Universidad Autónoma de Zacatecas
"Francisco García Salinas"
Unidad Académica de Ingeniería Eléctrica
Doctorado en Ciencias de la Ingeniería
(DOCII)

Reconocimiento de voz a través de técnicas híbridas utilizando modelos Markovianos y nuevos tipos de redes neuronales

Que como parte de los requisitos
para obtener el grado de:

Doctor en Ciencias de la Ingeniería

presenta:

Aldonso Becerra Sánchez

Zacatecas, Zac. Noviembre de 2017

Universidad Autónoma de Zacatecas

"Francisco García Salinas"

Unidad Académica de Ingeniería Eléctrica

Doctorado en Ciencias de la Ingeniería

(DOCII)

Reconocimiento de voz a través de técnicas híbridas utilizando modelos Markovianos y nuevos tipos de redes neuronales

Que como parte de los requisitos
para obtener el grado de:

Doctor en Ciencias de la Ingeniería

presenta:

Aldonso Becerra Sánchez

dirigido por:

Dr. José Ismael de la Rosa Vargas

Sinodales:

Dr. José Ismael de la Rosa Vargas **Director de tesis**	Firma
Dr. Efrén González Ramírez **Co-director de tesis**	Firma
Dr. José de Jesús Villa Hernández **Vocal**	Firma
Dr. Daniel Alaniz Lumbreras **Vocal**	Firma
Dr. Antonio de Jesús Balvantín García **Vocal**	Firma

Dr. Jorge de la Torre y Ramos Director de la Unidad Académica de Ingeniería	Dr. José Ismael de la Rosa Vargas Responsable del Programa de Doctorado

Zacatecas, Zac. Noviembre de 2017

Resumen

El módulo de reconocimiento de voz dentro de un sistema de dialogo hablado se ha convertido en un punto clave con el paso del tiempo. Las mejoras que se le pueden hacer con los nuevos enfoques y técnicas han mostrado el camino evolutivo que se puede dar en muchos procesos de entrenamiento y definición de arquitecturas con el fin de obtener mejores tasas de reconocimiento. En este sentido, el presente trabajo tiene como objetivo investigar esquemas que permitan mejorar las tasas de error por palabra (WER). El trabajo se fundamenta en la idea del uso de la arquitectura de red neuronal profunda y modelos ocultos de Markov (RNP-MOM), la cual se basa en gran medida en el comportamiento del enfoque de modelo de mezclas Gaussianas y modelos ocultos de Markov (MMG-MOM). En primera instancia se hacen comparaciones experimentales en el funcionamiento de ambos enfoques tomando como punto de partida un corpus de voces personalizado en Español de la parte norte central de México, basado en una tarea de marcado telefónico a través de reconocimiento de dígitos numéricos y nombres completos de personas, con independencia de locutor, con dependencia de texto, de tamaño mediano y con palabras conectadas. En el primer caso de estudio experimental se obtuvo una mejora relativa del $30\,\%$ usando el modelo acústico de redes neuronales (WER de $1.49\,\%$), en comparación con el modelo clásico de mezclas Gaussianas ($2.12\,\%$). En el segundo caso de estudio se consiguió una mejora relativa de $20.71\,\%$ en la tasa de error por palabras del enfoque conexionista (redes neuronales, WER de $3.33\,\%$) con respecto al modelo de mezclas Gaussianas ($4.20\,\%$). En las tareas de reconocimiento presentadas se muestra que los enfoques actuales cimentados en modelos conexionistas, con origen en la in-

teligencia artificial, superan en la mayoría de los procesos de reconocimiento a los enfoques tradicionales de mezclas Gaussianas. Con el fin de conseguir mejoras en los modelos recientes de reconocimiento de voz, en la segunda parte del trabajo se proponen nuevas funciones de costo para entrenar una red neuronal, denominando a estas funciones como mapeadas no uniformes. Estas funciones permiten obtener mejores tasas de reconocimiento en comparación con la función convencional de entropía cruzada dentro del entrenamiento de una red neuronal profunda, utilizando para ello el algoritmo de retro-propagación y una optimización con el gradiente descendente. Los resultados obtenidos (se consiguió una mejora relativa de 12.3% y 10.7% con los dos enfoques planteados, con respecto al modelo base de entropía cruzada) han mostrado mejoras en las tasas de error por palabra, sugiriendo que las funciones de costo propuestas tienen argumentos para ser consideradas como alternativas interesantes en este tipo de tareas. No obstante, se debe seguir en la labor de probar este y nuevos mecanismos de función de costo con diferentes corpus de voces y en diversos entornos con y sin ruido ambiental, además de considerar variaciones radicales en los origenes de voz de los locutores.

Abstract

The speech recognition module within a spoken dialogue system has become a key factor over time. The improvements that can be made with the new approaches and techniques have shown the evolutionary path that can be carried out in many processes of training and architecture definition in order to obtain superior recognition rates. In this sense, the present research has as objective to investigate new schemes to improve the word error rates (WER). The present work is based on the idea of using the deep neural networks and hidden Markov models (DNN-HMM) architecture, which relies heavily on the behavior of the Gaussian mixture models and hidden Markov models (GMM-HMM) approach. First, experimental comparisons are made taking into consideration both approaches. The research process has been performed by using a corpus of personalized voices in Spanish from the northern central part of Mexico, based on a connected-words phone dialing task through the recognition of digit strings and personal name lists. The specified recognition task is defined as speaker-independent, text-dependent and mid-vocabulary. In the first experimental case study, a relative improvement of $30\,\%$ was obtained using the acoustic model based on neural networks (WER of $1.49\,\%$), compared to the classic acoustic model based on Gaussian mixtures ($2.12\,\%$). In the second case study, a relative improvement of $20.71\,\%$ was achieved with the connectionist approach (neural networks, WER of $3.33\,\%$) with regard to the Gaussian mixture model ($4.20\,\%$). The presented recognition task shows that the current approaches based on connectionist models, originated in artificial intelligence, surpass the traditional approaches of Gaussian mixtures in most of the speech recognition tasks. With the purpose of obtaining improvements in

the recent speech recognition models, the second part of the thesis proposes new cost functions to train a neural network, calling these functions as non-uniform mapped criteria. These functions allow superior recognition rates in comparison with the conventional cross-entropy function within the training of a deep neural network, by using the back-propagation algorithm and an optimization with the gradient descent procedure. The obtained results (a relative improvement of 12.3% and 10.7% was achieved with the two proposed approaches, with respect to the conventional model of cross-entropy) have shown improvements in the word error rates, suggesting that the proposed cost functions have arguments to be considered as interesting alternatives in this type of tasks. Nevertheless, we must continue with the work of testing this and new cost function mechanisms with different voice corpus in several conditions with and without environmental noise, in addition to considering radical variations in the speakers' speech sources.

Agradecimientos

El presente trabajo de tesis es un esfuerzo en el cual participaron muchas personas de forma directa o indirecta, leyendo, opinando, corrigiendo, dando ánimo, acompañando en los momentos débiles y en los momentos fuertes, a ellos quiero agradecer que por fin se haya concluido satisfactoriamente esta investigación.

Agradezco primero a la Universidad Autónoma de Zacatecas por permitirme realizar mis estudios Doctorales, a mis maestros por haber compartido conmigo sus enseñanzas, consejos, conocimientos y ánimos para seguir adelante. En este sentido también quiero agradecer al CONACYT por el apoyo proporcionado durante la estancia doctoral.

Gracias a mis asesores de Tesis, el Dr. José Ismael de la Rosa y Dr. Efrén González Ramírez por su paciencia y experiencia transmitida. A los Doctores José de Jesús Villa, Daniel Alaniz y Antonio de Jesús Balvantín por sus observaciones pertinentes y acertadas.

A todos ellos gracias.

VI

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

A la cadena de eventos desde la concepción del mensaje en el cerebro del locutor hasta su recepción en el cerebro del escucha se le denomina *cadena de habla*. Tratando de emular este mecanismo, los sistemas de reconocimiento automático de voz (RAV) están destinados a producir sequencias de palabras (transcripción) de una señal de voz grabada a partir de cualquier tipo de micrófono. Con el paso del tiempo las mejoras tecnológicas en el reconocimiento de voz han sido enfocadas en cuatro directrices [1]: a) procesamiento en el *front-end*, b) modelado acústico, c) modelado del lenguaje, y d) búsqueda de la hipótesis y la combinación de sistemas.

La mayoría de los sistemas recientes de reconocimiento de voz modelan la variabilidad de la señal de voz a través de los modelos ocultos de Markov (MOM), y establecen el grado de concordancia de las observaciones acústicas de entrada (una representación corta de la voz humana) a un estado del modelo de Markov usando modelos de mezclas Gaussianas (MMG). Este enfoque ha sido usado en el reconocimiento de voz por varios años [2–4]. Los modelos de mezclas Gaussianas son modelos muy eficientes, y es complicado que otro enfoque pueda alcanzar una mejor precisión en el *modelado acústico*. Además, estos pueden ser ajustados de manera discriminativa después de que han sido generativamente entrenados con el propósito de maximizar su probabilidad [5]. A pesar de que estos modelos son adecuados para generar los datos observados en cada estado del modelo de Markov, los modelos de mezclas Gaussianas tienen un defecto serio: son ineficientes para modelar datos que se encuentran en o cerca de un

conducto múltiple no lineal en el espacio de datos [6].

Una manera alternativa de llevar a cabo la función de las mezclas Gaussianas es utilizar redes neuronales de alimentación hacia adelante (*feed-forward*), pero especificamente hablando las denominadas redes neuronales profundas (RNPs), las cuales tienen el potencial de aprender mejores modelos de datos que se encuentran en o cerca de un conducto múltiple no lineal [6]. El reciente modelado acústico sustentado en el uso de redes neuronales ha evidenciado que puede mejorar a las mezclas Gaussianas en la mayoría de las tareas de reconocimiento [7–13]. Aunque dos décadas atrás las redes neuronales de una sola capa fueron empleadas en tareas de reconocimiento de voz [14, 15], ni el hardware ni los algoritmos de entrenamiento eran apropiados en grandes cantidades de datos o varias capas ocultas. Pero hoy en día los avances en aprendizaje-máquina y hardware de computadora hacen posible superar esas limitaciones [16].

Una consideración importante en el modelado con redes neuronales es el actual proceso de aprendizaje en dos etapas [10, 17]: i) inicialización de los pesos (con o sin pre-entrenamiento) y ii) el entrenamiento con el algoritmo discriminativo de retro-propagación (*fine-tuning*). Un paso de gran ayuda en el resurgimiento de las redes neuronales es el pre-entrenamiento, una forma alternativa a la inicialización aleatoria clásica de los pesos, la cual puede ser llevada a cabo mediante algunas estrategias como pre-entrenamiento discriminativo (DPT) [7], pre-entrenamiento con redes de creencia profunda (DBN) [9, 18, 19], *denoising autoencoder* [20, 21], pre-entrenamiento híbrido [22] y *dropout* [20, 23, 24].

Un elemento esencial en un sistema de diálogo hablado es el módulo de reconocimiento de voz, el cual debe operar sin ninguna ayuda humana (acarreando el término de automático). Este módulo proporciona un enlace fundamental entre la computadora y el usuario final. Con el propósito de que los sistemas de diálogo sean cada vez mejor adaptados a la interacción entre el usuario y la interfaz, se deben seguir desarrollando técnicas que permitan alcanzar tasas altas de interacción efectiva.

Bajo este sentido, partiendo del hecho de que diversos investigadores están en un proceso de búsqueda constante de mejorar las tasas actuales de reco-

nocimiento de voz, principalmente basándose en modelos conexionistas (redes neuronales), el objetivo del presente trabajo es revisar dos de las metodologías principales para el reconocimiento de voz, para posteriormente poder analizar y definir una propuesta o sugerencia que permita brindar alternativas metodológicas para disminuir las tasas de error en el reconocimiento automático de voz, parte importante de un sistema de diálogo hablado convencional. Para poder hacer esta aportación, se parte de la introducción del enfoque convencional de MMG-MOM (modelo de mezclas Gaussianas-modelo oculto de Markov), y la contraparte del modelo no lineal reciente de RNP-MOM (red neuronal profunda-modelo oculto de Markov); donde el principal cambio entre dichos sistemas está localizado en el *modelado acústico*. De esta forma, se plasman en el presente trabajo de investigación dos propuestas basadas en modificar la función objetivo del proceso de entrenamiento de una red neuronal. En primer lugar, se define la función denominada *entropía cruzada mapeada no uniforme*, la cual busca minimizar la ambigüedad de la pertenencia de una observación acústica de entrada a un estado de MOM, basándonos en la información que nos proporcionan los estados competidores; para ello hacemos uso del concepto de extropía. En segundo término, se hace la fusión de la *entropía cruzada mapeada no uniforme* con un concepto denominado en la literatura como *entropía cruzada enfatizada*, la cual busca dar prioridad o enfatizar aquellas observaciones acústicas difíciles de clasificar, desenfatizando aquellas observaciones más precisamente clasificadas. Esta fusión permite definir una segunda propuesta denominada *entropía cruzada mapeada impulsada o enfatizada*. Estos lineamientos propuestos serán descritos con detalle en el capítulo 4.

1.1. La señal de voz, producción, percepción y caracterización acústico-fonética

En este apartado se mencionan algunos fundamentos de la producción y caracterización de las señales de voz.

1.1.1. La señal de voz

Aunque el problema de generación de voz no está totalmente bien comprendido, las señales de voz son generadas por movimientos coordinados de diferentes partes de la anatomía humana. A través del control del diafragma y de los músculos conectados a la cavidad bucal, el aire a presión puede ser concentrado debajo de la laringe, y provee energía para producir las señales de voz. Los sonidos de la voz son producidos por la posición de las cuerdas vocales en la laringe, cuando son puestas en vibración por el flujo de aire. Los sonidos no vocalizados (silencios) son producidos sin la vibración de las cuerdas vocales y a menudo implican la generación de turbulencia en el tracto vocal. De esta manera, las características acústicas de los sonidos resultan de los cambios en la posición de la lengua, la mandíbula y los labios.

La información que es comunicada a través de la voz es intrínsecamente discreta y está compuesta por símbolos llamados “fonemas”. Cada una de las lenguas existentes tienen su propio conjunto particular de fonemas y los rangos de los conjuntos oscilan de 30 a 60 fonemas.

Puesto que la voz humana está representada por la combinación de sonidos de frecuencias diferentes, estos se obtienen haciendo vibrar las cuerdas vocales cerca de 100 veces por segundo para los hombres y 200 veces por segundo para las mujeres. La excitación de estas vibraciones causa resonancia en el tracto vocal en un rango de 200 Hz a 5000 Hz. La voz humana contiene información lingüística dentro del rango de 200 Hz a 8000 Hz. En general, existen 2 condiciones que deben ser imperativamente respetadas en un sistema de comunicación de voz [25, 26]:

1. Se debe preservar el contenido del mensaje en la señal de voz.

2. La representación de una señal de voz debe hacerse en una forma conveniente para transmitirla o almacenarla.

1.1.2. Manipulación y aplicación de una señal de voz

La manipulación de una señal de voz se puede llevar a cabo una vez que convertimos la señal acústico-fonética en una señal eléctrica continua, luego se transforma en una señal discreta y posteriormente la tratamos matemáticamente [25] (ver Figura 1.1).

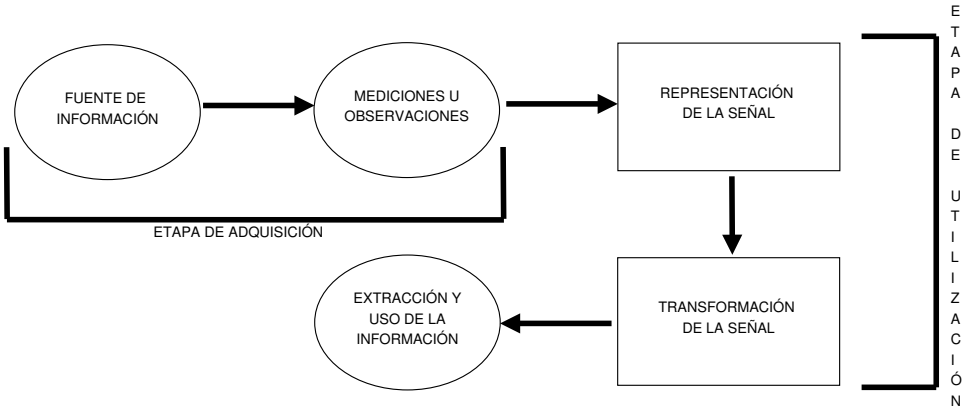

Figura 1.1: Esquema general de procesamiento de una señal de voz

La extracción y la utilización de la información contenida en un mensaje se lleva a cabo por escuchas, pero la meta final es que esta se haga mediante máquinas (de forma automática); de tal manera, los propósitos pueden ser:

1. Identificación de locutor.
2. Verificación de locutor.
3. Reconocimiento de texto, por mencionar algunos.

1.1.3. El proceso de producción de voz

El proceso de producción de voz comienza cuando el locutor formula un mensaje que él quiere transmitir a un escucha por medio del habla. En este sentido la comunicación oral es la transferencia de información de una persona a otra por medio de una señal acústica denominada voz, la cual consiste en variaciones

de presión generadas por el tracto vocal del locutor. Tales variaciones de presión se propagan como ondas a través del aire y alcanzan el oido de los escuchas, quienes procesan y decifran las ondas convirtiéndolas en un mensaje (ver Figura 1.2) [27, 28]. Una vez que la señal de voz es generada y propagada, comienza el

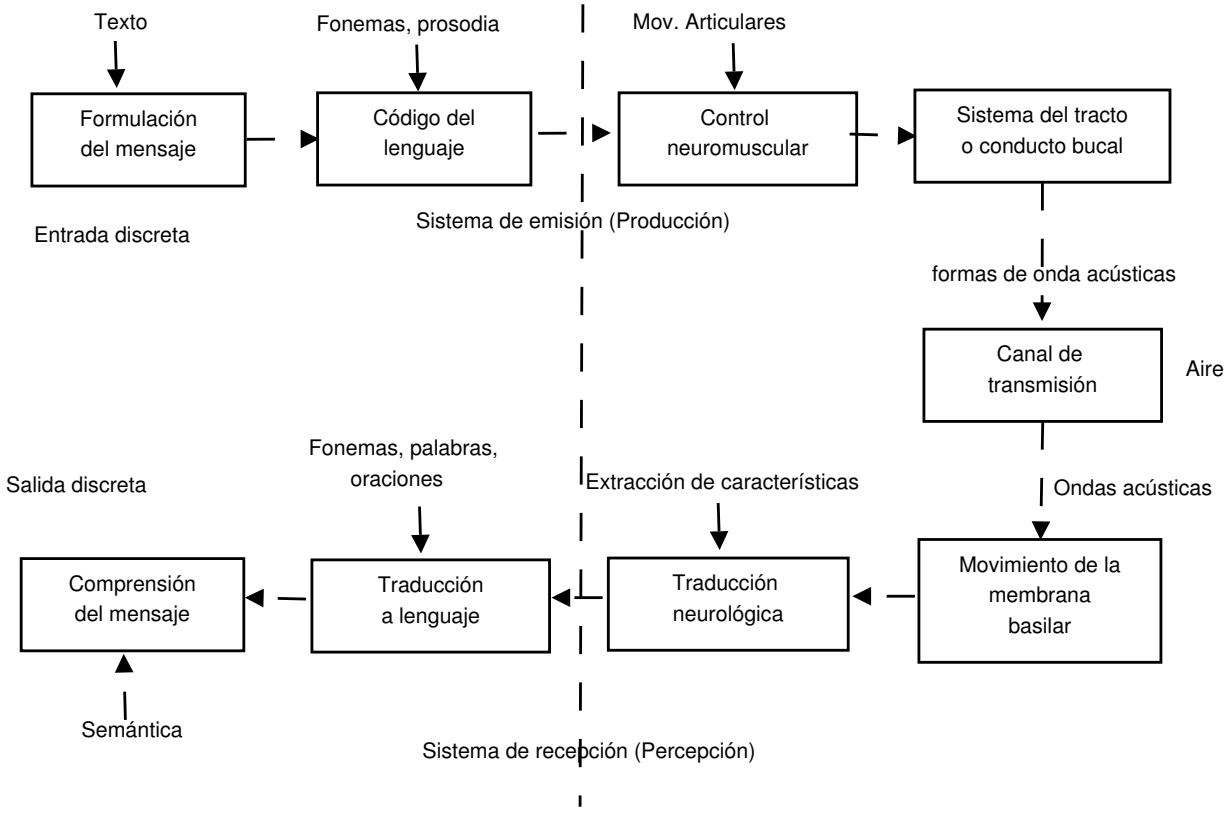

Figura 1.2: Proceso de producción y percepción de voz entre dos personas

proceso de percepción (reconocimiento de voz). En esta etapa se procesa la señal acústica a lo largo de la membrana basilar, que proveé un análisis espectral de la señal de entrada. El proceso de traducción neurológico convierte este espectro en señales que activan el nervio auricular (extracción de características). Al final de dicha actividad neurológica, entorno al nervio auricular, la señal es convertida en un código de lenguaje y finalmente se lleva a cabo la comprensión del mensaje [25].

A la cadena de eventos desde la concepción del mensaje en el cerebro del locu-

tor hasta el arribo del mensaje al cerebro del escucha se le conoce como *cadena de habla*.

El proceso de producción del habla se resume en [29]:

1. El tracto vocal, que comienza con la apertura de las cuerdas vocales o glotis y termina en los labios, está compuesto por: la faringe (conexión entre esófago con la boca) y la boca o cavidad oral.

2. El tracto o conducto nasal comienza con el velum y finaliza en los orificios nasales. Cuando el velum se mueve hacia abajo, la cavidad nasal se acopla con el tracto vocal para producir sonidos nasales.

Por ende el sistema de producción de voz (locutor) se puede dividir en tres grupos de órganos que participan (ver Figura 1.3 y 1.4) [25, 26]:

1. Órganos de respiración: pulmones, bronquios, tráquea.

2. Órganos de fonación: laringe, cuerdas vocales y glotis.

3. Órganos articulatorios: labios, mandíbula, dientes, velum, faringe, cavidad oral y cavidad nasal.

Previa inhalación del aire en los pulmones, la voz se produce generalmente cuando el aire está inhalado; al expanderse el diafragma, los órganos de respiración proporcionan un flujo de aire a los órganos de fonación, es ahí donde el sonido adquiere sus características primarias y en donde las cuerdas vocales juegan un rol importante. Luego se agregan otras características impuestas por los órganos articulatorios, brindando una señal acústico-fonética.

1.1.4. Sonidos de voz y sus características

Los órganos de respiración proporcionan un flujo de aire a los órganos de fonación, donde el sonido adquiere sus características primarias, al cual se le agregan propiedades adicionales por los órganos de articulación, al restringir el

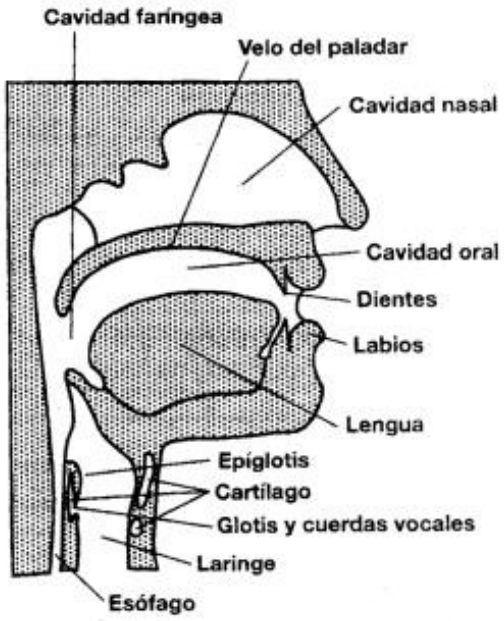

Figura 1.3: Esquema del aparato fonatorio humano

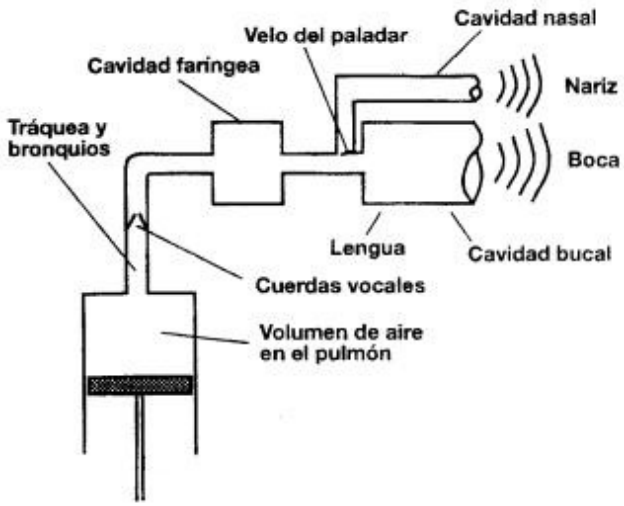

Figura 1.4: Modelo físico del sistema fonatorio

paso del aire en algún punto del tracto vocal, dientes, etc.

Fonemas. Las "unidades sonoras" que se utilizan para el habla son llamadas "fonemas" y estos dependen de las características impuestas según sea el lenguaje. Por convención los fonemas están denotados entre diagonales y se dividen en varios grupos [25, 27, 28]:

1. Vocales orales: $/a/, /e/, /i/, /o/, /u/$.
2. Vocales nasales: $an \rightarrow /\hat{a}/, on \rightarrow /\hat{o}/$.
3. Semi-vocales: $/w/, /r/, /y/$.
4. Diptongos: $/ay/, /ey/, /au/$.
5. Consonantes nasales: $/n/, /m/$.
6. Fricativas (vocalizadas:$/v/, /j/$ y no vocalizadas: $/f/, /s/$).
7. Plosivos (vocalizados: $/b/, /d/, /g/$ y no vocalizados: $/p/, /t/, /k/$).
8. Líquidos: $/l/$.

1.1.5. Esquemas a considerar en la representación de la señal de voz

La señal de voz es una señal que varía lentamente en el tiempo, es decir, cuando esta es examinada sobre un periodo suficientemente pequeño (ente 5 y 100 ms), sus características son cuasi-estacionarias. Sin embargo, sobre largos periodos de análisis (0.2 segundos en adelante) las características de la señal cambian reflejando los diferentes sonidos que serán vocalizados o no vocalizados; para estos largos periodos la señal de voz es no estacionaria. En la práctica, para poder analizar la señal de voz, es necesario considerar una segmentación de la señal en periodos lo suficientemente pequeños, de tal manera que se cumpla con la condición de estacionariedad o cuasi-estacionariedad [25].

Existen dos grupos de métodos clásicos para el análisis de señales de voz. El primero está fundamentado en el análisis puramente temporal de $x(t)$, mientras

que el segundo se basa en una representación espectral de $X(\omega)$, para el cual es necesario utilizar los conceptos de transformada de Fourier de $x(t)$. Esta última representación saca ventaja del contenido frecuencial y de cómo se reparte la energía en el espectro (cómo varía la forma espectral). De la cual podemos tener la energía instantánea ($|x(t)|^2$) y la espectral ($|X(\omega)|^2$). La señal de voz humana, de vocalización de aves, etc., tienen la particularidad de presentar modulaciones en frecuencia, lo cual quiere decir que son señales no estacionarias. El análisis espectral clásico que se fundamenta en el análisis de Fourier, implica que las características espectrales de la señal son estacionarias. Una forma de preservar la resolución en frecuencia es analizar la señal de voz observándola en duraciones cortas, lo cual es el principio de introducción de ventanas de análisis. Este método de ventaneo permite reducir los efectos de las variaciones temporales y limita el efecto de la no estacionariedad.

1.2. El reconocimiento de voz en un sistema de diálogo hablado

Es conocido que los sistemas de diálogo para la interacción hombre-máquina fundamentan su propósito en lograr un flujo de señales de voz en ambas direcciones, logrando mediante el habla una comunicación entre ambas entidades. La idea base se enfoca en lograr que el usuario se comunique con la máquina y esta tenga la capacidad de identificar lo que se le está diciendo y esté en posibilidad de responder apropiadamente, habitualmente mediante una señal acústica de salida. En todo este proceso obviamente lo ideal es que se pueda realizar sin importar el tópico del que se trate, aunque normalmente se hace basándose en cierto contexto o escenario específico. El reconocimiento de voz es el módulo de entrada de un sistema de diálogo y su relevancia es decisiva en el funcionamiento global del sistema.

1.2.1. Componentes de un sistema de diálogo

Un sistema de diálogo está compuesto por los siguientes elementos (ver Figura 1.5) [30]:

1. *Módulo de reconocimiento de voz (SR - Speech Recognition)*. El propósito del módulo es reconocer las señales de voz del locutor, teniendo como salida un texto equivalente a lo hablado por el usuario.

2. *Módulo de procesamiento de lenguaje natural (NLP - Natural Language Processing)*. Teniendo como entrada la salida del SR, este módulo obtiene una idea de lo que el usuario intentó decir, buscando dentro de un catálogo de sentencias la que corresponde, dando como salida el significado.

3. *Módulo de manejo de diálogo (DM - Dialogue Management)*. Teniendo como entrada el significado de lo que el usuario intentó decir con su señal de voz, proporciona como salida una respuesta a lo que el usuario solicita; esta salida es proporcionada en forma de texto.

4. *Módulo de conversión de texto a voz (TTS - Text-To-Speech)*. Teniendo como entrada la cadena de texto del módulo de DM, proporciona una señal de voz equivalente, la cual es emitida al usuario.

Este trabajo se enfoca exclusivamente en el módulo de SR (una generalización de este módulo es el RAV - reconocedor automático de voz), el cual está compuesto por las siguientes fases generales [2, 3, 28, 31, 32]:

1. Preparación de la base de datos o corpus de voces.

2. Procesamiento de la señal de voz.

3. Generación de modelos y entrenamiento.

4. Decodificación de la señal recibida.

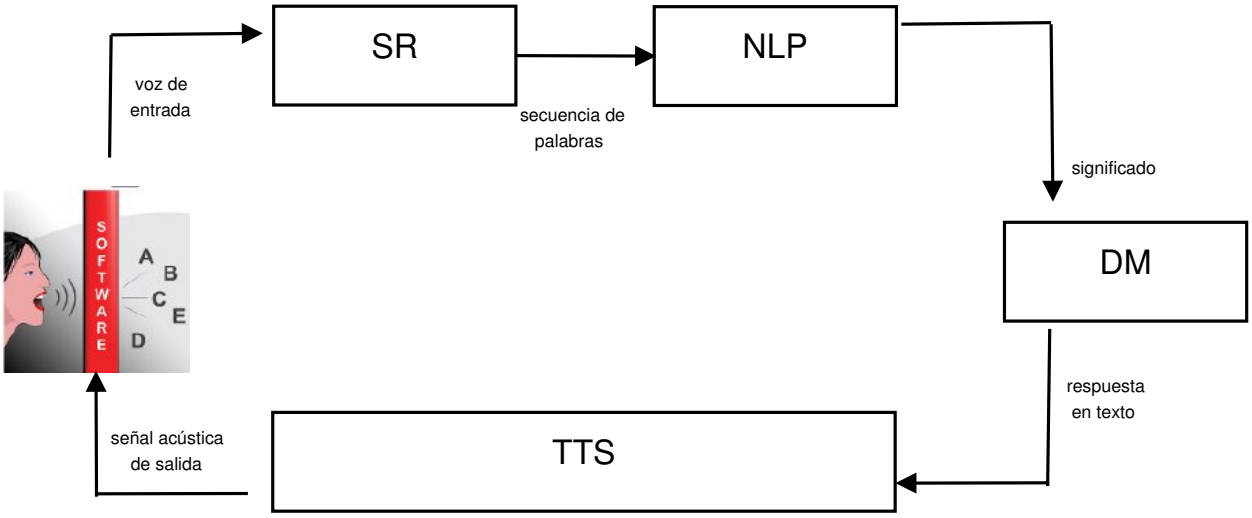

Figura 1.5: Esquema general de un sistema de diálogo hablado

El proceso de reconocimiento de voz es complejo de manera natural, ya que involucra muchos factores asociados y características implícitas en ellos, por ejemplo, se debe considerar el idioma hablado, reglas gramaticales y pronunciaciones; además se debe tener en cuenta que siempre suelen existir complicaciones como el ruido ambiental (ruidos de computadoras, impresoras, máquinas de escribir, teléfonos, conversaciones de personas, automóviles, por mencionar algunos), la distorsión (ocasionada por el micrófono principalmente) y los ruidos articulatorios del locutor (la manera de hablar de la o las personas involucradas en el entrenamiento y pruebas del sistema, así como estados de ánimo).

1.2.2. Consideraciones de un sistema de RAV

El problema del reconocimiento automático de voz (RAV) se puede dividir en dos tareas principales [33]:

1. Reconocimiento automático del habla (RAH). Se puede considerar como la generalización del reconocimiento automático de voz y su propósito es reconocer la secuencia de palabras equivalentes al mensaje de voz emitido por el usuario.

2. Reconocimiento automático de locutor (RAL). Estos sistemas pueden ser dependientes o independientes del vocabulario; es decir, en el primer caso, el locutor debe articular un mensaje fijo que ha sido determinado con anterioridad, mientras que en el segundo, el locutor dispone teóricamente de la completa libertad para seleccionar el texto de su mensaje de reconocimiento.

   a) Identificación automática de locutor (IAL). Debe hacer un juicio razonablemente exacto acerca de quién, de una lista de locutores potenciales está hablando.

   b) Verificación automática de locutor (VAL). Debe determinar si un locutor es realmente quien dice ser.

Adicionalmente, es importante tomar en cuenta algunas cuestiones relacionadas con la manera en que se llevará a cabo el proceso de reconocimiento, entre las cuales se resaltan [31, 34]:

- *Dependencia o independencia de locutor.* Dependiendo de si un sistema se considera para un único locutor o no importe quién o quiénes sean los usuarios del mismo.

- *Tamaño del vocabulario.* Es habitual distinguir un sistema por la longitud de las palabras que es capaz de reconocer, por ejemplo, es pequeño si a lo mucho contempla 100 palabras, mediano si está entre 101 y 999, o grande si su funcionamiento se basa en reconocer palabras en cantidades mayores a 1000. Considerando este tamaño, es recomendable seleccionar las unidades lingüísticas a usar, por ejemplo, si es pequeño es habitual usar “palabras”, si es mediano puede ser “sílaba” o “fonema”, y en dado caso que sea extenso es preferible utilizar “fonemas” o “fonos”.

- *Velocidad del habla.* Si se utilizan pausas prolongadas entre cada palabra, por ejemplo valores de cientos de milisegundos, se dice que es reconocimiento de palabras aisladas, en caso que sea una emisión de habla sin pausas extensas se conoce como sistema de habla continuo.

- *Contexto de uso del sistema*. Es natural pensar que un sistema de reconocimiento tenga más eficiencia si se utiliza en un entorno de aplicación específico, reduciendo drásticamente el número y contexto de las palabras involucradas.

1.3. Métodos de análisis para el reconocimiento de voz

En este apartado se mencionan algunas de las principales técnicas y líneas de trabajo sobre las que se fundamenta el reconocimiento de voz.

1.3.1. Primera aproximación del reconocimiento automático de voz

El RAV se puede considerar un área de la inteligencia artificial que tiene como tarea convertir una señal de voz a su transcripción de texto utilizando para ello la computadora. Para ir abordando este tema se irán mostrando algunos tópicos relevantes.

El habla está representada a través de una secuencia de vectores denominados *observaciones acústicas* [35], que son el resultado de un proceso de extracción de características de la señal acústica digitalizada. Una observación acústica O está definida por la siguiente secuencia de vectores $O = \{\vec{o}_1, \vec{o}_2, ..., \vec{o}_T\}$; donde $\vec{o}_t$ es el vector de habla en el tiempo t. Dada una observación acústica, el objetivo del reconocimiento de voz es encontrar la correspondiente secuencia de palabras $W = \{w_1, w_2, ..., w_m\}$.

El reconocimiento automático del habla habitualmente se formula basándose en el teorema de Bayes, quedando su solución definida por la ecuación:

$$\widehat{W} = \underset{w}{\operatorname{argmax}}\ p(W|O) = \underset{w}{\operatorname{argmax}}\ \frac{p(W)p(O|W)}{p(O)}, \tag{1.3.1}$$

donde $\widehat{W}$ es la secuencia de palabras con mayor probabilidad dadas las observaciones O, debido a que la maximización de la ec. (1.3.1) es realizada manteniendo la observación acústica O fija, la ecuación se reformula como:

$$\widehat{W} = \underset{w}{\operatorname{argmax}}\ p(W|O) = \underset{w}{\operatorname{argmax}}\ p(W)p(O|W), \tag{1.3.2}$$

donde el término $p(O|W)$ es conocido como *modelo acústico* y el término $p(W)$ es denominado *modelo de lenguaje* (ver Figura 1.6) [2, 3, 28, 32, 34].

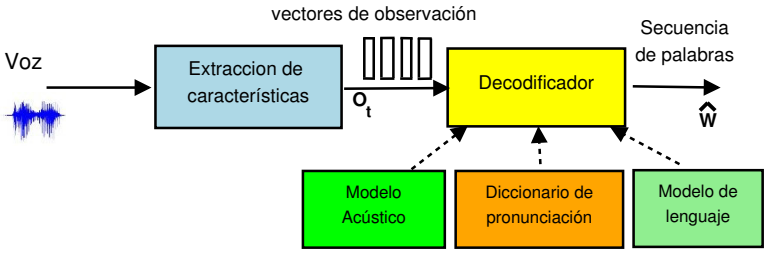

Figura 1.6: Arquitectura general del modelo de reconocimiento de voz (SR)

En la Figura 1.6 se puede observar el esquema general de un sistema de reconocimiento. Primeramente, la señal de voz (forma de onda) es pre-procesada usando un *audio front-end*, que extrae las características acústicas de dicha señal (por ejemplo, secuencia de tramas de características de igual duración). Algunos ejemplos de estas características son FBANK, MFCC, MFSC, LPC, PLP, entre otras [36]. El modelo acústico (*acoustic model*) es un modelo estadístico de las características extraídas por el *front-end*. Típicamente este modelo es un modelo generativo (como MMG) de las características condicionadas de diferentes clases de sonidos hablados o *fonemas*. Este modelo es usado para calcular la probabilidad de generar una observación acústica de una transcripción hipotética. El diccionario de pronunciación (*pronunciation dictionary*) mapea cada palabra en el lenguaje elegido conforme a su pronunciación (secuencia de fonemas que componen cada palabra). Normalmente una palabra tiene varias formas de pronunciarse, en cada caso el diccionario deberá contener varias entradas para esa palabra. Este diccionario es usado para proporcionar restricciones de las secuencias de sonidos que son posibles. El modelo del lenguaje (*language model*) es un modelo

estadístico de la secuencia de palabras en el lenguaje elegido y es usado para proporcionar una clasificación relativa a las secuencias de palabras. La mayoría de los reconocedores utilizan modelos de lenguaje de *n-gramas* debido a la facilidad de incorporarse con otros componentes. El decodificador (*decoder*, denotado por argmax_w) es un módulo que transcribe una cadena de habla [3], y lo hace generando transcripciones candidatas con respecto a las restricciones del diccionario y al modelo de lenguaje. Cada candidato es calificado usando los puntajes del modelo acústico, representando la probabilidad de observación dado el candidato, y se califica tomando como base el modelo de lenguaje, representando una probabilidad previa del candidato (una secuencia de palabras que representa la transcripción).

Métodos para el reconocimiento automático de voz

Podemos decir que existen cuatro *metodologías* fundamentales en reconocimiento del habla y que corresponden con otras tantas clases de técnicas [25, 34]:

1. *Reconocimiento de patrones*. La aproximación basada en el contraste de patrones supone que la oración formulada puede representarse como una secuencia de unidades del habla (palabras), cada una representada por un cierto patrón o conjunto de patrones. Se establece un criterio de distancia o similitud que permita comparar una unidad de habla con cada uno de los patrones de referencia almacenados y que sirva para determinar el patrón que mejor se ajuste, en algún sentido, a la unidad de entrada.

2. *Sistemas basados en el conocimiento*. Los sistemas basados en el conocimiento tratan de emular un conjunto de conocimientos sobre el habla, puesto en juego por un ser humano en su tarea de comprensión de un discurso. Para ello hace uso de técnicas de construcción de sistemas basados en reglas y sistemas expertos, desde el mismo nivel acústico-fonético, o bien desde niveles superiores.

3. *Modelos estocásticos*. Supone un avance en la capacidad de generalización. Una característica diferenciadora es la utilización de modelos probabilísticos

en lugar de determinísticos, teniendo así capacidad de integración para una solución simultánea del problema de segmentación y el de clasificación.

4. *Modelos neuronales o conexionistas.* Alternativa capaz de realizar un trabajo similar a los modelos estocásticos sin tener tantas restricciones y tiempos aceptables.

A partir de estas metodologías se han fusionado modelos que aterrizan en esquemas variados.

1.3.2. Funcionamiento de un RAV

Para que un RAV proporcione como salida la cadena de texto que se emitió por el locutor, como vimos anteriormente, se tiene que dar una serie de etapas, desde la digitalización de la señal hasta el proceso de decodificación o búsqueda de patrones. En este apartado se aborda el procedimiento a seguir.

1.3.2.1. Preparación de la base de datos o corpus de voces

Debido a la naturaleza del sistema, al ser de vocabulario pequeño, medio o extenso, se debe disponer del corpus de voces que permita tanto el entrenamiento como las pruebas. Dependiendo también de si es dependiente o no de locutor, la cantidad de locutores diversos que deberán estar presentes en el corpus acrecentará, incluso por cada locutor se debe dar una serie de repeticiones de las elocuciones, incluyendo diversos tipos de formas de pronunciación. El corpus de voces se puede crear de manera personalizada para el entorno en el que se desea tratar (viajes, oficinas, escuelas, bibliotecas, etc.) o se puede adquirir alguno de los ya existentes, como TIMIT, English SpeechDat (M), AURORA, EUROMI, Albayzin, SALA, por mencionar algunos, y que dependerá del idioma deseado [31, 34]. Una limitación de estos tipos de corpus existentes radica en el precio de adquisición que ronda arriba de cientos de dólares, además de que depende mucho de la región geográfica de interés y la presencia o no del tipo de idioma deseado. Desde

este momento se deben tener presentes las unidades mínimas para el reconocedor, ya que serán las que se utilizarán por el reconocedor y que serán las que estarán modeladas en el sistema. Para elegir la más apropiada se suelen tener en cuenta algunos factores, como la precisión (capacidad de representación de la elocución en diversos escenarios), la entrenabilidad (capacidad de estimar los parámetros con datos necesarios) y su capacidad de ser generalizable (las nuevas observaciones deben ser reconocidas con los sistemas ya entrenados). Las unidades básicas más comúnmente usadas son palabras y fonemas (sonido de una letra) o fonos (realización acústica de una letra en diferentes contextos) [37]. En los sistemas independientes de locutor y de tamaño considerable, se recomienda usar fonos o fonemas y las unidades derivadas de ellos como tri-fonos, alófonos o fonos dependientes del contexto (fonos que consideran el vecino de la izquierda y el vecino de la derecha). Habitualmente cuando se hace la grabación de las señales de voz se suelen utilizar parámetros con una tasa de muestreo de 4410 Hz, 8000 Hz, o 16000 Hz, además de usarse 16 bits por muestra, formato .wav y un solo canal (mono).

1.3.2.2. Extracción de características: *audio front-end*

Como se mencionó anteriormente en la Figura 1.6 de la página 15, la señal acústica puede ser representada y compactada por una serie de características que la definen y que permiten poder representarla. Estos atributos permiten distinguir entre diferentes tipos de objetos, los tipos o categorías en los cuales se pueden clasificar los objetos (para nuestro caso serán señales acústicas) son conocidos como clases [38]. Mediante el reconocimiento de patrones se puede realizar la extracción de características de una señal de voz y poder clasificarla, colocándola en la *clase* correspondiente. En el RAV se necesitan características que sean invariantes en el tiempo, a cambios en amplitud y que no sean sensibles a la duración de las palabras [35].

Las principales partes del proceso de modelado de la señal de voz (*audio front-end*) [39] comienzan con un contorno espectral (*spectral shaping*), el cual es el proceso de convertir la señal de voz analógica en una señal digital (conversión

A/D). Este también involucra algún tipo de filtrado, enfatizando componentes importantes de las frecuencias. El segundo elemento, *el análisis espectral*, implica analizar el espectro con el fin de capturar los aspectos sobresalientes de la señal. Para analizar el espectro, la señal muestreada (obtenida del paso anterior) es dividida en tramas (intervalos cortos de tiempo, con una longitud normal de 20-25 ms, traslapados 10 ms). La *transformación paramétrica* es el tercer elemento, el cual es el nombre para el ajuste de las mediciones anteriores (las características son integradas en un solo vector).

Para el caso del reconocimiento, la prioridad es extraer de la señal acústica suficiente información que permita el reconocimiento de fonemas, palabras o frases que contengan estos fonemas tanto como sea posible. Por tanto, el reconocimiento del habla extrae de la señal de entrada la información acústica de la palabra o frase pronunciada por el locutor. Esta operación normalmente nos permite obtener un conjunto de parámetros o coeficientes en menor cantidad que las muestras de entrada. Al mismo tiempo se conserva, en la mayoría de los casos, la representación correcta de las diferentes unidades que constituyen la voz [40]. El propósito principal del proceso de reconocimiento de voz continua es hacer concordar una señal de entrada $x[n]$ con un conjunto de palabras o frases de acuerdo a ciertos criterios adecuados. La primera etapa de este proceso habitualmente es conocido como la parametrización, y por ende su principal tarea es reducir los datos convirtiendo la señal de entrada en parámetros, mientras virtualmente se conservan todas las características de la información de la señal de voz.

Los principales algoritmos clásicos de análisis espectral incluyen características basadas en bancos de filtros, análisis de Fourier y basados en predictores lineales. Además, estos pueden ser complementados con coeficientes cepstrales [39]. Los coeficientes cepstrales en frecuencias de Mel (MFCC) son uno de los parámetros (características acústicas) de voz más útiles, desarrollados por Davis and Mermelstein [35, 36, 41, 42]. Por ejemplo, para los MFCC, la salida resultante para cada trama de observación es un vector de 39 coeficientes (un solo vector de observación o_t, ver Figura 1.7).

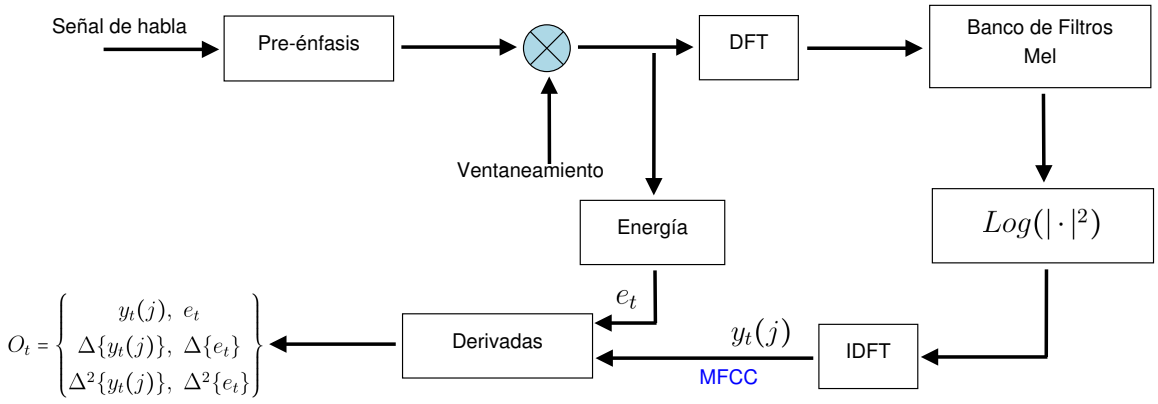

Figura 1.7: Proceso de construcción de los MFCC dentro del audio front-end en un RAV

El proceso de construcción de los MFCC sigue un conjunto de etapas:

1. *Digitalización y cuantificación de la señal.* Involucra convertir la señal analógica recibida del transductor en una señal digital, a través de muestreo (usando el teorema de muestreo) y cuantificación (representar cada muestra obtenida a través de un número entero, para ello se utiliza cierta cantidad de bits, siendo lo más habitual 16, contemplando valores entre -32768 y 32767) a través de la ecuación:

$$x[n] = x_a(n\Delta T), \tag{1.3.3}$$

donde $\Delta T = \frac{1}{F_s}$, y F_s es la frecuencia máxima de la señal o frecuencia de Nyquist.

2. *Pre-énfasis en altas frecuencias.* Con la señal de voz discretizada ($x[n]$) se aplica un filtro FIR de primer orden en las altas frecuencias, dado por:

$$y[n] = x[n] + \alpha x[n-1], \tag{1.3.4}$$

donde α es un valor configurable, habitualmente con 0.95 o 0.97.

3. *Ventaneo.* Dada la naturaleza del mecanismo de producción de voz, y por

ende del tracto vocal, la señal de voz es no estacionaria, variando constantemente en el tiempo en cuanto a sus características estadísticas. El ventaneo es un procesado que se hace a la señal resultante del paso anterior con el fin de extraer características espectrales sobre ventanas (habitualmente de 20 ms o 25 ms) para obtener tramos cuasi-estacionarios (los cuales se asume son constantes en el tiempo). Se considera que para el proceso de ventaneo (ver ec. (1.3.5)) se requieren factores como el ancho de la ventana en milisegundos, el desplazamiento (separación y solapamiento de cada ventana contigua), así como la forma de la ventana (comúnmente Hamming, ver ec. (1.3.6)).

$$x_s[n] = y[n]h_{h_s}[n], \tag{1.3.5}$$

donde h_{h_s} es una ventana de Hamming que tiene valores constantes para todas las tramas (frames) s, y está dada por:

$$h_{h_s}[n] = \begin{cases} (1-\alpha) - \alpha\cos(\frac{2\pi n}{\|\text{frame}\|}), & \text{si } 0 \le n < \|\text{frame}\|, \\ \\ 0, & \text{de otra manera}, \end{cases} \tag{1.3.6}$$

donde $\|frame\|$ es el tamaño de la ventana a analizar; si α toma el valor de 0.46 la ventana es de Hamming (si α es 0.5, la ventana se considera de Hanning).

4. *Análisis con transformada de Fourier.* Involucra aplicar la DFT (Discrete Fourier Transform) de cada ventana $x_s[n]$, usando para ello la transformada rápida de Fourier (FFT):

$$X_s[k] = \sum_{n=0}^{N-1} x_s[n]e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, 2, ..., N-1, \tag{1.3.7}$$

donde $X_s[k]$ es la DFT de cada trama producida por el ventaneo.

5. *Banco de filtros Mel.* Tomando en cuenta una transformada discreta de Fou-

rier $X_s[k]$ de una señal de entrada, se define un banco de filtros M, con $(m = 0, 1, 2, \ldots, M)$, donde el filtro m es un filtro triangular dado por la ecuación:

$$H_m[k] = \begin{cases} 0, & si\ k < f(m-1), \\ \frac{k-f(m-1)}{f(m)-f(m-1)}, & f(m-1) \le k \le f(m), \\ \frac{k-f(m-1)}{f(m+1)-f(m)}, & f(m) \le k \le f(m+1), \\ 0, & k > f(m+1). \end{cases} \tag{1.3.8}$$

Estos filtros calculan el promedio del espectro alrededor de cada frecuencia central. Se define f_h como la frecuencia más alta y f_l como la frecuencia más baja del banco de filtros en Hz, F_s es la frecuencia de muestreo en Hz, M el número de filtros y N el tamaño de la FFT. Los puntos límite $f(m)$ son uniformemente espaciados en la escala de Mel, dado por:

$$f(m) = \frac{N}{F_s}\beta^{-1}\Big(\beta(f_l) + m\frac{\beta(f_h) - \beta(f_l)}{M+1}\Big), \tag{1.3.9}$$

donde la escala de Mel está definida por:

$$\beta(f) = 1125\ \ln\Big(1 + \frac{f}{700}\Big), \tag{1.3.10}$$

y su inversa definida por:

$$\beta^{-1}[b] = 700\ \Big(e^{\frac{b}{1125}} - 1\Big). \tag{1.3.11}$$

De esta manera se puede calcular el logaritmo de la energía de cada filtro por la ecuación:

$$S(m) = \ln\Big(\sum_{k=0}^{N-1} |X_s[k]|^2 H_m[k]\Big), \quad 0 < m < M. \tag{1.3.12}$$

6. *Análisis cepstral.* El cepstrum en frecuencias de Mel es la transformada coseno discreta de las salidas de los M filtros (simplificación de la IDFT, dado que el espectro logarítmico es una función real simétrica), dado por:

$$c(m) = \sum_{m=0}^{M-1} S(m)\cos\Big(\pi n\Big(\frac{m-1/2}{M}\Big)\Big), \tag{1.3.13}$$

donde M varía para diferentes implementaciones de 24 a 40, para el RAV, generalmente se usan 12 o 13.

7. *Coeficientes cepstrales dinámicos.* A los coeficientes obtenidos en la sección anterior, se les agrega la información de la energía. Esto se realiza por cada trama ventaneada:

$$E = |x_s[n]|^2. \tag{1.3.14}$$

Dado que la señal de voz no es constante, los cambios en el tiempo son trascendentales. Una forma de capturar esta información es calcular los coeficientes delta, por tanto los vectores de observación finales se formarán por:

- 12 coeficientes MFCC c_k (resultado del análisis cepstral).
- 12 coeficientes de primer orden delta ($\Delta c_k = c_{k+2} - c_{k-2}$).
- 12 coeficientes de segundo orden delta ($\Delta\Delta c_k = \Delta c_{k+2} - \Delta c_{k-2}$).
- 1 coeficiente de energía E.
- 1 coeficiente delta de energía ΔE.
- 1 coeficiente delta-delta de energía $\Delta\Delta E$.

En total, si se sigue el proceso completo, resultan 39 coeficientes por trama de señal de voz.

1.3.2.3. Modelado acústico y clasificación de patrones

Como se vió en la Figura 1.6 de la página 15, el siguiente componente del proceso del RAV es el modelo acústico, que involucra dos componentes [34].

1. *Clasificación de patrones.* Tiene como objetivo fundamental realizar la agrupación de los vectores acústicos de características (observaciones) en clases, proporcionando una manera de medir el grado de pertenencia de una observación a una determinada categoría. Entre los principales mecanismos para realizar esta labor se encuentran la cuantificación vectorial, las redes neuronales y funciones de densidad de probabilidad (principalmente Gaussianas).

2. *Modelado acústico.* Tiene como objetivo realizar asociaciones temporales de vectores de observación agrupados con las respectivas unidades de reconocimiento, proporcionando una medida de acople de las observaciones de características dada la supuesta pertenencia a una determinada unidad de reconocimiento. Entre los mecanismos más comúnmente usados se encuentran los modelos ocultos de Markov, la alineación dinámica en el tiempo (DTW - dynamic time warping) y las redes neuronales recurrentes.

1.3.2.4. Modelo de lenguaje

Un modelo de lenguaje estadístico $p_\Gamma(W)$ (ver ec. (1.3.1)), con parámetros de n-grama Γ, representa la probabilidad a priori de una secuencia de palabras $W = \{w_1, w_2, \ldots, w_m\} \triangleq W_1^m$, y es calculada multiplicando las probabilidades de una palabra predicha w_i, condicionada sobre las $n-1$ palabras precedentes, w_{i-n+1}^{i-1}. Las probabilidades del n-grama pueden ser calculadas de acuerdo a la estimación de máxima verosimilitud [2, 3, 5]:

$$p_{ML}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i})}, \tag{1.3.15}$$

donde $c(\cdot)$ es el conteo de una secuencia de palabras, $p_\Gamma(W)$ es combinada con la probabilidad del modelado acústico $p_\Lambda(O|W)$, dados los parámetros del modelo de Markov Λ, con el fin de encontrar la secuencia de palabras más probable $\widehat{W}$ con referencia a la ec. (1.3.1). A pesar de que los modelos de lenguaje de n-gramas son efectivos en explotar las regularidades léxicas locales, sufren de insuficiencia en los datos de entrenamiento, información distante y generalización del modelo; lo

cual restringe las capacidades de predicción para reconocimiento de voz continua en vocabulario extenso (LVCSR). Sin embargo, existen métodos para mitigar estas circunstancias, tales como las técnicas de suavizado [5].

1.3.2.5. Diccionario de pronunciación.

Los diccionarios de pronunciación son usados para entrenar sistemas de procesamiento de voz transcribiendo la pronunciación de las palabras en unidades manejables como fonemas [43]. Por ejemplo, si se tiene la frase "Ella está enferma", su transcripción fonética pudiera ser /e/ /dZ/ /a/ /e/ /s/ /t/ /a/ /e/ /n/ /f/ /e/ /rr/ /m/ /a/, o si se tiene la frase "Llamar Juan Goncálvez Nuñez" su transcripción fonética sería /dZ/ /a/ /m/ /a/ /rr/ /x/ /u/ /a/ /n/ /g/ /o/ /n/ /k/ /a/ /l/ /b/ /e/ /s/ /n/ /u/ /jn/ /e/ /s/. De esta manera cada palabra en el diccionario contextual es traducida a su equivalente fonético, tomando en cuenta el idioma seleccionado y las consideraciones del alfabeto fonético internacional (IPA).

1.3.2.6. Decodificación o reconocimiento

Una vez modelado el sistema que se utilizará para la fase de identificación de lo que dice el locutor, se debe buscar toda posible cadena de palabras W para encontrar la que más se acopla a las secuencias de observaciones, tomando en cuenta la ec. (1.3.1) de la página 14. La idea siempre será encontrar el conjunto de palabras que mejor se ajuste a las restricciones fonéticas y de lingüística definidas en el modelo acústico. Esto se hace mediante el encuadre de las secuencias de observaciones de la entrada con la mejor secuencia de unidades de habla definidas en el modelo de red de conocimiento (modelado acústico). Esta fase también se aborda con más detalle en la secciones 2.3 y 3.3, donde se hace mención a los RAV tipo MMG-MOM y RNP-MOM. Para poder realizar el proceso de decodificación se requiere el uso del modelo de lenguaje, diccionario de pronunciación y en caso que se requiera un modelo léxico [32].

1.3.2.7. Rendimiento en los sistemas de RAV

Este es comúnmente especificado en términos de precisión y velocidad [34]. La precisión es usualmente medida o ponderada con la tasa de error por palabra (% WER), mientras que la velocidad es medida con un factor real de tiempo. En primer lugar, se realiza la alineación de la secuencia de palabras reconocidas con la secuencia de palabras de referencia (*dynamic string alignment*). La WER puede ser calculada como sigue:

$$\text{WER} = \frac{S + D + I}{N}, \tag{1.3.16}$$

donde S, D, I y N son el número de substituciones, eliminaciones, inserciones y el número total de palabras a analizar, respectivamente.

1.4. Planteamiento del problema

Si bien las líneas de trabajo del reconocimiento de voz han cambiado con la inclusión de las redes neuronales en el esquema del modelado acústico, estas aún no han resuelto todos los problemas relacionados con la disminución radical de las tasas de error en el reconocimiento. Debido a esto, nuevos esquemas de arquitectura en las redes neuronales [44], nuevos mecanismos en la utilización de la función objetivo [45–47], nuevos algoritmos de entrenamiento de la red [48] o incluso transformaciones y adaptaciones de las observaciones acústicas de entrada de la red [49, 50] han sido algunas de las líneas de trabajo en las que se han enfocado varios proyectos para lograr mejores resultados en la eficiencia en el área de reconocimiento de voz.

Por ende, surge la necesidad de plantearse nuevas líneas de seguimiento a los trabajos en la disciplina del reconocimiento automático de voz o del habla. En esta investigación por tanto surge la pregunta de *¿qué propuesta metodológica en el modelo RNP-MOM puede ser planteada con el fin de lograr resultados aceptables en las tasas de error de reconocimiento de voz?* La respuesta a esta pregunta sugiere un análisis de las metodologías actuales y las nuevas tendencias, para

que se tenga una línea base de partida con el fin de poder alcanzar los objetivos. En este sentido, la integración de modelos conexionistas (RNP) y modelos estocásticos (MOM) da pauta a su consideración para el planteamiento de nuevos esquemas de reconocimiento.

1.5. Justificación

Trabajos e investigaciones relacionadas al área del reconocimiento automático de voz han sido llevados a cabo por años [2–5, 7–15, 17], sin embargo, aún no se han logrado alcanzar las tasas de error significativamente pequeñas, considerando que el humano puede reconocer la voz en circunstancias muy adversas, cosa que los sistemas actuales del área no han llegado siquiera a ver posible. Por añadidura, la búsqueda de nuevos enfoques técnicos en el área de reconocimiento de voz se ve como una incesante línea a seguir. El nuevo enfoque de uso de redes neuronales y modelos ocultos de Markov es una de las líneas más novedosas en esta disciplina, y que ha arrojado resultados bastante prometedores [6, 9, 11, 51–54], y por consiguiente se requiere dar continuidad a las técnicas y procedimientos que puedan mejorarlo. La idea base siempre será lograr una cercanía con el reconocimiento de voz a través del humano. Los nuevos sistemas automáticos que implican la utilización de la voz en su funcionamiento cada vez requieren nuevas formas de ir tratando los contextos y escenarios que se presentan a la hora de buscar tasas de error pequeñas.

La aplicación final de la manipulación de voz en general es quizá la consideración más importante que orilla a no dejar de lado esta rama, resaltando aplicaciones como [25]:

- Transmisión y almacenamiento digital (reducción o ampliación de ancho de banda).
- Síntesis de voz (producción sintética en robots, teléfonos).
- Identificación y verificación de locutor (validación de personas).

- Reconocimiento de voz (traducción de mensajes en texto, usando dispositivos electrónicos y móviles).
- Aplicaciones de ayuda a personas (ciegos, mudos).
- Mejoramiento de la calidad de señal (filtrado, restauración).

1.6. Objetivo

Proponer una variante metodológica para la construcción de sistemas de reconocimiento de voz basados en modelos estocásticos y esquemas conexionistas.

1.6.1. Objetivos Particulares

1. Realizar un comparativo entre la metodología clásica de MMG-MOM y la vertiente conexionista de RNP-MOM, con el fin de que sirva como punto de partida para la propuesta de investigación.
2. Llevar a cabo una propuesta inicial que busque mejorar a dos de las metodologías para reconocimiento de voz que fueron analizadas.
3. Establecer una comparativa entre la metodología que propondremos y las metodologías de referencia.

1.7. Hipótesis

Existe la posibilidad de plantear una nueva variante metodológica, basada en modelos estocásticos y esquemas conexionistas, para llevar a cabo el modelado de señales de voz y aplicarlo a tareas de reconocimiento de voz con una tasa de error pequeña de al menos $0.5\,\%$ comparable con lo que se maneja en los métodos actuales del estado del arte.

1.8. Organización de la tesis

El resto del documento de tesis queda integrado por los siguientes capítulos. En el capítulo 2 se describen los aspectos esenciales del reconocimiento de voz desde el punto de vista de los modelos ocultos de Markov y las mezclas Gaussianas (MMG-MOM). Este enfoque convencional es explicado en sus fases de entrenamiento y de decodificación. La idea de los MOM es dar un trato versátil a la variabilidad temporal de la señal de voz, aunado a la búsqueda de pertenencia de una secuencia de observaciones de entrada a un estado ligado de tri-fono del MOM en particular a través de las mezclas Gaussianas (integración de clasificación de patrones y modelado acústico). Dejando abierta la inquietud de cómo poder mejorar su desempeño a través de modelos más recientes.

En el capítulo 3 se mencionan los procesamientos no lineales de la señal de voz desde la perspectiva de los modelos híbridos que integran los modelos conexionistas o redes neuronales y los modelos estocásticos o modelos ocultos de Markov (RNP-MOM). Se hace énfasis en cómo las redes neuronales profundas y el aprendizaje profundo pueden substituir el cálculo de probabilidad de emisión de observaciones en los modelos de Markov. Logrando de esta forma una mejora en las tasas de reconocimiento.

En el capítulo 4 se presentan las aportaciones metodológicas del presente trabajo con el fin de lograr dar un paso adelante en la intención de buscar alternativas que ayuden a mejorar las tasas de reconocimiento de voz. El aporte se enfoca en la descripción de funciones de costo alternativas (dentro del entrenamiento de una red neuronal) a la convencional entropía cruzada, resultando en un nuevo enfoque que es denominado función de costo de entropía cruzada no uniforme mapeada. La cual puede dar una visión particular o específica a las tramas que son tratadas con el fin de localizar su pertenencia a un estado ligado de tri-fono de MOM y poder obtener en consecuencia las probabilidades de emisión de observaciones, que serán el substituto de las mezclas Gaussianas en el enfoque MMG-MOM.

En el capítulo 5 se describen los principales lineamientos de métodos y mate-

riales para la realización de los experimentos, indicando el origen y tratamiento de los datos de entrada a los modelos, así como la gramática y el léxico usado para darles un procesamiento en el modelado. Al final se mencionan los principales hallazgos y su significancia desde el punto de vista de las tasas de reconocimiento.

En el capítulo 6 se obtienen las notas de discusión y conclusiones del trabajo con el fin de recapitular los hechos trabajados y las posibles ventajas y desventajas de lo que se presenta en la investigación, dejando al final posibles líneas futuras de seguimiento a los mecanismos utilizados.

Capítulo 2

Modelado convencional de la señal de voz

Se describe el mecanismo clásico del reconocimiento del habla utilizando los modelos ocultos de Markov y las mezclas Gaussianas. Para ello se mencionan los elementos principales de estos dos esquemas de modelado.

En la sección 2.1 se estudian los modelos ocultos de Markov, su funcionamiento, entrenamiento y aplicación dentro del campo del reconocimiento de voz. En la sección 2.2 se describen los principales lineamientos en el funcionamiento de las mezclas Gaussianas dentro del reconocimiento de voz, así como su rol dentro de él. En la sección 2.3 se mencionan las bases para la fusión de estos dos enfoques (MMG-MOM) y cómo su integración permite el funcionamiento del RAV.

2.1. Modelos ocultos de Markov (MOM)

Un modelo oculto de Markov es un modelo estocástico que es utilizado para modelar fenómenos aleatorios variantes en el tiempo. El modelo de Markov es utilizado para modelar la evolución de fenómenos aleatorios denotados por estados discretos como una función de tiempo, donde la transición de un estado a otro es aleatoria. Supongamos que un sistema puede estar en uno de S estados diferentes, y que en cada instante del tiempo discreto este se puede mover a otro estado de manera aleatoria, con una probabilidad de transición en el tiempo t

dependiente solamente sobre el estado del sistema en el tiempo t (ver Figura 2.1). En la Figura 2.1 se muestran tres estados, desde el estado 1, cualquier transición es posible, se observan cuáles son las probabilidades de pasar de un estado a otro. Tenemos que q_t denota un estado en el tiempo t, y donde q_t puede tomar uno de los valores $Q = \{q_1, q_2, \ldots, q_S\}$; el estado inicial se selecciona de acuerdo a una probabilidad π_i, esto es $\pi_i = P(q_1 = i)$, donde $i = 1, 2, \ldots, S$ [25, 55].

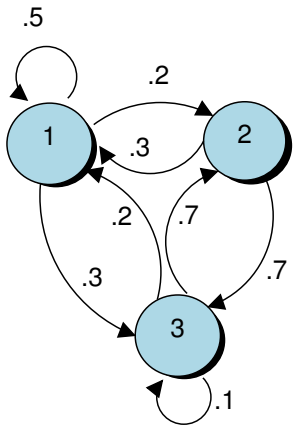

Figura 2.1: Diagrama de estados probabilísticos: modelo de Markov simple

De acuerdo a la descripción anterior, la probabilidad de transición depende solo del estado actual: $P(q_{t+1} = j|q_t = i, q_{t-1} = k, q_{t-2} = l, \ldots) = P(q_{t+1} = j|q_t = i)$. Esta estructura de probabilidades es llamada *propiedad de Markov*, y la secuencia aleatoria de estados $\{q_0, q_1, q_2, \ldots\}$, es llamada *secuencia de Markov o cadena de Markov*. Esta secuencia es la salida del modelo de Markov. Podemos determinar la probabilidad de llegar al siguiente estado sumando todas las probabilidades de los caminos para llegar ahí: $P(q_{t+1} = j) = P(q_{t+1} = j|q_t = 1)P(q_t = 1) + P(q_{t+1} = j|q_t = 2)P(q_2 = 2) + \cdots + P(q_{t+1} = j|q_t = S)P(q_t = S)$. Este cálculo puede ser hecho convenientemente en una matriz, por ejemplo

$$p_t = \begin{bmatrix} P(q_t = 1) & P(q_t = 2) & \cdots & P(q_t = S) \end{bmatrix}.$$

Así tenemos el vector de probabilidades de cada estado, y se tiene la matriz A conteniendo las probabilidades de transición

$$A = \begin{bmatrix} P(1|1) & P(2|1) & \cdots & P(q|1) \\ P(1|2) & P(2|2) & \cdots & P(q|2) \\ \vdots & & & \\ P(1|q) & P(2|q) & \cdots & P(q|q) \end{bmatrix},$$

donde $P(j|i)$ es una abreviación de $P(q_{t+1} = j|q_t = i) \triangleq a_{ij}$. Por ejemplo para la Figura 2.1, la matriz de probabilidades de transición sería:

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0 & 0.7 \\ 0.2 & 0.7 & 0.1 \end{bmatrix}.$$

Una asignación de probabilidad de estado estacionario es aquella que no cambia de un instante de tiempo al siguiente, de esta manera la probabilidad debe satisfacer la ecuación $p = pA$. Como una ley de probabilidad total, cada renglón de la matriz A debe sumar 1.

La idea detrás de un MOM puede ser ilustrada utilizando los problemas de urnas de probabilidad elemental (ver Figura 2.2). Supongamos que tenemos S diferentes urnas y donde cada una contiene sus propios conjuntos de bolas de colores. En cada instante de tiempo, una urna es seleccionada aleatoriamente de acuerdo al estado en el que esta se encontraba en el instante de tiempo previo (es decir, de acuerdo a un modelo de Markov). De esta manera, una bola se extrae aleatoriamente de una urna seleccionada en el tiempo t. La bola es lo que observamos como salida, y el estado actual es oculto.

En los MOM no se observa directamente el estado, en lugar de eso, cada estado tiene una distribución de probabilidad asociada. Cuando un MOM se mueve del estado q_t en el tiempo t, la salida observada o_t es un resultado de la variable aleatoria O_t que es seleccionada de acuerdo a la distribución $b(o_t|q_t = q)$, que se representa usando la notación $b(o_t|q_t = q) = B_q(o_t)$, o simplemente $b_q(o_t)$ (ver Figura 2.3).

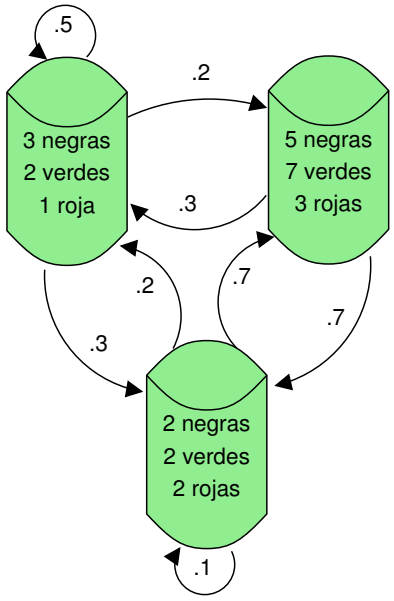

Figura 2.2: Ejemplo de concepto de modelos ocultos de Markov

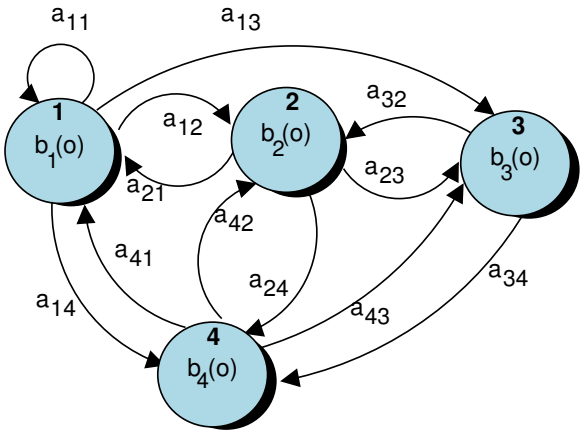

Figura 2.3: Modelo oculto de Markov de cuatro estados

En el ejemplo previo de las urnas, las probabilidades de salida dependían del contenido de las urnas. Una secuencia de salidas de un MOM es $\{o_0, o_1, o_2, \ldots\}$. La información inherente del estado no se observa directamente, está oculta. La distribución de probabilidad de cada estado puede ser de un tipo, y en general, cada estado puede tener su propio tipo de distribución. Sin embargo, en la prác-

tica suelen tener el mismo tipo de distribución pero con parámetros diferentes. Tenemos que M denota el número de posibles resultados de todos los estados y O_t es la variable aleatoria de salida en el tiempo t, con resultado o_t. Determinamos la probabilidad de cada posible salida sumando todas las probabilidades, $P(O_t = i) = P(O_t = i|q_t = 1)P(q_t = 1) + P(O_t = i|q_t = 2)P(q_t = 2) + \cdots + P(O_t = i|q_t = S)P(q_t = S)$. De tal manera que

$$r_t = \begin{bmatrix} P(O_t = 1) & P(O_t = 2) & \cdots & P(O_t = M) \end{bmatrix}$$

y

$$B = \begin{bmatrix} P(O_t = 1|q_t = 1) & \cdots & P(O_t = M|q_t = 1) \\ P(O_t = 1|q_t = 2) & \cdots & P(O_t = M|q_t = 2) \\ \vdots & & \\ P(O_t = 1|q_t = S) & \cdots & P(O_t = M|q_t = S) \end{bmatrix};$$

de tal manera que $b_{ji} = P(O_t = i|q_t = j) = b_j(o_t = i)$. Para el ejemplo de las urnas, con bolas de colores negro, verde y rojo, tenemos una correspondencia con los valores 1, 2 y 3 respectivamente.

$$B = \begin{bmatrix} 1/2 & 1/3 & 1/6 \\ 1/3 & 7/15 & 1/5 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}.$$

Donde cada renglón debe sumar uno. Entonces, las probabilidades de salida pueden ser calculadas por $r_t = p_t B$. Los parámetros de un MOM son descritos por el conjunto $\Lambda = \{A, \pi, B\}$, muy parecido a nuestros modelos de espacio de estados [55].

Por ende, un MOM se compone de un par de procesos estocásticos: una cadena *oculta* de Markov y un proceso *observable*, el cual es una función probabilística. Esto significa que los eventos observables en el mundo real (como las

observaciones acústicas) son modelados con distribuciones de probabilidad (posiblemente continuas), y que son la parte observable del modelo, asociados con un proceso Markoviano de primer orden de estados individuales de un tiempo discreto. En general este último no es ergódico. La semántica del modelo (correspondencia conceptual con un fenómeno físico) es usualmente encapsulada en la parte oculta; por ejemplo, en un RAV un MOM puede ser usado para modelar una palabra, donde cada estado de la parte oculta representa un fonema (o una unidad subfonética), mientras que la parte observable cuenta como las características de los eventos acústicos correspondientes en un espacio característico dado (como por ejemplo una señal acústica muestreada, representada de manera adecuada) [56]. De esta manera un MOM se define de la siguiente forma [57]:

1. Un conjunto Q de S estados, $Q = \{q_1, ..., q_S\}$, que son los distintos valores que el proceso estocástico oculto discreto puede tomar.

2. Una probabilidad de distribución del *estado inicial*, por ejemplo $\pi = \{P(q_i|t = 0), q_i \in Q\}$, donde t es un índice de tiempo discreto.

3. Una distribución de probabilidad que caracteriza las transiciones permitidas entre los diferentes estados, es decir $a_{ij} = \{P(q_j$ en el tiempo $t|q_i$ en el tiempo $t-1)$,$q_i \in Q, q_j \in Q\}$, donde las *probabilidades de transición* $a_{ij} \hat{=} A$ se asume que son independientes en el tiempo t.

4. Una *observación o espacio característico* O, el cual es un universo continuo o discreto para todos los posibles eventos observables (usualmente un subconjunto de R^d, donde d es la dimensionalidad de las observaciones).

5. Un conjunto de distribuciones de probabilidad (vistas como probabilidades de traslado o de salida), que describen las propiedades estadísticas de las observaciones para cada estado del modelo: $B = \{b_j(o_t) = P(o_t|q_j), q_j \in Q, o_t \in O\}$. Tradicionalmente asociados a modelos de mezclas Gaussianas.

Los MOM representan un paradigma de aprendizaje, en el sentido que los ejemplos del caso que va a ser modelado pueden ser obtenidos y utilizados en

conjunto con un algoritmo de entrenamiento con el fin de aprender las estimaciones adecuadas de $\Lambda = \{A, \pi, B\}$. Los algoritmos más populares para estos casos son el algortimo de forward-backward (o Baum-Welch) y el algoritmo de Viterbi [57, 58].

Los MOM pueden ser aplicados a reconocimiento de patrones, donde los patrones ocurren como eventos que se dan secuencialmente en el tiempo. La más exitosa de sus aplicaciones radica en el procesamiento de voz. Donde cada palabra o sonido (fonema) a ser reconocido está representado por un MOM, y donde la salida es un vector con ciertas características que se derivan de los datos de la voz. La variabilidad aleatoria en el vector de características y la cantidad de tiempo que cada característica produce es modelada por el MOM. La variabilidad en la duración de cada palabra es modelada por un modelo de Markov. La variabilidad de las salidas es modelada por una selección aleatoria dentro de cada estado. Por ejemplo, en un sistema con un vocabulario pequeño de N palabras hay N MOM (A_i, π_i, B_i), siendo cada uno entrenado para representar los parámetros de esa palabra. Esta es la fase de entrenamiento del problema de reconocimiento de patrones [55]. Para lograr el reconocimiento de una palabra desconocida, esta secuencia de vectores característicos es calculada, y la probabilidad de que esta secuencia de vectores sea producida por el MOM (A_i, π_i, B_i) es determinada para cada i. El MOM que produce la mayor probabilidad se selecciona como la palabra reconocida [25].

Mientras que los MOM han sido una metodología dominante para el modelado acústico en reconocimiento automático de voz durante décadas, muchas de sus debilidades también han sido bien conocidas y se han convertido en el foco de muchas investigaciones. Una de sus principales debilidades es la imposibilidad de representar la dependencia temporal en las características acústicas de la voz, que sin embargo, es una propiedad esencial de la dinámica de la voz [59].

2.1.1. Caracterización de los tres problemas fundamentales de los modelos ocultos de Markov

La idea de los modelos ocultos de Markov debe ser caracterizada por la resolución de tres problemas base [60].

2.1.1.1. Problema 1: Cálculo de la verosimilitud o probabilidad

Dado un modelo oculto de Markov $\Lambda = \{A, \pi, B\}$ y una secuencia de observaciones $O = \{o_1, \ldots, o_T\}$, determinar la verosimilitud $P(O|\Lambda)$. Para un MOM con S estados y una secuencia de T observaciones, existen S^T posibles secuencias ocultas. Para tareas reales, donde S y T son grandes, S^T es un número muy grande, y por tanto no podemos obtener la probabilidad de observación total calculando una probabilidad de observación por separado para cada secuencia de estados oculta y entonces acumularla. En lugar de ese procedimiento exponencial, se utiliza un algoritmo con eficiencia de $O(S^2T)$ llamado *Forward*. Este algoritmo es un tipo de procedimiento de *programación dinámica*, es decir, es un algoritmo que utiliza una tabla para almacenar valores intermedios mientras construye la probabilidad acumulando todas las probabilidades de todas las rutas posibles de estados ocultos que pudieran generar la secuencias de observaciones, pero lo hace de una manera eficiente retomando implícitamente cada una de estas rutas en una sola rejilla o cuadrícula.

Cada celda en la rejilla del algoritmo de *forward* $\alpha_t(j)$ representa la probabilidad de estar en el estado j después de haber visto las primeras t observaciones y dado el autómata Λ. El valor de cada celda $\alpha_t(j)$ es calculado acumulando las probabilidades de cada ruta que pudiera conducirnos a esa celda:

$$\alpha_t(j) = P(o_1, o_2, \ldots, o_t, q_t = j|\Lambda), \tag{2.1.1}$$

donde $q_t = j$ significa "la probabilidad de que el t-ésimo estado en la secuencia de estados sea el estado j". Por tanto, calculamos la probabilidad acumulando las extensiones de todas las rutas que conducen a la celda actual. Para un estado

dado q_j en el tiempo t el valor de $\alpha_t(j)$ está dado por

$$\alpha_t(j) = \sum_{i=1}^{S} \alpha_{t-1}(i) a_{ij} b_j(o_t), \tag{2.1.2}$$

donde $\alpha_{t-1}(i)$ es la probabilidad previa de la ruta de la celda del algoritmo *forward*, a_{ij} es la probabilidad de transición del estado previo q_i al estado actual q_j, y $b_j(o_t)$ es la probabilidad de observación de estado del símbolo de observación o_t dado el estado actual j. El algoritmo completo es desglosado como sigue:

function FORWARD (*observations* of len T, *state-graph* of len S) **returns** *forward-prob*

1 crear una matriz de probabilidad $forward[N+2, T]$
2 **for** each state s **from** 1 **to** S **do** ;paso de inicialización
3 $forward[s, 1] \leftarrow a_{0,s} * b_s(o_1)$
4 **for** each time t **from** 2 **to** T **do** ;paso de iteración o recursión
5 **for** each state s **from** 1 **to** S **do**
7 $forward[s, t] \leftarrow \sum_{s'=1}^{S} forward[s', t-1] * a_{s',s} * b_s(o_t)$
8 $forward[q_F, T] \leftarrow \sum_{s=1}^{S} forward[s, T] * a_{s,q_F}$;paso de terminación
9 **return** $forward[q_F, T]$

donde los estados 0 y F son el estado inicial y final del modelo oculto de Markov y no emiten observaciones.

2.1.1.2. Problema 2: Decodificación

Dada una secuencia de observaciones $O = \{o_1, \ldots, o_T\}$ y un modelo oculto de Markov $\Lambda = \{A, \pi, B\}$, descubrir la mejor secuencia de estados ocultos $Q = \{q_1, q_2, q_3, \ldots, q_T\}$. Para cualquier modelo, tal como un MOM, y que contiene variables ocultas, la tarea de determinar cuál secuencia de variables es el origen subyacente de alguna secuencia de observaciones es llamada tarea de **decodificación**. El algoritmo de decodificación más común para los MOM es el algoritmo de Viterbi. Al igual que el algoritmo de *forward*, el algoritmo de Viterbi es un tipo de programación dinámica y hace uso de una rejilla para tal efecto. La idea es procesar la secuencia de observaciones de izquierda a derecha rellenando las cel-

das de la rejilla. Cada celda de la rejilla de Viterbi $v_t(j)$ representa la probabilidad de que el MOM esté en el estado j después de haber visto las primeras t observaciones y pasado a través de la secuencia de estados más probable $\{q_0, q_1, \ldots, q_{t-1}\}$, dado el autómata Λ.

El valor de cada celda $v_t(j)$ es calculado recursivamente tomando la ruta más probable que pudiera conducir a esa celda. Formalmente cada celda se expresa como sigue

$$v_t(j) = \max_{q_0, q_1, \ldots, q_{t-1}} P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\Lambda). \tag{2.1.3}$$

Nótese que se está representando la ruta más probable tomando el máximo sobre todas las secuencias de estado previas posibles $\max_{q_0, q_1, \ldots, q_{t-1}}$. Dado que ya hemos calculado la probabilidad de estar en cada estado en el tiempo $t-1$, podemos calcular la probabilidad de Viterbi tomando la más probable de las extensiones de las rutas que conducen a la celda actual. Para un estado dado q_j en el tiempo t, el valor de $v_t(j)$ es calculado como sigue

$$v_t(j) = \max_{i=1}^{S} v_{t-1}(i) a_{ij} b_j(o_t), \tag{2.1.4}$$

donde $v_{t-1}(i)$ es la probabilidad previa de la ruta de la celda de la rejilla de Viterbi, a_{ij} es la probabilidad de transición del estado previo q_i al estado actual q_j, y $b_j(o_t)$ es la probabilidad de observación de estado del símbolo de observación o_t dado el estado actual j. El algoritmo completo es desglosado como sigue:

function VITERBI (*observations* of len T, *state-graph* of len S) **returns** *best-path*

1 crear una matriz de probabilidad de ruta $viterbi[N+2, T]$
2 **for** each state s **from** 1 **to** S **do** ;paso de inicialización
3 $viterbi[s, 1] \leftarrow a_{0,s} * b_s(o_1)$
4 $backpointer[s, 1] \leftarrow 0$
5 **for** each time t **from** 2 **to** T **do** ;paso de iteración o recursión
6 **for** each state s **from** 1 **to** S **do**
7 $viterbi[s, t] \leftarrow \max_{s'=1}^{S} viterbi[s', t-1] * a_{s',s} * b_s(o_t)$
8 $backpointer[s, t] \leftarrow \operatorname{argmax}_{s'=1}^{S} viterbi[s', t-1] * a_{s',s}$

9 $viterbi[q_F, T] \leftarrow \max_{s=1}^{S} viterbi[s, T] * a_{s,q_F}$;paso de terminación

10 $backpointer[q_F, T] \leftarrow \operatorname{argmax}_{s=1}^{S} viterbi[s, T] * a_{s,q_F}$

11 **return** the backtrace path by following pointers to states back in time from $backpointer[q_F, T]$

donde los estados 0 y F son el estado inicial y final del modelo oculto de Markov y no emiten observaciones.

2.1.1.3. Problema 3: Aprendizaje

Dada una secuencia de observaciones $O = \{o_1, \ldots, o_T\}$ y el conjunto de estados en el MOM, aprender los parámetros $\Lambda = \{A, \pi, B\}$. La entrada a tal algoritmo de entrenamiento sería una secuencia no etiquetada de observaciones O y un vocabulario de estados ocultos potenciales Q. El algoritmo estándar para el entrenamiento del MOM es el algoritmo de *forward-backward o Baum-Welch*, un caso especial del algoritmo de *Expectation-Maximization (E-M)*. El algoritmo nos permitirá entrenar las probabilidades de transición A y las probabilidades de emisión B. El algoritmo *forward-backward* inicia con una estimación inicial de los parámetros $\Lambda = \{A, B\}$. Entonces se ejecutan iterativamente dos pasos. Como otros casos del algoritmo de E-M, el algoritmo de *forward-backward* incluye dos etapas: el paso de la "esperanza" y el paso de la "maximización". En el paso de la "esperanza" se calcula la cuenta esperada de ocupación de estado γ y la cuenta esperada de transición de estado ξ a partir de las probabilidades A y B anteriores. En el paso de "maximización" se utiliza γ y ξ para recalcular las nuevas probabilidades A y B. El procedimiento general de *forward-backward* se muestra a continuación:

function FORWARD-BACKWARD(*observations* of len T, *output vocabulary V*, *hidden state set Q*) **returns** *HMM=(A,B)*

1 **initialize** A and B

2 **iterate** until convergence

3 **E-step**

$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\Lambda)} \ \forall\ t$ and j

$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(S)} \ \forall\ t,\ i,$ and j

M-step

$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{S}\xi_t(i,j)}$

$\hat{b}_j(\mathrm{v}_k) = \frac{\sum_{t=1,\ s.t.\ o_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$

return A,B

La probabilidad de la observación dado el modelo es la probabilidad *forward* de toda la sentencia completa (o la probabilidad *backward* de toda la sentencia), es decir, $P(O|\Lambda) = \alpha_T(S) = \beta_T(1) = \sum_{j=1}^{S}\alpha_t(j)\beta_t(j)$. ξ_t es la probabilidad de estar en el estado i en el tiempo t y en el estado j en el tiempo $t+1$, dada la secuencia de observaciones y el modelo. $\hat{a}$ es calculada dividiendo el número esperado de transiciones del estado i al estado j (el cual es acumulado sobre todo T de ξ) entre el número esperado de transiciones del estado i (acumulando todas las transiciones salientes del estado i). La fórmula para recalcular la probabilidad de observación: la probabilidad de un símbolo dado v_k a partir de un vocabulario de observación V dado un estado j, es decir $\hat{b}_j(\mathrm{v}_k)$. Entonces esta probabilidad de emisión se calcula dividiendo el número esperado de veces en el estado j y observar el símbolo v_k entre el número esperado de veces en el estado j. Para ello usamos la probabilidad de estar en el estado j en el tiempo t: $\gamma_t(j)$. Para el numerador de $b_j(\mathrm{v}_k)$ sumamos $\gamma_t(j)$ para todo tiempo t en el cual la observación o_t es el símbolo v_k, que es en el que estamos interesados. Para el denominador, sumamos $\gamma_t(j)$ sobre todo tiempo t. El resultado es el porcentaje de veces que estuvimos en el estado j y observamos el símbolo v_k; donde $\sum_{t=1,\ s.t.\ o_t=v_k}$ implica acumular sobre todo t para el cual la observación en el tiempo t fue v_k. Además, la frecuencia esperada de empezar en el estado j en el tiempo $t = 1$ equivale a $\hat{\pi} = \gamma_1(j)$.

Para entender el algoritmo se necesita definir la probabilidad de *backward* β, la cual es la probabilidad de ver las observaciones del tiempo $t+1$ hasta el final, dado que estamos en el estado i en el tiempo t:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T | q_t = i, \lambda), \tag{2.1.5}$$

y se calcula de manera similar al algoritmo de *forward*:

1. Inicialización:

$$\beta_T(i) = a_{iF}, \qquad 1 \leq i \leq S, \tag{2.1.6}$$

2. Recursión (los estados q_0 y q_F no emiten observaciones):

$$\beta_t(i) = \sum_{j=1}^{S} a_{i,j} b_j(o_{o+1}) \beta_{t+1}(j), \qquad 1 \leq i \leq S,\ 1 \leq t < T, \tag{2.1.7}$$

3. Terminación:

$$P(O|\Lambda) = \alpha_T(q_F) = \beta_1(0) = \sum_{j=1}^{S} a_{0j} b_j(o_1) \beta_1(j). \tag{2.1.8}$$

2.2. Modelos de mezclas Gaussianas

Un modelo de mezclas Gaussianas (MMG) es una función de densidad de probabilidad paramétrica representada por una suma ponderada de componentes de densidad Gausianos. Los MMGs son comúnmente utilizados como modelos paramétricos de distribuciones de probabilidad de características o mediciones continuas en un sistema biométrico, tal como características espectrales relacionadas al tracto vocal en los sistemas de reconocimiento del habla. Los parámetros del MMG son estimados a partir de datos de entrenamiento usando el algoritmo iterativo E-M (expectation-maximization) o una estimación MAP (maximum a posteriori) a partir de un modelo previo bien entrenado [61–64].

Un MMG es una suma de M componentes de densidad Gaussianos dada por la ecuación:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i), \tag{2.2.1}$$

donde x es un vector de datos continuo de dimensión D (por ejemplo, de características o medidas), w_i son los pesos de las mezclas, y $g(x|\mu_i, \Sigma_i)$ son los componentes de densidades Gaussianos. La Figura 2.4 muestra esta densidad de probabilidad compuesta por varias funciones Gaussianas con pesos w_i, donde cada función Gaussiana tiene su propia media μ_i y matriz de covarianza Σ_i. De esta manera, la densidad de probabilidad de una variable x, $p(x|\lambda)$, se calcula con base a los parámetros de la función compuesta.

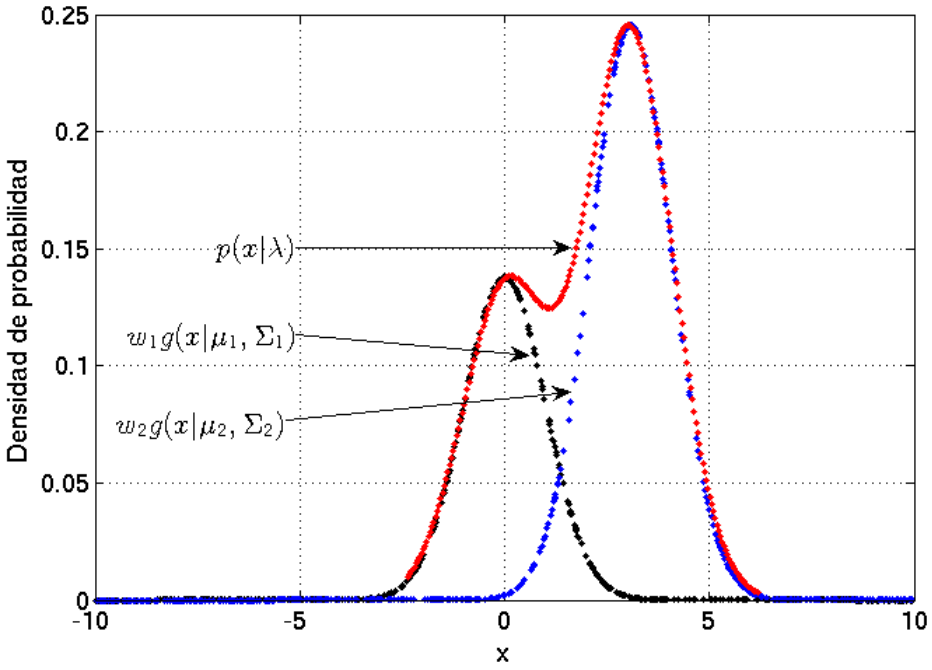

Figura 2.4: Ejemplo gráfico de un modelo de mezclas Gaussianas

Cada componente de densidad es una función Gaussiana de D variables de la forma

$$g(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)}, \tag{2.2.2}$$

con un vector de media μ y una matriz de covarianza Σ. Los pesos de las mezclas satisfacen la restricción de $\sum_{i=1}^{M} w_i = 1$. El MMG completo es parametrizado por

los vectores de medias, las matrices de covarianza, y los pesos de las mezclas de todos los componentes de densidad. Estos parámetros son representados de manera conjunta por la notación de la ecuación:

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, \qquad i = 1, \ldots, M. \tag{2.2.3}$$

Existen diversas variantes del MMG de la ec. (2.2.3). Las matrices de covarianzas pueden ser condicionadas o restringidas a ser diagonales. Además los parámetros pueden ser compartidos o ligados entre los diversos componentes Gaussianos, de tal forma que tendrían matrices de covarianza comunes para todos los componentes. La elección de la configuración del modelo (número de componentes, matrices de covarianza completas o diagonales, matrices comunes, y la ligadura de los parámetros) es normalmente determinada por la cantidad de datos disponibles para la estimación de los parámetros del MMG y cómo este es utilizado en una aplicación biométrica específica. Es importante mencionar que dado que los componentes Gaussianos están actuando de manera conjunta para modelar de manera completa las densidades de las características, las matrices de covarianza completas no son necesarias incluso si los vectores característicos no son independientes entre sí. La combinación lineal de Gaussianas básicas de covarianza diagonal es capaz de modelar las correlaciones entre elementos de vectores de características. El efecto de utilizar un conjunto de M matrices Gaussianas de covarianza completa puede ser equiparable al obtener un conjunto de Gaussianas de covarianza diagonal.

Así, dado un estado de MOM con su vector de media u_j y matriz de covarianza $\sum_j$, y un vector de observación o_t, la estimación de probabilidad Gaussiana multivariable es (la equivalente a la ec. (2.2.2)):

$$b_j(o_t) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(o_t-\mu_j)'\Sigma_j^{-1}(o_t-\mu_j)}. \tag{2.2.4}$$

La ec. (2.2.4) puede ser simplificada a la versión en la ec. (2.2.5), en la cual en lugar de usar una matriz de covarianza, solo se mantiene una media y una varianza para cada dimensión. De esta forma se estima $b_j(o_t)$ de un vector de

características de dimensión D dado un estado j-ésimo del MOM, usando solamente una Gaussiana multivariable de covarianza diagonal:

$$b_j(o_t) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{jd}^2}} e^{-\frac{1}{2}\frac{(o_{td}-\mu_{jd})^2}{\sigma_{jd}^2}}. \tag{2.2.5}$$

Entrenar una Gaussiana multivariable de covarianza diagonal es una generalización de entrenar una Gaussiana univariable.

Se ha demostrado que se pueden usar modelos de Gaussianas multivariable para asociar una probabilidad a una observación acústica de vector de características. Este modela cada dimensión del vector de características como una distribución normal. Pero una observación cepstral particular puede tener una distribución que no sea nada normal. Por esta razón se suele modelar la probabilidad de observación (b_j) no como una sola Gaussiana multivariable, sino más bien como una mezcla ponderada de Gaussianas multivariable (como se vio en la ec. (2.2.1)), y que se le denomina modelo de mezclas Gaussianas. La ec. (2.2.6) muestra la ecuación final para un MMG

$$p(x|\mu,\Sigma) = \sum_{i=1}^{M} w_i \frac{1}{\sqrt{2\pi|\Sigma_i|}} e^{(x-\mu_i)'\Sigma^{-1}(x-\mu_i)}. \tag{2.2.6}$$

Y en consecuencia para el cálculo de la probabilidad de emisión de observación, la ecuación resultante queda como:

$$b_j(o_t) = \sum_{m=1}^{M} w_{jm} \frac{1}{\sqrt{2\pi|\Sigma_{jm}|}} e^{(o_t-\mu_{jm})'\Sigma_{jm}^{-1}(o_t-\mu_{jm})}. \tag{2.2.7}$$

Es común usar las Gaussianas en sistemas biométricos, como reconocimiento de voz, debido a su capacidad de representar una clase amplia de distribuciones de muestras. Uno de los atributos poderosos de los MMG es su habilidad para formar aproximaciones suaves a densidades formadas de manera arbitraria. El modelo Gaussiano clásico uni-modal representa distribuciones de características a través de una posición (vector de medias) y una forma elíptica (matríz de covarianzas) y un cuantificador vectorial (VQ) o un modelo del vecino más cer-

cano que representa una distribución usando un conjunto discreto de funciones Gaussianas, cada una con su propia media y matriz de covarianza, para permitir una mejor capacidad de modelado.

El uso de un MMG para representar distribuciones de características en un sistema biométrico puede ser motivado también por la noción intuitiva que los componentes de densidades individuales pueden modelar algunos conjuntos subyacentes de clases ocultas. La forma espectral de la i-ésima clase acústica puede en dado caso ser representada por la media μ_i del i-ésimo componente de densidad, y las variaciones de la forma espectral promedio pueden ser representadas por la matriz de covarianzas Σ_i. Dado que todas las características usadas para entrenar el MMG son no etiquetadas, las clases acústicas están ocultas en el sentido que una clase de una observación es desconocida. Un MMG también puede ser visto como un MOM de un solo estado con una densidad de observación de una mezcla Gaussiana, o como una observación Gaussiana ergódica de un MOM fijo e igual a las probabilidades de transición. Asumiendo vectores de características independientes, la densidad de observación de los vectores de características obtenida de estas clases acústicas ocultas es una mezcla Gaussiana [61].

Estimación de los parámetros

Dados los vectores de entrenamiento y una configuración de MMG, se desean estimar los parámetros del modelo λ, que en algún sentido concuerdan mejor a la distribución de los vectores de características de entrenamiento. Existen varias técnicas disponibles para la estimación de los parámetros de un MMG [62].

Para entrenar la función de probabilidad del MMG se puede usar el algoritmo de Baum-Welch para obtener la probabilidad de que una cierta mezcla represente la observación, y de una forma iterativa actualizar la probabilidad. Se utiliza para ello la función ξ anterior para calcular la probabilidad de estado. Se define pues $\xi_{tm}(j)$ para suponer la probabilidad de estar en el estado j en el tiempo t con el m-ésimo componente de mezcla representando la probabilidad de salida o_t. Entonces se calcula $\xi_{jm}(j)$ de la siguiente forma [60]:

$$\xi_{tm}(j) = \frac{\sum_{i=1} S\alpha_{t-1}(j)a_{ij}w_{jm}b_{jm}(o_t)\beta_t(j)}{\alpha_T(F)}. \tag{2.2.8}$$

Si se tienen los valores de ξ de una iteración previa de Baum-Welch, entonces se puede emplear $\xi_{tm}(j)$ para recalcular la media, el peso de la mezcla y la covarianza utilizando las ecuaciones siguientes:

$$\hat{\mu}_{km} = \frac{\sum_{t=1}^{T} \xi_{tm}(k)o_t}{\sum_{t=1}^{T}\sum_{m=1}^{M} \xi_{tm}(k)}, \tag{2.2.9}$$

$$\hat{c}_{km} = \frac{\sum_{t=1}^{T} \xi_{tm}(k)}{\sum_{t=1}^{T}\sum_{m=1}^{M} \xi_{tm}(k)}, \tag{2.2.10}$$

$$\hat{\Sigma}_{km} = \frac{\sum_{t=1}^{T} \xi_{tm}(k)(o_t - \mu_{km})(o_t - \mu_{km})'}{\sum_{t=1}^{T}\sum_{m=1}^{M} \xi_{tm}(k)}. \tag{2.2.11}$$

2.3. Sistemas de reconocimiento de voz mediante el uso de MMG-MOM

En este apartado se mencionan los elementos requeridos para llevar a cabo el procedimiento de reconocimiento de voz utilizando enfoques estocásticos mediante los modelos ocultos de Markov y las mezclas Gaussianas.

2.3.1. Aplicando los MOM al reconocimiento de voz

Para el reconocimiento de voz, los estados ocultos de un MOM son fonos, partes de fonos, o palabras, y cada observación acústica (es generalmente obtenida cada 10 milisegundos, de tal forma que una señal de un segundo tiene 100 tramas) es información acerca del espectro y la energía de la señal de voz en algún punto del tiempo, y en donde el proceso de decodificación mapea esta secuencia de información acústica en fonos y palabras [60]. Los estados ocultos de un MOM pueden ser usados para modelar la voz en un número de formas diferentes. Para tareas de reconocimiento pequeñas, como reconocimiento de dígitos numéricos

o de respuestas tipo SI-NO, se pudiera construir un MOM cuyos estados correspondan a palabras completas. Sin embargo, para tareas más complejas se suelen utilizar los estados del modelo para representar unidades fonéticas (fonos), y las palabras serían secuencias de estas unidades fonéticas.

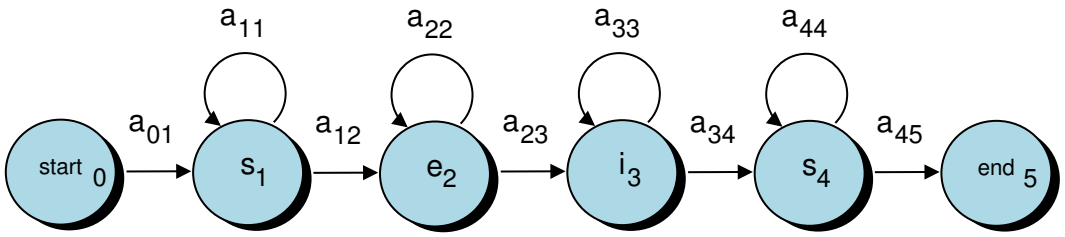

Figura 2.5: MOM para la palabra "seis", consistiendo de 4 estados de emisión y dos de no emisión

En la Figura 2.5 se puede observar un ejemplo de un MOM con fines de reconocimiento de voz, en el cual solo se permiten transiciones hacia el mismo estado o hacia delante, este tipo de modelos se denomina de **izquierda a derecha o red de Bakis**. En este tipo de esquemas se permiten las transiciones, así mismo para el caso en que se permitan las repeticiones de una entrada acústica en una cantidad variable de veces, ya que los fonos pueden tener una duración que varíe mucho, dependiendo del locutor, de la velocidad del habla, del contexto fonético o el nivel de prosodia de la palabra en cuestión. Por ejemplo, existen escenarios donde el fono `/a/` varia en longitudes de 7 a 387 milisegundos (1 a 40 tramas), mientras que fonos como la `/s/` puede variar de 7 milisegundos a más de 1.3 segundos (130 tramas) en algunas palabras.

Sin embargo, para escenarios donde la complejidad del reconocimiento se vuelve mayor, los MOMs como los de la Figura 2.5 se vuelven insuficientes; esto debido principalmente a que los fonos pueden durar más de un 1 segundo, es decir 100 tramas, y muchas de estas tramas no son acústicamente idénticas. Entonces las características espectrales de un fono, y la cantidad de energía varían drásticamente a lo largo del fono.

En la Figura 2.6 se pueden observar los cambios espectrales a lo largo del tiempo de los 4 fonemas de la palabra "seis". En la figura de la parte superior se

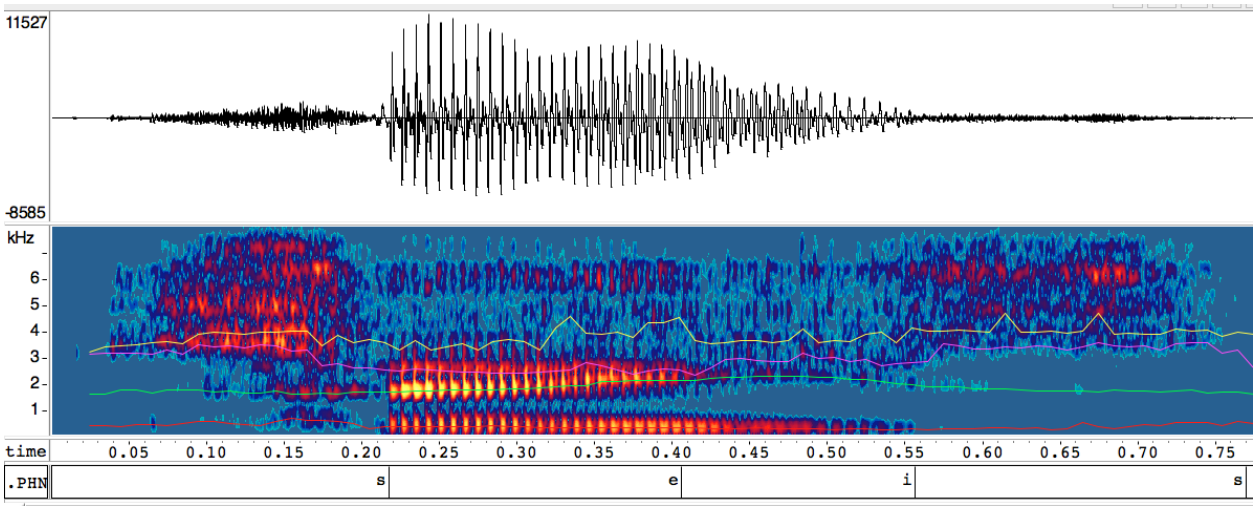

Figura 2.6: Espectrograma y forma de onda de los 4 fonos de la palabra seis. Nótese los cambios continuos en cada uno de los 4 fonemas con respecto al tiempo

muestra como varia la presión de aire de la señal de voz con respecto al tiempo; ahí se puede observar como cambia la amplitud o intensidad de la presión de aire para cada fonema. En la figura de la parte inferior, el eje x denota el tiempo en segundos, en el eje y se muestran los componentes de frecuencias de la señal de voz (los 4 formantes de frecuencias iniciales de la señal son mostrados por las líneas horizontales de colores). La amplitud de la señal (representando la energía) se puede notar en la intesidad de color rojo-naranja que aparece en los distintos componentes de fecuencias. Además, se puede apreciar en la Figura 2.6 el cambio de una fonema con respecto a su contexto, por ejemplo el fonema `/s/` varía cuando se encuentra al principio o al final de la palabra `/seis/`, dado que antes y después de él se localiza una fonema diferente. Adicionalmente, también el espectrograma señala cómo se da un cambio gradual de cada fonema cuando es seguido de otro. En general, el espectrograma nos muestra cómo se distribuye la energía a lo largo del tiempo y en sus componentes frecuenciales, enfatizando la intensidad de dichos componentes.

Para capturar el hecho de la naturaleza no homogénea de los fonos a lo largo del tiempo, en los escenarios de reconocimiento de voz continua con un vocabu-

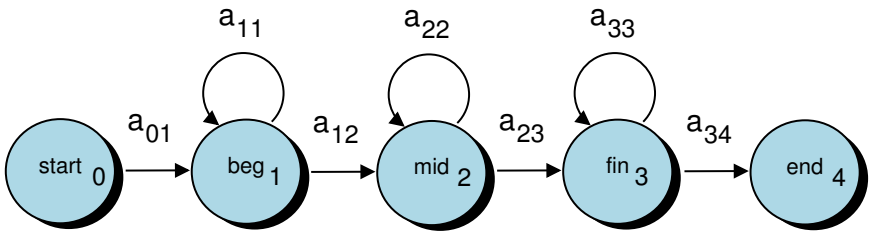

Figura 2.7: MOM estándar para un fono, el cual consiste en tres estados emisores de observaciones y dos no emisores

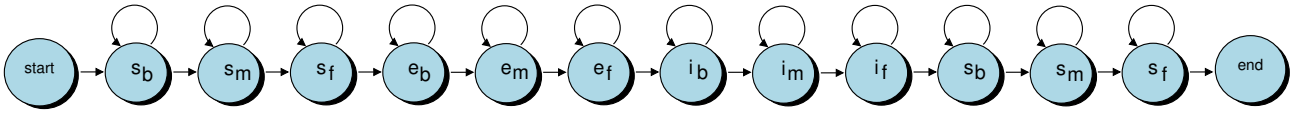

Figura 2.8: MOM compuesto para la palabra "seis". Se forma concatenando los modelos de fono individuales

lario extenso, generalmente se modela un fono con más de un estado en el MOM. La configuración más común es la que utiliza tres estados para cada fono; por tanto cada fono consiste de tres estados de emisión de observaciones además de dos estados no emisores de observaciones, tal como se muestra en la Figura 2.7. Se suele reservar el término de **modelo de fono** para referirse a un MOM completo de 5 estados, mientras que el término de **estado del MOM** para referirse a cada uno de los estados individuales de subfono del MOM. En este sentido, si se desea construir una palabra completa usando modelos de fonos, solo se substituye cada fono de la palabra con su respectivo modelo de fono de tres estados, tal como se muestra la Figura 2.8. Como se puede observar, se substituyen los estados inicial y final no emisores de observaciones para cada uno de los modelos de fono con transiciones directamente hacia el estado de fono emisor precedente y siguiente, dejando solo dos estados no emisores totales.

2.3.1.1. Creación de fonemas dependientes del contexto

Como se ha mencionado en el apartado anterior, en la práctica los efectos contextuales en un fonema causan grandes variaciones en la forma en que los

diferentes sonidos son producidos. Por tanto, para lograr una buena discriminación fonética, diferentes MOMs deben ser entrenados para cada contexto. La forma más simple y común es usar un tri-fono, donde cada fono tiene un MOM diferente para cada par único de vecinos en la izquierda y derecha. Por ejemplo, suponga la notación `x-y+z` para representar el fono `/y/` que ocurre después de un `/x/` y antes de un `/z/`. La frase "La nena" sería representada por la secuencia de fonos `/sil/ /l/ /a/ /n/ /e/ /n/ /a/ /sil/`, y si el MOM de tri-fono fuera usado, la secuencia que debería ser modelada constaría de

```
sil sil-l+a l-a+n a-n+e n-e+n e-n+a n-a+sil sil
```

Note que los límites de las palabras abarcan los contextos del tri-fono y que las dos instancias del fono `/n/` son representadas por diferentes MOMs debido a que sus contextos son diferentes. Este uso de los así llamados **tri-fonos de palabra cruzada** (cross-word tri-phones) proporciona la mejor precisión de modelado, pero conduce a complicaciones en el decodificador. Sistemas más simples resultan del uso de **tri-fonos de palabra interna** (word-internal tri-phones), donde el ejemplo de arriba quedaría como sigue

```
sil l+a l-a n+e n-e+n e-n+a n-a sil
```

Nótese que este tipo de modelado no cruza los límites entre las palabras al momento de hacer los tri-fonos dependientes del contexto. Aquí se requieren menos modelos diferentes, simplificando el problema de la estimación de los parámetros y el diseño del decodificador. Sin embargo, el costo es una incapacidad para modelar los efectos contextuales en los límites de las palabras y en procesos de voz fluida estos son considerables. El uso de distribuciones de emisión de observaciones que usan mezclas Gaussianas permite que cada distribución de estado sea modelada de una forma precisa. Sin embargo, cuando se usan tri-fonos resulta un sistema con muchos parámetros a entrenar. Por ejemplo, en un sistema de vocabulario extenso con tri-fonos de palabras cruzadas necesitará normalmente $60,000$ tri-fonos. En la práctica, alrededor de 10 componentes de mezclas dan buen rendimiento. Asumiendo que las covarianzas que se usan son todas diagonales, el reconocedor con 39 elementos en los vectores de observaciones acústicas requeriría alrededor de 790 parámetros por estado. En consecuencia, en total se

necesitarían 142 millones de parámetros.

El problema de muchos parámetros y poca cantidad de datos de entrenamiento es crucial en el diseño de un reconocedor de voz estadístico. Los sistemas primitivos han lidiado con el problema ligando todas las componentes Gaussianas juntas para formar una pileta que era entonces compartida entre todos los estados del MOM. En estos sistemas llamados sistemas de mezclas ligadas, solo los pesos del componente de la mezcla eran específicos del estado y podían ser suavizados por medio de interpolación con modelos independientes del contexto. Comparativas entre MOMs discretos, de densidad continua y de mezclas ligadas han mostrado que los de mezclas ligadas son mejores. Sin embargo, esto se daba debido a la falta de buenas técnicas de suavizado para sistemas de densidad contínua. Más recientemente, el suavizado basado en ligadura de parámetros se ha convertido en algo popular. En específico, los estados ligados y la ligadura de componentes basados en fonos han sido estudiados con el fin de averiguar su comportamiento [65]. Usando estas técnicas de ligadura con MOM de densidad continua se proporcionan buenos resultados en el rendimiento de la precisión del modelado. La idea en este enfoque es ligar estados que son acústicamente indistinguibles. Esto permite que todos los datos asociados con cada estado individual sean agrupados y por lo tanto proporcionan estimaciones más robustas para los parámetros del estado ligado. Esto se puede ver en la Figura 2.9, en la parte superior cada tri-fono tiene su propia distribución de salida privada. Después de la ligadura varios estados comparten distribuciones.

La elección común de cuáles estados ligar es realizada usando árboles de decisión fonéticos [2, 60, 66, 67]. Esto involucra construir un árbol binario para cada posición de fono y estado. Cada árbol tiene una pregunta fonética de SI/NO tal como “¿Es el contexto izquierdo un sonido nasal?” en cada nodo. Inicialmente todos los estados para una posición de estado de fono dado son colocados en el nodo raíz del árbol. Dependiendo de cada respuesta, la pileta de estados es sucesivamente dividida y esto continúa hasta que los estados han sido distribuidos hacia los nodos hoja. Todos los estados en el mismo nodo hoja son entonces ligados. Por ejemplo, en la Figura 2.10 se muestra el caso de la ligadura de los

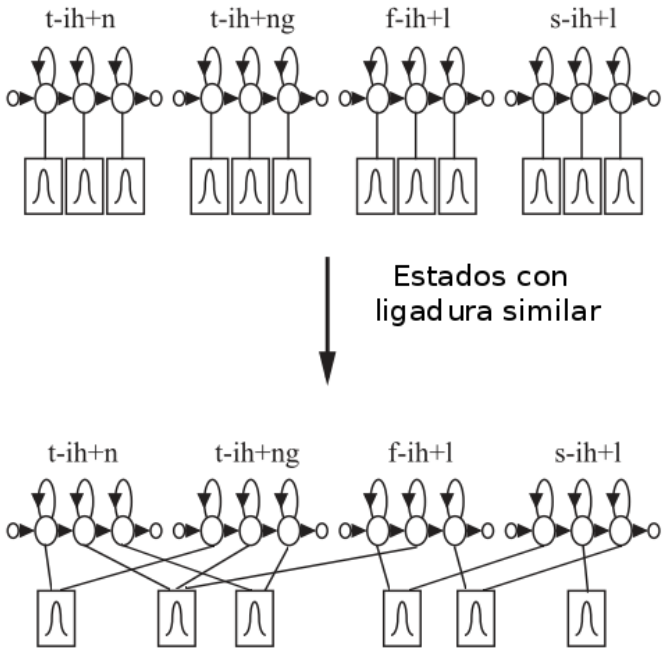

Figura 2.9: Ligadura de estados de acuerdo a variantes alofónicas (ejemplo tomado de sonidos en inglés)

estados centrales de todos los tri-fonos del fono inglés `/aw/` (como en la pronunciación de la palabra `out`). Aquí se observa que se distribuyen en el árbol dependiendo de las respuestas en las preguntas, esto concluye en alguno de los nodos terminales sombreados. En la Figura 2.10 se muestra que el estado central de `s-aw-n` se unirá al segundo nodo hoja de derecha a izquierda, ya que su contexto derecho es una consonante central y su contexto derecho es nasal, pero el contexto izquierdo no es un plosivo (central-stop).

Las preguntas en cada hoja son elegidas para maximizar la probabilidad de los datos de entrenamiento dado el conjunto final de ligaduras de estado. En la práctica, los árboles de decisión fonéticos proporcionan clusters compactos de estados de buena calidad, los cuales tienen suficientes datos asociados para estimar de una forma robusta las funciones de probabilidad de salida de las mezclas Gaussianas. Además, pueden ser usados para sintetizar un MOM para cualquier contexto posible si este aparece en los datos de entrenamiento o no,

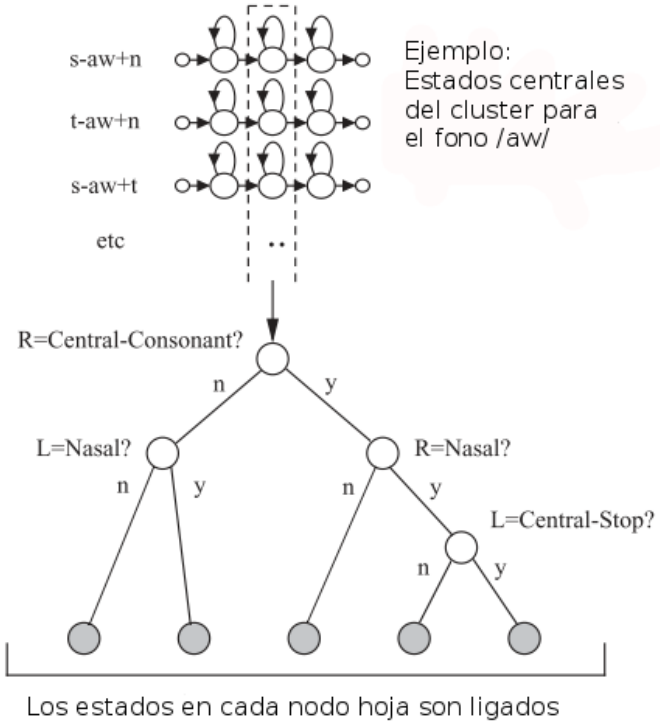

Figura 2.10: Agrupación de árbol de decisión para el ejemplo del fono inglés `/aw/`

simplemente descendiendo en los árboles y usando las distribuciones de estado asociadas con los nodos de hoja terminales. Finalmente, los árboles de decisión fonéticos pueden ser usados para incluir más que contextos de tri-fono simples.

2.3.1.2. Reconocimiento de voz usando el léxico y modelo del lenguaje

Es labor del decodificador segmentar simultáneamente las elocuciones en palabras e identificar cada una de ellas. Esta tarea es difícil dadas las variaciones en términos de cómo son pronunciadas las palabras de acuerdo a los fonos y también en cómo los fonos son articulados en características acústicas [60]. La verdadera tarea de la decodificación, en la cual se tienen que identificar los fonemas al mismo tiempo que se identifican y segmentan las palabras, es por supuesto más difícil. Por ejemplo, en una tarea de reconocimiento de dígitos (ver Figura 2.11) se puede notar que se utiliza el léxico para formar MOM de 5 esta-

dos para cada fonema o fono de cada palabra en el diccionario. A partir de ahí se forma el modelo resultante completo, donde se van uniendo los diferentes MOM de fonemas para formar palabras de reconocimiento y donde cada una al final puede tener o no un estado de emisiones de silencio; además se pueden repetir los dígitos tantas veces se defina en el modelo en cuestión, dependiendo de la tarea de reconocimiento.

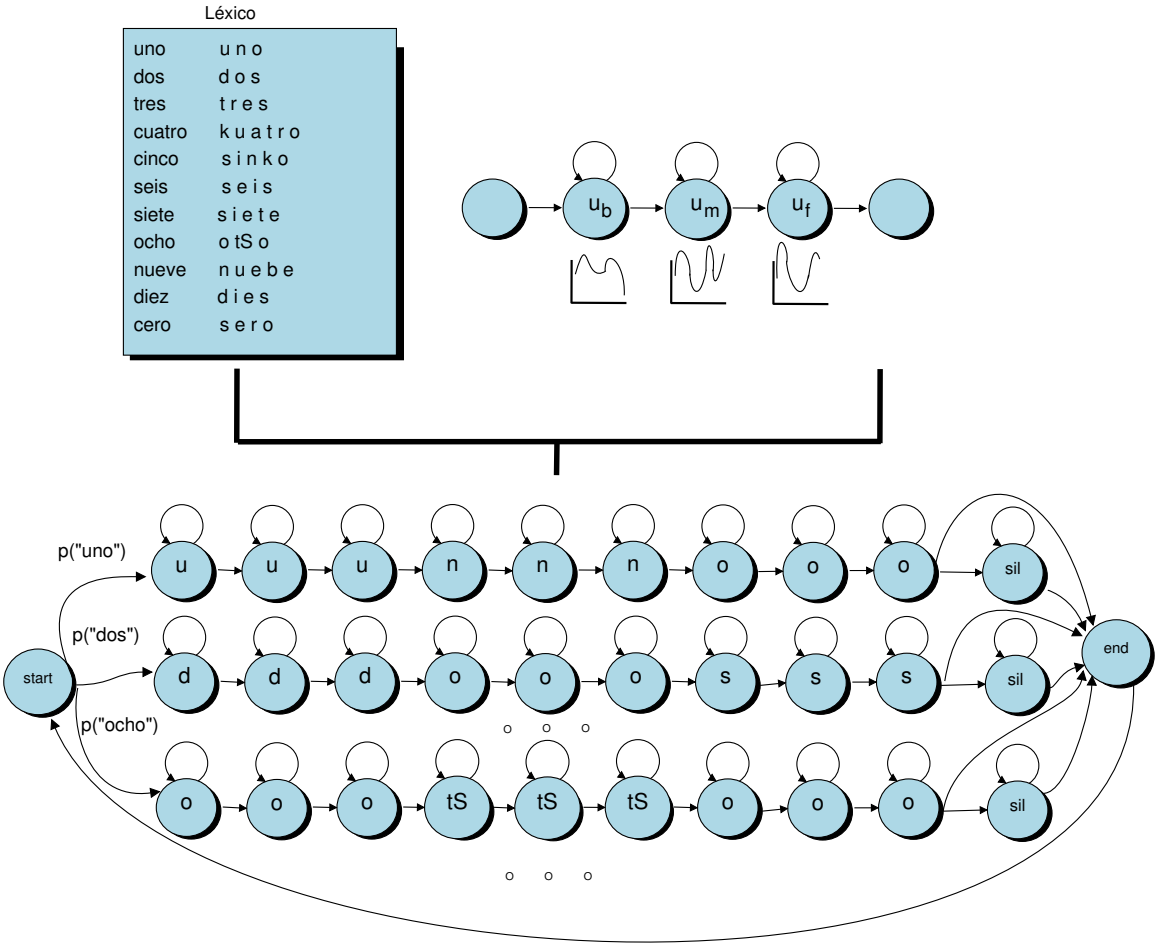

Figura 2.11: MOM para la tarea de reconocimiento de dígitos. El léxico especifica la secuencia de fonos, y cada MOM de fono está compuesto por tres subfonos, cada uno de los cuales con un modelo probabilístico de emisión de observaciones de tipo Gaussiano

Es común tener ciertas consideraciones respecto al modelo de lenguaje, por ejemplo, para tareas de reconocimiento de dígitos o de nombres de personas no

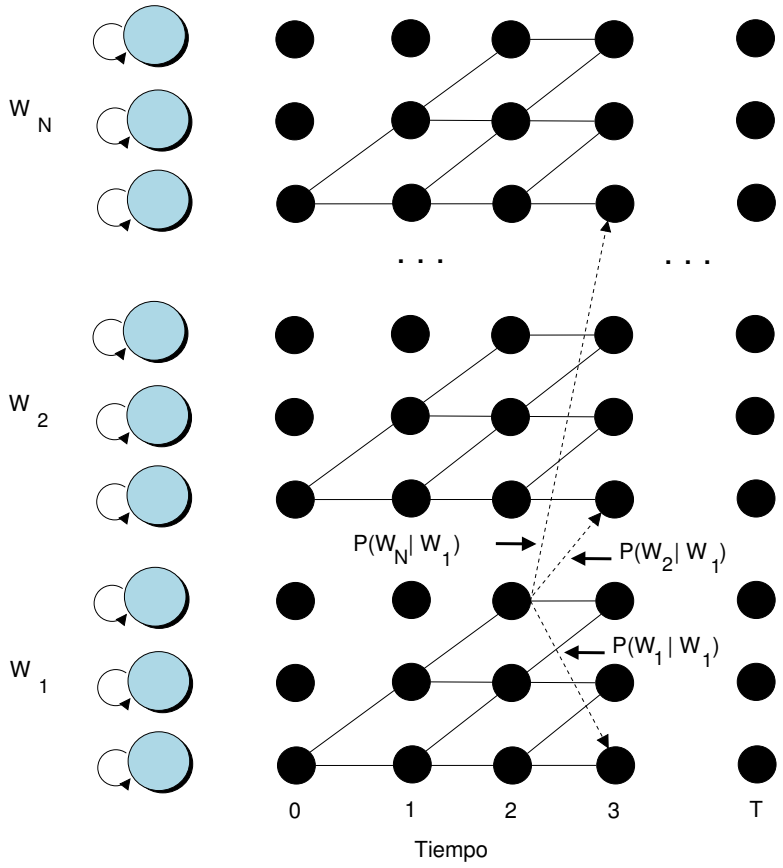

Figura 2.12: Rejilla de Viterbi de un MOM para un modelo de lenguaje de bi-grama

existen probabilidades de palabras, ya que en muchos casos (como reconocimiento de dígitos telefónicos y números de tarjetas de crédito) cada dígito tiene la misma probabilidad de aparición. En estos casos quizás la única limitación en el modelo es la gramática que se defina para la secuencia de palabras a reconocer, comúnmente definida por un esquema de gramática libre de contexto (GLC), en la cual se especifica cierta cantidad de repeticiones de dígitos. Pero para el caso de reconocimiento de nombres de personas, la limitante gramatical sería que debe ir primero un nombre y luego dos apellidos, por ejemplo. Si se quisiera considerar el reconocimiento de palabras continuas en un entorno libre y de vocabulario extenso, lo más normal es utilizar probabilidades de secuencias de palabras a través de n-gramas, siendo el bi-grama el más común.

En este sentido, en la Figura 2.12 se muestra una rejilla de Viterbi de un MOM para un modelo de bi-grama, en donde se muestra la habilidad del algoritmo para decodificar cadenas de palabras. Por ejemplo, se pudiera mejorar la matriz A para que incluya la probabilidad de transición entre palabras, a partir del fin de una palabra y hacia el inicio de otra. La probabilidad de transición en estos arcos, en lugar de venir de la matriz A dentro de cada palabra, viene del modelo de lenguaje $P(W)$. Una vez que la rejilla completa de Viterbi ha sido calculada para la elocución, se puede iniciar del estado más probable en el paso del tiempo final y proceder con punteros de retroceso para obtener la cadena más probable de estados y por consiguiente la secuencia más probable de palabras. En la Figura 2.13 se muestran los punteros del backtrace, los cuales son seguidos hacia atrás por la mejor ruta de estados, lo cual sucede en la w_2, eventualmente a través de w_N y w_1, resultando en la cadena final de palabras $w_1 w_n \cdots w_2$.

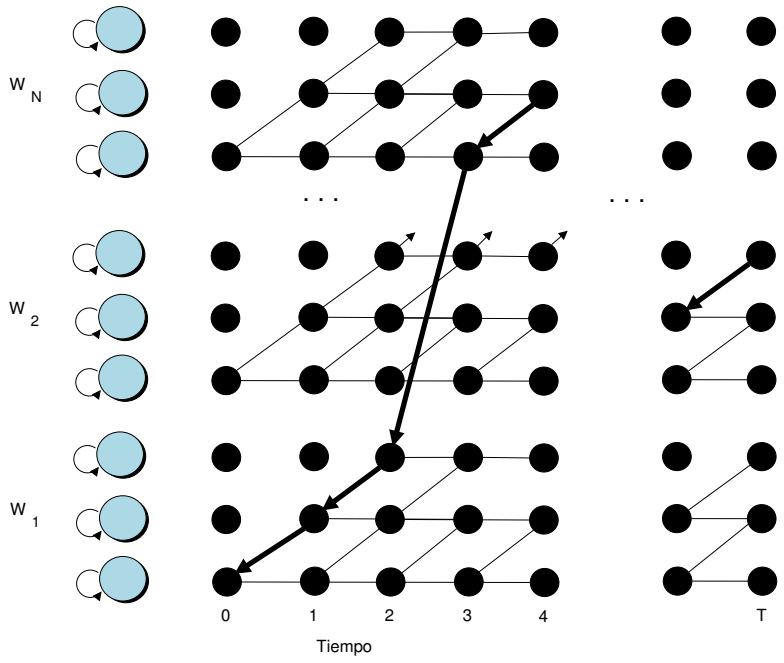

Figura 2.13: Viterbi backtrace en la rejilla del MOM. Se comienza en el estado final y el el resultado es la mejor cadena de fonos para la cual una cadena de palabras se deriva

El algoritmo de Viterbi es mucho más eficiente que el corrimiento exponencial

del algoritmo de *forward* para cada posible cadena de palabras. Sin embargo, aún es lento, pero investigaciones recientes en reconocimiento de voz se han enfocado en acelerar el proceso de decodificación. Por ejemplo, en entornos de reconocimiento de vocabulario extenso no se consideran todas las palabras posibles cuando el algoritmo está derivando rutas de una columna de estado a la siguiente. En su lugar, las rutas de probabilidades bajas son descartadas en cada paso temporal y no son extendidas a la siguiente columna de estado. Este procedimiento se implementa comúnmente a través de **beam search** [60], en la cual en cada tiempo t primero se calcula la probabilidad de la mejor (más probable) ruta/estado D. Entonces, se descarta cualquier estado que es peor que D por algún umbral fijo (beam width) θ. Se puede hablar a cerca de beam-search tanto en el dominio de probabilidad como en el de probabilidad logarítmica negativa. En el dominio de probabilidad cualquier estado/ruta cuyo valor sea menor que $\theta * D$ es descartado; en el dominio logarítmico negativo cualquier ruta cuyo costo sea mayor que $\theta + D$ es omitida. La beam search se implementa guardando para cada paso temporal una lista activa de estados. Solo transiciones de estas palabras son extendidas cuando se mueve al siguiente paso temporal. Debido a que en la práctica la mayoría de las implementaciones de Viterbi usan beam search, algunos en la literatura utilizan el término de **beam search o beam search de tiempo síncrono** en lugar de Viterbi.

Por tanto, en el proceso de decodificación se requiere un grafo de palabras, un diccionario de pronunciación o léxico y el conjunto de MOM dependientes del contexto. En el grafo de palabras (un MOM complejo) cada nodo es una red de MOM simples conectados. De esta forma se tiene un grafo de tres niveles (ver Figura 2.14).

Para una secuencia de observaciones $O = \{o_1, o_2, \ldots, o_T\}$ que está compuesta por T tramas, cada camino que comienza en el nodo inicial del grafo y concluye en el nodo terminal, y que por consecuencia pasaría por T estados (posiblemente con lazos incluidos) del MOM, es en tanto una posible hipótesis de reconocimiento.

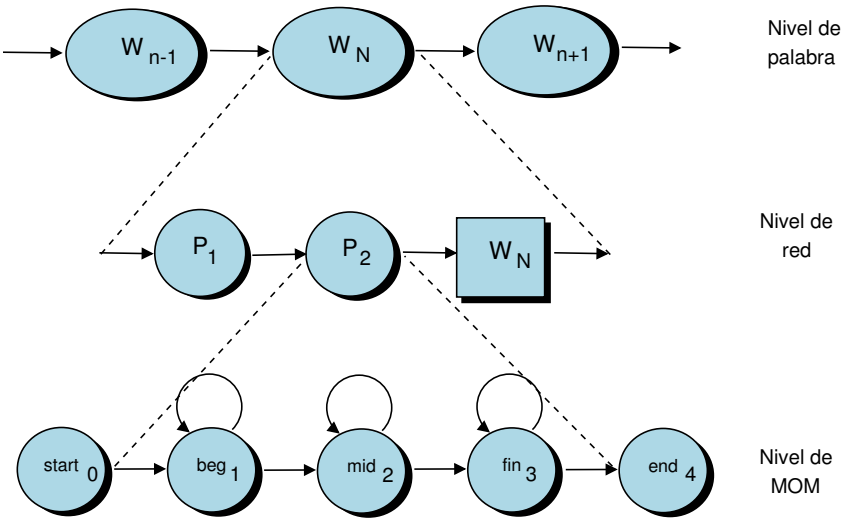

Figura 2.14: Niveles en un grafo de reconocimiento de voz

2.3.1.3. Decodificación usando algoritmos de token passing

El problema de la decodificación dentro del RAV radica en descubrir los caminos en donde el grafo de reconocimiento obtenga la probabilidad más alta. **Token passing** [68] es una manera realmente buena para entender (incluso para implementar) la búsqueda de Viterbi en los MOM. Un token representa el camino parcial a través del grafo de reconocimiento desde el tiempo 0 hasta el tiempo t. En el tiempo 0 el token está localizado en cada posible estado con el que se puede iniciar. En cada etapa los tokens son propagados, y en el momento en que existan bifurcaciones, los tokens son replicados en todas las posibles salidas y son distribuidos paralelamente. Cada vez que estos atraviesan las transiciones y los nodos, la probabilidad del token se va acumulando. Al final del recorrido del grafo solo unos cuantos elementos token permanecen.

Muchas veces es importante tener restricciones gramaticales en el contexto del RAV, principalmente debido a que se puedan mejorar las tasas de reconocimiento, y las cuales se pueden realizar usando el modelo del lenguaje comentado anteriormente. Estas restricciones permiten tener conexiones específicas entre los componentes del modelo y las palabras inmersas en él. La red o grafo resul-

tante será la utilizada en el esquema de decodificación. Si por ejemplo se utiliza la gramática libre de contexto, las reglas gramaticales que permitan la producción serán transformadas en una red sintáctica conforme se haya definido la gramática. En este escenario nos encontramos nodos terminales (palabras), no terminales (dividen las subredes para cada producción que genera la gramática) y las ligaduras (guardan los token y los valores calculados de probabilidad).

2.3.1.4. Flujo del reconocimiento de voz en MMG-MOM

La mayoría de los sistemas de reconocimiento de voz utilizan los MOM para lidiar con la variabilidad temporal de la voz, y usan los MMG para determinar lo bien que cada estado de un MOM se ajusta a una trama o a una pequeña ventana de tramas de coeficientes que representan la señal acústica de entrada [6]. En los sistemas de RAV, el enfoque de aprendizaje generativo más utilizado está basado en MMG-MOM [69], sobre el cual se han desarrollado diversos trabajos [2, 70–74]. Este modelo es parametrizado por $\Lambda = \{\pi, A, B\}$, donde π es un vector de probabilidades de estado inicial, $A = a_{ij}$ es una matriz de probabilidad de transición entre estados, y $B = \{b_1(o_t), \ldots, b_S(o_t)\}$ es un conjunto donde $b_j(o_t)$ representa el MMG del j-ésimo estado. Como vimos en la sección de MOM, $b_j(o_t)$ denota la probabilidad de emisión de la observación o_t dado un estado j en el tiempo t (ver Figura 2.15). Existen diversas distribuciones de observación para este caso, cuando las observaciones son discretas, las distribuciones $b_j(o_t)$ son funciones de masa y cuando las observaciones son continuas, las distribuciones son típicamente especificadas usando una familia de modelos paramétricos [69], comúnmente a través de mezclas Gaussianas, denotadas por la ec. (2.2.1).

La Figura 2.16 presenta el flujo de reconocimiento de voz con la arquitectura de MMG-MOM. En el paso 1 un vector de observación es extraído en un procedimiento de trama por trama. En el paso 2 un modelo fonémico es utilizado como base y la probabilidad de la mezcla Gaussiana ($\log b_j(o_t)$) se calcula para todos los nodos de estado activos. En el paso 3 el algoritmo de Viterbi calcula la rejilla $\delta_t(j)$ para el modelo acústico $p(O|W)$ contemplando todos los nodos de estado activos junto con el modelo de lenguaje (con el uso de n-gramas o gramática libre de

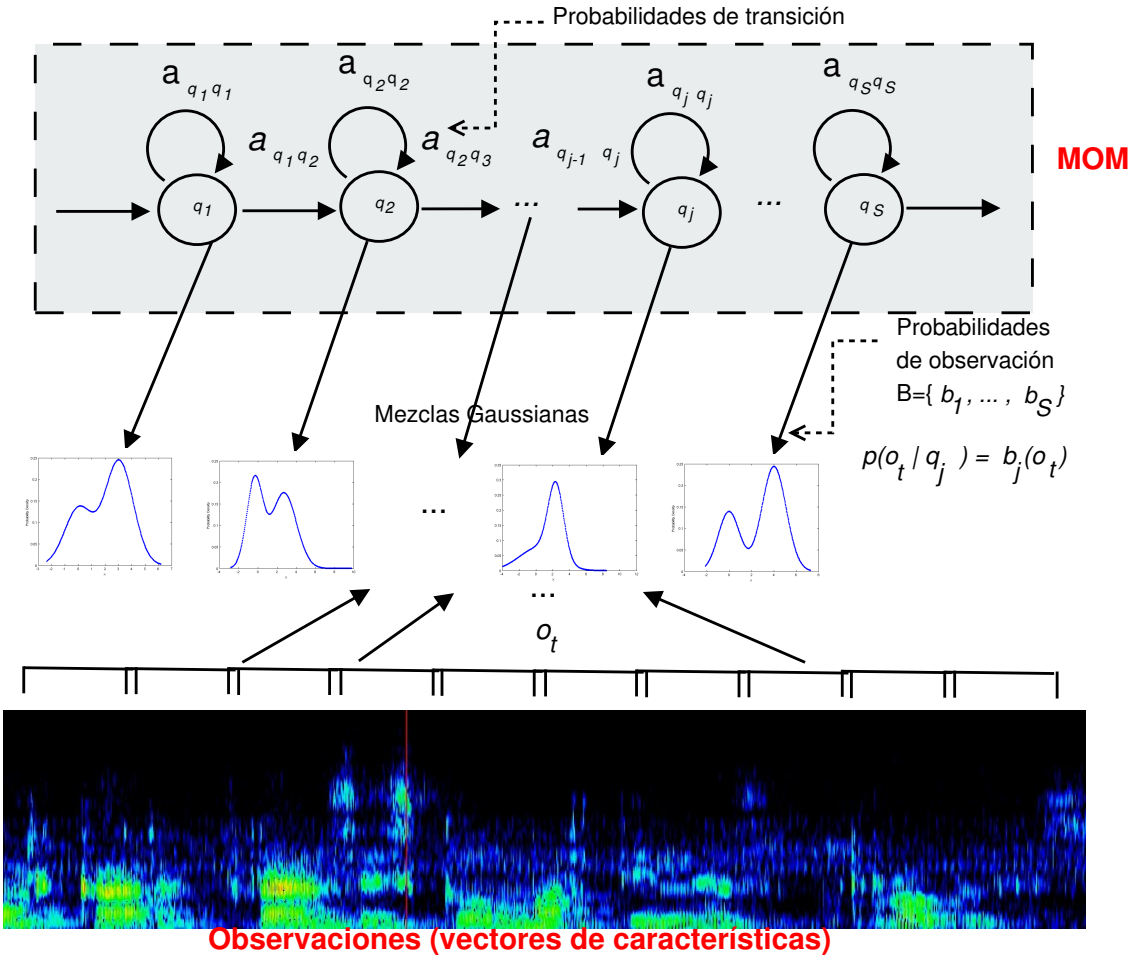

Figura 2.15: Esquema general de la arquitectura MMG-MOM

contexto - GLC) y su probabilidad a priori $p(W)$ (ver ec. (1.3.1)). En el paso 4 el algoritmo de *Beam pruning* se utiliza para descartar los nodos con las probabilidades acumuladas más bajas. Finalmente, en el paso 5 la sentencia con el valor máximo es generada como transcripción de salida [65].

Para este enfoque, como vimos anteriormente, en la fase de entrenamiento con el procedimiento convencional de la estimación de máxima verosimilitud (MLE), los parámetros del MOM son estimados maximizando la probabilidad de los datos de entrenamiento dadas sus correctas transcripciones, y MLE puede ser estimada iterativamente usando el algoritmo de Baum-Welch [75]. Por otro lado, los

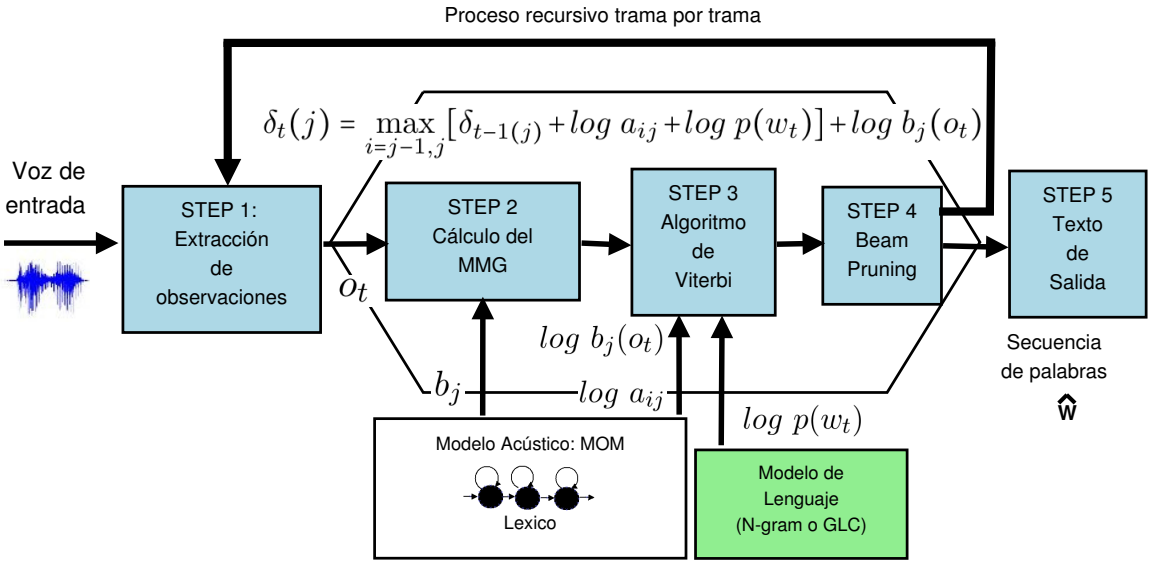

Figura 2.16: Flujo de datos en el sistema de reconocimiento de voz de MMG-MOM

algoritmos de entrenamiento discriminativos toman en cuenta las clases competidoras para optimizar los parámetros. Estos criterios son útiles para datos limitados o supuestos imperfectos en el modelo; maximum mutual information (MMI), minimum phone error (MPE) y minimum classification error (MCE) son algunos ejemplos [76].

Capítulo 3

Modelado no lineal de la señal de voz

En este capítulo se mencionan los principales elementos que cubren el modelado de la señal de voz desde puntos de vista alternativos a las mezclas Gaussianas, primordialmente se enfoca el trabajo en el uso de redes neuronales como estructura no lineal en el tratamiento de la señal de voz.

En la sección 3.1 se mencionan los principios del porqué existe el modelado no lineal de la señal de voz y los principales enfoques que lo abarcan, esto con el fin de modelar de mejor manera el proceso del RAV. En la sección 3.2 se describen las bases del aprendizaje profundo y su rol dentro de la inteligencia artificial. Finalmente, en la sección 3.3 se estudia el enfoque de RNP-MOM como un substituto del modelo convencional de las mezclas Gaussianas en el área del RAV.

3.1. No linealidad en el procesamiento de la señal de voz

Hablar es una habilidad motora que consiste de movimientos controlados y coordinados, desarrollada principalmente por órganos del tracto vocal (glotis, velum, lengua, labios) actuando sobre el aire del conducto respiratorio (traquea, laringe, faringe, boca, nariz) para producir sonidos de voz. Los órganos vocales

generan perturbaciones a las moléculas de aire en diferentes posiciones del tracto vocal, creando la fuente de las señales de voz. Las fuentes de voz más comunes son: 1) vibraciones cuasi-periódicas de las cuerdas vocales (fuente vocalizada); 2) ruido turbulento generado por el paso del aire a través de una presión cuasi-estrecha (generalmente provocada por la lengua) en la cavidad oral (fuente turbulenta); 3) ruido plosivo, seguido por la liberación de aire comprimido detrás de una obstrucción de la cavidad oral (fuente transitoria). La estructura compleja de los sonidos de voz no solo es debido a las características de la fuente que los genera, pero principalmente se debe a las características de la respuesta del tracto vocal que depende de la articulación del tracto vocal. Esta articulación cambia de acuerdo al movimiento de los órganos vocales que modifican su longitud, áreas de sección transversal y características de la respuesta [77].

La estructura de los sonidos de voz se genera por los efectos provenientes de las fuentes de sonidos y las características del tracto vocal. En la ausencia de sonido, el tracto vocal puede ser modelado como un tubo simple y las moléculas de aire pueden pasar por él como un oscilador lineal que responde a perturbaciones con pequeños cambios desde la posición de reposo. Las condiciones son extremadamente más complejas cuando los sonidos de voz son producidos desde el movimiento de los órganos vocales que cambian la condición del tracto vocal. Un modelo común del tracto vocal en estas condiciones es un conjunto de tubos superpuestos de diferente longitud y áreas transversales. Debido a esta longitud y área seccional, cada tubo es sujeto a diferentes presiones de aire, que a su vez genera diferentes fuerzas actuando sobre las moléculas de aire, causando un movimiento considerable. En este caso, una descripción lineal falla al describir esta dinámica compleja y por ende un enfoque no lineal debe ser usado [77].

Las aplicaciones de voz usualmente requieren del cálculo de un modelo de predicción lineal para el tracto vocal. Este modelo ha sido exitosamente aplicado durante varios años, pero tiene algunas desventajas. Principalmente, es incapaz de modelar las no linealidades que implican los mecanismos de producción de voz, y solo un parámetro puede ser corregido: el orden del análisis. Con modelos no lineales, la señal de voz es mejor ajustada y existe mayor flexibilidad

para adaptar el modelo a la aplicación [33, 78]. La forma más sencilla de predecir linealmente una muestra [79] es por medio de una combinación lineal de P muestras previas ponderadas por coeficientes de predicción (a_k), de acuerdo a

$$x[n] \cong \hat{x}[n] = \sum_{k=1}^{P} a_k x[n-k], \tag{3.1.1}$$

produciendo una señal residual o diferencia en la señal de

$$e[n] = x[n] - \hat{x}[n], \tag{3.1.2}$$

de esta manera la función general para un predictor puede ser escrita como

$$\hat{x}[n] = g(\underline{x}[n-1]) \cong x[n] \tag{3.1.3}$$

donde el vector de entrada $\underline{x}$ es

$$\underline{x}[n-1]^T = (x[n-1], x[n-2], \ldots, x[n-P]) \tag{3.1.4}$$

Para un predictor no lineal, la función $g(\cdot)$ debe ser no lineal. En la codificación de voz predictiva utilizando un predictor lineal, solo se requiere fijar el orden de predicción. Teóricamente, mientras más alto es el orden de predicción, más alta será la precisión de la predicción, pero hay una saturación en el rendimiento, especialmente para predicciones de orden elevado. Utilizar un predictor no lineal proporciona mayor factibilidad y mejores resultados, esto debido a que los modelos lineales son óptimos para señales Gaussianas, que no es el caso de las señales de voz [80]. Los modelos lineales son inadecuados para describir de manera precisa aspectos fonéticos como la no linealidad durante la generación de pulsos de la glotis, la presencia de flujo de aire turbulento en el tracto vocal, la correspondencia no lineal entre el tracto vocal y la fuente de la señal. El análisis por predicción no lineal parece ser un método válido para mejorar nuestro conocimiento, incluso sobre la caracterización de la voz esofágica [78].

En los últimos años se ha incrementado el interés en los modelos no lineales aplicados a la señal de voz. Este interés está basado en la no linealidad del

proceso de producción de la voz, y que incluyen aspectos como [33, 79]:

1. La señal residual del análisis predictivo [81].
2. La dimensión de correlación de la señal de voz [82].
3. La fisiología de los mecanismos de producción de voz [83].
4. Funciones de densidad de probabilidad [84].
5. Estadísticas de orden superior [85].

A través de predictores no lineales se pueden modelar las no linealidades existentes en la señal de voz, de tal manera que se pueden alcanzar mayores umbrales de predicción. Ante estas circunstancias, pocas aplicaciones han sido desarrolladas, principalmente debido a la elevada complejidad computacional y a la dificultad de analizar los sistemas no lineales. Sin embargo, los modelos por predicción no lineal han sido aplicados en el reconocimiento del habla de maneras muy directas, reemplazando en cierta medida a los modelos predictivos lineales.

Se contemplan dos enfoques hacia el análisis predictivo no lineal de la señal de voz [33, 79, 80, 86]:

1. *Predicción no paramétrica*: no se asume ningún modelo para la no linealidad. Es un método bastante simple, pero las mejoras sobre un modelo predictivo lineal son más bajas que con un modelo paramétrico no lineal. Un ejemplo de este esquema es un libro de códigos que categoriza varias entradas y salidas (ver ec. (3.1.5)) y el valor predicho puede ser calculado usando el vecino más cercano dentro del libro de códigos. Aunque este método es simple, un orden de predicción bajo debe ser usado. Los principales métodos abarcados en el enfoque no paramétrico suelen ser cuantificación vectorial interpolativa no lineal, métodos análogos de Lorenz, estimaciones por núcleos/densidad de la esperanza condicional,

$$(\underline{x}[n-1], \hat{x}[n]). \tag{3.1.5}$$

2. *Predicción paramétrica*: se asume un modelo de predicción. los principales enfoques utilizados son las series de Volterra (aproximación polinomial), modelos lineales locales, modelos auto-regresivos con umbral, modelos dependientes de estado y las redes neuronales (RBF, MLP, Kohonen Networks).

El uso de modelos predictivos no lineales basados en redes neuronales puede tomar ventajas de algún tipo de combinación entre diferentes predictores no lineales (diferentes redes neuronales, la misma arquitectura de red neuronal entrenada con diferentes algoritmos, o incluso la misma arquitectura y algoritmo de entrenamiento usando un 'bias' diferente y una inicialización aleatoria de pesos). Mientras que en las técnicas predictivas lineales las opciones son más reducidas. Datos de este tipo se pueden revisar en algunos trabajos donde se enfatiza esta situación [33, 79, 80, 86], obteniéndose reducciones en las tasas de error mediante predicción no lineal usando perceptrones multicapa, cuantificación vectorial y coeficientes LPC-CC; incluso obteniéndose mejor flexibilidad y eficiencia con las redes neuronales de perceptrón que con series de Volterra.

3.2. Procesamiento y aprendizaje profundo

El aprendizaje estructurado profundo, o más comúnmente llamado *aprendizaje profundo o aprendizaje jerárquico* ha surgido como una nueva área de investigación sobre el aprendizaje a través de las máquinas [18, 87]; que se suele considerar como una clase de técnica de aprendizaje basado en máquinas que aprovecha diversas capas de procesamiento no lineal para una extracción y transformación de características supervisadas o no supervisadas, así como para clasificación y análisis de patrones; estos niveles modelan relaciones complejas entre datos, donde los conceptos y características de niveles superiores son definidos en términos de los que aparecen en niveles inferiores, teniendo así una jerarquía de características denominada *arquitectura profunda* [88]. Seide et al. [7], basados en sus resultados, sugieren que las redes poco profundas tienen mucha menor efectividad de extracción que las redes profundas. Las arquitecturas poco profundas se han mostrado efectivas en la solución de diferentes problemas simples

y bien delimitados, pero su poder de representación y modelado es limitado y causa dificultades cuando se trabaja con aplicaciones complicadas del mundo real que involucran señales naturales como la voz humana, lenguajes y sonidos naturales, así como imágenes y escenas visuales. Aunado a esto, los mecanismos de procesamiento de información humana (audio y visión) sugieren la necesidad de arquitecturas profundas para extraer estructuras complejas y construir su representación interna de diversas entradas sensoriales.

Históricamente el concepto del aprendizaje profundo se originó en el uso de redes neuronales artificiales (específicamente en las redes neuronales de nueva generación). Donde comúnmente se hace referencia a redes neuronales *feed-forward* (de alimentación hacia adelante), o MLP (perceptrón multicapa) de varias capas ocultas, comúnmente llamadas redes neuronales profundas (RNP). En ciertos trabajos [18, 87] se ha introducido una clase de modelo generativo profundo llamada red de creencia profunda (deep belief network, DBN), la cual está compuesta por una pila de máquinas restrictivas de Boltzmann (restricted Boltzmann machines, RBM), cuyo componente principal es un algoritmo robusto de aprendizaje de capa por capa que optimiza los pesos en una complejidad lineal al tamaño y profundidad de la red.

De manera independiente y sorprendentemente, inicializando los pesos de un MLP con su correspondiente DBN configurada a menudo produce mucho mejores resultados que con pesos aleatorios. Un MLP con varias capas ocultas, o también denominada red neuronal profunda, es aprendido con un pre-entrenamiento DBN no supervisado, seguido de un ajuste de retro-propagación; a estas arquitecturas se les suele llamar DBN en la literatura [89–91]. Varios investigadores han sido cautelosos en distinguir las DNN de las DBN [92, 93], ya que cuando las DBN son usadas para inicializar el entrenamiento de las DNN, la red resultante es algunas veces llamada DBN-DNN [92].

3.2.1. Redes neuronales artificiales

Una *red neuronal* es un paradigma general computacional y matemático que modela las funciones de un sistema neuronal biológico (grupo de neuronas interconectadas); la red neuronal de primera generación recibió el nombre de perceptrón [94]. Existen numerosas representaciones de una neurona, quizás la más común es la observada en la Figura 3.1.

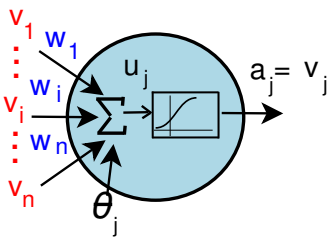

Figura 3.1: Modelo de nodo o unidad de salida (neurona) de McCulloch y Pitts

Una neurona está formada por dos partes: la función de red y la función de activación. La primera determina cómo las entradas de la red v_i son combinadas dentro de la neurona en el escalar u_j. En la ec. (3.2.1) se muestra una combinación lineal con pesos de las entradas,

$$u_j = \sum_{i=1}^{N} w_{ij} v_i + \theta_j, \tag{3.2.1}$$

donde w_{ij} son parámetros conocidos como pesos de la sinapsis de cada neurona i hacia la neurona j; la cantidad θ_j es conocida como elemento de tendencia o *bias* y es usado para modelar el hiperplano de decisión. Aunque existen otros tipos de funciones de combinación de las entradas, esta es la más común. La salida de la neurona, denotada por a_j, está relacionada a la entrada u_j a través de una transformación lineal o no lineal llamada función de activación, dada por:

$$a_j = f(u_j). \tag{3.2.2}$$

En varios modelos de redes neuronales, se han propuesto diferentes funciones

de activación, como por ejemplo la lineal: $f(u_j) = Du_j + E$, solo por mencionar alguna.

Topologías en las redes neuronales

En una red neuronal, múltiples neuronas son interconectadas para formar una red que facilita el cómputo distribuido. La configuración de la interconexión puede ser descrita eficientemente con un grafo dirigido; el cual consiste en un grupo de nodos (unidades de salida o neuronas) y arcos dirigidos (enlaces de sinápsis). Un modelo de red neuronal de perceptrón multicapa MLP consiste en una red con capas de alimentación directa (unidireccional o hacia adelante) de neuronas de McCulloch y Pitts, y donde cada neurona tiene una función de activación no lineal que es continuamente diferenciable. Un esquema típico es el que se muestra en la Figura 3.2 [94]. En donde cada círculo representa una neurona individual, estas neuronas son organizadas en capas, como se pueden observar etiquetadas como capa oculta 1, capa oculta 2 y la capa de salida; mientras que la entrada es etiquetada como capa de entrada, y en la cual normalmente no hay un modelo neuronal implementado. El término de capa oculta se refiere al hecho de que las salidas de estas neuronas alimentarán a neuronas de una capa superior, y por ende son ocultas al usuario que solo observa las salidas de las neuronas de la capa exterior.

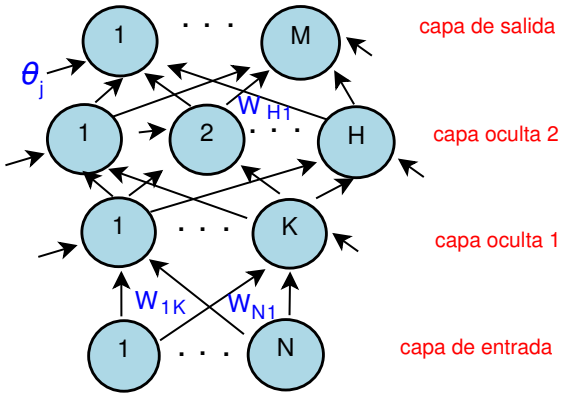

Figura 3.2: Perceptrón de 3 capas

3.2.2. Redes neuronales profundas

Las redes neuronales profundas (RNP) son un tipo de red neuronal artificial de alimentación directa (hacia adelante) que tiene más de una capa de unidades ocultas entre las entradas y las salidas [95]. Cada unidad oculta j utiliza típicamente la función logística, aunque también puede usar una función tangente hiperbólica [6], una unidad lineal rectificada (ReLU) [96, 97] o de tipo maxout [98, 99] como función de activación, esto con el fin de mapear (una transformación no lineal) sus entradas (z_j) a partir de la capa inferior a un estado escalar v_j, que es enviado a la capa superior, como se muestra en la ecuación [6, 100]:

$$v_j = \text{logistic}(z_j) = \frac{1}{1 + e^{-z_j}}, \qquad z_j = \theta_j + \sum_i v_i w_{ij}, \tag{3.2.3}$$

donde la función logistic representa una función sigmoidea, θ_j es el 'bias' de la unidad j, i es el índice de las unidades de la capa inferior y w_{ij} es el peso de una conexión de la unidad i de la capa inferior dirigida hacia la unidad j (capa superior).

Un número significativo de propuestas que se encuentran en la literatura utilizan las series de Volterra con no linealidad cuadrática, redes de funciones de base radial (RBF- radial basis function nets), que también implican un modelo no lineal cuadrático. En [79] se propone el uso de redes de perceptrones multicapa, debido a que tienen más flexibilidad en la no linealidad. Es fácil mostrar que un MLP con una función de transferencia (o de activación) sigmoidea [$f(u) = \frac{1}{1+e^{-u/T}}$] permite modelar no linealidades cúbicas. Se cree que es un hecho importante, dado que la no linealidad presente en el mecanismo de predicción de voz humana se debe probablemente a fenómenos de saturación en las cuerdas vocales.

En las décadas recientes se han reportado diversos estudios que han tratado con la predicción no lineal de la señal de voz. Muchos de ellos se han centrado en predicción paramétrica basada en redes neuronales, debido a que es un enfoque que ofrece mejoras sobre el análisis LPC. En los años recientes, la comunidad de reconocimiento de voz ha recuperado el interés sobre las redes neuronales, que fueron populares en los 80's y 90's pero que no pudieron superar significati-

vamente la exitosa combinación de los MOM con modelos acústicos basados en mezclas Gaussianas. Existen tres factores que influyeron para el resurgimiento de las redes neuronales como modelos acústicos de alta calidad [16]:

1. Haciendo las redes más profundas se puede lograr mejor desempeño, surgiendo el término de redes neuronales profundas (RNP) [7, 95, 101–103].

2. Inicializando razonablemente los pesos y utilizando hardware más poderoso hace posible que el entrenamiento de la red neuronal sea más eficiente [6, 10, 104, 105].

3. Utilizar un mayor número de unidades de salida (dependientes del contexto) realmente mejora su desempeño [8, 9, 12, 19, 53, 106].

Utilizar una red neuronal profunda y ancha puede acarrear una gran demanda computacional durante el proceso de entrenamiento y por eso es una de las razones del porqué hasta años recientes se comenzó de nuevo la reexploración de este tipo de aprendizaje profundo de una manera más seria [88]. En años recientes las RNP han sido exitosamente aplicadas en diferentes tareas de procesamiento de voz, como reconocimiento de fonemas, detección de palabras fuera de vocabulario, medidas de confianza, entre otras. Mientras que las RNP han generado avances significativos con respecto a los sistemas basados en MMG en diferentes tareas, pocas veces se han puesto a prueba para sistemas con entornos con ruido. Esto es especialmente de notar dado que las redes utilizan características simples en el dominio espectral y una función objetivo a nivel de trama y solo requiere un sencillo paso de decodificación. En contraste, los algoritmos de MMG-MOM son más complejos, dado que requieren múltiples pasos de reconocimiento, y en algunos casos emplean clasificadores múltiples.

Las arquitecturas del aprendizaje profundo a través de la computadora se pueden clasificar en [88]:

1. *Redes profundas para aprendizaje generativo o no supervisado*. Tratan de capturar la correlación de orden superior de los datos observados o visibles con propósitos de un análisis o síntesis de patrones cuando no existe

información disponible de las etiquetas de las clases que están sujetas al objetivo (no se conoce la salida). Entre ejemplos de ello tenemos a la máquina restrictiva de Boltzmann (RBM), redes de creencia profunda (DBN), máquina profunda de Boltzmann (DBM), auto-codificadores regularizados, entre otros.

2. *Redes profundas para aprendizaje supervisado*. Intentan proporcionar directamente la habilidad de discriminación para propósitos de clasificación de patrones, normalmente caracterizando la distribución a posteriori de clases (etiquetas) condicionadas con base a los datos visibles. Los datos de salida (datos deseados) son siempre conocidos de forma directa o indirecta. También se les suele denominar redes neuronales discriminativas. Ejemplos de ello son redes neuronales profundas, redes neuronales recurrentes (RNN), redes neuronales convolucionales (CNN), entre otras.

3. *Híbridas*. La meta es lograr una discriminación asistida, a menudo de manera significativa, con la salida de una red profunda discriminativa o generativa (no hay que confundir este esquema con RNP-MOM, a veces llamado también enfoque híbrido).

Un enfoque muy útil en ese sentido es el cómo un modelo de aprendizaje no supervisado o generativo (donde no necesariamente tiene que ser probabilístico; pero en caso que lo sea es fácil de interpretar y de incrustar en el dominio del conocimiento, fácil de hacer, fácil de manejar la incertidumbre, pero a la vez intratable para aprender e inferir en sistemas complejos) puede enórmemente mejorar el entrenamiento de una red, así como otro tipo de modelo de aprendizaje supervisado o discriminativo (los cuales suelen ser más eficientes para entrenarse y probarse, así como flexibles para construirse y más adecuados para problemas complejos) a través de una optimización o regularización.

Siniscalchi et al. [100] han comparado las RNPs y los perceptrones de una capa para clasificar fonemas y atributos fonémicos. Con evidencia experimental, varias redes de entre 5 y 7 capas ocultas son configuradas, y con hasta 2048 nodos ocultos por capa, mostrando que las redes profundas alcanzan mejoras

significativas sobre los perceptrones de una capa. Las RNPs han generado un progreso significativo con respecto a los sistemas basados en MMG, los cuales raramente han sido probados en entornos de ambiente ruidoso. Algunos trabajos [95, 107, 108] han evidenciado la efectividad de los modelos acústicos basados en redes multicapa en entornos de este tipo, encontrando que siendo entrenadas en entornos con datos de diversidad de condiciones, y con datos que incluyen compensación de ruido explícito alcanzan un buen nivel de desempeño.

3.3. Inspección de modelos híbridos: modelos ocultos de Markov y redes neuronales artificiales (RNP-MOM)

Los sistemas tradicionales de RAV son derivados de procesos de producción de voz con modelos basados en MOM, en los cuales las mezclas Gaussianas son usadas para modelar las probabilidades de observación de estados (la arquitectura MMG-MOM). Recientemente este enfoque clásico puede ser eficientemente substituido por las redes neuronales profundas.

Gracias a un procedimiento de pre-entrenamiento [18], los parámetros de la red neuronal pueden ser aprendidos en una forma más adecuada [101]. Además, otros factores que han contribuido al reciente éxito de las RNPs son la disponibilidad de más datos de entrenamiento y mejoras en la ingeniería de software. Uno de los parámetros críticos de los MOM es la distribución de probabilidad de observación de estado. Recientemente las así llamadas redes neuronales profundas dependientes del contexto han sido propuestas para reemplazar las mezclas Gaussianas con el fin de calcular estas probabilidades de observación de estado para todos los nodos ligados del MOM [9, 12].

Los sistemas de reconocimiento basados en MOM son efectivos en muchas circunstancias, pero sufren de algunas limitaciones importantes que complican su aplicabilidad en los sistemas de reconocimiento del habla en entornos del mundo real. Se han hecho intentos para superar estas limitaciones con la adopción

de las redes neuronales como un paradigma alternativo del RAV, pero no fueron exitosas cuando lidiaban con secuencias extensas de señales de voz. Después de los 80's y 90's algunos investigadores han comenzado a explorar una área que involucra la combinación de MOM y redes neuronales como una arquitectura híbrida [56].

Los sistemas de RAV basados en MMG-MOM son sensibles a la disparidad entre la etapa de entrenamiento y la de prueba, particularmente debido a la no concordancia producida por el entorno ruidoso. Normalmente, los algoritmos actuales fallan en el intento de proporcionar un entorno robusto para el reconocimiento en estas situaciones, principalmente debido a que los métodos para realzar las características de la señal intentan eliminar el ruido que afecta las observaciones antes de realizar el proceso de reconocimiento [107, 109]. Los métodos de adaptación del modelo dejan las observaciones sin cambios, en lugar de actualizar los parámetros del modelo del reconocedor para que sea más representativo en relación a la señal de voz observada [110–112]. Muchos de estos enfoques pueden ser mejorados utilizando entrenamiento con datos en diversas condiciones, así como con técnicas de entrenamiento adaptativas [95].

Recientemente se ha abordado una nueva forma de modelado acústico basado en lo que se conoce como RNP. Este modelado acústico está muy relacionado con el enfoque original de arquitecturas híbridas de red neuronal artificial y modelos de markov RNA-MOM [113], salvo por dos características clave. La primera, las redes son entrenadas para predecir estados acústicos dependientes del contexto llamados senones. La segunda, estas redes tienen más capas que las tradicionales redes entrenadas en décadas anteriores [95].

Existen varios paradigmas de aprendizaje profundo en el reconocimiento de voz, una arquitectura común es la que se describe a continuación.

3.3.1. Estructura de una red neuronal profunda

Como hemos comentado, una RNP es un MLP de varias capas ocultas entre sus entradas y salidas. Un MLP es comúnmente utilizado para clasificar una ob-

servación acústica o_t dentro de un conjunto Q de estados fonéticos dependientes del contexto. Es un clasificador no lineal que puede ser interpretado como un conjunto de modelos logarítmicos lineales, y en donde cada capa oculta modela la probabilidad del siguiente salto de un conjunto de variables binarias ocultas h dadas las variables visibles de entrada v, mientras que la capa de salida modela las probabilidades a posteriori.

En consecuencia, la estructura de una RNP es la misma que la de un perceptrón multicapa (MLP). Un perceptrón de $(L+1)$-capas es usado para modelar la probabilidad a posteriori $p_{q|o}$ (ver Figura 3.3) de un estado ligado q de un MOM dado un vector de observación o, es decir, la red modela la probabilidad a posteriori de que la observación o pertenezca a la clase q en el tiempo t, esto es $p(C_q|o_t)$. Las primeras L capas, $l = 0, 1, \ldots, L-1$ son capas ocultas que modelan las probabilidades a posteriori de las capas ocultas h^l dados los vectores de entrada v^l provenientes de la capa anterior, mientras que la capa superior L es utilizada para calcular la probabilidad a posteriori para todos los estados ligados usando la función softmax [7, 101, 103]:

$$p^l_{h_j|v}(h^l_j|v^l) = \frac{1}{1+e^{-z^l_j(v^l)}} = \sigma(z^l_j(v^l)),\ 0 \le l < L, \tag{3.3.1}$$

$$p^L_{q|v}(q|v^L) = \frac{e^{z^L_q(v^L)}}{\sum_{q'} e^{z^L_{q'}(v^L)}} = \mathrm{softmax}_q(z^L(v^L)), \tag{3.3.2}$$

$$z^L(v^L) = (w^l_{ij})^T v^l + \theta^l, \tag{3.3.3}$$

donde $v^0 = o_t$ es la observación acústica o_t en el tiempo t, w^l_{ij} (de la neurona i a la unidad j) y θ^l denotan la matriz de pesos y el vector de biases para la capa oculta l, h^l_j y $z^l_j(v^l)$ denotan el j-ésimo componente de tipo nodo oculto h^l, y su excitación $z^l(v^l)$ respectivamente; $\sigma(\cdot)$ es la función de activación $\mathrm{logistic}$, y q' es el conjunto de todos los estados ligados del modelo oculto de Markov asignados a la capa de salida, de tal forma que en la ecuación se pueda obtener una normalización en la función de activación. De esta forma, la probabilidad a

posteriori $p(q_t|o_t) = p^L_{q|v}(q|v^L) = p(C_q|o_t) = y^L_t(q)$.

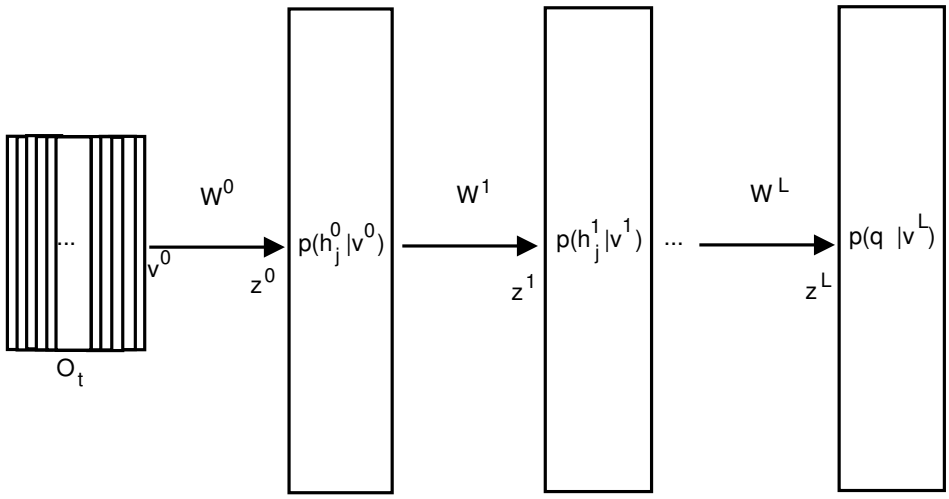

Figura 3.3: Generalización de red neuronal profunda

3.3.2. Inicialización de los pesos con pre-entrenamiento usando redes de creencia profunda

El problema de entrenar un perceptrón multicapa profundo, usando el algoritmo de retro-propagación, es que la función objetivo es no convexa y el algoritmo puede quedar atrapado fácilmente en mínimos locales [8]. Cuando más capas ocultas son agregadas al modelo [114], se vuelve más difícil encontrar un buen óptimo local [54]. Por tanto, es muy importante inicializar los parámetros del perceptrón con algún método eficiente de pre-entrenamiento. Hinton et al. [18] han propuesto utilizar un algoritmo de pre-entrenamiento basado en máquinas restrictivas de Bolztmann (restricted Boltzmann machines, RBMs) con el objetivo de poder inicializar los parámetros de la red neuronal profunda.

Las máquinas restrictivas de Bolztmann son un modelo generativo de dos capas basado en una función de energía asignada a cada configuración de vectores de estados visibles y ocultos [115]. Entonces seremos capaces de inicializar todos los pesos de la red neuronal usando los pesos de conexión de las máquinas de

Bolztmann. De esta manera, una pila de RBMs puede ser entrenada capa por capa y ser usada para los parámetros de la RNP; además, la primera capa de la red neuronal corresponde a máquinas restrictivas de Boltzmann de tipo Gaussiana-Bernoulli y cada una de las otras capas ocultas corresponden a máquinas de Boltzmann de tipo Bernoulli-Bernoulli [102].

Esta pila de RBMs es llamada red de creencia profunda (deep belief net, DBN) [18], y el proceso es llamado pre-entrenamiento [9]. Después de eso, a la RNP se le agrega una capa de salida de tipo softmax con pesos inicializados aleatoriamente para posteriormente aplicar el algoritmo estándar de retropropagación con el gradiente descendente (stochastic gradient descent, SGD), y utilizar este mecanismo para afinar todos los parámetros dentro de la red con un criterio de entrenamiento supervisado [101]. En las RBMs, la unidades visibles corresponden a vectores de entrada, y las unidades ocultas equivalen a detectores de características. Las RBMs pertenecen a modelos basados en energía, cuya probabilidad conjunta está definida por medio de la función de energía [9, 19, 54]

$$E(v,h) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} c_j h_j - \sum_{i,j} v_i h_j w_{ij}, \tag{3.3.4}$$

donde v_i y h_j son los estados de la unidad visible i y la unidad oculta j, a_i y c_j son sus biases, y w_{ij} corresponde al peso de conexión entre ellos. La ec. (3.3.4) requiere una leve modificación para representar una máquina de Boltzmann de tipo Gaussiana-Bernoulli [6]. La probabilidad de una configuración particular de las unidades ocultas y visibles está dada en términos de la energía de esa configuración según

$$p(v,h) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}. \tag{3.3.5}$$

Debido a que no existen conexiones directas entre las unidades ocultas en una RBM, la distribución condicional de capa a capa se obtiene por $P(v_i = 1|h) = \text{sigmoid}(a_i + \sum_j h_j w_{ij})$ y $P(h_j = 1|v) = \text{sigmoid}(c_j + \sum_i v_i w_{ij})$. Las RBMs son entrenadas con un criterio de máxima verosimilitud, y de una forma aproximada, usando el algoritmo de divergencia de contraste (contrastive divergence, CD) [54, 115]. Las

reglas de aprendizaje están definidas como:

$$\Delta w_{ij} = \eta(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}), \tag{3.3.6}$$

$$\Delta a_i = \eta(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}), \tag{3.3.7}$$

$$\Delta c_j = \eta(\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}), \tag{3.3.8}$$

donde η es la tasa de aprendizaje, $\langle v_i h_j \rangle_{\text{data}}$ es la esperanza que corresponde a la frecuencia con la cual la unidad visible v_i y la unidad oculta h_j están unidas en el conjunto de entrenamiento, y $\langle v_i h_j \rangle_{\text{model}}$ es la esperanza equivalente bajo la distribución definida por el modelo. Las ecuaciones (3.3.7) y (3.3.8) indican la misma regla de aprendizaje que la ec. (3.3.6), excepto que las primeras corresponden a estados individuales. Una diferencia importante con respecto a las redes neuronales clásicas es la inicialización de los pesos con este esquema de entrenamiento (pre-entrenamiento) en lugar de utilizar una inicialización aleatoria antes de la afinación final de los parámetros (fine-tuning). Sin embargo, Seide et al. [8] han demostrado que el pre-entrenamiento ayuda, pero no es un factor crucial para redes de más de 5 capas ocultas.

3.3.3. Entrenamiento de la red (fine-tuning después del pre-entrenamiento)

Los perceptrones multicapa a menudo son entrenados con el algoritmo de retro-propagación optimizado con el gradiente estocástico descendente (SGD) [7, 8, 101, 103]:

$$(w_{ij}^l, \theta^l) \leftarrow (w_{ij}^l, \theta^l) + \eta \frac{\partial J}{\partial (w_{ij}^l, \theta^l)},\ 0 \le l \le L, \tag{3.3.9}$$

donde J es la función objetivo y η es la tasa de aprendizaje. Se pueden utilizar criterios a nivel de trama para J, tales como la entropía cruzada (CE) [45] o la

entropía cruzada enfatizada [47]; mientras que entre los criterios discriminativos a nivel de secuencia tenemos el maximum mutual information (MMI), minimum phone error (MPE) y state-level minimum Bayesian risk (sMBR) [45, 46]. En los sistemas de RAV, por ejemplo, J es configurada para maximizar la probabilidad logarítmica total a posteriori sobre todas las T muestras de entrenamiento $O = \{o_1, \ldots, o_T\}$ dadas las etiquetas de estado correspondientes q_t, es decir

$$J(O) = \sum_{t=1}^{T_{corpus}} \log p_{q|o}(q_t|o_t). \tag{3.3.10}$$

Esto es equivalente a minimizar el criterio de entropía cruzada

$$J_{CE}(w, \theta; O, d) = -\sum_{t=1}^{T_{corpus}} \sum_{q=1}^{C_{clases}} d_t(q) \log p(C_q|o_t) = -\sum_{t=1}^{T_{corpus}} \sum_{q=1}^{C_{clases}} d_t(q) \log y_t^L(q) \tag{3.3.11}$$

donde $d_t(q) = p_{emp}(q_t|o_t)$ es la probabilidad empírica (la salida deseada, calculada con el conjunto de entrenamiento utilizando el algoritmo embebido de Viterbi y un sistema de mezclas Gaussianas base) que la observación o_t pertenezca a la clase q (estado de tri-fono ligado de MOM) en el tiempo t, y $y_t^L(q) = p_{rnp}(q_t|o_t)$ es la misma probabilidad estimada a partir de la RNP. Además, minimizar la función de entropía cruzada es equivalente a minimizar el criterio de la divergencia de Kullback-Leibler (KLD) [20]. En la mayoría de los casos se suele utilizar una etiqueta de clase restrictiva, entonces $d_t(q) = \delta_{q,s}$; donde δ es la delta de Kronecker, y s es la etiqueta de clase en el conjunto de entrenamiento para la observación o. De esta manera, el criterio de la entropía cruzada se puede reducir a la verosimilitud logarítmica negativa (negative log-likelihood, NLL)

$$J_{NLL}(w, \theta; O, d) = -\sum_{t=1}^{T_{corpus}} \log p(C_s|o_t) = -\sum_{t=1}^{T_{corpus}} \log y_t^L(s), \tag{3.3.12}$$

donde C_s es el estado de tri-fono ligado (senón) objetivo (salida deseada) indicado por el alineamiento forzado en el algoritmo embebido de Viterbi (la etiqueta de clase para la observación o_t).

Para completar el entrenamiento de la red neuronal, los gradientes (ya que se tienen varias capas) pueden ser calculados como sigue:

$$\begin{aligned}
&\frac{\partial J}{\partial w_{ij}^l} = \sum_t v_t^l (\omega_t^l e_t^l)^T, \qquad \frac{\partial J}{\partial \theta^l} = \sum_t \omega_t^l e_t^l, \\
&e_t^L = (\log \operatorname{softmax})'(z^L(v_t^L)), \\
&e_t^{l-1} = w_{ij}^l \cdot \omega_t^l \cdot e_t^l, \qquad \text{for } \ 0 \le l < L, \\
&\omega_t^l = \begin{cases} \operatorname{diag}(\sigma'(z^l(v_t^l))), & \text{for } \ 0 \le l < L, \\ 1, & \text{else}, \end{cases}
\end{aligned} \tag{3.3.13}$$

donde la señal de error $e_t^l = \partial J / \partial v_t^{l+1}$ es retro-propagada; las derivadas componente a componente de las funciones de activación sigmoideas son $\sigma_j'(z) = \sigma_j(z) \cdot (1 - \sigma_j(z))$ y la derivada de la función logarítmica de softmax es calculada como $(\log \operatorname{softmax})_j'(z) = \delta_{q_t,j} - \operatorname{softmax}_j(z)$; y δ es la delta de Kronecker. Este procedimiento es conocido como *algoritmo de retro-propagación de error* [116].

3.3.4. Decodificación

En los modelos acústicos basados en RNP, la red neuronal genera probabilidades a posteriori de las unidades de modelado sobre las características acústicas de entrada. En un entorno de MOM, el modelo acústico es siempre formulado como [9, 54]:

$$p(O|W) = \sum_q p(O, q|W) p(q|W), \tag{3.3.14}$$

$$\cong \max_q \ \pi(q_0) \prod_{t=1}^{T} a_{q_{t-1} q_t} \prod_{t=0}^{T} p(o_t|q_t), \tag{3.3.15}$$

donde el término $p(O|W)$ es el modelo acústico en la ec. (1.3.1), o_t es el vector de observación acústica en el tiempo t, W es el conjunto total de palabras de referencia, q_t es el estado del MOM en el tiempo t, a_{ij} es la matriz de probabilidades de

transición entre los estados del MOM y $\pi(q_0)$ es la probabilidad inicial del MOM. En los sistemas de reconocimiento basados en MMG, la probabilidad de observación $p(o_t|q_t) = b_{q_t}(o_t)$ es directamente modelada por las mezclas Gaussianas, y significa la probabilidad que el estado q emita la observación o en el tiempo t. Sin embargo, en los sistemas basados en RNP, la probabilidad de observación de estado (verosimilitud escalada) $p(o_t|q_t)$ es derivada de acuerdo a la ec. (3.3.16). Los modelos acústicos que utilizan redes neuronales pueden ser entrenados usando el algoritmo embebido de Viterbi utilizando como base los MMG, aunque sin embargo Senior et al. [117, 118] han explorado un enfoque para entrenar modelos de redes neuronales sin necesidad de utilizar mezclas Gaussianas, así como tambien Zhang et al. [119]. En el caso de los MMG, las unidades de modelado son enviadas como unidades de modelado para las RNP por medio de un alineamiento forzado, las características acústicas de entrada son etiquetadas, y entonces la red neuronal pre-entrenada es ajustada discriminativamente usando el algoritmo de retro-propagación. Por tanto, cuando un esquema de MMG-MOM se usa, la fase de decodificación puede ser realizada a través del cálculo de probabilidades de emisión de estado usando las redes neuronales como:

$$p(o_t|q_t) = \frac{p(q_t|o_t)p(o_t)}{p(q_t)}, \tag{3.3.16}$$

donde o_t son los vectores de observaciones acústicas a los cuales se les agregan las tramas vecinas dentro de una ventana de contexto, y q_t es el estado ligado del MOM de tipo tri-fono basado en un árbol de decisión fonético como en un sistema de tri-fono de MMG-MOM normal. La probabilidad a posteriori $p(q_t|o_t)$ está dada por la capa de salida de la RNP para el estado ligado correspondiente q_t, y $p(o_t)$ puede ser ignorada ya que es irrelevante a q_t. La probabilidad a priori $p(q_t)$ se calcula aproximadamente contando las tramas pertenecientes a cada estado q_t basándose en un alineamiento forzado de estado usando un esquema como el MMG-MOM o algún otro modelo [7, 53, 101, 102].

La red neuronal es entrenada para predecir los estados dependientes del contexto (tri-fonos) en la forma de senones (estados ligados). Este proceso de entre-

namiento puede ser llevado a cabo empleando un enfoque que incluye mejoras como [95]: a) entrenamiento con datos de diversas condiciones, b) mejora en las características con el fin de eliminar las distorsiones antes del entrenamiento, c) la incorporación de un modelo de ruido dentro de la propia red.

Mejoras en las tasas de error son alcanzadas de manera significativa a través de modelos híbridos (RNP-MOM) con redes neuronales pre-entrenadas con redes de creencia profunda [6, 9, 11]. Además, los estudios hechos en [51, 52] muestran que aunque el uso de redes neuronales es más complejo que los modelos Gaussianos, mejores resultados son obtenidos en las tasas de error, incluso en sistemas de vocabulario extenso [53, 54].

En la figura 3.4 se puede apreciar de manera condensada cómo el mecanismo de red neuronal substituye el cálculo de las probabilidades de emisión de estado dada una observación acústica, $b_j(o_t)$.

Además, en la figura 3.5 se observa el flujo de decodificación en el proceso de RAV (el cual es el mismo mostrado en la figura 2.16) que posiciona el lugar de aplicación de las redes neuronales dentro del proceso de cálculo de emisión de estados, reemplazando a las mezclas Gaussianas en dicho proceso.

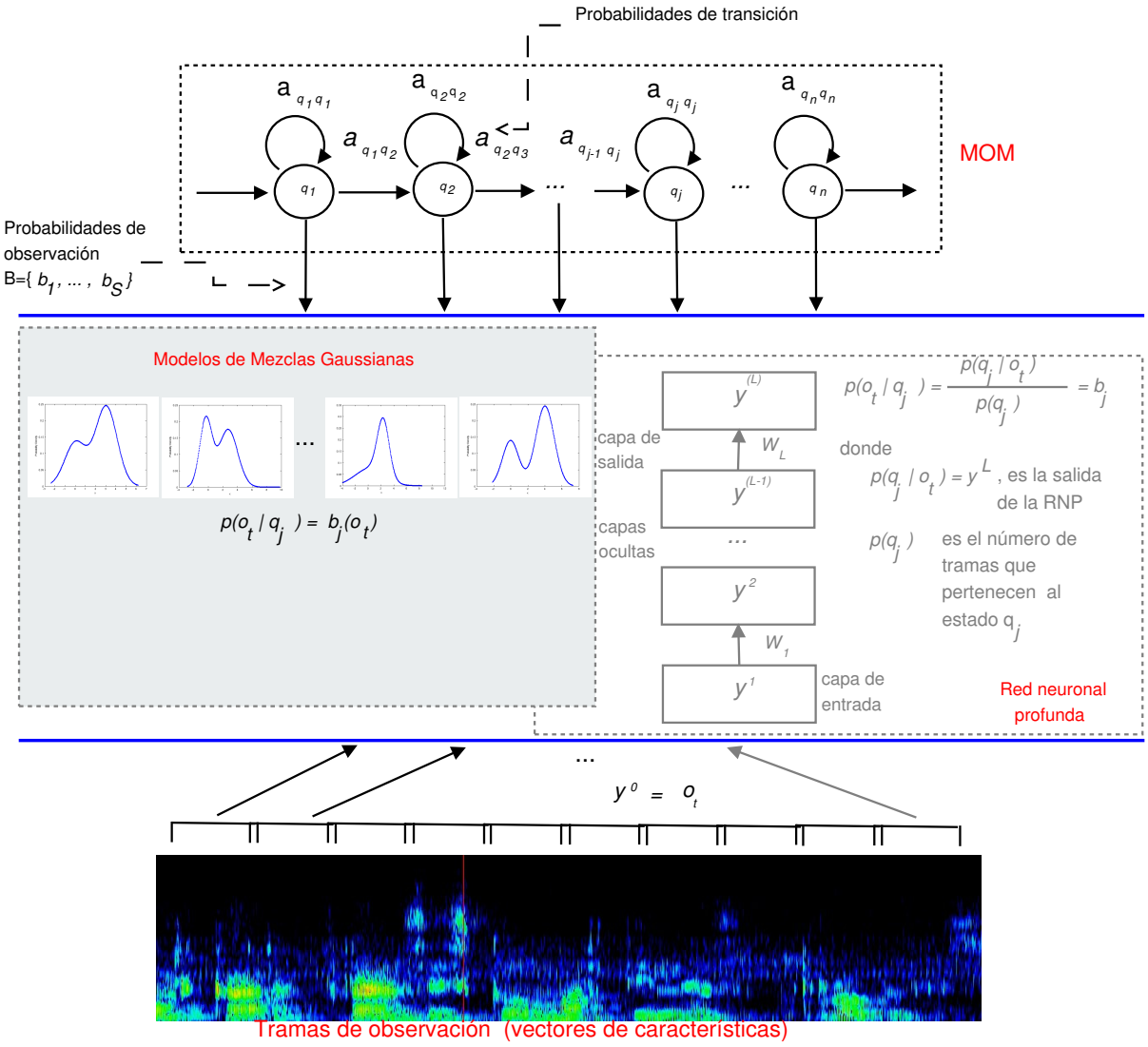

Figura 3.4: Arquitectura de RAV donde las mezclas Gaussianas son substituidas por las redes neuronales

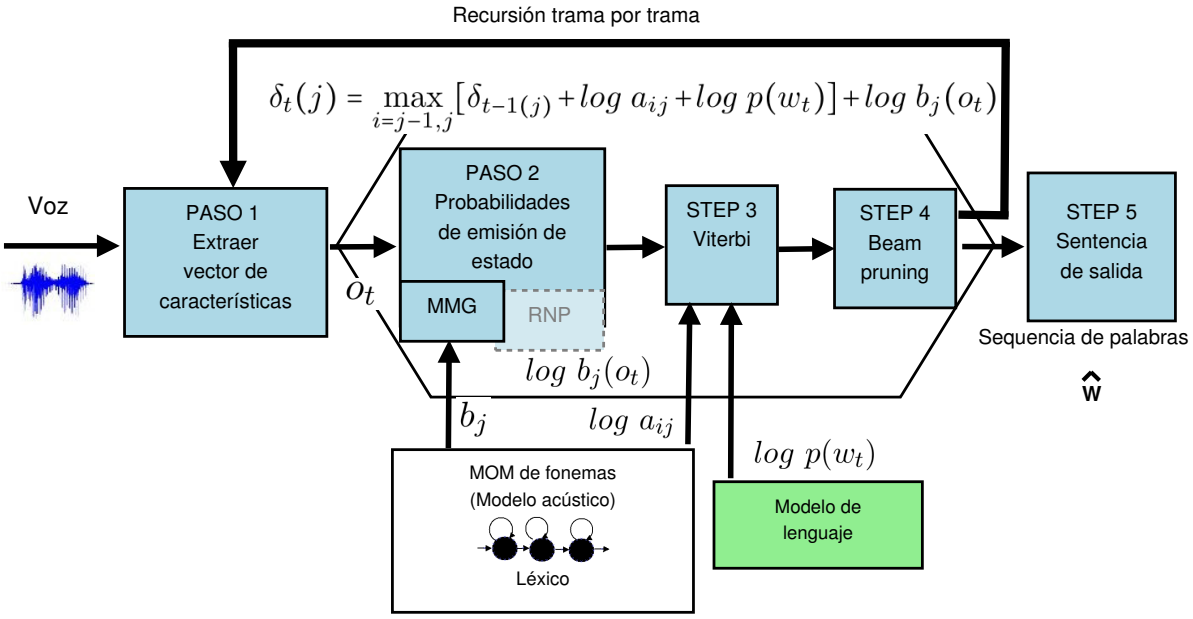

Figura 3.5: RNP (redes neuronales profundas) substituyendo el cálculo de probabilidades de emisión de estado en el flujo de datos del decodificador de RAV

Capítulo 4

Entrenamiento de redes neuronales mediante funciones de costo no uniformes en modelos híbridos de reconocimiento de voz

Los sistemas de RAV han evolucionado recientemente usando las redes neuronales con el fin de encontrar la concordancia de las observaciones acústicas de entrada (tramas) a un estado del MOM [6, 10, 14, 56, 120, 121]. El modelado acústico con redes neuronales artificiales ha mostrado que estas pueden obtener mejores tasas de reconocimiento por palabras (WER) en comparación con los esquemas de MMG-MOM en la mayoría de las tareas de reconocimiento de voz en diferentes ambientes (por ejemplo, [6–9, 53, 54, 95, 122–125]). El así denominado enfoque de RNP-MOM es comúnmente entrenado con ayuda del algoritmo embebido de Viterbi tomando como base los MMGs [126], aunque también el modelo con redes neuronales puede ser hecho explorando enfoques de entrenamiento sin MMGs [117–119].

Distintos criterios de entrenamiento pueden ser usados para optimizar los pesos de una red neuronal, tales como [127–130]: i) entrenamiento a nivel de trama y ii) entrenamiento discriminativo por secuencia. Es común emplear la entropía cruzada (CE) para un entrenamiento basado en tramas [131], mientras

que MMI (maximum mutual information) y MPE/sMBR (minimum phone error / state level minimum Bayes risk) son ejemplos de criterios basados en secuencias [45]. Pocas tareas se han enfocado en explorar criterios de entrenamiento a nivel de trama, considerada como una base subyacente en el RAV basado en RNP; de hecho el entrenamiento a nivel secuencia requiere una RNP bien entrenada basándose en un criterio a nivel de trama [47, 132].

Aprovechamos el presente capítulo para resaltar nuestra principal contribución, la cual hace referencia primordialmente en presentar dos nuevas variaciones de la función de costo a nivel de trama, que serán minimizadas en fase de entrenamiento de una RNP con el propósito de mejorar las tasas de error por palabras en un sistema de RAV. El primer enfoque propuesto se basa en el concepto de *extropía* [133], una función dual complementaria de la entropía. Así, obteniendo la extropía de la salida de la red neuronal con respecto al conjunto de datos de entrenamiento, se puede calcular un tipo de entropía cruzada transformada (algo parecido a la entropía par de Shannon [134, 135] o el error de reconstrucción [136–138]). La medida de extropía es restada a la entropía cruzada, transformándola en una variación de sí misma, denominada aquí como *entropía cruzada mapeada no uniforme*, debido a que su nuevo comportamiento es no uniforme; es decir, probabilidades objetivo similares obtenidas de la salida de la red neuronal pueden poseer diferente extropía (extraída de los estados competidores del senón objetivo) aunque tengan el mismo valor a posteriori y por consecuencia alcanzando una CE ajustada. Este esquema intenta eliminar la ambigüedad de las tramas difíciles, tratando de hacer más específica su pertenencia a un senón. El mapeo permite ofrecer un trato particular o específico a cada senón basándose en sus estados competidores, consiguiendo un modelo adaptativo robusto. El segundo método propuesto aplica una fusión a este enfoque mapeado con el método de CE enfatizada o impulsada ideado por Huang et al. [47], quienes han experimentado con aprendizaje a partir de problemas difíciles. En este caso, la nueva función de pérdida formulada enfatiza las tramas difíciles con probabilidades a posteriori objetivo bajas y a la vez desenfatiza la importancia de aquellas tramas con una alta predicción de la red neuronal.

En la sección 4.1 se indican las principales razones para formular la propuesta de este capítulo. En la sección 4.2 se describe la idea de la extropía y cómo puede ser esta utilizada como complemento en el cálculo de la divergencia en la red neuronal. En la sección 4.3 se describe el mecanismo de adaptación de la función de entropía cruzada mapeada. En la sección 4.4 se indican las principales aportaciones de la fusión de la función de costo propuesta con respecto a la entropía cruzada convencional y la impulsada.

4.1. Motivación

Cuando las redes neuronales son aplicadas al reconocimiento de voz, estas pueden calcular las probabilidades de emisión de estado en un enfoque de RNP-MOM, en lugar de emplear mezclas Gaussianas en una arquitectura de MMG-MOM. Las investigaciones en años recientes han mostrado mejoras con las redes neuronales en este campo. Sin embargo, aún existe un camino que transitar en el RAV, y particularmente, el uso de redes neuronales profundas aún puede ser mejorado en varias formas. Sabemos que la salida de una red neuronal proporciona la probabilidad de que una trama de entrada o_t (vector de observaciones acústicas) pertenezca a un estado de MOM específico q_j (estado de tri-fono ligado o senón) en el tiempo t, y que esta probabilidad es subsecuentemente empleada para calcularla la correspondiente probabilidad de emisión de estado $p(o_t|q_j) \triangleq b_j$, la cual es requerida por el modelo oculto de Markov. Motivados por este hecho, las tramas de entrada necesitan ser clasificadas tan preciso como sea posible, pero algunas de ellas poseen una alta probabilidad de pertenencia a varios estados, manifestando ambigüedad en su clasificación. Esta situación se debe a que esas probabilidades son muy cercanas la una a la otra. Esto implica que existe más de un estado compitiendo fuertemente con el estado deseado (el estado correcto, denominado senón objetivo). La intención es pues eliminar tanto como sea posible esta ambigüedad; para alcanzar esto, podemos utilizar la información que nos proporcionan dichos estados competidores. La medida de extropía, una función dual complementaria de la entropía, puede ayudarnos en estas circunstancias.

4.2. Concepto de extropía

Si una cantidad observable X es considerada con posibles valores contenidos en el rango $R(X) = \{x_1, x_2, \ldots, x_N\}$, el vector $p_N = \{p_1, p_2, \ldots, p_N\}$ está compuesto de valores de funciones de masa de probabilidad (*pmf*) evaluados para X sobre la partición de eventos $[(X = x_1), (X = x_2), \ldots, (X = x_N)]$. Recordando un poco, la entropía de Shannon (valor de incertidumbre en una partición de eventos) en X o en p_N está determinada por

$$H(X) = H(p_N) = -\sum_{i=1}^{N} p_i \log(p_i). \tag{4.2.1}$$

Esta entropía puede tener una función dual complementaria llamada *extropía* [133], entonces la extropía en X o en p_N equivale a

$$H_c(X) = H_c(p_N) = -\sum_{i=1}^{N} (1 - p_i) \log(1 - p_i), \tag{4.2.2}$$

y puede ser considerada como una medida *exterior* a la observación X: la medida exterior de todas las posibilidades de no ocurrencia es complementaria a la entropía (medida interior) de la única posibilidad de ocurrencia. En otras palabras, la *pmf* p_N tiene una función de masa complementaria $q_N = (N-1)^{-1}(1_N - p_N)$, una distribución de improbabilidad (opuesta a p_N) de los valores posibles de X, y su correspondiente medida de incertidumbre puede ser calculada usando la función de extropía. La complementariedad de su relación está basada en la generalización de la idea de un evento complementario a una cantidad complementaria.

4.3. Entropía cruzada mapeada no uniforme

La entropía cruzada es utilizada para medir la incertidumbre de una partición de eventos relacionada a una diferente, la contraparte de la entropía de Shannon (la incertidumbre de una sola partición de eventos). Tomando en cuenta su

relación, las propiedades de la entropía pueden ser empleadas en la función objetivo de la entropía cruzada. Puede pensarse que cada probabilidad a posteriori de un senón (una *pmf*, $y_t^L \mathrel{\hat{=}} p_N$), generada a partir de la salida de la red neuronal (capa softmax), tiene su función de masa complementaria (q_N); por tanto, la medida de extropía de la salida de la red puede también ser calculada. Entonces, en cierta forma, la función de costo contempla una medida logarítmica total esperada de la divergencia entre la salida deseada y la probabilidad predicha por la red neuronal.

Retomando estas ideas, y con el fin de definir una variación de la función de entropía cruzada, un tipo de CE transformada puede ser formulada, algo parecido a la medida de divergencia (para una sola partición de eventos) llamada entropía par de Shannon [134, 135] o el error de reconstrucción [136–138] (usado en los autoencoders). Así, restando la medida de extropía (calculada a partir de la salida de la red neuronal con respecto al conjunto de datos de entrenamiento) de la función de CE, puede ser formulada la *entropía cruzada mapeada no uniforme* como sigue

$$\begin{aligned} J_{CE}^{mapped} = &- \sum_{t=1}^{T_{corpus}} \sum_{q=1}^{C_{classes}} d_t(q) \, \log \, y_t^L(q) \\ &+ \sum_{t=1}^{T_{corpus}} \sum_{q=1}^{C_{classes}} (1 - d_t(q)) \, \log \, (1 - y_t^L(q)). \end{aligned} \tag{4.3.1}$$

Este método lleva a cabo una mapeo de la entropía cruzada (medida interior) con su correspondiente valor de extropía (una medida exterior: la extropía cruzada) con el propósito de ajustar los valores de la función de costo original. En este método las entropías cruzadas de los senones objetivo con probabilidad similar sufren una transformación de sus valores. Esto implica que cada probabilidad a posteriori objetivo, con valor similar generado a partir de la función softmax en un esquema de red neuronal profunda (en tareas de RAV), puede tener una medida de extropía diferente debido a la naturaleza de los valores logarítmicos acumulados para el complementario dual de $y_t^L(s)$. El presente enfoque pretende eliminar la ambigüedad de las tramas difíciles, intentando hacer más específica

su pertenencia a un senón, permitiendo ofrecer un trato particular o específico a cada tipo de senón objetivo basado en sus estados competidores. Un senón con una alta extropía tiene menos estados competidores (la pertenencia de una trama a un senón en particular es más pronunciada); en consecuencia, un senón con una extropía baja tiene más estados competidores, es decir, la pertenencia de una trama a un senón en particular es más ambigua.

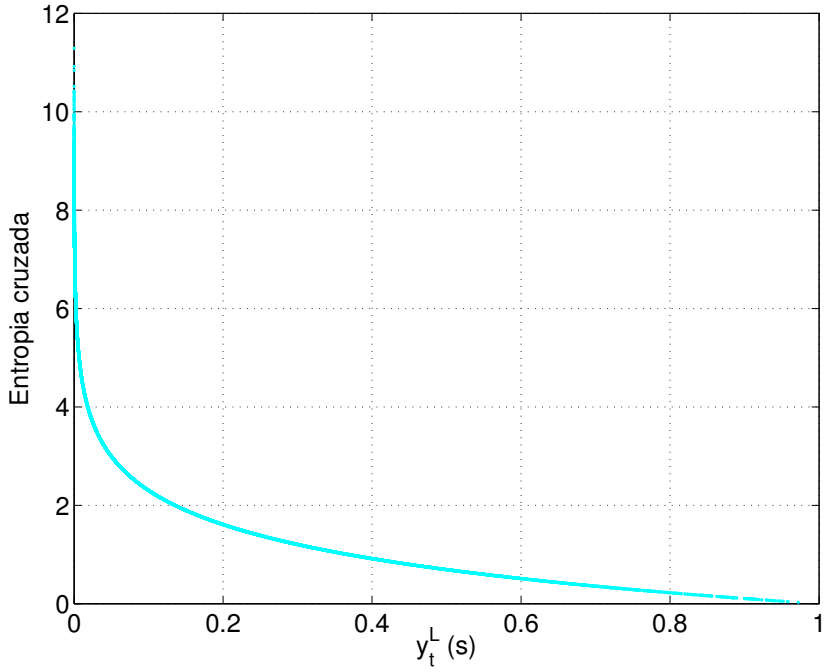

Figura 4.1: Forma de la función de entropía cruzada con respecto a las probabilidades a posteriori objetivo $y_t^L(s)$ para las muestras de la tarea de RAV

Por ejemplo, la Figura 4.1 muestra la medida de entropía cruzada con respecto a la salida de la red (senón objetivo) de un conjunto muestra de 25.6k tramas provenientes de señales de voz (la entrada de un reconocedor de voz). Puede ser inferido que probabilidades similares de la red tienen valores similares en la entropía cruzada, y obviamente mientras más alta la probabilidad a posteriori objetivo, más bajo el valor de la entropía cruzada. La Figura 4.2 muestra el valor de extropía (las mismas 25.6k tramas) con respecto a $y_t^L(s)$, donde las probabili-

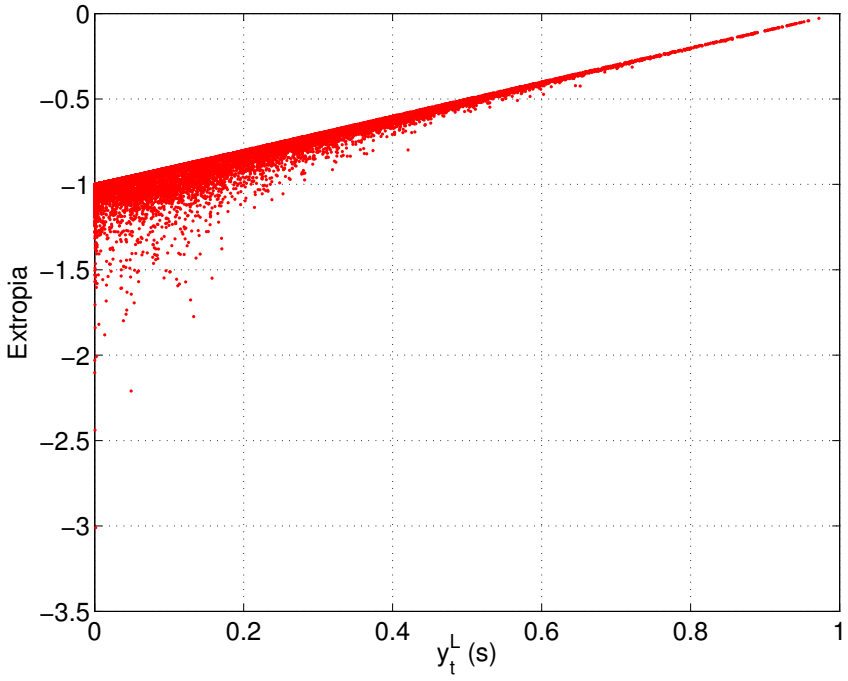

Figura 4.2: Forma del factor de mapeo de extropía con respecto a las probabilidades a posteriori objetivo $y_t^L(s)$ para las muestras de la tarea de RAV

dades similares pueden tener diferentes valores en la extropía; además, mientras más alta es la probabilidad, más baja es la extropía. Los senones objetivo con una probabilidad alta tienen menor extropía (valor absoluto) debido a que ellos tienen senones competidores con probabilidades muy bajas (no hay ambigüedad ciertamente hablando). Además, los senones objetivo con probabilidades similares bajas poseén una extropía variada ocasionada por dos motivos: i) existen varios senones competidores (extropía baja y por lo tanto una ambigüedad más pronunciada de las tramas) y ii) existen pocos estados competidores (extropía alta y por consiguiente una ambigüedad menos pronunciada).

De esta forma, la función objetivo final mapeada (ver Figura 4.3) tiene valores no uniformes en una entropía cruzada para probabilidades a posteriori objetivo similares, enfatizando aquellas tramas (donde el valor de extropía es más bajo y

su correspondiente nuevo valor de entropía cruzada es más alto con respecto a otros similares) con más incertidumbre debido al número de estados competidores; y desenfatizando aquellas tramas con más certidumbre (con menos estados competidores), incluso si sus senones objetivo tienen la misma probabilidad que otros.

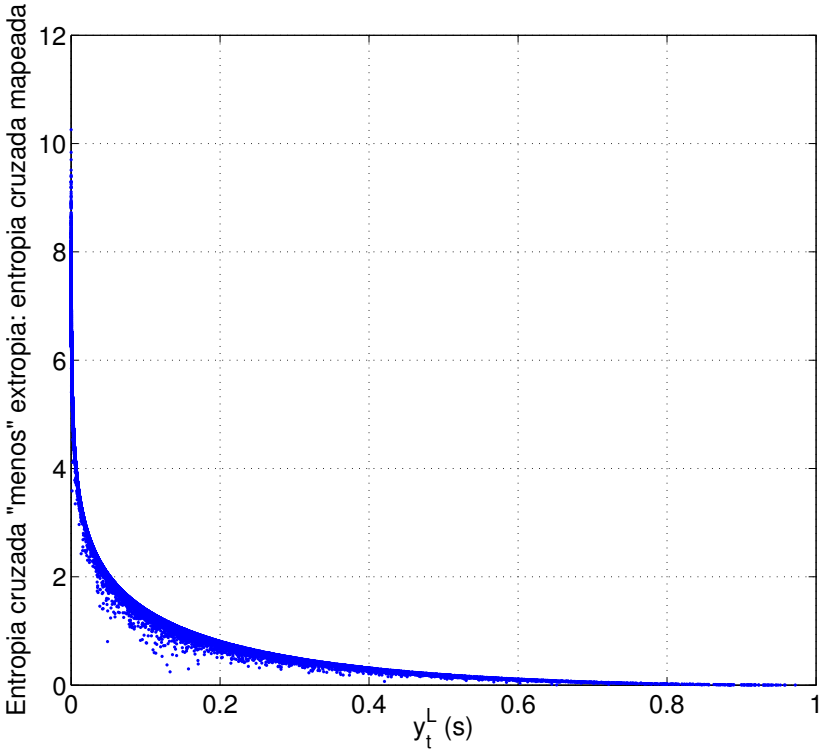

Figura 4.3: Forma de la función objetivo de entropía cruzada mapeada con respecto a las probabilidades a posteriori objetivo $y_t^L(s)$ para las muestras de la tarea de RAV

La Tabla 4.1 muestra ejemplos sencillos como supuestos en referencia a cómo se calculan tanto la entropía cruzada como la extropía; de igual forma se indica cómo existe una relación entre ambas a través de sus valores. Los ejemplos presentados son designados con la idea hipotética simple de que existen solo 3 posibles salidas en la red neuronal (3 estados de MOM). El "primer caso" tiene un supuesto de que el senón objetivo poseé una salida de la red de 0.8 (una alta

Tabla 4.1: Ejemplos de valores para la entropía cruzada (CE) y la extropía cruzada (CE_x) con respecto a las Figuras 4.2 y 4.3

$y_t^L(q)$	$\log y_t^L(q)$	$d_t(q)$	$1 - y_t^L(q)$	$\log 1 - y_t^L(q)$	$1 - d_t(q)$	CE	CE_x	CE - CE_x
Primer caso								
0.80	-0.0960	1	0.20	-0.6989	0			
0.10	-1	0	0.90	-0.4500	1			
0.10	-1	0	0.90	-0.4500	1			
						0.0960	-0.0900	0.0060
Segundo caso								
0.40	-0.3979	1	0.60	-0.2218	0			
0.30	-0.5228	0	0.70	-0.1549	1			
0.30	-0.5228	0	0.70	-0.1549	1			
						0.3979	-0.3098	0.0881
Tercer caso								
0.40	-0.3979	1	0.60	-0.2218	0			
0.39	-0.4089	0	0.61	-0.2146	1			
0.21	-0.6777	0	0.79	-0.1023	1			
						0.3979	-0.3169	0.0810

probabilidad de que la trama de entrada pertenezca a este estado), y existen 2 estados con una probabilidad de 0.1 de que la trama de entrada pertenezca a ellos. Como se puede apreciar, el valor de la entropía cruzada (0.096) menos el valor de la extropía cruzada ($-[-0.09]$) genera una entropía cruzada mapeada de 0.006, un valor relativamente bajo, indicando que el error (CE) de la clasificación no es pronunciado. Por otro lado, el "segundo caso" muestra una probabilidad a posteriori objetivo de 0.4 y una probabilidad a posteriori de 0.3 para los dos estados competidores. En este caso, el valor de entropía cruzada equivale a 0.3979 y el valor correspondiente de extropía es -0.3098, resultando en un error mapeado a ser retro-propagado de 0.0881. En el segundo caso, las probabilidades de los 2 estados competidores son muy cercanas al senón objetivo, sugiriendo que la ambigüedad de la clasificación de la trama es alta; en contraparte con el "tercer caso", donde la misma probabilidad a posteriori objetivo de 0.4 es obtenida por la

red neuronal, pero los dos estados competidores ahora tienen probabilidades de 0.39 y 0.21, respectivamente. En este caso, el valor de CE (0.3979) menos el valor de CE$_x$ (-0.3169) genera un error de 0.081, un error ligeramente mas pequeño que el del segundo caso, sugiriendo que la ambigüedad de la clasificación de la trama es menos pronunciada debido a que el tercer caso tiene un estado competidor (el estado con valor de 0.39) más cercano al senón objetivo (0.4), aunque ambos senones objetivos tengan la misma probabilidad a posteriori. De acuerdo a esto, se puede determinar que el tercer caso tiene solo un estado competidor con respecto a dos estados competidores en el segundo caso.

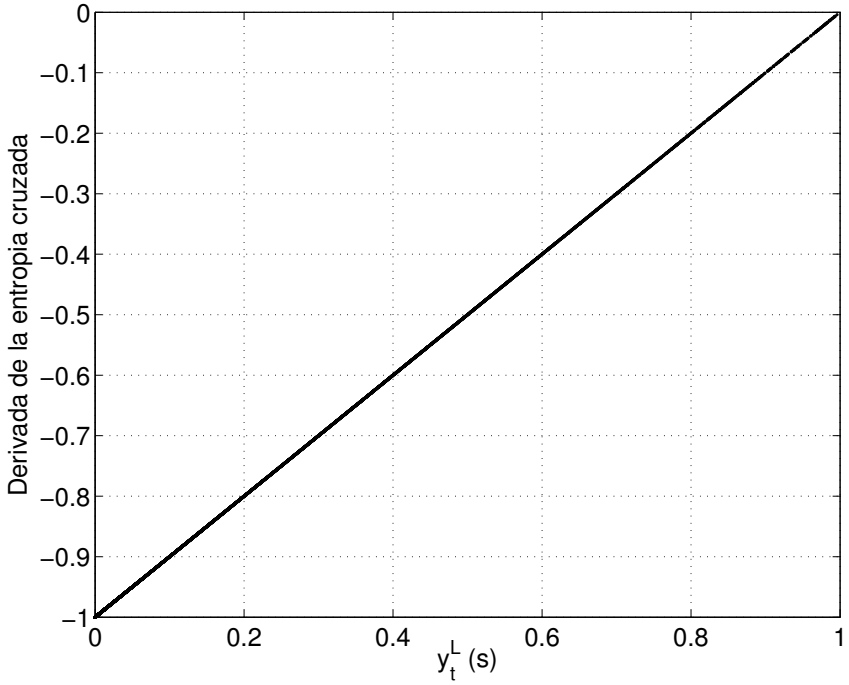

Figura 4.4: Derivada de la función objetivo clásica de entropía cruzada con respecto a las probabilidades a posteriori objetivo $y_t^L(s)$ para las muestras de la tarea de RAV

Tomando la derivada parcial de la ec. (4.3.1) con respecto al vector de excitación z^L de la capa de salida, el vector de error a ser retro-propagado a las capas ocultas previas está dado por:

$$\xi_t^L = \frac{\partial J_{CE}^{mapped}}{z^L} = (y_t^L - d_t) + y_t^L \left(\sum_{q=1}^{C} y_t^L(q) \frac{1 - d_t(q)}{1 - y_t^L(q)} - \frac{1 - d_t}{1 - y_t^L} \right), \tag{4.3.2}$$

donde el segundo término del lado derecho de la ecuación tiene el efecto de un factor de mapeo sobre la derivada de la entropía cruzada clásica, $\frac{\partial J_{NLL}}{z^L} = y_t^L - d_t$ (ver ec. (3.3.13) y Figura 4.4), causando que la derivada pueda ser convertida en la que se muestra en la Figura 4.5, en la cual el efecto de variaciones no uniformes de la extropía se puede percibir. Se puede notar que el error ya no decrece linealmente (a medida que la probabilidad $y_t^L(s)$ se vuelve más imprecisa) y en una forma uniforme, es decir, algunas tramas con probabilidades objetivo similares tienen un error más bajo a ser retro-propagado (aquellas con menos ambigüedad).

4.4. Mejoras en la entropía cruzada mapeada no uniforme a través del énfasis en probabilidades objetivo bajas

La entropía cruzada mapeada propuesta intenta eliminar la ambigüedad de las tramas difíciles, tratando de enfocarse en las tramas con probabilidades a posteriori distribuidas entre varios senones. Varias tramas pueden ser desenfatizadas dada su pequeña ambigüedad de predicción, sin embargo, algunas de ellas pueden ser interpretadas como senones erróneos. Esto puede pasar cuando la trama tiene una probabilidad alta pero en un senón equivocado, y por tanto el senón objetivo tiene una probabilidad a posteriori baja dada esa trama.

Con el propósito de lidiar con esto, la función objetivo de entropía cruzada mapeada puede ser fusionada con la CE impulsada ideada por Huang et al. [47], quienes han experimentado con aprendizaje en una RNP tomando en cuenta la idea del aprendizaje a partir de problemas difíciles: enfatizando las tramas difíciles con probabilidades a posteriori objetivo bajas y desenfatizando la importancia

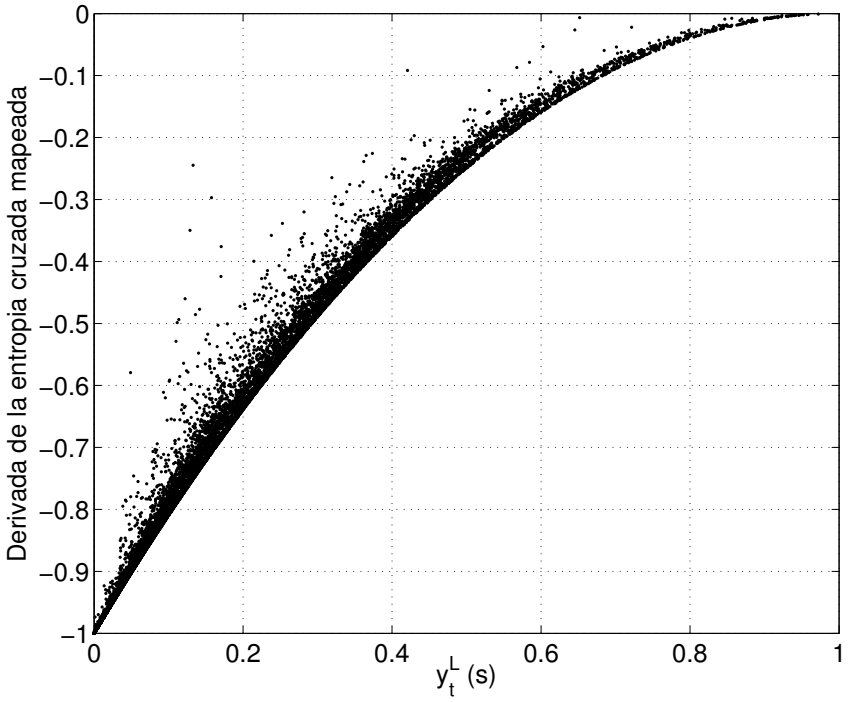

Figura 4.5: Forma de la derivada de la función objetivo de entropía cruzada mapeada con respecto a las probabilidades a posteriori objetivo $y_t^L(s)$ para las muestras de la tarea de RAV

de aquellas tramas con una alta predicción de la RNP. La idea es aprender más de tramas difíciles, las cuales son propensas a ser erróneamente predecidas (probabilidades a posteriori bajas $y_t^L(s)$). Esta *función objetivo impulsada* se define como sigue

$$J_{CE}^{boosted} = -\sum_{t=1}^{T_{corpus}} (1 - y_t^L(s))^{\alpha} \log y_t^L(s), \tag{4.4.1}$$

donde un término de ponderación ha sido agregado a la entropía cruzada con el fin de impulsar las tramas difíciles, así las tramas con una $y_t^L(s)$ baja son enfatizadas; y α denota el orden del impulso. Su correspondiente derivada con

respecto al vector de excitación de la capa de salida está dado por

$$\xi_t^L = \frac{\partial J_{CE}^{boosted}}{z^L} = f_i \ (y_t^L - d_t), \tag{4.4.2}$$

donde f_i equivale a

$$f_i = (1 - y_t^L(s))^{\alpha-1}(1 - y_t^L(s) - \alpha y_t^L(s) \log y_t^L(s)), \tag{4.4.3}$$

donde el factor de importancia f_i impulsa el segundo término $y_t^L - d_t$. El factor de importancia se aproxima a 0 cuando $y_t^L(s)$ se acerca a 1 (el factor de importancia de las tramas correspondientes cae cuando la exactitud en la predicción crece y viceversa), poniendo más atención a tramas difíciles (aquellas con una probabilidad a posteriori objetivo baja).

A partir de estas aseveraciones, la fusión entre la entropía cruzada mapeada y la impulsada se formula como sigue:

$$\begin{aligned} J_{CE}^{bm} = &- \sum_{t=1}^{T}\sum_{q=1}^{C} d_t(q)(1 - y_t^L(q))^{\alpha} \log y_t^L(q) \\ &+ \sum_{t=1}^{T}\sum_{q=1}^{C}(1 - d_t(q)) \ (y_t^L(q))^{\alpha} \log (1 - y_t^L(q)), \end{aligned} \tag{4.4.4}$$

donde los términos de ponderación complementarios fueron agregados tanto a la entropía cruzada como a la extropía con respecto a la ec. (4.3.1). Su correspondiente derivada con respecto al vector de excitación de la capa de salida está dada por:

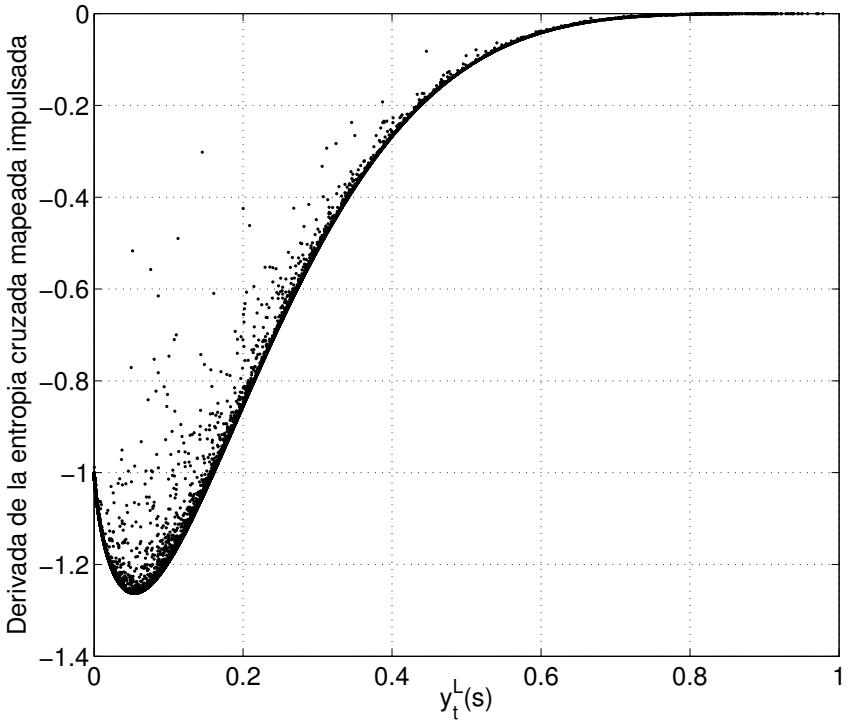

Figura 4.6: Forma de la derivada de la función objetivo de entropía cruzada mapeada impulsada (orden de impulso $\alpha = 4$) con respecto a las probabilidades a posteriori objetivo $y_t^L(s)$ para las muestras de la tarea de RAV

$$\begin{aligned}\xi_t^L = \frac{\partial J_{CE}^{bm}}{z^L} =& (1-y_t^L(s))^{\alpha-1}(1-y_t^L(s)-\alpha y_t^L(s)\,\log\, y_t^L(s))(y_t^L-d_t)\\ &-(y_t^L)^{\alpha}(1-d_t)\left(\frac{y_t^L}{1-y_t^L}-\alpha\,\log\,(1-y_t^L)\right)\\ &+y_t^L\sum_{q=1}^{C}(y_t^L(q))^{\alpha+1}\frac{1-d_t(q)}{1-y_t^L(q)}\\ &-y_t^L\sum_{q=1}^{C}\alpha(y_t^L(q))^{\alpha}(1-d_t(q))\,\log\,(1-y_t^L(q)).\end{aligned} \tag{4.4.5}$$

La Figura 4.6 muestra el efecto de los *factores de importancia* en la entropía

cruzada mapeada impulsada a través de la derivada del modelo de la ec. (4.4.5). Como se puede apreciar, las probabilidades a posteriori objetivo bajas son enfatizadas (tramas difíciles), proyectando más error del que realmente es, mientras que se puede notar el impacto del valor de la extropía sobre las probabilidades similares (representaciones no uniformes de cada una), proporcionando un poder adaptativo y un modelo robusto.

Capítulo 5

Experimentación y resultados

En este capítulo se mencionan los lineamientos metodológicos y de materiales utilizados en el trabajo de investigación, incluyendo el corpus de voces, el software utilizado y los procedimientos operativos para el procesado de las fuentes de datos, para finalmente alcanzar las tasas de reconocimiento de voz en cada tarea realizada.

En la sección 5.1 se mencionan los elementos intervinientes en el procedimiento de obtención y uso del corpus de voces empleado en el presente trabajo de investigación. En la sección 5.2 se muestra la parte de la interfaz gráfica de usuario desarrollada con el fin de facilitar el uso y configuración general de las llamadas a procedimientos implicados en el proceso de RAV a través de Kaldi (una herramienta que permite manipulación y reconocimiento de voz usando diferentes esquemas). Finalmente, en la sección 5.3 se muestran las configuraciones y detalles de los casos de estudio elaborados con el fin de realizar las pruebas y experimentos con las arquitecturas dentro de las tareas de RAV.

5.1. Preparación de los datos

El corpus de voces es el elemento base para el seguimiento del proceso de reconocimiento automático de voz, para lo cual, en este apartado se menciona cómo se obtuvo y qué tratamiento se le dio.

5.1.1. Definición de la gramática del reconocedor y su contexto

El proceso de creación de modelo de lenguaje o gramática depende de si se usará un modelo de lenguaje estadístico o algún tipo de gramática. El caso de estudio de reconocimiento de voz se ha realizado utilizando un ambiente de marcado telefónico (cadenas de dígitos y listas de nombres personales) con un corpus de voces personalizado de palabras conectadas en Español, dependiente del texto (vocabulario), independiente de locutor y de tamaño mediano. Se definió una gramática libre de contexto (GLC) siguiendo la notación BNF-extendida (Backus-Naur Form) [139], la cual se guardó en el archivo `gram` (ver Apéndice A.1).

En la Figura 5.1 se puede observar el grafo de la gramática de reconocimiento definida en el trabajo, donde cada vértice representa una palabra y las aristas representan las transiciones permitidas entre ellas.

Por lo que el resultado de la gramática libre de contexto especificada es una secuencia de sentencias como las siguientes:

- LLAMAR ALAN SANMIGUEL MEDINA
- MARCAR LUCIANO SALMON CABAÑAS
- TELEFONO CUATRO UNO SIETE SIETE NUEVE OCHO DOS OCHO OCHO OCHO
- MARCAR HERNAN COBOS PANIAGUA
- TELEFONO OCHO CUATRO UNO CUATRO TRES CINCO CUATRO OCHO SEIS UNO
- MARCAR JUAN GRANADA MADRID
- ...

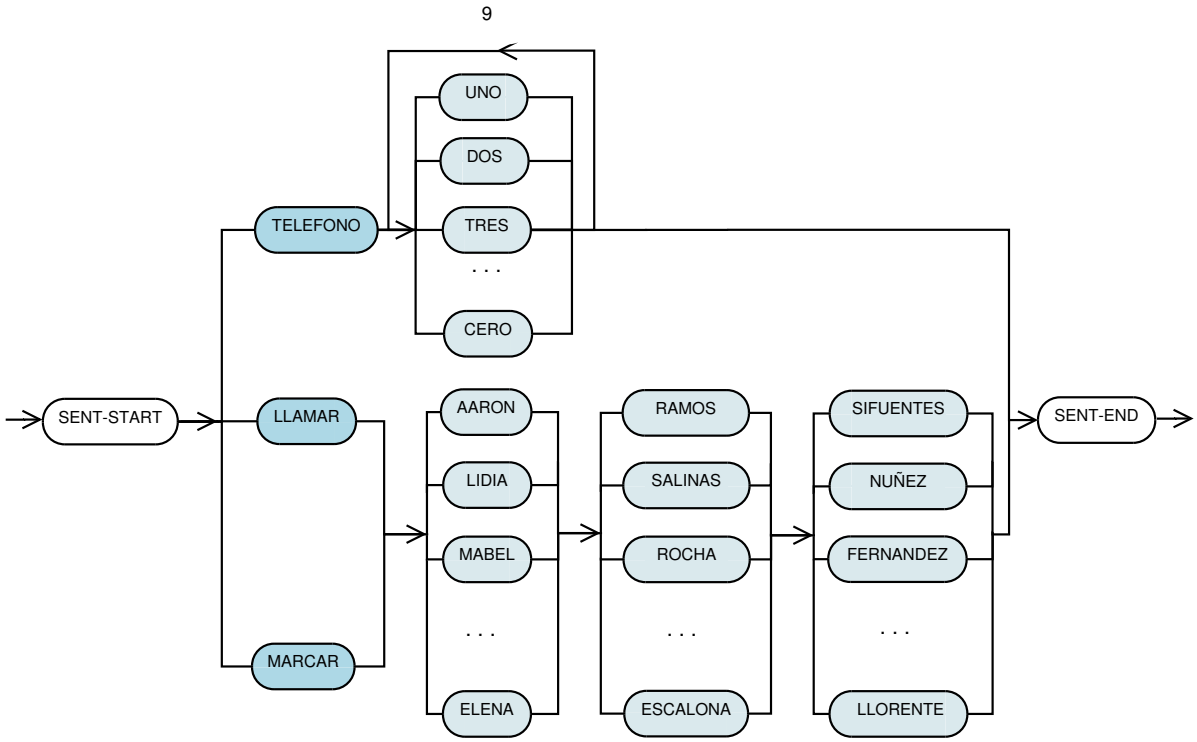

Figura 5.1: Grafo de palabras correspondiente a la gramática definida

- TELEFONO DOS SIETE SEIS UNO SIETE CERO CUATRO SIETE OCHO UNO
- LLAMAR BRUNO MENDIZABAL PEÑALVER
- TELEFONO TRES NUEVE UNO SEIS SEIS CINCO UNO OCHO OCHO NUEVE
- LLAMAR RAFAEL OLIVA NIEVES

Esta gramática (modelo de lenguaje) ha sido creada iniciando con la notación BNF extendida (con cadenas de dígitos de extensión 10, así como secuencias de nombres completos con dos apellidos). Empleando el Toolkit de HTK [140], la GLC es convertida a una red de palabras por medio de una notación de bajo nivel llamada *standard lattice format* (SLF) en HTK, en la cual cada instancia de

palabra y cada transición palabra a palabra es explícitamente listada (los datos son guardados en un archivo denominado `wdnet`, a través del uso del comando `HParse gram wdnet`):

VERSION=1.0
N=1413 L=2796
I=0 W=<eps>
I=1 W=<eps>
I=2 W=TELEFONO
I=3 W=UNO
I=4 W=<eps>
I=5 W=DOS
I=6 W=TRES
I=7 W=CUATRO
I=8 W=CINCO
I=9 W=SEIS
I=10 W=SIETE
I=11 W=OCHO
I=12 W=NUEVE
I=13 W=CERO
I=14 W=UNO
I=15 W=<eps>
... ...
I=138 W=ROSALINDA
I=139 W=JESUS
I=140 W=JUAN
I=141 W=JOSE
I=142 W=FRANCISCO
I=143 W=GERARDO I=144 W=DANIEL
I=145 W=SAMUEL
I=146 W=SARA

I=147 W=LUCERO
I=148 W=ALEJANDRO
I=149 W=ESMERALDA
I=150 W=SERGIO
... ...
J=0 S=103 E=1
J=1 S=0 E=2
J=2 S=2 E=3
J=3 S=3 E=4
J=4 S=5 E=4
J=5 S=6 E=4
J=6 S=7 E=4
J=7 S=8 E=4
J=8 S=9 E=4
... ...

en donde I representa el número de nodo, W la palabra en cuestión, J el salto del nodo S al nodo E; y $< eps >$ representa una cadena vacía.

Además, este formato HTK de red de palabras tuvo que ser transformado (por medio de un script en perl que toma como entrada el archivo `wdnet`, quedando el resultado guardado en el archivo `.../data/lang/G.fst`, dentro de la carpeta de Kaldi) a un archivo transductor (en formato de texto AT&T) para posteriormente ser usado en el Toolkit de Kaldi [141] (ver Apéndice A.2).

Esta notación está creada basándose en la gramática del formato OpenFST (finite state transducer) [142], la cual es usada en Kaldi como base para crear el grafo o transductor. Para ejemplificar esto, tómese el transductor de la Figura 5.2. El estado inicial es la etiqueta 0. Puede haber solo un estado inicial. El estado final es 2 con el peso final de 3.5. Cualquier estado con un peso final distinto de infinito es un estado final. Existe un arco (o transición) del estado 0 al estado 1 con la etiqueta de entrada a, la etiqueta de salida x, y un peso de 0.5. Este FST trasduce, por ejemplo, cadenas como ac a xz con peso de 6.5 (la suma de los pesos de los arcos y el peso final).

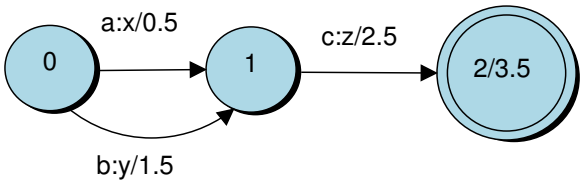

Figura 5.2: Transductor de ejemplo en OpenFST

El transductor FST para el RAV del presente trabajo queda formado como se muestra en la Figura 5.3.

5.1.2. Definición del diccionario de pronunciación (lexicon)

Como se mencionó anteriormente, el diccionario de pronunciación indica el desglose de los componentes fonéticos de cada palabra a utilizar en el contexto del diccionario. Siguiendo las notaciones del IPA (alfabeto fonético internacional), para la tarea de reconocimiento del presente trabajo se tiene el diccionario mostrado en el Apéndice A.3 (`.../data/local/dict/lexicon.txt`). Cada palabra del diccionario es convertida a su significado fonético siguiendo los fonos (datos guardados en el archivo `.../data/local/dict/nosilence_phones.txt` para los sonidos sonoros, y en `.../data/local/dict/silence_phones.txt` para los no sonoros) listados en la Tabla 5.1.

5.1.3. Adquisición o grabación del corpus de voces para entrenamiento y para pruebas

Las grabaciones de los audios son realizadas a partir de secuencias aleatorias (`HSGen -l -n 4000 wdnet lexicon.txt >datasetprompts`, para generar por ejemplo 4000 sentencias aleatorias) usando HTK, que usa de base la gramática y el diccionario de pronunciación. Del total del conjunto de datos de entrenamiento original propuesto se descartaron algunas grabaciones, ya que no contaban con lineamientos elementales, por ejemplo, no disponían de la frecuencia de muestreo correcta, y/o la calidad del audio generado era demasiado baja.

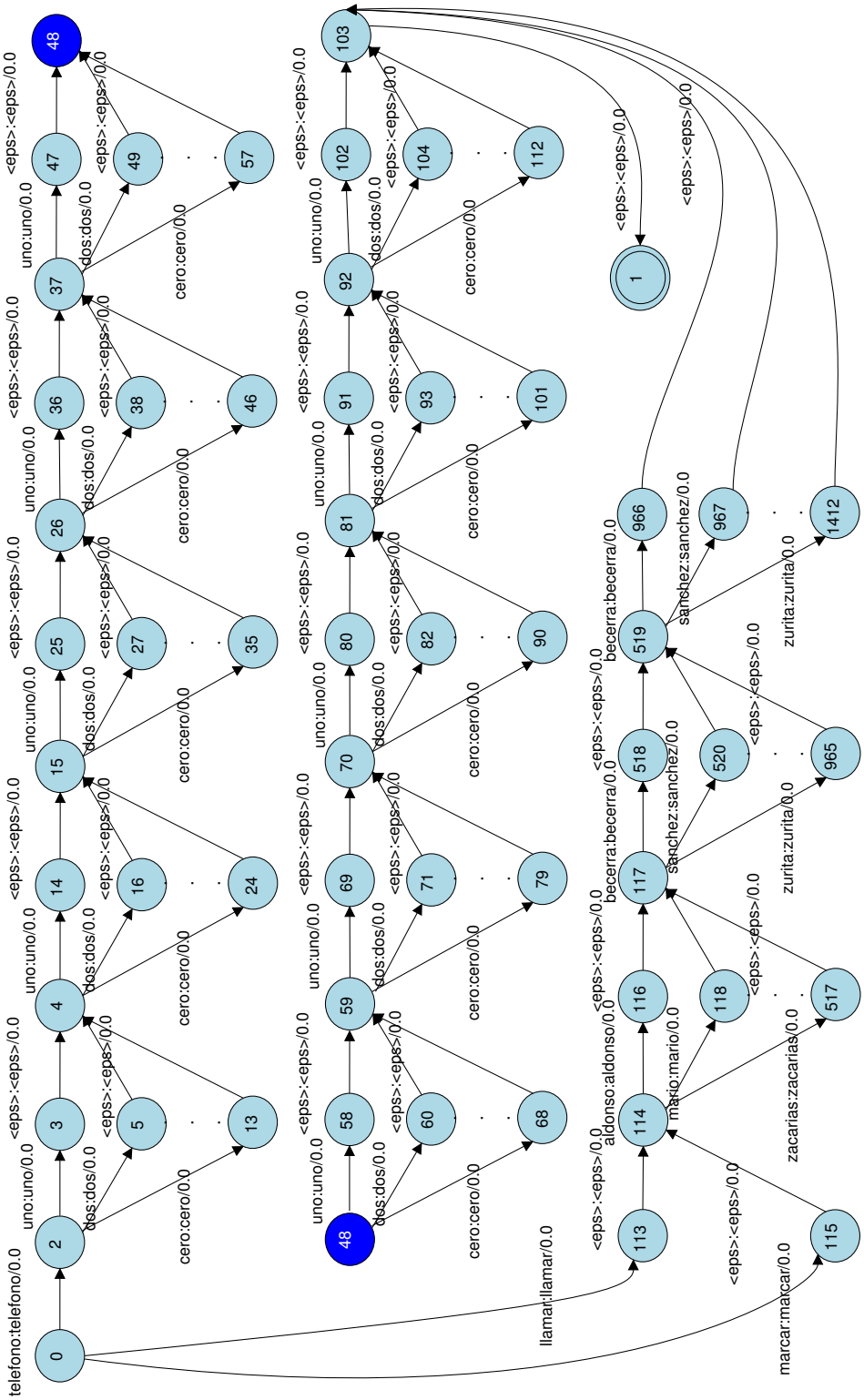

Figura 5.3: Transductor (en formato OpenFST) de la gramática del presente trabajo de investigación

Tabla 5.1: Representación de fonos para la tarea de reconocimiento de voz en español latino

Sonido (fono)	**Representación adaptada del IPA**
a	a
l	l
d	d
o	o
n	n
s, c	s
e	e
j	x
r (ere)	r
f	f
b, v	b
i	i
t	t
r (erre)	rr
k, c, q	k
m	m
ll	dZ
u	u
p	p
g	g
ñ	jn
ch	tS

El corpus de voces completo para el presente trabajo de investigación consta de una mezcla de voces humanas y elocuciones de aplicaciones en línea de texto a voz). Las aplicaciones en línea utilizadas fueron ispeech, oddcast (SitePal) y vocalware. Muchos de los archivos de audio grabados tienen varios tipos de ruido con la intención de hacer más robusto el sistema: distorsión del micrófono, sonido de lluvia, automóviles, animales, volumen alto y bajo, y en general el ruido ambiental ordinario. La edad de los participantes humanos oscila entre 18 y 26 años. Todos los audios fueron grabados usando una codificación PCM, con 16 bits por muestra, una tasa de muestreo de 16000 Hz con un solo canal.

El programa *Wavesurfer* 1.8.8 (ver Figura 5.4) fue empleado para la realización de estas grabaciones en formato wav, utilizando los micrófonos que las computadoras portátiles de los usuarios traen integrados, los cuales varían en todas las marcas.

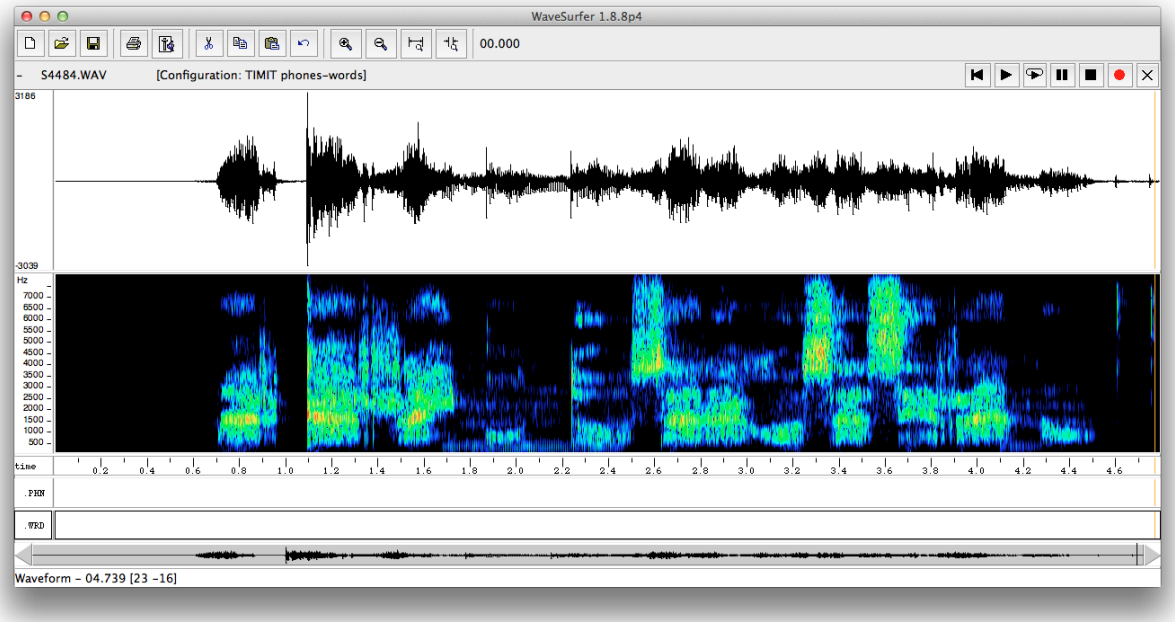

Figura 5.4: Wavesurfer: programa de manipulación de audio

Los archivos resultantes en Kaldi para el manejo de los datos de entrenamiento y pruebas quedan resumidos de la siguiente forma

1. **Entrenamiento**: en la carpeta `.../data/train/` se encuentra guardado el archivo de `text`, que contiene algo como:

   ...
   s2015-00000004_S0000011611 TELEFONO SEIS CINCO CINCO CUATRO NUEVE OCHO OCHO TRES SIETE DOS
   s2015-00000004_S0000011621 LLAMAR SELENE NUÑEZ PADILLA
   s2015-00000004_S0000011631 LLAMAR FEDERICO VILLA VAZQUEZ

s2015-00000004_S0000011641 MARCAR ANTONIA MONTES GARCIA
s2015-00000004_S0000011651 TELEFONO OCHO NUEVE CERO CUATRO CINCO TRES DOS UNO CERO CUATRO
s2015-00000005_S0000010021 MARCAR LISA ROCHA LOPEZ
...
en donde cada renglón está compuesto por un identificador de sentencia y posteriormente la sentencia generada de la gramática.

El archivo `wav.scp` contiene algo como:
...
s2015-00000004_S0000011611 data/train/sig/S0000011611.WAV
s2015-00000004_S0000011621 data/train/sig/S0000011621.WAV
s2015-00000004_S0000011631 data/train/sig/S0000011631.WAV
s2015-00000004_S0000011641 data/train/sig/S0000011641.WAV
s2015-00000004_S0000011651 data/train/sig/S0000011651.WAV
s2015-00000005_S0000010021 data/train/sig/S0000010021.WAV
...
en donde cada renglón lleva su número de identificación de registro y la ruta del archivo de audio.

El archivo `utt2spk` contiene algo como:

...
s2015-00000004_S0000011611 s2015-00000004
s2015-00000004_S0000011621 s2015-00000004
s2015-00000004_S0000011631 s2015-00000004
s2015-00000004_S0000011641 s2015-00000004
s2015-00000004_S0000011651 s2015-00000004
s2015-00000005_S0000010021 s2015-00000005
...

que sirve para diferenciar a un locutor y las sentencias de él (por ejemplo se pueden ver los locutores s2015-00000004 y s2015-00000005).

En el archivo `spk2gender` se puede ver algo como:

...
s2015-00000002 m
s2015-00000003 m
s2015-00000004 m
s2015-00000005 f
s2015-00000006 m
s2015-00000007 m
...

que sirve para denotar el sexo del locutor.

2. **Pruebas**: para las pruebas de la tarea de RAV, en la carpeta `.../data/test/` se encuentra guardado el catálogo de archivos similares a los mostrados en el apartado de entrenamiento.

Además, cabe mencionar que en la carpeta `.../data/train/mfcc`, y en la correspondiente de `test`, se almacenan los archivos de características acústicas MFCC generados con Kaldi. Las transformaciones y adaptaciones finales de las características acústicas quedan almacenadas en `data-fmllr-tri3b` de Kaldi, tanto para entrenamiento como para pruebas.

5.1.4. Etiquetado de señales de voz de entrenamiento

A estas alturas se puede ir integrando el proceso completo de entrenamiento en un sistema de reconocimiento basado en MOM. Ya se han mencionado en el capítulo 2 ciertos aspectos del proceso de entrenamiento, por ejemplo, cómo son calculados los parámetros de un MMG a través del algoritmo de expectation-maximization. También se vio cómo se puede utilizar el proceso de entrenamiento de una red neuronal en el capítulo 3. Sin embargo, queda pendiente cómo son obtenidos datos de entrenamiento en los cuales cada trama es etiquetada con una identidad de fono. Asumiendo un diccionario de pronunciación o léxico, entonces

la estructura del grafo de estados del MOM es pre-especificada (ver Figura 2.11) como la estructura base. En general, los sistemas de reconocimiento de voz no tratan de aprender la estructura del MOM de palabras individuales. Por lo tanto se requiere entrenar la matriz B y las probabilidades de transiciones diferentes a cero (lazos y de siguiente fono) en la matriz A. Todas las otras probabilidades en la matriz A son ajustadas a cero y nunca cambian.

Un posible método de entrenamiento simple es el entrenamiento de palabras aisladas **etiquetadas manualmente** [60], en el cual separamos las matrices B y A para los MOMs para cada palabra basándose en los datos de entrenamiento de un alineamiento manual. Se nos proporciona por ejemplo un corpus de voces de dígitos, donde cada instancia de un dígito pronunciado es almacenada en un archivo ".wav", con el inicio y fin de cada palabra y fono segmentado manualmente. Dada esta base de datos etiquetada manualmente, se pueden calcular las probabilidades de observación Gaussianas B y las probabilidades de transición A contándolas solamente en los datos de entrenamiento. La matriz A es específica para cada palabra, pero B sería compartida por todas las palabras si el mismo fono ocurre en múltiples palabras.

No obstante, los datos de entrenamiento de etiquetas manuales son raramente utilizados en sistemas de reconocimiento de voz continua. Una de las razones es que es muy costoso emplear humanos para etiquetar manualmente los límites fonéticos, podría tomar hasta 400 veces de tiempo real (400 horas de tiempo para etiquetar una hora de señal de voz). Otra razón es que los humanos no etiquetan fonéticamente muy bien para unidades menores que un fono. Los sistemas de reconocimiento no son mejores que los humanos en encontrar límites, pero sus errores son por lo menos consistentes entre los conjuntos de entrenamiento y pruebas.

Por estas razones, los sistemas de RAV entrenan cada MOM de fono embebido en una sentencia completa, y la segmentación y el alineamiento de fono son realizados sistemáticamente como parte del proceso de entrenamiento. El proceso completo de entrenamiento del modelado acústico es llamado **entrenamiento embebido**. En este sentido, por ejemplo, para entrenar un sistema de recono-

cimiento de dígitos se necesitará un corpus de entrenamiento de secuencias de dígitos (y en nuestro caso también de cadenas de nombres completos) hablados por locutores. Asumiendo que el corpus de entrenamiento está separado en archivos individuales ".wav", cada uno conteniendo secuencias de dígitos (o nombres completos de personas) hablados. Para cada archivo *wav* se ocupará saber la secuencia correcta de dígitos o nombres que en él se pronuncian. Por esta razón se asocia a cada archivo *wav* su transcripción (una cadena de palabras). Se ocupa también un diccionario de pronunciación (léxico) y un conjunto de fonos, definiendo entonces un conjunto de MOMs de fonos (sin entrenar). A partir de estas transcripciones, del léxico y los MOMs de fono, se construye un MOM para una "oración completa" para cada sentencia requerida (ver Figura 5.5), el cual está listo para ser alineado y entrenado con respecto a las características cepstrales extraídas a partir de los archivos de audio.

A partir de esta instancia se encuentran las condiciones para entrenar las matrices A y B para el MOM. Lo interesante del paradigma basado en el algoritmo Baum-Welch para el entrenamiento embebido de los MOMs es que con lo que se dispone ya es el total de datos de entrenamiento que se requiere (no se ocupa una transcripción fonética o saber dónde inicia o termina cada palabra). El algoritmo de Baum-Welch acumula todas las posibles segmentaciones de palabras y fonos, usando $\xi_j(t)$, la probabilidad de estar en el estado j en el tiempo t y generar la secuencia de observaciones O. Para realizar este proceso se ocupa una estimación inicial para las probabilidades de observación y de transición, a_{ij} y $b_j(o_t)$. La forma más simple de hacer esto es con una **inicialización plana**. En este mecanismo se ajusta en cero cualquier transición del MOM que queremos que tenga una estructura de cero, tales como las transiciones de estados actuales a estados precedentes. El cálculo de la probabilidad γ en el algoritmo de Baum-Welch incluye el valor previo de a_{ij}, de esta manera los valores en cero no cambiarán. Entonces se hacen equiprobables el resto de las probabilidades (las que no son ceros) de transición de los MOMs. Así las dos transiciones de salida de cada estado (el lazo y la del salto siguiente) deberían tener una probabilidad de 0.5 cada una. Para las Gaussianas, este proceso inicializa la media y la varianza

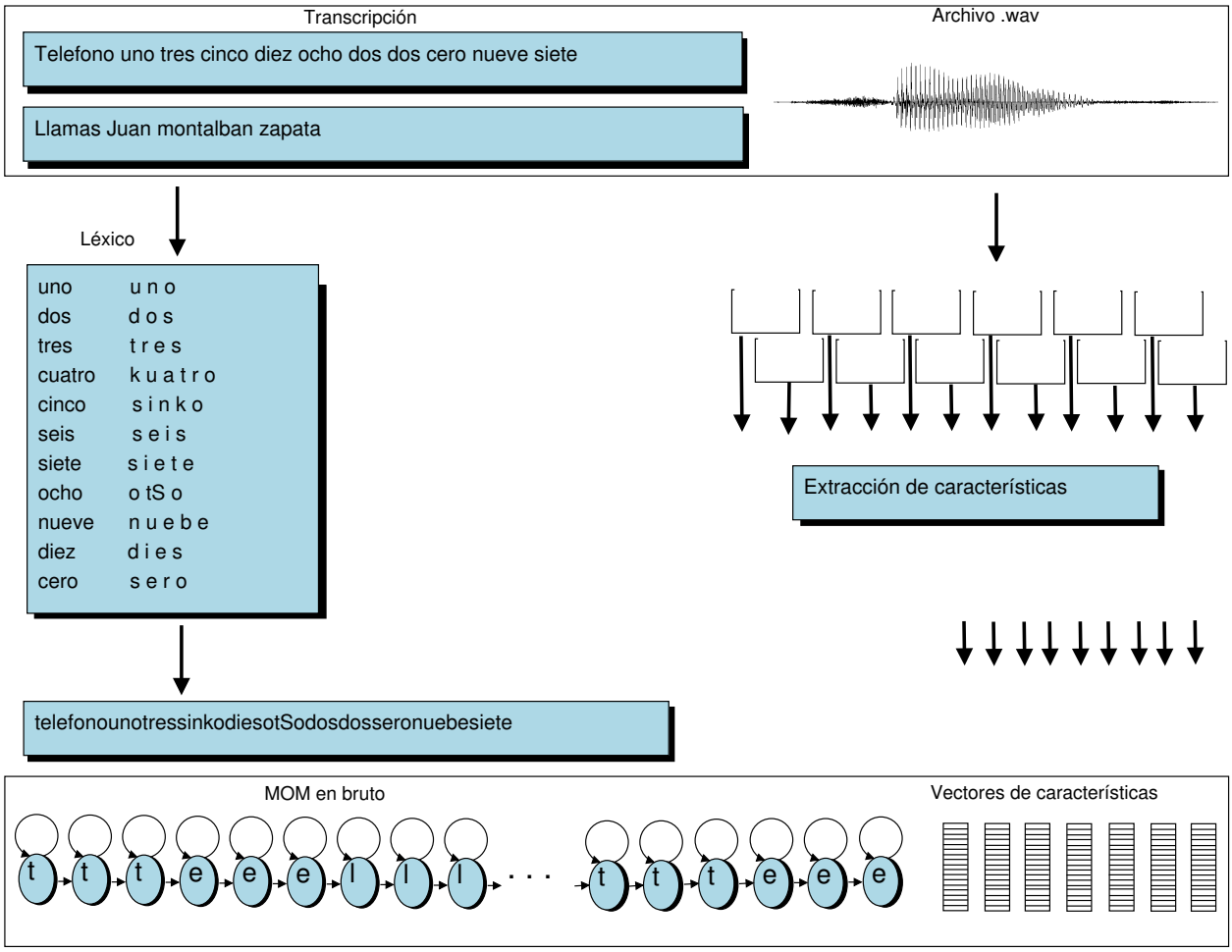

Figura 5.5: Entrada para el algoritmo de entrenamiento embebido

de manera idéntica a la media y varianza global de los datos de entrenamiento completos.

Ya con las estimaciones iniciales de A y B, para un sistema estándar MMG-MOM, se ejecutan varias iteraciones del algoritmo Baum-Welch sobre el conjunto de entrenamiento completo. Cada iteración modifica los parámetros del MOM, y se detiene cuando el sistema converge. Durante cada iteración se calculan las probabilidades *forward* y *backward* para cada oración dadas las probabilidades iniciales de A y B, entonces se re-estiman estas mismas. Entonces se aplican las variaciones del algoritmo de E-M para actualizar la media y varianza para

Gaussianas multivariable.

El algoritmo de Baum-Welch es usado repetidamente como un componente del proceso de entrenamiento embebido. Baum-Welch calcula $\xi_j(t)$, la probabilidad de estar en el estado j en el tiempo t, y se usa el algoritmo *forward-backward* para acumular todas las posibles rutas que estuvieron en el estado j emitiendo el símbolo o_t en el tiempo t. Esto permite acumular la cuenta para re-estimar la probabilidad de emisión $b_j(o_t)$ para todas las rutas que pasaron el estado j en el tiempo t. Sin embargo, el algoritmo de Baum-Welch por sí solo puede consumir mucho tiempo. Entonces se utiliza una aproximación eficiente de él usando el algoritmo de Viterbi. En el cual, si recordamos, solo se contempla la ruta más probable para pasar a través del estado j en el tiempo t. Así, en lugar de ejecutar E-M en cada etapa del algoritmo embebido, se ejecuta repetidamente Viterbi. Ejecutando el algoritmo de Viterbi de esta manera sobre los datos de entrenamiento se suele llamar al proceso como **alineamiento forzado de Viterbi** o **alineamiento forzado**.

En el entrenamiento de Viterbi (a diferencia de la decodificación de Viterbi en el conjunto de prueba) se conoce cuál secuencia de palabras se asignaría a cada secuencia de observaciones. Por ende, se puede forzar al algoritmo a pasar por ciertas palabras, ajustando a_{ij} apropiadamente. Viterbi forzado es una simplificación del algoritmo de decodificación regular de Viterbi, ya que solo tiene que encontrar la secuencia correcta de estados (subfonos), pero no tiene que encontrar la secuencia de palabras. El resultado del proceso es un alineamiento forzado: la mejor ruta de estados correspondiente a una secuencia de observaciones de entrenamiento. Entonces se puede usar este alineamiento de estados de MOM a observaciones para acumular la cuenta y re-estimar los parámetros del MOM. Resulta que el algoritmo forzado de Viterbi también se utiliza en el entrenamiento embebido de modelos híbridos como sistemas de RNP-MOM.

5.1.5. Extracción de características acústicas de las señales de voz de entrenamiento y de prueba

Las observaciones acústicas usadas para la tarea de RAV son las que se manejan por defecto en el recetario de scripts de Kaldi [141], que corresponden a 13 coeficientes MFCC con normalización cepstral de media y varianza (CMVN), los cuales fueron empalmados en el tiempo tomando una ventana de 11 tramas (5 en cada lado de la trama actual), seguido de una de-correlación y reducción de dimensionalidad a 40 usando el linear discriminant analysis (LDA). Las observaciones resultantes son además de-correlacionadas por medio de maximum likelihood linear transform (MLLT), también se hace un speaker adaptive training (SAT) a través del esquema de feature-space maximum likelihood linear regression (fMLLR) transform estimada por hablante [143]. La señal de voz fue inicialmente analizada usando una ventana de 25 ms traslapada con 10 ms en el empalme y un coeficiente de pre-énfasis de 0.97,

5.2. Interfaz gráfica de usuario creada para Kaldi

Para la manipulación más sencilla de los parámetros de configuración generales de los módulos del reconocedor de Kaldi, se decidió crear una interfaz sencilla (utilizando el bash de linux) para configurar en línea de comandos los principales elementos intervinientes en él. Cabe mencionar que solo se manejan en esta interfaz algunas de las opciones de las que dispone la configuración de Kaldi, en este caso, las que se ocuparon para la obtención de resultados en este trabajo.

5.2.1. Menú principal

El menú principal para el acceso a la interfaz de Kaldi (Figura 5.6) muestra cuatro opciones. En la primera (*Reset task data*) se proporciona el mecanismo para eliminar los datos de corrimientos anteriores en el sistema base de archivos de

características acústicas de Kaldi, borrando archivos (de tipo MFCC) como los que se encuentran en las carpetas de *.../data/test/mfcc*, *.../data/train/mfcc*, *.../data/test/log*, *.../data/train/log*, *.../data/test/split**, y *.../data/train/split**; así como los archivos de *.../data/test/cmvn.scp*, *.../data/train/cmvn.scp* y *.../data/test/feats.scp*, *.../data/train/feats.scp*. La opción de acceso a la ejecución del MMG-MOM aparece como *Run GMM-HMM Baseline system*, y en la cual se puede hacer el corrimiento del sistema base de RAV en sus diferentes opciones. En la opción de *Run DNN-HMM system* se da acceso a las configuraciones del sistema basado en RNP-MOM.

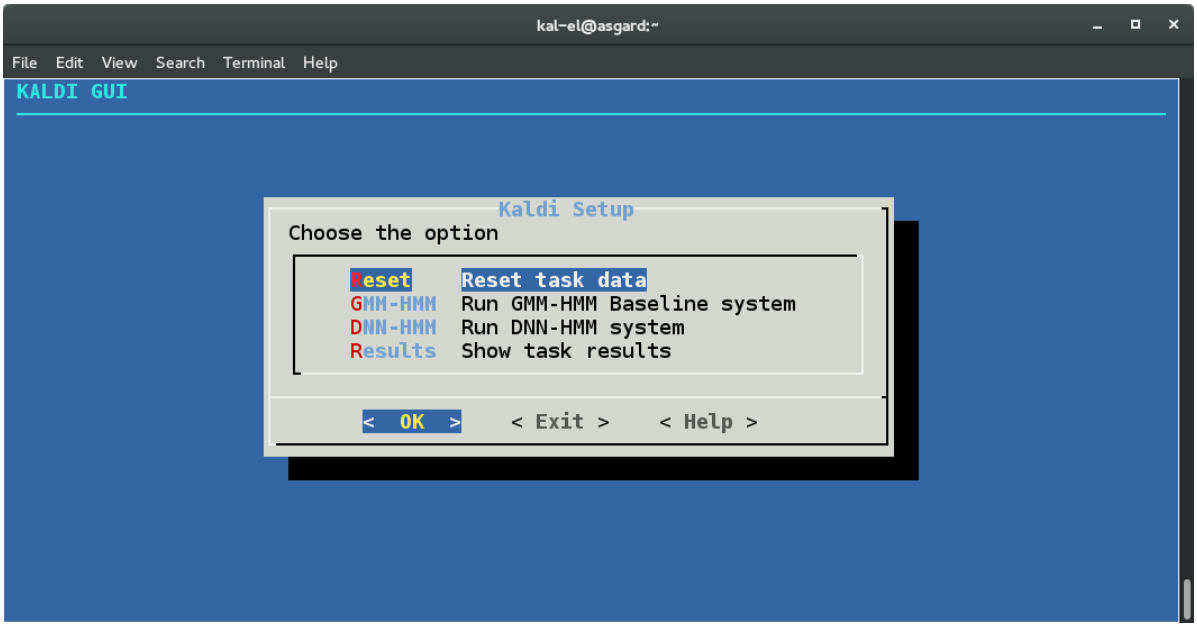

Figura 5.6: Menú principal del GUI creado para Kaldi

Adicionalmente se puede tener acceso a los resultados de WER de los corrimientos actuales en la carpeta de *.../exp* seleccionando *Results*. En *Help* se muestra la ruta donde se guarda el archivo de *log* general del GUI (*.../log_fui/*.log*).

5.2.2. Ejecución del sistema base de MMG-MOM

En el sistema base de los MMG-MOM se tienen varias opciones en el corrimiento (ver Figura 5.7). La primera alternativa es el corrimiento del RAV en modo de mono-fonos con características acústicas MFCC (*Monophone system*). En la segunda opción se tiene la forma de reconocimiento contemplando tri-fonos (ya en un entorno dependiente del contexto, *Tri-phone system*). En la selección de *Tri-phones + Delta + Delta Delta* se hace el procedimiento de RAV con tri-fonos y características acústicas MFCC con la primera y segunda derivada. En *Tri-phones + LDA + MLLT* se ejecuta el proceso de tri-fonos con características MFCC base y el análisis discriminativo lineal y con transformación MLLT. En la opción de *Tri-phones + LDA + MLLT + SAT (fMLLR)* se ejecuta el proceso de reconocimiento con características iguales a la alternativa anterior, además de que se hace un entrenamiento adaptativo por locutor mediante fMLLR. En la última opción que aparece en el menú se pueden ejecutar todas las opciones juntas, una después de la otra, con el fin de obtener resultados para la comparación con el sistema de RNP-MOM.

En esta sección se ejecutan primero los procesos de RAV con entrenamiento de *maximum likelihood* para posteriormente tomarlos como inicio para reconocimiento con base a un entrenamiento discriminativo con MPE y MMI, por ejemplo.

5.2.3. Configuración de parámetros de las redes neuronales

En la Figura 5.8 se pueden ver las opciones para la ejecución del RAV mediante redes neuronales. En la primera opción (*Stacked-RBMs parameters*) se pueden configurar los parámetros generales de funcionamiento de la parte de la pila de máquinas restrictivas de Boltzmann (pre-entrenamiento con red de creencia profunda - DBN). En la opción de *Back-propagation parameters* se pueden configurar los parámetros generales del funcionamiento del algoritmo de retro-propagación mediante el gradiente descendente. En la opción de *CD DNN-HMM with fine-tuning after DBN pretraining* se ejecuta el módulo de RAV de redes neuronales con modelos ocultos de Markov con pre-entrenamiento. En la última opción (*DNN trained*

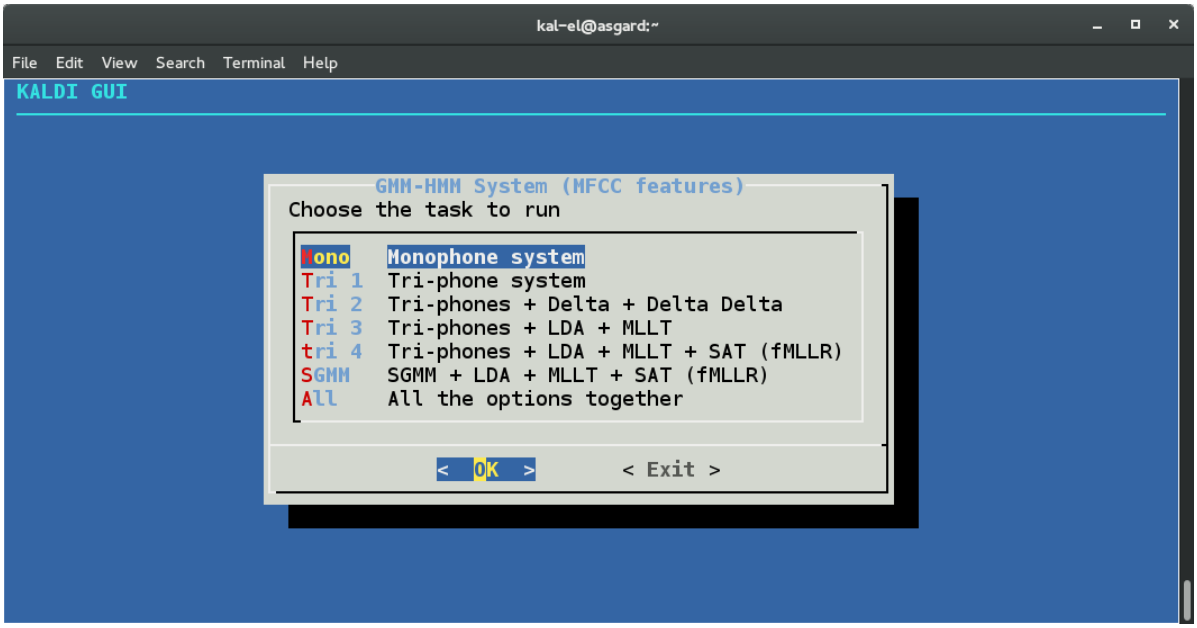

Figura 5.7: Acceso a las alternativas de corrimientos base del modelo MMG-MOM

with Back-propagation and random weights) se ejecuta el RAV con pesos aleatorios en el entrenamiento.

En la Figura 5.9 se definen los parámetros de configuración dentro de una pila de máquinas de restrictivas de Boltzmann. Se define el número de capas, el número de nodos por capa, el número de iteraciones del entrenamiento de la primera capa y de las demás capas. El número de tramas para el empalme que se usarán como entrada de la red. Finalmente se especifican las tasas de aprendizaje para la primera capa y para las restantes. También se incluye la opción de resetear los parámetros a su valor común. En la Figura 5.10 se observan los parámetros ya configurados. En la Figura 5.11 se muestra la pantalla de captura de datos de los parámetros.

En la Figura 5.12 se definen los parámetros de configuración dentro del algoritmo de retro-propagación (fine-tuning). Se puede configurar el número de capas ocultas, el número de nodos por capa, el número máximo de iteraciones para el algoritmo de entrenamiento en el modelo basado en trama, así como el número de

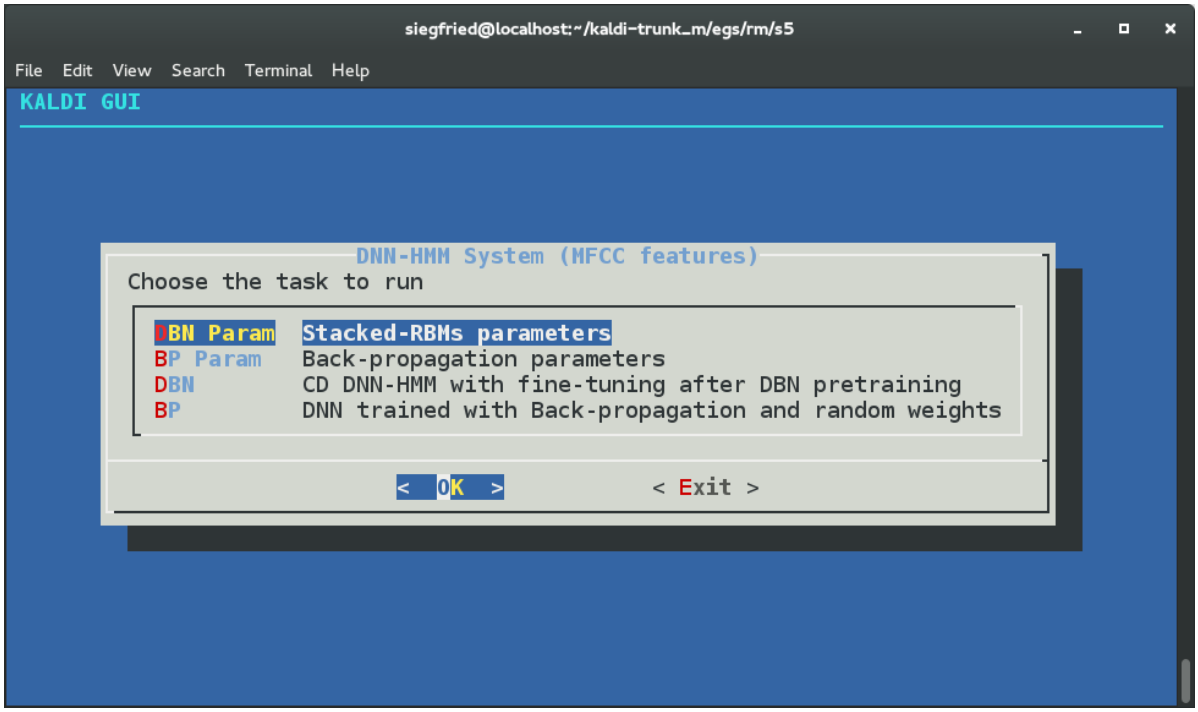

Figura 5.8: Acceso a las alternativas de corrimientos de modelos de RNP-MOM

iteraciones para entrenamiento discriminativo, el tamaño del minibatch (actualización por bloques), el número de tramas empalmadas a tomar como entrada en la red, el factor de escalamiento que se aplicará a la tasa de aprendizaje en el criterio de entropía cruzada. También se configura el porcentaje para cuando inicia el proceso de aplicación del factor de escalamiento sobre la tasa de aprendizaje en el proceso de entropía cruzada. Posteriormente viene el porcentaje base para indicar cuando el proceso de entrenamiento termina (cuando ya no existe una mejora en el proceso de validación que supere este porcentaje). También viene el orden de impulso para el caso del entrenamiento discriminativo por bMMI. Aparece también la bandera de activación o no del descarte de tramas en el entrenamiento por MMI. Finalmente aparecen las tasas de aprendizaje para el entrenamiento basado en trama y basado en discriminación. También se incluye la opción de resetear los parámetros a su valor común. En la Figura 5.13 se observan los parámetros ya configurados.

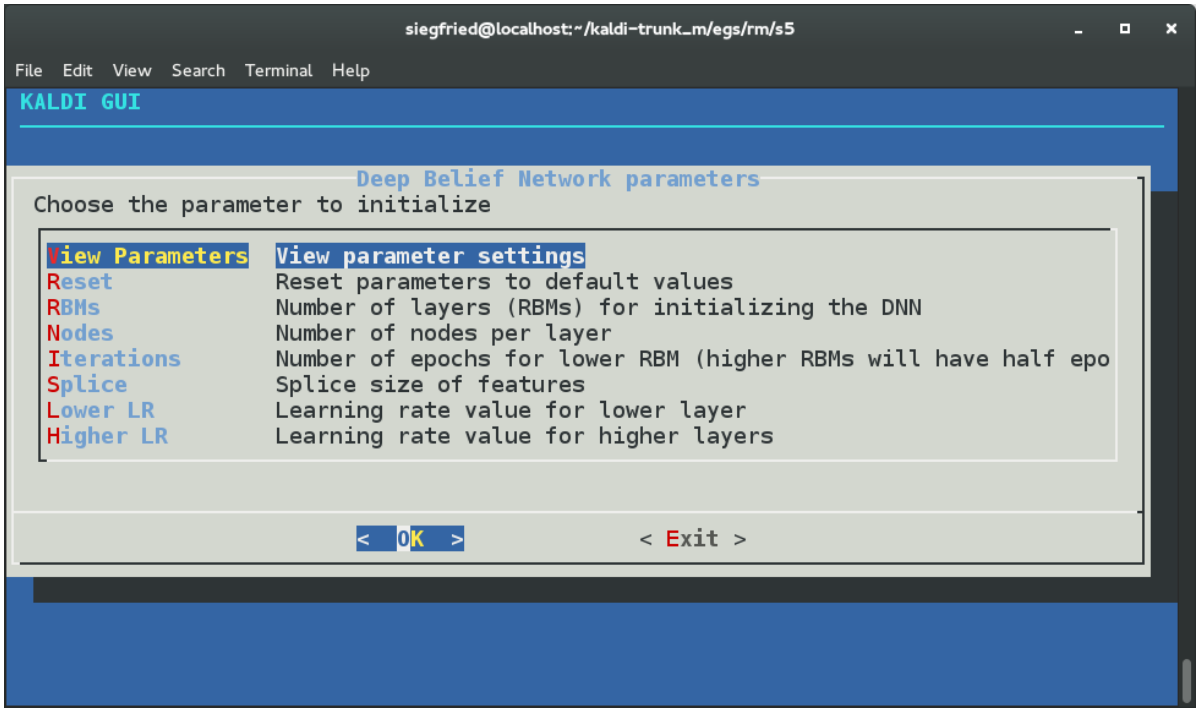

Figura 5.9: Parámetros de configuración de la DBN

5.3. Tareas de reconocimiento automático de voz: casos de estudio

Con el fin de mostrar los resultados de aplicación de las metodologías de reconocimiento, en esta sección se mencionan los resultados de tres tareas de reconocimiento.

5.3.1. Caso de estudio 1

El caso de estudio 1 se ha desarrollado utilizando un ambiente de marcado telefónico (con cadenas de dígitos y listas de nombres personales) con un corpus de voces personalizado, dependiente del texto (vocabulario), de tamaño mediano, independiente de locutor y de palabras conectadas en Español de México. El conjunto de datos consiste de 1340 sentencias de 76 hablantes (62 hombres y

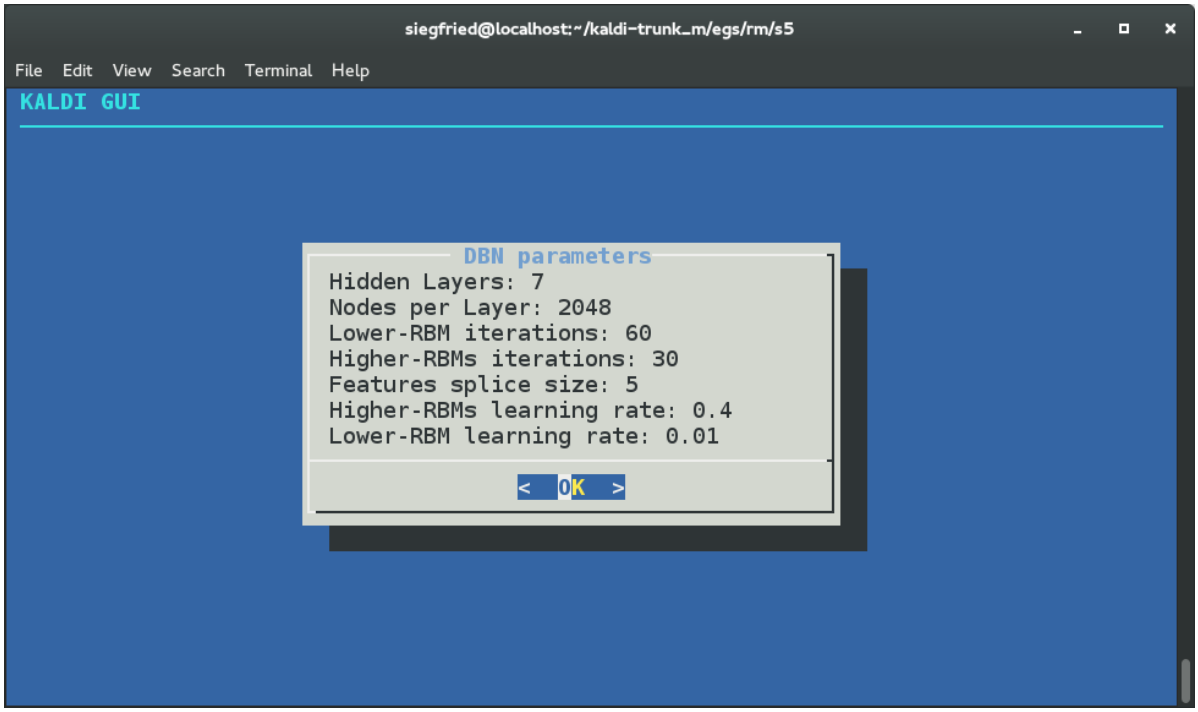

Figura 5.10: Visualización de los parámetros de la DBN

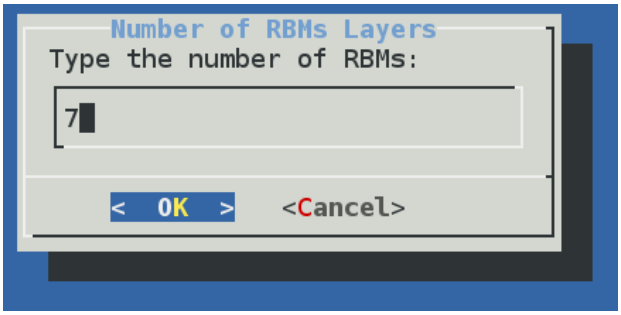

Figura 5.11: Pantalla de captura de datos de los parámetros de la red neuronal

14 mujeres); la edad de los participantes oscila entre 18 y 25.

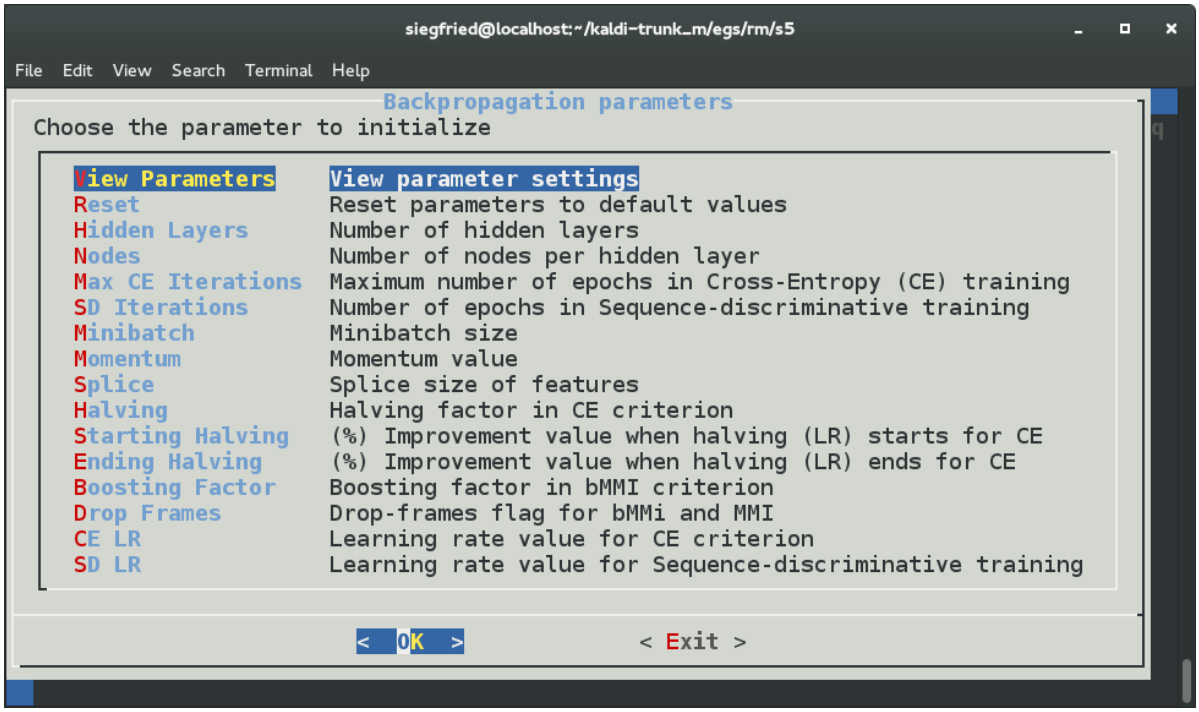

Figura 5.12: Parámetros de configuración de la RNP para el algoritmo de retro-propagación

5.3.1.1. Sistema base de MMG-MOM: definición de configuración y experimentos

El sistema base de MMG-MOM fue entrenado usando 1109 sentencias de 63 hablantes (52 hombres y 11 mujeres), y se usó para pruebas un conjunto de datos de 231 sentencias emitidas por 13 locutores (10 hombres y 3 mujeres). La tarea consiste de 1040 estados de tri-fonos ligados y 9k Gaussianas; utilizando más Gaussianas no mejora el resultado con una estimación de máxima verosimilitud para esta tarea. Las hojas del árbol de decisión fonético empleadas para la arquitectura de MMG-MOM corresponden a las unidades de salida del modelo de RNP-MOM respectivo [45], 1040 nodos en este caso.

La Tabla 5.2 muestra los resultados de WER para el sistema de mezclas Gaussianas. El modelado acústico inicial con mono-fonos alcanzó un WER de 5.37%. Además, el modelo de tri-fonos proporciona una ligera mejora, obteniendo un

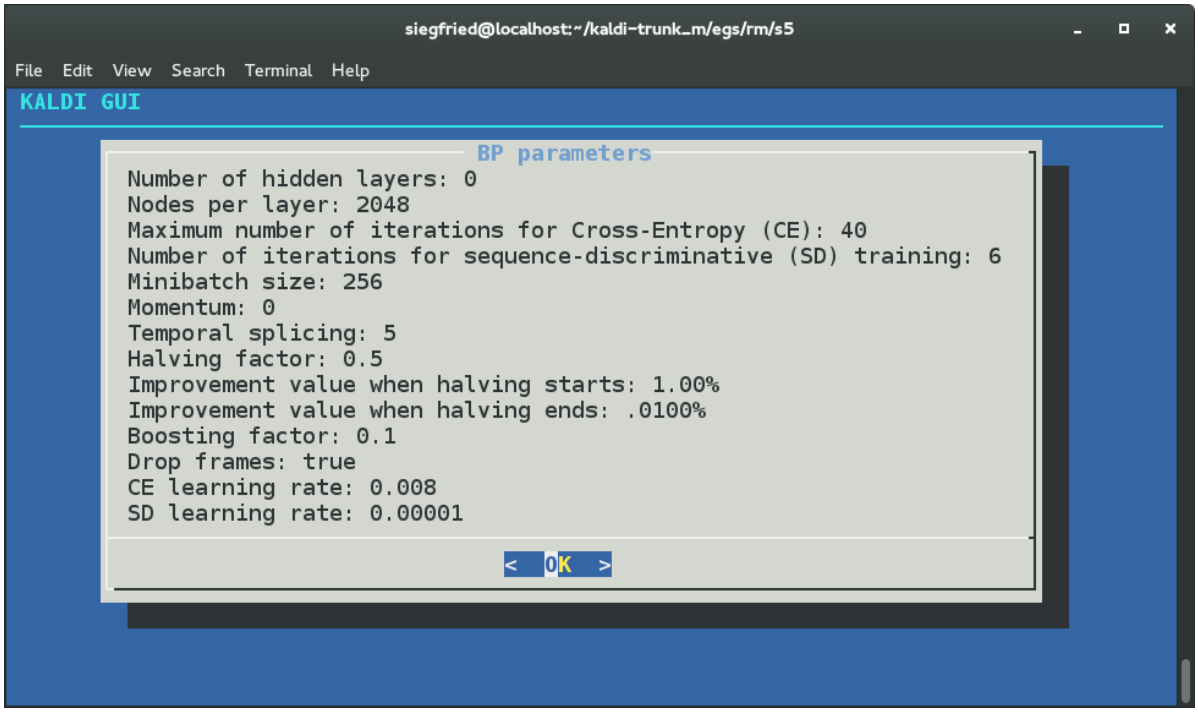

Figura 5.13: Visualización de los parámetros de la RNP

WER de $4.88\,\%$; la configuración de tri-fonos con deltas (primera y segunda derivada) ha alcanzado un WER de $4.88\,\%$. Estos experimentos fueron completados con el criterio de entrenamiento de máxima verosimilitud (MLE). Los modelos de tri-fonos con LDA+MLLT obtuvieron un WER de $5.80\,\%$, $5.94\,\%$, $6.79\,\%$ y $5.80\,\%$ para los criterios de entrenamiento de MLE, MMI, MPE y bMMI (boosted MMI, boosting factor = 0.05), respectivamente. Los resultados más sobresalientes en este esquema de tri-fonos fueron generados por los modelos con características SAT+MLLT+SAT (fMLLR), los cuales consiguieron un WER de $2.26\,\%$, $2.12\,\%$, $2.33\,\%$ y $3.18\,\%$ para MLE, MMI, MPE y bMMI, respectivamente. Es importante notar que la adaptación de características con LDA+MLLT+SAT, en el modelo de tri-fonos con mezclas Gaussianas, obtuvo $2.12\,\%$ en el WER (con el criterio de entrenamiento MMI), el mejor resultado para el modelado clásico en este caso de estudio. Además, también se muestran los resultados para una variante de los MMG, el subspace Gaussian mixture model (SGMM) [144]. Para este sistema se obtuvo un

WER de $2.33\,\%$ usando los criterios de MLE y bMMI, y para MMI la tasa de error obtenida fue de $2.26\,\%$.

Tabla 5.2: Resultados de WER (%) de diferentes configuraciones de MMG-MOM para el reconocimiento de cadenas de dígitos y listas de nombres en Español en el ambiente de marcado telefónico para el caso de estudio 1

Sistema/Criterio	MLE	MMI	MPE	bMMI
GMM mono	5.37	–	–	–
tri	4.88	–	–	–
tri+($\Delta + \Delta\Delta$)	4.95	–	–	–
tri+LDA+MLLT	5.80	5.94	6.79	5.80
tri+LDA+MLLT+SAT (fMLLR)	2.26	**2.12**	2.33	3.18
SGMM tri+LDA+MLLT+SAT (fMLLR)	2.33	2.26	–	2.33

5.3.1.2. Sistema de RNP-MOM: definición de configuración y experimentos.

El sistema de RNP ha sido entrenado utilizando 1109 sentencias (63 hablantes: 52 hombres y 11 mujeres), 112 de las cuales fueron empleadas como el conjunto de desarrollo (held-out) para propósitos de validación cruzada (8 hablantes: 5 hombres y 3 mujeres). Para pruebas, se utilizó un conjunto de datos de 231 elocuciones habladas por 13 locutores (10 hombres y 3 mujeres).

El proceso de *audio front-end* y de cálculos basados en las rejillas de Viterbi (fases de decodificación) han sido ejecutados sobre un CPU, una computadora Dell XPS 8700 series con un procesador Intel Core i7-4790 a 3.60GHz, y 16GB de DDR3 SDRAM a 1600MHz. El pre-entrenamiento y entrenamiento (fine-tuning) fueron ejecutados sobre una unidad de procesamiento gráfico de propósito general (GPGPU) NVIDIA GM107 (GeForce GTX 750 Ti), la cual contiene 2 GB de GDDR5 RAM a 5.4 Gbps y 640 núcleos de procesamiento. Se utilizó la librería CUDA 7.5.18 (en Fedora 21) para proporcionar acceso a las operaciones de matrices basadas en unidades de procesamiento gráficas.

Se definieron dos configuraciones de red neuronal: una con inicialización de pesos aleatorios y otra con pre-entrenamiento de tipo DBN. El entrenamiento se lleva a cabo usando el algoritmo de retro-propagación con actualizaciones de minibatches. Para el caso de cuando se utiliza la red de creencia profunda (DBN, máquinas restrictivas de Boltzmann apiladas) para inicializar los pesos de la RNP, el pre-entrenamiento de la primera capa (Gaussiana-Bernoulli RBM) se realizó por 60 épocas (una época es un pasada completa a través de todo el conjunto de datos de entrenamiento), y el resto de las capas (Bernoulli-Bernoulli) por 30 épocas. Más épocas no proporcionan una mejora significativa en nuestro caso de estudio, pero menos iteraciones de este proceso muestran resultados menos favorables. Cuando se usa el pre-entrenamiento basado en DBN, la máquina restrictiva de Boltzmann de tipo Gaussiana-Bernoulli se entrenó con una tasa de aprendizaje inicial de 0.01 y las RBMs de tipo Bernoulli-Bernoulli se entrenaron con una tasa de 0.4. Los pesos iniciales de la máquina restrictiva de Boltzmann fueron generados a partir de una distribución Gaussiana $N(0, 0.01)$.

5.3.1.3. Entrenamiento a nivel de trama usando la función de entropía cruzada

El primer bloque de redes neuronales entrenadas ha sido construido con una estructura de 1 a 7 capas ocultas (ver la parte superior izquierda de la Tabla 5.3), donde cada capa oculta tiene 2048 unidades sigmoideas y 1040 unidades de salida. En esta configuración, después del pre-entrenamiento DBN, la RNP es entrenada usando el algoritmo de retro-propagación con etiquetas de estado obtenidas por medio del alineamiento forzado a partir del modelo entrenado de MMG-MOM. Con una arquitectura similar, cuando se utiliza solamente el algoritmo de retro-propagación para entrenar la red, la configuración de esta tiene 1, 4 y 7 capas ocultas con 2048 unidades (ver la parte superior derecha de la Tabla 5.3). Todas las elocuciones y tramas son presentadas en forma aleatoria, y las configuraciones de la red utilizan el algoritmo de gradiente descendente para minimizar la entropía cruzada entre las etiquetas y las salidas de la red. El gradiente descendente utiliza minibatches de 256 tramas, y una tasa de aprendizaje inicial de

0.008, que se divide a la mitad cuando la mejora en la precisión de tramas en el proceso de validación cruzada entre dos épocas sucesivas cae por debajo del $1\,\%$. La optimización termina cuando la precisión de tramas incrementa menos de $0.01\,\%$. Esta definición de parámetros es la misma para ambas configuraciones: la red que utiliza pre-entrenamiento (DBN) y para la red que solo utiliza pesos aleatorios como inicialización (entrenada solo con retro-propagación, BP).

Tabla 5.3: Resultados comparativos (%) de WER entre diferentes configuraciones de RNP-MOM para el reconocimiento de cadenas de dígitos y listas de nombres en Español para el ambiente de marcado telefónico del caso de estudio 1. L es el número de capas ocultas, y N^l es el número de unidades por capa

LxN^l	DBN					BP	
	CE	MMI	bMMI	sMBR	MPE	CE	sMBR
1x$2k$	1.77	1.77	1.77	1.77	1.77	2.26	2.05
2x$2k$	1.77	1.70	1.70	1.70	1.70	-	-
3x$2k$	1.77	1.63	1.70	1.77	1.70	-	-
4x$2k$	1.63	1.63	1.63	1.56	1.56	6.51	6.44
5x$2k$	1.77	1.70	1.70	1.63	1.70	-	-
6x$2k$	1.91	1.84	1.84	1.84	1.84	-	-
7x$2k$	**1.49**	**1.49**	**1.49**	**1.49**	**1.49**	3.96	2.33
7x$3k$	1.63	1.56	1.56	1.63	1.63	-	-
1x$7k$	1.84	1.84	1.84	1.84	1.84	-	-
1x$16k$	2.26	2.19	2.26	2.19	2.12	2.55	2.48

Las Figuras 5.14-5.16 (para una configuración de red de 7, 4 y 1 capas ocultas, respectivamente) muestran los valores de entropía cruzada del proceso de entrenamiento (gráfica izquierda) y los valores correspondientes para el proceso de validación cruzada (gráfica derecha) con respecto al número de épocas usadas en el proceso respectivo. En estas figuras se puede apreciar que la métrica de error (entropía cruzada) en el modelo DBN se ve favorecido por el pre-entrenamiento, ya que la salida deseada es más cercana a la proporcionada por la red neuronal (la entropía cruzada tiende a cero). Por tanto, este esquema es más probable que obtenga tasas de reconocimiento altas en comparación con la arquitectura

de MMG-MOM y a la que utiliza solo el proceso de entrenamiento simple con el algoritmo de retro-propagación (BP) para la red neuronal. Esta situación es una de las más importantes circunstancias del porqué las nuevas redes neuronales superaron los modelos de mezclas Gaussianas. Es importante recordar que el proceso de validación cruzada se lleva a cabo con el fin de verificar que el entrenamiento es apropiado antes de pasar a la fase de pruebas. Finalmente, se puede notar que las redes pre-entrenadas requieren menor cantidad de épocas en fase de entrenamiento, y así mismo alcanzan de manera más rápida un nivel de convergencia.

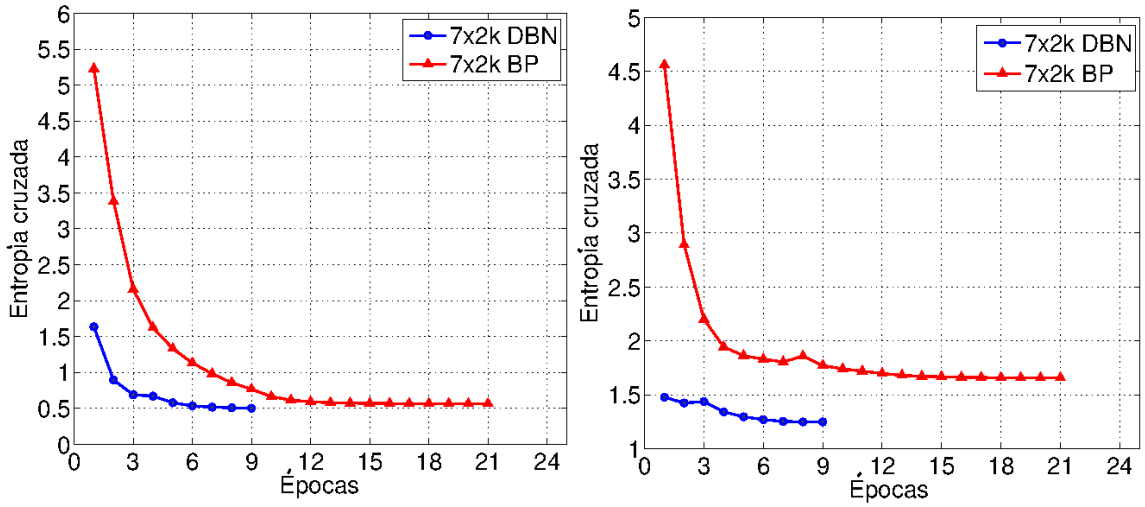

Figura 5.14: Valores de entropía cruzada en fase de ajuste de parámetros en entrenamiento (fine-tuning) para una configuración DBN y BP con 7 capas ocultas y $2k$ unidades sigmoideas. La gráfica de la izquierda corresponde a la fase de entrenamiento, y la gráfica de la derecha hace referencia al proceso de validación cruzada. El eje x es el número de épocas en cada fase

Los resultados de tasa de error para la configuración de red DBN (usando el criterio de entrenamiento a nivel de trama de entropía cruzada) con $2k$ unidades son $1.77\,\%$ de WER para 1, 2, 3 y 5 capas ocultas; $1.63\,\%$ para 4 capas, $1.92\,\%$ para 6 capas y $1.49\,\%$ de WER para 7 capas ocultas. Los mejores resultados para una red pre-entrenada son obtenidos con 7 capas ocultas (7x$2k$), $1.49\,\%$ de WER, una reducción relativa de error de $30\,\%$ con respecto al mejor resultado en el modelo

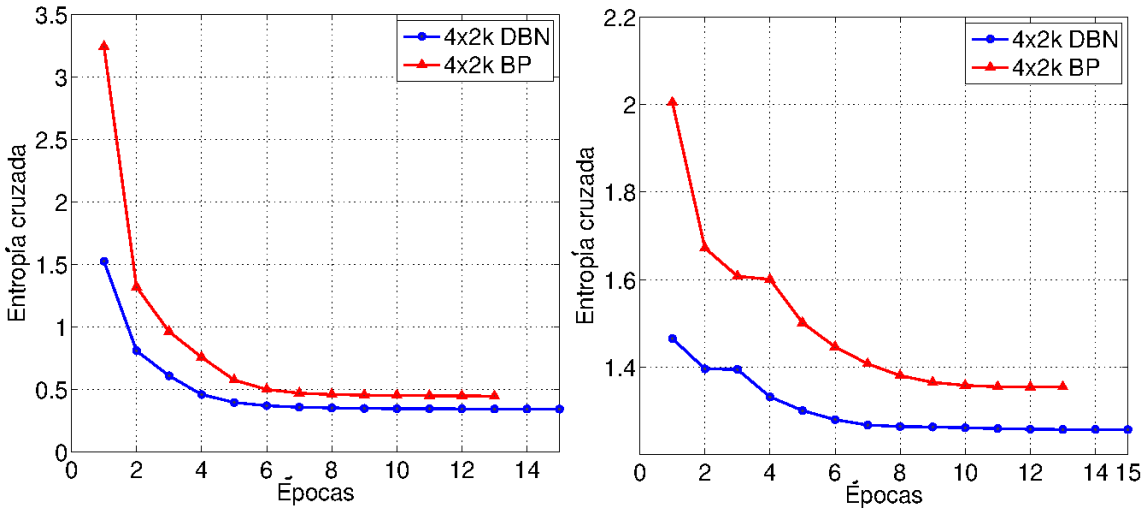

Figura 5.15: Valores de entropía cruzada en fase de ajuste de parámetros en entrenamiento (fine-tuning) para una configuración DBN y BP con 4 capas ocultas y $2k$ unidades sigmoideas. La gráfica de la izquierda corresponde a la fase de entrenamiento, y la gráfica de la derecha hace referencia al proceso de validación cruzada. El eje x es el número de épocas en cada fase

de mezclas Gaussianas. Para el caso de cuando la RNP solo es entrenada con BP (inicializada con pesos aleatorios), se alcanza un WER de $3.96\,\%$ con 7 capas ocultas y $2k$ unidades por capa (ver la parte superior derecha de la Tabla 5.3).

Como se puede ver, es posible que una red profunda, entrenada solo con BP, pueda fácilmente quedar atrapada en un mínimo local pobre [8], por ejemplo, en nuestros experimentos la configuración BP de 4x$2k$ (un WER de $6.51\,\%$). El resultado de tasa de error para la configuración de 1 capa oculta y $2k$ unidades, que es entrenada solo con BP, alcanzó un valor de $2.26\,\%$ en el WER. Este valor reafirma el supuesto que una red poco profunda es menos probable que quede atrapada en un mínimo local. La Figura 5.17 muestra un comparativo gráfico de las configuraciones de los esquemas de DBN (de 1 a 7 capas ocultas) y BP (de 1, 4 y 7 capas ocultas) con $2k$ unidades sigmoideas. Como se puede observar, el comportamiento de la gráfica es más estable en el caso de la red DBN, y por consiguiente tiende a decrementar mientras mayor es el número de capas. También se aprecia que los valores de WER para la red BP son más impredecibles y

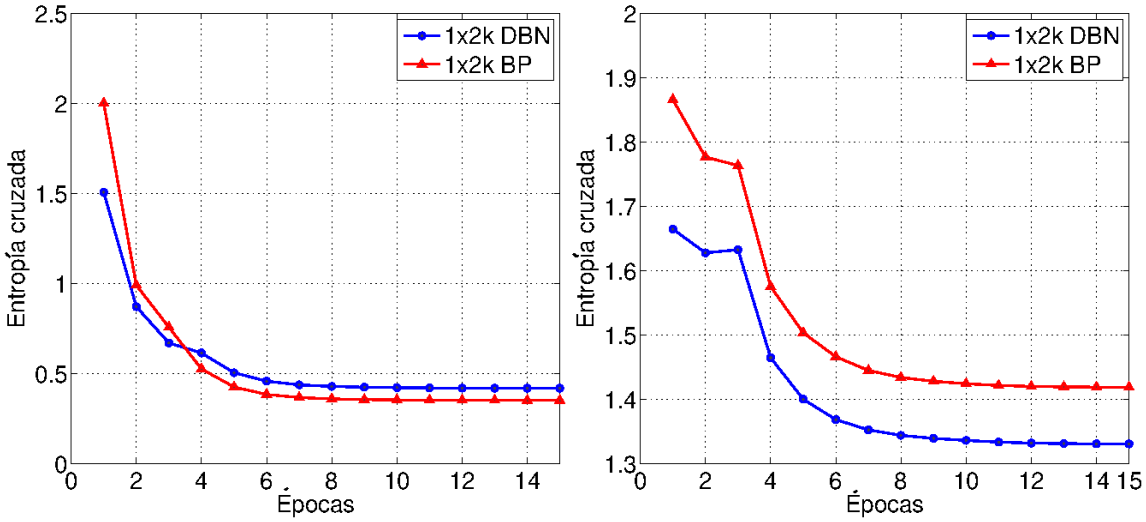

Figura 5.16: Valores de entropía cruzada en fase de ajuste de parámetros en entrenamiento (fine-tuning) para una configuración DBN y BP con 1 capa oculta y $2k$ unidades sigmoideas. La gráfica izquierda corresponde a la fase de entrenamiento, y la gráfica de la derecha hace referencia al proceso de validación cruzada. El eje x es el número de épocas en cada fase

tienden a oscilar en mayor medida, siendo capaz de caer en mínimos locales durante la fase de entrenamiento. En la Figura 5.17 se muestra la importancia de las redes neuronales modernas y el porqué existe un salto en las mejoras de las tasas de reconocimiento proporcionadas por esquemas de tipo MMG, y que una red tradicional difícilmente alcanza. Esto es observado cuando notamos que una red BP no excede el mejor porcentaje de error por palabra que brinda el MMG (WER de $2.12\,\%$).

Otras pruebas se describen en el segundo bloque de modelos de RNP (la parte inferior de la Tabla 5.3), por ejemplo, un esquema con DBN con 7 capas ocultas y $3k$ unidades (una tasa de error de $1.63\,\%$) no alcanza tan buenos resultados como los proporcionados por la configuración de 7x$2k$ unidades ($1.49\,\%$); por tanto, eso sugiere que una red más amplia no necesariamente produce una mejor precisión. Además, un modelo DBN con 1x$7k$ unidades (con una tasa de error de $1.84\,\%$) muestra el poder de una RNP con respecto a una red que no es profunda, incluso

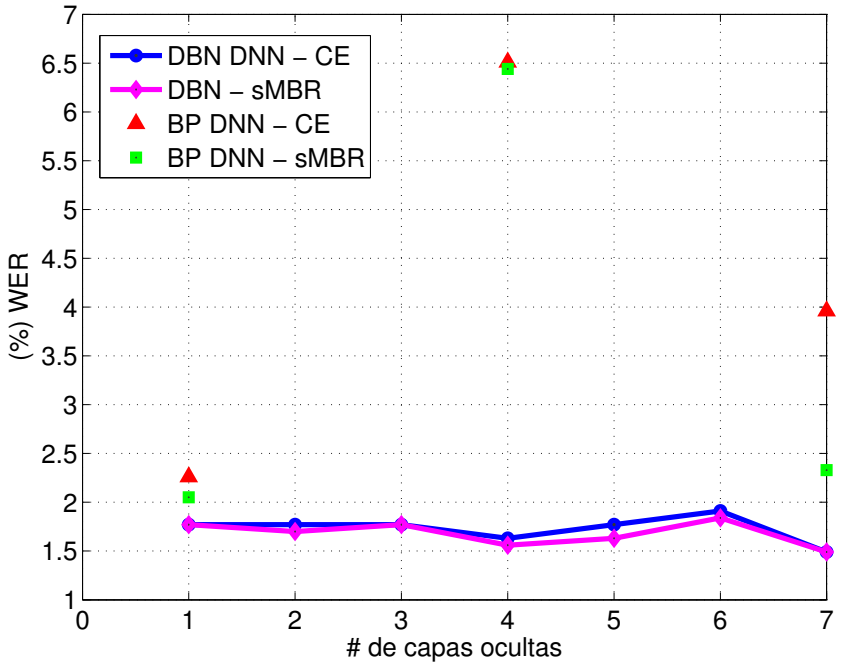

Figura 5.17: (Resultados de WER (%) por capa oculta para una red DBN y BP con los criterios de entrenamiento de entropía cruzada (CE) y sMBR. El número de capas contemplado para el proceso de entrenamiento de BP es de 1, 4 y 7

si esta tiene muchas unidades ocultas como el esquema de $1\text{x}16k$ (con una tasa de error de $2.26\,\%$). La red de $1\text{x}16k$ entrenada exclusivamente con BP tuvo un WER de $2.55\,\%$, una tasa de error no mucho más alta que la proporcionada por su equivalente en un modelo DBN, sugiriendo que en una red poco profunda el cambio del uso de pre-entrenamiento no es muy radical. Contemplando estos datos, sería oportuno mencionar que una red poco profunda, pero amplia, puede superar la tasa de error de la arquitectura de MMG, siempre y cuando esta red sea pre-entrenada.

En la Figura 5.18 se muestra la precisión de tramas de la predicción de la RNP con respecto al número de capas ocultas en el modelo DBN en comparación con el modelo BP. La Figura 5.18 muestra la precisión de tramas en la fase de entrenamiento (gráfica izquierda) y los correspondientes valores para el proceso

de validación cruzada (gráfica derecha). Hay que recordar que un modelo de RNP tiene como objetivo clasificar tramas (observaciones acústicas) en clases (estados ligados de tri-fono), para posteriormente calcular las probabilidades de emisión de estado ($b_j(o_t)$). Los valores presentados indican la precisión en la clasificación de las observaciones de entrada (vectores de características) con referencia a un estado de tri-fono del MOM (un estado ligado de tri-fono del MOM: senón). La idea es siempre tratar de alcanzar precisiones mayores en la clasificación de tramas, esto con la intención de identificar de mejor manera los tri-fonos y por consiguiente obtener tasas de reconocimiento superiores. En la Figura 5.18 se enfatizan estas circunstancias, ya que mejores tasas de clasificación en el enfoque DBN (con respecto al modelo BP) permiten modelar un RAV para mejorar las tasas de reconocimiento de voz de un enfoque de mezclas Gaussianas, el cual tradicionalmente había sido difícil de superar con una red neuronal clásica.

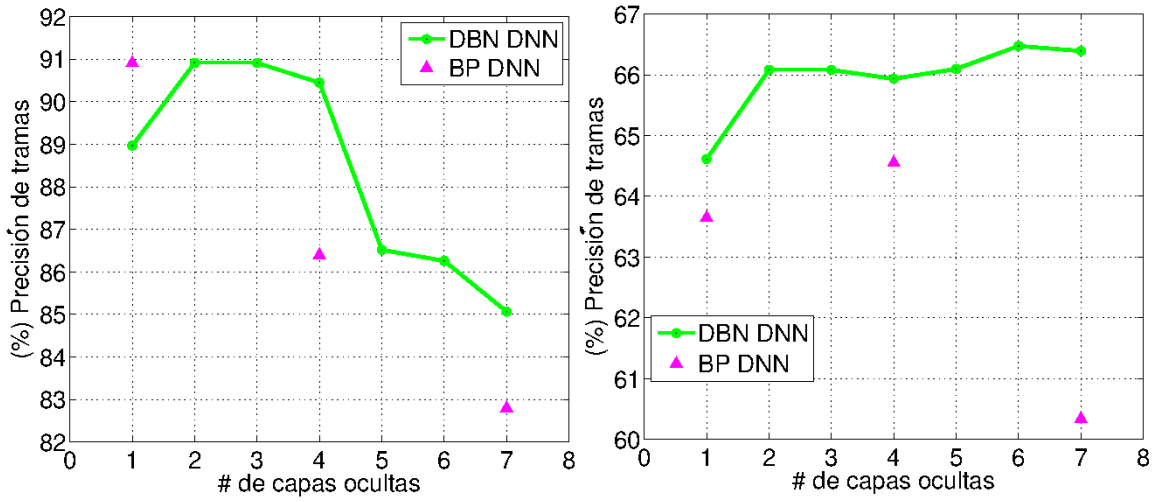

Figura 5.18: Precisión de tramas para las fases de entrenamiento (gráfica izquierda) y validación cruzada (gráfica derecha) en configuraciones DBN y BP tomando en cuenta el número de capas ocultas usadas en el caso de estudio 1. El número de capas considerado para el proceso de entrenamiento/validación-cruzada de BP es 1, 4 y 7

5.3.1.4. Entrenamiento secuencial-discriminativo

De forma similar a como el modelo de mezclas Gaussianas lo hace, el entrenamiento secuencial-discriminativo de las RNPs inicia con un conjunto de alineamientos y rejillas que son generados usando los modelos de RNP correspondientes entrenados con la función objetivo de entropía cruzada. Dichos modelos son usados como paso inicial para los criterios de entrenamiento secuencial-discriminativos (por ejemplo, MMI, bMMI, MPE y sMBR) [45]. Vesel et al. [45] sugieren que una tasa de aprendizaje fija de $1e^{-5}$ es adecuada para este tipo de entrenamiento, y que una heurística de rechazo de tramas (FR) conduce a aprendizajes más estables.

En esta etapa, los resultados muestran una mejora relativa en algunas configuraciones de red neuronal DBN. La Tabla 5.3 muestra que en algunos casos el entrenamiento discriminativo alcanza mejor precisión con respecto al entrenamiento a nivel de trama, sin embargo, el mejor resultado en WER para MMI, bMMI (boosting factor=0.1), MPE y sMBR es el mismo que el alcanzado con la función de CE, un WER de $1.49\,\%$ en la configuración de 7x$2k$. Cómo sucedió en el entrenamiento a nivel de trama con la entropía cruzada, en el entrenamiento discriminativo sin pre-entrenamiento las tasas de reconocimiento no sobrepasan los resultados ni de la red pre-entrenada ni de los esquemas de Gaussianas; excepto en el entrenamiento con BP y el criterio sMBR en una capa oculta y $2k$ unidades sigmoideas, el cual alcanzó una tasa de error por palabras de $2.05\,\%$, mejorando el WER de $2.12\,\%$ perteneciente al modelo Gaussiano. La Figura 5.17 muestra que el criterio sMBR proporciona una ligera mejora del WER con respecto al criterio de CE. La misma figura da una idea acerca del comportamiento del criterio sMBR en relación al criterio de CE, indicando leves mejoras en algunas capas alternas.

5.3.1.5. Análisis de tiempos de cómputo

El sistema base de MMG-MOM toma alrededor de 16 minutos para completarse, ejecutándose en un CPU. Además, se ha mostrado el potencial que tienen

Tabla 5.4: Resumen de tiempos de entrenamiento para algunas configuraciones en el enfoque de RAV usando RNP en el caso de estudio 1

Tipo $Lx N^l$	# Tiempo por época (min.)					# de épocas
DBN						
1x2k			0.40			60
2x2k			0.93			30
3x2k			1.26			30
4x2k			1.43			30
5x2k			1.53			30
6x2k			1.70			30
7x2k			1.80			30
1x16k			2.35			60
Fine-tuning	CE	MMI	bMMI	sMBR	MPE	
1x2k	0.28	0.54	0.37	0.57	0.69	15-6-6-6-6
2x2k	0.49	0.90	0.71	0.90	0.97	15-6-6-6-6
3x2k	0.49	0.86	0.79	0.93	0.94	15-6-6-6-6
4x2k	0.89	1.51	1.36	1.56	1.53	15-6-6-6-6
5x2k	1.10	1.90	1.73	1.91	1.93	16-6-6-6-6
6x2k	1.30	2.17	2.16	2.33	2.26	16-6-6-6-6
7x2k	1.52	2.54	2.53	2.57	2.59	9-6-6-6-6
7x3k	2.96	5.03	3.03	5.07	5.06	9-6-6-6-6
1x7k	0.73	1.13	1.01	1.20	1.20	16-6-6-6-6
1x16k	1.59	2.31	2.17	2.32	2.33	15-6-6-6-6
Solo BP						
1x16k	1.69	2.53	2.30	2.33	2.36	16-6-6-6-6
1x2k	0.28	0.51	0.36	0.59	0.62	15-6-6-6-6
4x2k	0.90	1.54	1.47	1.53	1.61	13-6-6-6-6
7x2k	1.50	2.50	2.40	2.60	2.53	21-6-6-6-6

los modelos de RNP en relación al modelado basado en MMG, pero es interesante analizar el consumo de tiempo para el entrenamiento de la red neuronal; dicho proceso es el más pesado en carga computacional. En esta sección se enfatiza esta parte, dejando un poco de lado la fase de decodificación dado su costo en carga

computacional reducido. La Tabla 5.4 muestra los resultados para algunos experimentos, por ejemplo, el WER más preciso (7x2k) toma en su pre-entrenamiento alrededor de $0.4 * 60 + 0.93 * 30 + 1.26 * 30 + 1.43 * 30 + 1.53 * 30 + 1.70 * 30 + 1.80 * 30 = 283.5$ minutos (4.72 horas), y la fase de entrenamiento con BP (fine-tuning) toma alrededor de $1.52 * 9 + 2.54 * 6 + 2.53 * 6 + 2.57 * 6 + 2.59 * 6 = 75.06$ minutos (1.25 horas). La Figura 5.19 muestra el consumo de tiempo para entrenamiento en algunas configuraciones de RNP. Como se puede notar, obviamente la tendencia es que mientras mayor es el número de capas ocultas empleado, mayor es el tiempo de pre-entrenamiento. Sin embargo, el tiempo invertido en el entrenamiento con 3 o más capas no difiere mucho, ya que la carga pesada es desarrollada en el pre-entrenamiento.

En el caso de cuando la red fue entrenada solo con el algoritmo de BP (configuraciones de 1, 4 y 7 capas ocultas), el consumo de tiempo es muy similar a su equivalente en la red DBN; no obstante, la precisión alcanzada en el reconocimiento con esta red DBN profunda es mayor, ahí radica la importancia de la inversión del tiempo. Por ejemplo, la tasa de error obtenida con 1 capa oculta y 16 unidades en el entrenamiento con la red DBN no es tan grande como el obtenido con la misma arquitectura pero entrenada solo con BP. Sin embargo, el consumo de tiempo en la fase de entrenamiento es casi el mismo (78.63 y 84.16 minutos, respectivamente). Esta situación se debe principalmente al poder del pre-entrenamiento en las redes profundas. En el entrenamiento secuencial-discriminativo usamos 6 épocas como máximo, con más iteraciones no se mejora substancialmente la precisión.

El consumo total para el entrenamiento del sistema de 7x2k toma alrededor de 6 horas (hemos observado que usando una GPU considerablemente acelera el entrenamiento, con un corrimiento en CPU el tiempo invertido sería mucho mayor). Para completar las etapas restantes en la tarea de RAV basada en RNP de este caso de estudio se requiere invertir tiempo en transformación/adaptación de características acústicas, y para generar alineamientos y la decodificación. El tiempo invertido en estos procesos es mucho menor que el implicado en el entrenamiento. El tiempo total invertido, partiendo desde cero, para entrenar y

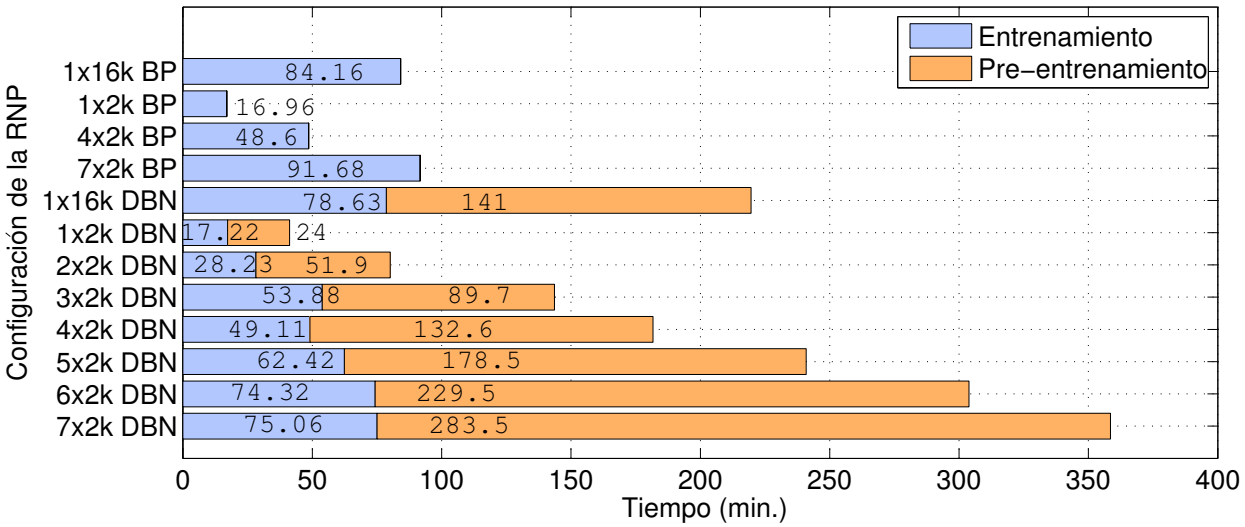

Figura 5.19: Resumen de tiempos de entrenamiento para algunos modelos de RNP en el caso de estudio 1. El consumo de tiempo es presentado para: a) RNPs diseñadas con pre-entrenamiento (DBN) + entrenamiento (fine-tuning, con BP) y b) RNPs entrenadas solo con el algoritmo BP

decodificar el sistema basado en RNP asciende aproximadamente a 7 horas. A este punto, surge una pregunta: ¿cuánto toma a una sentencia de esta tarea el ser reconocida en tiempo real? La respuesta puede ser difícil de contestar, sin embargo, el caso de estudio muestra algunos datos de tiempo para poder hacer esta estimación. Por decir, para calcular el tiempo invertido por cada sentencia de prueba, se asume que la fase de entrenamiento (incluyendo los pasos de alineamiento y cálculo de rejillas) ya ha sido realizado por la tarea. De esta manera, la fase de decodificación puede ser llevada a cabo en un sistema en tiempo real. Para este caso de estudio, el tiempo promedio invertido en fase de decodificación por cada sentencia es alrededor de 325 ms, un periodo de tiempo razonable tomando en cuenta 7 tareas paralelas (especificadas en Kaldi) para hacer el cálculo de los alineamientos, estadísticas a priori, y cálculo de rejillas para decodificación.

5.3.2. Caso de estudio 2

El caso de estudio 2 se ha desarrollado utilizando el mismo ambiente de marcado telefónico con un corpus de voces personalizado, dependiente del texto (vocabulario), de tamaño mediano, independiente de locutor y de palabras conectadas en Español de México. Este conjunto de datos consta de 1836 sentencias de 87 locutores (una mezcla de voces humanas y de elocuciones de programas de texto a voz, tales como ispeech, oddcast y vocalware).

5.3.2.1. Sistema base de MMG-MOM: definición de configuración y experimentos

El sistema base es entrenado utilizando 1585 sentencias de 73 locutores, y es probado con 251 sentencias (14 hablantes). La tarea consiste de 1203 estados ligados de tri-fono (senones) utilizando 9k Gaussianas en el modelado acústico. Los resultados del modelo MMG-MOM, utilizando el criterio de máxima verosimilitud (MLE), se muestran en la Figura 5.20: estos incluyen mono-fonos, y tri-fonos con la primera y segunda derivada, LDA, MLLT, SAT y fMLLR en la especificación de las características acústicas; en contraste con los correspondientes criterios de entrenamiento discriminativo, los resultados de las tasas de error para MMI, bMMI (boosted MMI, factor de impulso = 0.05) y MPE son mostrados en la Figura 5.21. El sistema con tri-fonos LDA+MLLT+SAT+fMLLR y el criterio de MMI obtuvo una tasa de error por palabras de 4.20 %, el valor más bajo para la tarea de mezclas Gaussianas.

5.3.2.2. Sistema de RNP-MOM: definición de configuración y experimentos.

La tarea basada en RNPs es entrenada usando 1585 sentencias (73 locutores), 180 de las cuales son usadas para el proceso de validación (3 locutores) de la red neuronal. Un conjunto de datos de 251 sentencias (14 locutores) ha sido utilizado para pruebas del sistema. Las fases de extracción de características y de decodificación han sido ejecutadas en un CPU, una computadora Dell XPS 8700

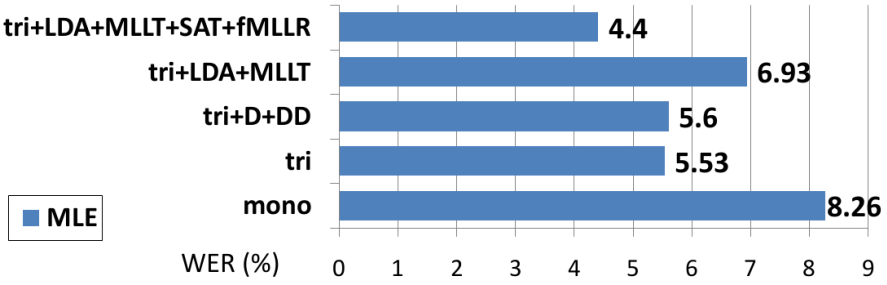

Figura 5.20: Resultados de WER (%) en la tarea de MMG-MOM usando MLE

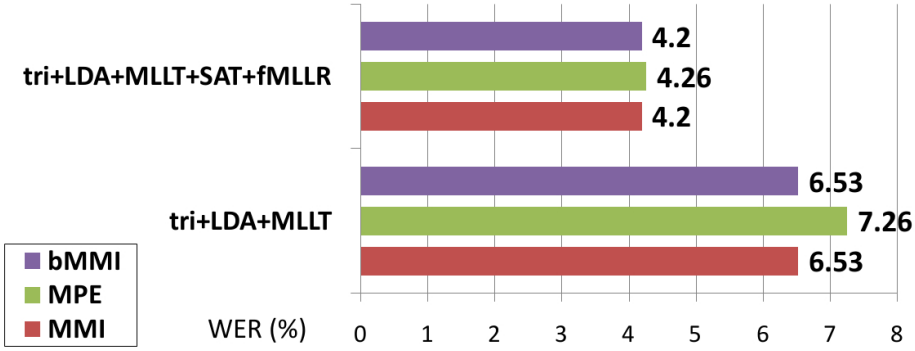

Figura 5.21: Resultados de WER (%) en la tarea de MMG-MOM con varias configuraciones en criterios discriminativos

series (Intel Core i7-4790 con una velocidad de reloj de 3.60GHz, y 16GB de 1600MHz DDR3 SDRAM). La fase de entrenamiento fue acelerada ejecutándola en una unidad de procesamiento gráfico de propósito general (GPGPU), una tarjeta NVIDIA GeForce GTX 750 Ti (la cual contiene 2 GB de GDDR5 RAM, reloj de la memoria de 5.4 Gbps y 640 núcleos de procesamiento). La librería CUDA 7.5.18 ha sido utilizada en Fedora 21 para proporcionar acceso a las operaciones de matrices basadas en el uso del GPU.

Dos configuraciones generales de los pesos han sido definidas: a) inicialización aleatoria y b) pre-entrenamiento usando la red de creencia profunda (DBN). Cuando se ha utilizado el conjunto de pesos con pre-entrenamiento para inicializar la RNP, el entrenamiento de la primera capa (máquina restrictiva de Bolztmann de tipo Gaussiana-Bernoulli) es llevado a cabo por 60 épocas (es donde

todos los datos de entrenamiento son considerados exactamente una vez) con una tasa de aprendizaje 0.01, y el resto de las capas (de tipo Bernoulli-Bernoulli) por 30 épocas, con una tasa de aprendizaje de 0.4. Los pesos iniciales de las máquinas de Boltzmann son extraídos de una distribución Gaussiana $N(0, 0.01)$.

5.3.2.2.1. Entrenamiento a nivel de trama usando la entropía cruzada (CE). Las tareas que usan redes neuronales utilizan varias configuraciones en el número de capas ocultas (ver Tabla 5.5), donde cada red neuronal tiene unidades sigmoideas, y 1203 unidades (senones) en la capa de salida. En la parte izquierda de la Tabla 5.5, después del pre-entrenamiento, la RNP es ajustada (fine-tuned) con el algoritmo de retro-propagación (BP$^{\text{FT}}$) utilizando etiquetas de estado provenientes del sistema base de MMG-MOM. El gradiente descendente utiliza minibatches de 256 tramas con el fin de minimizar la CE, y una tasa de aprendizaje de 0.008 que se divide a la mitad cuando la mejora en la precisión del procesos de validación entre dos épocas sucesivas cae por debajo de $1\,\%$. El entrenamiento termina cuando la precisión ya no alcanza un valor mayor de $0.01\,\%$. La tasa de error por palabra más baja para un modelo con pre-entrenamiento DBN y CE es obtenida con 7 capas ocultas y 2048 unidades por capa (7x2k), $3.46\,\%$ WER, una reducción de error relativa de $17.61\,\%$ con respecto al WER más bajo en un entorno de MMG. Cuando la RNP es entrenada solo con el algoritmo de retro-propagación, la WER de $4.73\,\%$ es alcanzada con 7x2k. En la Tabla 5.5 son mostrados los resultados de este tipo de red con pesos aleatorios, tanto para criterios a nivel de trama (CE) como para el mejor resultado obtenido en criterios discriminativos (DT).

Otras configuraciones son presentadas en la parte inferior de la Tabla 5.5, por ejemplo, una RNP con pre-entrenamiento DBN con 7 capas ocultas y 3k unidades no alcanzan tan buenos resultados como aquella de 7x2k, por tanto, esto sugiere que una red más ancha no necesariamente produce una mejor precisión. Además, el modelo con DBN de 1x7k muestra el poder de una RNP con respecto a una de poca profundidad, incluso aunque esta tenga muchas unidades ocultas como 1x16k.

Tabla 5.5: Resultados de WER (%) en el modelo RNP-MOM. L y N^l son el número de capas ocultas y unidades por capa, respectivamente. DT corresponde al entrenamiento discriminativo

LxN^l	*DBN+BP^{FT}*				*Solo BP*	
	CE	*MMI*	*sMBR*	*MPE*	*CE*	*DT*
1x2k	5.06	4.33	4.33	4.33	4.46	4.40$^{\text{MMI}}$
5x2k	4.33	4.26	4.26	4.26	4.06	4.06$^{\text{sMBR}}$
7x2k	3.46	3.40	**3.33**	**3.33**	5.53	4.73$^{\text{MMI}}$
9x2k	4.20	4.26	3.60	4.26	-	-
3x4k	4.06	4.06	4.06	4.06	-	-
7x3k	3.60	3.60	3.53	3.60	-	-
1x7k	4.86	4.86	4.20	4.20	-	-
1x16k	4.46	3.73	3.73	3.73	-	-

5.3.2.2.2. Entrenamiento secuencial-discriminativo. De manera similar que el enfoque de MMG-MOM, el entrenamiento secuencial-discriminativo de las RNPs comienza apartir de un conjunto de alineaciones y rejillas que son generadas usando los correspondientes modelos de RNP entrenados con la función costo de entropía cruzada. Estos modelos a nivel de trama son usados como paso inicial para los criterios de entrenamiento secuencial-discriminativo (es decir, MMI, MPE y sMBR) [45]. Vesel et al. [45] sugieren que una tasa de aprendizaje fija de $1e^{-5}$ es adecuada para un entrenamiento de este tipo, y que una heurística de rechazo de tramas (FR) conduce a un aprendizaje más estable. Los resultados muestran una mejora relativa en algunas configuraciones de RNP-DBN. La Tabla 5.5 muestra que en algunos casos, el entrenamiento secuencial-discriminativo alcanza mejor precisión con respecto al entrenamiento a nivel de trama, sin embargo, la mejor tasa en los modelos de RNP es obtenida con los criterios de MPE y sMBR (7x2k), una WER de $3.33\,\%$; es decir, una reducción relativa de $20.71\,\%$ en comparación con el WER más bajo en un modelo de MMG.

5.3.2.2.3. Tiempo de entrenamiento. El sistema de MMG-MOM base toma acerca de 16 minutos para completar la tarea completa en un CPU. El poder

del modelo acústico basado en RNP con respecto a las mezclas Gaussianas es remarcable, pero es interesante analizar el consumo de tiempo para el entrenamiento, el cual es la carga computacional más pesada. La Tabla 5.6 muestra algunos resultados de tiempo en los experimentos, por ejemplo, la configuración de la tasa de error más precisa (7x2k) toma para pre-entrenar acerca de $0.53 * 60 + 1.57 * 30 + 1.75 * 30 + 1.93 * 30 + 2.13 * 30 + 2.32 * 30 + 2.50 * 30 = 397.8$ minutos (6.63 horas), y el entrenamiento por BP (fine-tuning) toma acerca de $2.05 * 14 + 3.50 * 6 + 3.58 * 6 + 3.58 * 6 = 92.6$ minutos (1.54 horas). El tiempo total consumido por el entrenamiento de la configuración 7x2k toma acerca de 8.17 horas (usando un GPU considerablemente acelera el entrenamiento, con un CPU este tiempo sería considerablemente mayor).

Para completar los pasos de la tarea de RAV basada en RNPs, también se necesita invertir tiempo en la etapa de adaptación/transformación de características, y en generar las alineaciones y la fase de decodificación. Sin embargo, el tiempo invertido en estos procesos es menor, comparado al tiempo de entrenamiento de la RNP. El tiempo total invertido para entrenar y decodificar la mejor configuración del modelo de RNP-MOM desde cero es acerca de 9.5 horas.

Tabla 5.6: Resumen de tiempo de cálculo de los modelos de RNP

LxN^l	*# Tiempo por época (min.)*				*# de épocas*
DBN					
$(1-7)$x2k	0.53-1.57-1.75-1.93-2.13-2.32-2.50				60 - 30 - 30 - 30 - 30 - 30 - 30
BP^{FT}	CE	MMI	sMBR	MPE	
7x2k	2.05	3.50	3.58	3.58	14 - 6 - 6 - 6
Solo BP					
7x2k	2.06	3.54	3.61	3.58	18 - 6 - 6 - 6

¿Cuánto tiempo tarda una sentencia de esta tarea en ser reconocida en tiempo real? Se asume que la fase de entrenamiento (incluyendo los pasos de alineación y cálculo de las rejillas) ha sido ya realizada por la tarea. Así, la fase de decodificación puede ser llevada a cabo en un sistema en tiempo real. Para este caso de estudio, el tiempo promedio invertido en la fase de reconocimiento para cada

sentencia es acerca de 367ms (tomando en cuenta 7 trabajos en paralelo en la configuración de Kaldi).

5.3.3. Caso de estudio 3

El caso de estudio 3 ha sido desarrollado también sobre un ambiente de marcado telefónico dependiente del texto con un conjunto de datos de vocabulario de tamaño mediano e independiente de locutor, contemplando palabras continuas en Español de la parte central de México. Con el propósito de fortalecer el alcance del corpus de voces, este ha sido complementado con sentencias de audio generadas a través de aplicaciones en línea de texto a voz con acentos de características sonoras similares a la región. Las aplicaciones en línea usadas fueron ispeech, oddcast (SitePal) y vocalware. Los archivos de audio generados desde estas aplicaciones conforman un conjunto de 644 sentencias. El conjunto completo de datos incluyen 2547 sentencias de 114 hablantes (91 hombres y 23 mujeres: una mezcla de voces humanas y sentencias de texto a voz). La mayoría de los audios fueron grabados con varios tipos de ruido: distorsión de los micrófonos, lluvia, carros, animales, audios con volumen alto y bajo, y ruido ambiental en general. La edad de los participantes humanos oscila entre los 18 y 26 años.

5.3.3.1. Sistema base de MMG-MOM: definición de configuración y experimentos

El sistema base ha sido entrenado usando 2186 sentencias de 96 locutores (77 hombres y 19 mujeres), y este ha sido probado con 351 sentencias (18 hablantes: 14 hombres y 4 mujeres). La tarea consiste de 1444 senones y 9k Gaussianas. La mejor tasa de error por palabras alcanzada con el sistema de MMG es 3.66 % (con el criterio de entrenamiento de *maximum likelihood*, ML) (ver Tabla 5.7).

Tabla 5.7: Resultados del WER (%) para la tarea de RAV base del caso de estudio 3. Las tasas de error por palabras para el sistema basado en RNP son presentadas para un número variado de capas ocultas. El número correspondiente de épocas empleado en la fase de entrenamiento también se muestra.

Sistema	**# de capas**	**WER**	**# de épocas**
MMG-MOM / ML	-	3.66	-
RNP-MOM / CE			
	2	5.86	18
	3	5.76	18
	4	4.30	16
	5	4.30	16
	6	3.71	14
	7	**3.17**	16

5.3.3.2. Sistema de RNP-MOM: definición de configuración y experimentos.

La tarea de RAV basado en RNPs ha sido entrenada utilizando 2186 sentencias (96 hablantes: 77 hombres y 19 mujeres), 226 de las cuales han sido utilizadas para validación (9 locutores: 8 hombres y una mujer) de la red neuronal. Un conjunto de 351 sentencias (18 locutores: 14 hombres y 4 mujeres) ha sido empleado para las pruebas. Las fases de extracción de características y de decodificación han sido ejecutadas sobre un CPU, una computadora Dell XPS 8700 series (Intel Core i7-4790 con una velocidad de reloj de 3.60GHz, y 16GB de 1600MHz DDR3 SDRAM). La fase de entrenamiento se aceleró mediante la ejecución de la etapa sobre una unidad de procesamiento gráfica NVIDIA GeForce GTX 750 Ti (que contiene 2 GB de GDDR5 RAM, reloj de la memoria de 5.4 Gbps y 640 núcleos de procesamiento) (GPGPU). La librería CUDA 7.5.18 se empleó en Fedora 21 con el fin de proporcionar acceso a las operaciones a nivel matriz en un entorno de GPU.

La configuración se definió con un pre-entrenamiento DBN (deep belief network) [18]. En este modelo el entrenamiento de la capa Gaussiana-Bernoulli

se llevó a cabo por 60 épocas con una tasa de aprendizaje de 0.01, y las capas Bernoulli-Bernoulli restantes fueron entrenadas por 30 épocas con una tasa de aprendizaje de 0.4. Los pesos iniciales de las máquinas restrictivas de Boltzmann se extrajeron de una distribución Gaussiana $N(0, 0.01)$. La tarea de RAV se entrenó con varias capas ocultas (de 2 a 7) con 2048 unidades sigmoideas por capa, y 1444 unidades en la capa de salida a través de etiquetas de estado obtenidas mediante la tarea del modelo de MMG base. El procedimiento del gradiente utiliza minibatches de 256 tramas para minimizar la función objetivo, y una tasa de aprendizaje inicial de 0.008, que se va dividiendo a la mitad cuando la mejora entre dos procesos de validación es menor de $1\,\%$. El proceso de entrenamiento termina cuando la mejora no supera el $0.01\,\%$. El mejor resultado del WER para nuestra tarea mediante el criterio de CE es de $3.17\,\%$ (obtenido con 7 capas ocultas y 16 épocas en fase de entrenamiento); un desempeño más sobresaliente con respecto al sistema MMG base (ver Tabla 5.7).

En la tabla 5.8 se muestran los resultados de la tasa de error por palabras de las pruebas para el criterio de entropía cruzada impulsada o enfatizada con varios órdenes de impulso α y varias capas ocultas. Entendiendo que cuando $\alpha = 0$ equivale a la función de CE convencional. Aunque la entropía cruzada impulsada es un buen método, solo la arquitectura de 6 capas ocultas y un orden de impulso de $\alpha = 1$ (una WER de $3.08\,\%$ con 13 épocas en la fase de entrenamiento) pudo superar el límite de la CE ($3.17\,\%$ WER) en nuestras pruebas para esta tarea de reconocimiento. Como se puede observar, en general la red neuronal entrenada con la función de entropía cruzada impulsada toma menos épocas para alcanzar el límite impuesto en el procedimiento del gradiente descendente (la fase de fine-tuning) en comparación con la función de CE convencional.

La Tabla 5.9 muestra los resultados de las pruebas para un método alternativo ideado por Huang et al. [47], conocido como entropía cruzada con razón a posteriori logarítmica (log-posterior-ratio). La idea de este método se basa en conjuntamente minimizar la función de entropía cruzada y maximizar la proporción o razón a posteriori logarítmica entre el senón objetivo y el senón más competidor (donde λ es un factor para controlar el balance), engrandeciendo el margen

Tabla 5.8: Resultados del (%) WER para el criterio de CE impulsada con diferente orden de impulso (α) y varias capas ocultas para el caso de estudio 3. El número correspondiente de épocas en la fase de entrenamiento también se muestra.

	WER			**# de épocas**		
# de capas / α	**1**	**2**	**4**	**1**	**2**	**4**
2	4.59	5.08	5.08	13	13	13
3	5.03	3.96	4.88	17	16	13
4	4.79	4.25	4.20	16	12	12
5	4.25	4.20	4.83	14	15	13
6	**3.08**	3.52	3.32	13	12	13
7	3.52	3.37	3.32	14	12	12

entre ellos a tal grado que se incrementa el poder de generalización de la red. En los experimentos del presente trabajo, el esquema mencionado ligeramente mejoró el WER de la CE (una reducción relativa del $1.5\,\%$) en una red neuronal de arquitectura de 7 capas ocultas, mostrando su poder de predicción.

Tabla 5.9: Resultados del WER (%) para el criterio de CE/log-posterior-ratio con diferentes factores de balance (λ) y varias capas ocultas para el caso de estudio 3. El número correspondiente de épocas en la fase de entrenamiento también se muestra. Cuando $\lambda = 0$, CE/ratio equivale a la función CE clásica.

	WER			**# de épocas**		
# de capas / λ	**1e-03**	**2e-03**	**4e-03**	**1e-03**	**2e-03**	**4e-03**
2	6.40	5.86	5.86	18	18	17
3	5.22	5.22	5.76	18	17	17
4	4.30	4.30	4.35	16	16	16
5	4.49	4.49	4.49	16	16	16
6	3.71	3.71	3.71	14	14	14
7	3.17	3.17	**3.12**	16	16	16

Por otro lado, como se muestra en la Tabla 5.10, los métodos propuestos (con diferente orden de impulso α y varias capas ocultas) han alcanzado resultados superiores en el WER. Una red neuronal compuesta de 6 capas ocultas ha alcan-

zado el mejor puntaje en WER, una reducción relativa de este de $12.3\,\%$ y $10.7\,\%$ para la entropía cruzada mapeada (CE_m) y la fusión de esta con la entropía cruzada impulsada (CE_m^b), respectivamente, con respecto al criterio convencional de CE. Además, las épocas utilizadas en entrenamiento son solo 12 y 11, respectivamente. Si recordamos, la función de entropía cruzada mapeada equivale a la función de entropía mapeada impulsada cuando el orden de impulso α tiene el valor de 0.

Tabla 5.10: Resultados del WER (%) para los criterios de entropía cruzada mapeada con diferente orden de impulso (α) y varias capas ocultas para el caso de estudio 3. La función de entropía cruzada mapeada impulsada (CE_m^b) con orden de impulso $\alpha = 0$ equivale a la entropía mapeada (CE_m). El número correspondiente de épocas en la fase de entrenamiento también se muestra

	WER				**# de épocas**			
# de capas / α	**0**	**1**	**2**	**4**	**0**	**1**	**2**	**4**
CE_m^b	CE_m							
2	4.83	4.79	4.79	5.08	14	13	13	14
3	3.96	4.74	4.05	3.86	12	13	13	13
4	3.08	3.37	3.22	3.22	12	12	12	13
5	3.27	3.81	3.61	3.81	11	12	11	12
6	**2.78**	2.93	**2.83**	3.32	12	11	11	12
7	2.93	3.12	2.93	2.88	12	12	11	12

La función de entropía cruzada mapeada puede hacer la diferencia cuando existen tramas ambiguas (senones objetivo con muchos competidores, y por tanto la probabilidad a posteriori generada por la red está más distribuida entre ellos), ayudando a establecer una más definida pertenencia de ellos a un senón en particular. Si los métodos de impulso son aplicados a la función de mapeo de la CE, los esquemas proporcionan resultados interesantes (por lo menos una arquitectura, por orden de impulso α, proporciona resultados más bajos que los mostrados por la entropía cruzada convencional), ya que se enfocan sobre los senones objetivo con dificultad en su predicción correcta (estados con baja

probabilidad).

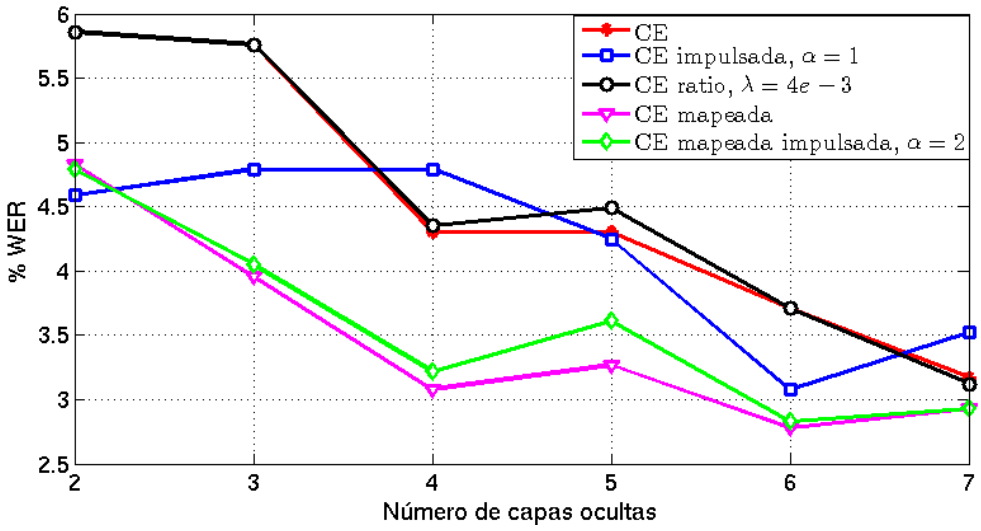

Figura 5.22: Los resultados de WER% para la tarea de RAV basada en RNPs con varios criterios de entrenamiento con su arquitectura más representativa en el presente documento.

La Figura 5.22 muestra los resultados de WER más representativos con respecto al número de capas ocultas empleadas para cada método discutido en el documento. El esquema base de CE es el punto inicial. La arquitectura de CE/log-posterior-ratio es solo un poco mejor cuando se usan 7 capas ocultas. La entropía cruzada impulsada con $\alpha = 1$ es más estable a lo largo del número de capas. El sistema basado en RNP es en general un método mejor tanto con la entropía cruzada mapeada como con la entropía mapeada impulsada a lo largo de la mayoría de las capas ocultas empleadas en la tarea de reconocimiento actual. El puntaje mejor del WER por método se muestra en la Figura 5.23, como se puede ver, los métodos de CE mapeada obtuvieron menores tasas de error por palabra sobre nuestro vocabulario.

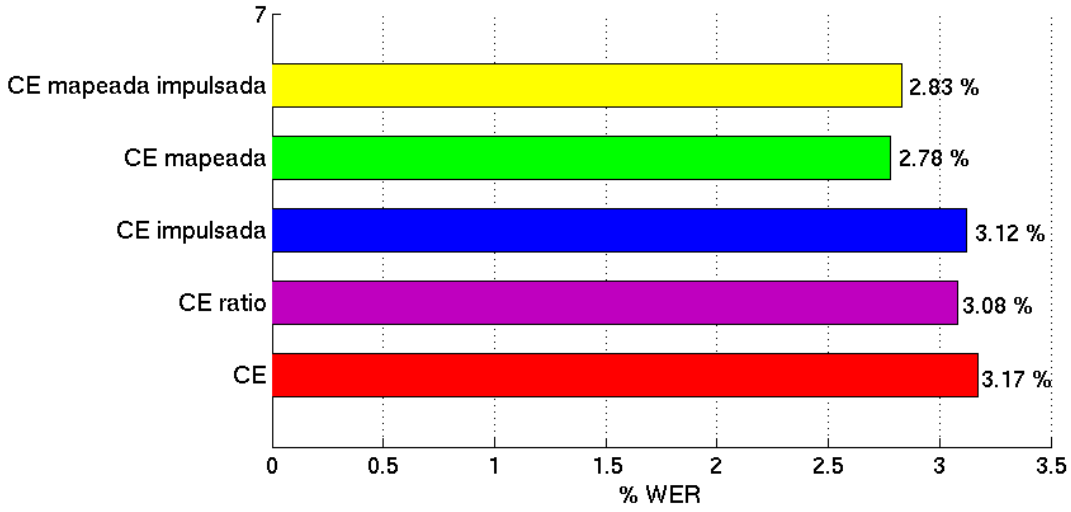

Figura 5.23: Los resultados de WER% para la mejor configuración de varios criterios de entrenamiento de RNP: entropía cruzada (CE), entropía cruzada impulsada con $\alpha = 1$, entropía cruzada/log-posterior-ratio (CE ratio) con $\lambda = 4e-03$, entropía cruzada mapeada (CE$_m$) y entropía cruzada mapeada impulsada (CE$_m^b$) con $\alpha = 2$

5.4. Diferencias entre los modelados acústicos basados en MMG y RNP

Algunos puntos claves de los modelos acústicos presentados en los casos de estudio se resumen en este apartado.

a El potencial que muestra el proceso de inicialización de los pesos de la RNP basándonos en un pre-entrenamiento. Los resultados presentados han mostrado que una red neuronal inicializada con algún tipo de modelo de aprendizaje puede proporcionar mejores y más rápidos niveles de convergencia. Esta situación es soportada por el concepto denominado *producto de expertos* (que es la combinación de múltiples modelos de variables latentes de los mismos datos, es decir, variables no observables pero que son posibles de inferir; dichos modelos multiplican sus distribuciones de probabilidad y posteriormente se normalizan) [6, 145]; a diferencia de los modelos de

Gaussianas, que son un modelo de *suma de expertos*. Los MMG no utilizan sus parámetros en una forma muy eficiente, ya que cada parámetros solo es usados en una pequeña parte de los datos, en contra parte al modelo de RNP con pre-entrenamiento DBN.

b El modelado acústico basado en redes neuronales ofrece la ventaja del esquema dependiente del contexto, ya que los vectores de características de entrada de la red son concatenados uniendo varias tramas consecutivas (una ventana de contexto); a diferencia de los MMG, que asumen entradas no correlacionadas debido al uso de matrices diagonales de covarianza.

c Un esquema de RNP puede modelar circunstancias simultáneas (diferentes capas ocultas con varias neuronas) para un vector de características acústicas, pero los MMG solo pueden modelar un dato por un componente de mezcla simple.

Capítulo 6

Discusión y aportes finales

El proceso de reconocimiento de voz ha trascendido con el paso del tiempo, desarrollado sobre varias técnicas probabilísticas o determinísticas, las cuales han probado de una o de otra forma sus ventajas y desventajas. De igual forma, los modelos recientes de mezclas Gaussianas y del uso de redes neuronales, dentro del modelado acústico, han sido de las ramas más sobresalientes en los últimos años. De acuerdo a los resultados obtenidos y a las pruebas realizadas, se puede ver que falta camino por recorrer en esta área; por consiguiente, los procedimientos metodológicos y técnicos involucrados en el RAV seguirán cambiando en los años subsecuentes.

6.1. Conclusiones

El propósito del módulo de reconocimiento de voz dentro de un sistema de diálogo hablado es proporcionar un mejor mecanismo de interacción entre el usuario y la computadora. Un rendimiento aceptable en este módulo facilita un mejor enlace entre aquellos elementos involucrados. La disciplina del RAV comenzó hace varias décadas, y con el paso del tiempo ha ido evolucionando. Después de los modelos de mezclas Gaussianas, los modelos recientes han mejorado de manera considerable la eficiencia del reconocimiento, tomando como base varios campos del conocimiento, tanto en hardware como en software, diferentes algoritmos de entrenamiento y mejores técnicas de modelado acústico-fonético.

Varias tareas de reconocimiento de voz se han llevado a cabo por años, y por

mucho tiempo los modelos basados en mezclas Gaussianas han sido de las arquitecturas más predominantes y comunes en esta línea, las cuales han mostrado que alcanzan resultados bastantes buenos. Enfoques recientes en el área de RAV han encontrado en las redes neuronales artificiales un posible modelo que permita incrementar la precisión en las tasas de reconocimiento. Existen beneficios de utilizar las redes neuronales en los enfoques de RAV, sin embargo, también se tienen complicaciones en su uso, por ejemplo, el tiempo de cómputo del entrenamiento es considerablemente mayor con respecto a los modelos de mezclas Gaussianas, incluso con el apoyo de las unidades de procesamiento gráfico. Probablemente es necesario proporcionar un equilibrio entre las tasas de reconocimiento y el tiempo de cómputo en ciertas tareas de reconocimiento. El uso de las RNP provee una alternativa para investigaciones futuras, y por consiguiente diferentes esquemas de entrenamiento han ido encaminando su desarrollo.

Como se vio en los diferentes experimentos presentados en el presente documento, tomando como base un corpus de voces personalizado en Español de la parte norte-central México, los resultados en general muestran que el uso de redes neuronales en el modelado acústico proporciona mejores tasas de reconocimiento en comparación con los modelos tradicionales de mezclas Gaussianas. En el primer caso de estudio se obtuvo una mejora relativa del $30\,\%$ usando el modelo acústico de redes neuronales (WER de $1.49\,\%$), en comparación con el modelo clásico de mezclas Gaussianas ($2.12\,\%$). En el caso de estudio 2 se obtuvieron mejoras en los resultados de tasas de reconocimiento utilizando enfoques de RNPs, con reducciones relativas del WER de $20.71\,\%$ con respecto a las tasas de reconocimiento dentro de un enfoque de MMG ($3.33\,\%$ de WER para la arquitectura profunda, y $4.20\,\%$ para el mejor resultado en el modelo Gaussiano).

Se puede asumir que un corpus de voces más grande puede proporcionar resultados similares. Con los resultados mostrados se puede entender que desarrollando transformaciones/adaptaciones en las características acústicas, y modificando los modelos acústicos, las tasas de reconocimiento por palabras pueden verse favorecidas. También es de mencionarse que los beneficios de un mecanismo de pre-entrenamiento como el que utiliza las máquinas de Boltzmann ayu-

dan de manera remarcable en las reducciones de las tasas de reconocimiento en RNPs; con pesos aleatorios, estas redes pueden verse atrapadas en mínimos locales.

Una de las varias líneas de trabajo en las que se puede desglosar un enfoque procedimental-analítico, con el fin de obtener mejores tasas de reconocimiento dentro de un sistema de diálogo con RAV, está cimentada en modificaciones en la función de entrenamiento del modelado acústico. En este sentido, las funciones de minimización y algoritmos de entrenamiento son una base fundamental. Dos variaciones a la función de costo tradicional de la entropía cruzada dentro de una red neuronal se realizaron en este trabajo con el fin de obtener mejores tasas de reconocimiento. La noción básica detrás de los modelos de entrenamiento propuestos es el mapeo de la función de costo a una representación no uniforme de la entropía cruzada. Esta función está basada en el hecho de que probabilidades a posteriori similares correspondientes a senones objetivo son en realidad diferentes debido a su medida de extropía. Esta postura ocasiona que aunque se tengan las mismas medidas de entropía cruzada, dichos senones poseen una medida dual complementaria diferente, y por consiguiente los hace variar, tratándolos en realidad en una forma diversa (a cada senón objetivo) y por consiguiente se obtiene un modelo más robusto. De esta manera, la entropía cruzada mapeada propuesta aquí permite tratar la ambigüedad de las tramas que tienen una probabilidad a posteriori distribuida entre varios senones. Además, esta función puede ser fusionada con la función de entropía cruzada impulsada con el fin de definir una nueva variante que enfatice aquellas tramas con más dificultad en su predicción (las tramas con una probabilidad a posteriori objetivo baja). Y como se vio en los resultados obtenidos en el caso de estudio 3, se obtuvo una mejora relativa del WER de 12.3% y 10.7% con la función de entropía cruzada mapeada y la función de entropía cruzada mapeada impulsada, respectivamente, en comparación con el modelo de CE tradicional.

Finalmente se puede mencionar que los objetivos del documento y de la investigación fueron cumplidos, y eso nos ayuda a concluir que la hipótesis de investigación no se rechaza.

6.2. Trabajo Futuro

Si bien es cierto que el corpus de voces empleado en el presente trabajo es relativamente diferente en contexto y tamaño a los presentados en otros trabajos, sería interesante hacer más pruebas con estos esquemas en varias condiciones con el fin de detectar puntos débiles y fuertes entre ellos. Actualmente, también sería bastante sobresaliente realizar integraciones de metodologías que hagan fusiones y variaciones de dichos métodos. Se pueden obtener mecanismos discriminativos o a nivel de trama que permitan rescatar algunas variaciones que provean buenos resultados. Adicionalmente, el área de la modificación en las arquitecturas subyacentes de las redes neuronales (ya sea profundas o convolucionales, por ejemplo) es una línea bastante explorada y que ha tenido buenas variaciones dentro de los enfoques actuales. Además de que se pueden presentar trabajos referentes también a alternativas en procesos denominados de reconocimiento de voz extremo a extremo (end-to-end speech recognition). De igual manera un punto que suele ser bastante interesante a poder considerar es integrar estos nuevos enfoques de mejoras en tasas de reconocimiento en técnicas de verificación e identificación automática de locutor. Otra área de interés en directrices de voz es el uso de enfoques acústicos que no tomen los modelos Gaussianos como base de la etiqueta en el entrenamiento de las RNPs, o incluso mecanismos alternativos al gradiente descendente, tales como enfoques Hessianos y variantes posibles en algoritmos genéticos, entre otros.

Referencias

[1] G. Saon y J. Chien, "Recent Developments in Large Vocabulary Continuous Speech Recognition," en *Proc. APSIPA ASC*, 2012.

[2] M.J.F. Gales y S.J. Young, "The application of hidden Markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no 3, pp. 195–304, 2007.

[3] S. Young, "HMMs and Related Speech Recognition Technologies," *Springer Handbook of Speech Processing, J Benesty, MM Sondhi and Y Huang (eds)*, chapter 27, 539-557.

[4] S. Young, "Large Vocabulary Continuous Speech Recognition: a Review," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 45-57, 1996.

[5] G. Saon and J. Chien, "Large-Vocabulary Continuous Speech Recognition Systems: A Look at Some Recent Advances," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 18-33, 2012.

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, y B. Kingsbury, "Deep Neural Networks for Acustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, Noviembre 2012.

[7] F. Seide, G. Li, X. Chen, y D. Yu, "Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription," en *Proc. ASRU*, pp. 24-29, 2011.

[8] F. Seide, G. Li, y D. Yu, "Conversational speech transcription using context-dependent deep neural networks," en *Proc. Interspeech, 2011*, pp. 437-440, 2011.

[9] G. Dahl, D. Yu, L. Deng, y A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Speech and Audio Proc.*, vol. 20, no. 1, pp. 30 – 42, 2012.

[10] A. Mohamed, G. Dahl, y G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[11] N. Jaitly, P. Nguyen, A. Senior, y V. Vanhoucke. "Application of Pretrained Deep Neural Networks to Large Vocabulary Conversational Speech Recognition," *UTML TR 2012-001*, 2012.

[12] D. Yu, L. Deng, y G. Dahl, "Roles of pretraining and fine-tuning in context-dependent DNN-HMMs for real-world speech recognition," *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Diciembre. 2010.

[13] T. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, y A. Mohamed, "Making Deep Belief Networks effective for large vocabulary continuous speech recognition," en *Proc. ASRU*, 2011.

[14] N. Morgan y H. Bourlard, "An introduction to hybrid HMM/connectionist continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 12, no. 3, pp. 25-42, 1995.

[15] E. Trentin y M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition," *Neurocomputing*, vol. 37, pp. 91-126, 2001.

[16] L. Deng, G. Hinton, y B. Kingsbury, "New Types of Deep Neural Network Learning for Speech Recognition and Related Applications: an Overview," en *Proc. ICASSP*, 2013.

[17] A. Mohamed, G. Dahl, y G. Hinton, "Deep Belief Networks for phone recognition," en *Proc. NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[18] G. Hinton, S. Osindero, y Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527-1554, 2006.

[19] G. Dahl, D. Yu, L. Deng, y A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN- HMMs," en *Proc. ICASSP*, 2011.

[20] Y. Dong y L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, Springer, London, 2015.

[21] Y. Bengio, P. Lamblin, D. Popovici, y H. Larochelle, "Greedy Layer-Wise Training of Deep Networks," en *Proc. Neural Information Processing Systems (NIPS)*, 2006.

[22] T. Sainath, B. Kingsbury, y B. Ramabhadran, "Improving training time of deep belief networks through hybrid pre-training and larger batch sizes," en *Proc. Neural Information Processing Systems (NIPS) Workshop on Log-linear Models*, 2012.

[23] S. Zhang, Y. Bao, P. Zhou, H. Jiang, y D. Li-Rong, "Improving deep neural networks for LVCSR using dropout and shrinking structure," en *Proc. ICASSP*, pp. 6899–6903, 2014.

[24] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, y R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detector," *arXiv preprint arXiv:1207.0580*, 2012.

[25] L. Rabiner, y B. Juang, *Fundamentals of Speech Recognition*, N.J.: Prentice-Hall, 1993.

[26] L. Deng, y D. O'Shaughnessy, *Speech Processing: An Dynamic and Optimization-Oriented Approach*, N.Y.: Marcel Dekker Inc., 2003.

[27] L. Rabiner, y R. Schafer, *Theory and Application od Digital Speech Processing*, N.J.: Prentice-Hall, 2009.

[28] L. Rabiner, y R. Schafer, "Introduction to Digital Speech Processing," *Foundations and Trends in Signal Processing* vol. 1, no. 1-2, pp. 1-194, 2007.

[29] F. Jelinek, *Statistical Methods for Speech Recognition*, EUA: MIT Press, 1998.

[30] H. Strik, A. Russel, H. Van Den Heuvel, C. Cucchiarini y L. Boves, "A Spoken Dialog System for the Dutch Public Transport Information Service," *Internacional Journal of Technology*, vol 2, pp. 121-131, 1997.

[31] U. García, "Módulo de reconocimiento de voz a texto independiente del locutor para sistemas de diálogo," Tesis de Licenciatura, Pontíficia Universidad Católica de Perú, Lima, Perú, 2009.

[32] N. Jaitly, "Exploring deep learning methods for discovering features in speech signals," Tesis Doctoral, Universidad de Toronto, Toronto, EUA, 2014.

[33] M. Faundez-Zanuy, y M. Chetouani, "Nonlinear Speech Processing: Overview and Posibilities in Speech Coding," *Progress in Nonlinear Speech Processing, LNAI*, pp. 170,189, 2007.

[34] M.A. Anusuya y S.K. Katti, "Speech Recognition by Machine: A Review," *(IJCSIS) International Journal of Computer Science and Information Security*, vol 6, no. 2, pp. 181-205, 2009.

[35] J. Guevara, "Recuperación de información en textos hablados," Tesis de Maestría, Universidad Nacional de Trujillo, Trujillo, Perú, 2010.

[36] G. Wang, "Context-Dependent Acoustic Modelling for Speech Recognition," Tesis de Doctorado, Universidad Nacional de Singapur, Singapur, 2014.

[37] X. Huang, A. Acero, y H. Hon, *Spoken language processing: a guide to theory, algorithm, and system development*, NJ: Prentice Hall PTR, 2001.

[38] E. Gose, R. Johnsonbaugh y S. Jost, *Pattern recognition and image analysis*, NJ: Prentice-Hall, Inc., 1996.

[39] J. W. Picone, "Signal modeling techniques in speech recognition," en *Proc. IEEE*, vol. 81, no. 9, pp. 119–121, 1993.

[40] R. Vergin, y D. O'Shaughnessy, "Generalized Mel Frecuency Cepstral Coefficients for Large-Vocabulary Speaker-Independent Continuos-Speech Recognition," *IEEE Transactions of Speech and Audio Processing*, vol. 7, no. 7, pp. 525-532, 1999.

[41] S. Davis y P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357–366, 1980.

[42] S. Gupta1, J. Jaafar, W. Ahmad y A. Bansal, "Feature extraction using MFCC," *Signal and Image Processing: An International Journal (SIPIJ)* vol. 4, no. 4, 2013.

[43] F. Stahlberg, T. Schlippe, V. Stephan y T. Schultz, "Towards Automatic Speech Recognition Without Pronunciation Dictionary, Transcribed Speech and Text Resources in the Target Language Using Cross-Lingual Word-to-Phoneme Alignment," en *Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2014, pp. 73-80.

[44] S. Zhang, Y. Bao, P. Zhou, H. Jiang, y L. Dai, "Improving deep neural networks for LVCSR using dropout and shrinking structure," en *Proc. ICASSP*, pp. 6849-6853, 2014.

[45] K. Vesely, A. Ghoshal, L. Burget, y D. Povey, "Sequence-discriminative training of deep neural networks," en *Proc. Interspeech*, pp. 2345-2349, 2013.

[46] K. Vesely, M. Hannemann, y L. Burget, "Semi-Supervised training of Deep Neural Networks," en *Proc. ASRU*, pp. 267-272, 2013.

[47] Z. Huang, J. Li, Ch. Weng, yCh. Lee, "Beyond Cross-Entropy: Towards Better Frame-Level Objective Functions for Deep Neural Network Training in Automatic Speech Recognition," en *Proc. Interspeech*, pp. 1214-1218, 2014.

[48] B. Kingsbury, T. Sainath, y H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," en *Proc. Interspeech*, 2012.

[49] F. Seide, G. Li, X. Chen, y D. Yu, "Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription," en *Proc. ASRU*, pp. 24-29, 2011.

[50] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, y G. Hinton, " Deep Belief networks using discriminative features for phone recognition," en *Proc. ICASSP*, pp. 5060-5063, 2011.

[51] J. Niu, L. Xie, L. Jia, y N. Hu, "Context-Dependent Deep Neural Networks for Commercial Mandarin Speech Recognition Applications," en *Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1-5 ,2013.

[52] N. Jaitly, y G. E. Hinton, "Using an autoencoder with deformable templates to discover features for automated speech recognition," en *Proc. InterSpeech*, pp. 1737-1740, 2013.

[53] K. Yao, D. You, F. Seide, H. Su, L. Deng, y Y. Gong, "Adaptation of Context-Dependent Deep Neural Networks for Automatic Speech Recognition," en *Spoken Language Technology Workshop (SLT), IEEE* , pp. 366-369, 2012.

[54] X. Li, Y. Yang, Z. Pang, y X. Wu, "A Comparative Study on Selecting Acoustic Modeling Units in Deep Neural Networks based Large Vocabulary Chinese Speech Recognition," *Neurocomputing*, vol. 170, pp. 251-256, 2015.

[55] T. K. Moon, y W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, N.J.: Prentice Hall, 2000.

[56] E. Trentin, y M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition," *Neurocomputing*, vol. 37, pp. 91-126, 2001.

[57] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," en *Proc. IEEE*, vol. 77. no. 2, 257-286, 1989.

[58] J. Bilmes, "What HMMs can do," *IEICE TRANS. INF. and SYST*, vol. E89-D, no. 3, 2006.

[59] M. Faundez-Zanuy, "Structured-Based and Template-Based Automatic Speech Recognition - Comparing parametric and non-parametric approaches," en *Proc. InterSpeech*, 2007.

[60] D. Jurafsky y J. Martin, *Speech And Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Pearson, New Jersey, 2008.

[61] D. Reynolds, "Gaussian Mixture Models," Tesis de Maestría, Universidad Nacional de Trujillo, Perú, 2010.

[62] G. McLachlan, *Mixture Models*,NY: Marcel Dekker, 1988.

[63] A. Dempster, N. Laird, y D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, vol 39, no. 1, pp. 1–38, 1977.

[64] D. A. Reynolds, T.F. Quatieri, y TR.B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol 10, no. 1, pp 19–41, 2000.

[65] H. Noguchi, K. Miura, T. Fujinaga, T. Sugahara, H. Kawaguchi, y M. Yoshimoto, "VLSI Architecture of GMM Processing and Viterbi Decoder for 60,000-Word Real-Time Continuous Speech Recognition," IEICE Transactions ELECTRON, vol. E94C no. 4, pp. 458-467, 2011.

[66] S. Young, J. Odell, y P. Woodland, "Tree-Based State Tying for High Accuracy Acoustic Modelling," en *Proc. de Human Language Technology Workshop*, pp. 307-312, Plainsboro NJ. Morgan Kaufman Publishers Inc., 1994.

[67] Z. H. Ling, S Y. Kang, H. Zen, A. Senior, y M. Schuster, "Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32 no. 3, pp. 35-52, 2015.

[68] S. Young, N.H. Russell, y J.H.S Thornton, "Token passing: a simple conceptual for connected speech recognition systems," *Relatorio técnico*, 1989.

[69] L. Deng y X. Li, "Machine Learning Paradigms for Speech Recognition: An Overview," *IEEE Transactions on audio, speech, and language processing*, vol 21, no. 5, pp. 1060-1089, 2013.

[70] L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, y P. Mermelstein, "Phonemic Hidden Markov Models with Continuous Mixture Output Densities for Large Vocabulary Word Recognition," *IEEE Transactions on signal processing*, vol. 39, no. 7, pp. 1677-1681, 1991.

[71] B. H. Juang, S. E. Levinson y M. Sondhi, "Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains," *IEEE Transactions on information theory*, vol. IT-32, no. 2, pp. 307-309, 1986.

[72] D. Yu, L. Deng y F. Seide, "The Deep Tensor Neural Network With Applications to Large Vocabulary Speech Recognition," *IEEE Transactions on audio, speech and language processing*, vol. 0, no. 0, pp. 1-9, 2012.

[73] J. Gauvain y Ch. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on speech and audio processing*, vol. 2, no. 2, pp. 291-298, 1994.

[74] G. Heigold, H. Ney y R. Schlüter, "Investigations on an EM-Style Optimization Algorithm for Discriminative Training of HMMs," *IEEE Transactions*

on audio, speech and language processing, vol. 21, no. 12, pp. 2616-2626, 2013.

[75] H. Jiang, "Discriminative training of HMMs for automatic speech recognition: A survey. Computer Speech and Language," vol. 24, no. 4, pp. 589-608, 2010.

[76] G. Heigold, H. Ney, R. Schlüter, y S. Wiesler, "Discriminative training for automatic speech recognition: Modeling, criteria, optimization, implementation, and performance," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 58-69, 2012.

[77] A. Esposito, y M. Marinaro, "Some Notes on Nonlinearities of Speech," *Nonlinear Speech Modeling, LNAI*, pp. 1-14, 2005.

[78] L. Landini, C. Manfredi, V. Positano, M. F. Santarelli, y N. Vanello, "Nonlinear prediction for oesophageal voice analysis," *Medical Engineering & Physics, LNAI*, vol. 24, pp. 529-533, 2002.

[79] M. Faundez-Zanuy, "Nonlinear Speech Processing: Overview and Posibilities in Speech Coding," *Nonlinear Speech Modeling, LNAI*, pp. 15-42, 2005.

[80] M. Faundez-Zanuy, S. McLaughlin, A. Esposito, A. Hussain, J. Schoentgen, G. Kubin, W. B. Kleijn, y P. Maragos, "Nonlinear Speech Processing: Overview and Applications," *Control and Intelligent Systems*, vol 30, no. 1, pp. 1,10, 2002.

[81] J. Thyssen, H. Nielsen, y S.D. Hansen, "Non-linear short-term prediction in speech coding," en *Proc. ICASSP*, pp. I-185 , I-188, 1994.

[82] B. Townshend,"Nonlinear prediction of speech," en *Proc. ICASSP*, vol. 1, pp. 425-428, 1991.

[83] H.M. Teager, "Some observations on oral air flow vocalization," *ASSP*, vol. 82, pp. 559-601, 1980.

[84] G. Kubin, "Nonlinear processing of speech," *Chapter 16 on Speech coding and synthesis*, Editors W.B. Kleijn y K.K. Paliwal, Ed. Elsevier, 1995.

[85] J. Thyssen, H. Nielsen, y S.D. Hansen, "Non-linearities in speech," en *Proc. IEEE workshop Nonlinear Signal y Image Processing, NSIP*, 1995.

[86] M. Faundez-Zanuy, "Modelado predictivo no lineal de la señal de voz aplicado codificación y reconocimiento de locutor," Tesis Doctoral, Universitat Politecnica de Catalunya, Barcelona, España, 1998.

[87] G. Hinton, y R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313. no. 5786, pp. 504-507, 2006.

[88] L. Deng y Y. Dong, *Deep Learning: Methods and Applications*, Washington: Now Plublishers, 2014.

[89] G. Dahl, D. Yu, L. Deng, y A. Acero, "Context-dependent DBN-HMMs in large vocabulary continuous speech recognition," en *Proc. ICASSP*, 2011.

[90] A. Mohamed, D. Yu, y L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," en *Proc. Interspeech*, 2010.

[91] A. Mohamed, G. Dahl, y G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, and Language Proc.* vol. 20, no. 1, 2012.

[92] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, y B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012.

[93] G. Dahl, D. Yu, L. Deng, y A. Acero, "Context-dependent, pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio, Speech, and Language Proc.*, vol. 20, no. 1, pp. 30-42, 2012.

[94] Y. Hen Hu, y J. Hwang, *Handbook of Neural Networks Signal Processing*, Florida: CRC Press, 2002.

[95] M. L. Seltzer, D. Yu, y Y. Wang, "An Investigation of deep neural networks for noise robust speech recognition," en *Proc. ICASSP*, vol. 13, 7398-7402, 2013.

[96] A. Maas, A. Hannun, y A. Ng, "Rectifier nonlinearities improve neural network acoustic models," en *Proc. International Conference on Machine Learning*, 2013.

[97] G.E. Dahl, T.N. Sainath, y G. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," en *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8609-8613, 2013.

[98] X. Zhang, J. Trmal, D. Povey, y S. Khudanpur, "Improving Deep Neural Network Acoustic Models using Generalized Maxout Networks," en *Proc. ICASSP*. doi: 10.1109/ICASSP.2014.6853589, 2014.

[99] M. Cai, Y. Shi, y J. Liu, "Deep maxout neural networks for speech recognition," in *Proc. ASRU*, pp. 291-296, 2013.

[100] S. M. Siniscalchi, D. Yu, L. Deng, y Ch. Lee, "Exploiting deep neural networks for detection-based speech recognition," *Neurocomputing*, vol. 106, no. 2013, 148-157, 2012.

[101] J. Pan, C. Liu, Z. Wang, Y. Hu, y H. Jiang, "Investigation of Deep Neural Networks (DNN) for Large Vocabulary Continuous Speech Recognition: Why DNN Surpass GMMs in Acoustic Modeling," en *Proc. International Symposium on Chinese Spoken Language Processing*, pp. 301-305, 2012.

[102] Y. Miao, y F. Metze, "Improving Low-Resource CD-DNN-HMM using Dropout and Multilingual DNN Training," en *Proc. Interspeech*, pp. 2237-2241, 2013.

[103] X. Chen, A. Eversole, G. Li, D. Yu, y F. Seide, "Pipelined Back-Propagation for Context-Dependent Deep Neural Networks," en *Proc. Interspeech*, 2012.

[104] L. Deng, D. Yu, y J. Platt. "Scalable stacking and learning for building deep architectures," en *Proc. ICASSP*, 2012.

[105] D. Yu, L. Deng, G. Li, y F. Seide, "Discriminative pretraining of deep neural networks," *U.S. Patent Filing*, Nov. 2011.

[106] T. Sainath, A. Mohamed, B. Kingsbury, y B. Ramabhadran, "Convolutional neural networks for LVCSR," en *Proc. ICASSP*, 2013.

[107] D. Macho, L. Mauuary, B. Noé, Y.M. Cheng, D. Ealey, D. Jou-vet, H. Kelleher, D. Pearce, y F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," en *Proc. ICSLP*, pp. 16-20, 2002.

[108] S. Sharma, D. Ellis, S. Kajarekar, P. Jain, y H. Hermansky, "Feature extraction using non-linear transformation for robust speech recognition on the aurora database," en *Proc. ICASSP*, vol. 2, pp. II1117 –II1120, 2000.

[109] L. Deng, D. Yu, Y. Gong, y A. Acero, "High-performance HMM adaptation with joint compensation of additive and convolutive distortions via Vector Taylor Series," en *Proc. ASRU*, vol. 7, pp. 65-70, 2007.

[110] Y. Hu, y Q. Huo, "An HMM compensation approach using unscented transformation for noisy speech recognition," en *Proc. ISCSLP*, pp. 346–357, 2006.

[111] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, y A. Acero, "A Minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition," en *Proc. ICASSP*, vol. 8, pp. 4041-4044, 2008.

[112] M. L. Seltzer, K. Kalgaonkar, y A. Acero, "Acoustic model adaptation via linear spline interpolation for robust speech recognition," en *Proc. ICASSP*, pp. 4550-4553, 2010.

[113] S. Renals, N. Morgan, H. Bourlard, M. Cohen, y H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. Speech and Audio Proc.*, vol. 2, pp. 161-175, 1994.

[114] Y. Xu, J. Du, L.R. Dai, y Ch. Lee, "An Experimental Study on Speech Enhancement Based on Deep Neural Networks," *IEEE Signal Processing Letters*, vol. 21 no. 1, pp. 1070-9908, 2014.

[115] G. Hinton, "A practical guide to training restricted boltzmann machines," *Tech. Rep. UTML TR 2010-003*, 2010.

[116] D.E., Rumelhart, G. Hinton, y R.J. Williams, "Learning representations by back-propagating errors," *Nature*, no. f323, pp. 533-536, 1986.

[117] M. Bacchiani, A. Senior, y G. Heigold, "Asynchronous , Online , GMM-free Training of a Context Dependent Acoustic Model for Speech Recognition," en *Proc. European Conference on Speech Communication and Technology*, 2014.

[118] A. Senior, G. Heigold, M. Bacchiani, y H. Liao, "GMM-free DNN training," en *Proc. ICASSP*, 2014.

[119] C. Zhang y P. C. Woodland, "Standalone training of context-dependent deep neural network acoustic models," en *Proc. ICASSP*, 2014.

[120] H. Bourlard y N. Morgan, "Connectionist speech recognition: a hybrid approach," Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[121] S.M. Siniscalchi, T. Svendsen, y Ch. Lee, "An artificial neural network approach to automatic speech processing," *Neurocomputing*, vol 140, pp. 326-338, 2014.

[122] A. Becerra, J. I. de la Rosa, y E. González, "A case study of speech recognition in Spanish: From conventional to deep approach," en *Proc. IEEE ANDESCON*, 2016.

[123] A. Becerra, J. I. de la Rosa, y E. González, "Speech recognition in a dialog system: from conventional to deep processing," *Multimedia Tools and Applications*, 2017. DOI: 10.1007/s11042-017-5160-5

[124] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, y J. Glass, "A complete KALDI recipe for building Arabic speech recognition systems," en *Proc. Spoken Language Technology (SLT)*, 2014, pp. 525-529.

[125] H. Seki, K. Yamamoto, y S. Nakagawa, "Comparison of Syllable-Based and Phoneme-Based DNN-HMM in Japanese Speech Recognition," en *Proc. Int. Conf. of Advanced Informatics: Concept, Theory and Application (ICAICTA)*, 2014, pp. 249-254.

[126] X. Li, C. Hong, Y. Yang, y. Wu, "Deep Neural Networks for Syllable based Acoustic Modeling in Chinese Speech Recognition," en *Proc. Signal and Information Processing Association Annu. Summit and Conf. (APSIPA)*, 2013.

[127] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, y Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol 22. no. 12, pp. 1713-1725, 2014.

[128] P. Zhou, H. Jiang, L. Dai, Y. Hu, y Q. Liu, "State-Clustering Based Multiple Deep Neural Networks Modeling Approach for Speech Recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol 23. no. 4, pp. 631-642, 2015.

[129] S. Wiesler, P. Golik, R. Schluter, y H. Ney, "Investigations on sequence training of neural networks," en *Proc. ICASSP*, 2015, pp. 4565-4569.

[130] K. Vesely M. Hannemann, y L. Burget, "Semi-supervised training of deep neural networks," en *Proc. ASRU*, 2013, pp.267-272.

[131] H. Su, G. Li, D. Yu, y F. Seide, "Error back propagation for sequence training of Context-Dependent Deep Networks for conversational speech transcription," en *Proc. ICASSP*, 2013, pp. 6664-6668,

[132] T. N. Sainath, B. Kingsbury, H. Soltau, y B. Ramabhadran, "Optimization techniques to improve training speed of deep neural networks for large speech tasks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 11, pp. 2267-2276, 2013.

[133] F. Lad, G. Sanfilippo, y G. Agró, "Extropy: complementary dual of entropy," *Statistical Science*, vol. 30, no. 1, pp. 40-58, 2015.

[134] J. Burbea y R. Rao, "On the Convexity of Some Divergence Measures Based on Entropy Functions," *IEEE Trans. Inf. Theory*, vol. 28, no. 3, pp. 489-495, 1982.

[135] R. Rao, "Use of diversity and distance measures in the analysis of qualitative data," in *Multivariate Statistical Methods in Physical Anthropology*, G. N. Van Vark and W. W. Howells, Eds. Dordrecht, Holland: D. Reidel Publishing Company, 1984, pp. 49-67.

[136] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.

[137] P. Vincent, H. Larochelle, Y. Bengio, y P. Manzagol, "Extracting and composing robust features with denoising autoencoders," en *Proc. Int. Conf. on Machine Learning (ICML)*, 2008, pp. 1096-1103.

[138] Z. Yang, S. Zhong, A. Carass, S. H. Ying, y J. L. Prince, "Deep Learning for Cerebellar Ataxia Classification and Functional Score Regression," *Lecture Notes in Computer Science*, vol. 8679, pp. 68-76, 2014.

[139] R. Scowen, "Extended bnf - generic base standards," en *Proc. of the Software Engineer-ing Standards Symposium*, pp. 25-34, 1993.

[140] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, y P.Woodland, "The HTK Book (for version 3.4)," *Cambridge University Engineering Department*, 2006.

[141] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, y K. Vesely, "The Kaldi speech recognition toolkit," en *Proc. ASRU*, 2011.

[142] C. Allauzen, M. Riley, J. Schalkwyk,W. Skut, y M. Mohri, "OpenFst: a general and efficient weighted finite-state transducer library," en *Proc. CIAA*, 2007.

[143] S. Rath, D. Povey, K. Vesel, y J. Cernock, "Improved feature processing for deep neural networks," en *Proc. Interspeech*, 2013.

[144] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembekb, N. Goel, M. Karafiát, A. Rastrowh, R. Rose, P. Schwarz, y S. Thomash, "The subspace Gaussian mixture model - A structured model for speech recognition," *Computer Speech and Language*, vol. 25, no. 2, pp. 404-439, 2011.

[145] G. E. Hinton GE, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Compututation*, vol. 14, pp. 1771–1800, 2002. doi: 10.1162/089976602760128018

Apéndice A

Definición de especificaciones de corpus de voces

A.1. Grámatica libre de contexto en formato BNF-extendido

$digit = UNO | DOS | TRES | CUATRO | CINCO | SEIS | SIETE | OCHO | NUEVE | CERO;

$fname = ALDONSO | MARIO | LUPE | RAUL | MIGUEL | CARMEN | SELENE | FABIO | CRISTIANO | ROSA | ANIBAL | LISA | ANGEL | LORENZO | CARMELITA | OCTAVIO | MARIANA | JAIME | IVAN | MARTA | CARLOS | ROSALINDA | JESUS | JUAN | JOSE | FRANCISCO | GERARDO | DANIEL | SAMUEL | SARA | LUCERO | ALEJANDRO | ESMERALDA | SERGIO | ALFONSO | FEDERICO | ANTONIA | GISELA | RAMIRO | ROGELIO | GUILLERMO | ANASTACIO | EDUARDO | ABELARDO | ALDAIR | SONIA | ISMAEL | EFREN | NARCISO | ADRIANA | ANA | LILIANA | SOFIA | DIEGO | FELIPE | JULIETA | JAVIER | EMILIO | EMILIANO | ATILANO | FABIOLA | FELIX | SANTIAGO | SEBASTIAN | DORA | LEONARDO | LUCAS | TOMAS | GABRIEL | DAVID | MANUEL | VICENTE | RAFAEL | FERNANDO | OLIVER | JULIAN | PEDRO | LARIZA | CRISTIAN | ELIAS | ANTONIO | ESTEBAN | LUCIANO | ALAN | CIRO | JOEL | CIPRIANO | JIMENA | ALONDRA | AMAYA | BENJAMIN | MARISOL

| MELQUIADES | CANDIDO | BRUNO | JACINTO | MAXIMILIANO | LUIS | CASIMIRO | NAPOLEON | GUILLERMINA | ADALBERTO | JAIRO | JORGE | AGUSTIN | ALFREDO | ALAIN | ARIAN | AARON | OMAR | PORFIRIO | ERIC | FLORINDA | ANTONIETA | HORTENCIA | LISANDRA | CAROLINA | VANESA | VILMA | LUCRECIA | TRANQUILINO | NEPOMUSENO | SOCRATES | RODOLFO | LAURENCIA | CRISANTEMO | GUADALUPE | SOLOVINO | LEONEL | ISRAEL | ROBERTO | JULIO | ENRIQUE | RODRIGO | AUXILIADORA | GUSTAVO | FAUSTO | FRIDA | FILIBERTO | EUCLIDES | EUGENIO | EULALIA | URBANO | TRANSITO | ULISES | TERESA | TRINIDAD | TEODORO | TAMARA | GREGORIO | GLORIA | ZAIDA | ZULEMA | SOCORRO | SILVIA | LETICIA | NORMA | KAREN | KASANDRA | JACOBO | JERONIMO | JOSUE | OFELIA | OLIMPIA | ORLANDO | OSIRIS | RAIMUNDO | ROLANDO | ROXANA | RUPERTO | RUTH | ROMEO | TEOFILO | RICARDO | RENATO | REBECA | RAQUEL | REINA | QUENTINA | QUIRINA | WENCESLAO | WENDY | WALDO | DOMINGO | DULCINEA | DIONISIO | HECTOR | HUMBERTO | HERNAN | HERODES | BENIGNO | BENITO | BEATRIZ | BRENDA | BONIFACIO | BLANCA | BERNARDO | BLAS | BELINDA | BALTAZAR | ABIGAIL | ABASOLO | ADELAIDA | ALMUDENA | AMARANTO | ANACLETO | ANGELICA | AQUILES | ARISTIDES | ASUNCION | ARNOLDO | ARMANDO | INDALECIO | ISIDORO | ISABEL | IMANOL | PENELOPE | PAULINO | PRISCILA | PRUDENCIA | PAULA | PALOMA | PANFILO | SEGISMUNDO | SILVANO | SILVESTRE | PETRA | SANCHO | YOLANDA | YASIR | CECILIO | CONSTANTINO | CORNELIO | CRISTOBAL | CLAUDIA | CRISANTO | CAYETANO | CARIDAD | FAUSTINO | FERMIN | FILOMENO | FLORENCIA | ADELMO | ALINA | AMADEO | AMBROSIO | AMELIA | AMPARO | ANASTASIA | ANDRES | ANGELES | ANGUSTIAS | AQUILINO | ARACELI | AURELIO | AZUZENA | BALDOMERO | BARTOLOME | BASILIO | BERNARDINO | BIANCA | BRUNILDA | KARINA | CARMELO | PERLA | CECILIA | CELESTINA | CATALINA | CELINA | CELSO | CESAREO | CINTIA | CIRILO | CLEMENTE | CLEOPATRA | CONCEPCION | CONRADO | CONSTANCIO | CRISOSTOMO | CONSUELO | DAGOBERTO | DEBORA | DEMETRIO | EDGARDO | EDMUNDO | ELENA | ELEONOR | ELIZABETH | ELOISA | ELOY

| ELSA | ELVIRA | ENCARNACION | ENGRACIA | ERASMO | ERNESTO | ESPERANZA | ESTEFANIA | EUDOSIA | EUFEMIO | EUSEBIO | EUSTAQUIO | EVANGELINA | EVARISTO | EZEQUIEL | GASPAR | GENOVEVA | GERTRUDIS | GLADIS | GODOFREDO | GRISELDA | HERIBERTO | HERMELANDO | HERMINIO | HIGINIO | HIPOLITO | HOMERO | HUGO | IGNACIO | IMELDA | IÑIGO | IRENE | ISAAC | ISAIAS | JAZMIN | JEREMIAS | JOAQUIN | JONATAN | JOSEFINA | JONAS | JULIANA | JUSTO | KATIA | LAZARO | LEANDRO | LEONIDAS | LEOPOLDO | LIDIA | MABEL | MACARENA | MAGDALENA | MAGALI | MAITE | MALVINA | MARCELO | MARGARITA | MATILDE | MAURICIO | MIRIAM | MONICA | NATALIA | NELIDA | NESTOR | NICOLAS | NIDIA | NOEMI | OLIVERIO | OSVALDO | OSCAR | OLIVIA | PATRICIA | PIEDAD | POMPEYA | QUINTINA | QUIRIACO | REINALDO | REMEDIOS | ROSAMUNDA | ROSENDO | ROSARIO | SALOMON | SAMANTA | SANSON | SERAFIN | SERENA | SOLEDAD | SUSANA | SORAYA | TADEO | TANIA | TATIANA | TEOBALDO | TEODOSIO | TIMOTEO | TRIANA | URIEL | URSULA | URANIA | VALENTINA | VENTURA | VERONICA | VICTOR | VICTORIANO | VIOLETA | VIRGILIO | VIVIANA | VLADIMIR | WILFREDO | ZORAIDA | ZENAIDA | ZACARIAS;

$lname = BECERRA | SANCHEZ | CASTILLEJOS | MORALES | RAMOS | MACIAS | GONZALEZ | BLANCO | LARA | GARCIA | LOPEZ | MEDINA | ROSALES | GAMBOA | CANTO | SALINAS | RONALDO | OLIVA | LLAMAS | JASO | MONTERO | ALVAREZ | ROBLES | SOTO | NUÑEZ | FERNANDEZ | ROBLEDO | AVELINO | CASAS | MADERO | HIDALGO | JARA | RODRIGUEZ | ZAPATA | VILLA | SOLIS | ALVEZ | JIMENEZ | ROCHA | VAZQUEZ | MARTINEZ | ESPARZA | RAMIREZ | TORRES | PADILLA | SIFUENTES | MONTES | URRUTIA | QUIÑONES | SOLER | VIDAL | FERRER | GOMEZ | PORTUGAL | SOUSA | GONCALVEZ | PEREIRA | SILVA | CASTRO | RUIZ | LEON | ARAGON | ACEVEDO | ADAME | AGUILAR | ESCOBAR | ESCALONA | LATORRE | LERMA | BALLESTEROS | SANTOS | SEGURA | NAVARRO | LLORENTE | OLMO | REVELES | ROJO | SALCEDO | SALMON | ROCA | ROMUALDO | OJEDA |

OCAMPO | OBREGON | NADAL | MURILLO | GRANADA | LAGOS | COBOS | DORIGA | ELIZALDE | ARCE | CHAVEZ | LEGUIZAMON | MONTENEGRO | VELAZQUEZ | TOLEDO | VALDEZ | BRAVO | FRANCO | DUARTE | OLIVERA | MOYANO | MAIDANA | ACUÑA | LEIVA | SORIA | AVILA | BARRIOS | MENDOZA | PAEZ | AGUERO | MENDEZ | ROLDAN | RIVERO | MANSILLA | VARGAS | CACERES | CORREA | FIGUEROA | CORDOBA | CORONEL | ARIAS | LOMELI | CARDOZO | VILLALBA | VERA | PONCE | OROZCO | MUÑOZ | LEDESMA | QUIROGA | CARRIZO | PERALTA | DOMINGUEZ | GODOY | RIOS | FERREYRA | CABRERA | JUAREZ | LUNA | YAÑEZ | ORTIZ | ROJAS | MOLINA | GUTIERREZ | AGUIRRE | HERRERA | SUAREZ | BENITEZ | ACOSTA | FLORES | RAJUELA | COLMENARES | ROMERO | ALFARO | ALMEIDA | ANGULO | ARENAS | AZNAR | ASENJO | AVILES | ARROYO | BAEZA | BARRANCO | BARRERA | BARRIGA | BAUTISTA | BERMUDEZ | BOTELLA | BRIONES | CABAÑAS | CABALLERO | CADENAS | CARDENAS | CALDERON | CALZADA | CAMACHO | CAÑAS | CARRERAS | CASANOVA | CASTAÑEDA | CASTELLANOS | CASTILLA | CEPEDA | CESPEDES | COLINA | COLLADO | CONTRERAS | CUADRADO | CUEVAS | CHACON | DUEÑAS | ECHEVERRIA | ESPAÑA | ESTRADA | FABREGAS | FAJARDO | FONSECA | FUENTES | DELAFUENTE | GALLARDO | GARRIDO | GORDILLO | GUZMAN | HARO | HUERTA | HURTADO | IBAÑEZ | IBARRA | INFANTE | IZQUIERDO | JAUREGUI | JURADO | JORDAN | LAGO | LINARES | LUMBRERAS | LLANOS | LOCADIA | MADRID | MADRIGAL | MARCOS | MELENDEZ | MIRANDA | MONTALBAN | MOSQUERA | MONDRAGON | NAVARRETE | NIÑO | OLIVARES | ORDOÑEZ | OSUNA | PACHECO | PATIÑO | PAVON | PEÑA | PEÑALVER | PIÑEIRO | PORTILLO | PUENTE | QUINTANILLA | REBOLLO | ROPERO | ROSELL | SAAVEDRA | SALAMANCA | SANJUAN | SANMIGUEL | SANTAMARIA | SERNA | SIERRA | SEVILLA | SOLANO | SORIANO | TAMAYO | TELLEZ | TEJERA | TORRECILLA | TOLOSA | URIARTE | VALENZUELA | VALVERDE | VAQUERO | VALLES | VALBUENA | VERGARA | VIÑEDO | VILLALOBOS | VIÑA | VILLAR | ZAMORA | ZAMORANO | ZABALETA | ZORRILLA | ZARAGOZA | TORO | TENORIO | TALAVERA | TELLO | SERRANO | SALVADOR | SALCIDO | SAINZ |

SABATER | RUBIO | RUEDA | ROMA | RIMANDO | RAYAS | QUEVEDO | POZO | PIZARRO | PINILLA | PENEDO | PEÑAS | PEDROZA | LUEVANO | PAREJO | PANIAGUA | PALOMARES | PALAU | PALACIOS | PALMA | ONTIVEROZ | NORIEGA | NIEVES | MUÑIZ | MILLA | MENDIZABAL | MENA | MATA | MARIÑO | MANRIQUES | SANSORES | LONGORIA | PASTOR | CARMONA | PASCUAL | CANO | IGLESIAS | DELGADILLO | VILLEGAS | DELACRUZ | VELAZCO | GIL | CALIXTO | ADRIANO | ACEVES | AHUMADA | ALCAZAR | ALDRETE | ALMAZAN | ALMARAZ | ALMIRON | ANDRADE | DEANDA | ARELLANO | ARCEO | AREVALO | ARTEAGA | ALMANZA | NUNGARAY | AYALA | ASTUDILLO | AVALOS | ALMADA | VACA | VENEGAS | BARROSO | BERNAL | BEZARES | BOLAÑOS | BORJA | BRICEÑO | BUSTOS | CALERO | CAMARGO | CANELAS | CAÑAVERAL | CANSECO | CASASOLA | CEBALLO | CIENFUEGOS | COLUNGA | CORRAL | CORVERA | COSTAS | CUELLO | CUENCA | CUESTA | CUMBRES | CUELLAR | CUERO | DAVILA | DIAZ | DURON | ENCINAS | ENCISO | ESCALERA | ESCALANTE | ESPINOZA | ESQUIVEL | FRESNO | FUERTE | GALLEGOS | GASCON | GAVILANES | GIRON | GODIN | GUERRA | GUERRERO | HEREDIA | HOYOS | ITURBIDE | ITURRALDE | LAGUNILLA | LABASTIDA | LAGUNA | LIMA | LIMONES | LUDUEÑA | LUJAN | MANSO | MARRON | MARQUEZ | MEJIA | MELGAR | MENCHACA | MENDIOLA | MESA | MIER | MONTEAGUDO | MONTAÑO | MONTALVO | MORATA | MUJICA | NAVAS | NEGRETE | OLMOS | OLMEDO | ORDOÑA | PALENCIA | PARGA | PELAEZ | PEÑUELAS | PEREZ | PEREDA | PINTO | PIZAÑA | POSADAS | PRADO | PUENTES | QUINTANO | QUIROZ | REAL | REBOLLEDO | REINOSO | RENTERIA | SALAS | SANROMAN | SANTILLAN | TRANCOSO | VIDALES | VILLAVERDE | ZUÑIGA | ZURITA;

$name = $fname $lname $lname;

(SENT-START (TELEFONO ($digit $digit $digit $digit $digit $digit $digit $digit $digit $digit) | (LLAMAR | MARCAR) $name) SENT-END)

en donde "|" denota alternativas, "[ ]" es opcionalidad, "{ }" denota cero o más repeticiones, " <> " equivale a una o mas repeticiones y "– > " denota regla de producción; así como SENT-START y SENT-END denotarán inicio y fín de cada sentencia.

A.2. Archivo transductor de la gramática libre de contexto

0 2 TELEFONO TELEFONO 0.00
103 1 <eps> <eps> 0.00
2 3 UNO UNO 0.00
3 4 <eps> <eps> 0.00
5 4 <eps> <eps> 0.00
6 4 <eps> <eps> 0.00
7 4 <eps> <eps> 0.00
8 4 <eps> <eps> 0.00
9 4 <eps> <eps> 0.00
10 4 <eps> <eps> 0.00
11 4 <eps> <eps> 0.00
12 4 <eps> <eps> 0.00
13 4 <eps> <eps> 0.00
2 5 DOS DOS 0.00
2 6 TRES TRES 0.00
2 7 CUATRO CUATRO 0.00
2 8 CINCO CINCO 0.00
2 9 SEIS SEIS 0.00
2 10 SIETE SIETE 0.00
2 11 OCHO OCHO 0.00
2 12 NUEVE NUEVE 0.00
2 13 CERO CERO 0.00

4 14 UNO UNO 0.00
14 15 <eps> <eps> 0.00
16 15 <eps> <eps> 0.00
17 15 <eps> <eps> 0.00
18 15 <eps> <eps> 0.00
19 15 <eps> <eps> 0.00
20 15 <eps> <eps> 0.00
21 15 <eps> <eps> 0.00
22 15 <eps> <eps> 0.00
23 15 <eps> <eps> 0.00
24 15 <eps> <eps> 0.00
4 16 DOS DOS 0.00
4 17 TRES TRES 0.00
4 18 CUATRO CUATRO 0.00
4 19 CINCO CINCO 0.00
4 20 SEIS SEIS 0.00
4 21 SIETE SIETE 0.00
4 22 OCHO OCHO 0.00
4 23 NUEVE NUEVE 0.00
4 24 CERO CERO 0.00
15 25 UNO UNO 0.00
25 26 <eps> <eps> 0.00
27 26 <eps> <eps> 0.00
28 26 <eps> <eps> 0.00
29 26 <eps> <eps> 0.00
30 26 <eps> <eps> 0.00
31 26 <eps> <eps> 0.00
32 26 <eps> <eps> 0.00
33 26 <eps> <eps> 0.00
34 26 <eps> <eps> 0.00
35 26 <eps> <eps> 0.00

15 27 DOS DOS 0.00
15 28 TRES TRES 0.00
15 29 CUATRO CUATRO 0.00
15 30 CINCO CINCO 0.00
15 31 SEIS SEIS 0.00
15 32 SIETE SIETE 0.00
15 33 OCHO OCHO 0.00
15 34 NUEVE NUEVE 0.00
15 35 CERO CERO 0.00
26 36 UNO UNO 0.00
36 37 <eps> <eps> 0.00
38 37 <eps> <eps> 0.00
39 37 <eps> <eps> 0.00
40 37 <eps> <eps> 0.00
41 37 <eps> <eps> 0.00
42 37 <eps> <eps> 0.00
43 37 <eps> <eps> 0.00
44 37 <eps> <eps> 0.00
45 37 <eps> <eps> 0.00
46 37 <eps> <eps> 0.00
26 38 DOS DOS 0.00
...
114 160 ABELARDO ABELARDO 0.00
114 161 ALDAIR ALDAIR 0.00
114 162 SONIA SONIA 0.00
114 163 ISMAEL ISMAEL 0.00
114 164 EFREN EFREN 0.00
114 165 NARCISO NARCISO 0.00
114 166 ADRIANA ADRIANA 0.00
114 167 ANA ANA 0.00
114 168 LILIANA LILIANA 0.00

114 169 SOFIA SOFIA 0.00
114 170 DIEGO DIEGO 0.00
114 171 FELIPE FELIPE 0.00
114 172 JULIETA JULIETA 0.00
114 173 JAVIER JAVIER 0.00
114 174 EMILIO EMILIO 0.00
114 175 EMILIANO EMILIANO 0.00
114 176 ATILANO ATILANO 0.00
114 177 FABIOLA FABIOLA 0.00
114 178 FELIX FELIX 0.00
114 179 SANTIAGO SANTIAGO 0.00
114 180 SEBASTIAN SEBASTIAN 0.00
114 181 DORA DORA 0.00
114 182 LEONARDO LEONARDO 0.00
114 183 LUCAS LUCAS 0.00
114 184 TOMAS TOMAS 0.00
114 185 GABRIEL GABRIEL 0.00
114 186 DAVID DAVID 0.00
114 187 MANUEL MANUEL 0.00
114 188 VICENTE VICENTE 0.00
114 189 RAFAEL RAFAEL 0.00
114 190 FERNANDO FERNANDO 0.00
114 191 OLIVER OLIVER 0.00
114 192 JULIAN JULIAN 0.00
114 193 PEDRO PEDRO 0.00
114 194 LARIZA LARIZA 0.00
114 195 CRISTIAN CRISTIAN 0.00
114 196 ELIAS ELIAS 0.00
114 197 ANTONIO ANTONIO 0.00
114 198 ESTEBAN ESTEBAN 0.00
114 199 LUCIANO LUCIANO 0.00

114 200 ALAN ALAN 0.00
114 201 CIRO CIRO 0.00
114 202 JOEL JOEL 0.00
114 203 CIPRIANO CIPRIANO 0.00
114 204 JIMENA JIMENA 0.00
114 205 ALONDRA ALONDRA 0.00
114 206 AMAYA AMAYA 0.00
114 207 BENJAMIN BENJAMIN 0.00
114 208 MARISOL MARISOL 0.00
114 209 MELQUIADES MELQUIADES 0.00
114 210 CANDIDO CANDIDO 0.00
114 211 BRUNO BRUNO 0.00
114 212 JACINTO JACINTO 0.00
114 213 MAXIMILIANO MAXIMILIANO 0.00
114 214 LUIS LUIS 0.00
...
519 1399 QUINTANO QUINTANO 0.00
519 1400 QUIROZ QUIROZ 0.00
519 1401 REAL REAL 0.00
519 1402 REBOLLEDO REBOLLEDO 0.00
519 1403 REINOSO REINOSO 0.00
519 1404 RENTERIA RENTERIA 0.00
519 1405 SALAS SALAS 0.00
519 1406 SANROMAN SANROMAN 0.00
519 1407 SANTILLAN SANTILLAN 0.00
519 1408 TRANCOSO TRANCOSO 0.00
519 1409 VIDALES VIDALES 0.00
519 1410 VILLAVERDE VILLAVERDE 0.00
519 1411 ZUÑIGA ZUÑIGA 0.00
519 1412 ZURITA ZURITA 0.00
1 0

A.3. Diccionario de pronunciación

AARON a r o n
ABASOLO a b a s o l o
ABELARDO a b e l a rr d o
ABIGAIL a b i g a i l
ACEVEDO a s e b e d o
ACEVES a s e b e s
ACOSTA a k o s t a
ACUÑA a k u jn a
ADALBERTO a d a l b e rr t o
ADAME a d a m e
ADELAIDA a d e l a i d a
ADELMO a d e l m o
ADRIANA a d r i a n a
ADRIANO a d r i a n o
AGUERO a g u e r o
AGUILAR a g i l a rr
AGUIRRE a g i rr e
AGUSTIN a g u s t i n
AHUMADA a u m a d a
... ...
BENITO b e n i t o
BENJAMIN b e n x a m i n
BERMUDEZ b e rr m u d e s
BERNAL b e rr n a l
BERNARDINO b e rr n a rr d i n o
BERNARDO b e rr n a r d o
BEZARES b e s a r e s

BIANCA b i a n k a
BLANCA b l a n k a
BLANCO b l a n k o
... ...
CLAUDIA k l a u d i a
CLEMENTE k l e m e n t e
CLEOPATRA k l e o p a t r a
COBOS k o b o s
COLINA k o l i n a
COLLADO k o dZ a d o
COLMENARES k o l m e n a r e s
COLUNGA k o l u n g a
CONCEPCION k o n s e p s i o n
... ...
GAVILANES g a b i l a n e s
GENOVEVA x e n o b e b a
GERARDO x e r a r d o
GERTRUDIS x e rr t r u d i s
GIL x i l
GIRON x i r o n
GISELA x i s e l a
GLADIS g l a d i s
GLORIA g l o r i a
GODOFREDO g o d o f r e d o
GODOY g o d o i
GOMEZ g o m e s
... ...
SEBASTIAN s e b a s t i a n
SEGISMUNDO s i x i s m u n d o
SEGURA s e g u r a
SEIS s e i s

SELENE s e l e n e
SENT-END SIL
SENT-START SIL
SERAFIN s e r a f i n
. . . . . .
ZARAGOZA s a r a g o s a
ZENAIDA s e n a i d a
ZORAIDA s o r a i d a
ZORRILLA s o rr i dZ a
ZULEMA s u l e m a
ZUÑIGA s u jn i g a
ZURITA s u r i t a
<UNK>SIL
!NULL SIL

donde SENT-START y SENT-END representan silencios, así como <UNK> y !NULL.

Apéndice B

Publicaciones

- Presentación en el Andean Council International Conference (ANDESCON), Arequipa, Perú, octubre 19-21, 2016.

  A. Becerra, J. I. de la Rosa, and E. González, "A case study of speech recognition in Spanish: From conventional to deep approach," in *Proc. IEEE ANDESCON*, 2016. DOI: 10.1109/ANDESCON.2016.7836212

- Artículo indexado en Multimedia Tools and Applications, 6 septiembre, 2017.

  A. Becerra, J. I. de la Rosa, and E. González, "Speech recognition in a dialog system: from conventional to deep processing," *Multimedia Tools and Applications*, 2017. DOI: 10.1007/s11042-017-5160-5

- Artículo aceptado para ser presentado en el IEEE International Autumn Meeting on Power, Electronics and Computing, Ixtapa, México, noviembre 8-10, 2017.

  A. Becerra, J. I. de la Rosa, and E. González, A. D. Pedroza, J. M. Martínez, N. I. Escalante, "Speech Recognition using Deep Neural Networks Trained with Non-uniform Frame-Level Cost Functions," in *Proc. IEEE ROPEC*, 2017.

- Artículo indexado en ***revisión*** en Multimedia Tools and Applications, 1 agosto, 2017.

  A. Becerra, J. I. de la Rosa, E. González, A. D. Pedroza, and N. I. Escalante, "Training deep neural networks with non-uniform frame-level cost function for automatic speech recognition," *Multimedia Tools and Applications*, 2017.

Índice alfabético