Implementación de una red neuronal recurrente para generar descripción y clasificación de arañas basado en respuestas de una cuenta de Twitter.

José Alejandro López Quel
21001127
Universidad Galileo
Statistical Learning II

Abstract

En este paper, se presenta un caso de generación de texto el cual recibe como parámetro una imagen e indica la especie y si la araña posee importancia médica basada en las respuestas de un aracnólogo en su cuenta de Twitter. Se emplea un modelo de image caption implementando una red neuronal recurrente. Por lo que se analiza el texto de cada una de las interacciones de la cuenta y se extrae para entrenar el modelo para generar la clasificación.

1 Introducción

Al principio se consideraba imposible que un ordenador pudiera describir una imagen. Con el avance de las técnicas de aprendizaje profundo y los grandes volúmenes de datos disponibles, ahora es posible construir modelos que pueden generar leyendas que describan una imagen.

El aprendizaje profundo es uno de los campos de estudio que más rápido avanza e investiga y que se está abriendo camino en toda la vida cotidiana. Es un subcampo del aprendizaje automático que se ocupa de los algoritmos y se inspira en la estructura y el funcionamiento del cerebro. El avance del aprendizaje profundo radica en la disponibilidad de datos, ya que actualmente estos se generan en grandes cantidades, siendo las redes sociales una de las fuentes de generación de datos más grande.

A partir de ello es posible encontrar distintos tipos de información compartida por expertos en distintos temas que pueden llegar a ser útiles para las personas. Siendo este el foco de este problema se tiene la existencia de una cuenta de Twitter llamada @Arachno_cosas la cual es administrada por un aracnólogo experto en clasificación de arañas. Por lo que él en su cuenta proporciona información sobre las arañas que muchos de los usuarios de la red social comparten. Por ello se decide demostrar el impacto de la implementación de redes neuronales recurrentes y así generar un modelo de image captioning para reproducir lo que realiza el aracnólogo siendo esto describir las arañas que se comparten con su cuenta de Twitter.

Para este caso se dispone del set de imágenes almacenados en un directorio y un archivo generado a partir de la extracción correspondiente de la plataforma de Twitter, en donde se almacena el nombre de la imagen y el texto del tweet asociado. A continuación, se detalla el procesamiento de dicha información para realizar la clasificación y así realizar la predicción del texto que debe acompañar a las imágenes que se utilizan.

2 Obtención de Datos

Para obtener los datos se emplea la librería de python llamada *Tweepy* la cual emplea el API de Twitter para leer los datos de la plataforma, en este caso se analiza la cuenta de @Arachno_Cosas, `https://twitter.com/Arachno_Cosas`. Se lee las respuestas con mención que realiza el autor de la cuenta que contienen una imagen, por lo que se extrae el texto de la respuesta brindada y la imagen de la interacción.

Figure 1: Ejemplo de texto a extraer de la cuenta Twitter

Una vez capturada la información con ayuda de la *Tweepy* se almacenan los datos en un archivo csv el cual contiene dos columnas, una con el texto del autor de la página y la otra con el nombre de la imagen correspondiente.

3 Pre-procesamiento de Datos

Para este caso, se normaliza el texto de los tweets de las personas que interactúan con la cuenta, por lo que se elimina cualquier tipo de caracteres que no corresponden a caracteres alfanuméricos, tales como signos de admiración, caracteres especiales, saltos de línea, o bien, links que pueda llegar a tener el texto del tweet extraído.

0	¡Hola, \n@MaXx_imiliano\n! Gracias por compart...	1.jfif
1	¡Hola, \n@catalina_parr\n! Gracias por tu cons...	2.jfif
2	¡Hola, \n@maxxeff4\n! Gracias por compartir. P...	3.jfif
3	¡Hola, \n@JavierG650ER\n! Gracias por tu consu...	4.jfif
4	¡Hola, \n@headcrusher666\n! Gracias por tu con...	5.jfif

Figure 2: Dataset antes de procesamiento

0	Hola MaXximiliano Gracias por compartir. Perte...	1.jfif
1	Hola catalinaparr Gracias por tu consulta. Se ...	2.jfif
2	Hola maxxeff4 Gracias por compartir. Pertenece...	3.jfif
3	Hola JavierG650ER Gracias por tu consulta. Per...	4.jfif
4	Hola headcrusher666 Gracias por tu consulta. S...	5.jfif

Figure 3: Dataset después de procesamiento

Posterior a ello se define una función para generar un diccionario, el cual contiene como llave el nombre de la imagen y como items cada uno de los textos extraídos de los tweets. En este caso se observa que cada uno de ellos posee un patrón como se muestra a continuación.

```
'Hola catalinaparr Gracias por tu consulta. Se trata de una integrante de la familia Araneidae del género Neoscona probablemente. No son consideradas de importancia médica NIMWhite heavy check mark Saludos'
```

Figure 4: Texto procesado

Cada tweet se compone de 4 partes,

- Saludo inicial, el cual siempre lleva el saludo y la mención al usuario que realizó la consulta y un agradecimiento por la misma.
- Descripción de la familia a la que pertenece la araña.
- Oración indicadora de importancia médica.
- Mensaje de despedida.

Por lo que en el pre-procesamiento del texto se ignora la primera y la ultima oración, siendo estas el saludo inicial y el mensaje de despedida. Y se enfoca el entrenamiento en las partes más importantes que son la descripción de la araña y si esta es de importancia médica o no. Dentro de la generación del diccionario con las descripciones de la imagen, se agrega a estas descripciones etiquetas de inicio y final de la siguiente manera:

```
['<start> Pertenece a la familia Araneidae del género Neoscona probablemente <end>',
 '<start> Se les conoce como arañas de jardín o de tela orbicular <end>',
 '<start> No son consideradas de importancia médica NIMWhite heavy check mark <end>',
 '<start> Se trata de una integrante de la familia Araneidae del género Neoscona probablemente <end>',
 '<start> No son consideradas de importancia médica NIMWhite heavy check mark Saludos <end>',
 '<start> Pertenece a la familia Araneidae del género Neoscona probablemente <end>',
 '<start> No son consideradas de importancia médica NIMWhite heavy check mark Saludos <end>',
 '<start> Pertenece a la familia Theridiidae del género Steatoda probablemente <end>',
 '<start> No son consideradas de importancia médica NIMWhite heavy check mark Saludos <end>',
 '<start> Se trata de una integrante de la familia Araneidae <end>']
```

Figure 5: Ejemplo de descripciones de las imágenes con etiquetas.

Luego de esto se utiliza la capa TextVectorization de Keras para vectorizar los datos de texto, es decir, para convertir las cadenas originales en secuencias de números enteros donde cada número entero representa el índice de una palabra en un vocabulario. Se utiliza un esquema de normalización de cadenas personalizado en este caso: quitar los caracteres de puntuación excepto "<" y ">", y el esquema de división por defecto (dividir en espacios en blanco).

Se decide que se generará pares de imágenes y sus correspondientes descripciones utilizando un objeto *tf.data.Dataset*. El proceso consta de dos pasos:

- Leer la imagen del disco
- Tokenizar las descripciones correspondientes a la imagen

4 Metodología y experimentación

Para esta investigación se utiliza la siguiente metodología la cual consta de 3 modelos:

- CNN: utilizada para extraer las características de la imagen.
- TransformerEncoder: Las características de la imagen extraída se pasan a un codificador basado en Transformer que genera una nueva representación de las entradas
- TransformerDecoder: Este modelo toma la salida del codificador y los datos de texto (secuencias) como entradas e intenta aprender a generar la descripción de la imagen.

Para la CNN se emplea un modelo pre-entrenado denominado EfficientNet-B0. Se tiene que esta arquitectura no fue desarrollada por los mismos ingenieros que EfficientNet, sino por la propia red neuronal. Desarrollaron este modelo utilizando una búsqueda de arquitectura neuronal multiobjetivo que optimiza tanto la precisión como las operaciones en coma flotante. Tomando B0 como modelo de referencia, los autores desarrollaron una familia completa de EfficientNets, desde B1 hasta B7, que alcanzó una precisión de vanguardia en ImageNet, siendo al mismo tiempo muy eficiente frente a sus competidores. Se tiene que este modelo esta compuesto de la siguiente forma:

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 6: Modelo EfficientNet-B0

Se puede ver que la arquitectura utiliza 7 bloques residuales invertidos, pero cada uno tiene una configuración diferente. Estos bloques también utilizan el bloque de compresión y excitación junto con la activación de la oscilación.

4.1 Experimento 1

Para el experimento 1 se utiliza un optimizador ADAM con el learning rate por defecto y se emplea una función de perdida Sparse Categorical Crossentropy. Por lo que se obtienen que el modelo obtiene un accuracy de 0.7003 para el conjunto de datos de validación.

```
Epoch 1/100
2/2 [==============================] - 82s 40s/step - loss: 30.7660 - acc: 0.1979 - val_loss: 16.5536 - val_acc: 0.574
Epoch 2/100
2/2 [==============================] - 63s 35s/step - loss: 11.0566 - acc: 0.6244 - val_loss: 11.2285 - val_acc: 0.648
Epoch 3/100
2/2 [==============================] - 63s 32s/step - loss: 7.0637 - acc: 0.7034 - val_loss: 10.0099 - val_acc: 0.6752
Epoch 4/100
2/2 [==============================] - 63s 35s/step - loss: 5.2750 - acc: 0.7530 - val_loss: 9.7271 - val_acc: 0.6826
Epoch 5/100
2/2 [==============================] - 63s 35s/step - loss: 4.1168 - acc: 0.7935 - val_loss: 9.7594 - val_acc: 0.6879
Epoch 6/100
2/2 [==============================] - 63s 35s/step - loss: 3.3450 - acc: 0.8169 - val_loss: 9.6652 - val_acc: 0.6996
Epoch 7/100
2/2 [==============================] - 63s 32s/step - loss: 2.8594 - acc: 0.8461 - val_loss: 9.7254 - val_acc: 0.7003
```

Figure 7: Resultados entrenamiento Experimento 1

4.2 Experimento 2

Para el experimento 2 se utiliza un optimizador ADAM con un learning rate de 0.01 y se emplea una función de perdida Sparse Categorical Crossentropy. Se comprueba que se obtiene resultados parecidos al Experimento 1. Con un accuracy de 0.7035 para el conjunto de datos de validación.

```
Epoch 1/100
2/2 [==============================] - 79s 35s/step - loss: 3.7328 - acc: 0.8096 - val_loss: 9.4909 - val_acc: 0.6848
Epoch 2/100
2/2 [==============================] - 62s 34s/step - loss: 2.7758 - acc: 0.8511 - val_loss: 9.7286 - val_acc: 0.7049
Epoch 3/100
2/2 [==============================] - 63s 35s/step - loss: 2.0499 - acc: 0.8867 - val_loss: 10.0567 - val_acc: 0.7035
```

Figure 8: Resultados entrenamiento Experimento 2

5 Discusión de resultados

Se obtiene que los dos experimentos realizados con el modelo planteado obtienen los mismos resultados similares de accuracy. Pero al observar la gráfica de la evolución del entrenamiento del modelo, se puede observar que el modelo del Experimento 1, se desempeña mejor que el del Experimento 2.

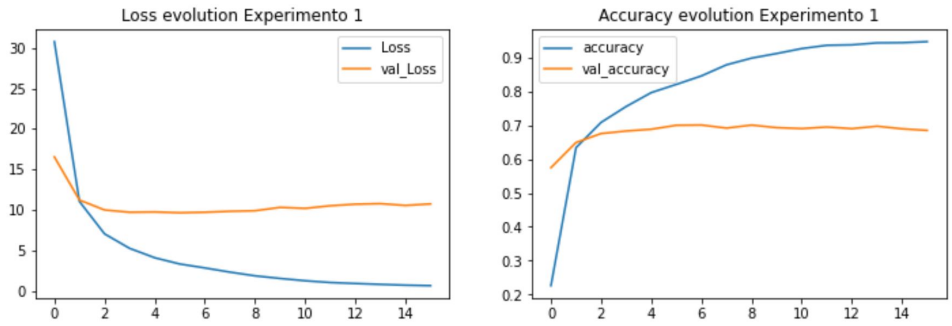


Figure 9: Resultados entrenamiento Experimento 1

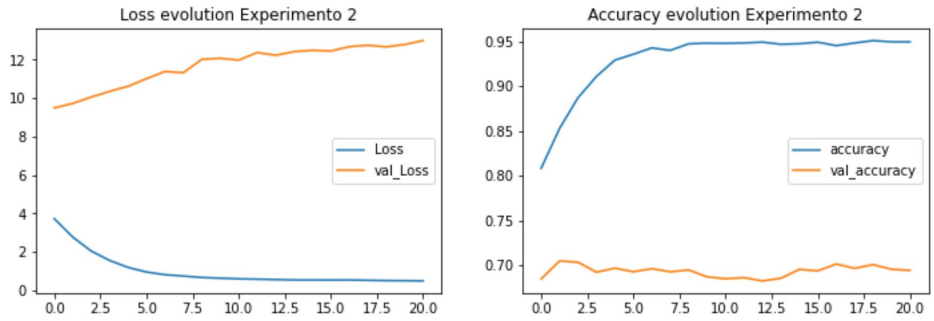
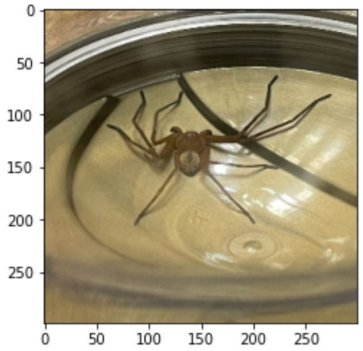


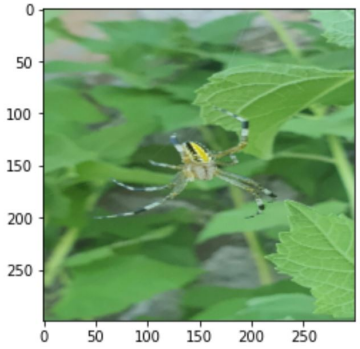
Figure 10: Resultados entrenamiento Experimento 2

Por lo que se selecciona como el mejor modelo al obtenido en el Experimento 1. Por último se genera una función para mostrar los resultados del texto que se predice para imágenes del set de evaluación.



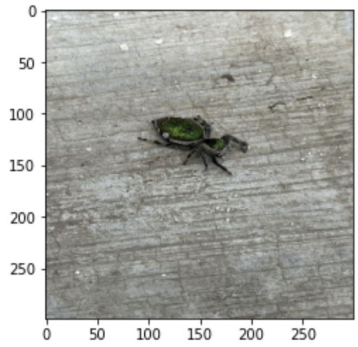
```
PREDICTED CAPTION: no son consideradas de importancia médica nimwhite heavy check mark saludos
```

Figure 11: Resultados predicción de texto ejemplo 1



```
PREDICTED CAPTION: no son consideradas de importancia médica nimwhite heavy check mark saludos
```

Figure 12: Resultados predicción de texto ejemplo 2



PREDICTED CAPTION: no son consideradas de importancia médica nimwhite heavy check mark saludos

Figure 13: Resultados predicción de texto ejemplo 3

6 Trabajo futuro

Se plantea generar un bot el cual analice la cuenta de Twitter @Arachno_cosas y genere los textos de las menciones que pueda llegar a tener. Al implementar el bot se debe tener en cuenta las 2 oraciones eliminadas en el procesamiento las cuales son el saludo inicial y el mensaje de despedida.

Referencias

[1] Roy, A. (2020, December 11). A Guide to Image Captioning - Towards Data Science. Medium. https://towardsdatascience.com/a-guide-to-image-captioning-e9fd5517f350

[2] Team, K. (2021). Keras documentation: Image Captioning. Keras. https://keras.io/examples/vision/image_captioning/

[3] Borad, A. (2021, July 10). Image Classification with EfficientNet: Better performance with computational efficiency. Medium. https://datamonje.medium.com/image-classification-with-efficientnet-better-performance-with-computational-efficiency-f480fdb00ac6