Traductor de Lenguaje Natural a Lenguaje de Señas

Eduardo Ruiz Rios, *Ingeniería Informática (Universidad Veracruzana*, Blvd. Adolfo Ruiz Cortines 455, Costa Verde, 94294 Veracruz, Ver., México, zs19002907@estudiantes.uv.mx*)*

***Resumen*—La visión artificial, también conocida como visión por computadora, forma parte de las tecnologías que conforman la industria 4.0. Su objetivo es simular la visión humana para identificar objetos, procesar, analizar, obtener y generar información a partir de las imágenes o vídeos de tal manera que los agentes inteligentes puedan *"comprender"* su entorno y puedan interactuar con él.**

El propósito de este proyecto es mostrar la potencia de combinar el procesamiento de imágenes con algoritmos de aprendizaje de manera que podamos dotar a nuestros programas con cierto grado de aprendizaje para que sean capaces de resolver una amplia variedad de problemas. En el presente se muestra cómo con el procesamiento de imágenes en conjunto con los algoritmos de aprendizaje podemos crear un traductor de lenguaje natural a lenguaje de señas.

Palabras clave—Redes neuronales, procesamiento de imágenes, visión por computadora, agentes inteligentes, aprendizaje profundo.

I. Introducción

De acuerdo con los datos recolectados del INEGI a través del censo realizado en 2020, en Veracruz Ignacio de la Llave hay una población de 8,062,579 personas en total. De esta población aproximadamente 300 mil individuos cuentan con alguna discapacidad del habla o auditiva que no les permite comunicarse adecuadamente. Actualmente en México, ha habido poco apoyo hacia programas de integración social para lograr que personas con este tipo de discapacidades sean capaces de desarrollarse plenamente. Es decir, no hay personas capacitadas en el sector público ni privado para poder entablar una comunicación utilizando el lenguaje de señas mexicano. Con el propósito de integrar a esta parte de la población a la sociedad y como promoción de una cultura de inclusión, se crea un programa capaz de traducir lenguaje natural (escrito) a lenguaje de señas. Para ello toma la sentencia en lenguaje de natural y mediante un procesamiento de imágenes y del lenguaje logra determinar la secuencia de imágenes de lenguaje de señas que equivalen a la oración original. Más adelante en la sección *III. Diseño e implementación* se explicará mejor cómo funciona el programa.

II. Estado del arte

Una de las principales problemáticas en el estudio de los datos en inteligencia artificial es la clasificación de las características ya sean de bases de datos de las cuales se puedan cuantificar y obtener descriptores que sean lo más exactos posibles. Para ello es necesario saber a qué conjuntos pertenecen. Para llevar a cabo lo anterior, es necesario conocer las diversas técnicas de la ciencia de datos, como las redes neuronales y el *machine learning*.

Una de las primeras proposiciones de las neuronas artificiales fue en el año de 1943 por Warren McCulloch, un neuropsicólogo, y Walter Pitts, en colaboración con la Universidad de Chicago. El concepto de redes neuronales ya había sido propuesto anteriormente por Alan Turing en 1948, en su artículo *"Intelligent Machinery"* con la creación de la prueba de Turing, una máquina compuesta de múltiples compuertas lógicas tipo NAND interconectadas. La máquina de Turing se considera precursora de las redes neuronales modernas.

Las redes neuronales han avanzado en gran medida en los últimos años para el procesamiento del lenguaje natural. Lo que se busca es simular el lenguaje natural de una neurona y así entrenar modelos de algoritmos que aprendan a resolver determinadas tareas. El desarrollo de este tipo de modelos se vio frenado en gran medida por las limitaciones de hardware de la época. Gracias a los avances tecnológicos de la informática y la electrónica, hoy en día es posible desarrollar estos modelos permitiéndonos aplicarlos para la generación de programas que *"entiendan"* el lenguaje natural.

III. Diseño e implementación

Para la realización de este traductor de lenguaje natural a lenguaje de señas se utilizó el lenguaje de programación Python en su versión 3.10.6, un lenguaje ampliamente usado para el campo de ciencia de datos y machine learning por las diversas librerías que existen para tal fin. Entre ellas se utilizaron las siguientes: Numpy una librería especializada en el tratamiento de matrices (dado que manipulamos imágenes y estas son representadas en la computadora como una matriz de pixeles), Matplotlib que nos permite crear gráficas y mostrar imágenes, Pandas para tratar con grandes cantidades de datos, Tensorflow para la creación de los modelos predictivos y redes neuronales y scikit-learn para el proceso de *one-hot encoding*.

Para comenzar necesitamos una fuente de datos. Podemos obtener nuestra fuente de datos de dos maneras:

1. Recolectándola nosotros mismos, o
2. Utilizando algunas de las que se pueden encontrar por internet

En mi caso utilicé una fuente de datos obtenida de las múltiples fuentes de datos que ofrece *Kaggle*. Independientemente del método de recolección de datos, será necesario hacer una clasificación de esta, es decir, del 100% que representa nuestra fuente de datos una parte la utilizaremos como conjunto de entrenamiento, otra parte

como conjunto de validación y otra parte como conjunto de prueba. En pocas palabras, el conjunto de entrenamiento, como su nombre lo indica, servirá para entrenar a nuestro modelo para que sea capaz de reconocer, clasificar y relacionar las letras del alfabeto con su equivalente en lenguaje de señas; en cuanto a volumen de datos se refiere, el conjunto de entrenamiento es el más extenso pues la intención es alimentar al modelo con la mayor cantidad de variaciones para que pueda predecir en el futuro satisfactoriamente. El conjunto de validación se utiliza para verificar que el modelo aprendió a clasificar e identificar satisfactoriamente. Finalmente, el conjunto de prueba sirve como una última verificación para asegurarnos que todo funcione correctamente y se realiza con datos que el modelo nunca ha visto.

La red neuronal con la que opera esta IA es una red neuronal convolucional. Este tipo de redes neuronales tienen las siguientes características:

1. Este tipo de redes neuronales son las más eficientes y aptas para los problemas de clasificación de imágenes.
2. Las redes neuronales convolucionales son redes neuronales compuestas por capas convolucionales.
3. Conforme los datos van avanzando en las capas de la red, esta es capaz de encontrar patrones más complejos.
4. Una vez que un patrón ha sido identificado puede identificarlo en cualquier otra posición.

Esta es la morfología o topología de la red neuronal propuesta:

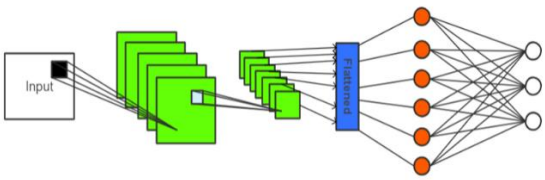

Figura 1. Red neuronal convolucional

Los pasos para llevar a cabo la tarea son los siguientes:

1. **Preparar los datos:** primero debemos entender cómo están los datos de nuestra fuente de datos. Nuestra fuente de datos consta de dos archivos de valores separados por comas (*comma separated values*, *csv* por sus siglas en inglés) que contiene los valores de los pixeles de cada imagen. Sino tuviésemos de esta manera nuestra fuente de datos, habría que hacerlo manualmente.
2. **Normalizar los valores:** ya que estamos trabajando con imágenes en escala de grises, cada píxel puede tomar valores dentro del intervalo [0,255]. El valor de 0 representa el mínimo valor y 255 el máximo que puede adoptar un píxel. El valor 0 representa el color negro y el 255 un color blanco. Todos los demás valores representan el color gris (el valor 127) y variaciones de él. Para facilitarle al modelo la manipulación de estos datos normalizaremos dichos valores en un intervalo de valores continuos de [0, 1] dividiéndolos entre 255.
3. **Binarización de las etiquetas:** llegados a este punto aplicamos el proceso de *one-hot encoding* a nuestras etiquetas. Las etiquetas, en esencia, son las categorías de nuestro problema. Para este caso las categorías corresponden a las letras del alfabeto. Para realizar este proceso, debemos contar con n vectores unidimensionales cuya dimensión sea igual a la cantidad de etiquetas que tenemos. Una vez esto solo resta colocar un 1 en la posición que representa a nuestra categoría y dejar las demás en ceros. Por ejemplo, supongamos que tenemos letras como en este caso, las etiquetas serían cada una de las letras del alfabeto. Para no extender demasiado la explicación utilizaremos solamente las letras *a*, *b* y *c*. Entonces tendríamos 3 vectores unidimensionales de tamaño 3. Entonces tendríamos:

$$a = [1\ 0\ 0]$$

$$b = [0\ 1\ 0]$$

$$c = [0\ 0\ 1]$$

4. **Separación de los datos de validación:** del 100% de los datos de nuestra fuente de datos, destinamos el 90% para entrenamiento y el 10% restante para la validación.
5. **Construcción del modelo:** construcción de la red neuronal.
6. **Entrenamiento del modelo:** entrenamos nuestro modelo con el 90% de los datos y verificamos su exactitud.

IV. Resultados

A partir del modelo propuesto se obtienen los siguientes resultados:

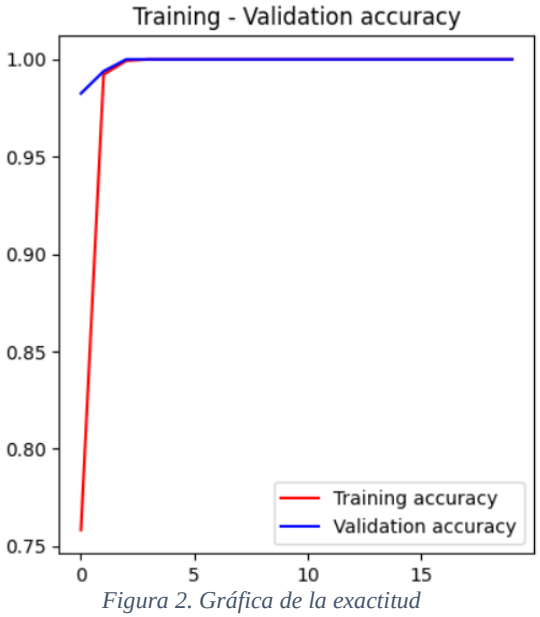

Figura 2. Gráfica de la exactitud

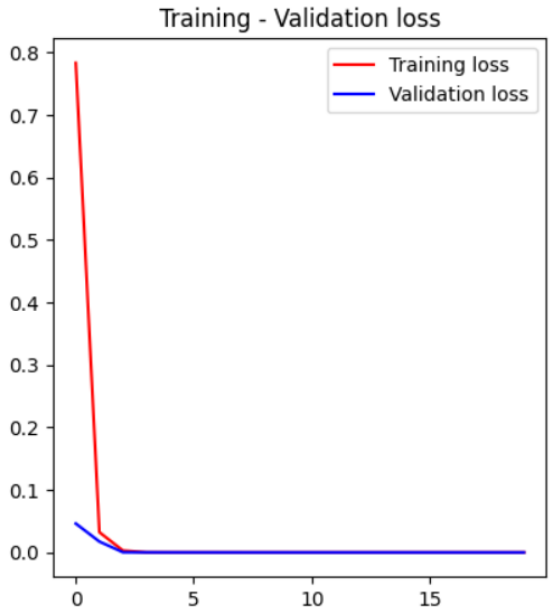

Figura 3 Gráfica de la pérdida

Como podemos notar, la exactitud para el conjunto de entrenamiento y de validación alcanzan rápidamente el valor de 1 o 100%. Similarmente, notamos como el error disminuye drásticamente casi hasta cero muy rápidamente. Esto es un indicativo de que la morfología del modelo funciona bien y se puede apreciar la curva de aprendizaje de este.

V. Conclusiones y trabajos futuros

Actualmente el programa solo es capaz de traducir lenguaje natural a lenguaje de señas lo cual es un avance en lograr que las personas con discapacidad para comunicarse puedan integrarse a la sociedad y para favorecer una cultura de inclusión. Sin embargo, esto es solo la mitad del camino. El modelo podría ser mejorado para incorporar una cámara que, a partir de la captura de vídeo, permita obtener las imágenes extraídas del vídeo de manera que ahora se pueda realizar la traducción inversa: de lenguaje de señas a lenguaje natural para así poder cumplir con el ciclo comunicativo emisor-receptor-emisor.

Referencias

[1] R. E. W. Rafael C. Gonzalez, Digital Image Processing, New Jersey: Prentice Hall.

[2] P. N. Stuart J. Russell, Artificial Intelligence : A Modern Approach, Pearson, 2021.

[3] INEGI. "Población. Discapacidad". Bienvenidos a Cuéntame de México, s.f.

[4] A. Chow. "Coding an AI to Recognize Sign Language with Tensorflow and Keras". Medium, 2021.

[5] J. Brownlee. "Why One-Hot Encode Data in Machine Learning? - MachineLearningMastery.com". MachineLearningMastery.com, 2020.