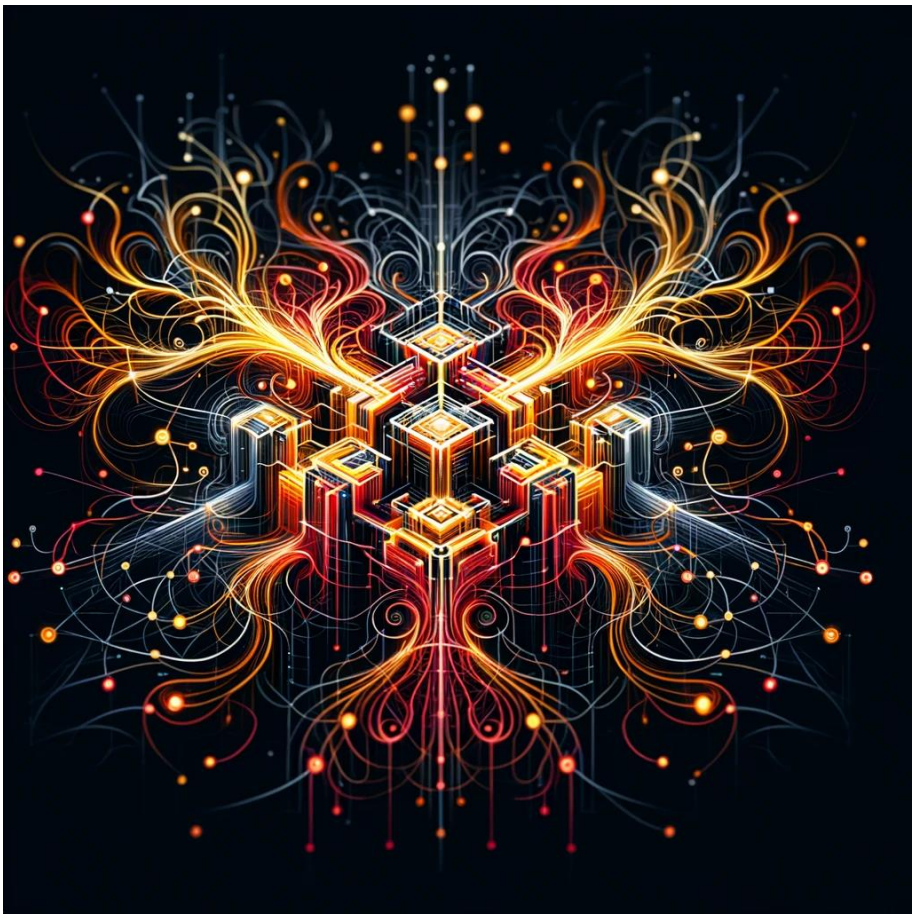
INTRODUCCIÓN A LAS REDES NEURONALES *TRANSFORMERS*



"Illustration depicting a transformer neural network" (DALL-E)

José Ángel Martínez-Huertas (jamartinez@psi.uned.es)

Modelos de redes neuronales

Máster en Metodología de las Ciencias del Comportamiento y de la Salud

Universidad Nacional de Educación a Distancia

ÍNDICE

Este documento pretende ser una introducción conceptual (que no analítica) a las redes neuronales llamadas *Transformers* en el contexto del Máster en Metodología de las Ciencias del Comportamiento y de la Salud en la Universidad Nacional de Educación a Distancia. Así, este documento asume que la/el estudiante que se enfrente a los *Transformers* tiene una base conceptual y analítica suficiente como para entender algunos conceptos fundamentales de las redes neuronales clásicas como "retropropagación", "nodos" o "capas ocultas". En cualquier caso, los contenidos presentados en este documento pretenden explicar conceptualmente todos los pasos que comúnmente se utilizan para estimar estos modelos.

1. ¿Qué es un *Transformer*?

Básicamente, un *Transformer* es una arquitectura de red neuronal que se encarga de procesar secuencias de datos de manera eficiente. Los datos que puede procesar esta arquitectura de red neuronal van desde texto hasta imágenes. Además, se dice que se hace un procesamiento eficiente de los datos, ya que son mucho más rápidos que otros modelos encargados de procesar secuencias de información porque trabajan en paralelo y porque contiene un componente nuevo que requerirá nuestra atención en las siguientes páginas: los mecanismos de atención.

Esta arquitectura de red neuronal fue propuesta por Vaswani et al. (2017) en un artículo titulado "*Attention is All You Need*":

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems 30 (NIPS 2017).

El título del artículo es toda una declaración de intenciones y, como podréis imaginar, nos desvela la importancia de los mecanismos de atención que veremos más adelante. Hasta la aparición de esta arquitectura de redes neuronales, los modelos de redes neuronales recurrentes y los modelos de redes neuronales convolucionales eran la solución óptima cuando éstos incluían codificadores y decodificadores. Sin embargo, en su versión clásica, la red neuronal *Transformer* prescinde por completo de la recurrencia típica de las redes neuronales recurrentes y de las operaciones de convolución propias de las redes neuronales convolucionales. Esta arquitectura basa todo su funcionamiento en los llamados mecanismos de atención.

2. Componentes de la arquitectura

De manera muy resumida, podemos afirmar que los *Transformers* tienen cinco componentes principales. A continuación, vamos a verlos de manera muy resumida y por separado para, después, explicar la arquitectura general del modelo de redes neuronales.

1. Los mecanismos de atención, también conocidos como atención auto-dirigida (*self-attention mechanisms*). Básicamente, esta parte del algoritmo permite que el modelo se centre en distintas partes de la entrada para poder llevar a cabo su aprendizaje y sus predicciones.

2. Las capas de codificación y decodificación, que tienen la misma estructura que la de los autocodificadores o *autoencoders*. En general, los *Transformers* suelen tener una estructura de codificador-decodificador, donde el codificador procesa la información de la entrada y el decodificador genera la salida a partir del resumen de la información de la entrada generado por las capas intermedias. Cada uno de estos componentes, es decir, el codificador y el decodificador, están compuestos a su vez por múltiples capas que tienen mecanismos de atención y redes neuronales normales (denominadas también en algunos contextos como prealimentadas o *feed-forward neural networks*, en contraposición con otras arquitecturas más complejas).
3. Las conexiones residuales y la normalización de la capa, que son conceptos que veremos con más detalle en el siguiente apartado, pretenden evitar que se pierda el gradiente[1]. Para ello, las distintas capas del codificador y del decodificador tienen una conexión residual a su alrededor y un proceso de normalización. Las conexiones residuales permiten que el output de una subcapa sea la suma de su input y su output, lo que facilita el flujo de información a través de la red.
4. Aunque no es un componente como tal, podemos considerar que el procesamiento en paralelo es clave para entender las ventajas del uso de los Transformers, ya que estos pueden procesar todos los elementos de la información de la entrada a la vez, siendo más rápidos y eficientes. Esto

[1] En redes neuronales profundas, dado que el cálculo del gradiente supone la multiplicación de distintas participaciones de los nodos en el error global de la red, suele darse el caso en el que el gradiente estimado se pierde o es nulo. Para ello, se han inventado algunas propuestas que pretenden corregirlo.

entra en contraposición a otros modelos como las redes neuronales recurrentes que debían procesar la información de manera secuencial.

5. Los *embeddings* y la codificación posicional (*posicional encoding*) son elementos clave para trabajar con la información de la entrada. Básicamente, como ya hemos visto en otros temas de la asignatura, los *embeddings* son representaciones vectoriales de las palabras y nos permiten representar numéricamente su significado. Esto, obviamente, es importante en modelos que trabajan con texto, pero no en modelos que trabajan con imágenes. Por su lado, la codificación posicional (*posicional encoding*) pretende dar cuenta de la posición relativa de las palabras u otras unidades de información en la secuencia analizada, ya que el propio modelo no puede procesar secuencialmente la información. Este es, básicamente, el input o la entrada que recibe el modelo.

3. Desglosando la arquitectura

La Figura 1 presenta la arquitectura del *Transformer* propuesto por Vaswani et al. (2017). Básicamente, en la arquitectura de codificador-decodificador del modelo, el codificador mapea la secuencia de la entrada y la transforma en una representación más abstracta. Una vez que tiene esta representación más abstracta, el decodificador genera la respuesta de la salida. Hasta aquí, la lógica es muy similar a la de los autocodificadores (*autoencoders*). A su vez, esta arquitectura general también inciorporada los mecanismos de auto-atención de manera aplicada (*stacked self-attention*) y va dato a dato, a través de capas completamente conectadas tanto en el codificador como el decodificador. Toda esta información queda recogida en la

arquitectura presentada en la Figura 1, pero vamos a explicarla con mayor detalle a continuación.

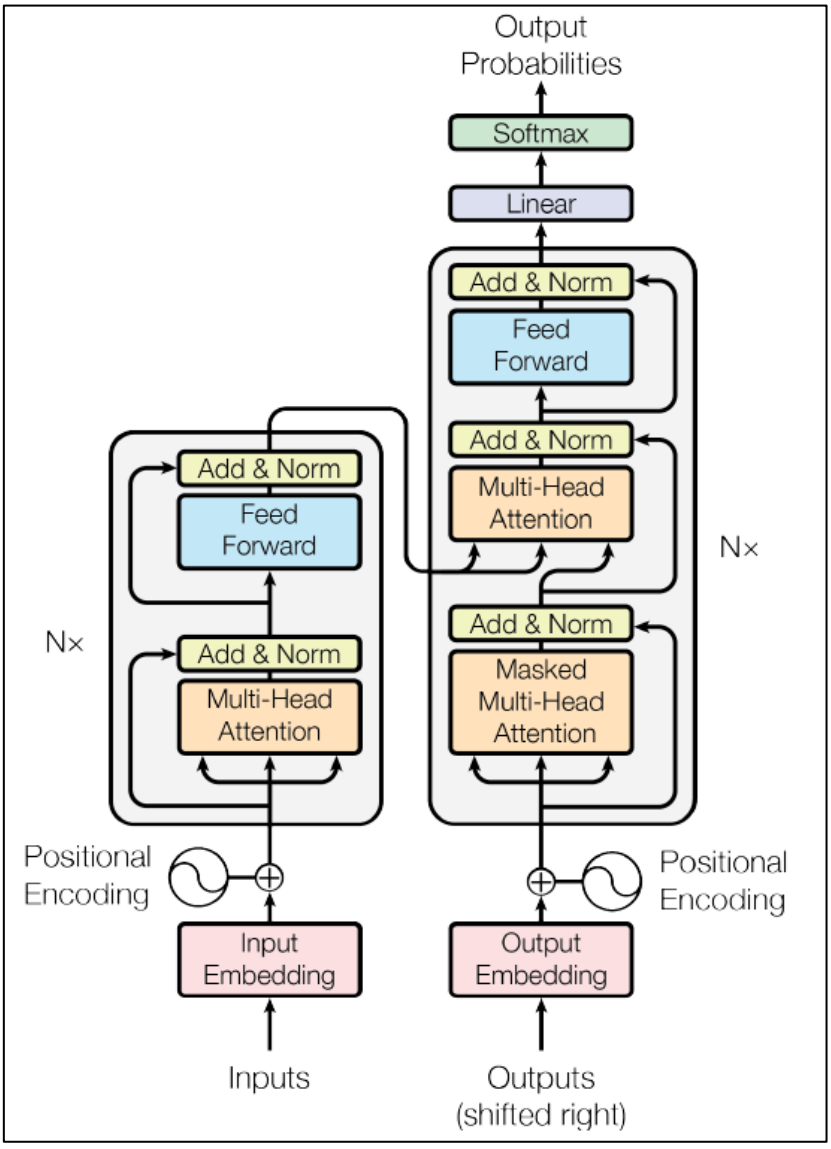


Figura 1. Arquitectura del *Transformer* (tomado de Vaswani et al., 2017).

En el caso de este modelo concreto, el codificador está compuesto por un total de 6 capas apiladas y cada una de las capas tiene dos sub-capas. La primera de las sub-capas es lo que hemos llamado mecanismo de atención (concretamente: *multi-head self-attention mechanisms*). La segunda solo captura la posición de la información en la secuencia. A su vez, se utilizan conexiones residuales alrededor de estas dos sub-capas, y se aplica una normalización. A su vez, el decodificador también está compuesto por 6 capas apiladas pero, aparte de las dos sub-capas ya mencionadas, también incorpora una tercera que aplica mecanismos de atención (*multi-head self-attention mechanisms*) sobre la información de la salida que están modificador para que las predicciones para la posición *i* de la secuencia solo tengan en cuenta los datos presentes desde el inicio de la secuencia hasta (*i-1*). De nuevo, se aplican conexiones residuales alrededor de las sub-capas, y se aplica una normalización.

Los mecanismos de atención, que son la clave para entender esta arquitectura de redes neuronales, se encargan de mapear la información recibida. Básicamente, permite al modelo ponderar la importancia de diferentes partes de la entrada. En el sub-mecanismo de atención aditiva (*additive attention*), podemos decir que cada palabra que se recibe en la entrada del modelo se transforma en tres vectores diferentes: una consulta (Q), una clave (K) y un valor (V). Los mecanismos de atención se aplican calculando un producto escalar entre Q y K, y aplicando una función *softmax* para obtener pesos normalizados que luego se aplican a los valores estimados (V). La expresión formal del output de dicho mecanismo de atención aplicado a un conjunto de consultas (Q que compondrían una matriz **Q**) sería:

$$Atención(Q, K, V) = softmax\left(QK^T \Big/ \sqrt{d_k}\right)V$$

donde $1/\sqrt{d_k}$ sería un factor de escala para normalizar la estimación.

Por otro lado, en el sub-mecanismo de atención *multi-head attention*, se proyectan las distintas consultas (Qs), claves (Ks) y valores (Vs) para poder aplicar los mecanismos de atención en paralelo. En este caso concreto, se utilizan 8 capas paralelas de atención. Para ello, se concatenan los distintos elementos y su expresión formal es la siguiente:

$$MultiHead(Q,K,V) = Concat\big(Atención(QW_i^Q, KW_i^K, VW_i^V)\big)W^O$$

donde W_i^Q, W_i^K, y W_i^V representan los parámetros de las matrices proyectadas.

La Figura 2 presenta gráficamente ambos mecanismos de atención.

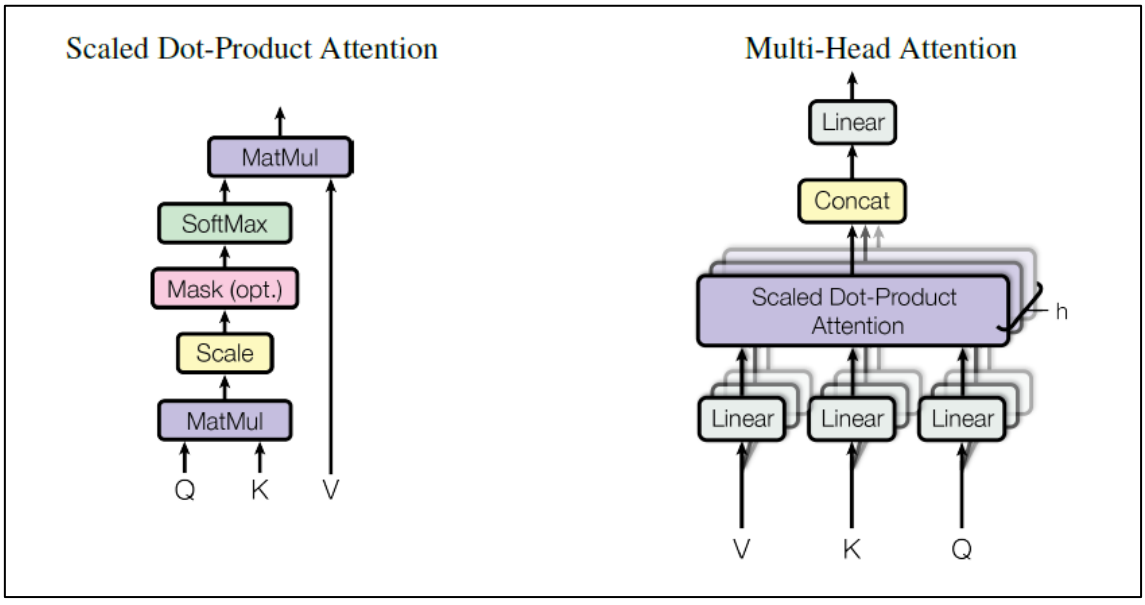


Figura 2. Mecanismos de atención (tomado de Vaswani et al., 2017).

Siguiendo con las explicaciones del modelo *Transformer* de Vaswani et al. (2017), los mecanismos de atención (*multi-head attention*) se aplican de tres formas diferentes:

1. En las capas de codificación y decodificación, las consultas (Q) provienen de la capa anterior del decodificador, mientras que las claves (K) y los valores (V) vienen de la salida del codificador.
2. El codificador tiene capas de auto-atención (*self-attention*) donde todos los valores (Q, K y V) vienen de la anterior capa *k-1* del codificador para que cada una de las posiciones de la nueva capa *k* del codificador pueda atender todas las posiciones de la capa anterior *k-1*.
3. El decodificador también tiene capas de auto-atención (*self-attention*) que permiten que todas las posiciones de la capa *k* atiendan todas las posiciones del decodificador hasta el momento (es decir, todas las posiciones de todas las capas hasta *k-1*).

Aparte, cada una de las capas del codificador y decodificador tienen una red completamente conectada que no deja de ser una doble transformación lineal con una activación ReLU entre ambos para conectar cada una de las capas.

4. BERT: *Bidirectional Encoder Representations from Transformers*

En esta sección vamos a presentar brevemente un modelo real y muy popular de lenguaje que nos permitirá ver la complejidad de las aplicaciones que utilizan este tipo de arquitectura de redes neuronales. El modelo BERT (derivado de su nombre completo

en inglés: *Bidirectional Encoder Representations from Transformers*) fue publicado por Devlin et al. (2018), un grupo de investigadores de Google:

> Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805.

BERT es un modelo de procesamiento de lenguaje natural desarrollado. Básicamente, el éxito del modelo BERT se puede encontrar en su bidireccionalidad, ya que permite que las palabras de una oración se analicen en todas las direcciones, teniendo en cuenta tanto las palabras anteriores como las posteriores (y no en una sola dirección, como hacían otros modelos previos), lo que permite entender el contexto y la intención detrás de cada palabra de manera más efectiva. Otra de las características que han hecho que este modelo sea tan popular es que se trata de un "modelo pre-entrenado" que se aplica y/o adapta para la resolución de tareas más específicas y se puede aplicar, por tanto, a una gran cantidad de tareas dependiendo del objetivo del investigador.

Una de las cosas más curiosas del modelo BERT es que, aunque se basa en la arquitectura de redes neuronales *Transformer*, solo utiliza el componente del codificador. Así, se pueden diferenciar tres partes principales en el modelo BERT: los *embeddings* (que no dejan de ser vectores que representan el significado de las palabras), un conjunto de capas de codificadores tal y como se han presentado para los *Transformers* (no como los de los autocodificadores o *autoencoders*), y una transformación final de la representación vectorial predicha por el codificador en un vector *one-hot* que representa una palabra. Así, cabe destacar también que existen varias versiones de este modelo en inglés cuya complejidad es enorme, ya que el modelo base tiene 12 capas de codificadores *Transformers* y las versiones más completas ascienden hasta las 24 capas de codificadores *Transformers*.