IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 Journal UGC CARE Listed (Group-I) Volume 10, Issue 6 May 2021

A Comprehensive Study on Sequence-Aware Recommender Systems Using Deep Learning.

Dr. Diwakar Ramanuj Tripathi [1], Dr. Vrushali Pramod Parkhi [2]

[1] Head, Department of Computer Science, S.S. Maniar College of Computer & Management, Nagpur, India

[2] Officiating Principal, S.S. Maniar College of Computer & Management, Nagpur, India

Abstract

Various challenges, such as recommendation systems, have been tackled using deep learning, a subset of machine learning. In a sequential recommendation system, neural networks are employed to model the temporal dynamics of user activity. These systems leverage deep learning techniques to consider the context of past interactions and the time intervals between events, aiming to provide more accurate and personalized recommendations.

Recurrent Neural Networks (RNNs) are frequently utilized in sequential recommendation due to their capability to capture sequential patterns and depict dependencies between items, enabling the prediction of future behavior. Another prevalent architecture for managing sequential data is Long Short-Term Memory (LSTM) Networks. Deep learning offers the advantage of effectively handling large-scale, high-dimensional data, enabling the learning of intricate, nonlinear representations.

Deep neural networks also facilitate the representation of dynamic behaviors and allow for real-time adjustments in systems. Nonetheless, despite these advantages, deep learning-based sequential recommendation systems encounter significant challenges. One such challenge is the substantial data and processing resources often required, making them less suitable for small datasets. Additionally, the interpretability of these models poses another hurdle, as the complexity of deep learning models can hinder their interpretation and understanding, potentially impacting certain applications.

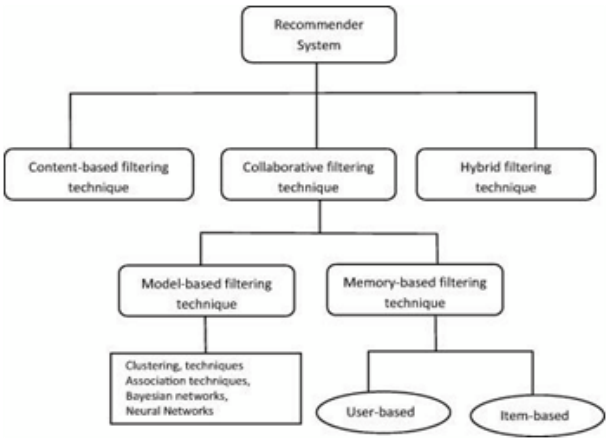
Keywords: Sequence-Aware, Deep Learning, Sequential Recommendation, Personalized Recommendations, Neural Networks, User Behavior Modeling, Sequence Modeling, Attention Mechanisms, RNN, LSTM

A kind of recommendation system called a sequential recommendation system with deep learning makes use of a neural network to simulate the temporal dynamic of user activity. By taking into account the context of prior encounters and the interval between events, these systems seek to offer recommendations that are more precise and individualised.

Due to its capacity to learn intricate and non-linear representations of the input, deep learning has gained popularity in recommendation systems recently. The Recurrent Neural Networks (RNN), which are able to represent the dependencies between items and predict future behaviour by capturing sequential patterns, are one of the most often utilised deep learning architectures in sequential recommendation. Long Short-Term Memory (LSTM) Networks are a popular architecture that can handle sequential input and avoid the vanishing gradient problem, which is a typical difficulty in conventional RNNs.

Many different applications, including e-commerce, music streaming, and online video platforms, use sequential recommendation systems. These systems can offer more precise and individualised recommendations by taking previous interactions in to account, which can improve user experience and engagement. Deep learning is also used in these systems, which enhances their capacity to handle large-scale, high-dimensional data and allows for real-time modifications.

Deep learning has benefits for sequential recommendation systems, but there are still some issues to be solved. The requirement for significant volumes of data and computer resources is one of the major difficulties. In some applications, the interpretability of these models might also be a barrier because deep learning models are frequently challenging to interpret and understand.



RELATED WORK

Recommendation System with RNN

According to research, Recurrent Neural Networks (RNN) are considered highly effective for sequence- aware recommendation. Deep recurrent neural networks leverage multiple levels of representation, akin to those successful in deep networks, along with the adaptable utilization of long-range context, thereby enhancing the capabilities of RNNs.

Problems: Vanishing gradient.

No long-term memory.

A sequential recommendation system based on Recurrent Neural Networks (RNN) is a type of recommendation system that utilizes RNNs to model the temporal dynamics of user behaviour. The main idea behind using RNNs in sequential recommendation is to capture the dependencies between items and predict future behaviour by learning sequential patterns in the data.

Recurrent Neural Networks (RNNs) possess the capability to process sequential data while maintaining an internal state that evolves with each time step. This feature enables them to consider the contextual information from past interactions, making them suitable for recommendation systems. For instance, in a music streaming service, an RNN-based recommendation system can leverage a user's listening history to suggest similar songs that align with their preferences.

One prevalent strategy in employing RNNs for sequential recommendation involves using them as a scoring mechanism. This entails assigning scores to items based on a user's previous engagements, with the highest- scoring items being recommended. Alternatively, RNNs can be utilized as generative models, producing a sequence of items predicted to attract user interaction. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are popular variations of RNNs in sequential recommendation systems, engineered to mitigate challenges like the vanishing gradient problem.

Despite their benefits, leveraging RNNs in sequential recommendation poses certain challenges. One notable hurdle is handling the vast amounts of high-dimensional data, demanding substantial computational resources. Moreover, RNNs' performance can be sensitive to hyperparameter selection, necessitating meticulous tuning efforts to achieve optimal results.

In conclusion, RNNs are a powerful tool for sequential recommendation systems as they are able to capture the dependencies between item sand predict future behaviour by learning sequential patterns in the data. However, they require significant computational resources and can be sensitive to hyperparameter selection, and further research is needed to address these challenges. An RNN-based recommendation system typically consists of several main components: an encoder, a recurrent layer, and a decoder. The encoder takes in the input data, which can include information about the user, the item, and the context of the interaction, and converts it into a hidden representation that can be understood by the recurrent layer.

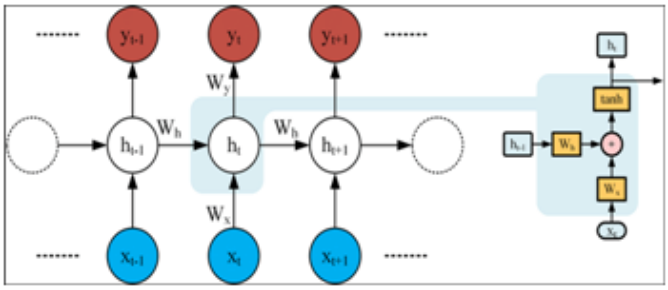
The recurrent layer, typically comprising a Long-Short Term Memory (LSTM) or Gated Recurrent Unit (GRU) network, is responsible for processing the hidden representation and updating its internal state, thus capturing sequential patterns in the data. Subsequently, the decoder utilizes the output from the recurrent layer to generate the final recommendations. The essence of employing RNNs in recommendation systems lies in their capability to grasp temporal dependencies among items, facilitating the prediction of future behavior. To achieve this, past interactions serve as input to the RNN, enabling it to discern patterns from the data, which are then stored within its internal state. As new data is introduced, the RNN leverages this stored information to make predictions regarding the items the user is likely to interact with next.

To calculate output of any given point h.

One of the significant advantages of employing Markov Chains in recommendation systems is their ability to incorporate the temporal aspect of user interactions, which other methods such as Collaborative Filtering might overlook. Additionally, they excel at handling sparse data and addressing the cold start problem.

It's worth noting that while Markov Chains are effective, they're just one of several methods used in recommendation systems. Collaborative Filtering, Hybrid methods, and Content-based filtering are among the alternatives. Moreover, the choice of method heavily depends on the dataset and the specific problem being addressed.

$$Z\ <t> = [X * W\ + h* w\ ] + b$$



RNN with a single hidden layer

Markov Chain

Markov Chain can be used in recommendation systems by modeling user behaviour as a sequence of states. Each state represents an item that a user has interacted with, such as viewing a product, adding it to a cart, or purchasing it. The transition probabilities between states represent the likelihood that a user will transition from one item to another. To use a Markov Chain in a recommendation system, you would first create a matrix that represents the transition probabilities between different items in your dataset. This matrix can be created by analyzing user interactions with the items.

After creating the matrix, it becomes a powerful tool for generating user recommendations through simulating a random walk. This entails starting at the current item a user is engaged with and leveraging transition probabilities to determine the next item they're likely to interact with. The items most frequently visited during this random walk are then recommended to the user.

Collaborative filtering is a widely used technique in recommendation systems aimed at predicting a user's preference for an item by analyzing the preferences of other users. This method relies on a users-items matrix and has gained significant traction due to its success in recent years. It can be categorized as either memory- based or model-based.

There are two primary types of collaborative filtering: user-based and item-based. In user-based collaborative filtering, recommendations are made based on the preferences of users who exhibit similar tastes. The system identifies users with similar past ratings and suggests items liked by those users to the active user.

On the other hand, item-based collaborative filtering recommends items similar to those the active user has previously enjoyed. It identifies items that bear resemblance to the ones liked by the user in the past and suggests them.

Both approaches rely on identifying patterns within the data to predict the preferences of the active user. While collaborative filtering algorithms are effective in making recommendations, they face challenges such as the cold start problem. This occurs when new users or items lack sufficient data for accurate recommendations to be generated.

Model-Based collaborative filtering

Collaborative filtering, specifically model-based, stands out as a popular technique within recommendation systems. This method tailors recommendations to individual users by leveraging a model, such as matrix factorization, to delve into the intricate patterns and connections within the data. Through training on historical user interactions—like

preferences, views, or purchases—the model learns to anticipate a user's interests by examining the behaviors of other users who

Its effectiveness shines in scenarios where vast datasets are available, as it excels in uncovering concealed correlations and trends within the data. Moreover, this approach remains dynamic, continuously updating the model as fresh data streams in, thereby enhancing recommendation accuracy over time. Industries spanning e-commerce, media, and social networking have embraced model-based collaborative filtering to enhance user satisfaction and foster engagement.

User-Based collaborative filtering

"In recommendation systems, user-based collaborative filtering is a widely employed technique aimed at providing personalized suggestions to users based on their shared preferences. This methodology operates by recommending products that similar users have either liked or purchased, drawing comparisons between the preferences of one user and those of others. The underlying principle of user-based collaborative filtering lies in the assumption that individuals who exhibit similar preferences for certain products are likely to share comparable interests in other items as well.

The process typically begins by analyzing the historical behavior or preferences of the target user to identify a cohort of users with similar tastes. Subsequently, the system examines the products that this group of comparable users has previously favored or purchased, leveraging this information to recommend items that the target user may not have encountered yet but are likely to enjoy based on the preferences of their comparable peers."

User-based collaborative filtering does have some drawbacks, though. One drawback is that it has a limited range of suggestions for novel or uncommon goods. It can be unable to provide recommendations for products outside of the user's regular experience because it is dependent on past behaviour and preferences. Additionally, it may be impacted by the "cold start" issue, which occurs when making recommendations for new users without a history of preferences.

The "sparsity" problem, which makes it challenging to identify comparable users when there are a lot of users or objects, is another restriction that might impair user- based collaborative filtering. User-based collaborative filtering may also be impacted by the "scalability" issue, which arises when there are a lot of users or objects and it becomes challenging to compute recommendations in real-time.

Supervised Learning

In recommendation systems, user-based collaborative filtering is a technique utilized to offer suggestions to users based on their shared preferences. This method recommends products that similar users have shown interest in or purchased, by comparing one user's preferences to those of others. The core principle of user-based collaborative filtering posits that individuals with similar product preferences will likely have comparable preferences for other items.

By analyzing users' past behaviors or preferences, this approach initially identifies a group of users who are similar to the target user. Subsequently, it examines the products that

these comparable users have already favored or purchased, before recommending items that the target user has not yet encountered but that comparable users have enjoyed.

Self-Supervised Learning

A potent method that can be utilized to enhance the effectiveness of recommendation systems is self- supervised learning. Without the requirement for explicit labelling, it enables the model to learn from user-interacted data, such as clicks, purchases, or ratings. This can be achieved by either utilizing supervised learning techniques like clustering to group similar individuals or things or by training the model to predict missing or distorted data from the interactions.

This can aid the model's ability to understand user habits and preferences and generate more accurate recommendations. In order to obtain even higher performance, self-supervised learning can also be employed in conjunction with other strategies like supervised learning or reinforcement learning. Overall, self-supervised learning is a viable method for enhancing recommendation system effectiveness and offering customers customized recommendations.

PROBLEM DEFINITION

A recommendation system using RNNs can be limited in their ability to handle large and complex data sets, and may not be as effective in incorporating feedback or multiple objectives into the learning process.

PROPOSED SYSTEM

Deep Reinforcement Learning

Deep reinforcement learning (RL) offers a promising avenue for boosting the effectiveness of recommendation systems by teaching models to make decisions based on a reward signal. The primary goal of recommendation systems is to deliver personalized suggestions that are likely to be clicked on or engaged with by consumers. This can be achieved by using the quantity of clicks or interactions on recommended items as a reward signal.

Through a process of suggesting items and receiving feedback in the form of rewards, the model learns to refine its recommendations. It adjusts its suggestions based on its understanding of which actions are more likely to yield higher rewards. This adaptive approach is particularly valuable in dynamic contexts where user preferences may evolve over time.

To further enhance the performance of recommendation systems, deep RL can be complemented with other techniques such as supervised learning or self-supervised learning. One of the key advantages of employing RL in recommendation systems is its ability to handle sparse and non-stationary data. This is particularly useful in scenarios where some users have limited interactions or certain items receive very few clicks. Additionally, RL enables the system to continuously learn from user interactions and adapt to changing preferences, thereby improving its overall effectiveness.

To learn successfully, RL-based recommendation systems need a lot of data and can be computationally expensive. To guarantee that the model is producing suggestions that are consistent with the general objectives of the recommendation system, the reward function must also be carefully chosen.

Overall, deep RL is a promising method for enhancing the effectiveness of recommendation systems and giving customers individualized recommendations, but it needs careful planning and execution to be effective.

In the proposed model, the DRL model helps The recommendation system to provide a more efficient and productive way of recommendation with two types of approach in DRL.

1) Model-Based Deep Reinforcement learning.

2) Model-Free Deep Reinforcement learning.

1) Model-Based Deep Reinforcement learning.:

This method integrates the advantages of model-based

reinforcement learning (RL) with deep neural networks. In model-based RL, the system develops the ability to forecast the consequences of various actions within the environment. This enables the agent to strategize ahead and enhance the quality of its decisions.

Model-Free Deep Reinforcement learning:

Model-free deep reinforcement learning (RL) is a method that leverages the advantages of model-free RL along with deep neural networks. Unlike traditional RL approaches that aim to learn an explicit model of the environment, model-free RL enables agents to make decisions solely based on the rewards they receive, without the need to explicitly model the environment.

CONCLUSION

The proposed system helps in the recommendation system due to its ability to incorporate a notion of reward or feedback, allowing the system to learn from its own performance and continually improve over time. Additionally, DRL can handle large and complex data sets, and can be used to optimize multiple objectives simultaneously.

Works Cited

1. Quadrana, Massimo, Paolo Cremonesi, and Dietmar Jannach. “Sequence-aware recommender systems.” ACM Computing Surveys (CSUR) 51, no. 4 (2018): 1-36.
2. Livne, Amit, Moshe Unger, Bracha Shapira, and Lior Rokach. “Deep context-aware recommender system utilizing sequential latent context.” arXiv preprint arXiv:1909.03999 (2019).
3. Said, Alan, and Alejandro Bellogín. “Comparative recommender system evaluation: benchmarking recommendation frameworks.” In Proceedings of the 8th ACM Conference on Recommender systems, pp.129-136. 2014.
4. Balderas, David, Pedro Ponce, and Arturo Molina. “Convolutional long shor term memory deep neural networks for image sequence prediction.” Expert Systems with Applications 122 (2019): 152-162.
5. Wang, Shoujin, Longbing Cao, Yan Wang, Quan Z.

6. Sheng, Mehmet A. Orgun, and Defu Lian. "A survey on session-based recommender systems." ACM Computing Surveys (CSUR) 54, no. 7 (2021): 1-38.
7. Xie, Xu, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. "Contrastive learning for sequential recommendation." In 2022 IEEE 38th international conference on data engineering (ICDE), pp. 1259-1273. IEEE, 2022.
8. Ahmed, Ahmed Adam, and Naomie Salim. "Markov Chain Recommendation System (MCRS)." Int. J. Nov. Res. Comput. Sci. Softw. Eng 3 (2016): 11-26.
9. Afsar, M. Mehdi, Trafford Crump, and Behrouz Far. "Reinforcement learning based recommender systems: A survey." ACM Computing Surveys 55, no. 7 (2022):1-38.
10. Yakhchi, Shahpar, Amin Beheshti, Seyed-Mohssen Ghafari, Mehmet A. Orgun, and Guanfeng Liu. "Towards a deep attention-based sequential recommender system." IEEE Access 8 (2020): 178073-178084.
11. Carroll, Micah D., Anca Dragan, Stuart Russell, and Dylan Hadfield-Menell. "Estimating and penalizing induced preference shifts in recommender systems." In International Conference on Machine Learning, pp. 2686-2708. PMLR, 2022.
12. Li, Seth Siyuan, and Elena Karahanna. "Online recommendation systems in a B2C E-commerce context: a review and future directions." Journal of the association for information systems 16, no. 2 (2015): 2.