MultiLayer Perceptron para reconocimiento de personas por su voz

1st Rubio Montiel, Ignacio 100
Departamento de Computer Science
Universidad de Ingeniería y Tecnología
Lima, Perú
ignacio.rubio@utec.edu.pe

2nd Zapata Gallegos, Neo Marcelo 100
Departamento de Computer Science
Universidad de Ingeniería y Tecnología
Lima, Perú
neo.zapata@utec.edu.pe

3rd Vidal Castro, Edir Sebastian 100
Departamento de Bioingeniería
Universidad de Ingeniería y Tecnología
Lima, Perú
edir.vidal@utec.edu.pe

***Abstract*—El reconocimiento de una persona mediante su voz es una tarea importante para la ejecución de ciertas tareas. La voz puede ser interpretada por su espectro en el espacio de la frecuencia para reducir los vectores que la caracterizan a lo largo del tiempo en un vector característicos de dimensionalidad reducida. Utilizando 24 actores con 59 archivos de audio WAV por cada uno de ellos, se ha implementado MFCCs para la extracción de un vector característico por cada archivo. Además, se ha desarrollado una red neuronal MLP utilizando estos vectores característicos para su entrenamiento de tal manera que dado un vector determine a qué actor pertenece.**

***Index Terms*—MultiLayer Perceptron, red neuronal, identificación de voz, Mel-frecuency cepstral coefficients.**

I. Introducción

En el campo de la inteligencia artificial la identificación de voz es un área que tiene diversas aplicaciones como en seguridad, ciencias sociales, psicología, etc. Uno de los métodos para la realización de esta tarea es mediante la implementación de redes neuronales los cuales han obtenido grandes resultados en la clasificación de imágenes, predicción de comportamientos, generación de datos y muchos más. Una de las arquitecturas de redes neuronales mayormente utilizadas por la facilidad en su entendimiento es la MultiLayer Perceptron (MLP) la cual consiste en una red neuronal con, como su nombre sugiere, múltiples capas de neuronas interconectadas. Para el siguiente trabajo se busca la identificación de voz de 24 actores distitnos mediante la implementación de una red neuronal MLP. Los vectores característicos serán generados utilizando los coeficientes cepstrales de frecuencia Mel (MFFCs) por cada archivo WAV existente (59 archivos por actor). Estos serán introducidos a la red neuronal desarrollada para su entrenamiento. La finalidad del algoritmo es el poder identificar el actor dado un vector de entrada.

II. Estado del arte

A. Obtención de vectores característicos

La obtención de vectores característicos tiene el propósito de representar una serie de matrices con dimensiones grandes a un vector con dimensiones reducidas, permitiendo un análisis y procesamiento más eficiente cuando se tiene un dataset muy grande [1].

En el caso de las señales de audio, el método más utilizado es de los coeficientes cepstrales de frecuencia Mel (MFFCs) siendo este el método dominante para la identificación de locutores, interpretación de discursos y otras áreas relacionadas a esta área [2]. MFFC utiliza la escala Mel para dividir las bandas de frecuencia en sub-bandas y utilizar la transformada discreta del coseno para obtener los coeficientes denominados MFCCs [3]. Cabe resaltar su conveniencia al momento de procesar voces ya que la escala Mel está basada en como las personas diferencian las frecuencias.

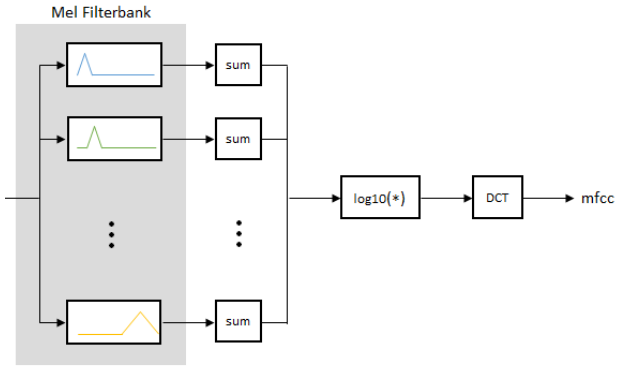

Fig. 1: Diagrama para obtención de coeficientes cepstrales de frecuencia Mel (MFFCs) [4].

B. Redes neuronales artificiales

Las redes neuronales artificiales (ANNs) son un tipo de red computacional que se inspira en las redes neuronales biológicas basándose en la interconexión de neuronas con otras mediante capas que juntas constituyen a la red completa en un sistema [5]. Los nodos de entrada de las ANN reciben señales de entrada, los nodos de la capa oculta calculan estas señales de entrada y los nodos de la capa de salida calculan la salida final procesando los resultados de la capa

oculta mediante funciones de activación [5]. Una capa puede tener solo una docena de unidades o millones de unidades, según la complejidad de las redes neuronales para aprender los patrones ocultos en el conjunto de datos [5]. Las ANN se utilizan en una amplia gama de aplicaciones, incluido el reconocimiento de voz, la clasificación de imágenes y la detección de objetos como mencionado anteriormente. Las ANN se entrenan de tal forma que todos sus pesos y umbrales se establecen inicialmente en valores aleatorios, y los datos de entrenamiento se alimentan a la capa de entrada, pasan a través de las capas sucesivas, se multiplican y se suman de manera compleja hasta que finalmente llega, radicalmente transformado, a la capa de salida; los pesos y umbrales se ajustan continuamente hasta que los datos de entrenamiento con las mismas etiquetas produzcan resultados similares de forma constante. [5].

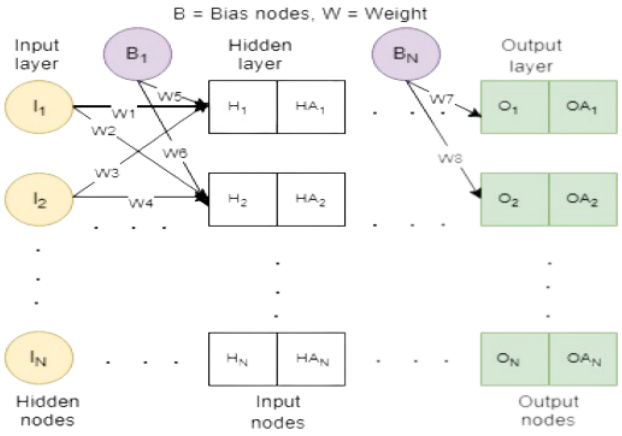

Fig. 2: Arquitectura típica de una red neuronal artificial (ANN) [5].

La red neuronal mayormente utilizada vendría a ser la MultiLayer Perceptron (MLP) la cual consiste en múltiples capas de perceptrones también denominados neuronas artificiales [6]. Lo que caracteriza esta arquitectura de red neuronal es que está completamente interconectada de tal manera que cada perceptrón de una capa están concetados a cada perceptrón de la siguiente capa [6].

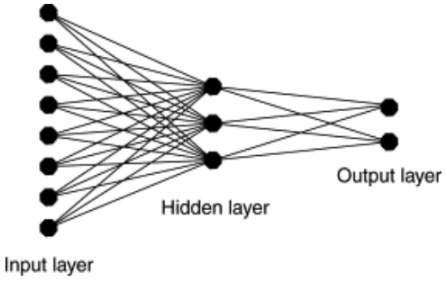

Fig. 3: Arquitectura simplificada de una MultiLayer Perceptron (MLP) [6].

Esta arquitectura es favorable al tener suficientes datos, dado que con suficientes unidades en cada capa oculta se puede aproximar virtualmente cualquier función a cualquier precisión deseada; las MLP son técnicamente aproximadores universales [6]. Cabe resaltar que esto solo es posible con un dataset suficientemente grande, por lo que las MLP son herramientas útiles al momento de solucionar problemas complejos cuando existe la data necesaria.

C. *Funciones de activación*

De por si las redes neuronales como se han explicado hasta el momento son esencialmente modelos de regresión lineal, por lo que es importante introducir una función que modifique esto. Las funciones de activación son funciones matemáticas que son utilizadas en las redes neuronales artificiales para introducir la no linealidad en la red neuronal, permitiendo que la red pueda aproximar cualquier función continua con suficiente entrenamiento [7]. La función de activación recibe la suma de los pesos de los datos de entrada y le agrega un bias para determinar si la neurona debe activarse o no, por lo que el propósito de la misma es transformar la suma de estos pesos de entrada al nodo a un valor de salida que alimente la siguiente capa oculta o el mismo output de la red neuronal [8].

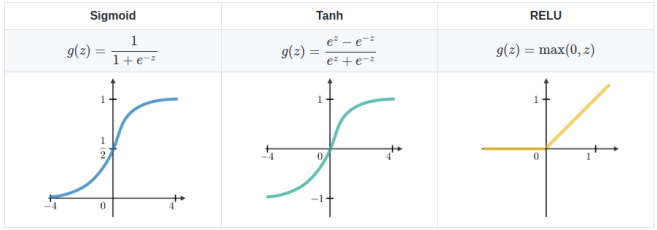

Fig. 4: Funciones de activación Sigmoide, Tanh y ReLu.

Algunas de las funciones de activación convencionalmente utilizadas son la función tangente hiperbólica (Tanh) y la función sigmoide [7]. Por otro lado, existen funciones de activación más modernas que solucionan algunas deficiencias funciones tradicionales como la función de unidad lineal rectificada (ReLu) [8]. A continuación se presenta una tabla con algunas recomendaciones sobre el uso de las funciones de activación mencionadas:

Tabla I: Recomendaciones sobre el uso de las funciones de activación Tanh, Sigmoide y ReLu [8].

Función de activación	Comentario	Uso
Tanh	Gradiente propenso a desaparecer	En redes neuronales recurrentes
Sigmoide	Gradiente propenso a desaparecer y zigzagueo	En puertas lógicas booleanas
ReLu	Función más popular en capas ocultas. Utilizar LReLu si ReLu "muere"	La primera opción a utilizar

III. Experimentos

A. Coeficientes cepstrales de frecuencia mel

Sobre la obtención de coeficientes cepstrales de frencuencia mel (MFCCs) se ha optado por utilizar la biblioteca librosa en Python especializada en el análisis de música y fragmentos de audio en general [9]. Por cada archivo de audio se procedió a obtener los vectores característicos con 128 coeficientes cepstrales de frecuencia mel. Una vez obtenido estos, se procedió a sacar la media por cada coeficiente para obtener un vector característico de 1x128 por cada archivo de audio. Todos los vectores característicos se guardaron en un archivo dataset.csv con su autor correspondiente al vector en el mismo.

Actor	Audio type	Param_1	Param_2	Param_3	Param_4	Param_5	Param_6	Param_7	Param_8	...
Actor_01	Song	-590.232544	51.974358	-15.977171	11.776335	-1.170663	-3.819702	-12.761039	-7.536566	...
Actor_01	Song	-569.296265	50.261189	-16.246378	11.208460	0.157604	-5.491686	-12.515923	-7.113371	...
Actor_01	Song	-577.159363	51.260155	-12.301028	14.358865	-1.930636	-2.710555	-10.627647	-7.980994	...
Actor_01	Song	-578.583801	49.643364	-15.457994	13.520749	-2.213725	-1.458224	-13.004834	-8.181676	...
Actor_01	Song	-589.882080	57.384579	-12.488736	13.282143	1.861456	-3.881037	-12.396720	-4.318160	...
...	...	...	...	...	...	...	...	...	...	...
Actor_24	Speech	-586.506653	24.936182	-17.393667	-1.750550	-15.307640	-9.822910	-15.646090	-15.538772	...
Actor_24	Speech	-532.467224	37.716980	-14.932145	-5.954791	-15.144546	-13.899151	-10.687799	-12.034179	...
Actor_24	Speech	-541.251648	29.595709	-18.001362	-3.139160	-14.282516	-17.555689	-12.232482	-12.344251	...
Actor_24	Speech	-492.652527	23.887983	-6.026659	1.677622	-11.062459	-5.152931	-10.083343	-9.132248	...
Actor_24	Speech	-517.984802	29.571215	-3.909699	-1.117704	-5.258939	-7.113936	-11.250309	-6.680300	...

Fig. 5: DataFrame con la forma de los vectores característicos almacenados en dataset.csv.

Posteriormente se dividió el 70% de los datos en el archivo para ser utilizados en el entrenamiento de la red neuronal (training.csv) y el 30% para la validación de la misma (testing.csv).

B. MultiLayer Perceptron

1) Configuración de la red: La MultiLayer Perceptron (MLP) fue configurada de distintas maneras en el presente proyecto para poder visualizar distintas métricas de rendimiento de la red neuronal como una función de la cantidad de capas ocultas y neuronas en cada una de estas. Por ende, se cuenta con una MLP donde se varian el número de capas ocultas (1 a 3), y número de neuronas por capa (50, 100 y 200).

2) Funciones de activación: Se ha optado por utilizar distintas funciones de activación y verificar con las mismas métricas de rendimiento utilizadas en la sección anterior si existe un cambio significativo o no. Se han utilizado las funciones de activación Tanh, Sigmoide y ReLu.

3) Resultados: Debido a la naturaleza de la función sigmoide y tanh se encontró que ambas funciones de activación causan que la red neuronal se saturara y, por consiguiente, no aprenda. Esto no sucede utilizando la función de activación ReLu, por lo que a continuación se muestra las métricas de rendimiento obtenidas a partir de esta misma:

Tabla II: Error con la función de activación ReLu

Layers	Neurons	Training Error	Testing Error
1	50	0.0613396	0.08028
1	100	0.0507844	0.0626947
1	200	0.0819546	0.0804184
2	50	0.0826735	0.0747892
2	100	0.0833333	0.0833333
2	200	0.0833333	0.0833333
3	50	0.0833333	0.0833333
3	100	0.0833333	0.0833333
3	200	N/A	N/A

Tabla III: Precision, Accuracy, y F1 Score para cada configuración con ReLu

Layers	Neurons	Precision	Accuracy	F1 Score
1	50	0.0123457	0.0123457	0.0123457
1	100	0.0438957	0.0438957	0.0438957
1	200	0.0397805	0.0397805	0.0397805
2	50	0.0164609	0.0164609	0.0164609
2	100	0.0411523	0.0411523	0.0411523
2	200	0.0288066	0.0288066	0.0288066
3	50	0.0411523	0.0411523	0.0411523
3	100	0.0466392	0.0466392	0.0466392
3	200	N/A	N/A	0.00137174

Se puede apreciar por las métricas de rendimiento obtenidas que la MLP con 3 capas ocultas, 100 neuronas por capa oculta y utilizando la función de activación ReLu muestra la mejor configuración frente a otras.

IV. Conclusiones

Se puede concluir por los resultados obtenidos que al aumentar las cantidad de capas ocultas en la MLP planteada se obtienen resultados más precisos. Además, al aumentar el número de neuronas por capa oculta se obtuvieron métricas de rendimiento significativamente diferente, mostrando mayor eficiencia al utilizar 100 neuronas. Por otro lado, con la aplicación de las funciones de activación se esperan resultados más precisos, con la función de activación ReLu siendo la que mejores métricas devolvió. Se puede resaltar la gran utilidad de las MLP al momento de reconocer la persona a la que le pertenece el audio introducido. Sin embargo, el coste computacional por ser una red completamente interconectada o la baja cantidad de datos disponibles pueden ser factores a tener en cuenta al momento de elegir esta arquitectura de red neuronal.

V. Material Complementario

Los códigos utilizados para la elaboración del presente trabajo se encuentran en el siguiente repositorio: GitHub.

VI. Bibliografía

[1] M. C. Golumbic, "Perfect Graphs," Annals of Discrete Mathematics, vol. 57, ch. 3, pp. 51-80, 2004.

[2] A. Sithara, T. Abraham, M. Dominic, "Study of MFCC and IHC Feature Extraction Methods With Probabilistic Acoustic Models for Speaker Biometric Applications (ICACC-2018)," Procedia Computer Science, no. 143, pp. 267-276, 2018.

[3] A. Awais, Y. Yu, A. Ahmed, S. Kun, S. Hayat, T. Tu, "Speaker Recognition Using Mel Frequency Cepstral Coefficient and Locality Sensitive Hashing," 2018 International Conference on Artificial Intelligence and Big Data, 2018.
[4] The MathWorks, Inc., "Speaker Identification Using Pitch and MFCC," Matlab version: R2023a, 2023.
[5] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, "State-of-the-art in artificial neural network applications: A survey," Cell Press Heliyon, vol. 4, no. 11, 2018.
[6] N. L. W. Keijsers, "Neural Networks," Encyclopedia of Movement Disorders, pp. 257-259, 2010.
[7] S. R. Dubey, S. K. Singh, B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," Neurocomputing, vol. 503, pp. 92-108, 2022.
[8] T. Szandała, "Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks," Bio-inspired Neurocomputing, pp. 203-224, 2020.
[9] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, O. Nieto, "librosa: Audio and music signal analysis in python," Proceedings of the 14th python in science conference, pp. 18-25, 2015.