Reconocimiento del Lenguaje de Señas mediante Aprendizaje Profundo

García Avendaño Martín Jesús
martin.garcia@iimas.unam.mx

Resumen

Este proyecto explora la utilidad del aprendizaje profundo en la identificación del lenguaje de señas mediante una red convolucional la cual ha sido entrenada con un conjunto de datos de mas de 87000 fotografías que ilustran las distintas letras del alfabeto ingles de señas.

1. Introducción

FIGURA 1: Letra 'V'del alfabeto de señas ingles.

El lenguaje de señas representa desde su invención en el siglo XV una alternativa para todas aquellas personas que por algún motivo se ven privadas de la capacidad auditiva y del habla sin embargo en pleno siglo XXI, las personas sordo-mudas sufren de un alto rango de exclusión, ya que estos normalmente solo pueden comunicar de forma clara sus ideas con otras personas que presentan esta característica y algunas veces con su circulo familiar aunque este no siempre es el caso, de esta forma se ven impedidos de comunicarse con el resto de las personas que los rodean haciendo que conseguir un empleo o pedir comida en un restaurante sea una tarea casi imposible, por lo que es de vital importancia poner manos a la obra en el desarrollo de soluciones que permitan a aproximadamente 72 millones de personas comunicarse con el resto del mundo.

La tecnología y su constante avance le han permitido a la humanidad dar solución a gran cantidad de problemas incluyendo el lenguaje de señas, desde hace tiempo se han empezado a buscar alternativas para dar "voz.ª estas personas que no la tienen, hasta hace unos años los brazaletes, pulseras y guantes con sensores estaban presentando grandes avances en la resolución de este problema sin embargo estos resultaban sumamente costosos de fabricar y muy poco prácticos en la vida útil.

El desarrollo a pasos agigantados del aprendizaje profundo y la ciencia de datos en general durante los últimos años ha permitido la aplicación de estos a casi cualquier problema incluyendo el lenguaje de señas que se ha convertido en un problema de particular interés para los investigadores especialmente para la gente que se dedica a la visión computacional como Oscar Koller, Hermann Ney, Richard Bowden que presentaron un trabajo relacionado en la conferencia sobre visión computacional en 2015, el interés de toda clase de investigadores recae en la complejidad del proyecto ya que este se encuentra rodeado de un sin numero de variables al querer ser implementado en un entorno real que no solo supondría el reconocimiento de letras si no de frases.

Las redes neuronales convulocionales se asemejan a las neuronas en la corteza visual de un cerebro biológico, son una variante de un perceptron multicapa pero estas trabajan con matrices bidimensionales lo que las hace idóneas para trabajar con imágenes.Las redes convoluciones funcionan básicamente de la siguiente forma: Supongamos que le pasamos a la red una imagen de 28 píxeles de alto por 28 de ancho que en realidad en una matriz de 784 valores entre el 0 y el 255 pero normalizados estos valores van del 0 al 1,por lo que en la red usaremos 784 neuronas,este numero de neuronas solo es valido si se aplica a una imagen de 1 solo color es decir escala de grises pero si tenemos una imagen a color tendremos 28x28x3 el tres se debe al espectro RGB y al hacer la operación nos da 2352 es decir ocuparemos 2352 neuronas con una imagen de color, construyendo de esta forma nuestra capa de entrada, ya que tenemos nuestra capa de entrada procederemos ha hacer lo que distingue a nuestra

red las **convoluciones** que consisten en tomar grupos de pixeles cercanos de la imagen de entrada y sacar el producto escalar de estos y el kernel que es una pequeña matriz supongamos de 3x3 arrojando una nueva matriz de salida que sera nuestra **nueva capa de neuronas ocultas**, si la imagen fuera a color tendríamos un kernel de 3x3x3, se suman las matrices de esos tres filtros, se le suma una unidad bias y se obtiene una sola matriz, los kernel cumplen la función de filtrado para extraer la mayor cantidad de características de la imagen. En cada convolución se aplican varios kernels por ejemplo 32 es decir tendríamos 32 filtros y por lo tanto 32 matrices de salida las que llamaremos el **featuring map** o mapa de características que se podrían interpretar como 32 nuevas imágenes y cada una posee diferentes atributos de la imagen original.Estas 32 nuevas imágenes o **featuring map** componen la nueva **capa de neuronas ocultas** la cual tendrá un valor en el caso de una imagen de un solo canal de 25088 neuronas lo cual constituye un poder computacional grande por lo que la mejor opción es realizar un muestreo tratando de conservar las características mas importantes de cada matriz por ejemplo con el muestreo Max-Poling, recorreremos cada matriz de las 32 de nuestro featuring map y haremos grupos de cuatro es decir 2x2, de esta forma se conserva solo el valor mas alto de estos 4 que tomamos arrojándonos una nueva matriz pero ahora de la mitad del tamaño original y de esta forma terminamos nuestra primera convolución la cual es capaz de detectar características primitivas, la segunda convolución identificara características mas finas y así sucesivamente con las demás convoluciones. Una vez que se han finalizado las convoluciones ha la ultima matriz obtenido de estas le aplicaremos la función softmax que conecta a esta ultima matriz con la capa de salida final la cual tendrá la cantidad de neuronas correspondientes a las clases que estamos identificando, los resultados que arroja la red están en el formato **one hot encoding** de unos y ceros por lo que la función **softmax** se encarga de pasar esos 0 y 1 en probabilidades

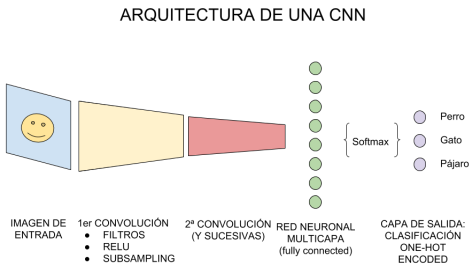

FIGURA 2: Diagrama de funcionamiento de una CNN.

1.1. Datos

El conjunto de datos de lenguaje de señas del alfabeto inglés consta de 87000 imágenes las cuales están divididas en 29 carpetas,26 carpetas pertenecen a cada una de las letras del alfabeto desde la A hasta la Z excluyendo la letra Ñ ya que esta no es propia del lenguaje ingles, la carpeta correspondiente a cada letra contiene 3000 fotografías de dicha letra en diferentes posiciones y con diferente exposiciones de luz con el fin de darla mayor variabilidad ente cada foto. Las 3 carpetas restantes corresponden a la seña de .ªspacio", "delz a imágenes vacías es decir donde no se percibe ninguna mano etiquetada como "nothing.ªl igual que las carpetas correspondientes a las letras estas igual tienen 3000 elementos cada una. Las dimensiones de todas las imágenes son de 200×200 se encuentran a color en el espectro RGB por lo que las dimensiones completas de las imágenes son 200×200, un ejemplo de las imágenes de este conjunto de datos se muestra en la Figura 1.

2. Desarrollo

2.1. Procesamiento de los datos

Lo primero que se hizo fue definir el tamaño de las imágenes originalmente las imágenes eran de un tamaño de 200x200 al ser de color obedecían a las dimensiones 200x200x3 sin embargo al ser un gran numero de imágenes es imposible trabajar con las dimensiones originales por lo que se redujo el tamaño a un tamaño de 64x64x3, después se procedió a la lectura de las imágenes y a su recorte para que obedecieran a las medidas de 64x64x3. Se decidió trabajar con el tamaño de imagen 64x64 ya que fue la dimensión que mejores resultados arrojo, por ejemplo 28x28 dejaba ir muchas características haciendo que los resultados a la hora de la predicción no fueran los mejores. También se definió el numero de clases que son 29 uno correspondiente a cada letra al espacio a la nada y a borrar. Posteriormente se normalizaron los datos en un rango 0 a 1 y posteriormente se definieron los lotes en tamaño de 32

FIGURA 3: Diferencia entre imagen original e imagen recortada.
1) Imagen recortada 64x64.
b) Imagen original 200x200.

Como se puede apreciar no existe una diferencia muy notable tanto a simple vista como a nivel de matriz

2.2. Modelado

El modelo fue una red convolucional apoyándonos en tensorflow, se usaron 5 capas, un batch de 0.3, 50 épocas y el muestreo se lleva acabo con la técnica de MaxPooly se establece una venta de (2,2), las cuales fueron el resultado de múltiples pruebas y resultaron estas los mejores . El modelo recibe 3 matrices de 64 números normalizados entre 0 y 1 por cada imagen. Se utilizo una función de activación **Relu**.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 60, 60, 32)        2432
_________________________________________________________________
activation (Activation)      (None, 60, 60, 32)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 30, 30, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 64)        18496
_________________________________________________________________
activation_1 (Activation)    (None, 28, 28, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 64)        36928
_________________________________________________________________
activation_2 (Activation)    (None, 12, 12, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
dropout (Dropout)            (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 4, 4, 64)          36928
_________________________________________________________________
activation_3 (Activation)    (None, 4, 4, 64)          0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 128)               32896
_________________________________________________________________
dense_1 (Dense)              (None, 29)                3741
=================================================================
Total params: 131,421
Trainable params: 131,421
Non-trainable params: 0
```

FIGURA 4: Características de la red empleada.

Cabe destacar que la red aunque fue programada con 50 epocas esta se detenía entre la época 14 y cuando las imágenes eran pocas pero cuando se trabajo con las 87000 la red se detuvo en la epoca 6, la red se detuvo antes de finalizar las 50 epocas ya que fue programada para que cuando el modelo ya no mejorar la red se detuviera.

3. Resultados y análisis

Los resultados obtenidos fueron bastante bueno como se puede apreciar en las gráficas y la tabla a continuación el desempeño de la red fue fantástico.

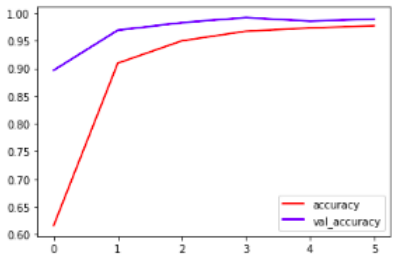

FIGURA 5: Gráficas de perdida.

4. Resultados y análisis

Como se puede apreciar en las gráficas de perdida a medida que las épocas en la red van avanzando la perdida y el error

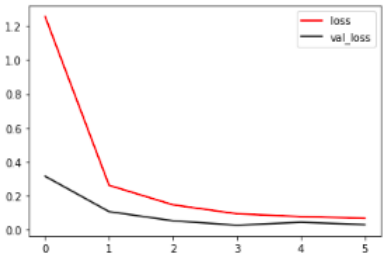

FIGURA 6: Gráficas de perdida.

es menor. Podemos destacar la efectividad de la red para obtener características ya que durante las pruebas en busca de los mejores parámetros nunca fue necesario el implementar mas de 5 capas para poder extraer las características suficientes para que la red pudiera clasificar siempre por arriba del 0.5 aun cuando se le pasaron solo 100 imágenes.

Finalmente con 5 capas y con 87000 imágenes se logro un resultado de **0.99**

5. Conclusiones

La red neuronal resulto ser bastante buena a la hora de identificar las letras del alfabeto ingles de señas por lo que puede ser una base para desarrollar proyectos a futuro, se debe mencionar que el conjunto de datos es un conjunto bastante bueno ya que todas las imágenes son de la misma mano y tomadas en el mismo lugar por lo que la identificación de los objetos que no son la mano fue bastante fácil para la red dado que el entorno nunca vario. Se pudo comprobar la efectividad de el aprendizaje profundo en esa tarea especifica por lo que sin duda alguna vale la pena seguir explorando el potencial de el aprendizaje profundo en el problema de lenguaje de señas.

Como trabajos a futuro se sugiere probar con otros conjuntos de datos para de esta forma la red pueda hacerse mas robusta y pueda presentar buenos resultados con cualquier imagen que se le presente, una vez que la red sea bastante robusta se podría empezar a experimentar ya no solo con el alfabeto si no con frases completas.

6. Bibliografia

http://cs231n.stanford.edu/reports/2016/pdfs/214_Report.pdf
https://www.temjournal.com/documents/vol3no4/journals/1/articles/vol3no4/SignLanguageRecognitionusingNeuralNetworks.pdf

https://link.springer.com/article/10.1007/s42979-021-00612-w

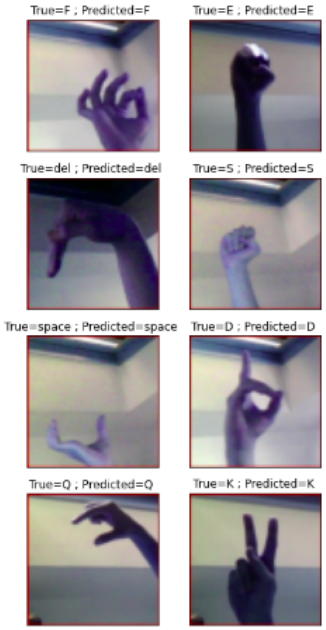

FIGURA 7: Ejemplos de los resultados arrojados por la red.

FIGURA 8: Otros resultados de los ejemplos arrojados por la red.