**Capítulo 16: Procesamiento de lenguaje natural con RNNs y atención**

**Introducción**

\_ En este capítulo, se explora el uso de redes neuronales recurrentes (RNNs) y mecanismos de atención en el campo del procesamiento de lenguaje natural (NLP).

Procesamiento de lenguaje natural (NLP): es una rama de la inteligencia artificial que se centra en la comprensión y generación del lenguaje humano. Es una tarea desafiante debido a la complejidad y ambigüedad del lenguaje natural.

RNN: es un tipo de arquitectura de redes neuronales diseñada para modelar secuencias de datos. Las RNNs son especialmente adecuadas para tareas de procesamiento de lenguaje natural, ya que pueden capturar la dependencia entre las palabras en una oración o en un texto.

Atención: demás de las RNNs, se presenta el concepto de atención, que es un mecanismo que permite a la red centrarse en partes específicas de la entrada durante la generación de la salida. La atención mejora la capacidad de las redes para capturar relaciones importantes y seleccionar la información relevante en un contexto dado.

**Sentiment Analysis**

Concepto

\_ El análisis de sentimientos es una técnica de procesamiento de lenguaje natural que se utiliza para determinar la actitud o la emoción expresada en un texto. El objetivo del análisis de sentimientos es identificar si un texto tiene un sentimiento positivo, negativo o neutro. En este contexto, el análisis de sentimientos se aplica a textos como reseñas de productos, comentarios en redes sociales, opiniones de usuarios, entre otros. Es especialmente útil para empresas y organizaciones que desean comprender la opinión y el sentimiento de los clientes hacia sus productos o servicios. El análisis de sentimientos puede proporcionar información valiosa para las empresas, ya que les permite comprender la satisfacción del cliente, identificar problemas o áreas de mejora, monitorear la reputación en línea y tomar decisiones informadas sobre estrategias de marketing y atención al cliente.

\_ Al aplicar el análisis de sentimientos, se procesa el texto de entrada y se utiliza el modelo entrenado para predecir el sentimiento asociado.

Aprendizaje profundo o Deep Learning: es una técnica específica dentro del campo más amplio del aprendizaje automático, que utiliza redes neuronales profundas para abordar problemas complejos y extraer características significativas de los datos.

Desarrollo del código

\_ En este caso se utilizará el conjunto de datos de reseñas de IMDb, ya que sirve de ejemplo para realizar tareas de análisis de sentimientos o clasificación de textos. Este conjunto de datos es ampliamente utilizado en el campo del procesamiento del lenguaje natural debido a su disponibilidad, tamaño razonable y desafío asociado. El objetivo de utilizar este conjunto de datos es entrenar modelos de aprendizaje automático para comprender y predecir el sentimiento asociado a una determinada reseña de película. Al tener una etiqueta binaria que indica si una reseña es positiva o negativa, se puede utilizar este conjunto de datos para realizar tareas de clasificación binaria.

\_ El conjunto de datos de reseñas de IMDb se considera el "hola mundo" del procesamiento del lenguaje natural, ya que es un punto de partida común para los principiantes que desean explorar y comprender los conceptos y técnicas básicas de análisis de sentimientos en el procesamiento del lenguaje natural.

Token de relleno (padding token): cuando trabajamos con datos secuenciales, como frases o textos, es común que las secuencias tengan diferentes longitudes. Para alimentar estas secuencias a un modelo de ML, necesitamos que todas tengan la misma longitud. El token de relleno se utiliza para agregar valores adicionales (por lo general, ceros) al final de las secuencias más cortas, de modo que todas tengan la misma longitud.

Token de inicio de secuencia (start-of-sequence token o SOS token): este token se coloca al principio de una secuencia para indicar el inicio de una nueva secuencia o frase. Ayuda a los modelos a capturar mejor la estructura y las relaciones dentro de una secuencia de palabras.

Embedding: es una representación vectorial de una palabra o un objeto. Es una técnica que asigna palabras a vectores. Cada dimensión en el vector representa una característica o un atributo específico de la palabra. El objetivo del embedding es capturar las similitudes semánticas y sintácticas entre las palabras. Se espera que palabras similares o relacionadas tengan representaciones de embedding cercanas entre sí en el espacio vectorial.

Época: representa una vuelta completa del modelo sobre todo el conjunto de datos de entrenamiento. Durante cada época, el modelo analiza los datos de entrenamiento por lotes, aprende de ellos y ajusta sus parámetros. Cuantas más épocas se realicen, más oportunidades tiene el modelo de mejorar su rendimiento y aprender patrones en los datos. Es como si el modelo estuviera estudiando repetidamente el mismo material de entrenamiento para comprenderlo mejor con cada pasada.

API funcional en Keras: permite construir modelos más complejos que involucran múltiples entradas, salidas o conexiones entre capas no lineales. Además, proporciona una mayor flexibilidad para personalizar la arquitectura del modelo, compartir capas entre diferentes ramas del grafo y manipular los datos de entrada y salida de cada capa.

Enfoque secuencial en Keras: es una forma sencilla y directa de construir modelos de redes neuronales. En este enfoque, las capas se apilan secuencialmente una encima de la otra para formar el modelo. Cada capa tiene una única entrada y una única salida, y la información fluye de manera lineal a través de las capas.

Capas GRU: son capas recurrentes que permiten a las redes neuronales modelar y procesar secuencias de datos de manera más eficiente y efectiva.

Masking: es una técnica utilizada para ignorar o marcar ciertos valores en los datos de entrada. La máscara se utiliza para indicar qué partes de las secuencias son reales y qué partes son valores de relleno. Cuando se aplica la máscara a una secuencia, los valores marcados como relleno se ignoran durante el procesamiento. Esto es útil para evitar que el modelo se confunda con los valores de relleno y se centre en los elementos importantes de la secuencia.

* En el código mencionado, la capa Embedding tiene el parámetro mask\_zero=True, lo que indica que se debe aplicar una máscara a los valores de cero en las secuencias de entrada. Esto significa que los valores de relleno se marcarán y se ignorarán durante el procesamiento posterior, lo que ayuda al modelo a centrarse en los elementos relevantes de la secuencia.

Reusing Pretrained Embeddings: reutilizar incrustaciones preentrenadas es una estrategia efectiva para aprovechar el conocimiento adquirido en grandes corpus de texto y mejorar el rendimiento en tareas de procesamiento de lenguaje natural al transferir este conocimiento general a dominios más específicos. En lugar de entrenar los vectores de palabras desde cero en un conjunto de datos específico, se aprovecha el conocimiento adquirido en un corpus más amplio. Ventajas:

* Reducen los recursos computacionales para entrenar los vectores de palabras desde cero.
* Se evita el problema de tener un conjunto de datos insuficiente.
* Las incrustaciones preentrenadas ya contienen información valiosa y pueden ayudar a mejorar el rendimiento del modelo.

\_ El proyecto TensorFlow Hub facilita la reutilización de componentes de modelos preentrenados en tus propios modelos. Estos componentes de modelo se llaman módulos.