Reconocimiento de voz

voice recognition

Autor: Jorge Orobio Auz

*IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: j.orobio@utp.edu.co

***Resumen*— Los sistemas biométricos se han convertido en una realidad cotidiana para las personas. Día a día vemos como su uso se ha extendido a las actividades comerciales, privadas, gubernamentales, entre otras. En este artículo se presentan los conocimientos y herramientas básicas necesarias para llevar a cabo el reconocimiento de voz y los diferentes métodos por los cuales se puede llegar a esta solución como redes neuronales con backpropagation, o DTW con modelos ocultos de Markov.**

***Palabras clave—* reconocimiento de voz**

**, modelos de markov, inteligencia artificial, backpropagation, computación.**

1. INTRODUCCIÓN

Los seres humanos poseen una alta capacidad para reconocer rostros y voces aún en escenarios donde existan altos niveles de variabilidad y ruido. Diseñar sistemas automáticos que emulen esta propiedad natural de los humanos, constituye una tarea compleja y con muchas limitaciones. Probablemente una de las primeras interrogantes sea ¿los rostros y voces son diferenciables como medidas biométricas? Afortunadamente en los últimos años se han realizado una gran cantidad de investigaciones que afirman esta interrogante, en especial el área de la biometría.

El reconocimiento de voz tiene la dificultad principal en la extracción de características de la voz, debido a que cuenta con diversos problemas a solucionar, como lo es que una persona no puede pronunciar dos veces igual la misma palabra, debido entre otras cosas: al estado de ánimo, la salud, y la fuerza de pronunciación, el tiempo, la entonación, etc. Aún y con estas dificultades, se han desarrollado algoritmos para poder determinar un nivel de coincidencia entre las pronunciaciones para poder realizar un reconocimiento eficaz, logrando encontrar algunos algoritmos que lo realizan con mayor desempeño, debido a su construcción, y sus características internas de proceso para el reconocimiento de patrones, entre ellos destacan las ANN, el DTW y Los HMM.

**CAPTURA DE LA VOZ.**

El archivo de audio usado para este análisis es entregado directamente en un formato compatible con el equipo y listo para trabajar, por lo que para este ejercicio no se tendrá tanta atención sobre este apartado pero dado el caso de que esta parte no se haya desarrollado se explicará brevemente cuál sería el procedimiento seguir:

Para la realización de la captura de la voz se propone registrar 5 veces una misma frase con cinco personas diferentes, cada una de ellas grabó dicha frase con estados de ánimo diferentes para construir un dataset más variado.

**ETAPA DE AJUSTE**

Una vez obtenida la señal del micrófono y transportada a nuestro ambiente, es decir, tener un vector de sonido, se realiza el siguiente paso que es la normalización, ya que es necesario ajustar los tamaños temporales de los grupos de estudio.

El ajuste consiste en quitar la parte de silencio de la entrada a todos los archivos y dejarlos en un comienzo semejante para poder aplicar el algoritmo a señales con un comienzo en una palabra, sin que el silencio del inicio intervenga. El método que se emplearía para la normalización fue simplemente una comparación entre los puntos del vector de sonido y verificar dónde se encontraría el inicio de la pronunciación mediante un cambio repentino de valor

Teniendo ya las señales alineadas en un valor inicial igual, el siguiente proceso necesario para nuestro sistema es el dar un cambio de manejo de variables, en lugar de tener la señal en el tiempo, la tendremos que manejar en términos de la frecuencia, lo que implica realizar una transformación, usando una Codificación de Predicción Lineal (LPC), con la cual obtendremos unos coeficientes que nos servirán para el entrenamiento de la red, los valores de transición del HMM.

**OBTENCIÓN DE LOS COEFICIENTES POR LPC**

La recursión de Levinson-Durbin es una importante solución para la obtención de coeficientes de filtros FIR[2] es por lo que se toma la señal después del ajuste para realizar el cambio a la frecuencia, mediante la obtención de los coeficientes por LPC, mediante dicho algoritmo. Se ajustó sobre el número de coeficientes para obtener con dicho algoritmo, para dejarlo establecido con 30 coeficientes de cada palabra, de esta manera, si la palabra era muy larga en pronunciación o corta, se aseguraba que los coeficientes de todas fueran iguales y con ello poder realizar una comparación y determinar la concordancia o diferencia entre las pronunciaciones.

**SOLUCIONES**

A partir de este punto se pueden implementar diferentes soluciones que pueden plantear la solución tomando como base las redes neuronales, el primer método posible encontrado durante este análisis revela que el reconocimiento de voz puede ser efectuado mediante mapas de kohonen estos mapas son redes neuronales que tienen la peculiaridad de tener un mapa organizado de sus pesos, mediante el algoritmo de entrenamiento. por otra parte se encuentra, también dentro del campo de redes neuronales tenemos la RNA backpropagation cuyo fin estaría limitado a diferencias los audios de un conjunto de audios entregado. Dejando claro las dos metodologías que se pueden usar haremos énfasis en la solución mediante mapas de kohonen

**SOLUCIÓN POR BACKPROPAGATION**

**Código de Predicción Lineal**

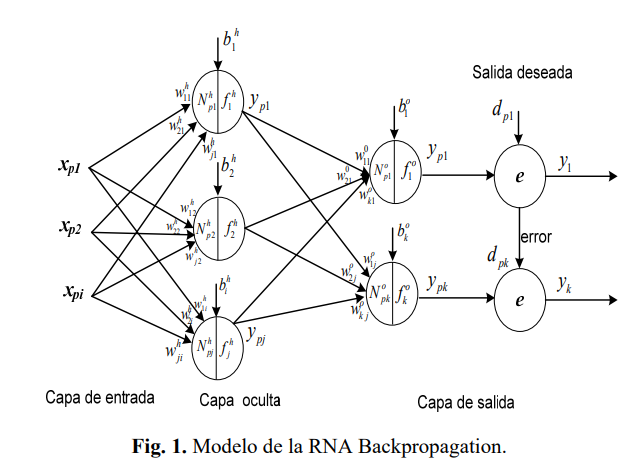
Una gran parte de las aplicaciones relacionadas con el tratamiento del habla, están basadas en el análisis de LPC, dado que es capaz de extraer la información lingüística y eliminar la correspondiente a la persona en particular. La predicción lineal modela la zona bucal humana como una respuesta al impulso infinita, que produce la señal de voz. El término predicción lineal se refiere al método para predecir o aproximar una muestra de una señal en el dominio del tiempo s[n] basada en varias muestras anteriores s[n - 1], s[n - 2], s[n -M].

**La Red Backpropagation**

En 1986, Rumelhart, Hinton y Williams, basados en otros trabajos formalizaron un método para que una red neuronal aprendiera la asociación que existe entre los patrones de entrada a la misma y las clases correspondientes, utilizando más niveles de neuronas que los que utilizó Rosenblatt para desarrollar el Perceptrón.

Este nuevo método es conocido como Backpropagation (retropropagación del error) que es un tipo de red con aprendizaje supervisado, el cual emplea un ciclo propagación-adaptación de dos fases.

Una vez aplicado un patrón de entrenamiento a la entrada de la red, este se propaga desde la primera capa a través de las capas subsecuentes de la red, hasta generar una salida, la cual es comparada con la salida deseada y se calcula una señal de error para cada una de las salidas, a su vez esta es propagada hacia atrás, empezando de la capa de salida, hacia todas las capas de la red hasta llegar a la capa de entrada, con la finalidad de actualizar los pesos de conexión de cada neurona, para hacer que la red converja a un estado que le permita clasificar correctamente todos los patrones de entrenamiento. La estructura general se muestra en la Figura. 1.



**Algoritmo de entrenamiento de la Red.**

A continuación se presenta el algoritmo empleado para el entrenamiento de la RNA Backpropagation. [1][2][5].

1. Inicializar los pesos de la red (w) con valores aleatorios pequeños.

2. Mientras la condición de paro sea falsa realizar los pasos (3-6).

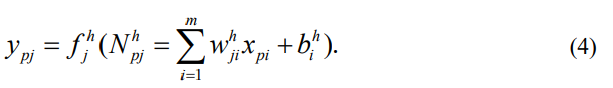
3. Se presenta un patrón de entrada, (xp1, xp2,…,xpi) y se específica la salida deseada que debe generar la red (dp1, dp2,…,dpk).

4. Se calcula la salida actual de la red, para ello se presentan las entradas a la red y se va calculando la salida que presenta cada capa hasta llegar a la capa de salida (y1,y2,…,yk). Los pasos son los siguientes:

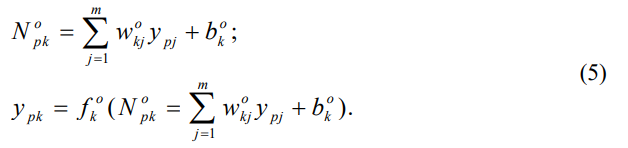
a) Se determinan las entradas netas para las neuronas ocultas procedentes de las neuronas de entrada.



b) Se aplica la función de activación a cada una de las entradas de la neurona oculta para obtener su respectiva salida.



c) Se realizan los mismos cálculos para obtener las respectivas salidas de las neuronas de la capa de salida.



5. Determinación de los términos de error para todas las neuronas:

a) Cálculo del error (salida deseada–salida obtenida).

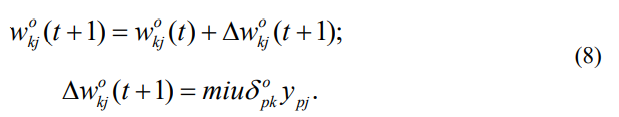


b) Obtención de la delta (producto del error con la derivada de la función de activación con respecto a los pesos de la red).

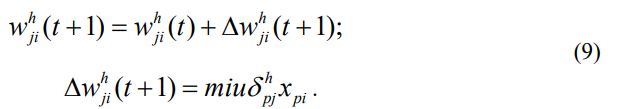


6. Actualización de los pesos. Se emplea el algoritmo recursivo del gradiente descendente, comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada.

a) Para los pesos de las neuronas de la capa de salida:



b) Para los pesos de las neuronas de la capa oculta:



7. Se cumple la condición de paro (error mínimo ó número de iteraciones alcanzado logrado).

**SOLUCIÓN POR MAPAS DE KOHONEN**

Los mapas autoorganizados (SOFM por sus siglas en inglés) o mapas de Kohonen son redes neuronales que tienen la peculiaridad de tener un mapa organizado de sus pesos, mediante el algoritmo de entrenamiento. Los SOFM poseen un gran potencial de aplicabilidad práctica, clasificando patrones, Fig. 2. Archivo de audio de la palabra “casa” capturado en simulink. Fig. 4. Estructura de una SOFM. cuantificando vectorialmente, reduciendo dimensiones, extrayendo rasgos, etc. Este modelo tiene dos variantes denominadas LVQ (Learning Vector Quantization) y TPM (Topology Preserving Map) o SOM (Self Organizing Map), ambas se basan en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red, aunque difieren en las dimensiones de éstos, siendo de una sola dimensión en el caso de LVQ y bidimensional o tridimensional en la red SOM. Estas redes se tratarán con mayor profundidad en secciones posteriores. El aprendizaje en el modelo de Kohonen es de tipo Off-line, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento. En la etapa de aprendizaje se fijan los valores de las conexiones (feedforward) entre la capa de entrada y la salida. Esta red utiliza un aprendizaje no supervisado de tipo competitivo, las neuronas de la capa de salida compiten por activarse y sólo una de ellas permanece activa ante una determinada información de entrada a la red, los pesos de las conexiones se ajustan en función de la neurona que haya resultado vencedora

En fase de aprendizaje cada neurona del mapa sintoniza con diferentes rasgos del espacio de entrada. El proceso es el siguiente: tras la presentación y procesamiento de un vector de entrada x(t), la neurona vencedora modifica sus pesos de manera que se parezca un poco más a x(t). De este modo, ante el mismo patrón de entrada, dicha neurona responderá en el futuro todavía con más intensidad. Sin embargo, el modelo de mapa de Kohonen aporta una importante novedad, pues incorpora a este esquema relaciones entre las neuronas próximas en el mapa. Esto quiere decir que el mapa puede clasificar en la fase de aprendizaje algunas variantes del patrón, como por ejemplo la pronunciación de la palabra “A” más de una vez por la misma persona, que dicha palabra no es igual en ninguna de las veces. Con ello la ANN puede clasificar de manera correcta las variaciones que se pueden presentar cuando un hablante pronuncie la misma palabra.

El algoritmo de aprendizaje autoorganizado basa su funcionamiento en las distancias euclídeas para determinar la neurona ganadora y la cual se asemejara a la entrada, actualizando sus pesos con la regla de la siguiente ecuación



**Ecuación 10**

Obteniendo con este aprendizaje un mapa autoorganizado que nos dejará en cada neurona una entrada, así si lo aplicamos en el reconocimiento de voz tendríamos un peso y una neurona exclusiva para cada fonema y sus variantes del patrón, teniendo con ello una gran posibilidad de segmentación. Con esto tomamos 30 muestras de cada palabra para pasar los coeficientes de todas las neuronas al algoritmo de entrenamiento de la red neuronal, y con esto obtener índices y pesos de cada palabra, que serán la base para el reconocimiento de las palabras, dichos índices son autoorganizativos.

El algoritmo de reconocimiento de la red llamada mapa auto organizativo (SOM por sus siglas en Inglés), se realiza cuando entra una palabra con los coeficientes del LPC y comienza a compararla con las ya entrenadas para ver en qué parte del mapa cae y así poder determinar la pronunciación de la palabra.

E. Algoritmo de DTW

Para utilizar el algoritmo DTW, se tomaron las voces captadas sin pasarlas por la conversión del LPC, ya que como el algoritmo está basado en el tiempo, no serviría tener los coeficientes del LPC que representan la frecuencia. Se tomó un patrón como base para poder comparar con las voces que se van a reconocer, por lo cual se hizo una selección meticulosa sobre cada una de las palabras, por lo que se decidió tomar un promedio de cada una de las palabras a reconocer para tenerlas como patrón de referencia.

F. Algoritmo de HMM

En primera instancia se realizó una conversión de coeficientes de LPC por letras para un fácil manejo en la evaluación del algoritmo de Viterbi con las probabilidades de transición. Dicha conversión que toma los promedios obtenidos de cada una de las palabras de la base de datos, comparando cada coeficiente de los promedios con cada coeficiente recibido de la pronunciación efectuada, para obtener una distancia euclídea, y así la menor distancia entre el coeficiente de la palabra recibida y los promedios es la letra a la que se convertirá dicha secuencia.

Una vez tenida la secuencia de letras de la palabra pronunciada, se introduce al algoritmo de Viterbi para evaluar el comportamiento de la palabra y su probabilidad de que corresponda a una de las palabras guardadas en la base de datos. El HMM que se utilizó para la evaluación del algoritmo de Viterbi se presenta en la ecuación 10.

El algoritmo de Viterbi se inicializa con una de las letras establecidas, dependiendo de cual de ellas tuvo el mayor número de repeticiones en la secuencia de letras, por ejemplo si se pronunció la palabra casa y al realizar la conversión a letras se obtuvo que en la secuencia se tienen más letras A, se inicializa con las probabilidades de A, si hubiera sido B, se inicializará con las probabilidades de B, y así sucesivamente.

Finalizando la evaluación se obtienen once probabilidades, donde la mayor probabilidad será la palabra pronunciada.

G. Modelo de Mayoría

Una vez que se tienen los reconocimientos por parte de cada uno de los algoritmos establecidos, se procede a realizar una decisión de mayoría que nos dará el resultado final del sistema y determinará que palabra se pronunció. En la figura 6 se presenta el diagrama de flujo del reconocimiento de voz empleado en este documento, tomando en cuenta que XI es un vector ya ajustado y Xi es el vector de los coeficientes de LPC

CONCLUSIÓN

En conclusión tenemos diferentes métodos para realizar el reconocimiento de voz mediante redes neuronales, cada uno de estos puede ser utilizado pero dependiendo de la metodología se requiere de un mayor esfuerzo para obtener resultados y afecta drásticamente la eficiencia de la metodología, por lo que el método a elegir va a depender de los planes que se tengan a futuro con dicho reconocimiento de voz

REFERENCIAS

Referencias en la Web:

***Carrasco, M. A., Portugal, R., & Peralta, B. (2006). Reconocimiento biométrico de audio y rostro: un sistema viable de identificación. Pontificia Universidad Católica de Chile Departamento de Ciencia de la Computación.***

***Ortega, C. A. D. L., Romo, J. C. M., & González, M. M. (2006). Reconocimiento de voz con redes neuronales, DTW y modelos ocultos de Markov. Conciencia Tecnológica, (32), 0.***

***Cruz-Beltrán, L., & Acevedo-Mosqueda, M. (2008). Reconocimiento de voz usando redes neuronales artificiales backpropagation y coeficientes lpc. In 6to Congreso Internacional de Cómputo en Optimización y Software. CiCos (pp. 89-99).***