Sistema de Recomendación Secuencial usando Redes Neuronales Recursivas

Jackson Vera Pineda
Kevin Manzano Rodríguez
Roger Fuentes Rodríguez

https://github.com/jackverneda/sri-summer-2024

Resumen Este informe describe la implementación de un sistema de recomendación secuencial basado en redes neuronales recurrentes (RNN) utilizando capas LSTM (Long Short-Term Memory) y GRU (Gated Recurrent Units) haciendo uso además de modelos de lenguaje para capturar la semántica de los productos en embeddings. Este sistema tiene como objetivo predecir el siguiente producto en el que un usuario podría estar interesado, basándose en sus interacciones anteriores.

Keywords: Recomendación secuencial, Redes Neuronales Recursivas, LSTM, GRU, Deep Learning

1. Introducción

En la era del comercio electrónico y las plataformas de contenido, la capacidad de ofrecer recomendaciones personalizadas a los usuarios se ha convertido en un factor crucial para mejorar la experiencia del usuario y aumentar la retención. Un enfoque efectivo es predecir el siguiente producto que un usuario podría interactuar basándose en su historial de interacciones pasadas. Este problema se conoce como "recomendación secuencial", donde el objetivo es utilizar secuencias temporales de interacciones de usuarios para predecir sus futuras acciones.

El desafío principal en la recomendación secuencial es capturar las dependencias temporales y la evolución de las preferencias de los usuarios a lo largo del tiempo. Las Redes Neuronales Recursivas (RNN) y sus variantes, como LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Units*), han demostrado ser herramientas poderosas para modelar secuencias temporales debido a su capacidad para retener y procesar información a través de secuencias largas.

2. Formulación del Problema

Una tienda online de cosméticos requiere de un sistema de recomendación para sugerir productos de interés personal en un futuro a los usuarios, esto es un desafío debido a la gran cantidad de productos que posee y un número de clientes en constante crecimiento. Actualmente la tienda utiliza una búsqueda por coincidencias, esto es efectivo pero no tiene en cuenta sugerencias creativas. La tienda registra cada compra efectuada por el usuario, información valiosa que se dispone para la recomendación.

3. Propuesta de la Solución

Entre las soluciones conocidas figuran: Modelos Basados en Redes Neuronales Recursivas, Modelos Basados en Atención, Modelos Basados en Grafos y Modelos de Refuerzo.

Los Modelos Basados en Redes Neuronales Recursivas han demostrado ser especialmente efectivos para capturar dependencias temporales en secuencias de datos, lo que los hace adecuados para sistemas de recomendación secuencial [2]. Un ejemplo destacado es el uso de redes neuronales recurrentes (RNN) para recomendaciones basadas en sesiones, como se describe en el trabajo de Hidasi et al. [1]. Este enfoque utiliza variantes como GRU (Gated Recurrent Units) para mejorar la precisión de las recomendaciones basadas en el historial de interacciones de los usuarios. Además, la evaluación empírica de GRU en modelado secuencial ha mostrado resultados prometedores en comparación con LSTM [3].

En este trabajo se brinda un enfoque similar a [1] con diferencias en ciertos momentos del proceso. La principal diferencia con los modelos clásicos que utilizan el id del producto como entrada al modelo, es transformar este id en un vector de mayor dimensión dentro de un espacio de productos con relaciones implicitas entre ellos. La semántica de este espacio está dada por los nombres de los productos para que se asemeje al lenguaje natural. Por lo tanto para transformar los productos en vectores de este espacio se propone utilizar un modelo de embeddings. De esta forma si se quiere un espacio de productos de dimensión n, el título de un producto se convertiría en un vector de dimensión n y este sería la re representación en el espacio.

Además este tipo de problemas se trata como un problema de clasificación donde la salida de la red neuronal es un vector de probabilidades cuya dimensión es la cantidad de productos pero, para una cantidad grande de datos se hace costoso el computo. Por tanto en el presente se propone el uso de una regresión porque es más liviana con el riesgo de que el vector resultante no exista como producto, y sea necesario buscar el más cercano.

4. Implementación de la Solución

En esta solución, se emplea una red neuronal basada en LSTM y GRU para abordar el problema de la recomendación secuencial. El proceso se puede dividir en las siguientes etapas clave:

4.1. Fuente de los Datos

El sitio `https://amazon-reviews-2023.github.io/` es un dataset donde se registra los reviews de los usuarios a productos de Amazon. Tiene los reviews de los años 2023, 2018 y 2014, utilizamos de la más reciente la categoría `all beauty` que posee cerca de 31,6 millones y 112,6 mil de productos. Consideramos este sitio porque es cómodo la forma en la que se almacenan la relación

usuario-review-producto. Se arma la secuencia para cada usuario de los productos referenciados en orden cronólogico. Como consideración se piensa que los datos no describen bien el comportamiento real, porque la media de los usuarios que compran productos no sienten la necesidad de dejar review, solamente los casos extremos donde se sienten o bien muy satisfechos, o muy inconformes esto produce un sesgo.

4.2. Preparación de Datos

Una vez descargados los datos se cargan desde los archivos `All_Beauty.json` y `meta_All_Beauty.json` que contienen las interacciones de los usuarios y la información de los productos, respectivamente. Esta información es almacenada en `DataFrame` de la biblioteca de `pandas`, estructura de datos útil para la manipulación de datos tabulares.

Las secuencias son listas de reviews hechas por un usuario. Para conformarlas se extraen datos de la tabla de las interacciones, se agrupan por el id del usuario `userid` y se ordenan cronológicamente.

Los datos se cargan desde un archivo CSV que contiene las características de los productos (`emb.csv`) y un archivo JSON que contiene secuencias de interacciones de usuarios (`seq.json`). Cada secuencia representa un historial de productos con los que un usuario ha interactuado.

ajustar

Las redes neuronales recurrentes necesitan que las entradas tengan la misma dimensión. Dado que las secuencias de interacciones de los usuarios pueden variar en longitud, se aplica un proceso de *padding* para normalizar estas secuencias a una longitud fija. Esto asegura que todas las secuencias tengan la misma longitud antes de ser procesadas por la red neuronal.

agregar el proceso de padding de agregar 0s

A partir de las secuencias, se generan los conjuntos de entrenamiento y prueba. Cada secuencia X en el conjunto de datos contiene 3 elementos, los dos primeros se entran a la red, y el tercero sería la salida ideal (predicción).

4.3. Construcción del Modelo

- **Red Neuronal Recursiva:** Se utiliza una arquitectura basada en LSTM, que incluye una capa de *embedding* para convertir los identificadores de productos en vectores densos de tamaño 128. Esto permite que el modelo capture las relaciones entre productos de manera más efectiva.
- **Capas LSTM y Dropout:** La red incluye dos capas LSTM con 256 unidades cada una. La primera capa LSTM devuelve la secuencia completa de salidas mientras que la última capa LSTM devuelve solo el último estado oculto. Después de cada capa LSTM, se aplica un *dropout* con una tasa de 0.3 para reducir el riesgo de sobreajuste.
- **Capa de Salida:** La capa final es una capa densa con activación *softmax*, que genera una distribución de probabilidad sobre los 112590 productos. Esto permite predecir cuál será el siguiente producto en la secuencia de interacción del usuario.

4.4. Entrenamiento y Evaluación del Modelo

- **Entrenamiento:** El modelo se entrena utilizando el conjunto de datos de entrenamiento, ajustando los pesos del modelo para minimizar la pérdida categórica usando *sparse_categorical_crossentropy*. La métrica utilizada es *accuracy*.
- **Evaluación:** Tras el entrenamiento, el modelo se evalúa en un conjunto de prueba independiente para medir su capacidad de generalización. Las métricas de rendimiento incluyen la pérdida y la precisión (*accuracy*).

5. Comparación entre la Solución Oficial y la Solución Propuesta

A continuación, se presenta una comparación entre la solución oficial basada en LSTM y la nueva solución propuesta que utiliza GRU:

5.1. Arquitectura y Complejidad

- **Solución Oficial (LSTM):** La solución oficial utiliza una red neuronal con dos capas LSTM de 256 unidades cada una, seguidas de capas de dropout para prevenir el sobreajuste. Las LSTM son conocidas por su capacidad para retener información a largo plazo, pero a costa de una mayor complejidad computacional y tiempo de entrenamiento.
- **Solución Propuesta (GRU):** La solución propuesta utiliza una única capa GRU con 512 unidades. Las GRU son una alternativa más ligera a las LSTM, con una estructura más simple que puede ofrecer un rendimiento comparable en menor tiempo y con menor uso de recursos computacionales.

5.2. Funcionamiento y Rendimiento

- **Solución Oficial (LSTM):** Esta solución está optimizada para capturar relaciones complejas en secuencias largas, siendo adecuada para casos donde la dependencia a largo plazo es crítica. Sin embargo, esto se traduce en un mayor tiempo de entrenamiento y una mayor necesidad de ajuste de hiperparámetros.
- **Solución Propuesta (GRU):** La solución con GRU, al ser más simple, tiende a entrenar más rápido y es menos propensa a sobreajustar los datos en comparación con las LSTM. Aunque puede no capturar dependencias tan largas como las LSTM, en muchos casos proporciona un rendimiento similar con menor costo computacional.

5.3. Resultados Empíricos

- **Solución Oficial (LSTM):** En las pruebas realizadas, la solución con LSTM mostró una alta precisión en la predicción de los productos, especialmente en secuencias largas, aunque a costa de un tiempo de entrenamiento mayor.

- **Solución Propuesta (GRU):** La solución GRU alcanzó resultados similares en términos de precisión, pero con un tiempo de entrenamiento menor y un uso de memoria más eficiente, lo que puede ser ventajoso en entornos de producción con recursos limitados.

6. Conclusión

La comparación entre la solución basada en LSTM y la solución propuesta con GRU destaca que ambas tienen sus fortalezas y debilidades. La solución con LSTM es más robusta para capturar dependencias a largo plazo, pero a un mayor costo computacional. Por otro lado, la solución con GRU es más eficiente y rápida, haciendo un buen compromiso entre rendimiento y consumo de recursos. La elección entre una y otra dependerá de las necesidades específicas del entorno de producción y de los recursos disponibles.

Referencias

1. Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas y Domonkos Tikk. *Session-based Recommendations with Recurrent Neural Networks.* Proceedings of the 4th International Conference on Learning Representations (ICLR 2016). Recuperado de `https://arxiv.org/abs/1511.06939`
2. Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep Learning.* MIT Press, 2016. Disponible en `https://www.deeplearningbook.org/`
3. Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho y Yoshua Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.* arXiv preprint arXiv:1412.3555, 2014. Recuperado de `https://arxiv.org/abs/1412.3555`