UNIVERSIDAD NACIONAL DE ENTRE RÍOS

FACULTAD DE INGENIERÍA

BIOINGENIERÍA

Proyecto Final

“Reconocimiento de emociones en el habla mediante autocodificadores profundos”

Autor:

- Neri Esteban Cibau

Director:

- Hugo Leonardo Rufiner

Co-director:

- Enrique Marcelo Albornoz

Oro Verde, 22 de octubre de 2013

Contenido

Objetivos **2**

Resumen **3**

Capítulo 1. Introducción **5**

La señal de la voz. Síntesis y características 6

Análisis e identificación de las emociones 8

Técnicas aplicadas a la clasificación de emociones 10

Capítulo 2. Marco teórico y materiales **15**

Neurona 15

Generalidades sobre redes neuronales 20

Entrenamiento y aprendizaje 22

Algoritmo de Retropropagación 23

Extracción de características 27

Base de datos 29

Capítulo 3. Método propuesto **33**

Aprendizaje Profundo 33

Diseño del autocodificador profundo 34

Capítulo 4. Experimentos y resultados **39**

Capítulo 5. Discusiones y conclusiones **45**

Capítulo 6. Análisis económico **49**

Materiales 51

Mano de obra 51

Equipamiento 51

Procedimiento 51

Consideraciones y cálculo de presupuesto 53

Apéndice **57**

Simulador de Redes Neuronales Stuttgart 57

Bibliografía **59**

Objetivos

Objetivos generales

- Identificar mediante autocodificadores profundos las emociones en señales de habla tomadas de una base de datos.

Objetivos específicos

- Profundizar sobre las técnicas de procesamiento digital de señales de voz.

- Investigar sobre la teoría general acerca de redes neuronales y particularizar en lo referido a autocodificadores.

- Diseñar e implementar un autocodificador profundo utilizando un lenguaje de programación o sistema apropiado.

- Realizar pruebas para evaluar la eficacia del autocodificador implementado en el reconocimiento de emociones de señales tomadas de una base de datos especializada.

- Integrar los conocimientos adquiridos durante la carrera de Bioingeniería para comprender, analizar y plantear una posible solución a un problema específico dentro del campo de incumbencia de un Bioingeniero.

Resumen

Actualmente muchas aplicaciones de la vida real están basadas en una interfaz mediante la cual el usuario interactúa con una dispositivo utilizando su voz. En los seres humanos este medio de comunicación se encuentra muy desarrollado a tal punto que somos capaces de percibir emociones simplemente escuchando a otra persona. Sin embargo esto no resulta sencillo de implementar en un sistema de reconocimiento de emociones. En primer lugar se deben analizar y extraer características adecuadas de la señal de habla dado que utilizar la señal cruda no resulta conveniente. En segundo lugar, debe desarrollarse un clasificador capaz de identificar las emociones a partir de dichas características. Una de las herramientas que tenemos al alcance para esta tarea son las redes neuronales artificiales y dentro de ellas el Perceptrón Multicapa. Una red con varias capas ocultas permite representar mejor tareas que requieren altos niveles de abstracción como es el caso del reconocimiento de emociones, pero surge el problema de poder entrenarlas correctamente mediante el algoritmo de retroprogación. Como posible solución a este inconveniente se propone un nuevo método que utiliza autocodificadores para ir entrenando sucesivamente las capas de una red con 2 o mas capas ocultas. Para evaluar el método se plantea la clasificación de 7 emociones de un corpus (incluyendo el estado emocional neutral) y se expermientan distintas estructuras de clasificador y parámetros de entrenamiento. Se logró un desempeño del clasificador de 70,48% superando a los resultados obtenidos mediante entrenamiento estándar.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
N. Cibau, H. L. Rufiner & E. M. Albornoz; "Reconocimiento de emociones en el habla mediante autocodificadores profundos (Undergraduate project)"
Facultad de Ingeniería - Universidad Nacional de Entre Ríos, 2013.

Capítulo 1. Introducción

El desarrollo continuo de nuevas tecnologías y la introducción de sistemas interactivos demanda a su vez medios de comunicación mas naturales entre hombres y máquinas, que se asemejen a los utilizados por lo seres humanos entre sí. Uno de los desafíos mas importantes que se presentan a la hora de disminuir esa brecha es el de poder dotar a una máquina de la capacidad de extraer información de esa comunicación sin que haya sido explicitada por el usuario. El canal de la voz, como medio de interacción, es ampliamente estudiado e implementado en diversidad de dispositivos usados cotidianamente. Sin embargo, estos son sistemas de Reconocimiento Automático del Habla (RAH) que solo interpretan el contenido lingüístico, dejando de lado una parte importante de la información.

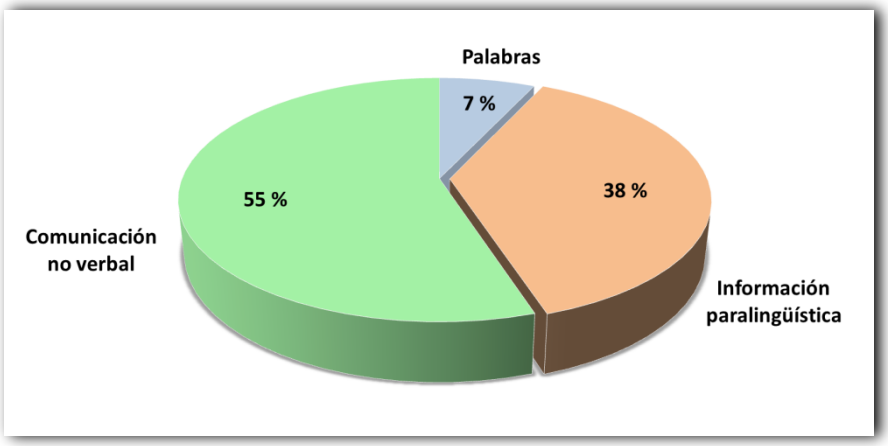

Figura 1 – Distribución de los componentes en la comunicación humana

La Figura 1 muestra la importancia que tiene cada uno de los tres componentes que intervienen en la comunicación “cara a cara” entre dos personas. Fue el resultado de estudios llevados a cabo por Mehrabian [MEHRABIAN, 1971] quien estableció que solo un 7% de la información está contenida en las palabras mientras que un 38% se transmite como información paralingüística (intensidad y tono de voz, duración de sílabas y palabras, etc). El restante 55% incluye comunicación no verbal a través del lenguaje corporal (expresiones faciales, gestos, señas, etc). Por lo tanto, desarrollar una interfaz óptima que logre asemejarse a una comunicación natural requeriría incluir estos tres componentes, pero uno de los inconvenientes que se presentan a la hora de inspeccionar la información no verbal es la necesidad de instalar cámaras que registren fotografías o video para poder detectar los movimientos de la persona. Esto no solo torna mas complejo el sistema sino que además restringe la libertad del usuario que debe mantenerse en cierta posición para poder ser captado correctamente. Por el contrario el contenido verbal y paralingüístico puede ser captado simplemente con un micrófono. Esta información está influenciada por el estado emocional del hablante, y siguiendo el camino inverso, el objetivo consiste en extraer las características representativas de esa información y aplicar técnicas apropiadas para determinar la emoción que las originó.

En el desarrollo de este capítulo se dará un breve repaso de las estructuras que participan en la emisión de voz y de algunas de las características de dicha señal. Posteriormente se presentarán los distintos enfoques referidos a la identificación y clasificación de emociones. Se abordarán las técnicas usadas actualmente para el reconocimiento automático de

emociones en el habla (REH) y sus aplicaciones. Finalmente se mencionarán los contenidos a desarrollar en los capítulos subsiguientes.

La señal de la voz. Síntesis y características

La voz es producida mediante el aparato fonador constituido por: pulmones, laringe y el tracto vocal, donde éste último comprende las cavidades supraglótica, faríngea, oral y nasal. Dentro de la laringe se encuentran las cuerdas vocales, y la abertura entre ambas se denomina glotis. Por otro lado, las cavidades que componen el tracto vocal poseen los llamados órganos articulatorios, que se pueden dividir en activos (lengua, mandíbula, paladar blando y labios) y pasivos (dientes, paladar duro y maxilar superior). Algunas de estas estructuras pueden observarse en la Figura 2.

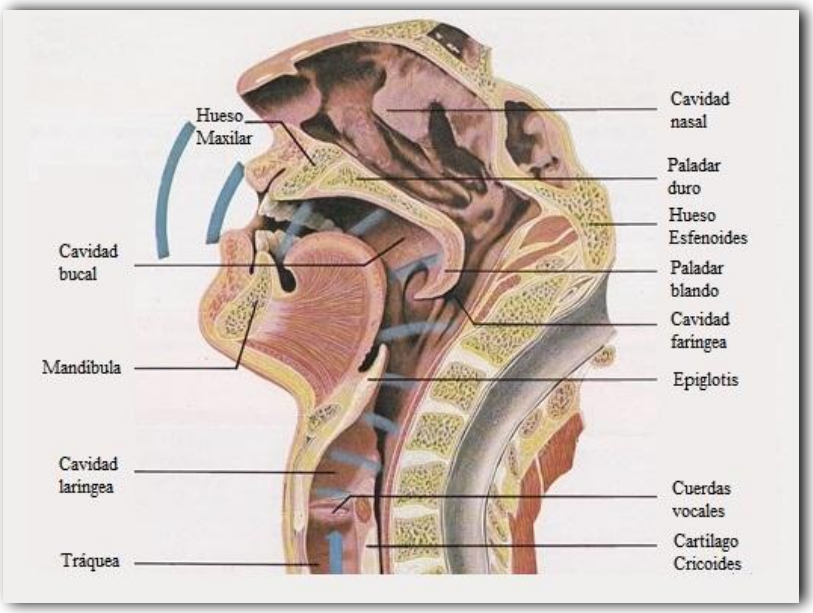

Figura 2 - Corte sagital. Estructuras y órganos del aparato fonador. Extraído del sitio web: *http://gentenatural.com/medicina/sintomas/disfonia-anatomia.html.*

La emisión de un sonido comienza mediante un flujo de aire proporcionado por los pulmones que al pasar por la laringe causa la vibración de las cuerdas vocales y el aire experimenta una turbulencia. Pero antes de que el sonido sea emitido el aire debe atravesar los órganos articulatorios los cuales actúan como filtros que modelan la señal de voz aportando componentes frecuenciales acorde a la resonancia que se experimenta en cada estructura [Rufiner, 2009][Guzman, 2010].

La manera más simple de clasificar los sucesos que se dan en las continuas transiciones temporales de la señal de voz es de acuerdo a como se disponen las cuerdas vocales, así un sonido puede ser *tonal* o *no tonal.* El primero de ellos se produce cuando las cuerdas vocales están tensadas y oscilan ante el flujo aire que pasa a través de la glotis, originando una serie de pulsos de presión cuasi-periódicos, la frecuencia de esos pulsos se denomina frecuencia fundamental. En los sonidos no tonales las cuerdas vocales se encuentran abiertas pero estrechando el conducto vocal lo que ocasiona turbulencia del flujo de aire. En la Figura 3 puede verse la forma típica de la señal de estos sonidos. Se puede incluir en la clasificación a otro tipo de suceso, el silencio, que si bien no es un sonido la combinación de sonidos y silencios conforman la señal final.

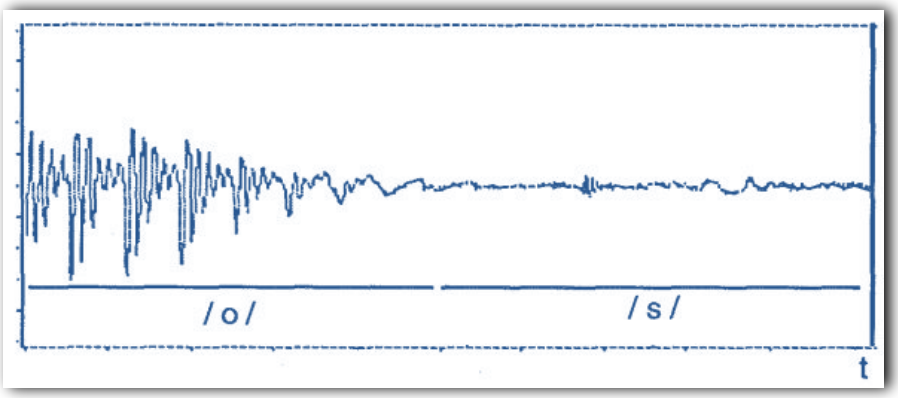

Figura 3 - Representación temporal de un sonido tonal /o/ y uno no tonal /s/ al pronunciar la palabra "dos". (Adaptada con permiso de [Pericas, 1993])

Cada una de las unidades básicas con las cuales se puede representar una lengua se denomina *fonema,* estrictamente son modelos teóricos cuya combinación de dos o más permite representar cada una de las palabras de una lengua. Cabe aclarar que un fonema es una abstracción mental de los sonidos del habla, mientras que la realización acústica del mismo es lo que se denomina fono o sonido y puede variar dependiendo del individuo e incluso ser diferentes para un mismo individuo. A su vez los fonemas se pueden clasificar en *vocálicos* y *consonánticos* dependiendo de las características acústicas y los gestos articulatorios que los producen. Cada palabra no es sólo representada por una serie de fonemas y silencios, sino que se dan transiciones entre cada uno de ellos que son de naturaleza continua y responden al cambio en las configuraciones de los órganos del aparato fonador.

Para un primer análisis de la señal del habla tomaremos el caso de las vocales por ser el más sencillo. Como se mencionó anteriormente, las distintas secciones del tracto vocal da lugar a resonancias que pueden identificarse claramente en la señal de voz como picos de energía para ciertas frecuencias, éstos son denominados *formantes*. De este modo, los valores formánticos distintos es lo que permite diferenciar perceptualmente una vocal de otra.

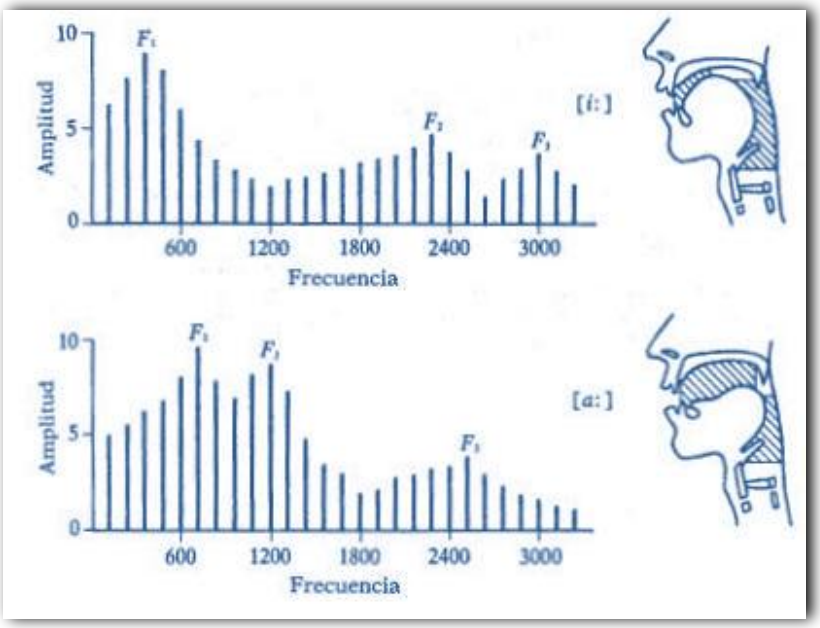

Figura 4 - Espectro de frecuencias de las vocales "i" (arriba) y "a" (abajo) donde se indican los formantes principales. Los pulsos son los originados en la glotis y corresponden a la frecuencia fundamental F_0. (Adaptada con permiso de [Llamazares, 2012])

La forma de identificar cada formante es mediante F_1, F_2, etc., asignando el valor de F_0 para la frecuencia fundamental. [Rufiner, 2009] [Navarro Mesa]. Como lo muestra la Figura 4, la representación en el dominio de la frecuencia permite distinguir características de la señal que en el dominio temporal están "ocultas" o son difíciles de apreciar.

Análisis e identificación de las emociones

Sabemos por nuestra experiencia cotidiana que las emociones están relacionadas con diferentes sistemas de nuestro cuerpo sobre los cuales producen alteraciones temporales. Desde un punto de vista fisiológico, originan respuestas en: el sistema muscular (dando lugar a las expresiones faciales y otros gestos corporales), la voz, actividad del Sistema Nervioso Autónomo y Sistema Endócrino. Por otro lado, dan lugar a cambios conductuales que nos llevan a reaccionar de diferente manera frente al entorno (personas, objetos, ideas, etc). El mecanismo es tan complejo que no se ha llegado a una definición rigurosa. El Diccionario de la Lengua Española (DRAE) en su 22ª Edición (2001) define emoción como: "*alteración del ánimo intensa y pasajera, agradable o penosa, que va acompañada de cierta conmoción somática*". Por su parte, la Asociación Estadounidense de Psicología (APA, American Psychological Association) establece que una emoción es un "*episodio reltivamente breve de respuestas evaluativas fisiológicas, conductuales y subjetivas sincronizadas*".

Existen múltiples teorías y autores que abordan el tema de las emociones coincidiendo en ciertos aspectos y confrontando en otros. A continuación se presentan las diferentes posturas y el enfoque adoptado por cada una de ellas. [Chóliz, 2005]

- ⇒ **Evolucionista:** sostiene que las emociones desempeñan una función adaptativa, tanto como facilitadoras de la respuesta apropiada ante las exigencias ambientales, como inductoras de la expresión de la reacción afectiva a otros individuos. Uno de los postulados principales de esta orientación es el de la existencia de emociones básicas, universales e innatas, necesarias para la supervivencia y que derivan de reacciones similares en los animales inferiores. El resto de emociones ("emociones derivadas") se generan por combinaciones específicas de aquéllas. Los autores que desarrollaron esta teoría son C. Darwin, P. Ekman y R. Plutchik.
- ⇒ **Psicofisiológica:** propone que la emoción aparece como consecuencia de la percepción de los cambios fisiológicos producidos por un determinado evento. En el caso de que no existan tales percepciones somáticas la consecuencia principal sería la ausencia de cualquier reacción afectiva. Además presupone que cada reacción emocional se podría identificar por un patrón fisiológico diferenciado (hipótesis de James-Lange). Este enfoque fue iniciado por James en 1884.
- ⇒ **Neurológica:** iniciada por Canon en 1931, en contraposición a la postura Psicofisiológica, plantea que las reacciones fisiológicas y viscerales no definen la cualidad de la reacción emocional, sino en todo caso la intensidad de la misma. Por lo tanto, lo verdaderamente relevante en la génesis de la emoción es la actividad del sistema nervioso central, actuando sobre la corteza para originar la experiencia cualitativa de la emoción, y sobre el sistema nervioso periférico.
- ⇒ **Conductista:** propuesta por J.B. Watson a partir de 1920, plantea que una emoción es un patrón de reacción heredado que contiene profundos cambios en los mecanismos corporales, sobre todo en el sistema límbico. Dicho patrón se manifiesta en niños pero con la edad se hace cada vez menos notorio. Este planteamiento fue modificado por B.F. Skiner y propuso que si bien las respuestas afectivas se basan en la fisiología y el estímulo externo, no considera los cambios fisiológicos como esencia de la emoción. Aunque existen otros análisis, todos identifican la emoción a partir exclusivamente de las manifestaciones externas (conductuales).
- ⇒ **Teoría de la activación general:** Lindsley (1951) argumenta que existe un único estado de activación general que caracterizaría a todas las emociones y que las

diferencias entre unas y otras sería cuestión de grado. Es decir, la activación es un continuo con diferentes grados de actividad que puede ser medido a través de las respuestas somáticas.

⇒ **Cognitiva:** algunos autores (Marañón, 1924; Schachter y Singer, 1962; Mandler, 1975) expresan que la emoción es causada por dos sucesos consecutivos, en primer lugar se produce una reacción de tipo fisiológico (activación) ante un estímulo y luego una interpretación cognitiva es la que determina la cualidad de la emoción. Posteriormente, Arnold (1960) señala que incluso antes de la activación ya se produce una evaluación global del estímulo como bueno o malo. Desde entonces, los autores que defienden esta teoría fueron progresivamente restando importancia a la respuesta fisiológica inicial y planteando que lo necesario para que se produzca una emoción son los procesos cognoscitivos implicados.

En este trabajo nos enfocaremos en la Teoría Evolucionista, que toma en cuenta las expresiones faciales (Figura 5 – Expresiones faciales que caracterizan las seis emociones básicas según la teoría Evolucionista. (Adaptado de http://www.grimace-project.net/)) para describir las emociones y que, como se mencionó antes, describe tres características muy importantes: la existencia de emociones básicas cuya combinación permite lograr el resto de los estados emocionales; considerar las emociones como innatas de lo que se deduce que cada uno de nosotros nace con la capacidad de expresarlas; su universalidad, posibilitando que personas de diferentes grupos étnicos y culturas puedan expresarlas (y reconocerlas) sin problemas.

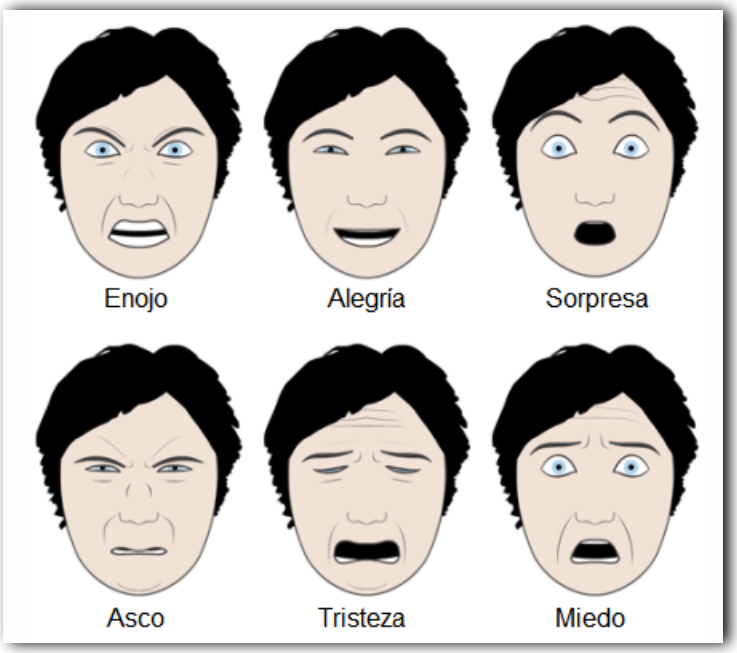

Figura 5 – Expresiones faciales que caracterizan las seis emociones básicas según la teoría Evolucionista. (Adaptado de http://www.grimace-project.net/)

Si bien uno reconoce y puede identificar los momentos en que se experiencian estas emociones, resulta difícil poder describirlas en forma precisa. En la Figura 6 – Experiencia subjetiva asociada a a cada una de las seis emociones básicas se han resumido cuales son las percepciones asociadas a cada emoción según [Choliz, 2005].

Enojo	Sensación de energía e impulsividad, necesidad de actuar de forma intensa e inmediata. Se experimenta como una experiencia aversiva, desagradable e intensa. Relacionada con impaciencia.
Alegría	Estado placentero, deseable, sensación de bienestar, auto-estima y autoconfianza.
Sorpresa	Aparece rápidamente y de duración momentánea hasta dar paso a una reacción emocional posterior. Mente en blanco momentáneamente. Sensación de incertidumbre por lo que va a acontecer.
Asco	Necesidad de evitación o alejamiento del estímulo. Si es oloroso o gustativo aparecen sensaciones gastrointestinales desagradables, tales como náusea.
Tristeza	Caracterizado por desánimo, melancolía, desaliento, pérdida de energía.
Miedo	Se trata de una de las emociones más intensas y desagradables. Genera aprensión, desasosiego , malestar, preocupación, recelo por la propia seguridad o por la salud, sensación de pérdida de control.

Figura 6 – Experiencia subjetiva asociada a a cada una de las seis emociones básicas

Hemos visto como es producida la voz y ciertas características de dicha señal. Se detallaron las posturas o perspectivas psicológicas respecto a las emociones, y sus características subjetivas. Esto nos da una base para comprender dos cuestiones claves para desarrollar un sistema de REH: las características que deben ser extraídas de la señal y que la técnica a usar para identificarlas. A continuación se hará una revisión de las aplicaciones en las que pueden implementarse dichos sistemas, se mencionarán cuales son las características de la señal de voz comunmente usadas y hará una revisión de las técnicas utilizadas en diferentes trabajos.

Técnicas aplicadas a la clasificación de emociones

Anteriormente se mencionó que la información contenida en un mensaje de voz (digamos una conversación) puede ser separada en dos partes: una explícita (verbal) y otra implícita (paralingüística). La parte explícita representa lo que se transmite por medio del lenguaje, es decir la semántica lingüística, que es independiente del emisor en cuestión. Por otro lado, la fracción implícita aporta información extra que enriquece el mensaje y está ligada íntimamente con particularidades del emisor como ser su sexo, edad y estado emocional [Ortego Resa, 2009].

Existen otras fuentes, además de la voz, que pueden proporcionar información, facilitando que un sistema pueda reconocer automáticamente las emociones, entre ellas se pueden

citar la imagen facial, gestos, posturas, frecuencia respiratoria y cardíaca [Vinhas, 2009]. Pero para ello se debe tener en cuenta una serie de limitaciones, como ser: hallar una base de datos que proporcione todas o algunas de esas señales, el manejo de gran cantidad de datos aumenta considerablemente el costo computacional, las aplicaciones en la vida real requerirían no solo de los dispositivos necesarios para registrar esas señales sino de la correcta conexión de los mismos al usuario.

Poder reconocer el estado de ánimo de un hablante utilizando sólo su señal de voz permite potenciar la capacidad de los sistemas de interacción humano-computadora basados en el habla, lo cual podría aplicarse a:

- ⇒ Sistemas semi-automáticos para diagnóstico de enfermedades psiquiátricas [Tacconi, 2008][Yuvaraj, 2012] [Dawel, 2012].
- ⇒ Reconocimiento de actitudes emocionales en niños mediante un software que permite diferentes escenarios de interacción visual-acústica con el usuario. [Yildirima, 2011]
- ⇒ Sistema telefónico de atención automática para emergencias que provea la asistencia adecuada y derivación a un operador humano con capacitación acorde a la situación en que se encuentra el usuario [Devillers, 2007].
- ⇒ *Call centers*, en este caso se trata de una interfaz humano-humano donde se monitorea conversaciones entre usuarios y agentes para llevar un control de calidad en la atención con el objetivo de lograr una comunicación agradable y la satisfacción del cliente [Devillers, 2006].
- ⇒ Optimizar los sistemas de síntesis de voz, aportándole información que permita la expresión de emociones para lograr una voz más natural [Murray, 2008].

El núcleo o idea fundamental detrás de estas aplicaciones es un sistema de clasificación de emociones a partir de la señal de voz que consta de dos etapas (aunque pueden realizarse estructuras jerárquicas con varios niveles de clasificación y por ende mas etapas). En la Figura 7 se representan en un diagrama de bloques estas dos etapas: extracción de características y clasificación. Ambas revisten la misma importancia ya que por un lado se deben seleccionar las características que mejor discriminen las emociones y para que luego un clasificador identifique esos patrones y así clasificarlas.

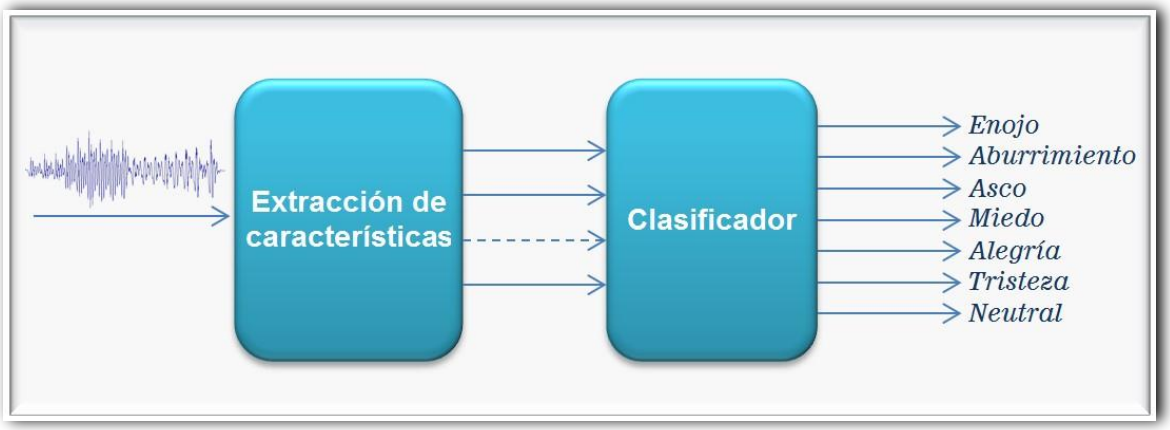

Figura 7 – Estructura básica de un sistema de REH de 7 emociones (Adapatado con permiso de [Albornoz-Rufiner, 2011]).

Con respecto a la primer etapa, mencionaremos cuáles son las características mas usadas (que serán tratadas luego en el Capítulo 2), entre ellas se encuentran: parámetros prosódicos como la frecuencia fundamental (F_0) y la Energía (E), de las cuales se toma su media y desvío estándar; y características espectrales como los coeficientes cesptrales en escala Mel (MFCC, del inglés Mel Frequency Cesptral Coefficients), la media del logaritmo del espectro (MLS, del inglés Mean Log Spectrum), coeficientes de predicción lineal (LPC, del inglés Linear Prediction Coefficients), coeficientes cepstrales de predicción lineal (LPCC,

del inglés Linear Prediction Cesptral Coefficients) y coeficientes de potencia en frecuencia logarítmica (LFPC, del inglés Log Frequency Power Coefficients).

Por otro lado, se han usado varios tipos de clasificadores en la bibliografía debido a que no existe un acuerdo en cual resulta mejor para la clasificación de emociones sino que cada uno posee sus ventajas y desventajas. Podemos citar:

- ⇒ Modelos ocultos de Markov (HMM, del inglés Hidden Markov Models), su uso se encuentra muy difundido en el REH dado que se utiliza en casi todas las aplicaciones de procesamiento del habla por su relación con el mecanismo de producción de la señal de voz [RABINER, 1993]. Consituye un modelo estadístico cuyo objetivo es determinar los parámetros desconocidos de un proceso (cadena) de Markov al que se esta modelando, y en el caso del REH ese proceso corresponde a la secuencia de características extraídas de la señal de voz.
- ⇒ Modelo de mezcla de Gaussianas (GMM, del inglés Gaussian Mixture Model), tienen como finalidad modelar un conjunto de variables continuas, donde cada una de ellas (se supone) tiene una distribución Gaussiana. Pueden considerarse como un caso especial de un HMM continuo de un solo estado [REYNOLDS, 1995].
- ⇒ Máquinas de soporte vectorial (SVM, del inglés Support Vector Machines), son un tipo de clasificador lineal cuyo objetivo es encontrar un hiperplano mediante el cual queden separados los patrones de entrada en un espacio cuyas dimensiones sean lo suficientemente altas para lograr dicha separa Cada patrón etiquetado es clasificado binariamente, por lo que el método de aprendizaje es supervisado [Tesis Marcelo, 2011]
- ⇒ Redes Neuronales Artificiales (RNA), se basan en modelizar el funcionamiento de una red neuronal biológica. Existen diferentes arquitecturas y aplicaciones, la mayoría asociada al reconocimiento de patrones. En este trabajo se utilizarán RNAs como método de clasificación y por lo tanto serán desarrolladas con mayor detalle mas abajo.

En la bibliografía se reportan resultados de experimentos que utilizan una característica particular extraída de la señal o una combinación de dos o más. Así mismo dichas características se prueban con uno o mas tipos de clasificadores. En [FOO, 2003] se utilizó un sistema basado en HMM para la clasificación de seis emociones de dos corpus propios (uno en Mandarín y otro en Birmano). Se emplearon MFCC, LFPC y LPCC que arrojaron una tasa de clasificación de 59,0%, 77,1%, 56,1%, respectivamente (promedio entre ambos corpus). Lee *et al* [lee, 2004] utilizaron HMMs para la clasificación de cuatro emociones y obtuvieron mejores resultados para HMMs basados en fonemas que en características prosódicas globales. Los GMM han sido experimentados por [SCHULLER, 2002] con el corpus FERMUS III (seis emociones básicas mas un estado neutral) con los que se obtuvo un 74,83% de clasificaciones correctas para reconocimiento independiente del hablante y fue de 89,12% para el caso dependiente del hablante. Los GMMs se han empleado con bases de datos conteniendo emociones distintas de las “seis básicas”, por ejemplo [SLANEY, 2003] utilizó el corpus BabyEars de tres emociones (aprobación, atención y prohibición) con el que obtuvieron un desempeño del clasificador del 75%.

Se dedicará un párrafo aparte para el caso de los trabajos en que fueron empleadas RNAs como herramienta de clasificación ya que servirá como punto de comparación de los resultados presentados luego en este trabajo. Algunas de las ventajas que presentan las RNAs, frente a HMMs y GMMs, en el reconocimiento de patrones es que se adaptan mejor a las “no linealidades” y su desempeño es mejor cuando se emplea una cantidad realtivamente baja de datos (patrones) de entrenamiento [AYADI]. El tipo de RNA al que mas se refiere dentro del área del REH es el Perceptrón Multicapa o PMC (generalmente con una sola capa oculta) debido a que su implementación y entrenamiento no resulta complicada pero los resultados obtenidos no son mejores frente a otros clasificadores como los vistos anteriormente. En [Hozjan, 2003] se utilizaron cuatro estructuras diferentes de PMC (con 26

neuronas en su capa oculta) con un corpus que contenía las seis emociones básicas. Los resultados que obtuvieron fueron de sólo 51,19% para el reconocimiento dependiente del hablante. Otros tres tipos de estructuras de PMC se usaron en [Petrushin, 2000] obteniendo un promedio de clasificación de 65% pero fue usada otra base de datos con cinco emociones. En [Truong y Leeuwen, 2007] distinguieron entre risas y voz por medio de clasificadores basados en fusión de GMM, SVM y PMC cuyo desempeño fue mejor cuando, ademas de caracaterísticas espectrales, se adicionaron características prosódicas. Más recientemente [Albornoz-Rufiner, 2011] utilizaron GMM, PMC y HMM para clasificar siete emociones de un corpus. Además, se propuso la combinación de ellos para diseñar clasificadores jerárquicos cuyo objetivo era separar las siete emociones en dos grupos, y luego discriminarlas individualmente. Se utilizaron características prosódicas (E y F_0) y espectrales (MFCC y MLS). El desempeño de los clasificadores individuales GMM, PMC y HMM fue de 63,49%, 66,83% y 68,57% respectivamente. Mediante un clasificador jerárquico se mejoró el desempeño logrando un 71,75%.

La motivación que lleva al uso de redes neuronales en los distintos campos de aplicación es el hecho que hay tareas en que las máquinas son más rápidas y eficientes que los seres humanos. Esto se debe principalmente a que el uso de reglas simbólicas no es un proceso natural al que el cerebro se encuentra habituado. Otro factor influyente, y limitante para los seres humanos, es el manejo de grandes volúmenes de información que para el caso de computadoras esto se reduce a aumentar la velocidad de cómputo o el tiempo que se debe estar ejecutando el proceso. Sin embargo el cerebro posee algunas características que lo hacen único: su gran capacidad de procesamiento paralelo, la tolerancia a fallas o daños, su inteligencia, razonamiento y conciencia. Éstas y algunas otras similitudes y diferencias se muestran en la Tabla 1.

	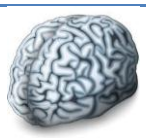	
Cant. de unid. de procesam.	10^{14} sinapsis	10^{8} transistores (por núcleo)
Tamaño de cada elemento	~µm	~nm
Potencia consumida	30 W	50 W (CPU)
Velocidad de procesamiento	100 Hz	GHz
Tipo de procesamiento	Paralelo Distribuido	Serial
Tolerancia a fallas	SI	NO
Inteligencia, razonamiento	SI	NO
Aprendizaje	SI	?

Tabla 1 – Comparativa de algunas características entre un cerebro humano y un microprocesador. (Extraído de [Orr, 1999])

Algunos de las tipos de aplicaciones de las redes neuronales son:

⇒ Modelización de distintos niveles de razonamiento: síntesis y reconocimiento de voz, visión, reconocimiento de emociones a través de audio y/o video, solución de problemas, entre otros.

⇒ Aprendizaje automático: encontrar la solución óptima a ciertos problemas (minimización de función de error), clasificación de patrones, mapeo de funciones en problemas de regresión, uso de memoria asociativa en motores de búsqueda de datos.

Estas capacidades de las redes neuronales permiten su utilización concreta en tareas complejas como: reconocimiento de patrones (en procesamiento de imágenes y voz), identificación de las fuentes a partir de una mezcla de señales originadas por ellas, robótica (navegación, visión, automatismo en industrias), software para reconocimiento de voz y lectura de texto en voz alta, detección de rostros, compresión de datos, predicción de comportamiento humano, meteorología, entre otras.

El desarrollo de teorías desde el campo de la neurociencia cognitiva, referidas a la forma en que el cerebro interpreta y procesa la información, expresado por ejemplo en [Elman et.al., 1996], ha llevado a una concepción diferente en inteligencia artificial que se denomina Aprendizaje Profundo o Deep Learning (en inglés). Mediante este enfoque, se pretende que una red de creencia profunda o deep belief network (en inglés) constituida por dos o más capas ocultas, aprenda altos niveles de representación en sus capas más profundas a partir de entradas constituidas por información de bajo nivel. De este modo queda representado en cada capa un conjunto de características o patrones diferente [Hinton, 2007]. Estas arquitecturas profundas son necesarias para poder imitar los procesos cognitivos superiores como la visión y el habla [Bengio, 2009].

El algoritmo de entrenamiento más usado es el de retropropagación, sin embargo cuando se aplica a redes profundas pierde eficiencia y las soluciones a las que se llegan son pobres. Una posible solución a este inconveniente fue dada por Hinton et.al. [Hinton et.al., 2006] que propuso un nuevo método consistente en realizar un pre-entrenamiento con el objetivo de inicializar los pesos de la red cerca de una solución óptima. Desde entonces se han publicado numerosos artículos que implementan este método. En [Mohamed et.al., 2012] fueron utilizadas redes profundas para predecir los estados de HMMs para el reconocimiento de fonemas de un corpus. Una revisión de técnicas similares a ésta se realizó en [Hinton, 2012] aplicadas a reconocimiento de voz. Brückner y Schuller [Brückner y Schuller, 2012] utilizando la señal de voz para la clasificación de diferentes tipos de personalidades por medio de una red profunda constituida por (una pila de) Máquinas de Boltzmann Restringidas (RBMs, del inglés Restricted Boltzmann Machines).

En este trabajo se propone un método inspirado en [Hinton et.al., 2006] para entrenar redes profundas y obtener así mejores resultados de generalización [Bengio, 2009]. El Capítulo 2 abordará el marco teórico general necesario para comprender el desarrollo subsecuente y se explicarán los materiales utilizados. En el Capítulo 3 se describirá el método propuesto. Los experimentos realizados y resultados obtenidos se expondrán en el Capítulo 4. Luego, en el Capítulo 5 se discutirán cuestiones sobre el método y resultados para elaborar las conclusiones del trabajo. Finalmente, en el Capítulo 6 se realizará un análisis económico.

Capítulo 2. Marco teórico y materiales

Los sistemas de REH, según se vio anteriormente, están formados por dos bloques o etapas. En la la primera de ellas se toma la señal de voz y mediante un adecuado procesamiento se extraen características que permiten representar y discriminar las emociones de mejor manera que si tomáramos los valores de la señal en crudo. Esta nueva representación sirve para facilitar la tarea en la segunda etapa cuyo objetivo será reconocer patrones dentro de ese conjunto de características y de este modo determinar a qué emoción corresponde. Una de las herramientas empleadas para realizar esa clasificación son las RNA y es el método que se ha optado en este trabajo. En el presente capítulo se abordarán los temas recién menciondos y que ayudará comprender un sistema de REH basado en RNA. Primeramente se introducirá el concepto de neurona y sus características para luego describir lo que es una RNA. Se verá en que consiste el entrenamiento de una red y se explicará el algoritmo mas utilizado para ello: *retropropagación del error*. Posteriormente se analizarán las características empleadas generalmente para el REH y por último será explicada la base de datos utilizada.

Neurona

Las RNA son una abstracción simplificada de las redes neuronales biológicas y su objetivo es imitarlas en aquellos aspectos que sean relevantes para reproducir su funcionamiento mediante una implementación computacional. De aquí en más se utilizarán indistintamente los términos red o red neuronal para referirnos a las RNA. Si bien no existe una definición universalmente aceptada, según la Agencia de Investigación de Proyectos Avanzados de Defensa (en inglés DARPA, Defense Advanced Research Projects Agency) “una red neuronal es un sistema compuesto por muchos elementos de procesamiento simple que operan en paralelo cuya función es determinada por la estructura de la red, la intensidad de las conexiones y el procesamiento llevado a cabo en elementos de cómputo o nodos”. [DARPA, 1998]

La unidad elemental de procesamiento y constitutiva de una red neuronal es precisamente la neurona, nodo o “unit” (en inglés) según la literatura, términos que se usarán indistintamente en este trabajo. En la Figura 8 se muestra un modelo muy simple de una neurona compuesta por un conjunto de *j* entradas y sobre las cuales se aplica una serie de funciones para producir el valor de salida.

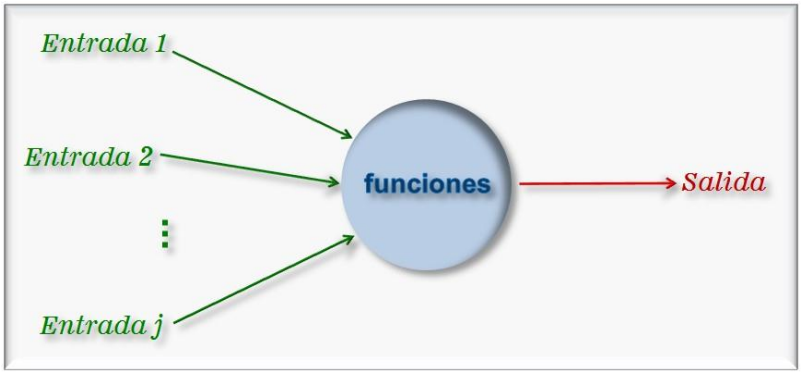

Figura 8 – Modelo simplificado de una neurona

Según la ubicación de una neurona en la red, cada entrada puede provenir de la salida de otra neurona o bien constituir una de las entradas de la red. Para cada configuración de

entradas se producirá un único valor de salida. Del mismo modo, dependiendo de dónde esté situada la neurona, ese valor será una salida de la red o en cambio originarse muchas instancias de la misma, es decir muchas salidas pero con el mismo valor, que actuarán de entradas de otras neuronas.

Mediante un modelo desarrollado, como se muestra en la Figura 9, se explicarán con mayor profundidad las características de una neurona utilizando notación apropiada. Los elementos que aparecen en dicha figura se referencian en la Tabla 2.

Tomamos arbitrariamente una neurona i a la cual ingresan n entradas provenientes, cada una de las salidas de la j-ésima neurona. Esas entradas tienen asociado un peso w_{ij}, que se encuentra entre la neurona j y la actual neurona i. Existe además una entrada adicional llamada **bias** que por convención se denomina x_0, cuyo valor es siempre igual a uno y su peso correspondiente w_0. Esto permite potenciar la capacidad de la red, como se analizará luego, ya que adiciona un término independiente de las demás entradas.

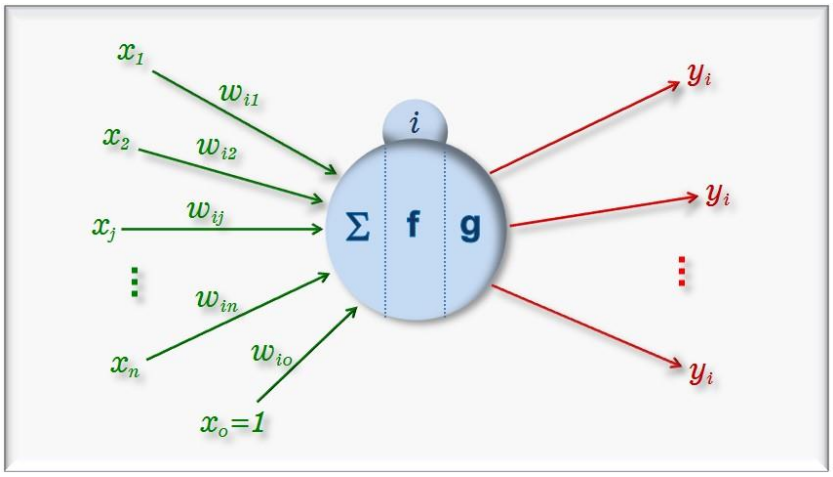

Figura 9 – Modelo desarrollado de una neurona

Tabla 2 – Referencias correspondientes a la Figura 9

i	nombre de la neurona
x_1, x_2, x_j, x_n	entradas
$w_{i1}, w_{i2}, w_{ij}, w_{in}$	pesos de cada entrada
$x_{io}=1$	bias
y_i	salidas
Σ	función de propagación
f	función de activación
g	función de salida

Para obtener el valor de salida y_i, hay tres funciones que se deben aplicar sucesivamente sobre las entradas. Cada una de ellas puede ser de variada morfologías otorgándole de esta manera diferentes propiedades a la red, se elección dependerá de la aplicación y tipos de señales usadas.

Función de propagación o excitación

Es la primera en aplicarse y consiste típicamente en una sumatoria de los productos entre el valor de cada entrada y su peso, incluyendo el bias. Se obtiene como resultado la *red de entrada* de la neurona net_i (red, en inglés) con el subíndice i que refiere al nombre de la neurona:

$$net_i = \sum_{j=1}^{n} w_{ij}x_j + w_{i0}x_0 = \sum_j w_{ij}x_j \quad (1)$$

Ahora, si consideramos un vector de pesos $\mathbf{w_i}$=(w_{i0},w_{i1},w_{i2},…,w_{in}) y el vector de entradas **x**=(x_{i0},x_{i1},x_{i2},…,x_{in}) para la neurona i, se puede expresar (1) en forma más compacta como un producto interno:

$$\sum_j w_{ij}x_j = \boldsymbol{w_i.x} \quad (2)$$

Además, cada entrada x_j es a su vez la salida y_j de la j-ésima neurona, haciendo **y**=(y_1, y_2,…, y_n) podemos escribir:

$$net_i = \sum_j w_{ij}x_j = \sum_j w_{ij}y_j = \boldsymbol{w_i.y} \quad (3)$$

Los pesos representan la intensidad de la conexión entre dos neuronas y pueden tomar cualquier valor real. Al igual que las sinapsis en el tejido biológico, son las variables que se irán modificando durante el entrenamiento, permitiendo que la red aprenda. Se habla de conexiones **excitatorias** cuando $w_{ij}>0$ e **inhibitorias** cuando $w_{ij}<0$. El caso $w_{ij}=0$ indica que no existe conexión, aunque esto no es permanente ya que dicho valor puede cambiar durante el entrenamiento.

Función de activación o transferencia

Es una función f que se aplica al resultado anterior para obtener el estado de activación a_i de una neurona:

$$a_i = f\left(\sum_j w_{ij}y_j\right) = f(net_i) \quad (4)$$

Función escalón. Esta función tiene la particularidad de que imita el potencial todo-nada de una neurona real. En el modelo esto equivale a decir que a_i pasará de un estado de reposo $a_i=0$ a un estado despolarizado o activado $a_i=1$ cuando net_i sea mayor o igual a cero (Figura 10.A). Su expresión es:

$$a_i = \begin{cases} 0 & si\ \ net_i < 0 \\ 1 & si\ \ net_i \geq 0 \end{cases}$$

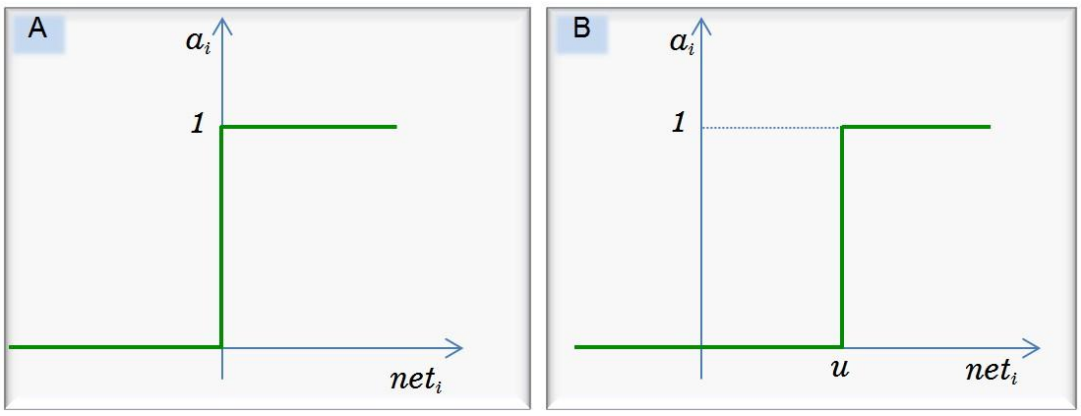

Figura 10 – A. Función de activación tipo escalón. B. Función escalón desplazada

Dado este comportamiento, las neuronas con esta función de activación son denominadas *binarias*.

Si desglosamos la expresión anterior y consideramos $x_0=1$ obtenemos:

$$a_i = f\left(\sum_{k=1}^{j} w_{ik}x_k + w_0x_0\right) = f\left(\sum_{k=1}^{j} w_{ik}x_k + w_0\right) \quad (5)$$

Analizando para un valor arbitrario $w_0<0$ y haciendo $u=-w_0$, la neurona pasará al estado activado recién cuando el primer término de (5) sea igual o mayor a u. Por lo tanto, la gráfica de la función estará desplazada hacia la derecha y u puede interpretarse como el umbral de despolarización, como ocurre en una neurona biológica (Figura 10.B).

Dado que $w_0 \in \Re$, el hecho de adicionar el bias a la neurona agrega un grado de libertad que permite variantes de la misma función (desplazamiento en este caso), otorgándole versatilidad a la neurona para originar la salida deseada. Una variante es la *función escalón simétrica*, en la cual a_i puede tomar los valores *-1* para el estado desactivado o *1* para el caso contrario (Figura 11).

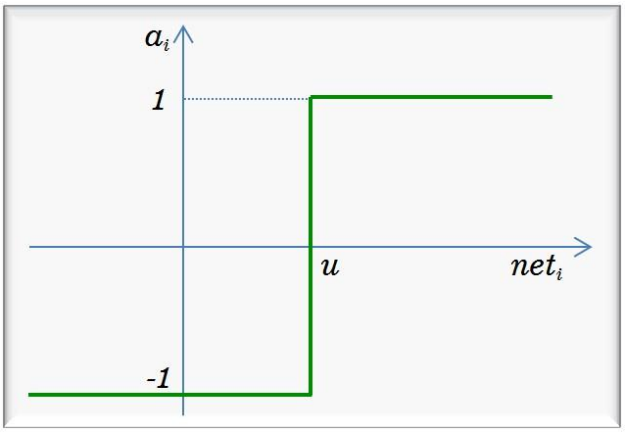

Figura 11 – Función escalón simétrica

Función identidad. Es una función que devuelve su propio argumento, para nuestro caso:

$$a_i = f(net_i) = net_i \quad (6)$$

El valor de activación de la neurona será, por lo tanto, idéntico a su red de entrada. También puede pensarse como una función lineal de pendiente positiva unitaria. Haciendo un análisis similar al que se hizo para la función escalón en (5), obtenemos:

$$a_i = \sum_{k=1}^{j} w_{ik}x_k + w_0 \quad (7)$$

Considerando arbitrariamente $w_0<0$ y $u=-w_0$ se puede ver que el término originado por el bias produce un desplazamiento (a la derecha en este caso) de la gráfica, tal como se observa en la Figura 12.

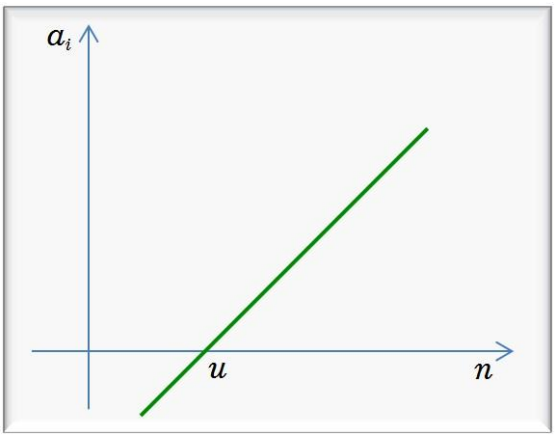

Figura 12 – Función identidad

Función logística estándar. Es una función, cuya gráfica presenta forma de "S", está definida por la fórmula:

$$a_i = \frac{1}{1-e^{-net_i}} \tag{8}$$

Al igual que en los casos anteriores, la gráfica se encuentra desplazada debido al bias (Figura 13). Mientras que net_i puede tomar valores teóricamente desde -∞ a ∞, los valores resultantes para a_i están acotados entre 0 y 1, como sucede en la función escalón, con la diferencia que aquí la transición entre estos valores es continua y suave. Esto permite que la función sea diferenciable, requisito esencial para implementar uno de los algoritmos de aprendizaje como es la *retropropagación.*

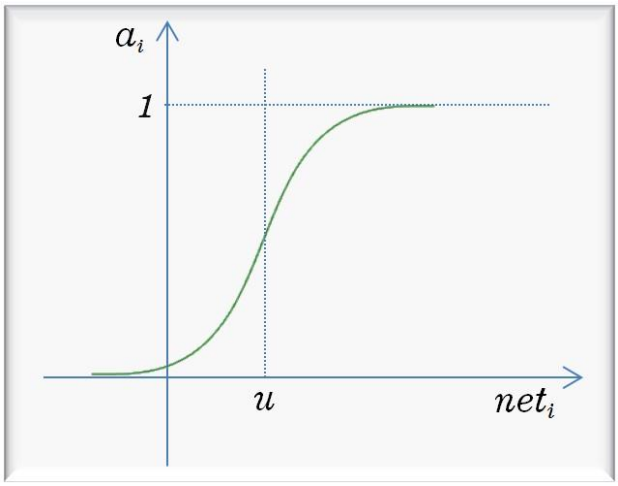

Figura 13 – Función logística estándar

Función tangente hiperbólica

De comportamiento similar a la función identidad pero de rango (-1, 1), definida como la relación entre el seno hiperbólico y el coseno hiperbólico, o de forma equivalente:

$$a_i = tanh\, net_i = \frac{senh\, net_i}{cosh\, net_i} = \frac{e^{net_i}-e^{-net_i}}{e^{net_i}+e^{-net_i}} \tag{9}$$

Su gráfica puede verse en la Figura 14 (considerando, como antes, el desplazamiento u producido por el bias). Al igual que la función logística, también es diferenciable y la elección de una u otra dependerá de la aplicación en particular.

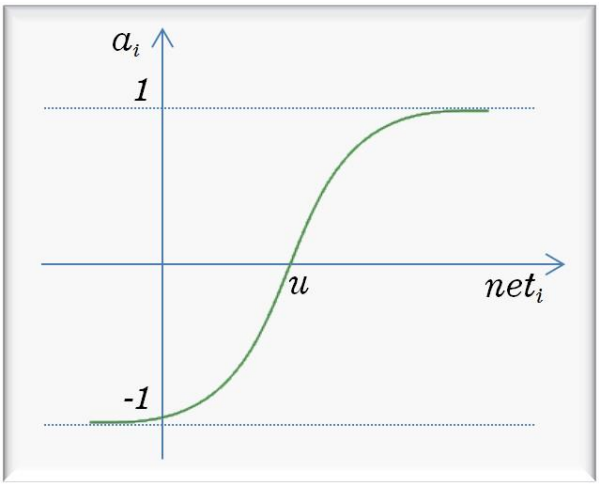

Figura 14 – Tangente hiperbólica

Función de salida

Es la encargada de producir la salida y_i de la neurona a partir de su estado de activación a_i. Si bien existen otras, las funciones más usadas son las mismas que para la función de activación, pero no se debe tener en cuenta el desplazamiento u. Por ejemplo, en una neurona con función de salida identidad será $y_i = a_i$.

Generalidades sobre redes neuronales

Existen diferentes formas de interconectar neuronas y disponerlas en una red, en base a esta topología puede clasificárselas. Sin embargo, aquí sólo nos referiremos a la estructura más usada que consiste en disponer las neuronas en capas. Así, se distinguirán una *capa de entrada*, *capas ocultas*, y una *capa de salida,* cada una de ellas compuestas por neuronas llamadas de entrada, ocultas y de salida, respectivamente. Solo se considerarán conexiones entre neuronas de capas adyacentes y no entre nodos de una misma capa. Dichas conexiones son unidireccionales con sentido entrada-ocultas-salida, por lo que estas redes son llamadas de *pre-alimentación* o *feedforward* (en inglés). Además, cada nodo está conectado con todos los de la capa siguiente, denominándose redes *totalmente conectadas*.

Lo explicado anteriormente se puede ver de forma más clara en la Figura 15, donde se muestra como ejemplo una red con su capa de entrada de dos nodos, una única capa oculta con tres nodos y la capa de salida con un nodo (para mayor claridad se han omitido los pesos y nombres de neuronas). Por simplicidad, los nodos de bias se han resumido en uno para toda la capa, esto es equivalente a uno para cada neurona dado que su estado es siempre uno (activado) y solo influye el peso de la conexión. El sentido unidireccional de las conexiones es graficado por medio de flechas; la salida de una neurona "apunta" a la entrada de la neurona que alimenta. Como se puede notar, las entradas de la red son nodos que no poseen entradas ni bias. El conjunto de valores que servirán de entrada a la red, llamado *vector o patrón de entrada,* consta de tantos elementos como nodos de entrada tiene la red (dos para el caso del ejemplo). Cada valor del vector será el estado de activación del nodo y también su salida, por lo tanto se pueden considerar como neuronas cuya función de salida

es la identidad. Por último, la salida de la red corresponderá con la de la única neurona de salida.

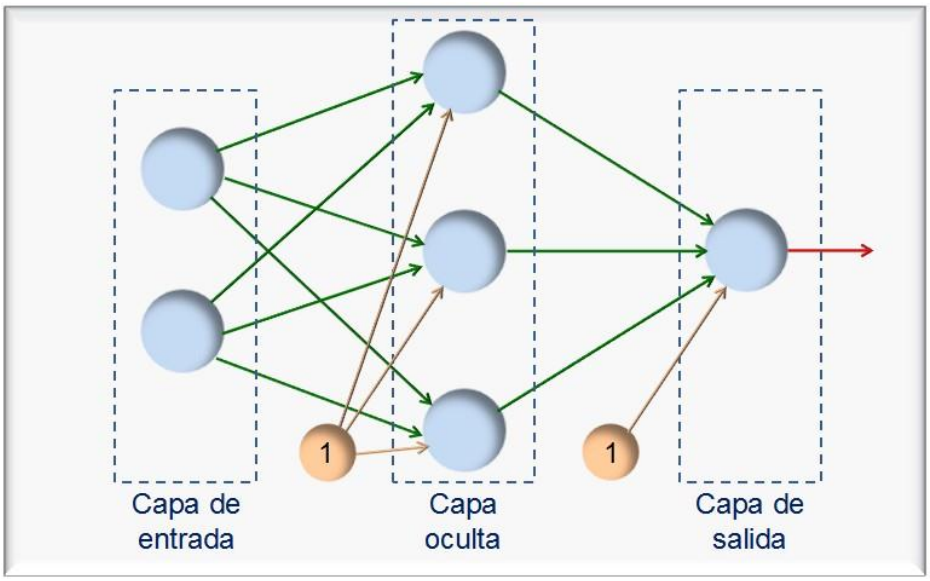

Figura 15 – Esquema de una red feedforward de dos capas, totalmente conectada compuesta de 2 nodos de entrada, 3 ocultos y 1 de salida.

Si bien se han nombrado tres capas, la red del ejemplo posee únicamente dos. Esto es debido a que la capa de entrada es solo representativa, y son las capas oculta y de salida las que reciben conexiones, con sus pesos asociados, que hacen posible el aprendizaje de la red. Visto de otro modo, existen solo dos conjuntos de pesos: el que está entre la capa de entrada y oculta, y el que se encuentra entre la capa oculta y de salida.

El *perceptrón* constituye la red neuronal más simple, fue introducido en 1958 por Frank Rosenblatt [Rosenblatt, 1958] con la idea de ilustrar burdamente algunas características del funcionamiento del sistema nervioso. Consistía en una red compuesta por nodos de entrada (cuya cantidad dependerá de la aplicación) y una única neurona de salida, es decir sin nodos ocultos. Posteriormente se dio el nombre de *perceptrón simple* a una red de una capa y *perceptrón multicapa* (PMC) a la que posee una o más capas ocultas.

Un perceptrón similar al de Rosenblatt es presentado en la Figura 16. Usando la nomenclatura indicada antes tenemos: las entradas x_1, x_2, x_0 (bias), sus pesos correspondientes w_1, w_2, w_0, la neurona i y su salida y_i. Consideramos una función de activación f del tipo escalón simétrico:

$$a_i = \begin{cases} -1 \;\; si \;\; net_i < 0 \\ \;\; 1 \;\; si \;\; net_i \geq 0 \end{cases}$$

El estado de activación -1 corresponde al valor booleano “falso”, y el estado 1 a “verdadero”. La función de salida g que será la identidad, por lo tanto la salida del perceptrón será idéntica a su estado de activación.

El perceptrón simple es básicamente un clasificador y dado que posee una única capa sólo es capaz de clasificar vectores de entrada que sean linealmente separables. Para solventar estas limitaciones se debe agregar al menos una capa oculta, conformando un PMC como el de la Figura 15. La adición de más capas ocultas potencia las capacidades de la red, permitiendo representaciones complejas como es el caso del habla, pero esto a su vez dificulta el correcto entrenamiento para lograr una solución adecuada.

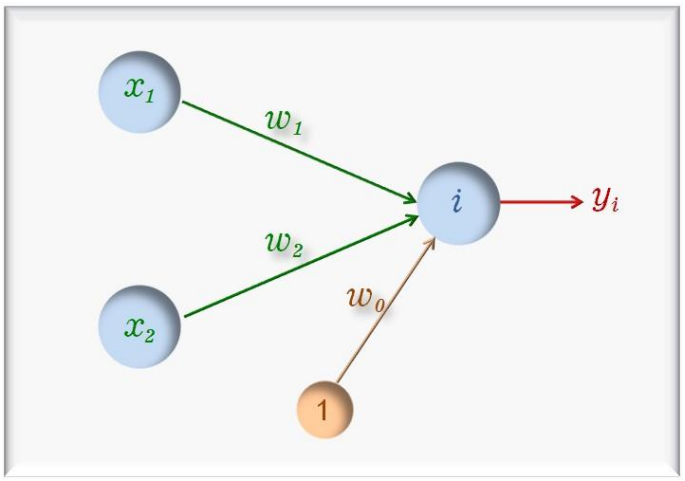

Figura 16 – Perceptrón simple con un solo nodo de salida

Entrenamiento y aprendizaje

Sin pretender dar una definición estricta, podemos decir que el aprendizaje, en sentido general, es un mecanismo de adaptación que genera modificaciones relativamente permanentes en la conducta y/o comportamiento mediante asociaciones entre estímulos del entorno y la respuesta generada. Para el caso de las redes neuronales podemos decir más precisamente que el aprendizaje se logra mediante la modificación de los pesos de las conexiones, a partir de un conjunto de datos que sirven de "entorno", para lograr el comportamiento requerido. Esto se lleva a cabo a través del *entrenamiento* durante el cual la actualización de pesos está asociada a minimizar una *función de costo* que permite estimar cuán lejos se está de la solución óptima. Existen tres paradigmas de aprendizaje:

- ⇒ **Supervisado:** su nombre indica que se requiere de un "maestro", esto es, la salida deseada que debe dar la red. Dado un subconjunto de pares ejemplo entrada-salida (x,y) donde $x \in X$, $y \in Y$, el objetivo es inferir una función de mapeo $M: X \rightarrow Y$ que se verifique para dichos ejemplos. En otras palabras, se pretende estimar la relación entrada-salida de todo el conjunto de datos a partir de un subconjunto del mismo. La función de costo está asociada a la discrepancia producida entre la salida de la red y la deseada. Este tipo de aprendizaje se aplica a tarea de reconocimiento de patrones (clasificación) y aproximación de funciones (regresión). Una de las restricciones para la aplicación de este paradigma es la necesidad de contar con información "rotulada" o "etiquetada" (labeled, en inglés), es decir que la salida y para cada x debe ser explicitada. Para muchas aplicaciones esta información no puede ser accedida fácilmente, además el proceso de rotular puede acarrear un costo elevado y ser engorroso ya que debe ser necesariamente llevado a cabo por seres humanos.
- ⇒ **Por refuerzo:** en este caso se tiene un conjunto X de datos de entrada con los que se pretende lograr un comportamiento específico de un sistema del entorno pero cuya dinámica se desconoce. Cada $x \in X$ sirve de entrada a la red, y su salida interactúa con el sistema que a su vez produce una respuesta. Si la respuesta es incorrecta se "penaliza" a la red. A partir de esto, el objetivo es minimizar la cantidad de futuras penalizaciones. Tiene sus aplicaciones en problemas de control, juegos y programación dinámica.
- ⇒ **No supervisado:** aquí se tiene un conjunto de datos X pero no existe maestro ni penalizaciones. El objetivo de la red es crear representaciones de X mediante la minimización de una función de costo que depende de las entradas $x \in X$ y salidas correspondientes. Las representaciones quedan virtualmente almacenadas en la red a través de los pesos en las conexiones. Esto tiene utilidad en tareas de clasificación, reducción de dimensionalidad, filtrado de señales y agrupamiento o clustering (en in-

glés). Al contrario de lo que ocurre en el aprendizaje supervisado, aquí no se requiere de datos rotulados para el entrenamiento.

⇒ **Semi-supervisado:** reciente paradigma que surge como combinación de los tipos supervisado y no supervisado. Su potencialidad radica en aprovechar la gran disponibilidad de información no rotulada para el entrenamiento, y mejorar el desempeño de la red a través de una menor cantidad datos rotulados.

Según sea el paradigma de aprendizaje que se utilice, existen diferentes algoritmos de entrenamiento que son los procedimientos matemáticos mediante los cuales los pesos se van ajustando automáticamente.

Algoritmo de Retropropagación

Dentro del paradigma de aprendizaje supervisado, uno de los algoritmos de entrenamiento más utilizado es el de *retropropagación del error* o simplemente *retropropagación*. El mismo se aplica a redes feedforward totalmente conectadas y tiene por objetivo minimizar una función de error que determina en cada iteración la discrepancia entre la salida deseada y la salida actual de la red. Esto se entiende fácilmente para la capa de salida, ya que los valores deseados son conocidos. Pero para poder aplicar este mismo razonamiento a las capas ocultas y a la de entrada lo que se hace es propagar hacia atrás (retropropagar) el error, calculándolo para cada una de las neuronas de las capas anteriores (más cerca de la entrada). Acorde a ese error son ajustados los pesos con el objetivo de minimizarlo, de modo que cuando se presente la misma entrada (o una de características similares) la salida producida por la red se asemeje a la deseada.

Acorde a lo anterior, la aplicación del algoritmo de retropropagación puede dividirse en dos fases. En la primer fase se presenta a la red el patrón de entrada que es propagado hacia adelante a través de todas las capas para obtener la salida de la red. La segunda fase comienza con el cálculo de error entre la salida de la red y la deseada, luego su retropropagación y ajuste de pesos. Ambas fases constituyen una iteración o época, y el proceso de entrenamiento se lleva a cabo para una cantidad establecida de épocas o hasta que el error en la salida sea menor a un valor prefijado.

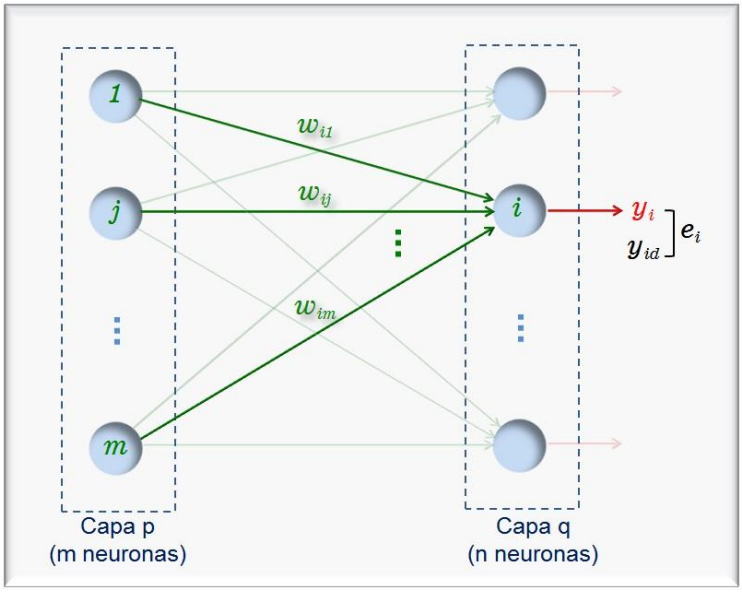

Figura 17 - Capas adyacentes de una red feedforward

A continuación se verán las fórmulas utilizadas para implementar el algoritmo. Tomaremos dos capas adyacentes de una red feedforward como se ve en la Figura 13. Una capa **p**

de **m** neuronas que será la que está más cerca de la entrada, y otra **q** de **n** neuronas más cerca de la salida. Referiremos genéricamente con i a las neuronas de la capa **q** y con j a las de la capa **p**. En cada época se presenta a la red un vector de entrada **x** y su correspondiente vector de salidas deseadas $\mathbf{y_d}$.

En la primera parte de esta fase la capa **q** corresponde a la de salida. Para una época dada, la red produce un vector de salida **y**, donde cada elemento y_i indica la salida de la i-ésima neurona de la capa **q**, que comete un error igual a:

$$e_i = y_{di} - y_i \tag{10}$$

A partir de este error individual se estima el error total en la salida (capa **p**) mediante la función cuadrática:

$$E_q = \frac{1}{2} \cdot \sum_i e_i^2 = \frac{1}{2} \cdot \sum_i (y_{di} - y_i)^2 \tag{11}$$

Esta función, llamada Suma de los Cuadrados de Errores o SSE (del inglés Sum of Squared Errors) depende de cada salida y por lo tanto está representada por una superficie en un espacio **n** dimensional (recordar que **n** es la cantidad de neuronas en la capa de salida). En la Figura 18 se muestra, a modo de ejemplo, la representación gráfica de la función de error en función de sólo dos pesos. Las irregularidades de la superficie, con infinidad de mínimos locales dificultan arribar a una solución óptima.

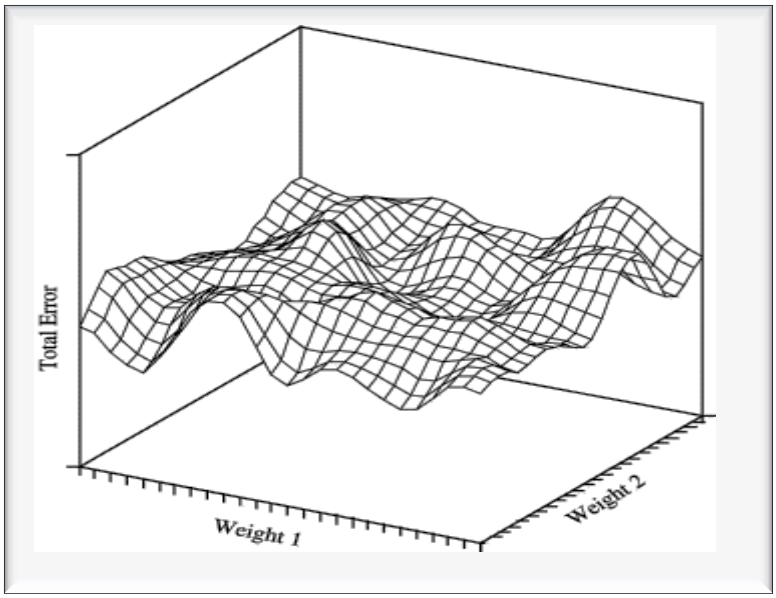

Figura 18 – Superficie de error esquemática tomando en cuenta solo dos pesos como variables. Extraída de [Randall]

El objetivo del algoritmo de retropropagación es encontrar un mínimo y la estrategia para ello es “movernos” en el sentido opuesto al de crecimiento (gradiente) de la función. Para ello calculamos la derivada respecto a cada salida cuyo resultado es simplemente la diferencia entre la salida deseada y la dada por la red.

$$E_i = \frac{dE_q}{dy_i} = -e_i = -(y_{di} - y_i) \tag{12}$$

En la segunda fase de este algoritmo, para poder ajustar los pesos que actúan sobre las entradas de cada neurona se debe conocer como el error es afectado por ellos. Consideramos una función de activación logística y función de salida identidad, por lo tanto:

$$y_i = \frac{1}{1 - e^{-x_i}} \tag{13}$$

Aplicamos la regla de la cadena para obtener la tasa de variación del error de la i-ésima neurona respecto a la variación de los pesos asociados a cada una de sus entradas:

$$\gamma_{ij} = \frac{dE_q}{dw_{ij}} = \frac{dE_q}{dy_i} \cdot \frac{dy_i}{dx_i} \cdot \frac{dx_i}{dw_{ij}} = -(y_{di} - y_i) \cdot y_i \cdot (1 - y_i) \cdot y_j \tag{14}$$

Donde, el resultado del primer factor se obtuvo en (12) y los otros dos de la siguiente manera:

$$\frac{dy_i}{dx_i} = \frac{1}{(1 - e^{-x_i})} \cdot \frac{(-e^{-x_i})}{(1 - e^{-x_i})} = \frac{1}{(1 - e^{-x_i})} \cdot \left[1 - \frac{1}{(1 - e^{-x_i})}\right] = y_i.(1 - y_i) \tag{15}$$

$$\frac{dx_i}{dw_{ij}} = \frac{d(y_j.w_{ij})}{dw_{ij}} = y_j \tag{16}$$

Denominamos con δ_i a:

$$\delta_i = (y_{di} - y_i) \cdot y_i \cdot (1 - y_i) = -E_i \cdot y_i \cdot (1 - y_i) \tag{17}$$

y expresamos γ_{ij} de forma simplificada:

$$\gamma_{ij} = -\delta_i \cdot y_j \tag{18}$$

Hasta aquí se calculó el error en la capa de salida, lo cual se simplifica al conocer la salida deseada por ser un dato requerido para el entrenamiento. Para obtener el error de las capas anteriores se lo debe retropropagar, y es lo que se explica a continuación.

Con ayuda de la Figura 19 analizaremos ahora como la salida de una neurona j de la capa **p** afecta a las **n** neuronas de la capa **q** y por ende al error total.

Nuevamente, aplicando regla de la cadena calculamos el producto entre la tasa de variación del error respecto a las entradas de la i-ésima neurona y la tasa de variación de esas entradas respecto a la salida proveniente de la j-ésima neurona:

$$E_j = \frac{dE_q}{dy_j} = \sum_{j=1}^{n} \left(\frac{dE_q}{dx_i} \cdot \frac{dx_i}{dy_j}\right) = \sum_{j=1}^{n} \left(\frac{dE_q}{dx_i} \cdot \frac{d(y_j.w_{ij.})}{dy_j}\right) = -\sum_{j=1}^{n} (\delta_i.w_{ij}) \tag{19}$$

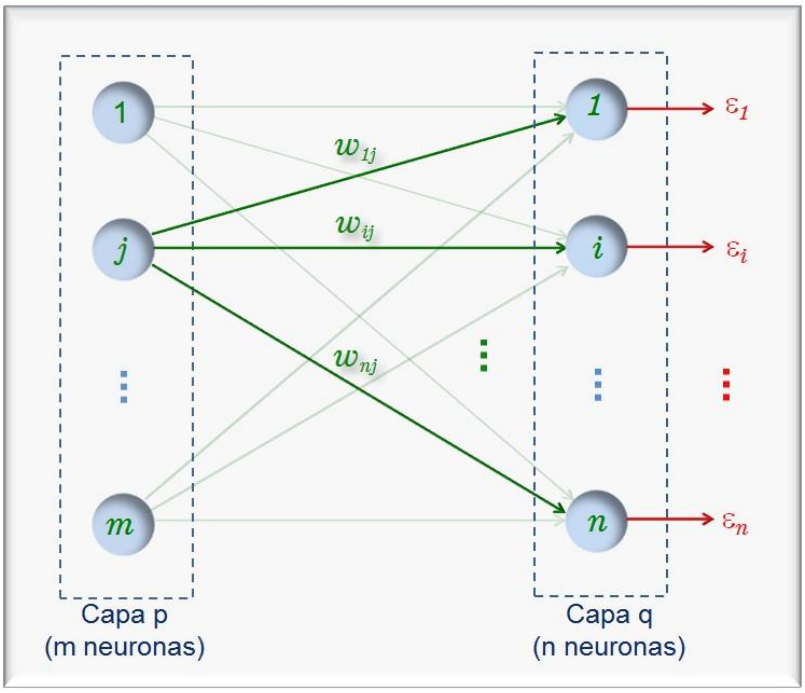

Figura 19 - Influencia de una neurona j de la capa p sobre el error en la capa siguiente.

El valor E_j resulta equivalente a E_i, ambos representan la variación del error en la capa **p** y **q** respectivamente, respecto a la variación de las salidas de las neuronas de esa capa. Siguiendo un razonamiento igual al de antes podemos extender el cálculo considerando una capa **r** anterior a la capa **p** y aplicando las ecuaciones (17) y (18):

$$\delta_j = -E_j \cdot y_j \cdot (1 - y_j) = y_j \cdot (1 - y_j) \cdot \sum_{j=1}^{n} (\delta_i . w_{ij}) \tag{20}$$

$$\gamma_{jr} = -\delta_j \cdot y_r \tag{21}$$

Los valores obtenidos en las ecuaciones (17) y (18) y en sus homólogas (20) y (21) son los necesarios para calcular la variación de los pesos tendientes a disminuir el error producido. La modificación de los pesos puede realizarse de diferentes maneras, la más sencilla es la *regla delta generalizada* que aplica a cada peso un incremento directamente proporcional a la tasa de variación del error. Dado que el incremento debe ir en sentido opuesto al crecimiento del error, tomará signo negativo.

$$\Delta w_{ij} \propto -\gamma_{ij} \tag{22}$$

Llamando w' a los nuevos pesos y adicionando una constante η llamada *tasa de aprendizaje* tenemos:

$$\Delta w_{ij} = \eta . \delta_j . y_i$$
$$w'_{ij} = w_{ij} + \Delta w_{ij} \tag{23}$$

La tasa de aprendizaje o learning rate (en inglés) se fija antes de comenzar el entrenamiento, determina cuán grande será la variación de los pesos y de este modo la rapidez de

aprendizaje. Su valor se encuentra típicamente entre 0,01 y 1. Con el objetivo de que el método alcance rápidamente un mínimo es preferible tomar un valor grande de η pero esto a su vez ocasiona oscilaciones que dificultan o impiden su convergencia. La elección del valor óptimo se basa en prueba y error.

Una variante a la regla delta generalizada fue propuesta por Rumelhart, Hinton y Williams (1986) y consiste en adicionar un término proporcional a la cantidad del último cambio realizado sobre un peso. Al coeficiente que pondera dicha cantidad se lo llama *momento* (μ) y aplicando (23) obtenemos:

$$\Delta w'_{ij} = \eta . \delta_j . y_i + \mu \cdot \Delta w_{ij}$$
$$w'_{ij} = w_{ij} + \Delta w'_{ij} \qquad (24)$$

Esta última variante de la regla delta generalizada es el método elegido para entrenar las redes en este trabajo.

Extracción de características

La selección de un adecuado conjunto de características representativas, de las emociones para nuestro caso, es una cuestión sumamente importante. Los sistemas de recoconocimiento de patrones usados en la etapa de clasificación (como se vio en un sistema de REH) verán afectados su desempeño por dicha elección. En el campo del REH hay ciertas características que son las mas comunmente empleadas en los trabajos, sin embargo lo correcto es analizar las señales involucradas en nuestra aplicación y en base a ello determinar cuales son las reelevantes para abordar nuestro problema. Sin embargo esto no es una tarea sencillla ya que requiere de cierto conocimiento y experiencia. En [Ayadi, 2011] se han presentado cuatro cuestiones importantes a la hora de extraer características de una señal:

- ⇒ Región de análisis de la señal
- ⇒ Cuales son las características adecuadas para la tarea en cuestión
- ⇒ Cual es el efecto de los procesamiento de voz usuales (filtrado y remoción de silencios) sobre el desempeño del clasificador
- ⇒ Si basta con usar características acústicas o se debería considerar información lingüistica, gestos faciales, entre otros.

Del primer punto surge una clasificación de las características entre aquellas *locales* que son obtenidas mediante un análisis por tramos pequeños de la señal y las *globales* que son una representación estadística de todas las características sobre una elocución completa.

Algunos aspectos a saber sobre la naturaleza estadística de la señal de voz son: la *función de densidad de probabilidad (fdp), estacionariedad y ergodicidad*. Cuando se tiene una gran cantidad de muestras, que sean representativas de una señal la fdp puede estimarse mediante un histograma de amplitudes. Por el contrario cuando se analiza para intervalos cortos (decenas de milisegundos) esto pierde validez ya que la fdp está muy influenciada por el tipo de sonido analizado. Una simplificación muy común es considerar la voz como un proceso estocástico ergódico, lo que a su vez está ligado con la suposición de estacionariedad. Es decir, si no se cumple esta última tampoco lo será la primera. Sin embargo se debe tener en cuenta que esto es relativo ya que cuando se analizan intervalos cortos (20-60 ms) la señal puede considerarse estacionaria en dichos tramos, en consecuencia se considera a la señal de voz como cuasi-estacionaria. Cuando se desea procesar y extraer información se debe elegir en forma adecuada la duración de dichos tramos. [Navarro Mesa].

El conjunto de características que fueron utilizadas aqué se seleccionó en base al trabajo previo de [Albornoz-Rufiner, 2011] donde realizaron exhaustivas pruebas tomando distintas combinaciones basadas en F_0, E, MFCC y MLS para entrenar un PMC. Dado que la base de

datos utilizada es la misma, se optó por el vector que arrojó el mejor resultado en sus experimentos compuesto por las siguientes 46 características:

12 MFCC medios	30 MLS medios	μ(F_0)	μ(E)	σ(F_0)	σ(E)

A continuación se explica brevemente como fueron obtenidas las características usadas para el entrenamiento de las redes, lo cual fue extraído de [Albornoz, 2011].

El análisis cepstral es una técnica que aplicada sobre una señal permite obtener información acerca de cómo fue originada. Se denomina cepstrum a lo que se obtiene como resultado de aplicar la transformada inversa de Fourier al logaritmo decimal del módulo de la transformada de Fourier de una señal. Llamando $\boldsymbol{FT}$ a la transformada de Fourier (del inglés Fourier Transform), $\boldsymbol{FT^{-1}}$ a su inversa, $\mathbf{s}$ a una elocución cualquiera del corpus, entonces el cepstrum C de $\mathbf{s}$ se calcula como:

$$C = FT^{-1}\{log(|FT\{\boldsymbol{s}\}|)\} \tag{25}$$

Introducimos el concepto de *escala de Mel* que constituye una escala de tonos, es decir, un ordenamiento de valores de frecuencias que son percibidas como igualmente espaciadas por el oído humano. A partir de esto se puede obtener la representación de los Coeficientes Cepstrales en Frecuencias Mel (MFCC, del inglés Mel Frequency Cepstral Coefficients) que consiste en tomar cada tramo de la transformada de Fourier de la señal, aplicarle un banco de filtros cuyas frecuencias centrales corresponden a la escala Mel. Se calcula el logaritmo de la potencia en cada una de estas bandas, se aplica una transformación de tipo coseno y finalmente se obtienen los MFCC mediante $\boldsymbol{FT^{-1}}$, que representan las amplitudes resultantes.

Otra característica utilizada es la media del logaritmo del espectro (MLS, del inglés Mean Logarithm Spectrum) que se calcula tomando la transformada discreta de Fourier v de cada elocución y calculando por tramos de la siguiente manera:

$$MLS(k) = \frac{1}{N_s} \cdot \sum_{n=1}^{N_s} log|v_s(n,k)| \tag{26}$$

donde k es la banda de frecuencias, N_s es el número de tramos de la elocución s y $v_s(n,k)$ es la transformada discreta de Fourier de la señal s para el tramo n.

Las restantes características que componen el vector de entrada a la red son la energía (E) y la frecuencia fundamental (F_0) de cada elocución. El cálculo de E se hizo a través del producto interno de cada tramo de la señal s consigo misma, esto es, su norma 2 al cuadrado:

$$E = s \cdot s = \|s\|_2^2 = \sum_i |s_i|^2 \tag{27}$$

donde el subíndice i denota cada elemento de la señal.

Para determinar la F_0 se siguió el procedimiento explicado en [Albornoz, 2011] que consiste en calcularla a partir de los coeficientes cepstrales. En el cesptrum puede observarse un pico de gran amplitud localizado en la frecuencia fundamental y luego otros de menor amplitud ubicados en los múltiplos de ésta. Teniendo en cuenta que el rango de F_0 se encuentra entre los 80 y 500 Hz para la voz humana, podemos ubicar el pico correspondiente en el cesptrum con una ventana de 15 ms como lo ejemplifica la Figura 20.

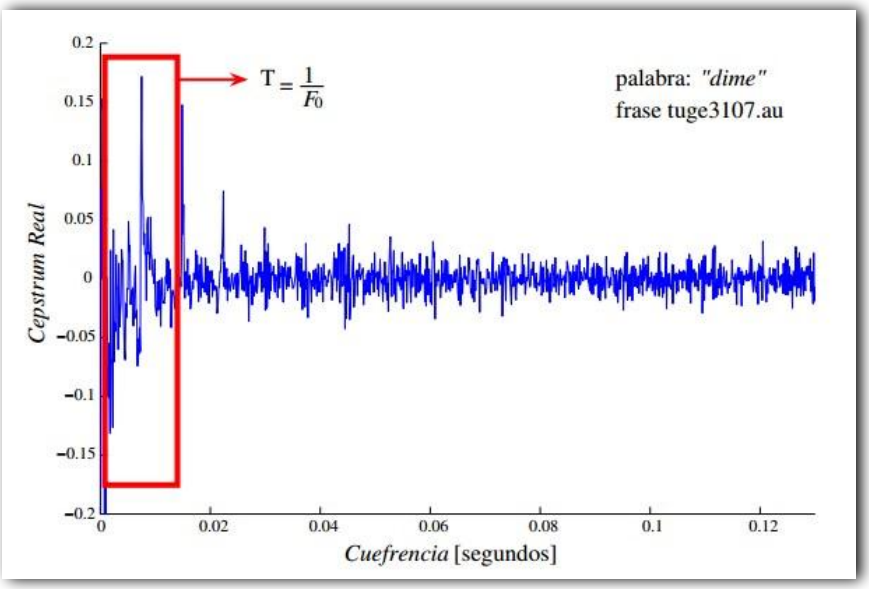

Figura 20 – Determinación de F_0 a partir de los coeficientes cepstrales. Extraído de [Albornoz, 2011].

Base de datos

No menos importante que el desarrollo de los métodos en una investigación es la elección de una correcta base de datos cuando se la requiere. La misma debe adecuarse respecto de las características que serán analizadas y de la cantidad de muestras necesarias para que los resultados sean valores estadísticos representativos. Los avances en minería de datos demandan bases de datos cada vez más completas. La minería de datos o data mining (en inglés) engloba a los procesos, dentro de las ciencias de la computación, destinados a analizar grandes volúmenes de datos con el objetivo de extraer información "oculta", no identificable directamente en la señal original, útil para su uso posterior.

En este trabajo se ha utilizado una base de datos en alemán denominada comúnmente EMO-DB y desarrollada en el Departamento de Acústica Técnica de la Universidad Técnica de Berlín, Alemania en 2005 [Burkhardt, 2005]. El corpus consta de 535 frases expresadas con siete emociones diferentes: enojo, aburrimiento, asco, miedo, alegría, tristeza y neutral.

Las grabaciones fueron realizadas en una cámara anecoica por un grupo de 10 actores, 5 hombres y 5 mujeres. Cada uno de ellos interpretó las mismas 10 frases simulando las siete emociones que fueron adquiridas digitalmente utilizando una frecuencia de muestreo de 48 KHz y luego submuestreadas a 16 KHz. Posteriormente, todas las grabaciones fueron sometidas a una evaluación de percepción llevada a cabo por 20 personas que debían clasificarlas según a que emoción correspondía y valorar con un porcentaje de 0 a 100 su naturalidad. Acorde a este procedimiento fueron seleccionadas solo aquellas grabaciones clasificadas correctamente al menos en el 80% de los casos y cuya naturalidad no fue menor del 60%. En la Tabla 3 se muestran el total de frases agrupadas por emoción: enojo (E), aburrimiento (A), asco (S), miedo (M), alegría (L), tristeza (T) y neutral (N).

Tabla 3 – Frases del corpus clasificadas por emoción

Emoción	E	A	S	M	L	T	N
Nro. de frases	127	81	46	69	71	62	79

El número de frases no es el mismo para cada emoción y por lo tanto usar esta base de datos así introduciría un sesgo en los resultados ya que, por ejemplo, la red aprendería mejor a reconocer “enojo” dada la significativa mayor cantidad de frases de esta emoción respecto a las demás. Para evitar esto se equipararon las cantidades a la menor de ellas (S), es decir que se tomaron solo 46 frases de cada emoción. Las mismas fueron seleccionadas aleatoriamente completando un total de 322 (46x7=322) elocuciones.

Para lograr un adecuado entrenamiento y verificación del desempeño de la red, se dividieron los datos en conjuntos de entrenamiento (En), prueba (P) y validación (V) cuya proporción de frases en cada uno era 60%, 20% y 20%, respectivamente. A a su vez se conformaron 10 particiones diferentes con la misma estuctura. En la Tabla 4 puede verse la distribución para una partición cualquiera.

Tabla 4 – Distribución de las frases en los conjuntos de entrenamiento, prueba y validación para una partición

Conjunto	% aprox.	Cant. de frases	Cant. de frases por emoción
En	60	196	28
P	20	63	9
V	20	63	9

De forma más detallada, en la Tabla 5 puede verse como está constituida cada una de las particiones, indicadas la cantidad de frases correspondiente a cada emoción y clasificadas por sexo, masculino (Ma) o femenino (Fe).

Tabla 5 – Distribución de frases en los conjuntos de entrenamiento (En), prueba (P) y validación (V), discriminadas entre las realizadas por hombres (H) y mujeres (M) para las 10 particiones.

		E		A		S		M		L		T		N	
		Ma	Fe	Ma	Fe	Ma	Fe	Ma	Fe	Ma	Fe	Ma	Fe	Ma	Fe
1	En	12	16	12	16	7	21	17	11	7	21	13	15	14	14
	P	3	6	5	4	1	8	6	3	4	5	4	5	5	4
	V	3	6	2	7	3	6	4	5	5	4	2	7	5	4
2	En	13	15	12	16	7	21	16	12	8	20	12	16	14	14
	P	4	5	5	4	2	7	3	6	3	6	2	7	5	4
	V	5	4	3	6	2	7	5	4	4	5	2	7	6	3
3	En	12	16	16	12	8	20	15	13	10	18	11	17	10	18
	P	5	4	2	7	2	7	5	4	3	6	5	4	3	6
	V	4	5	4	5	1	8	4	5	3	6	5	4	7	2
4	En	16	12	12	16	7	21	13	15	9	19	9	19	16	12
	P	5	4	4	5	1	8	7	2	6	3	5	4	6	3
	V	4	5	2	7	3	6	5	4	4	5	2	7	5	4
5	En	15	13	14	14	5	23	14	14	14	14	13	15	16	12
	P	5	4	2	7	2	7	6	3	3	6	2	7	7	2
	V	4	5	3	6	4	5	4	5	3	6	3	6	3	6

Tabla 5 – (continuación)

6	**En**	15	13	9	19	7	21	16	12	12	16	9	19	11	17
	P	0	9	5	4	2	7	6	3	4	5	5	4	5	4
	V	3	6	8	1	2	7	1	8	0	9	6	3	5	4
7	**En**	15	13	12	16	6	22	16	12	8	20	10	18	16	12
	P	4	5	5	4	4	5	5	4	5	4	3	6	4	5
	V	7	2	3	6	1	8	3	6	3	6	5	4	4	5
8	**En**	17	11	13	15	5	23	18	10	10	18	12	16	11	17
	P	4	5	5	4	2	7	2	7	2	7	4	5	7	2
	V	3	6	5	4	4	5	6	3	5	4	3	6	4	5
9	**En**	16	12	9	19	6	22	14	14	9	19	13	15	15	13
	P	3	6	4	5	1	8	5	4	3	6	2	7	2	7
	V	4	5	1	8	4	5	6	3	4	5	2	7	4	5
10	**En**	13	15	16	12	8	20	17	11	11	17	13	15	14	14
	P	3	6	1	8	1	8	3	6	3	6	5	4	6	3
	V	6	3	4	5	2	7	5	4	3	6	4	5	3	6

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
N. Cibau, H. L. Rufiner & E. M. Albornoz; "Reconocimiento de emociones en el habla mediante autocodificadores profundos (Undergraduate project)"
Facultad de Ingeniería - Universidad Nacional de Entre Ríos, 2013.

Capítulo 3. Método propuesto

Se presentaron en el capítulo 2 muchos de los conceptos que se engloban en el área de REH y que serán útiles para comprender el método propuesto en este trabajo. Previamente, en el capítulo 1 se analizó el estado del arte de las técnicas usadas actualmente para el REH y se vio que los resultados obtenidos para PMCs encuentran por debajo de los demás clasificadores como HMM o GMM. Una de las razones es que los PMCs utilizados constan de una sola capa oculta, lo cual es una limitante para el desempeño del clasificador. Sin embargo el hecho de agregar mas capas ocultas acarrea otros inconvenientes como se verán en este capítulo. Así mismo se explicará un método que permite entrenar estas redes profundas aplicado a un caso de REH.

Aprendizaje Profundo

El aprendizaje profundo es un concepto que se ha estado manejado desde hace dos décadas, pero los intentos de implementación no arribaron a resultados prometedores dado que, el ampliamente usado algoritmo de retropropagación, no resulta eficiente cuando se pretenden entrenar redes con dos o más capas ocultas a partir de pesos inicializados al azar. La razón de este inconveniente reside en la misma esencia del algoritmo, y se debe a que el gradiente de error se torna demasiado pequeño cuando es retropropagado hacia las capas más alejadas de la salida. Por lo tanto la variación de pesos resulta insignificante y la red permanece “estancada” en una solución muy pobre.

El nuevo método propuesto por [Hinton et.al., 2006] consiste en realizar un pre-entrenamiento de a una capa por vez con el objetivo de llevar los pesos a valores cercanos a una buena solución. Luego, entrenar la red completa mediante retropropagación y cuyos pesos fueron inicializados con los valores obtenidos previamente [Hinton & Salakhutdinov, 2006] [Brueckner, 2012]. De esta manera el pre-entrenamiento suple la deficiencia del algoritmo de retropropagación de alcanzar por sí solo una óptima solución a partir de pesos inicializados al azar en una red profunda. Si bien Hinton tomo cada capa como una RBM para realizar el pre-entrenamiento, en este trabajo se optó por utilizar autocodificadores y la justificación de esta elección se da a continuación.

Podemos describir de forma general una tarea de clasificación como un proceso en el cual, dadas ciertas clases, se debe seleccionar de un conjunto de características aquellas que mejor representan dichas clases. Cuando ese conjunto posee demasiadas características la tarea puede resultar complicada y sería mejor disponer de un conjunto más pequeño pero cuyos elementos sean representativos de los primeros. Aplicando este concepto a redes neuronales, lo que deseamos es poder obtener una representación de baja dimensión a partir de información de mayor dimensión. Este proceso de reducción de dimensión es llevado a cabo en un autocodificador, que consiste en una red multicapa con una capa central más pequeña (menor cantidad de neuronas) y cuyo objetivo es reconstruir en la salida los vectores de entrada [Hinton & Salakhutdinov, 2006]. Dado que la información de entrada es forzada a atravesar ese “cuello de botella” y luego debe ser reconstruida, la red aprende un código de menor dimensión. Un autocodificador entonces está constituido por una parte “codificadora” que incluye desde la capa de entrada hasta la central, y una parte “decodificadora” desde la capa central hasta la de salida. La representación de baja dimensión lograda en el autocodificador es entonces utilizada como el vector de entrada para un clasificador.

En el contexto de las redes neuronales un clasificador es un tipo de configuración de red cuya capa de salida posee tantas neuronas como clases. Para cada una de estas clases existen diferentes patrones de entrada a partir de los cuales la red debe aprender a decidir a

qué clase corresponde. Esto implica que para cada patrón sólo una neurona de salida se tomará como válida y para determinarlo existen diferentes criterios. Si la función de activación es logística las salidas serán valores comprendidos entre 0 y 1. En este trabajo, la regla aplicada para determinar la clase que será la salida de la red es “ganador toma todo”, esto quiere decir que la neurona cuyo valor de salida sea el más alto determinará la única clase que es salida de la red para cada patrón de entrada.

Un autocodificador de una sola capa oculta puede ser entrenado sin problemas mediante retropropagación, pero para el caso de autocodificadores profundos no lineales (que poseen 2, 3 o 4 capas ocultas) el algoritmo pierde eficacia, según se explicó anteriormente. Mediante una serie de pasos explicados más abajo se describe el método de entrenamiento para el autocodificador profundo utilizado en este trabajo. Luego, el agregado de una última capa convierte el autocodificador profundo (AP) en un clasificador profundo (CP).

Diseño del autocodificador profundo

El desarrollo del método propuesto en este trabajo, cuyo objetivo final es obtener el CP, será dividido en 8 etapas. Daremos un vistazo general a dicho clasificador (Figura 21) que ayudará a comprender de mejor manera los pasos previos. Consiste en una red con una capa de entrada **i**, dos capas ocultas $\mathbf{h_1}$, $\mathbf{h_2}$ y la capa de salida **o**. La cantidad de neuronas en cada capa se simboliza con $n_{i,}$ n_{h1}, n_{h2} y n_o, respectivamente. A modo esquemático las capas están graficadas como rectángulos cuyo tamaño es proporcional a la cantidad de neuronas que posee. Las capas adyacentes se encuentran totalmente conectadas, es decir que cada neurona posee salidas hacia todas las que se encuentran en la capa siguiente y dichas conexiones son graficadas mediante trama de puntos.

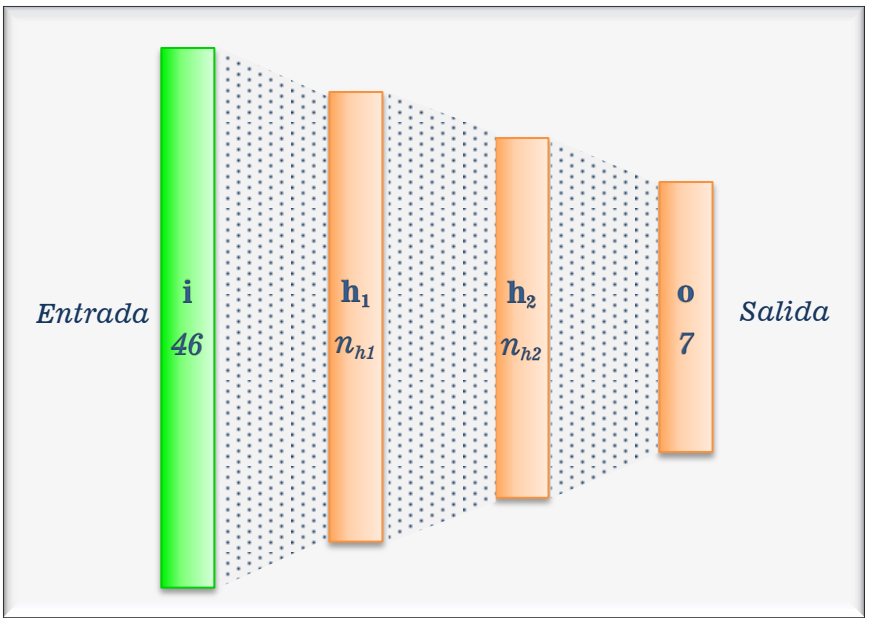

Figura 21 – Diagrama del CF de 3 capas. Las capas son representadas mediante rectángulos y las conexiones entre ellos por trama de puntos. Posee una capa de entrada i, dos capas ocultas $\mathbf{h_1}$, $\mathbf{h_2}$ y la capa de salida o. El cantidad de nodos en cada capa es indicada por n_i, n_{h1}, n_{h2} y n_o, respectivamente. En la capa de entrada la función de activación es lineal (sombreado verde), mientras que en el resto es logística.

Dado que el vector de entrada posee 46 elementos y son 7 las clases (emociones) que deben clasificarse, la cantidad de neuronas de entrada será n_i=46 y de salida n_o=7. El número de neuronas en las capas ocultas son grados de libertad que permitirán construir distintas configuraciones de redes. La única restricción impuesta es que n_i>n_{h1}>n_{h2}>n_o cu-

yo propósito es obligar a la red a efectuar una reducción de dimensionalidad progresivamente en cada capa. Las neuronas poseen función de activación lineal en la capa de entrada y logística para el resto. A su vez todas las neuronas que conforman el CP poseen función de salida lineal.

En las etapas, que se explicarán a continuación, el entrenamiento de las redes se lleva a cabo mediante el algoritmo de retropropagación añadiendo el término de momento (de aquí en más lo llamaremos ARPM), para cada época se calcula el SSE de entrenamiento (SSEe) y el de prueba (SSEp). Usaremos el término "mejor red" para referirnos a aquella red cuya configuración de pesos arrojó el menor SSEp durante el entrenamiento, y para la cual además se calculó el SSE de validación (SSEv). En las Etapas 7 y 8 se aplica el mismo criterio pero dado que es una proceso de clasificación se toma el error porcentual EP (EPe, EPp y EPv) que corresponde a las clasificaciones incorrectas.

Etapa 1

El objetivo de esta etapa es realizar el pre-entrenamiento de la primer capa, es decir inicializar los pesos de las conexiones entre **i** y $\mathbf{h_1}$. Para ello se construyó un autocodificador como el de la Figura 22, que consta de una capa oculta $\mathbf{h_1}$ con n_{h1} neuronas y capas de entrada y salida **i** con 46 neuronas, denominamos entonces a esta red como **i+h$_1$+i.** Luego del entrenamiento se seleccionó la mejor red.

Etapa 2

Se tomó la red de la Etapa 1 y fue eliminada la capa de salida, resultando un codificador **i+h$_1$** como el de la Figura 23. Luego, los patrones de entrenamiento, prueba y validación fueron pasados por la red y sus salidas constituirán los vectores de entrada para entrenar la red de la etapa siguiente.

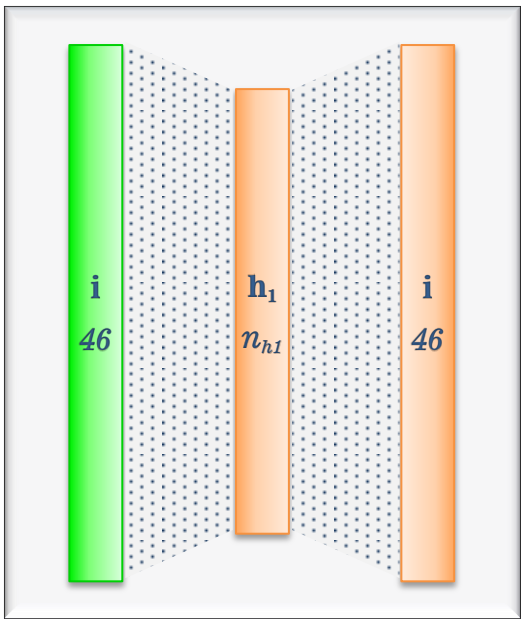

Figura 22 – Diagrama del autocodificador correspondiente a la Etapa 1 formado por una única capa oculta $\mathbf{h_1}$. La capa de entrada posee 46 nodos y función de activación lineal.

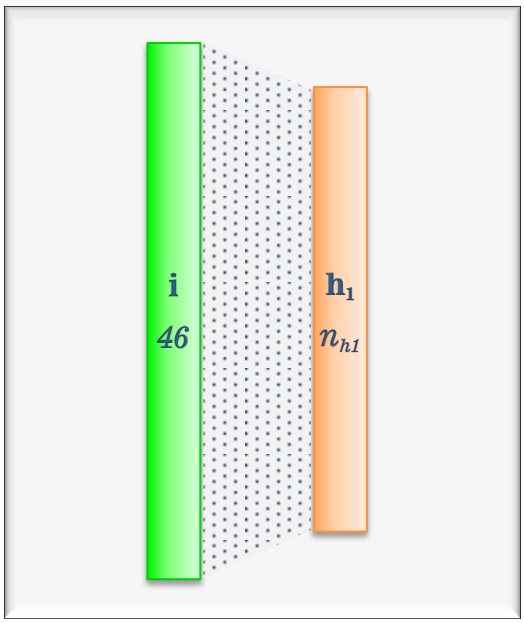

Figura 23 – Red i+h1 de la Etapa 2 que resulta luego de eliminar la capa de salida del autocodificador de la Etapa 1.

Etapa 3

Similar a lo realizado en la Etapa 1, se construyó un autocodificador $\mathbf{h_1+h_2+h_1}$ (Figura 24) que fue entrenado con los patrones obtenidos en el paso anterior y luego seleccionado el mejor. Con esto se logra inicializar los pesos entre las capas $\mathbf{h_1}$ y $\mathbf{h_2}$ del CP.

Etapa 4

Siguiendo el mismo procedimiento de la Etapa 2, se tomó el autocodificador seleccionado en la Etapa 3 y fue eliminada su capa de salida para obtener la red $\mathbf{h_1+h_2}$ que se muestra en la Figura 25.

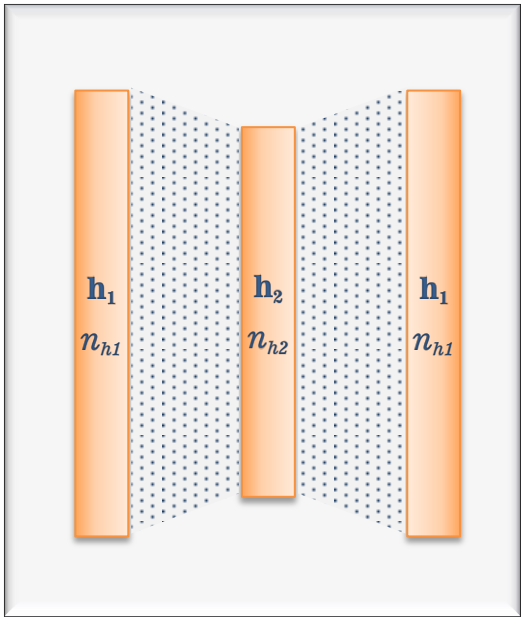

Figura 24 - Autocodificador de la Etapa 3 formado por una única capa oculta h_2. Las capas de entrada y salida poseen n_{h1} nodos. Posee función de activación lineal en la capa de salida.

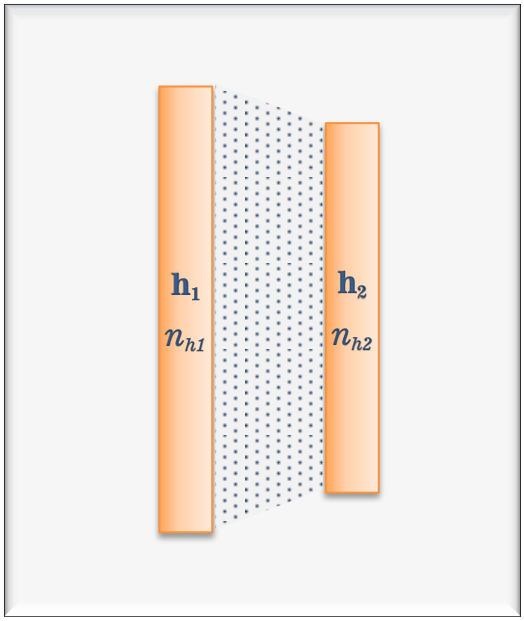

Figura 25 - Red h_1+h_2 de la etapa 4 que resulta luego de eliminar la capa de salida del autocodificador de la Etapa 3.

Etapa 5

Esta es la etapa más compleja y tiene como objetivo construir el AP. Primeramente, se toman las redes de las Etapas 2 ($\mathbf{i+h_1}$) y 4 ($\mathbf{h_1+h_2}$), que se unen para formar una red $\mathbf{i+h_1+h_2}$ mediante conexiones "uno a uno" entre las capas $\mathbf{h_1}$. Este tipo de conexiones son sólo un requisito del software utilizado, ya que poseen sus pesos igual a uno y no se modifican en el entrenamiento, es por ello que para simplificar la notación escribimos $\mathbf{i+h_1+h_2}$ en vez de $\mathbf{i+h_1+h_1+h_2}$ (de aquí en más se asume este criterio al unir redes). Luego, se hace una copia de esta red y es invertida resultando en una configuración $\mathbf{h'_2+h'_1+i'}$ (si bien las capas son idénticas, se usan los apóstrofes para distinguirlas, ya que su posterior ubicación en el AP será diferente). Finalmente, se ensamblan ambas partes para formar el AP $\mathbf{i+h_1+h_2+h'_1+i'}$ como se ve en la Figura 26. Una vez realizado el entrenamiento se seleccionó el mejor AP.

Etapa 6

Se tomó el AP seleccionado en el paso anterior y eliminando las capas $\mathbf{h'_2+h'_1+i'}$ se obtuvo el decodificador $\mathbf{i+h_1+h_2}$ como se muestra en la Figura 27. Los mismos patrones de entrada utilizados en la Etapa 5 fueron pasados por esta red para generar, mediante sus salidas, los patrones de entrenamiento para la red de la próxima etapa.

Etapa 7

Con el objetivo de inicializar los pesos de la última capa del CP se creó una red simple que consiste en una capa de entrada **h_2''** con n_{h2} neuronas y una capa de salida **o** de 7 nodos (Figura 28). A cada nodo corresponde una de las clases -emociones- ya que ésta capa constituirá luego la salida del clasificador. La red se entrenó utilizando los patrones de entrada generados en el paso anterior, y las salidas deseadas se tomaron a partir de los datos rotulados del corpus. En esta etapa la mejor red seleccionada fue la que arrojó el porcentaje mayor de clasificaciones correctas para los patrones del conjunto de prueba.

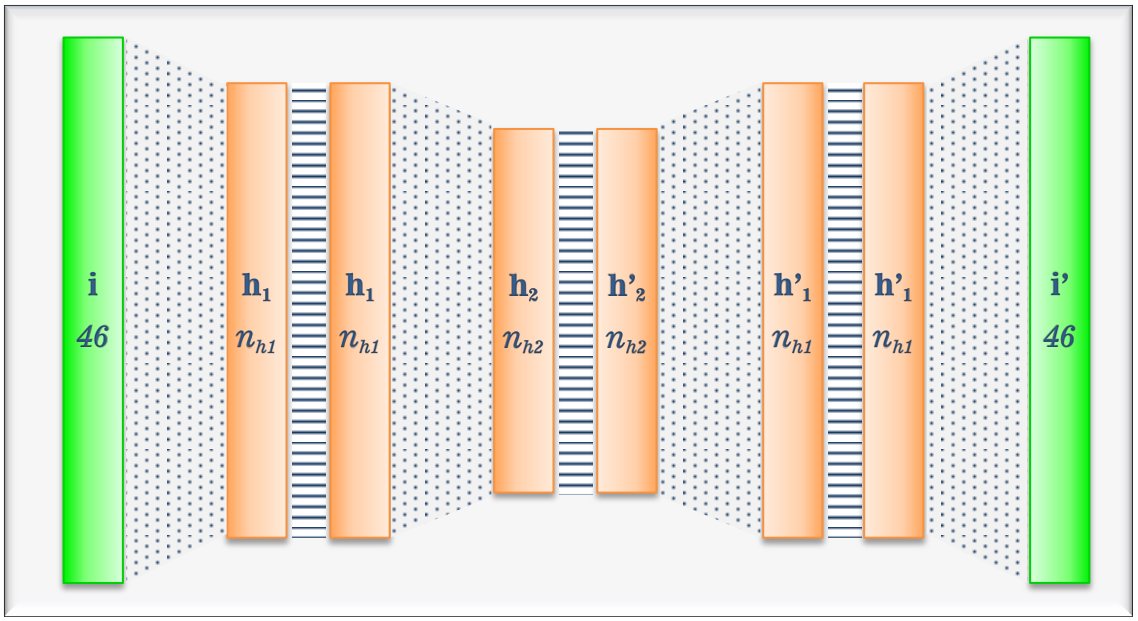

Figura 26 - Diagrama del AP i+h_1+h_2+h_1'+i' correspondiente a la Etapa 5. Las conexiones "uno a uno" se muestran con líneas paralelas.

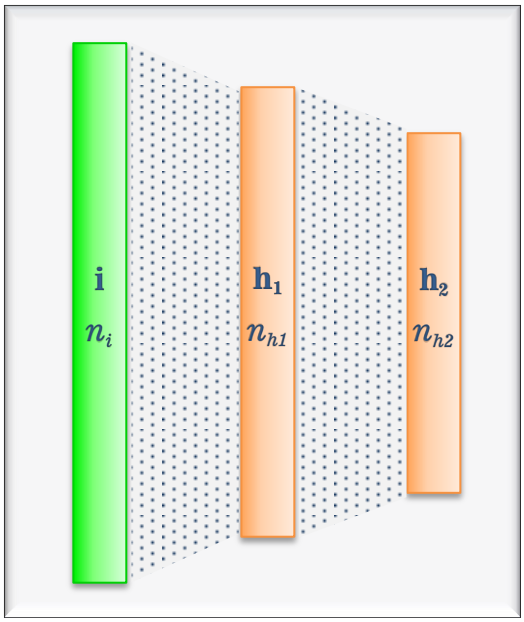

Figura 27 - Codificador i+h_1+h_2 obtenido en la Etapa 5, luego de eliminar la parte decodificadora del AP.

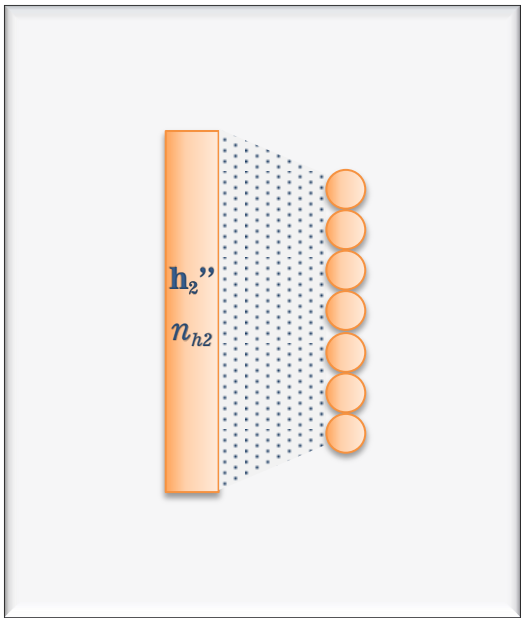

Figura 28 – Red de la Etapa 7 cuyo objetivo es inicializar los pesos de la capa de salida del CP.

Etapa 8

Hasta aquí se ha completado el pre-entrenamiento para cada capa y se procedió a unir las redes de las Etapas 6 y 7, formando el CP **i+h$_1$+h$_2$+o** mostrado en la Figura 29. Con los pesos inicializados gracias al proceso anterior, se realizó un entrenamiento final que actúa como un "ajuste fino" corrigiendo los pesos para mejorar el desempeño del clasificador. Se eligió la mejor red aplicando el mismo criterio que en la Etapa 7.

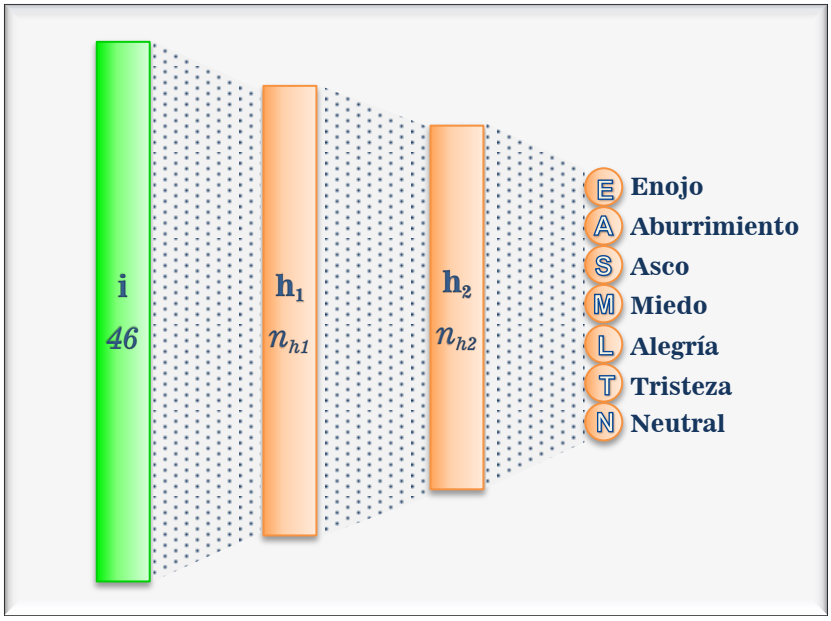

Figura 29 – CP obtenido al final del proceso completo. En la capa de salida se muestran esquemáticamente las 7 clases (emociones).

Se ha presentado en este capítulo un nuevo método para entrenar redes profundas. Dicho método se vale del aprendizaje no supervisado (para el caso de los autoencoders) para inicializar los pesos de una red cuyo proceso de aprendizaje es supervisado. En el capítulo siguiente se expondrán los resultados de experimentos realizados utilizando distintas estructuras de red y parámetros de entrenamiento.

Capítulo 4. Experimentos y resultados

Para llevar a cabo la implementación de las redes neuronales y realizar los procesamientos requeridos en una computadora o clúster se utilizó el Simulador de Redes Neuronales Stuttgart (*Stuttgart Neural Network Simulator* o SNNS, en inglés), un software desarrollado por el Instituto de Sistemas de Alto Rendimiento Paralelo y Distribuido de la Universidad de Stuttgart. El SNNS es de descarga gratuita y consta de cuatro partes:

- ⇒ Simulador del kernel: opera con la estructura de datos interna de las redes permitiendo realizar cualquier operación sobre ellas.
- ⇒ Interfaz gráfica de usuario "XGUI": construida sobre el kernel, posibilita la creación, simulación, representación y operaciones sobre las redes a través de una interfaz gráfica.
- ⇒ Interfaz para ejecución por lotes "batchman": programa que brinda la posibilidad de ejecutar los procesos para crear y entrenar redes en segundo plano.
- ⇒ Compilador de redes "snns2c": herramienta que permite convertir una red entrenada en SNNS a un archivo en lenguaje C que puede ser luego compilado en un ejecutable o implementado como una función dentro de una aplicación hecha en C.

El SNNS proporciona la interfaz gráfica para efectuar las tareas, pero dado que los procesos de entrenamiento y otras operaciones efectuadas sobre las redes requerían muchas horas de ejecución en CPU, se utilizó el programa batchman. Básicamente es un intérprete de instrucciones que utiliza un lenguaje particular basado en AWK, Pascal, Modula-2 y C. Algunas herramientas accesibles desde la interfaz gráfica no están disponibles en batchman. Para suplir aquellas que fueron necesarias en este trabajo y efectuar el resto de las operaciones se elaboraron scripts en Shell (.sh), Bash (.bsh) y Perl (.pl). Las curvas de error que se muestran en este capítulo se realizaron con Gnuplot. Todo se trabajó desde el sistema operativo Linux (distribución Ubuntu).

Como se mencionó en el capítulo anterior, se utilizó el ARPM que el SNNS tiene implementado y permite ajustar los siguientes cuatro parámetros:

η	tasa de aprendizaje
μ	momento
c	valor para evitar zona plana; término que es añadido a la derivada de la función de activación para evitar que el algoritmo se "estanque" en zonas planas de la superficie de error.
$\mathbf{e_{imax}}$	el error máximo tolerado en la salida de la i-ésima neurona de la capa de salida ($\mathbf{e_i}=y_{di} - y_i$). Los valores inferiores a $\mathbf{e_{imax}}$ serán retropropagados como cero ($\mathbf{e_i}=0$).

Estructuras de red y parametros utilizados

Se experimentaron 6 estructuras de redes que diferían en la cantidad de neuronas de sus capas ocultas (n_{h1} y n_{h2}). En la Tabla 6 se muestran estas estructuras y las redes que se fueron necesarias entrenar en las distintas etapas para lograrlas.

A su vez, en cada etapa de entrenamiento, se utilizaron dos combinaciones de parámetros del ARPM, llamados parámetros A (PA) y parámetros B (PB). Para los PA se utilizaron los valores típicos [MANUAL SNNS] en todas las etapas. Los PB se seleccionaron luego de

exhaustivas pruebas en las que se iban ajustando los parámetros con el objetivo de que las gráficas de error (SSE o EP según la Etapa) fueran más suaves.

Tabla 6 – Estructuras experimentadas del clasificador y redes entrenadas en las etapas previas

Estructura				Red entrenada en la etapa correspondiente				
n_i	n_{h1}	n_{h2}	n_o	Etapa 1	Etapa 3	Etapa 5	Etapa 7	Etapa 8
46	**27**	**17**	7	46+27+46	27+17+27	46+27+17+27+46	17+7	46+27+17+7
46	**30**	**20**	7	46+30+46	30+20+30	46+30+20+30+46	20+7	46+30+20+7
46	**33**	**23**	7	46+33+46	33+23+33	46+33+23+33+46	23+7	46+33+23+7
46	**36**	**26**	7	46+36+46	36+26+36	46+36+26+36+46	26+7	46+36+26+7
46	**39**	**23**	7	46+39+46	39+23+39	46+39+23+39+46	23+7	46+39+23+7
46	**42**	**23**	7	46+42+46	42+23+42	46+42+23+42+46	23+7	46+42+23+7

En estas pruebas además se estableció la cantidad de épocas de entrenamiento de modo que permitía el correcto aprendizaje de la red, y más allá de esa cantidad no se producía disminución del SSEp. Esto explica por qué para PB, cuya tasa de aprendizaje y momento son menores, se debió aumentar la cantidad de épocas. Los valores usados se muestran en la Tabla 7.

Tabla 7 – Parámetros usados en los experimentos

Estructura	PA					PB				
	Épocas	μ	η	c	e_{imax}	Épocas	μ	η	c	e_{imax}
Etapa 1	800	0,10	0,10	0,10	0,10	2500	0,08	0,05	0,07	0,07
Etapa 3	800	0,10	0,10	0,10	0,10	800	0,10	0,10	0,07	0,07
Etapa 5	800	0,10	0,10	0,10	0,10	4500	0,02	0,03	0,07	0,07
Etapa 7	4000	0,10	0,10	0,10	0,10	4000	0,10	0,10	0,07	0,07
Etapa 8	800	0,10	0,10	0,10	0,10	4000	0,01	0,02	0,07	0,07

Las combinaciones de las 6 estructuras de red con los PA y PB resulta en un total de 12 configuraciones. Para cada una de ellas se efectuaron tres corridas y se seleccionaron las que arrojaron el mayor porcentaje de aciertos del clasificador final (Etapa 8), éstos se muestran en la Tabla 8 y Tabla 9 como así también los resultados obtenidos en las etapas previas.

Tabla 8 – Error de validación (SSEv y EPv) obtenido en cada etapa para las 6 estructuras de red utilizando PA. Se agregó una columna con el promedio de SSEv de las Etapas 1 a 5.

Estructura	PA					
	Etapa 1 (SSEv)	Etapa 3 (SSEv)	Etapa 5 (SSEv)	Promedio Et. 1-5	Etapa 7 (EPv)	Etapa 8 (EPv)
46+27+17+7	2,87	1,39	11,83	**5,36**	42,86	**36,98**
46+30+20+7	2,77	1,26	12,02	**5,35**	41,11	**33,33**
46+33+23+7	2,69	1,36	11,90	**5,32**	39,52	**34,92**
46+36+26+7	2,54	1,45	12,89	**5,63**	39,21	**36,82**
46+39+23+7	2,61	1,58	11,48	**5,22**	40	**35,08**
46+42+23+7	2,62	1,64	9,81	**4,69**	39,68	**33,97**

En la Figura 30 se puede ver mas claramente como influye en el resultado final el entrenamiento mediante PA o PB. Para todas las estructuras el desempeño del clasificador final fue mejor cuando se utilizaron PB, siendo el valor mas alto de 70,48% para 46+27+17+17.

Tabla 9 – Error de validación (SSEv y EPv) obtenido en cada etapa para las 6 estructuras de red utilizando PB. Se agregó una columna con el promedio de SSEv de las Etapas 1 a 5.

Estructura	PB					
	Etapa 1 (SSE)	Etapa 3 (SSE)	Etapa 5 (SSE)	Promedio Et. 1-5	Etapa 7 (EPv)	Etapa 8 (EPv)
46+27+17+7	2,34	0,97	2,78	**2,03**	38,89	**29,52**
46+30+20+7	2,23	1,01	2,77	**2,00**	39,37	**29,68**
46+33+23+7	2,23	1,14	2,40	**1,92**	40,16	**30,63**
46+36+26+7	2,17	1,13	3,37	**2,22**	38,89	**30,95**
46+39+23+7	2,11	1,20	3,72	**2,34**	37,94	**31,27**
46+42+23+7	2,01	1,31	2,45	**1,92**	39,05	**30**

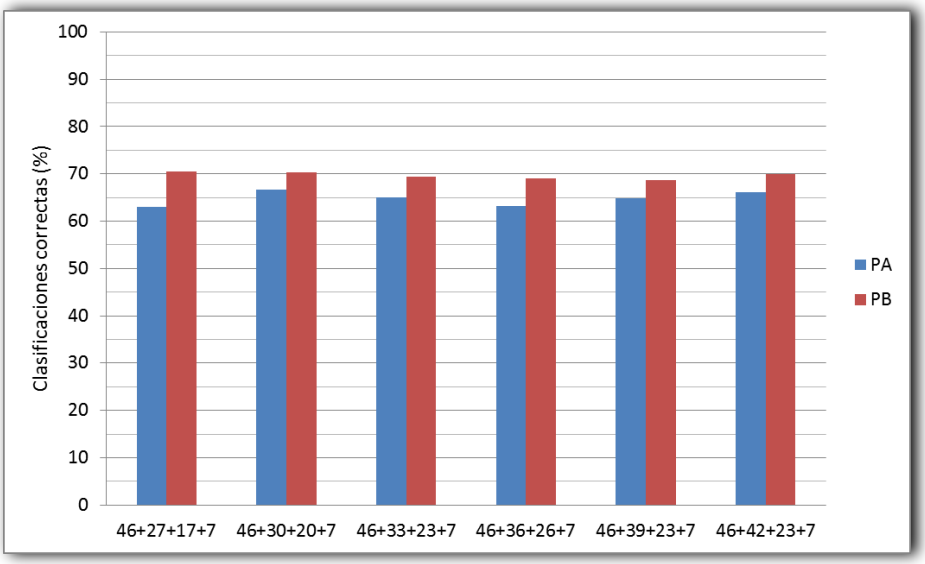

Figura 30 – Porcentaje de clasificaciones correctas de las 6 estructuras de clasificadores utilizando PA y PB para su entrenamiento.

Los valores presentadas hasta aquí sólo revelan los resultados finales, que si bien es lo que determina la elección de la mejor configuración, no nos da idea sobre cómo se desarrolló el aprendizaje de cada red. Para ello tomaremos el caso de la estructura 46+27+17+7 que arrojó el mejor resultado con PB y lo compararemos con los PA. En la Figura 31 se muestran las curvas de SEE y EP de la partición N°5 (tomada arbitrariamente). En las Etapas 1 y 2 las gráficas son similares entre PA y PB, decrecen monótonamente hasta un valor en el que se estabilizan. En la Etapa 5 se percibe una gran diferencia: para los PA las curvas de SEE encuentran un mínimo pero continuando el entrenamiento aumentan nuevamente, mientras que usando PB el SSEp decrece suavemente hasta un mínimo en el que permanece (notar que en la gráfica de PA el eje de ordenadas toma valores hasta 100 pero para PB solo hasta 10). Para la Etapa 7 no se presentan diferencias significativas . Por último, en la Etapa 8, puede notarse que con PB la curva resulta más suave (sin oscilaciones) debido a que **η** y **μ** son 10 veces más pequeños que PA (Tabla 7).

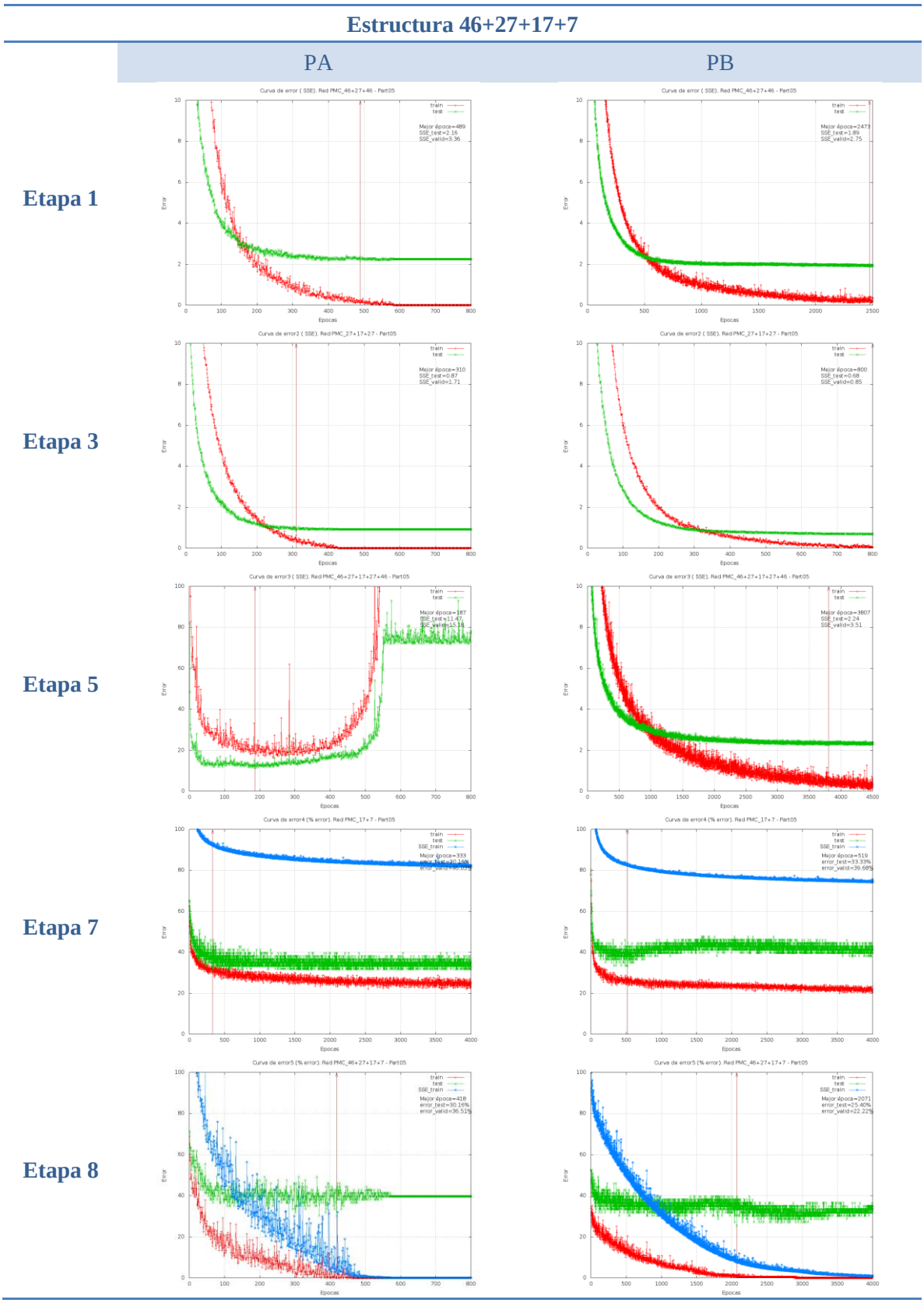

Figura 31 – Etapas de entrenamiento de la partición N°5 para el clasificador 46+27+17+7. Gráficas de SSE (Etapas 1, 3 y 5) y EP (Etapas 7 y 8), en rojo las del conjunto de entrenamiento y en verde las del conjunto de prueba. En las Etapas 7 y 8 se agregó la curva de SSE de entrenamiento (en celeste). La línea vertical sobre las gráficas indica la época en que se produjo el menor SSE de prueba.

Se comparará a continuación el método propuesto frente al entrenamiento estándar, es decir, aplicando el ARPM directamente al clasificador con pesos inicilizados al azar. Para tal fin se construyeron las mismas 6 estructuras de redes y cada una de ellas fue entrenada de la manera estándar. Los parámetros del algoritmo se seleccionaron, como antes, para lograr gráficas de error suaves, y se estableció $\boldsymbol{\eta}=\boldsymbol{\mu}=\mathbf{c}=\mathbf{e}_{imax}=0{,}05$. En la Tabla 10 se exponen los resultados (porcentaje de clasificaciones correctas del conjunto de validación) para las 6 estructuras entrenadas por ambos métodos. De forma mas clara puede verse en la Figura 32 la representación de los mismos datos en un gráfico de líneas.

Tabla 10 – Resultados de las 12 configuraciones experimentadas (porcentaje de clasificaciones correctas del clasificador final)

Estructura	Método Estándar	Método Propuesto	
		PA	PB
46+27+17+7	68,57	63,02	70,48
46+30+20+7	68,10	66,67	70,32
46+33+23+7	69,20	65,08	69,37
46+36+26+7	68,25	63,18	69,05
46+39+23+7	68,41	64,92	68,73
46+42+23+7	67,94	66,03	70,00

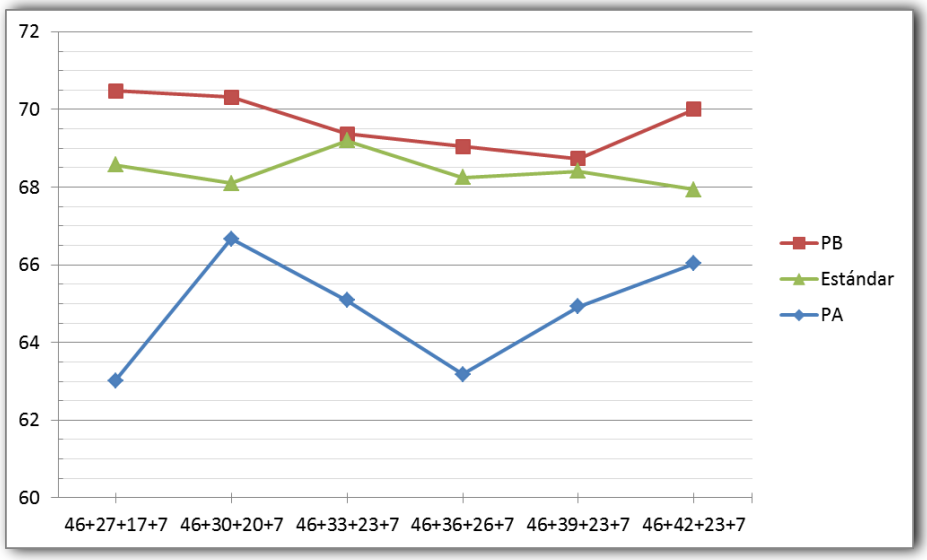

Figura 32 – Gráficas comparativas entre el entrenamiento estándar y el método propuesto (PA y PB). Valores tomados de la Tabla 10.

Variante del método propuesto

En la Etapa 1 del método propuesto, el autoencoder $\mathbf{i+h_1+i}$ tiene función de activación lineal en su capa de entrada pero su capa de salida es logística (Figura 22). Entonces, con la idea de poder representar de mejor manera la "simetría" del autoencoder se propuso que la capa de salida sea también lineal. De este modo se espera que pueda reconstruirse de mejor manera la entrada, con ello disminuir el error en dicha etapa y favorecer el desempeño del clasificador. Para ello se realizaron similares experimentos a los detallados anteriormen-

te (utilizando las mismas estructuras de red y parámetros de entrenamiento) pero contemplando esta modificación. Para diferenciarlos, llamaremos MP1 al método propuesto original y MP2 al que posee dicha variante en la Etapa 1. Dado que los PB son los que permitieron mejores resultados, solos presentaremos éstos para el MP2 (Tabla 11). Si comparamos con la Tabla 9, el error cometido en la Etapa 1 aumentó 3 a 4 veces para todas las estructuras de red cuando se aplicó el MP2, pero llamativamente fue menor (la mitad aproximadamente) en la Etapa 2. Los promedios de error de las Etapas 1, 3 y 5 son mayores en los 6 casos con el MP2.

Tabla 11 - Error de validación (SSEv y EPv) obtenido en cada etapa para las 6 estructuras de red utilizando el MP2 con PB. Se agregó una columna con el promedio de SSEv de las Etapas 1 a 5.

Estructura	MP2 (PB)					
	Etapa 1 (SSE)	Etapa 3 (SSE)	Etapa 5 (SSE)	Promedio Et. 1-5	Etapa 7 (EPv)	Etapa 8 (EPv)
46+27+17+7	6,09	0,67	3,59	**3,45**	41,43	**30,95**
46+30+20+7	8,66	0,81	3,75	**4,41**	42,06	**32,06**
46+33+23+7	6,77	0,72	3,54	**3,68**	44,45	**31,75**
46+36+26+7	7,34	0,81	3,81	**3,99**	44,92	**31,27**
46+39+23+7	8,21	0,74	4,67	**4,54**	44,60	**30,95**
46+42+23+7	7,43	0,76	3,52	**3,90**	44,29	**30,63**

Tomando los resultados de validación (EPv) comparamos los métodos MP1 y MP2 para determinar como influye la modificación realizada al entrenar la primer capa sobre el desempeño del clasificador final. En la Figura 33 se presentan ambas gráficas y puede notarse que para la mayoría de las estructuras el MP2 no produjo mejores resultados, excepto para la 46+39+23+7 aunque solo fue de 0,30 puntos.

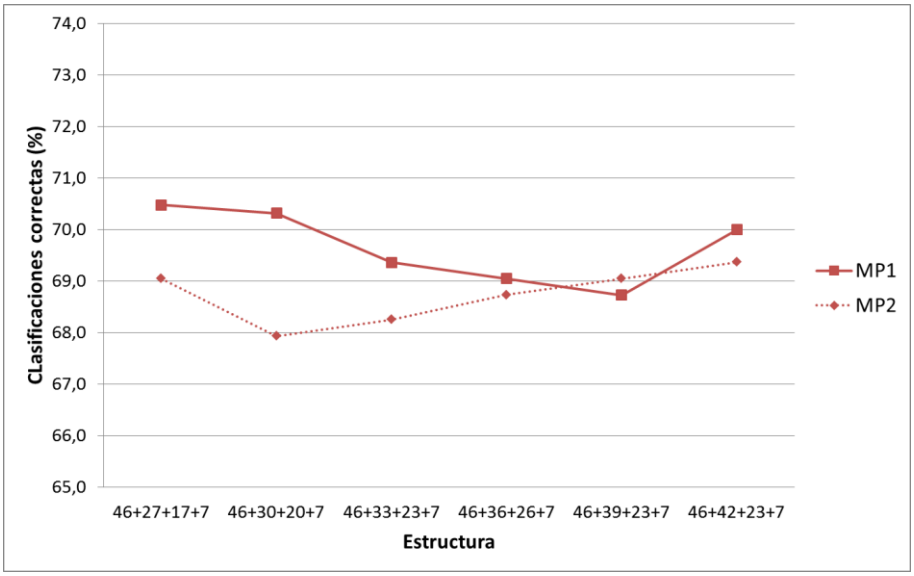

Figura 33 – Resultado comparativos entre desempeño de las 6 estructuras entrenadas por MP1 y MP2.

Capítulo 5. Discusiones y conclusiones

En el Capítulo anterior se presentaron los resultados para seis estructuras de clasificadores que fueron entrenados de tres maneras diferentes: por medio del método propuesto (MP1), utilizando una variación del mismo (MP2) y de la forma estándar. Además se probaron dos conjuntos de parámetros para el ARPM. Se aplicó a la clasificación de siete emociones de un corpus, tomando de cada elocución un conjunto de 46 características que integraban el vector de entrada a la red. Las mismas habían permitido alcanzar los mejores resultados utilizando otro tipo de clasificadores [Albornoz-Rufiner, 2011].

Discusiones

La elección y ajuste de de los parámetros requeridos para el algoritmo de entrenamiento no resultó una cuestión de menor importancia. Mas aún, esto agregó un grado de dificultad ya que no pueden estimarse a priori y solo por medio de exhaustivos experimentos basados en prueba y error se encontraron valores que posibilitaron mejores resultados. Para ejemplificar esto nos remitiremos a la Figura 34, donde se muestran las dos gráficas de error correspondientes a la Etapa 5 (extraídas de la Figura 31) del MP1. Utilizando PA las gráficas de SSEt (rojo) y SSEp (verde) decrecen rápidamente y permiten aproximarse a un mínimo (SSEp=11,47) antes de la época 200. Conforme continúa el entrenamiento los valores (grandes) de η y μ producen significativos cambios en los pesos ocasionando cierta inestabilidad. Luego de la época 500 esto se hace mas notorio y la red termina "alejándose" totalmente del mínimo. Lo contrario sucede con PB, la disminución del error se lleva a cabo lentamente, el SSEp alcanza un mínimo en el que se estabiliza (SSEp=2,24; 80% menor que con PA) en la época 3807. Aunque el SEEt continúa disminuyendo la capacidad de generalización de la red (SSEp) no lo hace, a partir de aquí se origina el inconveniente de sobreentrenamiento de la red.

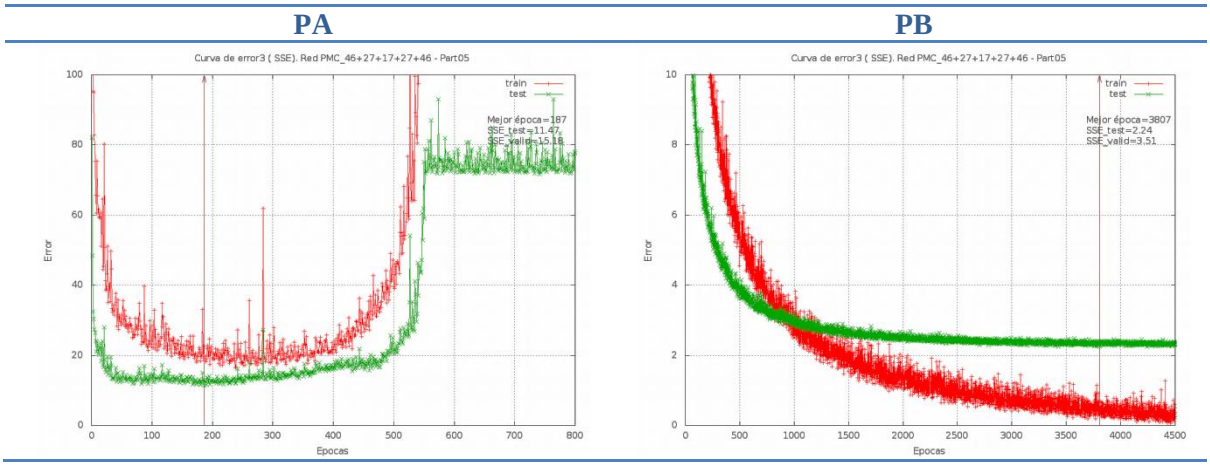

Figura 34 – Graficas de error de la Partición N°5, obtenidas en la Etapa 5 de entrenamiento del clasificador 46+27+17+7 por el MP1.

Un análisis similar al anterior (y referido a la misma estructura y partición) se puede hacer para la Etapa 8. La misma consiste en un tarea de clasificación y por lo tanto sería lógico entrenar la red tratando de disminuir EPt, pero el ARPM se basa en minimizar el SSEt (ya que es la esencia del algoritmo) e indirectamente EPt. Según se observa en la Figura 35, donde están graficadas las curvas de EPt (rojo), EPp (verde) y SSEt (azul), los PA producen curvas con muchas oscilaciones o picos en uno y otro sentido. Lo anterior se debe a que en cada época, si el SSEt es por exceso, el algoritmo trata de corregirlo pero la variación en los pesos es excesiva y termina produciendo un error por defecto, y viceversa. Si se piensa esto

en el espacio de pesos, el algoritmo va produciendo “saltos” sobre la superficie de error tratando de encontrar un mínimo local (SSEp=30,16), lo cual no resulta eficiente. Por el contacontario, los PB permiten que la red pueda acercase lentamente (sin grandes oscilaciones) a un un mínimo local “mejor” (SSEp=25,40).

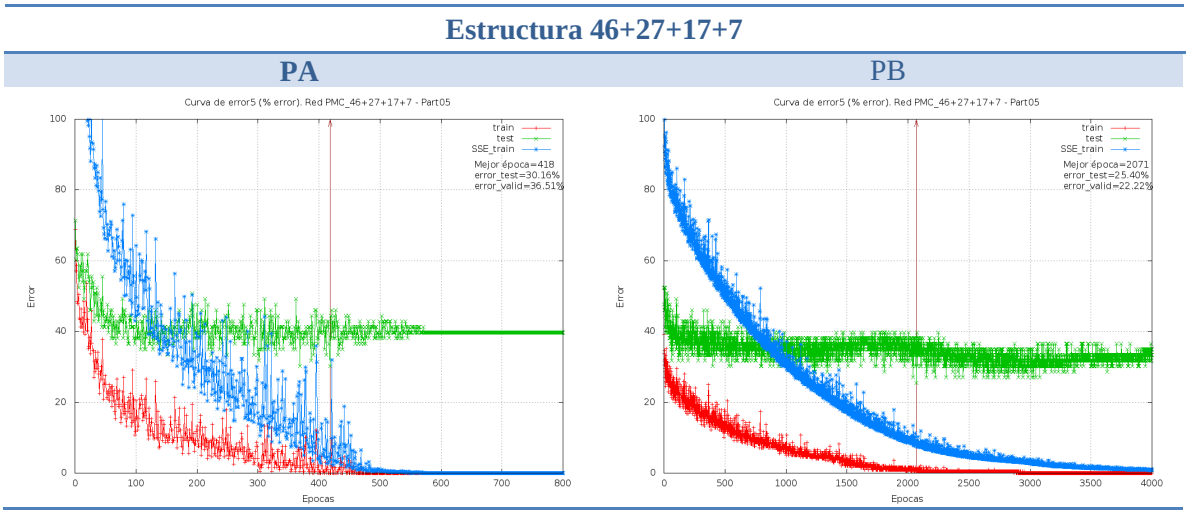

Figura 35 - Gráficas de error obtenidas en la Etapa 8 del entrenamiento del clasificador 46+27+17+7. Las mismas corresponden a la Partición N°5.

Otro punto que se debe analizar, y que tiene por propósito el método propuesto, es la inicialización de pesos en la última etapa, correspondiente clasificador final. En el capítulo anterior se vio que los resultados fueron mejores frente al entrenamiento estándar, sin embargo esto solo nos da un posible indicio de que el método cumplió su objetivo. Un indicador mas directo para ello es cuantificar el error que comete el clasificador en la primera época de entrenamiento. Intuitivamente podemos pensar que, si los pesos se encuentran inicialmente cerca de un mínimo local (lo que no ocurre necesariamente si son inicializados al azar) el error producido debería ser menor al comenzar el entrenamiento.

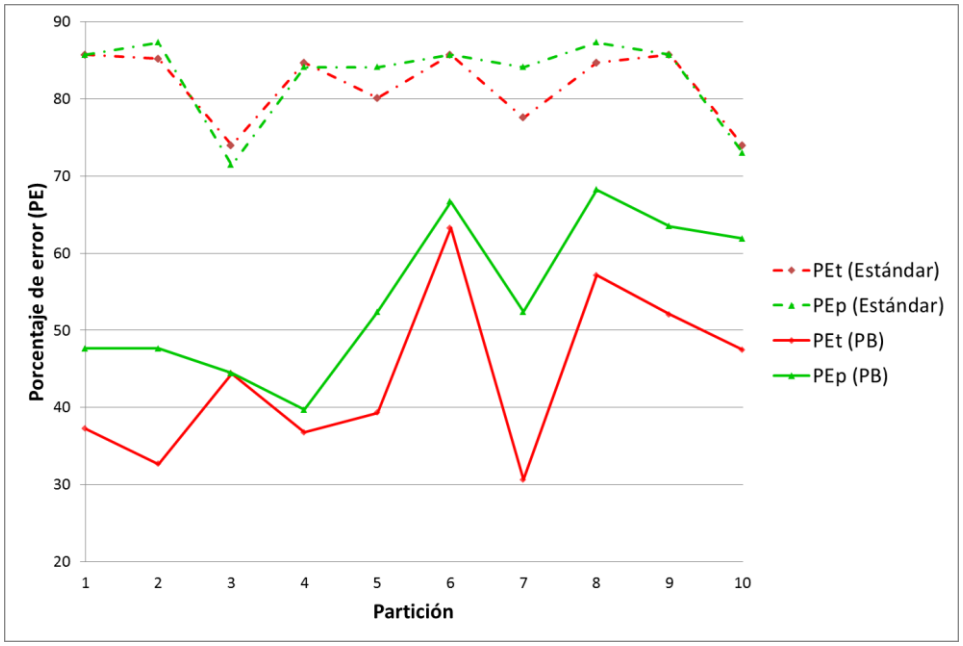

Figura 36 – Porcentaje de error para la primera época de entrenamiento en cada una de las 10 particiones. En línea continua se representa el método propuesto usando PB (estructura 46+27+17+7) y con trazo discontinuo el método estándar (estructura 46+33+23+7). El color rojo se usó para entrenamiento (PEt) y el verde para prueba (PEp).

En la Figura 36 se han graficado, para las 10 particiones, los errores cometidos por el clasificador (Etapa 8) al comenzar el entrenamiento. Las estructuras elegidas corresponden a las que dieron el mejor resultado: 46+33+23+7 para el método estándar y 46+27+17+7 para el MP1 usando PB, como se expuso en el capítulo anterior. Para ambos métodos las curvas de error de prueba (PEp) "acompañan" a las de entrenamiento (PEt), la cual resulta lógico ya que en la primer época la red no ha aprendido lo suficiente (solamente "vio" una vez cada patrón de entrenamiento) y comete casi el mismo error que cuando se le presenta un patrón nuevo (que no ha visto aún). Es importante resaltar que el error inicial (de entrenamiento y prueba) fue menor en todas las particiones cuando se usó el método propuesto lo cual pone en evidencia que efectivamente los pesos estaban inicializados cerca de una "buena solución".

Los experimentos realizados entrenando las redes mediante el MP2 no produjeron mejores resultados frente al MP1, como se vio en la Tabla 11. El mayor error promedio de las Etapas 1, 3 y 5 sugiere que la red no logró una mejor representación de la información en su estructura interna. Esto mismo puede corroborarse analizando los valores de la Etapa 7, que resultaron peores para MP2 y se debe a que esta última capa es entrenada con los patrones que fueron generados a partir de esa representación lograda en la red previamente. De todos modos las diferencias entre los resultados obtenidos por MP1 y MP2 no son realmente significativas como para determinar que un método es mejor que otro.

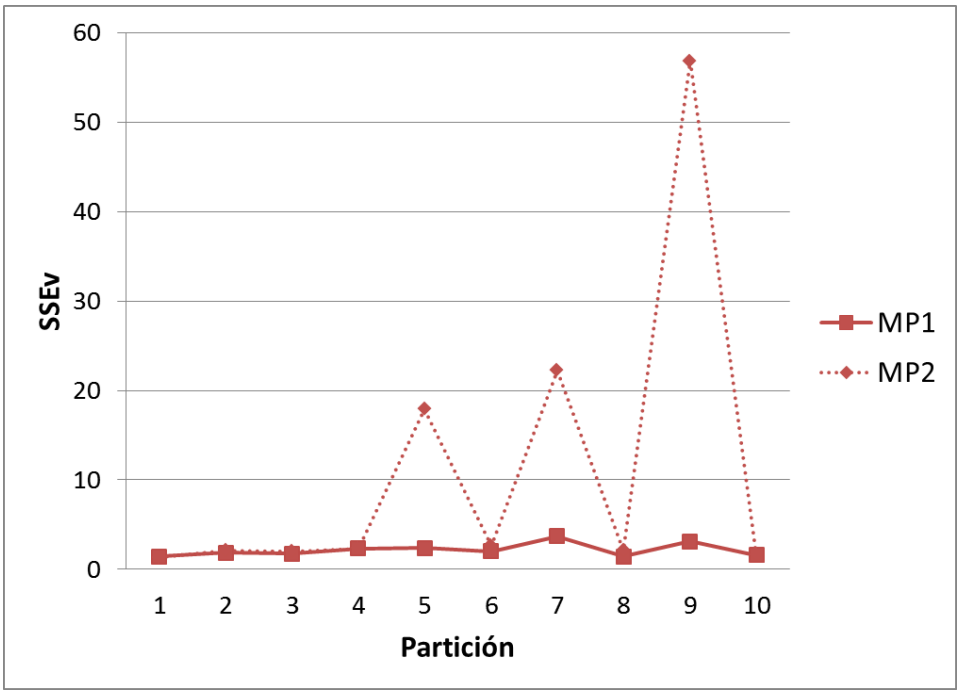

Figura 37 – Error de validación (SSEv) para cada partición producido en la Etapa 1, correspondiente a la estructura 46+36+26+7 entrenada por MP1 y MP2

Se puede profundizar un poco mas y evaluar que diferencias se producen en la Etapa 1 según se aplique el MP1 o MP2. En la Figura 37 se presenta el SSEv producido al entrenar la Etapa 1 del clasificador 46+36+26+7 por ambos métodos. Se puede percibir de forma clara que el SSEv en la mayoría de las particiones es similar, pero en tres de ellas (particiones 5, 7 y 9) se producen grandes diferencias que desvían el promedio hacia un valor mayor. En la Etapa 2 se produce un comportamiento similar y para las mismas particiones, pero en la Etapa 3 la red logra corregir estos desbalances y es por ello que los resultados finales (Etapa 8) terminan siendo similares. Se debe aclarar que esto no es un sesgo producido por desbalance en las particiones, ya que para otras estructuras ocurre en particiones diferentes a éstas.

Conclusiones y trabajo futuro

En este trabajo se ha presentado un método novedoso para el entrenamiento de estructuras de red profundas (con 2 o mas capas ocultas). Dicho método se aplico al entrenamiento de un clasificador profundo constituído por un PMC cuya tarea fue clasificar 7 emociones de un corpus. Se logró el mejor desempeño de 70,48% para el clasificador 46+27+17+7 lográndo mejorar los resultados frente a los obtenidos en trabajos previos [Albornoz-Rufiner, 2011], donde se obtuvo 66,83% utilizando también un PMC.

Las estructuras de red, parámetros utilizados y las consideraciones que se han propuesto en este trabajo representan un buen punto de partida para este nuevo método. Muchas simplificaciones se han realizado para poder enmarcar este trabajo en un Proyecto de Final de carrera, pero que amerita su continuidad para una investigación mas profunda. Una de las propuestas de trabajo a futuro es realizar experimentos con mayor cantidad estructuras, sin restricciones en la cantidad de neuronas en capas ocultas, y tratando de determinar el número óptimo de ellas que proveea una solución mejor. Otra consideración a tomar en cuenta en el entrenamiento del autocodificador profundo (Etapa 5) es la de aplicar “restricciones de raleza” (sparsity constraints, en inglés) en su capa media.

Como se ha propuesto en el Capítulo 6, la realización de una base de datos con características similares a la usada aquí permitiría aplicar el método propuesto y comparar resultados frente a dos idiomas. También se podría aplicar este método para tareas de reconocimiento de dígitos y reconocimiento de habla.

Capítulo 6. Análisis económico

La creación de bases de datos es una tarea que en la mayoría de los casos está ligada a usos específicos, es decir, se construyen para ser utilizadas en alguna aplicación que generalmente se engloba en el mismo proyecto de investigación. Esto no quita que puedan resultar útiles para otras aplicaciones, pero lo que se quiere resaltar es que raramente se considera crear una base de datos como objetivo final. Poder acceder a las mismas presenta ciertas ventajas:

- ⇒ Disponer de información necesaria rápidamente sin necesidad de tener que realizar la adquisición y procesamiento por cuenta propia
- ⇒ Muchas son ampliamente aceptadas y usadas en diversidad de trabajos, lo que permite la comparación de los resultados obtenidos.
- ⇒ Evitar incurrir en los costos que conllevaría crear una propia

Por otro lado, se presentan algunos inconvenientes como:

- ⇒ Pocas son accesibles gratuitamente
- ⇒ Que no cuente con la información suficiente (variables necesarias) para la aplicación en cuestión
- ⇒ Cantidad de datos insuficiente para realizar una evaluación estadística apropiada

En el Capítulo 2 se describió la base de datos utilizada en este trabajo (EMO-DB). Una de las principales caracterísitcas que se hubiese deseado es que la misma sea en español. Si bien las emociones constituyen un fenómeno innato al ser humano y no están afectadas por el idioma [Ekman, 1969], sería preferible trabajar en español y posibilitar entonces que futuras implementaciones del método propuesto aquí se trabajen en el mismo idioma. Otra observación para hacer sobre la EMO-DB es el desbalance en la cantidad de elocuciones por emoción, lo que obligó a reducir el número usado de ellas.

Debido a lo mencionado anteriormente y a la escasa disponibillidad de base de datos en español, se propone a continuación el desarrollo de un corpus que sirva para el REH y que pueda llevarse a cabo en la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos. El mismo estaría destinado no solo a utilizarse con el método propuesto aquí sino que quede disponible para otras aplicaciones que lo requieran en el ámbito académico o para uso externo previo acuerdo con la facultad. Se describirán cuales son las instalaciones, equipamiento, personal, procedimientos necesarios para ello y se hará un estimativo de los costos. Muchas de las consideraciones fueron tomadas del trabajo que explica como fue realizada la EMO-DB [Burkhardt, 2005].

Uno de los primeros interrogantes que se nos plantea es: ¿Cómo lograr representar fielmente las emociones? Para lo cual se tienen dos alternativas: de manera natural o actuada. La primera de ellas sería sin dudas la ideal ya que representa la emoción espontánea, sin ningún tipo de consideración o restricción, pero para poder obtenerlas se debe recurrir a fuentes como progrmas de TV, entrevistas o algún tipo de grabación que se haga en el “medio natural”. El inconveniente que presenta ello es que difícilmente pueda obtenerse una buena calidad de audio y existen muchas variables imposibles de controlar (ruidos, eco, distancia hablante-micrófono, etc) Además, la expresión de las personas puede estar afectada por el entorno con micrófonos, cámaras y exposición al público. La segunda alternativa consiste en simplemente solicitar a una persona que diga una frase “simulando” un estado emocional dado. El hablante, por lo tanto, utilizará ciertas “claves” percibidas en las conversaciones cotidianas para tratar de imitar esa emoción, lo cual es muy subjetivo y puede llegar a ser poco convincente. En el caso de que los hablantes sean actores profesionales dispondrán además de las técnicas aprendidas para lograr una mejor realización. Otro enfoque, que puede considerarse como intermedio entre los dos anteriores, sugiere que las

emociones puden ser inducidas a través de imágenes, narraciones y videos, logrando de éste modo situar a la persona en un estado emocional deseado. Esto posee las ventajas de lograr una emoción “natural” y a la vez poder realizar las grabaciones en con un ambiente y equipoas adecuados. Se debe preveer en estos casos el debido consentimiento de quiénes realizaran la prueba ya que pueden verses afectados por tales situaciones.

Como reglas generales para la construcción del corpus planteamos cuáles son las características deseables del mismo:

- ⇒ Idioma: español.
- ⇒ Contener las 6 emociones básicas y el estado emocional neutro [Ekman, 1969]. Esto además permitirá comparar resultados con el corpus EMO-DB.
- ⇒ Un razonable número de hablantes (varones y mujeres) que realicen todas las emociones permitiendo generalización sobre todo el grupo.
- ⇒ El contenido verbal de cada frase pronunciada debe ser el mismo para todos los hablantes de modo que permita comparabilidad frente a emociones y hablantes.
- ⇒ Las grabaciones deben hacerse con buena calidad de audio, minimizando el ruido de fondo.
- ⇒ Para poder realizar el filtrado inverso, las grabaciones deben realizarse dentro de una cámara anecoica (CA).
- ⇒ Lograr una cantidad de elocuciones balanceada para todas las emociones (similar cantidad para cada emoción)

Ítem	Función
Materiales	
2 mesas	Ubicar equipos utilizados
Cable	Alimentación y transm. de datos
Frases	Serán pronunciadas por los actores
Contenido multimedia	Inducción de emociones
Mano de obra	
10 actores (5 mujeres y 5 hombres)	Pronunciarán las frases a ser grabadas
2 Bioingenieros	Manejo de equipos, adquisición, procesamiento y análisis de datos. Deben poseer conocimientos sobre análisis y procesamiento de señal de habla.
Fonoaudiólogo	Diagnóstico de voz sana en actores Controlar pronunciación
10 o más personas	Test de percepción
Equipamiento	
Micrófono	-
Brazo para micrófono	Aislar vibraciones del piso
Placa de sonido	-
PC1	Adquisición (aloja la placa de sonido)
PC2	Inducción de emociones
Cable	Alimentación y transmisión de datos
Cámara anecoica	Aislación de ruido externo y minimización de reverberancia

Materiales

Aquí se considerarán los elementos tales como: 2 mesas (1 dentro de la CA) destinada a situar las PC y otros equipos, 4 sillas (2 dentro del la CA), brazo para micrófono (posibilita amurar a pared o techo para evitar captar vibraciones a través del piso) y los cables necesarios para la alimentación y conexión entre equipos.

Mano de obra

Para realizar las elocuciones se buscarán actores (posiblemente de zonas cercanas) por contacto directo o pudiendo hacer solicitudes a través de medios de comunicación. Se seleccionarán 5 masculinos y 5 femeninos para realizar el corpus. Los bioingenieros serán los encargados del montaje de los equipos, ajuste y control de su correcto funcionamiento durante las pruebas. Además deberán realizar la adquisición y procesamiento de las señales. Un fonoaudiólogo sera el encargado de determinar que los actores no presenten patologías en su voz y estará presente durante las grabaciones para verificar una pronunciación correcta de los actores. Para realizar el test de percepción sobre todas las elocuciones se formará un "jurado" compuesto por 10 o más personas (voluntarios, no necesariamente deban ser profesionales) quienes deberán clasificar y valorar las emociones a través de un criterio preestablecido.

Equipamiento

⇒ Micrófono: Shure MS58, de tipo unidireccional, con filtro para evitar ruidos por viento o aliento, posee respuesta en frecuencia adaptada a las voces (con rango medio y atenuación de bajos).

⇒ Placa de sonido: debería contar con un conversor ADC con más de 16 bits (el mercado ofrece placas de 20 y 24 bits, por ejemplo), esto es para ofrecer un buen rango dinámico (excursión de la señal) sin peligro de "recortes" en picos ("clipping" en inglés) y permitir luego del post procesamiento un señal sin degradación de al menos 16 bits (calidad CD y placas de sonido estándar). Propuesta: ASUS Xonar Essence ST (conexión PCI)

⇒ PC para adquisición: estará fuera de la CA y recibirá la señal del micrófono, será la que contenga la placa de sonido y deberá cumplir los requisitos de sistema que demande la misma.

⇒ PC para inducción de emociones: deberá estar fuera de la CA pero conectada a un monitor y parlantes que se encuentren dentro, la misma servirá para reproducir contenido multimedia hacia el actor antes de realizar la grabación. Esto servirá para ayudar a que logre el estado emocional correspondiente.

⇒ Cámara anecoica: se utilizará la que dispone actualmente la FIUNER. Posee 3,75 x 3 m, constituída por paredes dobles (externa de ladrillo común e interna de ladrillo cerámico hueco), la pared interna se encuentra recubierta con una malla metálica puesta a tierra. Se logró una cámara de aire mediante colocación de placas de yeso separadas 5cm de la pared interior. La cara interna de esas placas, techo (flotante) y piso están revestidos con 3 capas de lana de vidrio de distintas densidades y 8 cm de espesor cada una para reducir la reverberancia. Los datos fueron extraídos de [Aronson et al, 2006]

Procedimiento

Selección de actores

La búsqueda de actores puede realizarse contactando grupos de teatro, personas que trabajan en el medio o mediante publicación en medio de comunicación, estableciendo una cita donde se realizará la preselección.

Primeramente se le explicará a cada actor la tarea a realizar para lo cual debe dar su consentimiento. Luego, un fonoaudiólogo hará una evaluación de cada actor para determinar si presentan alteraciones funcionales de la voz. Para ello pueden utilizarse dos tipos de análisis:

- ⇒ Informal o subjetivo, consiste en la observación, palpación, escucha y auscultación de los sistemas muscular (tronco superior), respiratorio y fonatorio. Se identifican el tipo y modo respiratorio y las cualidades y modalidades fonatorias. No requiere de equipos pero si pericia por parte del profesional.
- ⇒ Formal u objetivo, se realiza utilizando programas de PC como el ANAGRAF y PRAAT. Se requiere de una PC con el programa instalado y micrófono.

Posteriormente, los individuos con voz sana harán una realización de cada emoción (pronunciando la misma frase con las 7 emociones) que no necesariamente debe realizarse en la CA, pero si será grabado y almacenado en una PC. El siguiente paso será llevado a cabo por un grupo de personas (no necesariamente debe ser el mismo que para el test de percepción pero es conveniente que no hayan participado en las grabaciones) que deberán escuchar las grabaciones y seleccionar 10 actores (5 de cada sexo) de acuerdo a la naturalidad y reconocibilidad de la emoción en sus realizaciones. Dichos actores serán nuevamente contactados, informados y citados para realizar las grabaciones (se debe tener en cuenta que demandará 2hs aproximadamente para cada actor, por lo que no todos deberán concurrir el mismo día).

Montaje de equipos

Dentro de la CA deberá ubicarse el micrófono, de forma colgante para evitar que capte las vibraciones a través del piso, y conectado a la entrada de la placa de sonido de la PC de adquisición (situada fuera de la CA). La ubicación del micrófono deberá ser tal que permita captar la voz para una persona de pie. También se deben colocar un monitor y parlantes dentro de la CA conectados a la segunda PC (fuera de la CA) que serán destinados a inducir las emociones en el actor antes de la realiazación de cada emoción.

Adquisición de señales

Para cada uno de los actores seleccionados se realizará lo detallada a continuación. Se le explicará la metodología a seguir y solicitará el consentimiento sobre la realización de la prueba, luego ingresará en la CA donde se le indicará a que distancia hablar del micrófono (se deberá poder ajustar la posición del micrófono para que resulte cómodo al actor) y otras consideraciones como: no gritar (ej: enojo) o hablar demasiado suave (ej: tristeza) ya que introduciría un sesgo para estas emociones, podrán realizar expresiones corporales siempre y cuando no perturbe la señal captada. Luego de realizar pruebas para ajustar la ganancia del mircófono, se le presentará el contenido audiovisual al actor y se tomará el tiempo que desee para “preparar” la realización de la emoción, cuando lo desee hará una seña para comenzar la grabación en la que deberá pronunciar las 10 frases (con el el intervalo que él disponga entre cada una). El mismo procedimiento se aplicará para las restantes 6 emociones (para el caso del estado emocional neutro no será necesario el contenido audiovisual).

Si durante la realización de una elocución se producen defectos (ruido, mala pronunciación, etc) advertidos oportunamente, debe solicitarse al actor que se efectúe nuevamente. Teniendo en cuenta que cada actor debe pronunciar 70 frases (lo cual puede tornarse tedioso) puede dividirse el proceso en 2 etapas, alternando entre 2 o mas actores. Sin embargo debe tenerse precaución para evitar confusiones en el registro y etiquetado de las señales.

El registro deberá hacerse en formato “.wav” utilizando una frecuencia de muestreo de 48 KHz, una profundidad (bit depth, en inglés) de 20 o 24 bits y de tipo monoaural. Se recomienda que el nombre de archivo correspondiente a la grabación de cada frase contenga al menos la siguiente información: identificador del actor, identificador de cada frase, identificador de emoción. Por ejemplo, “A02_F10_E.wav” correspondería a Actor 2, Frase 10, Enojo.

Como primer paso, primordial antes de cualquier modificación se debe realizar una (al menos) copia de seguridad de toda la información en otra unidad de disco duro (externo o de otra PC). El espacio en disco que demandará puede calcularse aproximadamente multiplicando: frecuencia de muestreo (48 kHz), profundidad de bits (24), duración promedio de cada elocución (3 s) y el total de elocuciones (700 = 10 actores x 10 frases x 7 emociones), obteniéndose 288,4 MB.

Test de percepción

Tiene por objetivo someter a evaluación cada elocución para lo cual un jurado deberá decidir a qué emoción corresponde y juzgar su naturalidad. Dicho jurado estará compuesto por 10 o mas personas, y cada una deberá estar frente a una PC en la cual se le mostrará una tabla (debidamente confeccionada) sobre la que deberá asentar su voto. Se les permitirá escuchar una por una las elocuciones (sólo una vez) en forma aleatoria, seguidamente deberán marcar en un casillero (o escribiendo la letra según lo acordado en este trabajo) a qué emoción corresponde y estableciendo un valor de naturalidad en una escala de 1 a 10 (10 correspondería a una emoción muy convincente).

Una vez obtenidos todos los resultados se promediarán para la misma elocución y seleccionarán solo aquellas que cumplan un determinado criterio, por ejemplo, que tenga clasificaciones correctas de más de 80% (8 de 10 personas la clasificaron correctamente) y naturalidad mayor o igual a 7. Aquí se puede originar el desbalance en la cantidad de elocuciones por emoción ya que el enojo es notoramiente mas fácil de identificar, luego le siguen asco, y estado neutral (según se vió en la Tabla 3). Surgen dos alternativas para lograr una distribución homogénea: truncar todas a la menor cantidad de ellas o considerarlo al momento de la adquisición y hacer mas realizaciónes para emociones restantes.

Procesamiento de señales seleccionadas

El procesamiento mínimo necesario será normalizar el nivel de intensidad (volumen) de cada elocución según el pico (excursión) máximo y eliminar los silencios al principio y final. Se puede aplicar filtrado si fuera necesario. Dependiendo de qué información se desee adicionar (sílaba acentuada, identificación y etiquetado de fonemas, frecuencia fundamental máxima y mínima, histogramas, etc) deberá hacerse el análisis necesario.

Disposición final del corpus

Una vez rotuladas todas las elocuciones y añadida la información deseada, se podrán almacenar en un CD (pero se debe considerar que solo admite audio en 16 bits y cuya frecuencia de muestreo sea de 44,1 KHz), en unidades de disco rígido o bien alojarlo en un sitio de internet. Además se debe adjuntar un informe técnico donde se detallarán todos los pasos realizados para la realización del corpus.

Consideraciones y cálculo de presupuesto

Se expondrán a continuación los aspectos reelevantes que servirán para determinar los costos asociados a la realización del corpus según los pasos detallados anteriormente.

Respectos al equipamiento que deberán adquirirse, se realizó un búsqueda de precios en internet a través del sitio *mercadolibre.com.ar* y casas de venta de accesorios de sonido (micrófono, brazo y cable). Se consideró que ambas PCs puede ser aportadas por la cátedra o laboratorio que lleve a cabo la realización de la base de datos, dado que no se requiere de características especiales (solo de conexión PCI para la placa de sonido).

A través del sitio web de la Asociación Argentina de Actores se consultó la escala salarial referida a actores teatrales en concepto de “ensayos” y la Convención Colectiva de Trabajo

N° 307/73 vigentes. De allí se extrajo el monto mensual y se dedujo el proporcional para obtener el costo aproximado que demandará la contratación de los mismos por 3hs.

Se consultó a la Fonoaudióloga Mariangel Albornoz Laferrara, M.P. 238 sobre el costo de realización de la evaluación funcional de la voz mediante análisis subjetivo que pueda llevarse a cabo en el lugar (como debe realizarse a todos los actores que se presenten en primera instancia se tomó una cantidad arbitraria de 15). Además se contempla la presencia de dicha profesional durante las grabaciones para indicar defectos en la pronunciación que requieran repetir la elocución lo cual se consideró como una entrevista con el paciente (a fines de cálculo).

Para las tareas de conexión y puesta a punto de equipos, manejo de software de adquisición, procesamiento y análisis de señales, rotulado y disposición final del corpus se considerarán un Bioingeniero junior con cocimiento básico del tema y un Bioingeniero senior con título de postgrado que se encuentre trabajando en el área de procesamiento del habla. Mediante consulta al Colegio de Ingenieros Especialistas de Entre Ríos (CIEER) se obtuvo el salario mínimo mensual que se calcula a través de una unidad llamada *Ingenio.* La remuneración es entonces de 80 Ingenios, y su valor (que depende de la categoría) se tomó según la actualización correspondiente a Noviembre de 2013. Considerando 45hs de trabajo semanales es estimó el monto a percibir por hora de trabajo, los valores se vuelcan en la tabla siguiente:

Categoría	Ingenios Mensuales	Valor del ingenio	Monto mensual	Monto hora
Bioingeniero Junior	80	$ 100,00	$ 8.000,00	$ 44,44
Bioingeniero Senior	80	$ 150,00	$12.000,00	$ 66,67

Para el Bioingeniero Junior se estimó un total de 85hs que se distribuyen de la siguiente manera: 8hs para la selección de actores, 30hs de sesión de grabaciones, 40hs destinadas a procesamiento y etiquetado de datos y las restantes son consideras para la conexión de equipos, puesta a punto, entre otros. Las horas de trabajo para el caso del Bioingenierio Senior se consideran menores ya que no necesariamente deberá estar en todo el proceso, pero sí para dar las indicaciones iniciales y estableces ciertos criterios acordes a su experiencia y conocimiento.

Finalmente, tomando en cuenta las consideraciones realizadas se elaboró un presupuesto detallado con los ítems mencionados anteriormente que se presenta en la Tabla 12. El plazo de duración de este proyecto se estima en 4 semanas. La primera estaría destinada a la convocatoria y selección de actores. En la semana siguiente se llevarían a cabo las grabaciones y las dos semanas restantes se destinarán a la evaluación, procesamiento y disposición final de los datos.

Fuente de financiamiento

Este proyecto puede llevarse a cabo a través del Programa de Investigación y Desarrollo (PID) de la UNER. Se debe tener en cuenta que la presentación de dicho proyecto debe hacerse desde un grupo de investigación perteneciente a la facultad. Esta fuente de financiamiento solo se puede destinar para bienes de consumo, servicios no personales y equipamiento pero no contempla salarios ni becas. Una propuesta del autor, es que la realización de esta base de datos sea objetivo de un Proyecto Final de Bioingeniería para lo cual puede solicitarse una de las “Becas para Auxiliares de Investigación” o “Becas Estímulo a las Vocaciones Científicas” (CIN) de la UNER.

Tabla 12 – Prespupuesto detallado para la realización de la base de datos

Ítem	Cantidad	Valor unit.	Valor total
Equipos y Materiales			
Micrófono (Shure SM-58)	1	1.529,00	1.529,00
Brazo para mic. (Servidata S57 de 1,2m)	1	1093,95	1093,95
Cable mic. (XLR-3/Plug 6,5mm de 4,5m)	1	69,00	69,00
Placa de sonido (ASUS Xonar Essence STX)	1	1.879,00	1.879,00
Cable p/monitor (Vga-Vga de 5m)	1	110,00	110,00
Subtotal Equipos y Materiales			**4.680,95**
Mano de obra			
Actores (3hs c/u)	10	116,17	1.161,70
Fonoaudiólogo (Análisis de voz)	15	250,00	3.750,00
Fonoaudiólogo (presencia en cada grabación)	10	140,00	1.400,00
Bioingeniero junior	85 hs	44,44	3.777,40
Bioingeniero senior	40 hs	66,67	2.666,68
Subtotal Mano de obra			**12.755,78**
TOTAL			**17.436,73**

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
N. Cibau, H. L. Rufiner & E. M. Albornoz; "Reconocimiento de emociones en el habla mediante autocodificadores profundos (Undergraduate project)"
Facultad de Ingeniería - Universidad Nacional de Entre Ríos, 2013.

Apéndice

Simulador de Redes Neuronales Stuttgart

Para llevar a cabo la implementación de las redes neuronales y realizar los procesamientos requeridos en una computadora o clúster se utilizó el Simulador de Redes Neuronales Stuttgart (*Stuttgart Neural Network Simulator* o SNNS, en inglés), un software desarrollado por el Instituto de Sistemas de Alto Rendimiento Paralelo y Distribuido de la Universidad de Stuttgart. El SNNS es de descarga gratuita y consta de cuatro partes:

- ⇒ Simulador del kernel: opera con la estructura de datos interna de las redes permitiendo realizar cualquier operación sobre ellas.
- ⇒ Interfaz gráfica de usuario "XGUI": construida sobre el kernel, posibilita la creación, simulación, representación y operaciones sobre las redes a través de una interfaz gráfica.
- ⇒ Interfaz para ejecución por lotes "batchman": programa que brinda la posibilidad de ejecutar los procesos para crear y entrenar redes en segundo plano.
- ⇒ Compilador de redes "snns2c": herramienta que permite convertir una red entrenada en SNNS a un archivo en lenguaje C que puede ser luego compilado en un ejecutable o implementado como una función dentro de una aplicación hecha en C.

Dado que los procesos de entrenamiento y otras operaciones efectuadas sobre las redes requerían muchas horas de ejecución en CPU, se utilizó el programa "batchman". Consiste en un intérprete de instrucciones que utiliza un lenguaje particular basado en AWK, Pascal, Modula-2 y C. Las funciones que es capaz de ejecutar el batchman se pueden dividir en cuatro grupos, dependiendo de su finalidad. En la Tabla 13 se encuentran listadas las usadas en el presente trabajo y algunas variables del sistema que pueden ser accedidas.

Tabla 13 – Funciones y variables del batchman

Funciones para establecer parámetros del SNNS	
setInitFunc()	Selección de función para inicializar pesos y parámetros
setLearnFunc()	Selección de algoritmo de entrenamiento y parámetros
setUpdateFunc()	Selección de modo de actualización de pesos
setShuffle()	Activa/Desactiva selección aleatoria de patrones
Funciones referidas a redes neuronales	
loadNet()	Carga una red existente (.net)
saveNet()	Guarda la red actual (.net)
saveResult()	Guarda archivo de resultados (.res)
initNet()	Inicializa los pesos de la red
trainNet()	Entrena una época (Una vez todos los patrones)
resetNet()	Establece todos los pesos en cero
testNet()	Genera las salidas de la red para un patrón de entrada
Funciones referidas a patrones	

loadPattern()	Carga archivo de patrones (hasta 5 archivos)
setPattern()	Selección del archivo de patrones a utilizar
Funciones especiales	
execute()	Ejecuta un comando o programa
print()	Imprime en pantalla
exit()	Sale de batchman
Variables internas de batchman	
SSE	Suma de los cuadrados de error en las salidas
CYCLES	Ciclos/Épocas entrenados hasta el momento
SIGNAL	Valor entero de una señal capturada durante la ejecución

Además fue necesario un kit de herramientas con ejecutables precompilados (incluidos en el SNNS) que son "invocados" desde el batchman; aquellos usados en este trabajo se muestran en la Tabla 14.

Tabla 14 – Descripción de los ejecutables que fueron usados

Nombre del ejecu-	Descripción
analyze	Calcula los porcentajes de clasificación a partir de los archivos de resultados generados durante el entrenamiento.
ff_bignet	Permite crear redes de tipo feedforward.
mkhead	Genera el encabezado de los archivos de patrones.
linknets	Une dos o más redes.

Bibliografía

[1] DARPA, DARPA Neural Network Study, AFCEA Intl Pr, 1988.

[2] A. Dawel et.al., "Not just fear and sadness: Meta-analytic evidence of pervasive emotion recognition deficits for facial and vocal expressions in psychopathy," *Neuroscience & Biobehavioral Reviews,* Vols. 36, Issue 10, 2012.

[3] A. Zell et. al., "Stuttgart Neural Network Simulator. User Manual, version 4.2," *Institute for Paralell and Distributed High Performance Systems (University of Stuttgart), Wilhelm-Schickard-Institute for Computer Science (University of Tubingen),* 2000.

[4] A. Stuhlsatz et. al., "Deep Neural Networks for Acoustic Emotion Recognition: Raising the Benchmarks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.

[5] L. Devillers y L. Vidrascu, "Real-life emotions detection with lexical and paralinguistic cues on human-human call center dialogs," in *Interspeech*, Pittsburgh, Pennsylvania, 2006.

[6] P. Ekman, E.R. Sorenson y W.V. Friesen, "Pan-cultural elements in facial displays of emotions," *Science,* vol. 164, no. 3875, pp. 86-88, 1969.

[7] F. Burkhardt et. al., "A Database of German Emotional Speech," *9th European Conference on Speech Communication and Technology - Interspeech 2005,* p. 1517–1520, 2005.

[8] J.A. Freeman y D.M. Skapura, Redes neuronales. Algoritmos, aplicaciones y técnicas de programación, Wilmington, Delaware, E.U.A.: Versión en español por Rafael García-Bermejo Giner. Addison-Wesley Iberoamericana S.A., 1993.

[9] G.E. Hinton, S. Osindero, Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation,* vol. 18, no. 7, pp. 1527-1554, 2006.

[10] G.E. Hinton y R.R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science,* no. 28, pp. 504-507, 2006.

[11] G.E. Hinton et. al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 82-97, 2012.

[12] M. Guzman, "Acústica del tracto vocal," 2010.

[13] I.R. Murray y J.L. Arnott , "Applying an analysis of acted vocal emotions to improve the simulation of synthetic speech," *Computer Speech & Language,* Vols. 22, Issue 2, p. 107–129, 2008.

[14] J. Adell Mercado, A. Bonafonte Cávez, D. Escudero Mancebo, "Analysis of prosodic features: towards modelling of emotional and pragmatic attributes of speech," *XXI Congreso de la Sociedad Española. Procesamiento del Lenguaje Natural - SEPLN 2005,* no. 35, p. 277–284, 2005.

[15] J. R. Deller Jr., J.G. Proakis y J.H. Hansen, Discrete-Time Processing of Speech Signals, Upper Saddle River, NJ, USA: Prentice Hall PTR, 1993.

[16] L. Devillers y L. Vidrascu, "Real-Life Emotion Recognition in Speech," *Speaker Classification II: Selected Projects. Lecture Notes in Computer Science,* vol. 4441, pp. 34-42, 2007.

[17] M. V. Llamazares, "Fonología y Fonética. Apunte de cátedra - Univ. de León, España.," 2012.

[18] A. Mohamed et. al., "Deep Belief Networks using Discriminative Features for Phone Recognition," in *ICASSP-2011, ISCA*, Portland, OR, USA, 2012.

[19] J. L. N. Mesa, "Procesador Acústico: el bloque de extracción de características," [Online]. Available: http://www.ulpgc.es/hege/almacen/download/25/25296/apuntesextraccioncaracterisitcas.pdf.

[20] C. Ortego Resa, "Tesis de grado: Detección de emociones en voz espontánea," *Universidad Autónoma de Madrid,* 2009.

[21] L. Rabiner y B-H Juang, Fundamentals of Speech Recognition, Prentice Hall PTR, 1993.

[22] F. J. H. Pericas, "Tecnicas de procesado y representación de la señal de voz para el reconocimiento del habla en ambientes ruidosos," 1993.

[23] R. Brückner y B. Schuller, "Likability Classification – A not so Deep Neural Network Approach," in *Proceedings INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association*, Portland, Oregon, USA, 2012.

[24] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review,* vol. 65, no. 6, pp. 386-408, 1958.

[25] K. R. Scherer, A blueprint for affective computing: a sourcebook, Oxford: Oxford University Press, 2010.

[26] B. Schuller, S. Steidl y A. Batliner, "The INTERSPEECH 2009 Emotion Challenge," *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association,* pp. 312-315, 2009.

[27] S. Yildirim, S. Narayananb y A. Potamianosc, "Detecting emotional state of a child in a conversational computer game," *Computer Speech & Language,* vol. 25. Issue 1, p. 29–44, 2011.

[28] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall, 1998.

[29] E. M. Albornoz, D. H. Milone y H. L. Rufiner, "Spoken Emotion Recognition using Hierarchical Classifiers," *Computer Speech and Language,* vol. 25, pp. 556-570, 2011.

[30] S. Young et. al., The HTK Book (for HTK Version 3.1), Cambridge University Engineering Department, England, (Dec. 2001).

[31] D. Tacconi et. al., "Activity and emotion recognition to support early diagnosis of psychiatric diseases," *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare 2008, Tampere, Finland,* pp. 100-102, 2008.

[32] V. Vinhas, L.P. Reis y E. Oliveira, "Dynamic Multimedia Content Delivery Based on Real-Time User Emotions. Multichannel Online Biosignals," *International Conference on Bio-inspired Systems and Signal Processing - Biosignals 2009,* pp. 299-304, 2009.

[33] G. Orr, N. Schraudolph y F. Cummins, "CS-449: Neural Networks. Lecture notes," Willamette University, Salem, Oregon, 1999.

[34] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning,* vol. 2, no. 1, p. 1–127, 2009.

[35] R. Yuvaraj, M. Murugappan y K. Sundaraj, "Methods and approaches on emotions recognition in neurodegenerative disorders: A review," *Industrial Electronics and Applications (ISIEA), 2012 IEEE Symposium on,* pp. 287-292, 2012.

[36] H. L. Rufiner, Análisis y modelado digital de la voz, Santa Fe: Ediciones UNL, 2009.

[37] K. P. Truong y D. A. van Leeuwen, "Automatic discrimination between laughter and speech," *Speech Communication,* vol. 49, no. 2, p. 144–158, 2007.

[38] J. L. Elman et.al., Rethinking Innateness: A Connectionist Perspective on Development, MIT press, 1996.

[39] G. Hinton, "Learning multiple layers of representation," *Trends in Cognitive Sciences,* vol. 11, no. 10, pp. 428-434, 2007.

[40] D.E. Rumelhart, G.E.Hinton y R.J. Williams, "Learning internal representations by error propagation," *Parallel distributed processing,* pp. 318-362, 1986.

[41] D. Randall Wilson y Tony R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural Networks,* vol. 16, no. 10, p. 1429–1451, December 2003.

[42] E. M. Albornoz, "Modelado de estructuras prosódicas para el reconocimiento automático del habla," Tesis doctoral, Santa Fe, Argentina., 2011.

[43] L. Aronson, P. Estienne, M. E. Torres, H. L. Rufiner, D. H. Milone & C. E. Martínez, "BEPPA: batería de evaluación de pacientes con prótesis auditiva," *Universidad Nacional de Entre Ríos, No. DNDA347522.,* 2006.

[44] M. El Ayadi, M. S. Kamel, F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition,* vol. 44, no. 3, p. 572–587, 2011.

[45] M. Chóliz, "Psicología de la emoción: el proceso emocional," 2005. [Online]. Available: http://www.uv.es/choliz/Proceso%20emocional.pdf.

[46] C. M. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee y S. Narayanan, "Emotion recognition based on phoneme classes," *Proc. ICSLP, Jeju, Korea,* pp. 889-892, 2004.

[47] D. Reynolds, T. Quatieri & R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process,* vol. 10, no. 1-3, pp. 19-41, 2000.

[48] D. Reynolds y C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Tranactions on Speech Audio Processing,* vol. 3, no. 1, p. 72–83, 1995.

[49] T. Nwe, S. Foo y L. De Silva, "Speech emotion recognition using hidden Markov models," *Speech Communication,* vol. 41, no. 4, p. 603–623, 2003.

[50] M. Slaney y G. McRoberts, "BabyEars: A recognition system for affective vocalizations," *Speech Communication,* vol. 39, no. 3-4, p. 367–384, 2003.

[51] B. Schuller, "Towards intuitive speech interaction by the integration of emotional aspects,"

2002 IEEE International Conference on Systems, Man and Cybernetics, vol. 6, p. 6, 2002.

[52] V. Hozjan y Z. Kacic, "Context-independent multilingual emotion recognition from speech signal," *International Journal of Speech Technology,* vol. 6, no. 4, pp. 311-320, 2003.

[53] V. Petrushin, "Emotion recognition in speech signal: experimental study, development and application," *Proceedings of the ICSLP 2000,* p. 222–225, 2000.