Comparación de algoritmos para reconocimiento de habla aislada independiente del hablante

Mariano Marufo da Silva, Claudio Verrastro, Juan Carlos Gómez

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Departamento de Ingeniería Electrónica, Av. Medrano 951 (C1179AAQ) Ciudad Autónoma de Buenos Aires, Argentina

mmarufodasilva@est.frba.utn.edu.ar

Recibido el 2 de noviembre de 2017, aprobado el 24 de enero de 2018

Resumen

Este trabajo describe la fundamentación teórica e implementación de un sistema de reconocimiento de habla aislada independiente del hablante usando Modelos Ocultos de Markov, Máquinas de Vectores de Soporte y Redes Neuronales Artificiales. La evaluación fue realizada utilizando un *corpus* multihablante compuesto por once palabras del español argentino, y su rendimiento en términos de porcentaje de reconocimiento fue comparado entre los tres métodos de clasificación implementados. La comparación fue efectuada tanto para condiciones ideales como también con tres niveles distintos de ruido de fondo. Los resultados muestran que para bajos niveles de ruido el sistema basado en HMM consigue el mejor rendimiento, mientras que para mayores niveles de ruido los sistemas basados en SVM y ANN superan al anterior.

PALABRAS CLAVE: RECONOCIMIENTO DE HABLA - MODELOS OCULTOS DE MARKOV - MÁQUINAS DE VECTORES DE SOPORTE - REDES NEURONALES ARTIFICIALES

Abstract

This work describes the theory and implementation of a speaker-independent, isolated speech recognition system using Hidden Markov Models, Support Vector Machines and Artificial Neural Networks. The evaluation was performed using a multi-speaker *corpus* composed by eleven words of the argentinean spanish and the performance of the three implemented methods was compared in terms of their recognition rates. The comparison was performed both for ideal conditions and with three different levels of background noise. Results show that for low noise levels, the HMM based system has the best performance, while for greater noise levels the SVM and ANN based systems perform better.

KEYWORDS: SPEECH RECOGNITION - HIDDEN MARKOV MODELS - SUPPORT VECTOR MACHINES - ARTIFICIAL NEURAL NETWORKS

Introducción

El estudio de Algoritmos de Reconocimiento Automático del habla (ASR) ha hecho que actualmente sea común para muchas personas realizar tareas como una búsqueda de información en la web o dar una instrucción a su teléfono celular por medio del habla.

Inicialmente se utilizaron técnicas de Dynamic Time Warping (DTW) (Alvarez, 2016), que con el paso del tiempo fueron reemplazadas por métodos de modelización del habla basados en Modelos Ocultos de Markov (HMMs). En los años 1980 se comenzó a investigar en métodos que reemplazaran a estos modelos por Redes Neuronales Artificiales (ANNs), y posteriormente, en los años 1990, por Máquinas de Vectores de Soporte (SVMs), obteniendo aún al día de hoy resultados exitosos (Juang, 2005). Si bien tanto ANN como SVM son métodos del estado del arte en reconocimiento de patrones, por sí mismos no pueden modelar eficientemente las variaciones temporales del habla. Es por eso que muchas veces se utilizan en realidad sistemas híbridos que combinen el poder de reconocimiento de los mismos con la capacidad de modelización del habla de los HMMs, existiendo entonces sistemas híbridos HMM/ANN o HMM/SVM.

En este trabajo se diseñaron tres algoritmos reconocedores de habla para su utilización en un sistema basado en palabras aisladas, independiente del hablante y de vocabulario pequeño. Como métodos de reconocimiento se utilizó tanto HMMs como también SVMs y ANNs, estos dos últimos diseñados de manera de poder ser implementandos sin requerir del uso del *framework* HMM. Se compararon los rendimientos de los tres algoritmos para condiciones de uso en laboratorio (muy buena relación señal a ruido), y con tres niveles distintos de ruido de fondo.

ASR: descripción general

Un sistema de ASR típicamente se compone de un módulo de extracción de parámetros acústicos que se encarga de calcular a partir de la señal de habla sus características útiles y otro de reconocimiento de los patrones pronunciados en el que se comparan las características obtenidas a partir de las pronunciaciones a reconocer con las características de los modelos conocidos por el sistema (su diccionario) para determinar a partir de procedimientos de reconocimiento de patrones cuáles fueron las pronunciadas por el hablante.

El siguiente es un diagrama en bloques de un sistema de ASR completo:

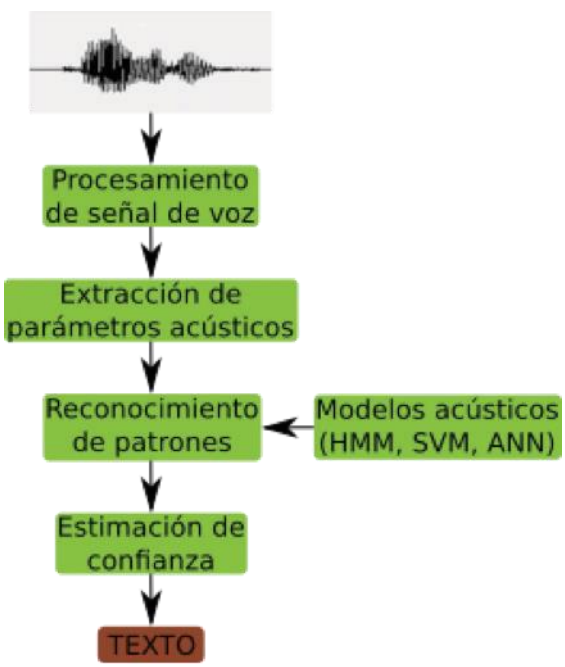


Fig.1. Diagrama en bloques de un sistema de ASR

Extracción de parámetros acústicos

Los parámetros utilizados suelen ser espectrales, siendo los más utilizados los llamados *Mel-frequency cepstral coefficients* (MFCCs). Para la obtención de los MFCC, se siguen los siguientes pasos a partir de la señal acústica (Martin, 2000):

1. Filtro de pre-énfasis: realza las componentes de alta frecuencia que sufren una atenuación debido al pulso glótico. Este filtro se caracteriza por la siguiente ecuación:

$$y_n = x_n - a * x_{n-1} \quad (1)$$

Donde a suele valer aproximadamente 0,95.

2. Ventaneo: se toman intervalos con una separación tal que queden solapados entre sí.

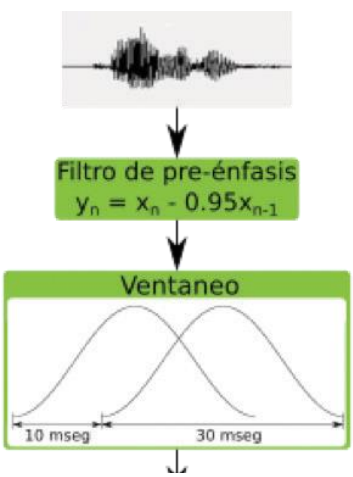
3. *|FFT|:* se calcula el módulo de la transformada de Fourier de cada intervalo.

4. *Escala de Mel*: se mapea la escala de frecuencias a la escala de Mel con filtros que imitan las diferentes sensibilidades en frecuencia del oído humano.

5. *Log:* se calcula el logaritmo a cada uno de los filtros en la escala de Mel.

6. *DCT:* se calcula la transformada coseno discreta a los logaritmos.

7. *Velocidad y aceleración:* se calculan la 1° y 2° derivada de la secuencia de frames analizada, las cuales permiten modelar la dinámica de variación espectral del habla.



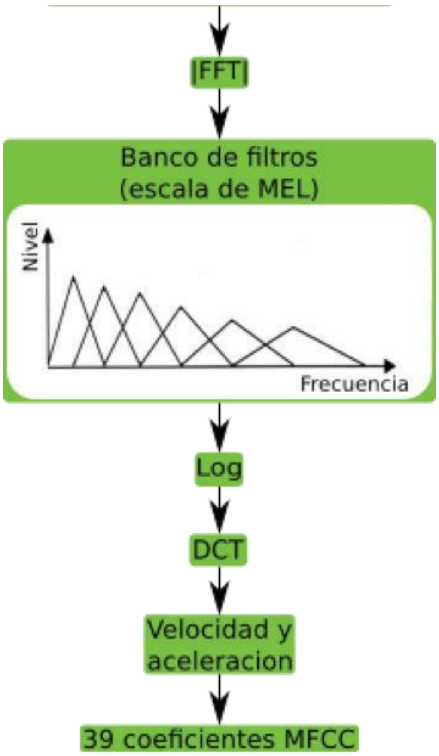


Fig.2. Pasos para la extracción de los coeficientes MFCC

Extracción de parámetros acústicos

El tipo de habla con el que se trabajó es el de palabras aisladas. Esto impone como condición al usuario el tener que dejar espacios de silencio entre la pronunciación de cada palabra y tiene la ventaja de que las bases de datos requeridas son más simples y que el algoritmo de reconocimiento presenta menor complejidad (Figura 3).

Modelos Ocultos de Markov

Se representan las características de las palabras pronunciadas a partir de la secuencia de vectores de observaciones O (las observaciones son alguna representación de ciertas características, como los coeficientes MFCC):

$$O = o_1, \ldots, o_T \quad (2)$$

donde cada o_t es el vector de observaciones correspondiente al instante t y T es la cantidad total de vectores de observaciones. Entonces, el objetivo del reconocimiento automático de palabras aisladas es el de resolver:

$$\arg\max_{1 \le i \le V} \left(P(w_i|O)\right) \quad (3)$$

Fig.3. Ejemplo de pronunciación de palabras aisladas

donde V es la cantidad de palabras en el diccionario y cada w_i es la palabra número i del mismo. Aplicando el teorema de Bayes obtenemos:

$$P(w_i|O) = \frac{P(O|w_i) * P(w_i)}{P(O)} \quad (4)$$

Es decir que para un set de probabilidades $P(w_i)$ iguales, la palabra más probable de haber sido pronunciada depende sólo de $P(O|w_i)$.

Modelos Ocultos de Markov para ASR

La producción del habla puede representarse por un modelo estadístico compuesto por estados con transiciones probabilísticas entre ellos y una función de densidad de probabilidad de emisión de distintos sonidos para cada uno de los estados (Rabiner, 1989). De esta forma, en este trabajo se modeló cada palabra del diccionario del sistema con un HMM.

Un modelo de Markov es una máquina de *N* estados finita que en cada instante *t* actualiza su estado, donde la transición del estado *i* al estado *j* se da a partir de la probabilidad discreta a*ij*. Según el estado *j* en el que se encuentre en cada instante t genera un nuevo vector de observaciones o_t a partir de la densidad de probabilidad de salida $b_j(o_t)$.

Entonces, el reconocimiento consiste en obtener la secuencia de vectores de observaciones O de la palabra pronunciada, seguido por un cálculo de las verosimilitudes para cada uno de los modelos $P(O|w_i)$ para $1 \leq i \leq V$ y finalmente de la selección del modelo que posea la máxima verosimilitud. Para esto generalmente se utiliza un algoritmo llamado *forward* (Young, 2006).

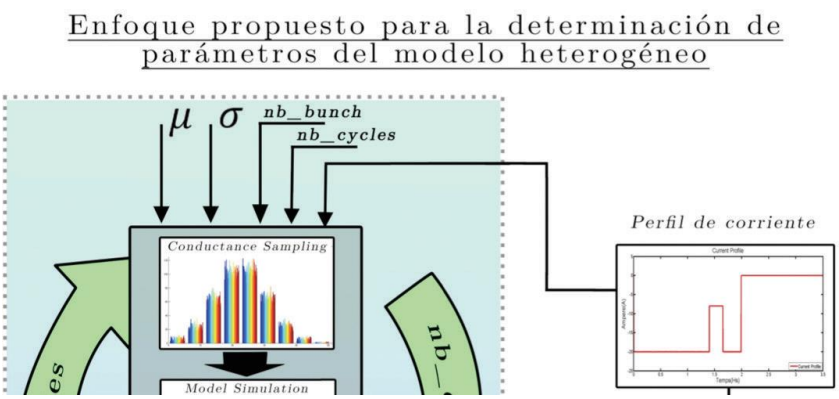


Fig.4. Ejemplo de representación de un HMM.

Máquinas de Vectores de Soporte

Las SVMs son clasificadores discriminativos capaces de lidiar con muestras de muchas dimensiones y garantizan su convergencia a un mínimo de la función costo asociada. Esto lo realizan estimando hiperplanos de decisión de manera directa, en lugar de modelar una distribución de probabilidad a partir de datos de entrenamiento, como lo hacen los HMMs (Ganapathiraju, 2004).

Los HMMs son modelos generativos ya que la clasificación está basada en la probabilidad de que el patrón incógnita haya sido generado por cada uno de los modelos que conforman el sistema de reconocimiento automático. Sin embargo, los problemas de clasificación pueden también ser resueltos, a veces con mejores resultados, a través de modelos discriminativos (Solera, 2007). Los clasificadores discriminativos pueden ser clave para crear modelos más robustos y precisos.

El funcionamiento de las SVMs se basa en maximizar un margen: la distancia entre la frontera de clasificación y las muestras (Figura 5). La minimización del riesgo empírico (ERM) puede ser utilizada para encontrar un buen hiperplano, aunque no garantiza una única solución. La minimización del riesgo estructural (SRM) impone un ordenamiento de los hiperplanos basándose en el margen. El hiperplano óptimo es aquél que maximiza el margen mientras minimiza el riesgo empírico, lo cual indirectamente garantiza una mejor generalización (Vapnik, 1998).

La distancia maximizada, o margen, es la responsable de las excelentes propiedades de generalización de las SVMs, ya que permite superar a la mayoría de los clasificadores no lineales en presencia de ruido, uno de los principales problemas de ASR. El margen indica cuánto ruido agregado a muestras limpias es permitido en el sistema sin que deje de clasificarlas correctamente.

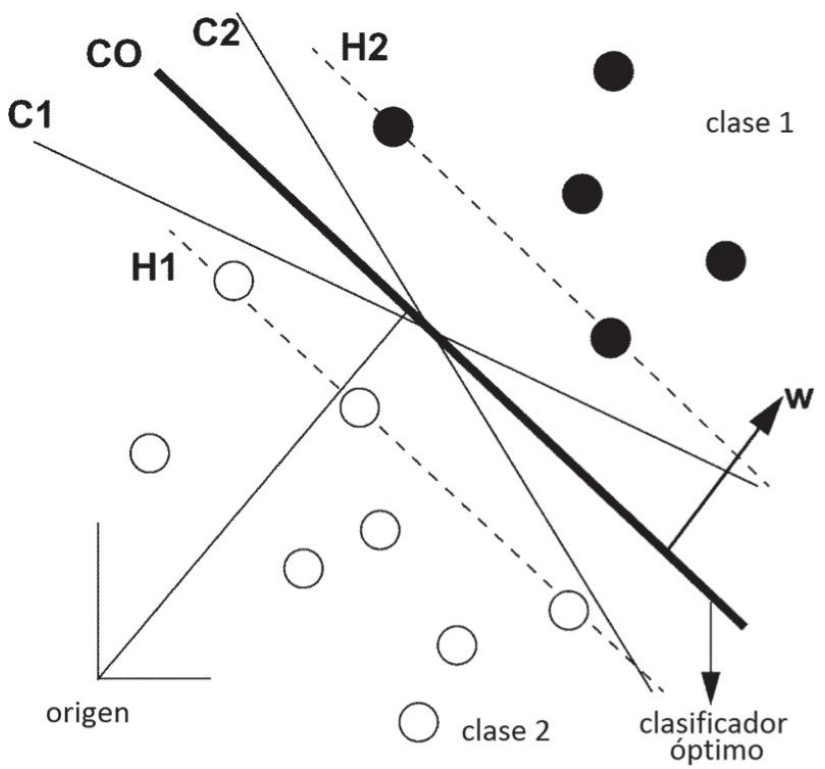


Fig.5. Diferencia entre minimización del riesgo empírico y del riesgo estructural. Cada hiperplano alcanza clasificación perfecta y, entonces, riesgo empírico nulo. Sin embargo, C0 es el hiperplano óptimo porque maximiza el margen (la distancia entre los hiperplanos H1 y H2). Los vectores que están ubicados sobre los hiperplanos H_1 o H_2 y cuya remoción cambiaría la solución hallada, son llamados vectores de soporte (adaptado de Ganapathiraju, 2004).

Si bien un clasificador SVM está definido en términos de los ejemplos de entrenamiento, no todos los ejemplos contribuyen a la definición del clasificador. Son los propios datos los que determinan cuan complejo será el clasificador, lo cual es totalmente opuesto al caso de las ANNs o los HMMs, donde la complejidad del sistema suele estar predefinida antes del entrenamiento (Ganapathiraju, 2004).

SVMs no lineales

Para los casos en los que los datos no están separados de forma lineal, se introduce el concepto del *kernel* (Schölkopf, 2002).

En el entrenamiento, los datos aparecen en forma de productos escalares, $x_i \cdot x_j$. Si primero se mapean los datos a algún otro espacio euclideo H, utilizando el mapeo $\Phi: R^d \rightarrow H$, entonces el algoritmo de entrenamiento dependerá de ellos sólo a través de productos escalares en H, es decir, en funciones de la forma $\Phi(x_i).\Phi(x_j)$. Entonces, si existe una función kernel K tal que $K(x_i),(x_j)=\Phi(x_i).\Phi(x_j)$, sólo se necesita usar K (incluso ni es necesario conocer Φ).

Todas las consideraciones anteriormente mencionadas se siguen cumpliendo, ya que se sigue tratando de una separación lineal, con la diferencia de que ahora es en un espacio diferente.

Los *kernels* probados en este trabajo fueron:

Lineal:

$$K(\bar{x_i}, \bar{x_j}) = \bar{x_i}^T . \bar{x_j} \quad (5)$$

Polinómico:

$$K(\bar{x_i}, \bar{x_j}) = (\bar{x_i}^T . \bar{x_j} + r)^d \quad (6)$$

RBF (Función de base radial):

$$K(\bar{x_i}, \bar{x_j}) = e^{-\frac{\|\bar{x_i} - \bar{x_j}\|^2}{2\sigma^2}} \quad (7)$$

SVMs para reconocimiento de habla

Las SVMs son clasificadores estáticos (los vectores con los que se trabaja deben tener todos la misma longitud) y por lo tanto tienen que ser adaptados para lidiar con la duración variable en las pronunciaciones del habla.

La Figura 6 muestra gráficamente el procedimiento que se implementó para la obtención de los vectores SVM de largo fijo, a partir de la señal de habla de largo variable. En primer lugar se realiza la extracción de parámetros MFCC a la señal acústica como ya fuera explicado previamente, obteniéndose una secuencia de largo variable de vectores de observación (a). Luego se divide la cantidad total de vectores de observación por 4, para agrupar a dichos vectores en 4 grupos de igual tamaño (b) (en caso de que la cantidad total no sea divisible por 4, se hace un redondeo). El siguiente paso es, para cada uno de estos grupos, calcular sus vectores promedio (ave) y desvío estandar (std) (c). Por último, se concatenan estos vectores promedio y desvío estandar corespondiantes a cada uno de los grupos, quedando como resultado el vector final a utilizar en SVM (d).

Nótese que al haber utilizado una cantidad fija de grupos, el largo del vector final será siempre tambien de largo fijo (independientemente del largo de la secuencia de habla original a la que representa). El largo de este vector es igual a la cantidad de parámetros MFCC utilizados multiplicados por el doble de la cantidad grupos, es decir: 39(*MFCC*).2(*ave y std*).4(grupos)=312.

Clasificación multi-clase

Las SVMs son en su naturaleza clasificadores del tipo binario (cada modelo discrimina entre dos clases), mientras que el ASR es un problema de clases múltiples.

Un punto fundamental en el diseño de clasificadores es si los mismos deben ser del tipo *one-versus-one* (uno contra uno), que aprenden a discriminar una clase de otra, o *one-versus-all* (uno contra todos), que aprenden a discriminar una clase de todas las demás. Los clasificadores *one-versus-one* son más pequeños, menos complejos y pueden ser estimados utilizando menos recursos que los clasificadores *one-versus-all.* Cuando el número de clases a clasificar es *N*, se necesita estimar *N.(N-1)/2* clasificadores *one-versus-one*, en comparación con N clasificadores *one-versus-all* (Ganapathiraju, 2004). Si bien los clasificadores one-versus-one pueden llegar a ser un poco más precisos que los clasificadores *one-versus-all,* se eligió en este desarrollo trabajar con clasificadores *one-versus-all,* para mayor eficiencia computacional y poder alcanzar un reconocimiento en tiempo real.

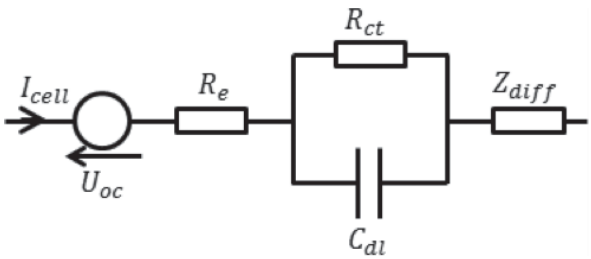


Fig.6. Procedimiento para la obtención del vector de largo fijo a partir de la señal de habla, la cual normalmente tiene un largo variable de entre 40 y 70 vectores de observación

Leave-one-Speaker-out Cross-Validation

La validación cruzada es una técnica de validación de modelos para determinar cómo van a generalizar los resutados de un análisis estadístico a un set de testeo independiente, es decir, para estimar cuán bien el modelo va a clasificar en la práctica.

En una ronda de validación cruzada se separan los datos en dos subconjuntos, se realiza un entrenamiento en uno de estos (llamado subconjunto de entrenamiento) y una validación de dicho entrenamiento en el otro (llamado subconjunto de validación). Para reducir la variabilidad, se realizan varias rondas de validación cruzada utilizando diferentes particiones del conjunto y los resultados de la validación en las distintas rondas son promediados.

Una de las principales razones para utilizar validación cruzada en lugar de validación convencional (particionar el set en dos subconjuntos de un 70% para entrenamiento y un 30% para testeo) es que no hay suficientes datos como para particionarlos en subconjuntos de entrenamiento y testeo sin perder precisión en la clasificación. En estos casos, una manera efectiva de estimar correctamente la precisión del modelo es usar validación cruzada (Seni, 2010).

Un caso particular de la validación cruzada de *K* iteraciones que se usa en reconocimiento de habla se llama *Leave-one-Speaker-out Cross-Validation*, y se utiliza para evaluar la robustez del reconocedor ante variaciones del hablante, determinando así sus características de generalización para el caso independiente del hablante.

Para el entrenamiento del modelo en cada iteración, se dejan de lado para el entrenamiento los datos correspondientes a uno de los hablantes y se utilizan para el procedimiento de validación. El procedimiento se repite para un hablante distinto en cada una de las *k* iteraciones, donde *k* es la cantidad de hablantes en el set de entrenamiento.

En este trabajo se utilizó *Leave-one-Speaker-out Cross-Validation* para estimar los sets de parámetros correspondientes a los modelos (en particular, los parámetros σ, r, d y C asociados a los kernels de los modelos SVM) con los que se obtendrían mejores tasas de reconocimiento en la generalización. Una vez encontrado el mejor set de parámetros, se volvió a entrenar con ellos el modelo por última vez, pero ahora utlizando todos los datos de entrenamiento.

Redes Neuronales Artificiales

El perceptrón es un tipo de clasificador binario lineal, es decir que realiza la clasificación basada en una combinación lineal entre un vec-

tor de pesos w y el vector de entrada x, de la siguiente manera (Freund, 1998):

$$f(x) = signo(w.x + b) \quad (8)$$

Donde *b* es el bias, el cual determina la posición del contorno de clasificación. El valor de f(x) (1 o -1) es utilizado para clasificar a la entrada x como positiva o negativa (o perteneciente o no a un grupo de datos).

El perceptrón es útil para clasificación lineal de datos binarios muy simples, pero para casos levemente más complejos (como por ejemplo, la operación booleana XOR (Abu-Mostafa, 2012)) es necesario agrupar varios perceptrones en capas, formando lo que se denomina perceptrón multicapa.

Redes Neuronales *Feed-Forward*

Las ANNs pueden pensarse como una forma de perceptrón multicapa suavizado, ya que en cada neurona la función signo es reemplazada por una función continua y derivable, como por ejemplo la función *logística o sigmoide:*

$$f(x) = \frac{1}{1+e^{-x}} \quad (9)$$

En el caso de las ANNs feed-forward (Figura 7), la información fluye en una sola dirección: desde la capa de entrada, a través de las capas ocultas y hasta la capa de salida. No hay ciclos o realimentaciones, a diferencia de otras como las ANNs recurrentes (Sak, 2014).

ANNs para reconocimiento de habla

Al igual que las SVMs, las ANNs también son clasificadores estáticos, ya que su capa de entrada posee una cantidad fija de neuronas. Por esta razón, se utilizó para este método el mismo procedimiento que en el caso de SVMs para generar el vector de largo fijo (Figura 6).

Por otro lado, en este caso no se utilizó el procedimiento de validación cruzada como para el caso de SVMs, ya que la herramienta utilizada (*toolbox* de ANNs de Matlab) no lo soporta. En su lugar se utilizó un sólo grupo de validación, compuesto por dos hablantes de la base de datos de entrenamiento. Una vez terminado el procedimiento de validación, la base de datos de entrenamiento completa fue utilizada para entrenar la ANN.

Generación de datos de entrenamiento con ruido

Una desventaja que presentan las ANNs respecto a las SVMs es que suelen requerir mucha mayor cantidad de datos para ser entrenadas. En caso de entrenar una ANN con datos insufi-

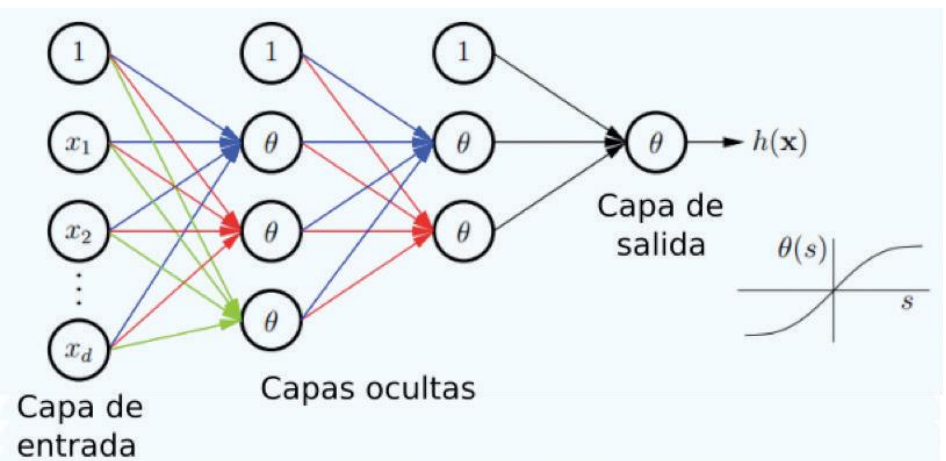


Fig.7. Ejemplo de una red neuronal *feed-forward* (adaptado de (Abu-Mostafa, 2012)).

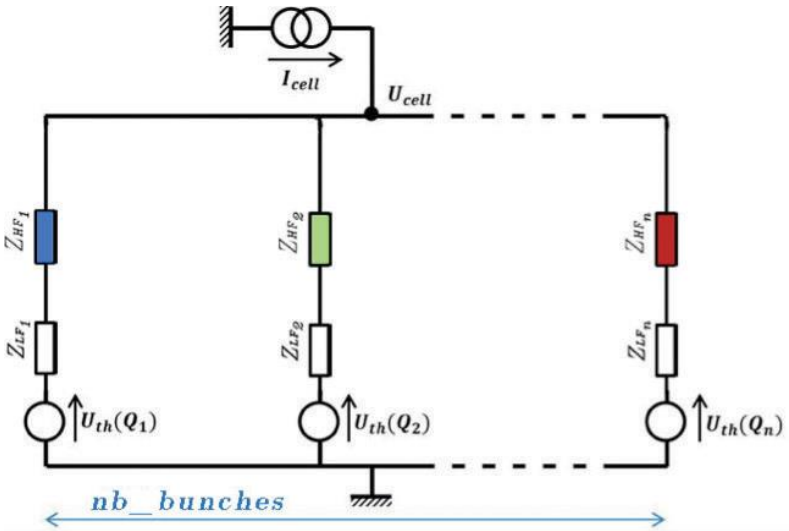


Fig.8. Performance observada al entrenar la ANN con la misma cantidad de datos (sin ruido) que para SVM

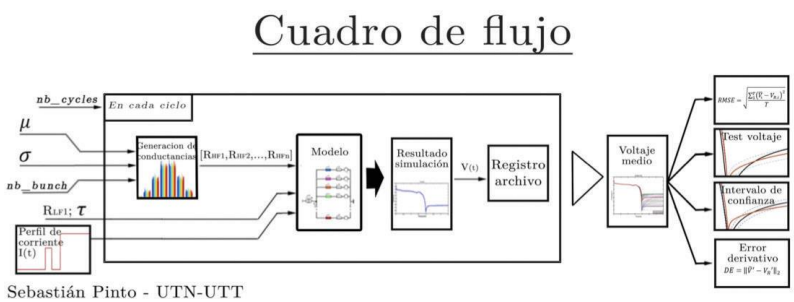


Fig.9. Performance observada al entrenar la ANN adicionalmente con copias ruidosas de los datos originales

Tabla 1. Rendimiento de los modelos HMM, SVM y ANN para distintos niveles de ruido.

Modelo	Configuración	Sin Ruido	SNR=50dB	SNR=40dB	SNR=30dB
HMM	16 estados	99.49%	98.99%	84.34%	28.28%
SVM	Kernel polinómico	96.46%	94.95%	94.95%	82.83%
ANN	1 capa oculta (30 neuronas)	91.41%	92.93%	90.40%	86.87%

cientes, la misma carecerá de una buena capacidad para generalizar (clasificar correctamente nuevos datos no presentes durante el entrenamiento), lo cual suele poder detectarse cuando el error de entrenamiento es mucho menor al error de testeo, como se observa en la Figura 8. Una manera de aumentar la capacidad de generalización de una ANN es generando mayor cantidad de patrones de entrenamiento. Esto puede conseguirse inyectando ruido en las entradas durante el entrenamiento (Sietsma, 1988), lo cual aumenta la robustez de la ANN bajo condiciones ruidosas (Yin, 2015). Además de inyectar ruido, también pueden aplicarse ciertas transformaciones a los datos para generar otros.

Resultados

Para los procesos de entrenamiento y testeo de los modelos se realizó la grabación de una base de datos en español argentino de 11 palabras (números 'cero' a 'diez') con las siguientes características (Marufo da Silva, 2016):

- Audios para entrenamiento: 13 hablantes (9 hombres, 4 mujeres), 99 pronunciaciones c/u (3 repeticiones de cada número a 3 velocidades distintas), 1287 audios en total.
- Audios para testeo: 6 hablantes (3 mujeres, 3 hombres), 33 pronunciaciones c/u (3 repeticiones de cada número), 198 audios en total.
- Todos los audios fueron grabados a 16KHz, mono y 16 bits.
- Todos los audios fueron grabados en una cámara anecoica para minimizar la influencia de los distintos tipos y niveles de ruido presentes en los audios grabados.

Para la realización del test se generaron además 3 copias de dichos audios con agregado de ruido, obteniéndose 3 versiones adicionales de cada audio con distintos niveles relación señal a ruido (SNR): 50dB, 40dB y 30dB. Este mismo procedimiento se realizó también para el caso de los audios de entrenamiento para ANN.

Para cada uno de los modelos (HMM, SVM y ANN) se probaron distintas configuraciones hasta encontrar para cada caso aquella que produjese mejores resultados. En la tabla 1 se observan los resultados comparativos entre las mejores configuraciones encontradas para cada modelo, para los casos: sin ruido, con 50dB de SNR, con 40dB de SNR y con 30dB de SNR.

Conclusiones

Se describió el desarrollo teórico y la implementación de un sistema de reconocimiento de habla aislada independiente del hablante, usando HMMs, SVMs y ANNs. La evaluación fue realizada utilizando un *corpus* multihablante compuesto por once palabras del español argentino, y su rendimiento en términos de porcentaje de reconocimiento fue comparado entre los tres métodos de clasificación implementados.

Los resultados para el caso HMM muestran altos niveles de reconocimiento para buenos niveles de SNR, pero que se degradan rápidamente a medida que aumenta el ruido.

Por otro lado, se comprobó la robustez de las SVMs bajo condiciones de ruido. Cabe destacar que no se requirió generar nuevas versiones de los audios agregando ruido para este caso, ya que las SVMs poseen una muy buena capacidad de generalización a partir de pocos datos de entrenamiento. Además, teniendo en cuenta que la complejidad de las SVMs está dada por la complejidad de los datos de entrenamiento, el agregado de datos ruidosos para entrenar hubiese aumentado notablemente la cantidad de vectores de soporte, aumentando así la complejidad del modelo.

Para el caso de las ANNs, teniendo en cuenta que su complejidad no está dada por los datos de entrenamiento, sino por la arquitectura preestablecida de la red, el agregado de ruido a los datos de entrenamiento disponibles pasa a ser un método adecuado cuando los mismos son insuficientes para entrenar a la red. En este caso, se comprobó que al agregar datos con distintos niveles de ruido, la capacidad de generalización de la ANN mejora ya que la misma tiende a ser menos sensible al ruido.

Referencias

ALVAREZ, A. G.; EVIN, D. A.; VERRASTRO, S., (2016) Implementation of a Speech Recognition System in a DSC. IEEE Latin America Transactions, 14(6), 2657-2662.

JUANG, B. H.; RABINER, L. R., (2005) Automatic speech recognition: a brief history of the technology development. Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara, 1.

MARTIN, J. H.; JURAFSKY, D., (2000) Speech and language processing. International Edition.

RABINER, L. R., (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77.2: 257-286.

YOUNG, S.; EVERMANN, G.; GALES, M.; HAIN, T.; KERSHAW, D.; X LIU, G.; MOORE, J.; ODELL, D.; OLLASON, D.; POVEY, V.; VALTCHEV, P.; WOODLAND, P., (2006) The HTK book. Cambridge University Engineering Department.

GANAPATHIRAJU, A.; HAMAKER, J. E.; PICONE, J., (2004) Applications of support vector machines to speech recognition. IEEE Transactions on Signal Processing 52.8: 2348-2355.

SOLERA-UREÑA, R. et al ,(2007) SVMs for automatic speech recognition: a survey. Progress in nonlinear speech processing. Springer Berlin Heidelberg. 190-216.

VAPNIK; V. N., (1998) Statistical Learning Theory. New York: Wiley.

SCHÖLKOPF, B. y SMOLA, A. J., (2002) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

SENI, G. y ELDER, J. F., (2010) Ensemble methods in data mining: improving accuracy through combining predictions. Synthesis Lectures on Data Mining and Knowledge Discovery, 2(1), 1-126.

FREUND, Y. y SCHAPIRE, R. E., (1998) Large margin classification using the perceptron algorithm. In Proceedings of the eleventh annual conference on Computational learning theory (pp. 209-217). ACM.

ABU-MOSTAFA, Y.; MAGDON-ISMAIL, M. y HSUAN-TIEN, L., (2012) Learning From Data, e-Chapter 7: Neural Networks. Amlbook.com.

SAK, H.; SENIOR, A. y BEAUFAYS, F., (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth Annual Conference of the International Speech Communication Association.

SIETSMA, J. y DOW, R. J., (1988) Neural net pruning-why and how. In IEEE International Conference on Neural Networks (Vol. 1, pp. 325-333). IEEE San Diego.

YIN, S.; LIU, C.; ZHANG, Z.; LIN, Y.; WANG, D.; TEJEDOR, J. y LI, Y., (2015) Noisy training for deep neural networks in speech recognition. EURASIP Journal on Audio, Speech, and Music Processing, 2015(1), 1-14.

MARUFO DA SILVA, M.; EVIN, D. A. y VERRASTRO, S., (2016) Speaker-independent embedded speech recognition using Hidden Markov Models. In Ciencias de la Informática y Desarrollos de Investigación (CACIDI), IEEE Congreso Argentino de (pp. 1-6). IEEE.