Resumen: An Attentive Survey of Attention Models.

Pablo Fernández Piñeiro

10 de julio de 2023

Índice

1. INTRODUCCIÓN

El (AM) se ha convertido en un concepto importante en la literatura de redes neuronales y es ampliamente utilizado en el campo de la inteligencia artificial. La atención se basa en la forma en que nuestros sistemas biológicos, como el procesamiento visual, se enfocan selectivamente en partes relevantes de la información. En problemas de procesamiento del lenguaje, habla y visión, ciertas partes de la entrada son más importantes que otras, y el modelo de atención permite que el modelo preste atención dinámicamente a esas partes relevantes para realizar la tarea de manera efectiva.

El rápido avance en la modelización de la atención se debe a varias razones: estos modelos son el estado del arte en múltiples tareas en procesamiento del lenguaje natural, visión por computadora, tareas multimodales y sistemas de recomendación. Además, ofrecen ventajas adicionales, como mejorar la interpretabilidad de las redes neuronales, lo cual es importante en aplicaciones que afectan la vida humana. También ayudan a superar desafíos asociados a las redes neuronales recurrentes, como el rendimiento degradado con entradas largas y la ineficiencia computacional en el procesamiento secuencial.

En este trabajo se presenta una encuesta breve pero completa sobre la modelización de la atención. Se explican la intuición detrás de la atención, se describen las funciones de atención propuestas y se presentan arquitecturas neuronales clave que utilizan la atención. También se exploran las aplicaciones en las que la atención ha sido ampliamente aplicada y se discute cómo la atención facilita la interpretabilidad de las redes neuronales. El objetivo es proporcionar una comprensión más amplia del modelo de atención y ayudar a los desarrolladores e ingenieros de inteligencia artificial a determinar el enfoque adecuado para sus aplicaciones específicas.

2. FUNDAMENTOS DE LA ATENCIÓN

La idea de la atención puede ser entendida utilizando un modelo de regresión propuesto por Naradaya-Wilson en 1964. Si nos dan un dataset para entrenar con n instancias, cada una con su valor “x” y su resultado, “y”. Se quiere predecir “y” para cada consulta nueva, “x”.

Un estimador básico predecirá la media de todos los resultados del dataset de entrenamiento. Naradaya-Watson propueso una mejor aproximación en la que el estimador utiliza una media ponderada donde los pesos se corresponden con la relevancia de cada instancia de entrenamiento. PAra la ponderiación, comunmente se utiliza un *normalized Gaussian kernel*, aunque se pueden utilizar otras medidas de similitud con normalización. Los autores mostraron que el estimador tiene:

1. **Consistencia**. Con los suficientes datos de entrenamiento converge a resultados óptimos.
2. **Simplicidad**. No tiene *free parameters*, la información está en los datos y no en los pesos.

3. MODELO DE ATENCIÓN

El primero uso de modelos de atención fue propuesto para una tarea de modelado secuencia a secuencia. Un modelo secuencia a secuencia consiste en una arquitectura codificador-decodficiador. El codificador es una RNN (Recurrent Nueral Network) que toma como entrada una secuencia de tokens, con un número fijo T de tokens por cada entrada, los transforma en en vectores de tamaño fijo también. El decodificador es también una RNN que toma el vector de tamaño fijo como entrada y genera una salida de otro tamaño.

3.1. Desafíos del tradicional codificador-decodificador

Hay dos desafíos muy conocidos en este marco de trabajo de codificador-decodificador. Primero,el codificador tiene que comprimir tod la información en un sólo vector de lonigutd fija que se pasaría al decodificador. Haciendo esto, se pueden producir pérdidas de información. Segundo, es incapaz de modelar la alineación entre secuencias de entrada y salida, que es un aspecto esencial de las tareas de salida estructuradas, como traducción o resumen. Intuitivamente, en tareas secuencia a secuencia, cada token de salida se espera que sea más influencia por algunas partes específicas de la secuencia de entrada. Sin

emnbargo el decodificador no tiene mecanismos para enfocarse selectivamente en los tokens de entrada relevantes mietras genera cada token de salida.

3.2. Idea clave

El modelo de atención (AM) tiene como objetivo mitigar estos desafíos al permitir que el decodificador acceda a toda la secuencia de entrada codificada h1, h2, ..., hT. La idea central es inducir pesos de atención α sobre la secuencia de entrada para priorizar el conjunto de posiciones donde se encuentra la información relevante para generar el siguiente token de salida.

3.3. Uso de la atención

El bloque de atención que se utilizaría en una arquitectura codificador-decpdificador es el que permite aprender automáticamente los pesos, que capturan la relevancia entre los estados ocultos del codificador y del decodificador. Estos pesos se utilizan para construir un vector de contexto, que se pasa al decodificador. El vector de contexto es una combinación ponderada de los estados ocultos del codificador y sus pesos de atención correspondientes. Esto permite que el decodificador acceda a toda la secuencia de entrada y se enfoque en las posiciones relevantes. El uso de la atención mejora tanto el rendimiento en la tarea final como la calidad de la salida debido a una mejor alineación. En contraste con la arquitectura tradicional, donde el vector de contexto es el último estado oculto del codificador, en el enfoque basado en atención, el vector de contexto es una combinación de todos los estados ocultos del codificador y sus pesos de atención correspondientes.

3.4. Aprendizaje de los pesos de atención

Los pesos de atención se aprenden incorporando una red neuronal de feed-forward adicional dentro de la arquitectura. Esta red de feed-forward aprende un peso de atención particular α_{ij} como función de dos estados, h_i (encoder hidden state) y s_{j-1} (decoder hidden state), que se toman como entrada por la red neuronal. Esta función se conoce como función de alineación, ya que evalúa qué tan relevante es el estado oculto del encoder h_i para el estado oculto del decoder s_{j-1}. Esta función de alineación produce puntuaciones de energía e_{ij} que luego se introducen en la función de distribución, que convierte las puntuaciones de energía en pesos de atención. Cuando las funciones .ªƨ "p"son diferenciables, el modelo de codificador-decoder basado en atención se convierte en una única función diferenciable y se puede entrenar conjuntamente con los componentes codificador-decoder de la arquitectura utilizando la retropropagación simple.

3.5. Modelo de atención generalizado

El modelo de atención mostrado, mapea una secuencia de claves a una distribución de atención según una consulta. Las claves son los estados ocultos del codificador y la consulta es el estado único oculto del decodificador. La distribución de atención enfatiza las claves relevantes para la tarea principal. Además, puede haber valores adicionales sobre los cuales se aplica la distribución de atención. El modelo de atención combina las claves y la consulta para calcular los pesos de atención utilizando funciones de alineación y distribución. Un ejemplo concreto es el estimador de regresión, donde la consulta es la instancia y las claves son los puntos de datos de entrenamiento, y los valores son las etiquetas correspondientes.

3.6. Funciones de alineamiento

Las funciones de alineación se dividen en dos categorías principales. La primera categoría compara las representaciones de consulta y claves, utilizando métodos como el producto punto o la similitud del coseno. También se pueden emplear transformaciones aprendibles para adaptar las representaciones a un espacio vectorial común. La segunda categoría combina las claves y la consulta para crear una representación conjunta, utilizando enfoques como la concatenación o la alineación aditiva. También existen funciones de alineación diseñadas para casos específicos, como la alineación basada en ubicación que solo depende

de la consulta. Además, se pueden utilizar características derivadas de elementos individuales en grupos para las funciones de alineación.

3.7. Funciones de distribución

Las funciones de distribución en el mecanismo de atención asignan pesos de atención a partir de los puntuaciones de las funciones de alineación. Las funciones más comunes son la *sigmoide logística* y la *softmax*. Estas funciones aseguran que los pesos de atención estén entre 0 y 1, y sumen 1. Sin embargo, también existen funciones de distribución *sparsemax* y *sparse entmax* que generan pesos de atención dispersos, asignando probabilidad distinta de cero solo a unos pocos elementos plausibles. Además, las redes de desatención compuestas introducen una función de distribución que permite eliminar tokens irrelevantes para la consulta. En general, hay diversas formas de formular el mecanismo de atención, y en este estudio se discuten diferentes tipos de atención, arquitecturas neuronales y aplicaciones.

4. TAXONOMÍA DE LA ATENCIÓN

La atención se divide en cuatro categorías generales, cada una con diferentes tipos de atención. Estas categorías no son excluyentes y se pueden considerar como dimensiones para aplicaciones específicas. Por ejemplo, Yang et al. [2016] utilizaron una combinación de atención multi-nivel, auto-atención y atención "soft".

4.1. Número de secuencias

La atención se clasifica en diferentes tipos según las características de las secuencias de entrada y salida. El tipo de atención distintiva se utiliza cuando las secuencias de entrada y salida son diferentes, y se aplica en tareas como la traducción, la generación de subtítulos de imágenes y el reconocimiento de voz. En estos casos, se busca establecer una relación entre elementos clave en la secuencia de entrada y los estados de consulta en la secuencia de salida.

Por otro lado, la co-atención se emplea cuando se trabaja con múltiples secuencias de entrada y se busca capturar las interacciones entre ellas. Este enfoque conjuntamente aprende los pesos de atención para cada secuencia de entrada, lo que permite capturar las relaciones entre los diferentes elementos en cada secuencia. La co-atención ha sido aplicada en tareas como la respuesta a preguntas visuales, donde se busca comprender tanto la imagen de entrada como la pregunta formulada para generar una respuesta adecuada.

En contraste, en tareas como la clasificación de texto y las recomendaciones, donde la salida no es una secuencia sino un resultado único, se utiliza la auto-atención. La auto-atención permite aprender la relevancia de cada token en la misma secuencia de entrada. En este caso, los estados clave y de consulta pertenecen a la misma secuencia, y los pesos de atención se calculan para cada par de tokens en la secuencia. Esto ayuda a capturar las relaciones y dependencias entre los diferentes elementos dentro de la secuencia.

4.2. Número de niveles de abstracción

En el caso más general, los pesos de atención se calculan sólo para la secuencia de entrada original. Este tipo de atención se denomina de un solo nivel. Sin embargo, la atención también puede aplicarse en múltiples niveles de abstracción de manera secuencial. El vector de contexto resultante del nivel de abstracción inferior se convierte en el estado de consulta para el nivel de abstracción superior. Además, los modelos que utilizan atención multi-nivel pueden clasificarse adicionalmente según si los pesos se aprenden de arriba hacia abajo (desde un nivel superior de abstracción hasta uno inferior) o de abajo hacia arriba.

Un ejemplo clave en esta categoría es el Modelo de Atención Jerárquica (HAM, por sus siglas en inglés), que utiliza el modelo de atención en dos niveles diferentes de abstracción: nivel de palabras y nivel de oraciones, para la tarea de clasificación de documentos. El HAM captura la estructura jerárquica natural

de los documentos, donde un documento está compuesto por oraciones y las oraciones están compuestas por palabras. La atención multi-nivel permite al HAM extraer palabras importantes en una oración y oraciones importantes en un documento. Primero construye una representación basada en atención de las oraciones con atención de primer nivel aplicada a la secuencia de vectores de incrustación de palabras. Luego, agrega estas representaciones de oraciones utilizando una atención de segundo nivel para formar una representación del documento. Esta representación final del documento se utiliza como un vector de características para la tarea de clasificación.

Las Stacked Attention Networks (SANs) también entran en esta categoría, ya que utilizan múltiples capas para refinar iterativamente la atención combinando información de la consulta y los resultados de capas anteriores de atención. Por ejemplo, los autores en Sun y Fu [2019] utilizaron SANs para la tarea de respuesta a preguntas de imágenes, donde múltiples capas de atención consultan la imagen varias veces para localizar progresivamente las regiones exactas en la imagen que son altamente relevantes para la respuesta. Los autores afirman que utilizar una presentación global de la imagen para predecir la respuesta da resultados subóptimos, ya que la atención se dispersa en muchos objetos en la primera capa. Sin embargo, cuando se utilizan múltiples capas de atención, las capas de atención de nivel superior utilizan el conocimiento de las capas de atención de nivel inferior (información visual) y el vector de consulta refinado (información de la pregunta) para extraer regiones más detalladas y pequeñas dentro de la imagen. También observaron que dos capas de atención son mejores que una, pero que tres o más capas no mejoran aún más el rendimiento.

4.3. Número de posiciones

En esta tercera categoría, las diferencias surgen de las posiciones de la secuencia de entrada donde se calcula la función de atención. La atención "soft" utiliza un promedio ponderado de todos los estados ocultos de la secuencia de entrada para construir el vector de contexto. Por otro lado, se ha propuesto un un modelo de atención "hard" que computa el vector de contexto a partir de estados ocultos muestreados estocásticamente en la secuencia de entrada. Estas categorías no son mutuamente excluyentes y se han propuesto métodos de aprendizaje variacional y de gradiente de políticas para superar las limitaciones. Luong et al. [2015b] propusieron dos modelos de atención, local y global, para la tarea de traducción automática, donde el modelo de atención local proporciona un compromiso paramétrico entre la atención "soft" y la atención "hard" al seleccionar una ventana alrededor de un punto de atención dentro de la secuencia de entrada.

4.4. Número de representaciones

La atención se utiliza para asignar pesos a diferentes representaciones de características en la secuencia de entrada, lo que permite capturar aspectos relevantes y descartar el ruido. Esto se conoce como Modelo de Atención Multi-representacional. Al combinar estas representaciones ponderadas, se obtiene una representación final. Este enfoque ha sido utilizado con éxito en varias tareas, como mejorar las representaciones de oraciones y abordar la polisemia en el procesamiento del lenguaje natural. La asignación de atención a diferentes aspectos de la entrada ayuda a seleccionar las características más importantes para cada contexto, mejorando así el rendimiento en las aplicaciones subsiguientes.

5. ARQUITECTURAS DE REDES NEURONALES ATENCIONALES

Las cuatro principales arquitecturas de redes neuronales utilizadas en conjunto con la atención son:

1. Codificador-Decodificador.
2. Transformer.
3. Redes de memoria que extienden la atención para más de una entrada.
4. Redes de atención en grafo (GAT).

5.1. Esquema Codificador-Decodificador

El uso inicial de la atención fue como parte de un marco de codificador-decodificador basado en RNN para codificar largas frases de entrada. En consecuencia, la atención se ha utilizado ampliamente con esta arquitectura. Un hecho interesante es que el mecanismo de atención puede tomar cualquier representación de entrada y reducirla a un único vector de contexto de longitud fija que se utiliza en la etapa de decodificación. Esto permite separar la representación de entrada de la salida. Se puede aprovechar esta ventaja para introducir codificadores-decodificadores híbridos, siendo el más popular una **Red Neuronal Convolucional (CNN)** como codificador y **una RNN o una *Long Short-Term Memory* (LSTM)** como decodificador. Este tipo de arquitectura es particularmente útil para tareas multimodales como la descripción de imágenes y videos, la respuesta a preguntas visuales y el reconocimiento de voz.

Sin embargo, no todos los problemas en los que tanto la entrada como la salida son secuenciales se pueden resolver con la formulación anterior. Las redes de punteros son otro tipo de modelos neuronales con las siguientes dos diferencias: (i) la salida es discreta y señala posiciones en la secuencia de entrada (de ahí el nombre de red de punteros) y (ii) el número de clases objetivo en cada paso de la salida depende de la longitud de la entrada (y, por lo tanto, es variable). Esto no se puede lograr utilizando el marco tradicional de codificador-decodificador, donde el diccionario de salida se conoce de antemano (por ejemplo, en el caso del modelado del lenguaje natural). Los autores lograron esto utilizando los pesos de atención para modelar la probabilidad de elegir el símbolo de entrada i como el símbolo seleccionado en cada posición de salida. Este enfoque se puede aplicar a problemas de optimización discreta como el problema del viajero de comercio y la clasificación.

5.2. Transformer

Los autores proponen la arquitectura Transformer, que elimina el procesamiento secuencial y las conexiones recurrentes en las redes neuronales recurrentes. En su lugar, utiliza el mecanismo de atención propia para capturar las dependencias globales entre la entrada y la salida. El Transformer se compone de múltiples capas de codificadores y decodificadores que aplican transformaciones lineales y atención multicabeza. También se utiliza codificación posicional para capturar la información de orden de la secuencia de entrada. Los Transformers han demostrado capturar dependencias a largo plazo, admitir el procesamiento paralelo y ser aplicables a diversas tareas en NLP, Visión por Computadora y procesamiento multi-modal. Sin embargo, presentan limitaciones como la fragmentación del contexto y la complejidad paramétrica. Diversas investigaciones se centran en mejorar los Transformers, analizando los cabezales de atención, extendiendo el tamaño del contexto y reduciendo el tiempo de cálculo y el consumo de memoria.

En cuanto a las investigaciones sobre los cabezales de atención en los Transformers, se ha encontrado que ciertos cabezales se especializan en relaciones particulares en el lenguaje, y la mayoría de ellos pueden ser eliminados sin afectar el rendimiento. También se ha analizado el comportamiento de los Transformers en cuanto a la atención a tokens especiales y la extracción de alineación de palabras. Otra línea de investigación se enfoca en ampliar el alcance de la atención, utilizando nuevas técnicas de codificación posicional y aprovechando estados ocultos compartidos entre segmentos. Además, se han propuesto enfoques como Sparse Transformers, Reformers y Performers para reducir el costo computacional y el consumo de memoria de los Transformers.

5.3. Redes de memoria

Aplicaciones como *Question Answering* y chatbots requieren aprender de una base de datos de hechos. Se utiliza una memoria externa y la atención para enfocarse en los hechos relevantes. Tres enfoques en la literatura combinan la memoria externa con la atención: End-to-End Memory Networks, Dynamic Memory Networks y Neural Turing Machines. Las redes de memoria constan de tres componentes: un proceso que lee la base de datos y la convierte en representaciones distribuidas, una lista de vectores de características que almacena la salida del lector (la "memoria"), y un proceso que utiliza el contenido de la memoria para realizar una tarea secuencialmente, poniendo atención en diferentes elementos de la memoria en cada paso de tiempo.

- **Las redes de memoria de extremo a extremo (MemN2N)** permiten el entrenamiento completo mediante retropropagación y requieren menos supervisión que las *Memory Networks*. Son

útiles en tareas de procesamiento del lenguaje natural como *Question Answering* y *Language Modeling*. Su arquitectura consta de dos partes principales: la primera encuentra los hechos relevantes para una consulta en una base de conocimientos mediante comparación y softmax, mientras que la segunda calcula la respuesta final utilizando un vector de contexto y atención sobre los hechos relevantes.

- **Dynamic Memory Networks (DMN)** utiliza un módulo de memoria episódica que selecciona qué partes de las entradas enfocar mediante el mecanismo de atención y genera una representación vectorial de la memoria. Repite este proceso de manera iterativa al condicionar la atención tanto en la pregunta como en la representación de memoria anterior, lo que permite que el módulo (i) se centre en diferentes entradas en cada iteración y (ii) recupere información adicional que se consideró irrelevante en iteraciones anteriores. Las preguntas activan compuertas que controlan qué entradas se proporcionan al módulo de memoria episódica. El estado final de la memoria episódica (después de múltiples episodios/iteraciones) se utiliza como entrada para el módulo de respuesta.

- **El Neural Turing Machine (NTM)** es un sistema que utiliza una representación de memoria continua junto con un controlador para realizar operaciones de lectura y escritura en la memoria. A diferencia de una Máquina de Turing convencional, el NTM es diferenciable y puede ser entrenado eficientemente. Utiliza el mecanismo de atención para acceder selectivamente a la memoria y realizar interacciones con una pequeña parte relevante de la misma. Además, el NTM es capaz de aprender algoritmos simples a partir de ejemplos de entrada y salida. Las redes de memoria ofrecen la ventaja de almacenar información en forma de memoria y utilizarla de manera efectiva mediante la atención selectiva. Los modelos MemN2N han mostrado un rendimiento superior en tareas como *Question Answering* y *Language Modeling* en comparación con los modelos RNN y LSTM. Por otro lado, los modelos DMN han logrado resultados destacados en *Sentiment Analysis* y *Part of Speech Tagging*.

5.4. Redes atencionales de grafos (GAT)

Los datos estructurados en forma de gráficos, como redes sociales, redes de citas, interacciones proteína-proteína, química, entre otros, presentan desafíos debido a la variabilidad en el número de vecinos en los nodos y la necesidad de eficiencia computacional. Generalizar las capas convolucionwales utilizadas en imágenes a capas convolucionales de gráficos requiere innovaciones para lograr eficiencia computacional y de almacenamiento, un número fijo de parámetros, capacidad de localización, capacidad de asignar importancia arbitraria a los vecinos y aplicabilidad a estructuras de gráficos arbitrarias e invisibles.

Los Graph Convolutional Networks (GCN) combinan la estructura local del grafo y las características de los nodos, pero asignan pesos explícitos no paramétricos basados en el grado del nodo. Para abordar esto, los Graph Attention Networks (GAT) utilizan la autoatención sobre las características de los nodos vecinos, asignando pesos basados en la importancia de los nodos. Los GAT son eficientes computacionalmente y permiten asignar diferentes importancias a los nodos del mismo vecindario a través de pesos de atención.

Además, se han propuesto extensiones de atención jerárquica para grafos heterogéneos, que contienen información más completa y semántica rica. Estas extensiones permiten aprender los valores de atención entre nodos y sus vecinos basados en metapath y combinar óptimamente vecinos y múltiples metapath en una estructura jerárquica.

6. APLICACIONES

Los modelos de atención se han convertido en un área activa de investigación debido a su intuición, versatilidad e interpretabilidad. Se han utilizado variantes de modelos de atención para abordar las características únicas de diversos campos de aplicación. En algunos casos, los modelos de atención han demostrado tener un impacto significativo en el rendimiento de la tarea en cuestión, mientras que en otros han ayudado a aprender mejores representaciones de entidades como documentos, imágenes y grafos. En algunos casos, la atención ha transformado por completo el campo de aplicación al convertirse en la técnica más popular.

Se describen modelos de atención en los siguientes dominios de aplicación: (i) Procesamiento del Lenguaje Natural (NLP), (ii) Visión por Computadora, (iii) Tareas Multi-Modales, (iv) Sistemas de Recomendación y (v) Sistemas Gráficos.

6.1. *Natural Language Processing (NLP)*

En el dominio del Procesamiento del Lenguaje Natural (NLP), la atención se utiliza para enfocarse en partes relevantes de las secuencias de entrada, alinear secuencias y capturar dependencias a largo plazo. En la traducción automática, la atención mejora la alineación de oraciones en diferentes idiomas y facilita la traducción de oraciones largas. En preguntas y respuestas, la atención ayuda a comprender mejor las preguntas y encontrar respuestas relevantes. En el análisis de sentimientos, la atención se utiliza para enfocarse en palabras importantes para determinar el sentimiento de entrada. Además, se han desarrollado modelos de atención para la clasificación de texto y el aprendizaje de representaciones de texto en NLP.

Los modelos de lenguaje pre-entrenados, como BERT, GPT y Transformer, han revolucionado muchas aplicaciones de NLP al capturar representaciones de lenguaje universal, permitir el ajuste fino para diversas tareas y facilitar el desarrollo de modelos de NLP. Estos modelos han logrado resultados destacados en una variedad de tareas de NLP, como traducción automática, resumen de textos, análisis de sentimientos y clasificación de texto.

6.2. Visión por computadora

La atención visual se ha vuelto popular en la visión por computadora para enfocarse en regiones relevantes de una imagen y capturar dependencias a larga distancia. Se utiliza en tareas como clasificación de imágenes, detección de objetos y generación de imágenes. La atención visual ayuda a seleccionar regiones importantes, reduce la complejidad computacional y mejora el rendimiento de los modelos.

Además de las redes neuronales convolucionales (CNN), los Transformers se están utilizando en tareas de visión para lograr eficiencia y escalabilidad. El Vision Transformer (ViT) utiliza la arquitectura Transformer en parches de imagen para clasificación y supera a las CNN con menos recursos computacionales. Otros enfoques, como el Detection Transformer (DETR), utilizan Transformers para detección de objetos, eliminando componentes diseñados a mano y mejorando la eficiencia. Los Transformers también se aplican a la generación de imágenes, donde se predice cada píxel de forma secuencial.

Estos avances en atención visual y Transformers están impulsando el desarrollo de arquitecturas eficientes, escalables y aplicables a diferentes dominios en la visión por computadora.

6.3. Tareas multi-modales

La atención se ha utilizado ampliamente en aplicaciones multi-modales para comprender las relaciones entre diferentes modalidades. En tareas de descripción multimedia, la atención se utiliza para encontrar partes relevantes de una imagen o video y generar una descripción en lenguaje natural. En el reconocimiento de voz, la atención es crucial para evitar el sobreajuste y considerar tanto la ubicación como el contenido de los fragmentos importantes en la secuencia de entrada. También se ha demostrado que la atención es efectiva en la comprensión de la comunicación humana cara a cara, donde se utiliza para descubrir interacciones entre diferentes modalidades en cada paso de tiempo.

Los Transformadores también se utilizan ampliamente en tareas de visión y lenguaje para aprender representaciones genéricas que capturan las relaciones multi-modales. Hay dos tipos principales de Transformadores utilizados en este campo: de flujo único y de flujo múltiple. Los modelos de flujo único procesan todos los datos de entrada en un solo Transformer, mientras que los modelos de flujo múltiple utilizan Transformers independientes para cada modalidad y luego aprenden representaciones multi-modales utilizando otro Transformer de co-atención.

6.4. Sistemas recomendadores

La atención ha sido ampliamente utilizada en sistemas de recomendación para diversas aplicaciones. En el perfilado de usuarios, la atención se utiliza para asignar pesos a los ítems con los que un usuario interactúa, capturando así sus intereses a largo y corto plazo de manera más efectiva. Además, se ha utilizado la atención para mejorar el filtrado colaborativo y los modelos secuenciales basados en RNN.

El aprendizaje de representaciones efectivas de usuarios e ítems es fundamental en los sistemas de recomendación. Se han propuesto enfoques que utilizan la atención para combinar las incrustaciones de usuarios e ítems aprendidas en diferentes dominios, generando una representación única. También se ha utilizado la atención jerárquica para aprovechar la estructura de las revisiones de artículos y aprender representaciones más efectivas de los ítems.

Además, la atención se ha aplicado para aprovechar la información auxiliar de manera más efectiva. Por ejemplo, se han construido grafos híbridos que vinculan las interacciones usuario-ítem y los atributos de los ítems, y se utilizan pesos de atención para calcular las incrustaciones de los nodos. También se ha utilizado la atención para atender a diferentes aspectos que afectan las preferencias del usuario, como el historial de carga, la influencia social y la admiración del propietario.

Por último, se ha propuesto un modelo de recomendación social que utiliza la atención para aprovechar los amigos de orden superior en una red social, permitiendo que el contexto de un usuario atienda a los contextos de sus amigos y utilice los intereses y conocimientos agregados para la recomendación.

6.5. Sistemas basados en grafos

Muchos conjuntos de datos importantes en el mundo real se presentan en forma de grafos o redes, como redes sociales, grafos de conocimiento, redes de interacción de proteínas y la World Wide Web. La atención se ha utilizado para resaltar elementos del grafo (nodos, aristas, subgrafos) que son más relevantes para la tarea principal. El enfoque común es calcular incrustaciones guiadas por atención de nodos, aristas, subgrafos o combinaciones de estos. La arquitectura de atención en grafos es eficiente, ya que se puede paralelizar en pares de vecinos de nodos, se puede aplicar a nodos de grafos con diferentes grados, y es aplicable directamente a problemas de aprendizaje inductivo, incluidas tareas en las que el modelo tiene que generalizar a grafos completamente desconocidos. A diferencia de las Redes Convolucionales en Grafos, el mecanismo de atención en grafos permite asignar diferentes importancias a nodos de un mismo vecindario, lo que aumenta la capacidad del modelo. Analizar los pesos de atención aprendidos puede tener beneficios en cuanto a interpretabilidad. La atención se ha utilizado en varias tareas de aprendizaje automático en datos estructurados en forma de grafo, incluyendo (i) Clasificación de Nodos, (ii) Predicción de Enlaces, (iii) Clasificación de Grafos y (iv) Generación de Secuencias a partir de Grafos. También se ha utilizado la atención para la predicción de hiperaristas en hipergrafos, que son utilizados para representar interacciones de orden superior en grafos mediante hiperaristas, es decir, aristas que conectan múltiples nodos. Los métodos existentes están diseñados principalmente para analizar interacciones de pares y, por lo tanto, no pueden capturar de manera efectiva las interacciones de orden superior en los grafos. Como resultado, los autores proponen un GAT basado en auto-atención para la predicción de hiperaristas en hipergrafos homogéneos y heterogéneos con tamaño de hiperarista variable.

7. ATENCIÓN PARA LA INTERPRETABILIDAD

El creciente interés en la interpretabilidad de los modelos de IA se debe tanto al rendimiento como a la transparencia y equidad de los modelos. Sin embargo, las redes neuronales, especialmente las arquitecturas de aprendizaje profundo, han sido criticadas por su falta de interpretabilidad. El modelado de la atención es interesante desde la perspectiva de la interpretabilidad, ya que nos permite inspeccionar directamente el funcionamiento interno de estas arquitecturas. Se plantea la hipótesis de que la magnitud de los pesos de atención se correlaciona con la relevancia de una región específica de entrada para la predicción de la salida en cada posición de una secuencia. Esto se puede lograr fácilmente visualizando los pesos de atención para un conjunto de pares de entrada y salida.

Varios estudios han demostrado que los modelos de atención pueden enfocarse en palabras relevantes y capturar relaciones entre palabras para la generación de resúmenes y la identificación de intereses de los usuarios. Además, se ha encontrado que los pesos de atención pueden alinear automáticamente oraciones en diferentes idiomas, a pesar de las diferencias en la estructura lingüística. También se han explorado aplicaciones interesantes de los pesos de atención en la detección de sesgos de género, la determinación del sentimiento de las reseñas y la identificación de posiciones iniciales en señales de audio.

A pesar de su popularidad para revelar el funcionamiento interno de las redes neuronales opacas, el uso de los pesos de atención para la interpretabilidad del modelo sigue siendo objeto de investigación. Algunos estudios han desafiado la idea de que los pesos de atención sean explicaciones válidas del comportamiento del modelo, argumentando que no están correlacionados con la importancia de las características típicas y que cambiarlos no afecta las predicciones del modelo de manera significativa. En general, la interpretabilidad de los modelos de atención sigue siendo un área de investigación activa.

8. CONCLUSIÓN

En este estudio, se discuten diferentes formas en las que se ha formulado la atención en la literatura y se intenta proporcionar una visión general de varias técnicas al discutir una taxonomía de atención, arquitecturas clave de redes neuronales que utilizan la atención y dominios de aplicación que han experimentado un impacto significativo. Se habla sobre cómo la incorporación de la atención en las redes neuronales ha llevado a mejoras significativas en el rendimiento, ha proporcionado una mayor comprensión del funcionamiento interno de las redes neuronales al facilitar la interpretabilidad y también ha mejorado la eficiencia computacional al eliminar el procesamiento secuencial de la entrada. Se espera que este estudio proporcione una comprensión de las diferentes direcciones en las que se ha investigado sobre este tema y cómo las técnicas desarrolladas en un área pueden aplicarse a otros dominios. Se concluye este estudio con algunas de las direcciones de investigación emergentes en el modelado de la atención.

8.1. Atención en tiempo real

Los modelos de traducción automática usualmente funcionan de manera secuencial, pero en aplicaciones en tiempo real se requiere la traducción antes de terminar la oración. Se han propuesto enfoques como la atención monotónica por fragmentos y la atención monotónica de múltiples cabezas para permitir la decodificación en tiempo real. Estos avances son relevantes para futuras investigaciones.

8.2. Atención autónoma

Introducir la atención en modelos de vanguardia como las CNN en Visión por Computadora ha mejorado su rendimiento. Se cuestiona si la atención puede ser una primitiva independiente en modelos de visión en lugar de un complemento a las convoluciones.Se han investigado modelos de visión puramente basados en atención, reemplazando las convoluciones espaciales por atención autocontenida en regiones locales, y se encontró que estos modelos pueden competir con los modelos de referencia en conjuntos de datos de visión.

8.3. Destilación de modelos

Las aplicaciones industriales, como los sistemas de recomendación y búsqueda, tienen restricciones estrictas de latencia para el servicio en línea de modelos. Sin embargo, los modelos pre-entrenados, como BERT, tienen cientos de millones de parámetros, lo que los hace inadecuados para el servicio en línea. La destilación de modelos tiene como objetivo comprimir un modelo grande y complejo en un modelo más simple sin perder su precisión. Se ha utilizado la auto-atención profunda para entrenar un modelo más pequeño imitando el módulo de auto-atención del modelo grande. Además, se ha observado que agregar un asistente de profesor también ayuda en la destilación de modelos Transformers pre-entrenados. También se ha empleado una estrategia profesor-estudiante específica para los modelos Transformers, donde el estudiante aprende del profesor a través de la atención.

8.4. Atención para la interpretabilidad

La investigación en la relación entre los pesos de atención y la interpretabilidad del modelo continúa siendo activa. Se puede investigar cómo modificar las distribuciones de atención de los modelos actuales para ofrecer justificaciones plausibles de las predicciones del modelo. Por ejemplo, se ha propuesto una celda LSTM modificada que genera pesos de atención más precisos, indicativos de palabras importantes y correlacionados con métodos de atribución basados en gradientes. Esta investigación sugiere que los modelos LSTM pueden beneficiarse de técnicas que promueven la diversidad entre los estados ocultos para mejorar la interpretabilidad y el rendimiento del modelo.

8.5. Anteción con aprendizaje automático

La búsqueda automatizada de arquitecturas de redes neuronales utilizando Neural Architecture Search (NAS) ha demostrado superar los diseños humanos en diversas tareas. Surge la pregunta de si es posible utilizar NAS para buscar la arquitectura óptima de un módulo de atención de alto orden. Se ha propuesto un enfoque novedoso llamado Higher Order Group Attention, que permite representar y utilizar atenciones de alto orden en el proceso de búsqueda. Este enfoque utiliza un método de búsqueda diferenciable para encontrar eficientemente el módulo de atención óptimo.

8.6. Atención multi-instancia

Los mecanismos de atención existentes se centran en elementos individuales en la memoria, pero la atención de múltiples instancias permite enfocarse en grupos de elementos adyacentes. La atención de área es una técnica que modela las claves de un área como el promedio de las claves de los elementos dentro de ella. También se pueden utilizar características derivadas, como la desviación estándar de las claves, para formar una representación más completa de cada área. Este enfoque es útil para explorar la atención en grupos de elementos con formas y tamaños variables.

8.7. Sistemas multi-agente

La comprensión y modelado del comportamiento de sistemas multiagente es esencial en diversas aplicaciones del mundo real, como vehículos autónomos y juegos multijugador. El mecanismo de atención, junto con los modelos generativos profundos, puede ser utilizado para modelar las interacciones dentro de estos sistemas. Se han propuesto enfoques que utilizan la atención para capturar el proceso de generación de comportamiento en sistemas multiagente y para identificar grupos de agentes y sus interacciones.

8.8. Escalabilidad

Los modelos Transformer grandes han demostrado ser altamente efectivos en diversas aplicaciones de procesamiento del lenguaje natural (NLP) y visión por computadora. Sin embargo, su entrenamiento y despliegue pueden ser costosos para secuencias largas debido a la complejidad cuadrática del mecanismo de autoatención. La investigación se centra en reducir esta complejidad a lineal sin perder rendimiento. Se han propuesto varias soluciones, como el uso de una función de enmascaramiento aprendible para controlar el alcance de atención, aproximaciones de atención lineal como Performers y el desarrollo de mecanismos de autoatención de bajo rango como el Linformer. Además, se han explorado enfoques como el Predictor de Conexiones que utiliza modelos LSTM para buscar patrones de atención óptimos en secuencias largas.

9. SEGUNDA PARTE: APLICACIONES PARA ESTE PROYECTO

Posibles aplicaciones en el contexto del proyecto de compartición de prototipos con el modelo ILVQ:

9.1. Recomendador centralizado

Un primer ejemplo sería el siguiente: Además de los nodos que están ejecutando el modelo ILVQ, se tiene un nodo ejecutando un modelo atencional. Cuando uno de los nodos ILVQ quiera compartir su conjunto de prototipos, se los envía al nodo recomendador y le envía también a qué nodos se los iba a enviar (los nodos destino se deciden según los protocolos de compartición definidos previamente), el nodo recomendador calcula a qué nodos sería relevante (estudiar qué sería relevante y qué no) enviar estos datos, y se lo envía.

Por ejemplo: El nodo 1 quiere compartir su conjunto de prototipos a los nodos 2 y 3. Este le envía su conjunto de prototipos al nodo recomendador junto con los IDs destino. El nodo recomendador calcula la relevancia del conjunto de prototipos del nodo 1 para los nodos 2 y 3, establece que para el nodo 2 no es relevante pero para el 3 sí, y reenvía el mensaje al nodo 3.

No muy útil puesto que no reduciría el número de mensajes enviados en ningún caso. (Estudiar solución mejor con un nodo recomendador). La ventaja que tiene sobre los distribuidos, **9.2** es que tendría el conjunto de prototipos más actualizados. Cuanto más desactualizados tenga cada nodo los conjuntos de prototipos de sus vecinos, probablemente reduzca sus prestaciones.

9.2. Recomendadores distribuidos

Sería de forma similar al ejemplo anterior pero en este caso los propios nodos que ejecutan el modelo ilvq, ejecutarían a su vez el modelo atencional recomendador. Esto sí reduciría el número de mensajes intercambiados pero aumentando el consumo de estos nodos (estudiar casos, mínimo mismos mensajes, estudiar también reducción de prestaciones en relación al número de mensajes que se han dejado de enviar).

9.2.1. A nivel de conjunto de prototipos

En este caso, cada nodo al recibir el conjunto de prototipos de los otros nodos, además de entrenarse con ellos, guarda en memoria el último conjunto recibido por cada nodo. Cuando este nodo decida compartir, mediante el modelo recomendador, obtiene si su conjunto de prototipos sería relevante para cada nodo destino, descartando el envío para los cuales no lo fuese.

Ejemplo: El nodo 1, siguiendo el protocolo de compartición obtiene como destino de su conjunto de prototipos, los nodos 2 y 3. A partir del conjunto de prototipos almacenado de cada nodo, le "pregunta" al modelo recomendador para cuáles sería relevante el envío de su conjunto, el modelo responde que la relevancia para el nodo 2 es muy baja, mientras que para el nodo 3 es relativamente alta. Decide enviar sólo al nodo 3 su conjunto de prototipos. Se ha ahorrado el envío de 1 mensaje.

9.2.2. A nivel de prototipos individuales

Este caso sería muy similar al caso anterior, con la diferencia de que ahora se estudia más concretamente cada prototipo para establecer cuáles serían realmente útiles utilizando el modelo recomendador. Se compara cada prototipo a compartir con el último conjunto almacenado de los nodos destino, los que el modelo establece como relevantes, son los que se comparten.

Este caso no reduciría tanto el número de mensajes enviados como el anterior, pero reduciría mucho el tamaño de estos, y al ser más preciso en qué prototipos son relevantes y no en el conjunto que sería un estudio *a grosso modo*. Por lo tanto mejoraría la relación de reducción de prestaciones respecto al número de mensajes/bytes enviados que en el caso anterior, **9.2.1**.

9.3. Clasificación de nodos

En aplicaciones relacionadas con grafos y nodos se podría hacer una clasificación de los nodos de la siguiente forma. Cada T segundos, los nodos envían su conjunto de prototipos (podría enviarle su capacidad predictiva también) a un nodo clasificador, este nodo, teniendo todos los conjuntos de prototipos, calcularía una puntuación de relevancia para cada nodo y se la enviaría. En funcion de la puntuación de relevancia obtenida, se incrementa o disminuye la probabilidad de compartición (incrementa o decremente el parámetro T asigando al inicio de la ejecución del programa, teniendo así un T variable para cada nodo e incluso un mismo nodo podría ser relevante en un instante de tiempo pero pasado un tiempo dejar de serlo).

9.3.1. Detección de nodos anómalos

Relacionado con la clasificación de nodos, **9.3**, se podría añadir un modelo para detectar nodos anómalos y reaccionar en consecuencia, ignorando sus prototipos o haciendo que se le envíen muchos prototipos de sus vecinos hasta que deje de tener resultados anómalos.

9.4. División en subgrafos

Dividir el conjunto de nodos en dos o más subgrafos en función de su desempeño. Los nodos más relevantes comparten más datos a los que son menos relevantes, los nodos de los peores subgrafos no comparten o reducen mucho el número de comparticiones a los más relevantes.

Por ejemplo: Tenemos 5 nodos, de los cuales el 1 y 2 tienen un muy buen desempeño, mientras que el 3,4 y 5, bastante deficiente. El modelo divide en dos subgrafos, el subgrafo 12, y el subgrafo 345. El subgrafo 12 sería el formado por los nodos 1 y 2, mientras que el 345, por los nodos 3,4 y 5. Los nodos del subgrafo 12, aumentarían la probabilidad de compartición de sus prototipos a todos los nodos del subgrafo 345, mientras que los nodos del subgrafo 345 dejarían de compartir al subgrafo 12. Esto resultaría en una mejora del rendimiento de los nodos del subgrafo 345 reduciendo el número de mensajes compartidos.
Muchas de estas aplicaciones podrían utilizarse conjuntamente, por ejemplo, se podría tener recomendadores distribuidos y a la vez cada cierto tiempo hacer una división en subgrafos, compartiendo los mejores nodos con los peores, algo así como realizando correcciones cada intervalo de tiempo T.

10. IMPLEMENTACIÓN

Las posibles implementaciones basadas en las aplicaciones descritas previamente, serían las siguientes:

10.1. Sistemas recomendadores

Para los sistemas recomendadores se utilizarían Transformer.

EVALUAR TAMBIÉN CÓMO UTILIZAR MODELO RECOMENDADOR, PODRÍA ENTRENARSE A LA VEZ QUE PREDICE COMO EL ILVQ, O PODRÍA SER PREENTRENADO.

10.2. Clasificación de nodos y división en subgrafos.

Para poder llevar a cabo cualquier aplicación relacionada con grafos tendría que realizarse con la arquitectura de Redes Atencionales de Grafos (GAT).