

Sistema de Recomendación Secuencial usando Redes Neuronales Recursivas

Jackson Vera Pineda
Kevin Manzano Rodríguez
Roger Fuentes Rodríguez

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

Abstract. Este informe describe la implementación de un sistema de recomendación secuencial basado en redes neuronales recurrentes (RNN) utilizando capas LSTM (Long Short-Term Memory). Este sistema tiene como objetivo predecir el siguiente ítem en el que un usuario podría estar interesado, basándose en sus interacciones anteriores. La capacidad de capturar dependencias temporales y la evolución de las preferencias del usuario a lo largo del tiempo son factores clave para mejorar la experiencia del usuario en plataformas de comercio electrónico y contenido. Además, se presenta una comparación entre la solución oficial basada en LSTM y una nueva solución propuesta que utiliza GRU (Gated Recurrent Units).

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 ítem 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 Descripción de la Solución

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

2.1 Preparación de Datos

- **Carga y Preprocesamiento:** Los datos se cargan desde un archivo CSV que contiene las características de los ítems (`emb.csv`) y un archivo JSON que contiene secuencias de interacciones de usuarios (`seq.json`). Cada secuencia representa un historial de ítems con los que un usuario ha interactuado.
- **Padding de Secuencias:** 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.
- **Generación de Conjuntos de Datos:** A partir de las secuencias, se generan los conjuntos de entrenamiento y prueba. Cada entrada X en el conjunto de datos contiene los dos primeros ítems de una secuencia, y la etiqueta y corresponde al tercer ítem de la secuencia, el cual se desea predecir.

2.2 Construcción del Modelo

- **Red Neuronal Recursiva:** Se utiliza una arquitectura basada en LSTM, que incluye una capa de *embedding* para convertir los identificadores de ítems en vectores densos de tamaño 128. Esto permite que el modelo capture las relaciones entre ítems 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 ítems. Esto permite predecir cuál será el siguiente ítem en la secuencia de interacción del usuario.

2.3 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*).

3 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:

3.1 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.

3.2 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.

3.3 Resultados Empíricos

- **Solución Oficial (LSTM):** En las pruebas realizadas, la solución con LSTM mostró una alta precisión en la predicción de los ítems, 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.

4 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.