Deep RL Recommender System

Elisa Chen, Beibei Du, Aditya John, Medha Sreenivasan

May 4, 2023

Deep Reinforcement Learning Recommender System

- A recommender system based on deep reinforcement learning (DRL).
- Uses a DRL algorithm to learn and provide personalized product recommendations.

Motivation and Goal

Motivation: The e-commerce industry has grown rapidly, leading to an overwhelming number of choices for consumers. To enhance user experience and increase customer satisfaction, personalized product recommendations have become crucial. There is a growing demand for innovative and effective recommender systems that can provide accurate recommendations in real time.

Goal: This project aims to explore session-based and sequential recommendation techniques for e-commerce use cases. By experimenting with different methods, including Deep Reinforcement Learning (DRL), we strive to improve the performance of recommendation systems and contribute to an enhanced user experience. We are comparing the performance of a GRU model to an hRNN model that also accounts for item features.

Data

We used two E-commerce datasets for our project:

- **Retail Rocket Dataset**: Collected from a real-world e-commerce website and consisted of raw data without any content transformation. Only the behavior data (events.csv) and item features (item_features_x.csv) were used.
- **H&M Dataset**: Contains information about transactions (transactions_train.csv) and item features (articles.csv). We used a subset of the entire dataset from August 2020 - October 2020 due to its large size.

Exploratory Data Analysis

Dataset 1 - Retail Rocket Dataset

- Category, events, and item properties datasets available for analysis.

- Events dataset contains information on views, adds to cart, and transactions.

- Analysis of events across time periods can provide insights into user behavior and preferences.

Dataset 2 - H&M Dataset

- Summary statistics and shape of dataset displayed.

- Preprocessing steps include data cleaning and sampling for training and testing.

- Popularity of each item is calculated using the 'pop.py' script.

Methodology

Feature Selection and One-Hot Encoding of Item Features

- Feature selection performed on top 500/600 most frequent properties for Retail Rocket/HM dataset.

- One-hot encoding is used to create an item feature matrix.

- Relevant columns filtered and saved as feature matrix dataset.

Re-defined Loss Function

- Item feature matrix passed through the feed-forward layer to create item embedding vectors.

- Modified loss function accounts for item embedding vectors.

- Details of implementation found in SNQN$_R R_F eatureVec.pyfile$.

Retail Rocket and H&M Results

Retail Rocket:

Model	HR@5		HR@10		HR@15		H
	Clicks	Purchase	Clicks	Purchase	Clicks	Purchase	Clicks
GRU	0.1925	0.3973	0.2437	0.4547	0.2721	0.4880	0.2920
hRNN	0.2237	0.4596	0.2672	0.5090	0.2929	0.5371	0.3118

H&M:

Model	HR@5		HR@10		HR@15		H
	Clicks	Purchase	Clicks	Purchase	Clicks	Purchase	Clicks
GRU	0.0175	0.0301	0.0245	0.0451	0.0305	0.0575	0.0360
hRNN	0.0174	0.0341	0.0317	0.0535	0.0405	0.0667	0.0485

Summary: - hRNN performs consistently better than GRU for clicks and purchases.

- Including item features leads to an 11-12% improvement in clicks and purchases for Retail Rocket.

- Similar improvements were observed for H&M.

- H&M results are likely lower due to the smaller data size.

Future Research

- **Hyperparameter Tuning Using Random Search**: The key hyperparameter values impacting the model output are learning rate, epoch number, and lambda. We are using 0.005, 5, and 0.15 for the three key hyperparameters respectively for our model training. Alternatively, hyperparameter tuning methods such as random search could be used to optimize the hyperparameter values in the future better.
- **Alternative Evaluation Metric**: We currently use HR and NDCG to evaluate the model. However, there are other metrics such as MRR and MAP that could also be used to evaluate the model performance. Using alternative metrics could help us gain a more nuanced and encompassing understanding of the model performance.
- **Increase capacity of computational resources**: More GPUs and storage could improve model performance. Training on more transactional data can also lead to better results.

GitHub Repository

Repository

https://github.com/belladu0201/Deep_RL_Recommender_System