los resultados de test con la herramienta “tool” y las otras “sX”.

Cuando existen varios grupos de valores de parámetros que nos ofrecen un mínimo WER, necesito seleccionar uno de ellos para hacer el test en las otras carpetas. Para ello se utiliza la herramienta “pick-point”, que trabaja en contra de los datos atípicos o outliers. Me dice cuál de los puntos es el más cercano en distancia euclídea con la media del conjunto. Será este el punto que selecciono para test en las otras “sX”.

2). Script 1stream_concatenacion.csh

```
#! /bin/tcsh -f
setenv ROOTNAME plpD2
setenv ROOTDIR $HOME/Escritorio/entorno2/icsi-scenic-tools-20120105/scripts/testbed/
setenv SCRIPTS $HOME/Escritorio/entorno2/icsi-scenic-tools-20120105/scripts/
setenv SOURCES $HOME/Escritorio/entorno2/icsi-scenic-tools-20120105/scripts/testbed/tools/
setenv LISTDIR $HOME/Escritorio/entorno2/icsi-scenic-tools-20120105/scripts/testbed/config/lists/
setenv FEADIR  $ROOTDIR/fea/
setenv NETSDIR $ROOTDIR/nets/
setenv CONFIG  $HOME/Escritorio/entorno2/icsi-scenic-tools-20120105/scripts/testbed/config/
setenv ACTDIR  $ROOTDIR/act/
setenv CONTEXT_AF 7
setenv NHIDDEN_AF 450
setenv CONTEXT_CONC 7
setenv NHIDDEN_CONC 1500

alias rm rm

######################
#Feature extraction
######################

#Change the dirs where the database is
sed 's#u/gelbart/raid/isolet-1.3#/home/isi/Escritorio/entorno2/icsi-scenic-tools-20120105/isolet-db/isolet#g' $LISTDIR/clean.wav.files.rand >&  $LISTDIR/clean.wav.files.rand2
sed 's#/Escritorio#/home/isi/Escritorio#g' $LISTDIR/clean.wav.files.rand >& $LISTDIR/clean.wav.files.rand2
sed 's#u/gelbart/raid/isolet-1.3#home/isi/Escritorio/entorno2/icsi-scenic-tools-20120105/isolet-db/isolet#g' $LISTDIR/noisy.wav.files.rand >&  $LISTDIR/noisy.wav.files.rand2
sed 's#/Escritorio#/home/isi/Escritorio#g' $LISTDIR/noisy.wav.files.rand >& $LISTDIR/noisy.wav.files.rand2

#################
#Net training
#################

 #Feature extraction for clean
#PLP features
feacalc -rasta false -dom cepstra -plp 12 -win 25 -step 10 -dither -hpfilter -delta 2 -list $LISTDIR/clean.wav.files.rand  -sam 16000 -o $FEADIR/clean-$ROOTNAME.pfile >& feaclean.out
```

```
# Adjust utterance lengths to match labels. This should not change any
# utterance by more than 2 frames!
feacat -deslenfile $CONFIG/labels/all.deslen -o $FEADIR/clean-$ROOTNAME-adj.pfile $FEADIR/clean-
$ROOTNAME.pfile
rm -r $FEADIR/clean-$ROOTNAME.pfile
mv $FEADIR/clean-$ROOTNAME-adj.pfile $FEADIR/clean-$ROOTNAME.pfile

#Compute statistics files for normalization
$SOURCES/norms $FEADIR/clean-$ROOTNAME.pfile

#PLP features for noisy
feacalc -rasta false -dom cepstra  -plp 12 -win 25 -step 10 -dither -hpfilter -delta 2 -list
$LISTDIR/noisy.wav.files.rand  -sam 16000 -o $FEADIR/noisy-$ROOTNAME.pfile >&feanoisy.out

# Adjust utterance lengths to match labels. This should not change any
# utterance by more than 2 frames!
feacat -deslenfile $CONFIG/labels/all.deslen -o $FEADIR/noisy-$ROOTNAME-adj.pfile $FEADIR/noisy-
$ROOTNAME.pfile
rm -r $FEADIR/noisy-$ROOTNAME.pfile
mv $FEADIR/noisy-$ROOTNAME-adj.pfile $FEADIR/noisy-$ROOTNAME.pfile

#Compute statistics files for normalization
$SOURCES/norms $FEADIR/noisy-$ROOTNAME.pfile

#For the five folds on clean
foreach fold (1 2 3 4 5)
foreach art_feature (1 2 3 4 5 6 7 8)

#Invoco a mis 7 entrenamientos anteriores (un total de 35 veces)
$SOURCES/train_concatenacion -context $CONTEXT_AF -trdir $NETSDIR/train/  -feadir $FEADIR/ clean
$fold $art_feature $NHIDDEN_AF
end
end

#Invoco a fwd_concatenacion (un total de 5 veces)
foreach fold (1 2 3 4 5)

 $SOURCES/fwd_concatenacion -context $CONTEXT_AF -trdir $NETSDIR/train/  -feadir $FEADIR/ clean
clean $fold $ACTDIR/clean-clean-$ROOTNAME-act-fold$fold.pfile 1 $NHIDDEN_AF 2 $NHIDDEN_AF 3
$NHIDDEN_AF 4 $NHIDDEN_AF 5 $NHIDDEN_AF 6 $NHIDDEN_AF 7 $NHIDDEN_AF 8 $NHIDDEN_AF

end

#For the five folds on noisy
foreach fold (1 2 3 4 5)
foreach art_feature (1 2 3 4 5 6 7 8)

#Invoco a mis 7 entrenamientos anteriores (un total de 35 veces)
$SOURCES/train_concatenacion -context $CONTEXT_AF -trdir $NETSDIR/train/  -feadir $FEADIR/ noisy
$fold $art_feature $NHIDDEN_AF
end
end
```

#Invoco a fwd_concatenacion (un total de 5 veces)
foreach fold (1 2 3 4 5)

$SOURCES/fwd_concatenacion -context $CONTEXT_AF -trdir $NETSDIR/train/ -feadir $FEADIR/ clean noisy $fold $ACTDIR/clean-noisy-$ROOTNAME-act-fold$fold.pfile 1 $NHIDDEN_AF 2 $NHIDDEN_AF 3 $NHIDDEN_AF 4 $NHIDDEN_AF 5 $NHIDDEN_AF 6 $NHIDDEN_AF 7 $NHIDDEN_AF 8 $NHIDDEN_AF
end

#For the five folds on noisy
foreach fold (1 2 3 4 5)
foreach art_feature (1 2 3 4 5 6 7 8)

#Invoco a mis 7 entrenamientos anteriores (un total de 35 veces)
$SOURCES/train_concatenacion -context $CONTEXT_AF -trdir $NETSDIR/train/ -feadir $FEADIR/ noisy $fold $art_feature $NHIDDEN_AF
end
end

#Invoco a fwd_concatenacion (un total de 5 veces)
foreach fold (1 2 3 4 5)

$SOURCES/fwd_concatenacion -context $CONTEXT_AF -trdir $NETSDIR/train/ -feadir $FEADIR/ noisy noisy $fold $ACTDIR/noisy-noisy-$ROOTNAME-act-fold$fold.pfile 1 $NHIDDEN_AF 2 $NHIDDEN_AF 3 $NHIDDEN_AF 4 $NHIDDEN_AF 5 $NHIDDEN_AF 6 $NHIDDEN_AF 7 $NHIDDEN_AF 8 $NHIDDEN_AF
end

#CONCATENO LOS 5 PFILES DE ENTRADA PARA QUEDARME SÓLO CON 1 (ANTES DEL ENTRENAMIENTO DE LAS 10 NN)
pfile_concat -o $ACTDIR/clean-cleanx10-$ROOTNAME-act.pfile $ACTDIR/clean-clean-$ROOTNAME-act-fold1.pfile $ACTDIR/clean-clean-$ROOTNAME-act-fold2.pfile $ACTDIR/clean-clean-$ROOTNAME-act-fold3.pfile $ACTDIR/clean-clean-$ROOTNAME-act-fold4.pfile $ACTDIR/clean-clean-$ROOTNAME-act-fold5.pfile

 # Adjust utterance lengths to match labels. This should not change any utterance by more than 2 frames!
feacat -deslenfile $CONFIG/labels/all.deslen -o $ACTDIR/clean-clean-$ROOTNAME-act-adj.pfile $ACTDIR/clean-cleanx10-$ROOTNAME-act.pfile
rm -r $ACTDIR/clean-cleanx10-$ROOTNAME-act.pfile
mv $ACTDIR/clean-clean-$ROOTNAME-act-adj.pfile $ACTDIR/clean-cleanx10-$ROOTNAME-act.pfile

#Compute statistics files for normalization

$SOURCES/norms $ACTDIR/clean-cleanx10-$ROOTNAME-act.pfile

#CONCATENO LOS 5 PFILES DE ENTRADA PARA QUEDARME SÓLO CON 1 (ANTES DEL ENTRENAMIENTO DE LAS 10 NN)
pfile_concat -o $ACTDIR/clean-noisyx10-$ROOTNAME-act.pfile $ACTDIR/clean-noisy-$ROOTNAME-act-fold1.pfile $ACTDIR/clean-noisy-$ROOTNAME-act-fold2.pfile $ACTDIR/clean-noisy-$ROOTNAME-act-fold3.pfile $ACTDIR/clean-noisy-$ROOTNAME-act-fold4.pfile $ACTDIR/clean-noisy-$ROOTNAME-act-fold5.pfile

```
# Adjust utterance lengths to match labels. This should not change any utterance by more than 2 frames!
feacat -deslenfile $CONFIG/labels/all.deslen -o $ACTDIR/clean-noisy-$ROOTNAME-act-adj.pfile $ACTDIR/clean-noisyx10-$ROOTNAME-act.pfile
rm -r $ACTDIR/clean-noisyx10-$ROOTNAME-act.pfile
mv $ACTDIR/clean-noisy-$ROOTNAME-act-adj.pfile $ACTDIR/clean-noisyx10-$ROOTNAME-act.pfile
#Compute statistics files for normalization
$SOURCES/norms $ACTDIR/clean-noisyx10-$ROOTNAME-act.pfile

#CONCATENO LOS 5 PFILES DE ENTRADA PARA QUEDARME SÓLO CON 1 (ANTES DEL ENTRENAMIENTO DE LAS 10 NN)
pfile_concat -o $ACTDIR/noisy-noisyx10-$ROOTNAME-act.pfile $ACTDIR/noisy-noisy-$ROOTNAME-act-fold1.pfile $ACTDIR/noisy-noisy-$ROOTNAME-act-fold2.pfile $ACTDIR/noisy-noisy-$ROOTNAME-act-fold3.pfile $ACTDIR/noisy-noisy-$ROOTNAME-act-fold4.pfile $ACTDIR/noisy-noisy-$ROOTNAME-act-fold5.pfile

# Adjust utterance lengths to match labels. This should not change any utterance by more than 2 frames!
feacat -deslenfile $CONFIG/labels/all.deslen -o $ACTDIR/noisy-noisy-$ROOTNAME-act-adj.pfile $ACTDIR/noisy-noisyx10-$ROOTNAME-act.pfile
rm -r $ACTDIR/noisy-noisyx10-$ROOTNAME-act.pfile
mv $ACTDIR/noisy-noisy-$ROOTNAME-act-adj.pfile $ACTDIR/noisy-noisyx10-$ROOTNAME-act.pfile

#Compute statistics files for normalization
$SOURCES/norms $ACTDIR/noisy-noisyx10-$ROOTNAME-act.pfile

#ENTRENO LAS 10 NN CON CADA UNA DE LAS ENTRADAS OBTENIDAS DE LA CONCATENACIÓN ANTERIOR

#For the five folds on clean
foreach fold (1 2 3 4 5)

$SOURCES/trainx10 -context $CONTEXT_CONC -trdir $NETSDIR/trainx10/  -feadir $ACTDIR/ clean-clean $fold $ROOTNAME $NHIDDEN_CONC
end

#For the five folds on noisy
foreach fold (1 2 3 4 5)

$SOURCES/trainx10 -context $CONTEXT_CONC -trdir $NETSDIR/trainx10/  -feadir $ACTDIR/ noisy-noisy $fold $ROOTNAME $NHIDDEN_CONC
end

################

#Tuning of the decoder based on the first fold for clean clean clean
################
$SOURCES/tune -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/ -feadir $ACTDIR/ clean-cleanx10 clean-cleanx10 $ROOTNAME-act $NHIDDEN_CONC

###############
#Testing on clean clean clean
################
```

```
#For the 5 folds
foreach fold (1 2 3 4 5)
   $SOURCES/testPP -context $CONTEXT_CONC -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/
-feadir $ACTDIR/ clean-cleanx10 clean-cleanx10 $fold $ROOTNAME-act $NHIDDEN_CONC
end

################
#Tuning of the decoder based on the first fold for clean clean noisy
################
$SOURCES/tune -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/ -feadir $ACTDIR/ clean-cleanx10
clean-noisyx10 $ROOTNAME-act $NHIDDEN_CONC

###############
#Testing on clean clean noisy
################

#For the 5 folds
foreach fold (1 2 3 4 5)
   $SOURCES/testPP -context $CONTEXT_CONC -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/
-feadir $ACTDIR/ clean-cleanx10 clean-noisyx10 $fold $ROOTNAME-act $NHIDDEN_CONC
end

################
#Tuning of the decoder based on the first fold for clean noisy noisy
################
$SOURCES/tune -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/ -feadir $ACTDIR/ clean-cleanx10
noisy-noisyx10 $ROOTNAME-act $NHIDDEN_CONC

###############
#Testing on clean noisy noisy
################

#For the 5 folds
foreach fold (1 2 3 4 5)
   $SOURCES/testPP -context $CONTEXT_CONC -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/
-feadir $ACTDIR/ clean-cleanx10 noisy-noisyx10 $fold $ROOTNAME-act $NHIDDEN_CONC
end

################
#Tuning of the decoder based on the first fold for noisy noisy noisy
################

$SOURCES/tune -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/ -feadir $ACTDIR/ noisy-noisyx10
noisy-noisyx10 $ROOTNAME-act $NHIDDEN_CONC

###############
#Testing on noisy noisy noisyy
################

#For the 5 folds
foreach fold (1 2 3 4 5)
```

```
    $SOURCES/testPP -context $CONTEXT_CONC -trdir $NETSDIR/trainx10/ -tedir $NETSDIR/testx10/
-feadir $ACTDIR/ noisy-noisyx10 noisy-noisyx10 $fold $ROOTNAME-act $NHIDDEN_CONC
end
```

3.) Etiquetas.m

```
d = fopen('all.align.ascii');
size(d)
A = fscanf(d,'%d');
N=size(A)
%Recorremos y guardamos cada una de las 3 columnas del All.align.ascii
J=1;
for i = 1:3:N(1),
        Primera(J) = A(i);
        J=J+1;
end

J=1;
for i = 2:3:N(1),
        Segunda(J) = A(i);
        J=J+1;
end

 J=1;
 for i = 3:3:N(1),
        Tercera(J) = A(i);
        J=J+1;
 end

 matrizAF=xlsread('isi.xlsx','hoja1','a1:h28');
 M=size(Tercera)
 p = fopen('all.alignAF_AM1.ascii','w');
 for fonema=1:1:M(2),
      fprintf(p,'%d ',Primera(fonema));
      fprintf(p,'%d ',Segunda(fonema));
      fprintf(p,'%d\n',matrizAF(Tercera(fonema)+1,2));
 end
 fclose(p);
 q = fopen('all.alignAF_AM2.ascii','w');
 for fonema=1:1:M(2),
      fprintf(p,'%d ',Primera(fonema));
      fprintf(p,'%d ',Segunda(fonema));
      fprintf(q,'%d\n',matrizAF(Tercera(fonema)+1,3));
 end
 fclose(q);
 r = fopen('all.alignAF_AM3.ascii','w');
 for fonema=1:1:M(2),
```

```
        fprintf(p,'%d ',Primera(fonema));
        fprintf(p,'%d ',Segunda(fonema));
        fprintf(r,'%d\n',matrizAF(Tercera(fonema)+1,4));
end
fclose(r);
 s = fopen('all.alignAF_AM4.ascii','w');
  for fonema=1:1:M(2),
         fprintf(p,'%d ',Primera(fonema));
        fprintf(p,'%d ',Segunda(fonema));
        fprintf(s,'%d\n',matrizAF(Tercera(fonema)+1,5));
  end
  fclose(s);
   t = fopen('all.alignAF_AM5.ascii','w');
  for fonema=1:1:M(2),
        fprintf(p,'%d ',Primera(fonema));
        fprintf(p,'%d ',Segunda(fonema));
        fprintf(t,'%d\n',matrizAF(Tercera(fonema)+1,6));
  end
  fclose(t);
   u = fopen('all.alignAF_AM6.ascii','w');
  for fonema=1:1:M(2),
        fprintf(p,'%d ',Primera(fonema));
        fprintf(p,'%d ',Segunda(fonema));
        fprintf(u,'%d\n',matrizAF(Tercera(fonema)+1,7));
  end
  fclose(u);
  v = fopen('all.alignAF_AM7.ascii','w');
  for fonema=1:1:M(2),
     fprintf(p,'%d ',Primera(fonema));
     fprintf(p,'%d ',Segunda(fonema));
     fprintf(v,'%d\n',matrizAF(Tercera(fonema)+1,8));
  end
  fclose(v);
fclose(d);
```

4.) Valores promedio de los experimentos

Teniendo en cuenta que (más detalles en [39]): $\mathbf{k_X}$ = perplejidad a la entrada; $\mathbf{k_{X|Y}}$ = perplejidad remanente representando la información de la entrada que no se ha transmitido hacia la salida en el clasificador; $\boldsymbol{\mu_{XY}}$ = factor de transmisión de información; $\mathbf{a(P_{XY})}$= precisión (accuracy) basada exclusivamente en los conteos de los aciertos de la matriz de confusión; $\mathbf{a'(P_{XY})}$ = precisión modulada por la entropía (EMA - Entropy Modulated Accuracy-) o corrección de la precisión en base a la perplejidad remanente; $\mathbf{q_X(P_{XY})}$ = factor NIT (Normalized Information Transfer factor), medida de calidad independiente de las distribuciones de entrada y salida; **1-CEN** = opuesto de la medida CEN (Confusion Entropy) para situarla entre los valores 0 y 1; **MCC** = Matthew Correlation Coefficient (MCC), una medida de calidad clásica.

PROMEDIOS DEL EXPERIMENTO DE REFERENCIA (BASELINE)

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-clean-plpD2-h800-cw5	25.999	1.140	22.807	0.969	0.877	0.877	0.969	0.968
clean-noisy-plpD2-h800-cw5	25.907	5.694	4.580	0.495	0.176	0.176	0.644	0.497
noisy-noisy-plpD2-h800-cw5	25.994	1.917	13.595	0.829	0.522	0.523	0.845	0.823

PROMEDIOS DE LA PRUEBA 1: 7 NN CON 800 NEURONAS EN LA CAPA OCULTA y cw5

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h800-cw5	25.998	1.215	21.400	0.948	0.823	0.823	0.955	0.946
clean-cleanx10-clean-noisyx10-plpD2-act-h800-cw5	25.920	6.797	3.856	0.437	0.147	0.148	0.630	0.446
clean-cleanx10-noisy-noisyx10-plpD2-act-h800-cw5	25.981	2.109	12.356	0.789	0.474	0.475	0.826	0.782
noisy-noisyx10-noisy-noisyx10-plpD2-act-h800-cw5	25.975	2.101	12.392	0.792	0.476	0.477	0.824	0.785

PROMEDIOS DE LA PRUEBA 2: 7 NN CON 100 NEURONAS EN LA CAPA OCULTA

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h100-cw5	25.998	1.240	20.983	0.944	0.807	0.807	0.950	0.942
clean-cleanx10-clean-noisyx10-plpD2-act-h100-cw5	25.946	7.710	3.390	0.392	0.130	0.130	0.635	0.418
clean-cleanx10-noisy-noisyx10-plpD2-act-h100-cw5	25.986	2.333	11.188	0.757	0.429	0.430	0.803	0.749
noisy-noisyx10-noisy-noisyx10-plpD2-act-h100-cw5	25.979	2.238	11.632	0.774	0.447	0.447	0.808	0.766

PROMEDIOS DE LA PRUEBA 3: 7 NN CON 450 NEURONAS EN LA CAPA OCULTA

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h450-cw5	25.998	1.210	21.506	0.951	0.827	0.827	0.956	0.949
clean-cleanx10-clean-noisyx10-plpD2-act-h450-cw5	25.912	6.425	4.042	0.454	0.156	0.155	0.631	0.458
clean-cleanx10-noisy-noisyx10-plpD2-act-h450-cw5	25.992	2.178	11.970	0.780	0.459	0.460	0.818	0.773
noisy-noisyx10-noisy-noisyx10-plpD2-act-h450-cw5	25.970	2.153	12.125	0.784	0.464	0.466	0.820	0.776

PROMEDIOS DE LA PRUEBA 4: 7 NN CON 3 VENTANAS DE CONTEXTO

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h800-cw3	25.998	1.225	21.236	0.945	0.816	0.817	0.953	0.943
clean-cleanx10-clean-noisyx10-plpD2-act-h800-cw3	25.950	7.532	3.474	0.398	0.133	0.134	0.611	0.405
clean-cleanx10-noisy-noisyx10-plpD2-act-h800-cw3	25.991	2.248	11.575	0.770	0.445	0.445	0.811	0.762
noisy-noisyx10-noisy-noisyx10-plpD2-act-h800-cw3	25.985	2.203	11.817	0.777	0.454	0.454	0.815	0.769

PROMEDIOS DE LA PRUEBA 5: 7 NN CON 7 VENTANAS DE CONTEXTO

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h800-cw7	25.999	1.205	21.578	0.951	0.830	0.830	0.956	0.949
clean-cleanx10-clean-noisyx10-plpD2-act-h800-cw5	-	-	-	-	-	-	-	-
clean-cleanx10-noisy-noisyx10-plpD2-act-h800-cw7	25.983	2.117	12.331	0.791	0.472	0.474	0.824	0.784
noisy-noisyx10-noisy-noisyx10-plpD2-act-h800-cw7	25.944	2.062	12.597	0.794	0.485	0.485	0.829	0.787

PROMEDIOS DE LA PRUEBA 6: 7 NN CON NHIDDEN AF 100 Y NHIDDEN CONC 800

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h800-cw5	25.998	1.261	20.645	0.937	0.793	0.794	0.947	0.935
clean-cleanx10-clean-noisyx10-plpD2-act-h800-cw5	25.964	7.994	3.272	0.379	0.125	0.126	0.630	0.406
clean-cleanx10-noisy-noisyx10-plpD2-act-h800-cw5	25.989	2.388	10.915	0.746	0.419	0.420	0.799	0.738
noisy-noisyx10-noisy-noisyx10-plpD2-act-h800-cw5	25.979	2.209	11.785	0.778	0.453	0.453	0.812	0.770

PROMEDIOS DE LA PRUEBA 7: 7 NN CON NHIDDEN AF 100 Y NHIDDEN CONC 1200

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h1200-cw5	25.998	1.235	21.058	0.945	0.810	0.810	0.951	0.943
clean-cleanx10-clean-noisyx10-plpD2-act-h1200-cw5	25.951	7.731	3.376	0.392	0.129	0.130	0.633	0.418
clean-cleanx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.988	2.351	11.090	0.755	0.425	0.427	0.802	0.747
noisy-noisyx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.983	2.231	11.668	0.775	0.448	0.449	0.809	0.767

PROMEDIOS DE LA PRUEBA 8: 7 NN CON NHIDDEN AF 450 Y NHIDDEN CONC 800

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h800-cw5	25.998	1.204	21.612	0.950	0.830	0.831	0.957	0.949
clean-cleanx10-clean-noisyx10-plpD2-act-h800-cw5	25.925	6.698	3.888	0.442	0.149	0.150	0.628	0.448
clean-cleanx10-noisy-noisyx10-plpD2-act-h800-cw5	25.991	2.175	11.979	0.777	0.460	0.461	0.819	0.769
noisy-noisyx10-noisy-noisyx10-plpD2-act-h800-cw5	25.978	2.147	12.146	0.788	0.466	0.467	0.820	0.780

PROMEDIOS DE LA PRUEBA 9: 7 NN CON NHIDDEN AF 450 Y NHIDDEN CONC 1200

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h1200-cw5	25.998	1.213	21.452	0.949	0.825	0.825	0.955	0.947
clean-cleanx10-clean-noisyx10-plpD2-act-h1200-cw5	25.916	6.466	4.015	0.453	0.155	0.154	0.628	0.456
clean-cleanx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.993	2.185	11.934	0.780	0.458	0.459	0.817	0.772
noisy-noisyx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.976	2.147	12.138	0.786	0.466	0.467	0.820	0.779

PROMEDIOS DE LA PRUEBA 10: 8NN NHIDDEN 800 CW5

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h800-cw5	25.999	1.213	21.440	0.950	0.824	0.825	0.955	0.948
clean-cleanx10-clean-noisyx10-plpD2-act-h800-cw5	25.938	7.290	3.617	0.420	0.137	0.139	0.618	0.431
clean-cleanx10-noisy-noisyx10-plpD2-act-h800-cw5	25.986	2.166	12.038	0.784	0.462	0.463	0.819	0.777
noisy-noisyx10-noisy-noisyx10-plpD2-act-h800-cw5	25.979	2.106	12.369	0.794	0.475	0.476	0.823	0.787

PROMEDIOS DE LA PRUEBA 11: 8 NN CON REJECT_LAST

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h800-cw5	25.998	1.208	21.530	0.950	0.828	0.828	0.956	0.948
clean-cleanx10-clean-noisyx10-plpD2-act-h800-cw5	25.917	6.940	3.780	0.432	0.144	0.145	0.623	0.439
clean-cleanx10-noisy-noisyx10-plpD2-act-h800-cw5	25.983	2.165	12.049	0.783	0.462	0.463	0.820	0.776
noisy-noisyx10-noisy-noisyx10-plpD2-act-h800-cw5	25.977	2.101	12.386	0.794	0.476	0.476	0.823	0.787

PROMEDIOS DE LA PRUEBA 12: 8 NN CON NHIDDEN AF 450, NHIDDEN CONC 1200 y cw5

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h1200-cw5	25.998	1.211	21.481	0.950	0.826	0.826	0.955	0.948
clean-cleanx10-clean-noisyx10-plpD2-act-h1200-cw5	25.915	6.495	4.000	0.450	0.154	0.154	0.632	0.455
clean-cleanx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.991	2.180	11.975	0.778	0.459	0.461	0.819	0.771
noisy-noisyx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.980	2.176	11.987	0.782	0.460	0.461	0.817	0.774

PROMEDIOS DE LA PRUEBA 13: 8 NN CON REJECT_LAST, NHIDDEN AF 450, NHIDDEN CONC 1200 y cw5

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h1200-cw5	25.998	1.222	21.298	0.946	0.818	0.819	0.954	0.944
clean-cleanx10-clean-noisyx10-plpD2-act-h1200-cw5	25.936	7.155	3.652	0.424	0.140	0.140	0.620	0.432
clean-cleanx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.993	2.225	11.724	0.773	0.449	0.451	0.813	0.765
noisy-noisyx10-noisy-noisyx10-plpD2-act-h1200-cw5	25.979	2.151	12.135	0.786	0.465	0.467	0.819	0.778

PROMEDIOS DE LA PRUEBA 14: 8 NN CON NHIDDEN AF 450, NHIDDEN CONC 1500 y cw5

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h1500-cw5	25.998	1.213	21.449	0.950	0.825	0.825	0.955	0.948
clean-cleanx10-clean-noisyx10-plpD2-act-h1500-cw5	25.907	6.394	4.070	0.454	0.156	0.157	0.634	0.458
clean-cleanx10-noisy-noisyx10-plpD2-act-h1500-cw5	25.992	2.196	11.887	0.776	0.455	0.457	0.817	0.768
noisy-noisyx10-noisy-noisyx10-plpD2-act-h1500-cw5	25.978	2.159	12.081	0.787	0.463	0.465	0.818	0.779

PROMEDIOS DE LA PRUEBA 15: 8 NN CON NHIDDEN AF 450, NHIDDEN CONC 2000 y cw5

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h2000-cw5	25.998	1.216	21.409	0.948	0.822	0.823	0.955	0.946
clean-cleanx10-clean-noisyx10-plpD2-act-h2000-cw5	25.940	7.216	3.619	0.419	0.139	0.139	0.620	0.427
clean-cleanx10-noisy-noisyx10-plpD2-act-h2000-cw5	25.991	2.198	11.852	0.773	0.455	0.456	0.817	0.766
noisy-noisyx10-noisy-noisyx10-plpD2-act-h2000-cw5	25.983	2.174	11.987	0.781	0.460	0.461	0.817	0.773

PROMEDIOS DE LA PRUEBA 16: 8 NN CON NHIDDEN AF 450, NHIDDEN CONC 1500 y cw7

Classfrs.	k_X	$k_{X\|Y}$	μ_{XY}	$a(P_{XY})$	$a'(P_{XY})$	$q_X(P_{XY})$	1-CEN	MCC
clean-cleanx10-clean-cleanx10-plpD2-act-h1500-cw7	25.999	1.230	21.168	0.942	0.813	0.814	0.953	0.940
clean-cleanx10-clean-noisyx10-plpD2-act-h1500-cw7	25.938	7.170	3.641	0.426	0.139	0.140	0.611	0.430
clean-cleanx10-noisy-noisyx10-plpD2-act-h1500-cw7	25.985	2.217	11.762	0.775	0.451	0.452	0.816	0.767
noisy-noisyx10-noisy-noisyx10-plpD2-act-h1500-cw7	25.974	2.084	12.506	0.793	0.480	0.481	0.827	0.785

GLOSARIO DE ACRÓNIMOS

- **ASR:** Automatic Speech Recognition (en español, esp: Reconocimiento Automático de Habla)
- **HSR:** Human Speech Recognition (esp: Reconocimiento Humano de Habla)
- **AF:** Articulatory Feature (esp: Característica Articulatoria)
- **ANN:** Artificial Neural Network (esp: Red Neuronal Artificial)
- **HMM:** Hidden Markov Model (esp: Modelo Oculto de Markov)
- **MFCC:** Mel Frequency Cepstral Coefficients
- **PLP:** Perceptual Linear Predictive (esp: Predicción Lineal Perceptual)
- **DTW:** Dynamic Time Warping
- **RASTA:** Relative Spectral Transform
- **FFT:** Fast Fourier Transform (esp: Transformada rápida de Fourier)
- **DCT:** Discrete Cosine Transform (esp: Transformada Discreta del Coseno)
- **SVM:** Support Vector Machine (esp: Máquina de Soporte de Vectores)
- **IFFT:** Inverse Fast Fourier Transform (esp: Transformada Discreta Inversa de Fourier)
- **LPC:** Linear Predition Coding
- **MLP:** Multilayer Perceptron (esp: Perceptrón Multicapa)

- **DB CLEAN:** Data Base Clean (esp: Base de datos limpia)
- **DB NOISY:** Data Base Noisy (esp: Base de datos ruidosa)
- **LOO:** Leave One Out
- **GMM:** Gaussian Mixture Model
- **WER:** Word Error Rate (esp: Tasa de Error de Palabra)
- **a, d, p parametres:** Acoustic scale, duration scale and phone detection penalty (esp: escala acústica, escala temporal y detección de fonema)

BIBLIOGRAFÍA Y REFERENCIAS

[1] Odette Scharenborg, Vincent Wan and Roger K. Moore, "*Towards capturing fine phonetic variation in speech using articulatory features*". Speech and Hearing Research Group, Departament of Computer Science, University of Sheffield, UK. January 2007.

[2] Hervé Bourlard and Nelson Morgan, "*Hybrid HMM/ANN Systems for Speech Recognition:Overview and New Research Directions*". In C. L. Giles and M. Gori (Eds.),
Adaptive Processing of Sequences and Data Structures. Springer-Verlag, Berlin, Germany, 1998.

[3] Hynek Hermansky, "*Perceptual linear predictive (PLP) analysis of speech*". Speech Technology laboratory, Division of Panasonic Technologies, California. 1990.

[4] Odette Scharenborg and Martin Cooke, "*Comparing Human and Machine Recognition Performance on a VCV Corpus*". Proc. Workshop on Speech Analysis and Processing for Knowledge Discovery, Aalborg. 2008.

[5] Katrin Kirchhoff, Gernot A. Fink and Gerhard Sagerer, "*Combining acoustic and articulatory feature information for robust speech recognition*". Speech comunication, 2000.

[6] Karen Livescu, Ari Bezman, Nash Borges, Lisa Yung, Özgür Çetin, Joe Frankel, Simon King, Mathew Magimai- Doss, Xuemin Chi, Lisa Lavoie, "*Manual transcription of conversational speech at the articulatory feature level*". in ICASSP, 2007.

[7] Boulard H and Morgan N, "*Hybrid HMM/ANN Systems for Speech Recognition: Overview and New Research Directions*". Technical report, IDIAP, Martigny, Switzerland. Intl. Comp. Science Institute, Berkeley, CA. UC Berkeley, Berkeley, CA, 1998.

[8] Karen Livescu, Özgür Çetin, Mark Haegawa-Johnson, Simon King, Chris Bartels, Nash Borges, Arthur Kantor, Partha Lal, Lisa Yung, Ari Bezman, Stephen Dawson-Haggerty, Bronwyn Woods, Joe Frankel, Mathew Magimai-Doss, Kate Saenko, "*Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU Summer Workshop"*. Technical report, Johns Hopkins University Center for Language and Speech Processing. 2007, in preparation.

[9] Joe Frankel, Mathew Magimai-Doss, Simon King, Karen Livescu and Özgür Çetin, *"Articulatory feature classifiers trained on 2000 hours of telephone speech"*.University of Edinburgh, ICSI, MIT. Submitted to ICASSP, 2007.

[10] Robert Mannel, "*Coarticulation and assimilation"*. Macquarie University, 2008.

[11] Richard P. Lippmann, *"An Introduction to computing with Neural Nets"*. IEEE ASSP Magazine, April, 4-22. 1987.

[12] Xavier Basogain Olabe, "*Redes neuronales artificiales y sus aplicaciones"*. Escuela Superior de Ingeniería de Bilbao. 2013.

[13] J. Trujillano Cabello, M. Badía Castello, J. March Llanes, A. Rodríguez Pozo, L. Serviá Goixart y A. Sorribas Tello, *"Redes Neuronales Artificiales en Medicina Intensiva"*. 2005.

[14] K. Livescu and J.R. Glass, *"Featured-based pronuntiation modeling with trainable asynchrony probabilities"*. In ICSLP, 2004

[15] I-Fan Chen and HSin-Min Wang, *"Articulatory Feature Asynchrony Analisys and Compensation in Detection-Based ASR"*. In Proc. of Interspeech, 2009. Taipei, Taiwan.

[16] Thomas Winkler, *"From Acoustic Mismatch Towards Blind Acoustic Model Selection in Automatic Speech Recognition"*. University of Bonn, 2013.

[17] C. P. Browman and L. Goldstein, *"Articulatory phonology: An overview"*. Haskins Laboratories Status Report on Speech Research, SR-111,112, 23-42.1992

[18] Discusión sobre Nasalidad en http://en.wikipedia.org/wiki/Manner_of_articulation. Visitada por última vez el 12/4/2015.

[19] Daniel Currie Hall. Interactive section. University of Toronto. Visitada por última vez el 12/4/2015.

[20] Wester, M. *"Syllable classification using articulatory-acoustic features"*. In: Proc. Eurospeech, Geneva, Switzerland, pp. 233–236. 2003.

[21] Ladefoged, P., 1982. *"A Course in Phonetics"*, second ed. Harcourt Brace Jovanovich.

[22] Lectura sobre fonación: http://gramatica.usc.es/~gamallo/aulas/linguaespanhola/AparatoFonadorLectura.pdf. Visitada por última vez el 12/4/2015.

[23] Lectura sobre fonación: http://rua.ua.es/dspace/bitstream/10045/4163/8/Lecci%C3%B3n_2.pdf. Visitada por última vez el 12/4/2015.

[24] Lectura sobre fonética: http://www.livingspanish.com/tipos-de-fonetica.htm. Visitada por última vez el 12/4/2015.

[25] Información sobre fonación: http://slideplayer.es/slide/163258/. Visitada por última vez el 12/4/2015.

[26] Yannis Stylianou, Marcos Faundez Zanuy and Anna Eposito, "*Progress in nonlinear speech procesing*". Pag. 191. 2007

[27] Información sobre la base de datos Isolet: https://archive.ics.uci.edu/ml/datasets/ISOLET. Visitada por última vez el 12/4/2015.

[28] Tipos de ruidos utilizados en la base de datos Isolet: http://www1.icsi.berkeley.edu/Speech/papers/eurospeech05-onset/isolet/noises.html

[29] H. Steeneken and F. Geurtsen. Description of the RSG-10 noise data- base. Technical report, TNO Institute for Perception, 1988.

[30] Grupo de habla ICSI: http://www1.icsi.berkeley.edu/Speech/icsi-speech-tools.html. Visitada por última vez el 12/4/2015.

[31] Estructura de redes neuronales:*http://redesneuronalesuat.galeon.com/ESTRUCTURA.html*. Visitada por última vez el 10/4/2015.

[32] Información sobre coarticulación: http://es.wikipedia.org/wiki/Consonante_coarticulada. Visitada por última vez el 12/4/2015.

[33] Crhistyan Czech , Fabian Miodownik y Alexis Ravaschio, "*Reconocimiento de locutores a partir de archivos en formato MP3*". Escuela de ingeniería de telecomunicación de la universidad de Vigo. PFC. 2005.

[34] K. Kirchhoff, "*Robust Speech Recognition Using Articulatory Information*". Ph D. dissertation, University of Bielefield, 1999.

[35] Ben Gold and Nelson Morgan, "*Speech and Audio Signal Processing*". John Wiley and sons, 2000.

[36] Ludek Muller Josef Psutka and Josef V. Psutka. "*Comparison of MFCC and PLP Parameterizations in the Speaker Independent Continuous Speech Recognition Task*". Technical report, University of West Bohemia, Czech Republic, 2001.

[37] Schwarz, P., Matejka, P. and Cernocky, J. *"Towards Lower Error Rates in Phoneme Recognition"* in Proc. TSD2004, Brno, Czech Republic, 2004.

[38] Francisco J. Valverde-Albacete and Carmen Peláez-Moreno, *"Two informations theoretic tools to assess the performance of multi-class classifiers"*. Universidad Carlos III de Madrid, Spain. Pattern Recognition Letters 31: 1665–1671. doi: 10.1016/j.patrec.2010.05.017, 2010.

[39] Francisco J. Valverde-Albacete and Carmen Peláez-Moreno, *"100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains The Accuracy Paradox"*. (Open Access) PLoS One, 9(1) doi:http://dx.doi.org/10.1371/journal.pone.0084217. 2014.

[40] Breve resumen de Reconocedores de voz en http://www.monografias.com/trabajos96/reconocimiento-voz/reconocimiento-voz.shtml. Visitada por última vez el 10/4/2015.

[41] Breve resumen de Clasificadores de voz en rua.ua.es/dspace/bitstream/10045/16038/.../RUA%20-%20Tema%206.ppt. Visitada por última vez el 10/4/2015.

[42] PFC Leticia Rueda, *"Mejoras en reconocimeinto del habla basadas en mejoras en la paramentrización de la voz"*. Abril 2011 : http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20110603LeticiaRueda.pdf