Traducción de Texto por Redes Transformer

Juan Marcelo Frutos Navarro y Santiago Dionisio Vargas García
Inteligencia Artificial
Facultad de Ingeniería, Universidad Nacional de Asunción

***Resumen*—El trabajo se enfoca en un área de Machine Learning llamado Procesamiento Natural del Lenguaje (NPL), específicamente en el área de traducción, en este caso del ingles al español**

I. Introducción

Transformer es un modelo de aprendizaje profundo que adopta mecanismos de atención,su uso primario es para el área de natural language processing (NLP).El mecanismo de atención mira una secuencia de entrada y decide en cada paso qué otras partes de la secuencia son importantes.

II. Antecesores

Los modelos antecesores son Bag of Words, solo considera si una palabra conocida aparece en un documento o no, no le importa el significado, el contexto y el orden en que aparecen. Debido a esto se implementó el uso de la redes RNN, que luego fueron mejoradas a las LSTM.

Los RNN procesan de forma secuencial, manteniendo un vector de estado que contiene una representación de los datos que se ven después de cada token. Para procesar el n-ésimo token, el modelo combina el estado que representa la oración hasta el token n-1con la información del nuevo token para crear un nuevo estado, que representa la oración hasta el token n. Teóricamente, la información de un token puede propagarse arbitrariamente hacia abajo en la secuencia, si en cada punto el estado continúa codificando información contextual sobre el token. En la práctica, este mecanismo es defectuoso: el problema del gradiente deja el estado del modelo al final de una oración larga sin información precisa y extraíble sobre los tokens anteriores.

Este problema fue abordado por mecanismos de atención. Los mecanismos de atención permiten que un modelo se extraiga del estado en cualquier punto anterior a lo largo de la secuencia. La capa de atención puede acceder a todos los estados anteriores y los pesa de acuerdo con una medida de relevancia aprendida, proporcionando información relevante sobre los tokens lejanos.

El artículo “Attention is all u need” describe los transformadores y lo que se denomina arquitectura secuencia a secuencia. Secuencia a secuencia (o Seq2Seq) es una red neuronal que transforma una secuencia determinada de elementos, como la secuencia de palabras en una oración, en otra secuencia. Los modelos Seq2Seq son particularmente buenos en la traducción, donde la secuencia de palabras de un idioma se transforma en una secuencia de palabras diferentes en otro idioma.

Los modelos Seq2Seq constan de un codificador y un decodificador. El codificador toma la secuencia de entrada y la mapea en un espacio dimensional superior (vector n-dimensional). Ese vector abstracto alimenta al decodificador que lo convierte en una secuencia de salida. La secuencia de salida puede estar en otro idioma, símbolos, una copia de la entrada, etc.

III. Modelo del sistema

La red neuronal del transformador recibe una oración de entrada y la convierte en dos secuencias: una secuencia de incrustaciones de vectores de palabras y una secuencia de codificaciones posicionales.

Las incrustaciones de vectores de palabras son una representación numérica del texto. Es necesario convertir las palabras a la representación incrustada para que una red neuronal pueda procesarlas. En la representación de incrustación, cada palabra del diccionario se representa como un vector. Las codificaciones posicionales son una representación vectorial de la posición de la palabra en la oración original.

El transformador suma las incrustaciones vectoriales de palabras y las codificaciones posicionales juntas y pasa el resultado a través de una serie de codificadores, seguidos de una serie de decodificadores,a diferencia de los RNN y LSTM, toda la entrada se alimenta a la red de forma simultánea en lugar de secuencial.

Cada uno de los codificadores convierte su entrada en otra secuencia de vectores denominados “encodings”. Los decodificadores hacen lo contrario: vuelven a convertir las codificaciones en una secuencia de probabilidades de diferentes palabras de salida. Las probabilidades de salida se pueden convertir en otra oración en lenguaje natural usando la función softmax.

Cada codificador y decodificador contiene un componente llamado mecanismo de atención, que permite el procesamiento de una palabra de entrada para incluir datos relevantes de otras palabras determinadas, mientras enmascara las palabras que no contienen información relevante.

Debido a que esto debe calcularse muchas veces, implementamos múltiples mecanismos de atención en paralelo, aprovechando la computación en paralelo que ofrecen las GPU. A esto se le llama mecanismo de atención de múltiples cabezales. La capacidad de pasar varias palabras a través de una red neuronal simultáneamente es una ventaja de los transformadores sobre LSTM y RNN.

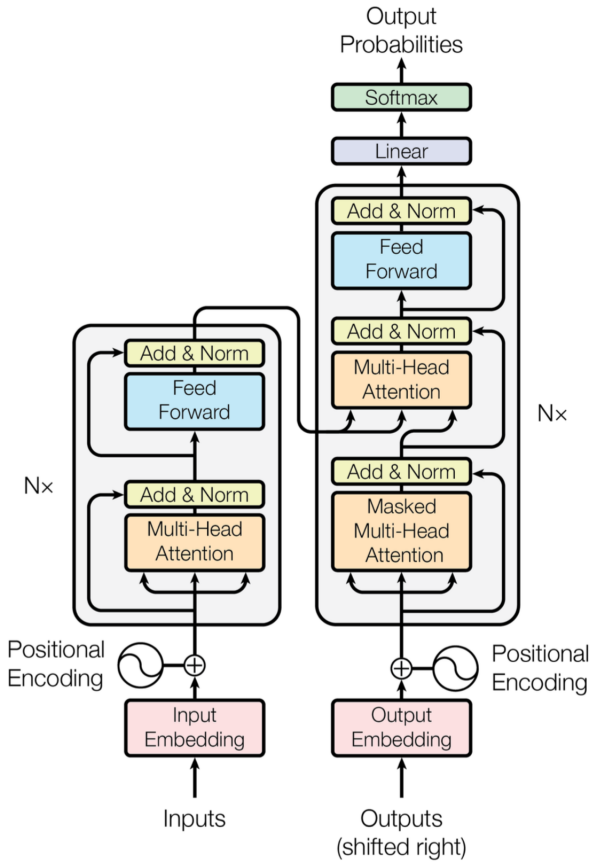

Figure 1: The Transformer - model architecture.

Figura 1. Arquitectura del Transformer

El codificador está a la izquierda y el decodificador a la derecha. Tanto el codificador como el decodificador se componen de módulos que se pueden apilar uno encima del otro varias veces, como lo describe Nx en la figura. Vemos que los módulos consisten principalmente en capas de Atención de Cabecera Múltiple y Feed Forward. Las entradas y salidas (oraciones de destino) se incrustan primero en un espacio n-dimensional ya que no podemos usar cadenas directamente. Una parte leve pero importante del modelo es la codificación posicional de las diferentes palabras. Dado que no tenemos redes recurrentes que puedan recordar cómo se introducen las secuencias en un modelo, necesitamos de alguna manera darle a cada palabra / parte de nuestra secuencia una posición relativa, ya que una secuencia depende del orden de sus elementos. Estas posiciones se agregan a la representación incrustada (vector n-dimensional) de cada palabra.

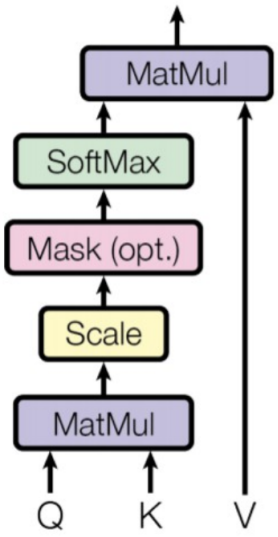

Figura 2. Scaled Dot Product Attention

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

La figura 2 es una descripción del mecanismo de atención implementado, que se puede entender mejor por la ecuación (1) donde Q es una matriz que contiene la representación vectorial de una palabra en la secuencia, K son todas las representaciones vectoriales de todas las palabras en la secuencia y V son los valores, que nuevamente son las representaciones vectoriales de todas las palabras en la secuencia. Para el codificador y decodificador, módulos de atención de múltiples cabezales, V consta de la misma secuencia de palabras que Q.Sin embargo, para el módulo de atención que tiene en cuenta las secuencias del codificador y del decodificador, V es diferente de la secuencia representada por Q.

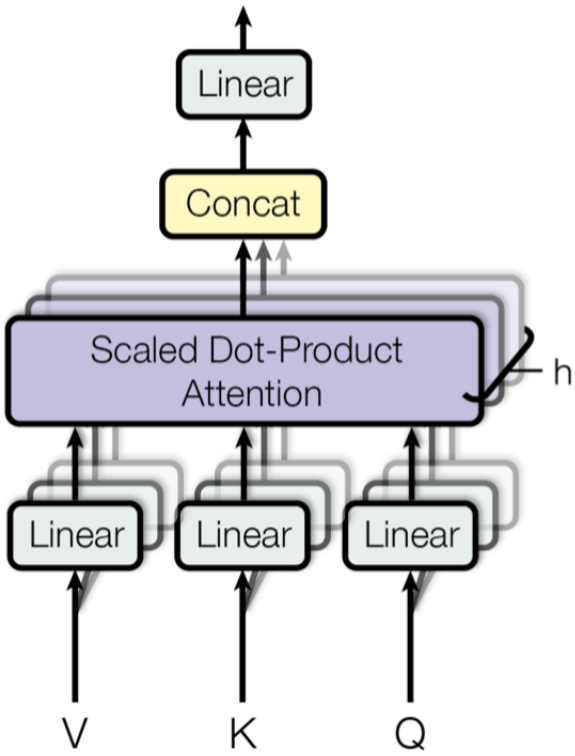

Figura 3. Multihead Attention

La figura 4 describe cómo este mecanismo de atención se puede paralelizar en múltiples mecanismos que se pueden usar uno al lado del otro. El mecanismo de atención se repite varias veces con proyecciones lineales de Q, K y V. Esto permite que el sistema aprenda de diferentes representaciones de Q, K y V, lo que es beneficioso para el modelo. Estas representaciones lineales se realizan multiplicando Q, K y V por las matrices de peso W que se aprenden durante el entrenamiento.

IV. Resultados

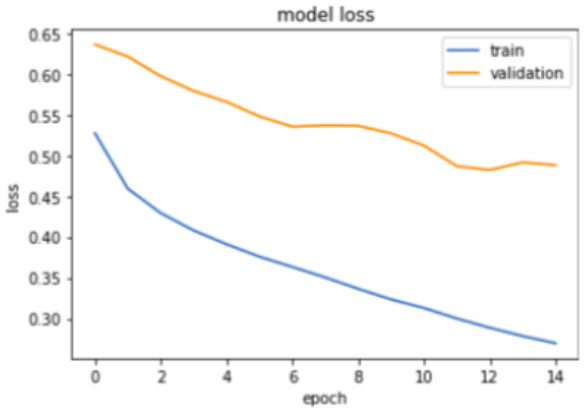

Figura 4. 2 heads, 50 batch, False Shuffle, Underfitting.

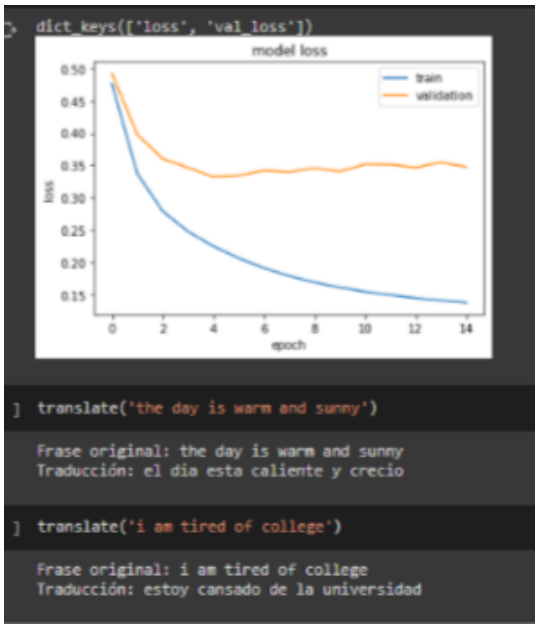

Figura 5. 4 heads, 50 batch, True Shuffle, Overfitting

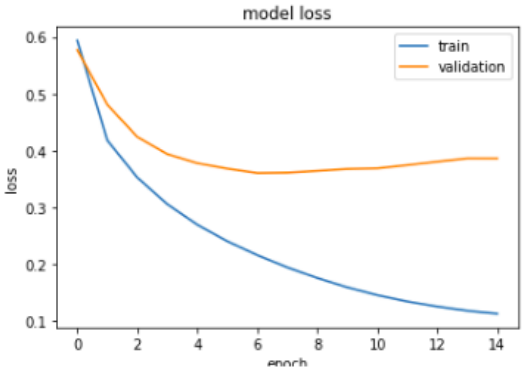

Figura 6. 2 heads, 256 batch, True Shuffle, Overfitting

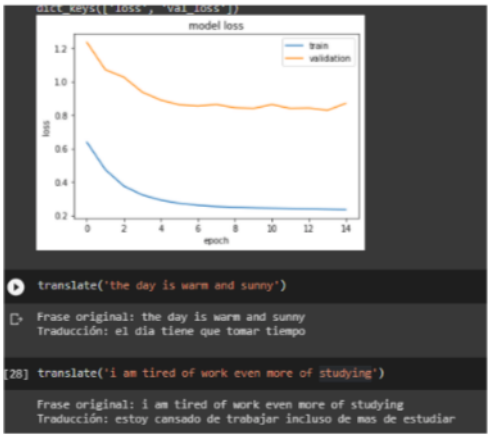

Figura 7. 2 heads, 256 batch, True Shuffle, Overfitting

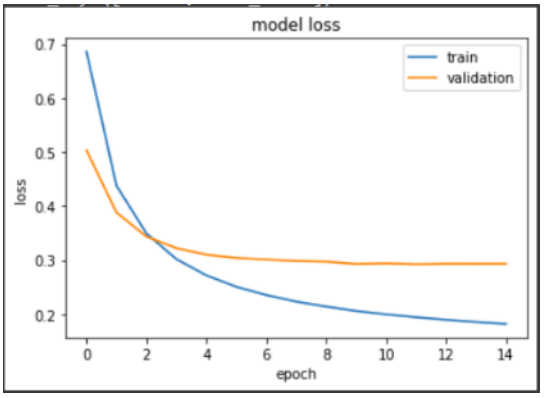

Figura 8. 2 heads, 256 batch, True Shuffle, Mejor resultado, primer dataset con la mitad de frase que las otras figuras

REFERENCIAS

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin, “Attention Is All You Need,” , 2017.