ISSN 1870-4069

Construcción de una red neuronal replicadora de datos y una aplicación a clonación de voz

Anel Ramírez Álvarez, Mario Anzures García,
José Alejandro Rangel Huerta

Benemérita Universidad Autónoma de Puebla, México

anel.ramirez.al@gmail.com, marioanzuresg@gmail.com,
jose.rangelhuerta@viep.com.mx

Resumen. Las redes neuronales son sistemas de ecuaciones que nos permiten modelar comportamientos acordes a su modelo de construcción y entrenamiento. Como sabemos las redes neuronales realizan una pequeña emulación del comportamiento del cerebro humano, gracias a que forman nuevas conexiones internas en función de su aprendizaje. En este artículo de investigación se realiza la construcción de una red neuronal capaz de replicar datos mediante la compensación de diferencias entre los datos, esta construcción de red está inspirada en los sistemas de clonación de voz, donde para cierto timbre de voz se busca llegar a otro timbre de voz distinto, presentando así una red neuronal capaza de clonar datos. Por último, se presenta una aplicación de esta construcción neuronal a una clonación de voz a pequeña escala, teniendo en cuanta que en general una red neural aprende sola, no se tendrá que solucionar un problema tan complejo como lo es la detección de patrones bajo criterios estáticos que reconstruyan una voz, sino que la misma red es capaz de aprender a llegar al resultado deseado siendo entrenada. Este sistema, aunque prometedor, tiene limitantes dado la cantidad de información a la que debe acceder, por lo que se realiza una propuesta de evaluación para la red neuronal bajo la clonación de voz.

Palabras clave: Red neuronal, extracción automática de patrones, replicar datos, red neuronal simétrica, entrenamiento, compensar diferencia entre datos, clonación de voz, deep learning.

Construction of a Data Replicating Neural Network and an Application to Voice Cloning

Abstract. Neural networks are systems of equations that allow us to model behaviors according to their construction and training model. As we know, neural networks perform a small emulation of the behavior of the human brain, thanks

to the fact that they form new internal connections based on their learning. In this research article, the construction of a neural network capable of replicating data by compensating for differences between the data is carried out. This network construction is inspired by voice cloning systems, where for a certain voice timbre it is sought to reach another different timbre of voice, thus presenting a neural network capable of cloning data. Finally, an application of this neural construction to small-scale voice cloning is presented, considering that in general a neural network learns by itself, it will not have to solve a problem as complex as pattern detection based on criteria. Static that reconstructs a voice, but the network itself is capable of learning to reach the desired result by being trained. This system, although promising, has limitations given the amount of information it must access, so an evaluation proposal is made for the neural network under voice cloning.

Keywords: Neural network, automatic pattern extraction, replicate data, symmetric neural network, training, compensate difference between data, voice cloning, deep learning.

1. Introducción

El uso de Deep Learning, como lo son redes neuronales, resulta útil ya que solucionar problemas complejos donde la única ventaja es la consistencia en los datos es de gran utilidad ya que estas encuentran solas los patrones necesarios para converger en la solución requerida. Algunos de estos problemas complejos donde las redes neuronales trabajan eficientemente son en la clasificación de imágenes y textos, reconstrucción de imágenes, predicción de sucesos, procesos de control, entre otros.

Estos sistemas, aunque versátiles, tiene sus limitantes por la cantidad de información a la que debe acceder para poder lograr dicho fin, por ejemplo, las personas aprendemos de experiencias, cuanto más experiencia en un trabajo tengamos mejor podría ser nuestra respuesta a dicha labor, las redes neuronales funcionan de forma similar, cuanta más información recibe más precisa es su respuesta ante el problema específico en el cual se diseñó, es decir, el aprendizaje profundo se beneficia enormemente del acceso a grandes cantidades de datos.

Otro aspecto a tener en cuanta cuando trabajamos con redes neuronales es el diseño de su construcción; el número de neuronas en la capa de entrada, capas ocultas y neuronas de la capa de salida, ya que según sea su forma de construcción podremos modelaremos soluciones a problemas específicos, por ejemplo, para los ámbitos de clasificación tenemos que, para x número de neuronas de entrada, según sea el tamaño del dato a clasificar, solo tendremos cierto N número de salidas según el número de valores a identificar.

Ahora abordaremos la construcción de una red replicadora de datos, donde para x entradas tendremos el mismo número de salidas. Esta propuesta de construcción surgió como inspiración de los sistemas de clonación de voz (voice cloning), que son tecnologías desarrolladas que buscan imitar voces, estos sistemas se encuentran ya muy

presentes en la actualidad siendo usados a nivel comercial ya desde hace años por múltiples empresas de tecnología. Aunque también se plantea la cuestión de los posibles usos malintencionados como copias digitales de los patrones de voz, las cuales podrían burlar los complejos sistemas de seguridad.

Distintas medidas están siendo tomadas para combatir los intentos de aquellos que quieran usar de forma maliciosa las grabaciones. Estos sistemas desarrollados por empresas de tecnología han logrado grandes avances en este rubro por lo que es importante tomar a consideración sus avances y aportaciones. En la actualidad estos sistemas usan técnicas de deep learning, procesamiento de lenguaje natural, acoplamiento a voces artificiales, y/o muchas otras técnicas, además del uso de tecnologías que den soporte a grandes cantidades de información a las cuales necesitan acceder. A continuación, daremos un marco de contexto de estas tecnologías, en la metodología abordaremos el proceso de solución al problema planteado y concluyendo en la aplicación de esta propuesta de solución.

1.1. Modelos de inteligencia artificial

En ciencias de la computación, [1] una máquina inteligente ideal es un agente flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea. Coloquialmente, el término inteligencia artificial se aplica cuando una máquina imita las funciones cognitivas que los humanos asocian con otras mentes humanas, como, por ejemplo: percibir, razonar, aprender y resolver problemas.

Varios ejemplos se encuentran en el área de control de sistemas, planificación automática, la habilidad de responder a diagnósticos y a consultas de los consumidores, reconocimiento de escritura, reconocimiento del habla y reconocimiento de patrones. Los sistemas de IA actualmente son parte de la rutina en campos como economía, medicina, ingeniería, el transporte, las comunicaciones y la milicia, y se ha usado en gran variedad de aplicaciones de software, juegos de estrategia, como ajedrez de computador, y otros videojuegos.

El entrenamiento de un modelo de red neuronal en esencia significa seleccionar un modelo de la serie de modelos permitidos (o, en un bayesiano marco, la determinación de una distribución en el conjunto de modelos permitidos) que minimiza el criterio de costo. Hay numerosos algoritmos disponibles para la formación de los modelos de redes neuronales; la mayoría de ellos puede ser vista como una aplicación directa de la teoría de optimización y la estimación estadística.

1.2. Redes neuronales

El objetivo de la red neuronal [2] es resolver los problemas de la misma manera que el cerebro humano, aunque las redes neuronales son más abstractas. Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso.

Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una

función limitadora o umbral, que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona.

Esta función se conoce como función de activación. Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional.

Para realizar este aprendizaje automático, normalmente, se intenta minimizar una función de pérdida que evalúa la red en su total. Los valores de los pesos de las neuronas se van actualizando, buscando reducir el valor de la función de pérdida. Este proceso se realiza mediante la propagación hacia atrás.

1.3. Aprendizaje de una red neuronal

Lo que ha atraído el mayor interés en las redes neuronales es la posibilidad de aprendizaje. Dada una determinada tarea a resolver, el aprendizaje [3] consiste en utilizar un conjunto de observaciones para encontrar la cual resuelve la tarea de alguna forma óptima. Hay tres grandes paradigmas de aprendizaje, cada uno correspondiente a una tarea de aprendizaje abstracto en particular.

Estos son el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo. Al igual que calcular una formula, es cómo funcionan las redes neuronales, estás se basan en el cálculo de sumas ponderadas las cuales al calcular el valor de los coeficientes son capaces de acoplarse a nuevos comportamientos de sistemas en análisis. A continuación, se mostrará la solución al problema a la construcción de red neuronal propuesta.

2. Trabajos relacionados

Existen ya muchos sistemas desarrollados que han logrado grandes avances en el rubro de la clonación de voz por lo que es importante tomar a consideración sus avances y aportaciones. Algunos de ellos son Lyrebird una empresa emergente con sede en Montreal fue lanzada en 2017. Su objetivo: utilizar la inteligencia artificial para "crear las voces artificiales más realistas del mundo".

Su software se alimenta con y así decir lo que quiera, con la misma voz. El modelo de aprendizaje automático se creó para determinar los factores que hacen que cada voz sea única. A esto lo llaman el ADN de la voz. Cada vez que prueban una nueva voz, el algoritmo averigua en qué se diferencia de las otras voces de nuestra base de datos y en qué se parece. Su inteligencia artificial basada en redes neuronales y algunos petaflops de potencia informática de alto rendimiento trabajando detrás de escena para darle a una persona o marca una voz (casi) real.

Existen también desarrollos creados por investigadores de IA con artículos publicados [4] y otros como el gigante chino Baidu, quien con sólo enunciar una oración su sistema es capaz de crear una copia sintética de la voz humana. Hasta no hace mucho la industria de la clonación de voces se concentraba en un nicho de mercado

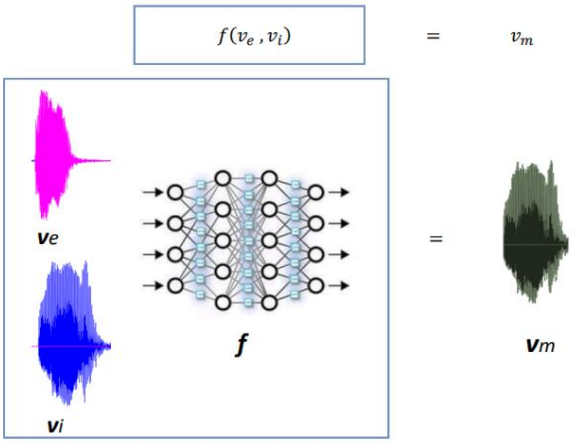

Fig. 1. Diagrama de representación de la ecuación 1.

muy reducido que buscaba atender las necesidades de aquellos que por motivos médicos o particulares [5].

3. Método

Como se vio en el apartado anterior esta construcción de red neuronal propuesta surge como inspiración de los sistemas de voice cloning, que usualmente utilizan en todo su proceso algún sistema de deep learning basados en una representación del conocimiento por medio de listas de datos de la voz. Para plantear una metodología, primero planteamos los requerimientos iniciales del Sistema, que son los siguientes: procesamiento de grandes cantidades de datos, sistema sometido a entrenamiento, sistema que converge en una solución, moldeable a modificaciones requeridas por el sistema, someter la información a modificaciones e iterar hasta encontrar la solución.

3.1. Propuesta de red neuronal

Se propone la construcción de una red simétrica de una sola capa y tres interconexiones para cada neurona, llamada en la literatura como Red Neuronal Monocapa. A continuación, se explica el motivo de cada uno de los parámetros mencionados que son requeridos por el sistema y el diagrama general de la red. La construcción general de la RN se puede reducir a la siguiente ecuación:

$$\boldsymbol{f}(ve[\text{amplitud}], vi[\text{amplitud}]) = vm[\text{amplitud}], \quad (1)$$

donde $\boldsymbol{f}$ representa la construcción general de la RN en función de ve (voz de entrad) y vi (voz imitada) para tener como salida la voz imitada. La ecuación anterior requiere

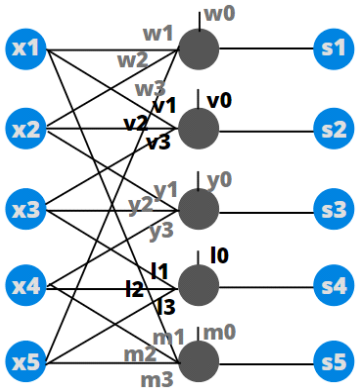

Fig. 2. Red mono capa simétrica de 5 neuronas.

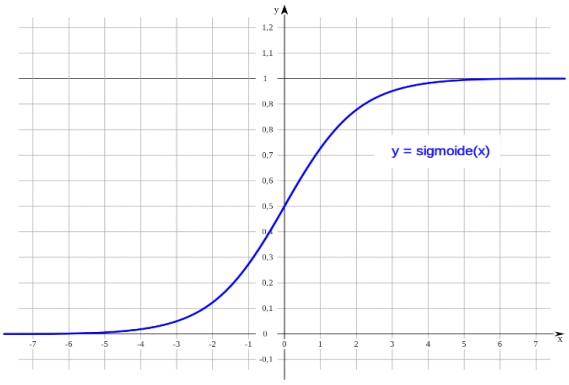

Fig. 3. Función de activación Sigmoide.

de una simetría del tamaño de número de datos necesarios, es decir, que el tamaño de audios debe ser del mismo número de muestras y sujetas al mismo tiempo de muestreo, por lo cual se fijó este requerimiento para la construcción de la RN.

La ecuación anterior $\boldsymbol{f}$(voz de entrada, voz a imitar) se desglosa por la RN con la totalidad de las sumas ponderadas que conforman la red. Para explicar en un nivel más detallado se muestra a continuación un diagrama de dicha ecuación.

A continuación, se explicará por puntos la construcción de la red a partir de las entradas y salidas requeridas por el sistema $\boldsymbol{f}$():

- Ya que buscamos crear un modelo en el que tendremos entradas y salidas de las mismas dimensiones, se propone construir una red simétrica que no modifique el tamaño de la red.
- Ya que para manipular datos simples sólo necesitamos una capa se propone el uso de una red monocapa. Para evitar así el uso de más variables que resultarían innecesarias para los resultados.
- El motivo de las tres interconexiones es por motivos de ahorro de variables.

Con esto en mente se construyó una red mono capa simétrica de 5 neuronas, es decir, que requerirá 5 datos de entrada vi, 5 datos de salida vm y 5 datos a los que se desea imitar vi.

Para definir la función de activación lo realizamos en función del resultado que necesitamos. Ya que las señales son normalizadas, es decir, el resultado está en el rango de [0,1] en pasos de hasta 0.1 décimas. Se propone el uso de la función sigmoide. Es muy importante decir que el peso de wi, vi, yi, li y mi. Se definió a partir de la siguiente ecuación: $-5 = w(0.1)$, ver la gráfica siguiente ya que la sigmoide comienza a converger en 1 y 0 para los valores de 5 y -5 respectivamente. despejando entremos que el peso es de $w = 50$ para ambos rangos entonces definimos los pesos en el intervalo de [-50, 50]. Donde tenemos que $y = f(x) = s$; es decir, que al tener valores en x de -5 a 5 tendremos como resultado para s un rango de valores de 0 a 1, que son los valores de nuestras señales de audio a variar.

Teniendo ya definida la función de activación a usar y los pesos que podrán ir recorriendo todos los valores para $\boldsymbol{s}$ de 0 a 1. Se muestran a continuación las ecuaciones que forman el diagrama de la Fig.2. Las ecuaciones siguientes son obtenidas de la generalización de la regla delta.

Ecuaciones requeridas para el sistema de red neuronal:

$$S_1 = f(w_1x_1 + w_2x_2 + w_3x_5 + w_0), \tag{2}$$

$$S_2 = f(v_1x_1 + v_2x_2 + v_3x_3 + v_0), \tag{3}$$

$$S_3 = f(y_1x_2 + y_2x_3 + y_3x_4 + y_0), \tag{4}$$

$$S_4 = f(l_1x_3 + l_2x_4 + l_3x_5 + l_0), \tag{5}$$

$$S_5 = f(m_1x_1 + m_2x_4 + m_3x_5 + m_0). \tag{6}$$

Ecuaciones para el cálculo del error:

$$\Delta_{S_1} = SD_1 - S_1, \tag{7}$$

$$\Delta_{S_2} = SD_2 - S_2, \tag{8}$$

$$\Delta_{S_3} = SD_3 - S_3, \tag{9}$$

$$\Delta_{S_4} = SD_4 - S_4, \tag{10}$$

$$\Delta_{S_5} = SD_5 - S_5. \tag{11}$$

Pesos o coeficientes de las sumas ponderadas:

$$w_O = w_o + \Delta_{S_1}, \tag{12}$$

$$w_1 = w_1 + \Delta_{S_1}x_1, \tag{13}$$

Tabla 1. Tabla de datos para la propuesta de red neuronal del sistema (Fig. 9).

Datos iniciales (voz original) *ve* ***Xi***	**Datos para imitar (voz a imitar) *vi*** ***SDi***	**Datos de resultado de la Red Neuronal (voz imitada) *vm*** ***Si***
0.2 a 0.1e	0.4 ar	0.40000000000000013
0.4 a	0.5 er	0.5000000000000002
0.5 a	0.55	0.5500000000000007
0.9 a	0	0.0020049821385272233 ≈ 0
0.7 a	0	0.0022907827937611648 ≈ 0

$$w_2 = w_2 + \Delta_{S_1} x_2, \quad (14)$$

$$w_3 = w_3 + \Delta_{S_1} x_5, \quad (15)$$

$$v_O = v_o + \Delta_{S_2}, \quad (16)$$

$$v_1 = v_1 + \Delta_{S_2} x_1, \quad (17)$$

$$v_2 = v_2 + \Delta_{S_2} x_2, \quad (18)$$

$$v_3 = v_3 + \Delta_{S_2} x_3, \quad (19)$$

$$y_O = y_o + \Delta_{S_3}, \quad (20)$$

$$y_1 = y_1 + \Delta_{S_3} x_2, \quad (21)$$

$$y_2 = y_2 + \Delta_{S_3} x_3, \quad (22)$$

$$y_3 = y_3 + \Delta_{S_3} x_4, \quad (23)$$

$$m_O = m_o + \Delta_{S_5}, \quad (24)$$

$$m_1 = m_1 + \Delta_{S_5} x_1, \quad (25)$$

$$m_2 = m_2 + \Delta_{S_5} x_4, \quad (26)$$

$$m_3 = m_3 + \Delta_{S_5} x_5, \quad (27)$$

$$l_O = l_o + \Delta_{S_4}, \quad (28)$$

$$l_1 = l_1 + \Delta_{S_4} x_3, \tag{29}$$

$$l_2 = l_2 + \Delta_{S_4} x_4, \tag{30}$$

$$l_3 = l_3 + \Delta_{S_4} x_5. \tag{31}$$

Condiciones de paro, como bien sabemos son a variar:

1. $\sum_{i=0}^{N=5} \Delta s_i \leq 0.0001$,
2. Número de iteraciones < 1000.

Este proceso se implementó en Python y se puso a prueba para ciertos datos de ejemplo, dándonos los resultados siguientes.

Los Pesos de wi, vi, yi, li y mi calculados por la Red Neuronal, se comprobaron en el sistema y posteriormente en la función de activación para validar la información resultante. Hay que recordar que los datos solución en modelos de redes neuronales cambian con una nueva corrida de la red, es decir, nuevos datos que de igual forma convergerán a la solución deseada. No son datos de solución únicos para algún modelo de entrenamiento, por ejemplo, en este caso para nuestra Tabla 1, habrá distintas combinaciones de valores para wi, vi, yi, li que solucionan estos datos de entrenamiento.

La propuesta de construcción se sometió a múltiples pruebas como los de la Tabla 1 y vemos como los resultados son los deseados, de igual forma se probó para más datos dando resultados igualmente favorables, a continuación, probaremos esta construcción para una cantidad mayor de datos, para el fin de este artículo, con datos de señales de audio.

3.2. Modificación de datos

El procesamiento digital de señales o DSP (sigla en inglés de digital signal processing) es la manipulación matemática de una señal de información para modificarla o mejorarla en algún sentido. A continuación, se presentan las técnicas más comunes para la modificación de sonoridad de señales de audio.

3.3. Manipulación de volumen de una señal de audio

Para la variación de volumen se modifica la amplitud de la señal, esto mediante la multiplicación de la señal por una contante. Si multiplicamos por una constante mayor a uno esta aumentará su volumen a dicha proporción y si se multiplica por una constante menor a uno esta disminuirá su volumen.

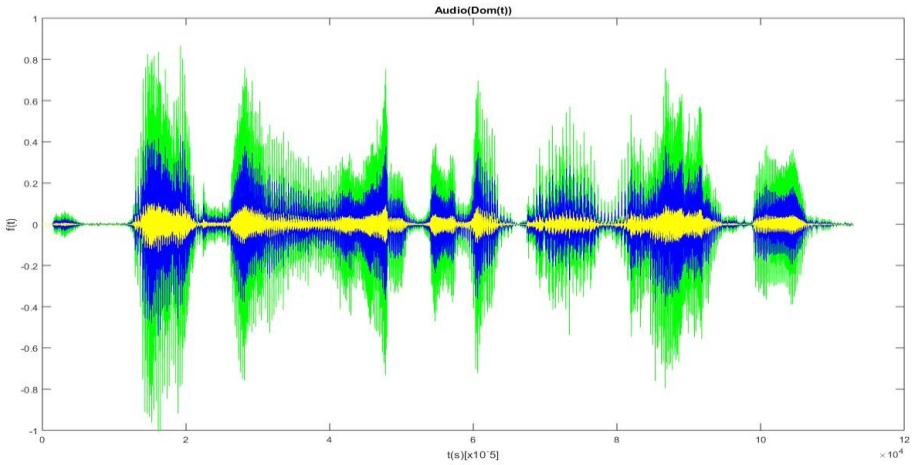

Fig. 4. Modificación de una señal de audio.

Aumento de volumen:

$$\text{Amplitud de la señal} \cdot X \geq \text{Amplitud de la señal,} \quad \text{donde es un entero} \geq 1. \tag{32}$$

Disminución de volumen:

$$\text{Amplitud de la señal} \cdot X < \text{Amplitud de la señal,} \quad \text{donde es un entero} < 1. \tag{33}$$

En la Fig. 4, podemos ver que la gráfica azul es la señal original, la gráfica verde es la original multiplicada por una constante positiva (aumento de volumen) y la gráfica amarilla es la original multiplicada por una fracción (disminución de volumen).

3.4. Implementación de una colección de datos a la red neuronal

Hasta aquí hemos visto uno de los requisitos para voice cloning que es el tratamiento de las señales, ahora se implementó la propuesta algorítmica por Redes Neuronales que modificara sistemáticamente los datos hasta lograr su fin, llegar a una nueva colección de datos deseada, ver Fig. 5. En las siguientes pruebas se usará Keras de Tensorflow para probar la propuesta de construcción de la RN para palabras simples.

3.5. Pruebas del modelo

Para la implementación del Sistema se realizó el entrenamiento con 5 palabras simples de la Tabla 2. Esto bajo la premisa de volver experta a la red en ciertas formas de conocimiento de sonidos similares para que a la hora de someterlo a pruebas pueda ser capaz de formar predicciones bajo conocimientos adquiridos, intentando llenar así lagunas de no conocimiento, dado que la red neuronal funciona bajo memoria de

Tabla 2. Tabla de datos para el entrenamiento de la red neuronal.

Audio Hombre	Audio Mujer
Palabra 1: casa	Palabra 1: casa
Palabra 2: bueno	Palabra 2: bueno
Palabra 3: rama	Palabra 3: rama
Palabra 4: coma	Palabra 4: coma
Palabra 5: codera	Palabra 5: codera
Palabra 6: hola	Palabra 6: hola
Palabra 7: correr	Palabra 7: correr

conocimientos, ver Fig. 6, donde intentamos predecir un audio no conocido (circulo verde) a partir del conocimiento de audios conocidos similares (circunferencias verdes).

Como sabemos para realizar cualquier uso de datos debemos de tener una preparación o consideraciones en los datos, en este caso las palabras se grabaron bajo las mismas condiciones de frecuencia de muestreo en la captura de la señal. Y posteriormente se normalizaron, es decir, se buscó el valor máximo de la señal y se dividido la señal entre este para que todos los audios tuviesen un rango de -1 a 1. De esta manera, la red neuronal no intentará compensar el volumen de las muestras (remediar el volumen de las señales), y solo se centrará en la sonoridad. Esto con el fin de mejorar la eficiencia de la red.

A continuación, se muestran los puntos para preparar las señales:

- Normalizar los audios, con el fin de que el factor de volumen no sea compensado por la red.
- Recortar la señal a la misma cantidad de datos para todos los audios (misma longitud de datos).
- Imprimir los datos en un array ordenado, preparado para Python.
- Implementación del trining con Keras.
- Pruebas de la red entrenada resultante en Matlab.

3.6. Implementación del sistema con keras de tensorflow

En Python usando el entorno de Colab se implementó la red neuronal de la Fig 7 con la ayuda de la API de Keras de Tensorflow, el uso de esta herramienta se debido a la flexibilidad de uso y acoplamiento a grandes cantidades de datos a los que necesitamos procesar. A continuación, se explican los pasos seguidos en el código.

1 Importamos Tensorflow y la librería de NumPy.

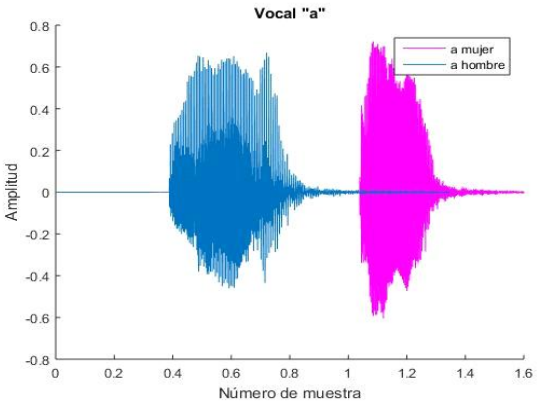

Fig. 5. Señales de audio de la vocal “a” en voces de una mujer y un hombre.

Fig. 6. Converger a la solución deseada no conocida por la red.

2 Declaramos los datos usados en la red neuronal, los datos de la Tabla 2.

3 Declarara en un solo array los datos de entrada y en un segundo array los datos de salida de la red neuronal, con los cuales se somete a entrenamiento la red.

4 Declarar el número de entradas de la capa de entrada de la red, en este caso es el tamaño de cada muestra de audio, construcción de la Fig. 2, 1900 neuronas en la capa de entrada y salida.

3.7. Resultados

Se entrenó la Red Neuronal con palabras simples. Con una voz de hombre e intentando llegar a una voz de mujer, de igual forma otra red se entrenó con voz de mujer para obtener la voz de un hombre respectivamente. Ver Fig. 7. La longitud de todas las palabras ingresadas fue de 19 mil datos en los arrays. Con lo cual el resultado de la red es de igual forma de 19 mil. Por lo que las pruebas se realizaron con palabras que puedan entrar dentro de este rango de longitud.

En la Fig. 8 y 9 podemos ver los datos de entrenamiento a los cuales fue sometido el sistema en Tensorflow.

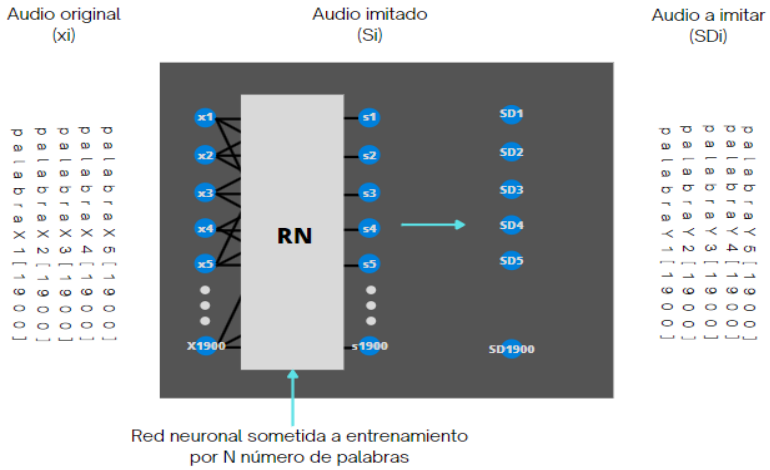

Fig. 7. Entrenamiento de la Red Neuronal propuesta.

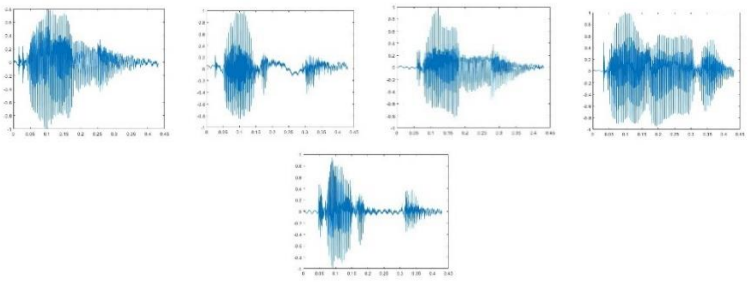

Fig. 8. Audios implementados en el entrenar el sistema: Tipos de Audios Deseados, Tono de voz de una Mujer.

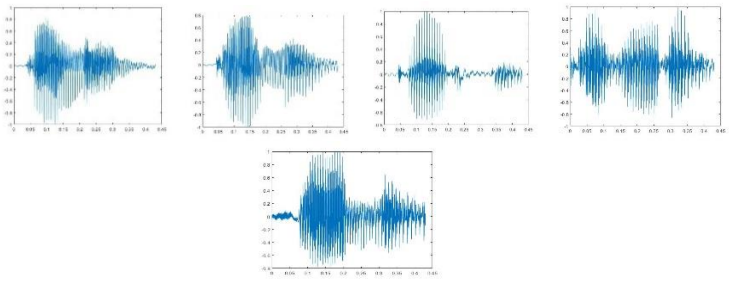

Fig. 9. Audios implementados en el entrenar el sistema: Audios de Entrada, Tono de voz de un Hombre.

En la Fig. 10. Podemos ver la estabilización de la función de pérdida de la red neuronal al intentar llegar a los resultados deseados, en función del número de épocas o iteraciones que tuvo que realizar la red para lograr minimizar errores, logrando así el

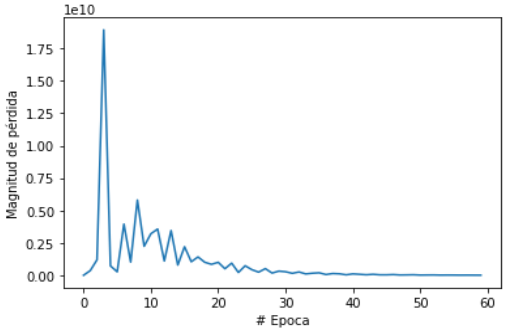

Fig. 10. Estabilización de la Red Neuronal para llegar al resultado deseado.

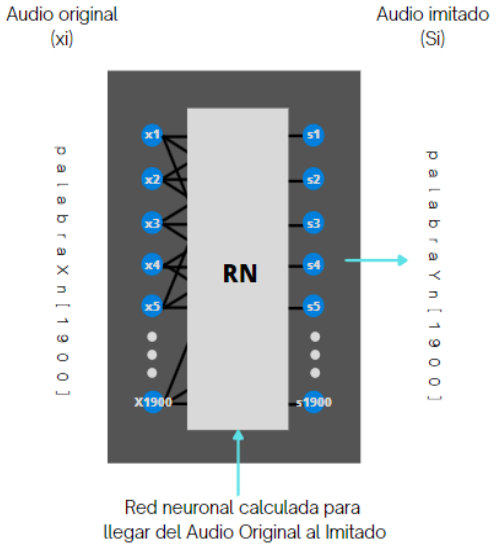

Fig. 11. Prueba de la Red Neuronal propuesta.

propósito de los resultados. Se implementaron las palabras conocidas por el sistema, es decir, se ingresaron todas las palabras con las que se entrenó el sistema en la etapa anterior, ver Fig. 11.

Esta etapa dio resultados correctos para todos los datos, es decir, se ingresó la palabra “coma” en la red neuronal con la voz de un hombre y sonó como resultado la palara “coma” con la voz de una mujer, es importante mencionar que el resultado arrojo los sonidos deseados, pero con ruido en ellas, ver Fig. 12.

Donde se escuchaba la palabra deseada junto con ruido. Existen técnicas de eliminación de ruido, pero esta implementación solo buscaba el estudio de la red neuronal propuesta de la Fig. 2.

Para la prueba final del sistema de igual forma se implementó la Fig. 11, pero para palabras no conocidas por el sistema, palabras no ingresadas en la etapa de

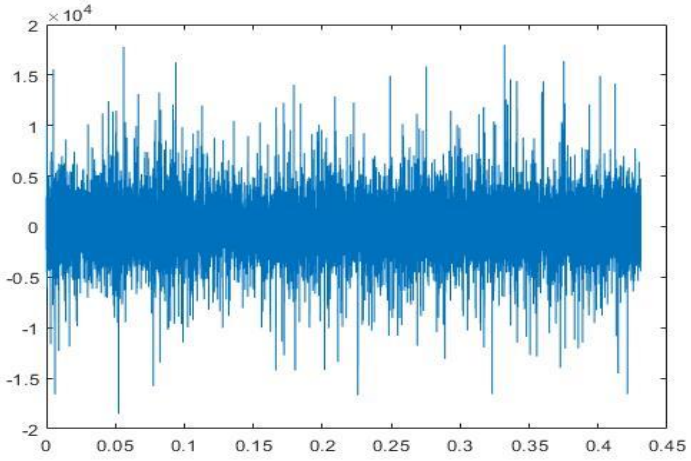

Fig. 12. Palabra imitada por la RN con ruido generado por la misma RN.

entrenamiento del Modelo, en base a la Fig. 6. Ingresando datos de voz de un hombre y dando como resultado un timbre de voz de mujer.

En esta etapa de prueba los resultados fueron favorables ya que todos se recreaba la voz de una mujer, pero con ciertos problemas de vocalización al pronunciar las palabras, dado lo complejidad del problema, ya que son sistemas que requieren de más información dado el funcionamiento de las redes neuronales. [6]

4. Conclusiones y trabajo futuro

La construcción de red neuronal propuesta logro resultados favorables al replicar los datos deseados de ciertos datos origen, el sistema se puso a prueba con múltiples valores, como el ejemplo de la Tabla 1, dando resultados siempre favorables. Dado esto se buscó la implementación de datos más complejos para simular a pequeña escala los sistemas de clonación de voz, donde la red neuronal termino construyendo un sistema para llegar de la voz de un hombre a la voz de una mujer, intentando en todo momento llegar a la voz a clonar.

Con ello podemos decir que la RN propuesta es capaz de replicar datos de N dimensiones, probando el sistema con pocos datos y hasta miles de datos, esto último llevándolo a temas de clonación de voz. Por lo cual, la red logró recrear sonidos simples artificialmente. Aunque no sea un sistema escalable por sus altos requerimientos y baja flexibilidad para acoplarse a palabras de distintos tamaños queda analizar cuestiones como aumentar el entrenamiento de la RN con cantidades masivas de datos para que pueda ser capaz de obtener más información de la identidad de los datos a los que la RN replica en el entrenamiento.

Referencias

1. Uday-Kamath, J. L., James, W.: Deep learning for NLP and Speech. USA: Springer (2019)
2. Navin, M.: Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition with TensorFlow and Keras. USA. Apress (2018)
3. Rudolph, R.: Redes Neuronales: Guia Sencilla de Redes Neuronales Artificiales. USA. CreateSpace Independent Publishing Platform (2018)
4. Ye, R., Weiss, F., Wolfgang, M., Melvin, J., Zhifeng, C., Yonghui, W.: Direct speech-to-speech translation with a sequence-to-sequence model.
5. Sercan, Ö., Arık, C., Kainan, P.: Neural Voice Cloning with a Few Samples
6. Anel, R.A.: Modelo de entrenamiento y testing de la RN. Neuralnetworktest (2022)