

UNDERGRADUATE PROJECT REPORT

|  |  |
| --- | --- |
| **Project Title:** | **Convolutional Generative Network for Next Item Recommendation** |
| **Surname:** | **Long** |
| **First Name:** | **Yicheng** |
| **Student Number:** | **202018010116** |
| **Supervisor Name:** | **Joojo Walker** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **May 6, 2024** |

**Chengdu University of Technology Oxford Brookes College**

**Chengdu University of Technology**

**BSc (Single Honours) Degree Project**

Programme Name: **Computer Science**

Module No.: **CHC 6096**

Surname: ……………………………………LONG………………………………………………………

FirstName: ……………………………………YICHENG………………………………………………

Project Title: ……Convolutional Generative Network for Next Item Recommendation……………

Student No.: …………………………………202018010116……………………………………………

Supervisor:……………..……………………Dr. Joojo Walker…………………………………………

2ND Supervisor (if applicable): **Not Applicable**

Date submitted: **May 5, 2024**

*A report submitted as part of the requirements for the degree of BSc (Hons) in Computer Science*

*At*

**Chengdu University of Technology Oxford Brookes College**

**Declaration**

**Student Conduct Regulations**:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

<https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/>

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at <https://www.brookes.ac.uk/students/sirt/student-conduct/>

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

2024-05-03 10:28:25.174000**I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them**.

Signature ……………………………………… Date ……………May 5th, 2024………………

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

2024-05-03 10:28:25.174000**I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.**

Signature ……………………………………… Date ……………May 5th, 2024………………

# **Acknowledgment**

I am extremely grateful for the support and assistance from my university, my supervisor, my parents, and many others, which enabled me to successfully complete my final report. I would like to express my sincerest thanks and respect to them.

First and foremost, I would like to express my heartfelt gratitude to Oxford Brookes University for providing me with an environment conducive to learning and personal growth. Here, I not only acquired professional knowledge but also enriched my personal development. I will always cherish every teacher, student, and staff member here and appreciate their guidance and assistance during this journey.

Secondly, I would like to thank my supervisor, Dr. Joojo Walker, for his patient guidance, strict requirements, and continuous encouragement throughout my project. His guidance and suggestions on topic selection, research, writing, and other aspects were invaluable and led me to achieve excellent results. His words and actions will inspire me to be more diligent and enterprising in my future studies and work.

Additionally, I want to thank my friends who patiently answered my questions and discussed new problems with me when I encountered difficulties. Our exchanges helped me progress faster.

Lastly, I want to thank my parents, who have always supported me when facing challenges. Their dedication and hard work have led to my achievements and honor today. Their understanding and support are the source of my motivation, and I thank my parents once again for their selfless dedication.

Once again, I want to thank everyone who supported and encouraged me during this journey. Without their valuable contributions, this project would not have been possible.

# **Table of Contents**

[**Acknowledgment** iii](#_Toc165625861)

[**Table of Contents** iv](#_Toc165625862)

[**Abstract** vi](#_Toc165625863)

[**Abbreviations** vii](#_Toc165625864)

[**Glossary** viii](#_Toc165625865)

[**Chapter 1 Introduction** 1](#_Toc165625866)

[**1.1** **Background** 1](#_Toc165625867)

[**1.2** **Aim** 2](#_Toc165625868)

[**1.3** **Objectives** 2](#_Toc165625869)

[**1.4** **Project Overview** 3](#_Toc165625870)

[**1.4.1** **Scope** 3](#_Toc165625871)

[**1.4.2** **Audience** 3](#_Toc165625872)

[**Chapter 2 Background Review** 3](#_Toc165625873)

[**Chapter 3 Methodology** 5](#_Toc165625874)

[**3.1** **Approach** 5](#_Toc165625875)

[**3.1.1** **A Convolutional Generation Model for Item Recommendation Based on Conversation** 5](#_Toc165625876)

[**3.1.2** **Calculation of the probability of the distribution of items in an interaction sequence** 5](#_Toc165625877)

[**3.1.3** **Neural network structure of the model** 6](#_Toc165625878)

[**3.1.4** **One-dimensional transformations and reshaping operations** 8](#_Toc165625879)

[**3.2** **Masked Convolutional Residual Network Techniques** 9](#_Toc165625880)

[**3.3** **Dropout mask technique** 11](#_Toc165625881)

[**3.4** **The last layer of the network and item prediction** 12](#_Toc165625882)

[**3.5** **Technology** 12](#_Toc165625883)

[**3.6** **Project Version Management** 13](#_Toc165625884)

[**Chapter 4 Implementation and Results** 13](#_Toc165625885)

[**4.1** **Datasets and Experiment Setup** 13](#_Toc165625886)

[**4.2** **Evaluation Metrics** 14](#_Toc165625887)

[**4.3** **Hyperparameter Optimization** 16](#_Toc165625888)

[**4.4** **Parameter Settings** 16](#_Toc165625889)

[**4.5** **Performance Comparison** 17](#_Toc165625890)

[**Chapter 5 Professional Issues** 22](#_Toc165625891)

[**5.1** **Project Management** 23](#_Toc165625892)

[**5.1.1** **Activities** 23](#_Toc165625893)

[**5.1.2** **Schedule** 24](#_Toc165625894)

[**5.1.3** **Project Data Management** 24](#_Toc165625895)

[**5.1.4** **Project Deliverables** 25](#_Toc165625896)

[**5.2** **Risk Analysis** 25](#_Toc165625897)

[**5.3** **Professional Issues** 26](#_Toc165625898)

[**5.3.1 Legal issues** 26](#_Toc165625899)

[**5.3.2 Social issues** 26](#_Toc165625900)

[**5.3.3 Ethical issues** 27](#_Toc165625901)

[**5.3.4 Environmental issues** 27](#_Toc165625902)

[**Chapter 6 Conclusion** 28](#_Toc165625903)

[**6.1 Reflection and Conclusion** 28](#_Toc165625904)

[**6.2 Potential Future Work** 29](#_Toc165625905)

[**References** 30](#_Toc165625906)

[**Appendices** 31](#_Toc165625907)

# **Abstract**

With the growth of data and information, the demand for recommendation systems has increased significantly, making more precise recommendations a major research direction in the field. This paper first reviews the development history of recommendation systems and previous studies on sequence recommendation. However, traditional recommendation techniques still face many challenges. To further identify factors affecting recommendation effectiveness, this study re-implemented an existing next-item recommendation model based on convolutional generative networks: NextItNet. This pre train model has the capability to handle sequences of arbitrary lengths and introduces residual learning, making it a mature sequential recommendation model. The performance of this model was evaluated on five different types of datasets. Experimental results indicate that datasets with larger volumes and higher user interaction counts, yet lower sparsity, perform better with the NextItNet model. Specifically, the model achieved a Recall and Hit of 67.25% simultaneously on the Yoochoose-buy dataset. Additionally, it was found that appropriately increasing the number of recommendations also improves the model's recommendation effectiveness, which is applicable to both large and small datasets. The implementation of this model in this study further demonstrate its reliability.

**Keywords:** Convolutional Generative Network; Next-item recommendation; Residual learning

# **Abbreviations**

|  |  |
| --- | --- |
| Abbreviations | Definition |
| CNN | Convolutional Neural Network |
| RNN | Recurrent Neural Network |
| CGN  GPU  UNIX  HV-CNN  Word2Vec  SBR  FedRAP  NARM | Convolutional Generative Network  Graphics Processing Unit  Timestamp in seconds  Horizontal Vertical Convolutional Neural Network  Word to Vector  Sequencial Recommendation  Federated Recommendation with Additive Personalization  Neural Attention Recommendation Machine |

# **Glossary**

|  |  |
| --- | --- |
| Keywords | Definition |
| Convolutional Generative Network | Convolutional Generative Network (CGN) is a deep learning architecture designed to generate data and use convolutional layers to process spatial and temporal information. It is often used for image generation, sequence prediction, other tasks that require capturing complex spatial or temporal structures, and recommendation systems. |
| Next-item recommendation | Next-item recommendation refers to the recommendation system predicting the next item or action that the user may be interested in based on their historical behavior or contextual information. This recommendation method aims to provide users with personalized suggestions to meet their current needs or interests. The next recommendation is widely applied in fields such as e-commerce, social media, music, and video streaming. |
| Residual learning | Residual learning is a deep learning method aimed at training models by learning the residuals between input and target (i.e. the difference between predicted and actual values). This method simplifies network training by introducing skip connections and residual blocks, effectively solving the problems of gradient vanishing and exploding during deep neural network training, and improving the training speed and performance of the model. |
| NextItNet model | An existing next recommendation model based on convolutional generative networks proposed by Yuan et al. [1]. This model combines a simple convolutional generative network and introduces residual learning, which will be explained in detail in Chapter 3. |
| Internal channels | In the convolutional layer of a convolutional neural network, multiple convolutional kernels (or filters) are usually defined, each of which generates an output channel when performing the convolution operation. And the parameters within each convolutional kernel are shared in order to perform feature extraction on the input data. |
| Relu | ReLU is a commonly used activation function used for each neuron in artificial neural networks. Its mathematical form is f (x)=max (0, x), which means that when the input is less than 0, the output is 0; When the input is greater than or equal to 0, the output is equal to the input. The simple form and computational efficiency of ReLU make it one of the commonly used choices in deep learning, helping to accelerate the training process of the model and, in some cases, improving the performance of the model. |

# **Introduction**

## **Background**

With the rapid development of the Internet, information is experiencing exponential growth, leading individuals to encounter a plethora of data daily, including news updates and ubiquitous product advertisements [2]. However, much of this information is irrelevant to users, prompting the need to deliver desired content efficiently. As a response to this challenge, recommendation systems have emerged. Various recommendation systems employ different strategies, including collaborative filtering systems, which predict user preferences for unseen items based on historical interactions between users and items [3], content-based recommendation systems, which rely solely on item features and attributes without needing user-item interaction records [4], and deep learning recommendation systems, which leverage deep learning techniques to learn complex relationships between users and items from large-scale data [5].

In recent years, sequence recommendation within deep learning recommendation systems has garnered increasing attention due to its ability to infer dynamic user preferences based on sequential user interactions, along with its real-time and personalized advantages. Particularly in applications focusing on real-time or continuous user experiences, sequence recommendation can capture latent features of sequential data. For instance, Last.fm users often enjoy a series of songs and videos within a period, where these items exhibit certain correlations or common characteristics [6]. However, sequence data is inherently complex. To better capture data features, sequence recommendation systems enhance performance by incorporating deep learning techniques. For example, recurrent neural networks (RNNs) have shown significant effectiveness in recommendation systems [7], [8], [9], and following the inspiration drawn from the remarkable performance of convolutional neural networks (CNNs) in processing image data [10], [11], CNNs have also been widely applied in recommendation systems.

Nevertheless, unlike traditional neural networks, this project adopts the NextItNet, a convolutional generative network-based next-item recommendation model proposed by Yuan et al. [1]. This model possesses superior capabilities in capturing contextual information and handling long sequential texts compared to traditional deep learning techniques.

The remaining chapters of this report are organized as follows: Chapter 2 mainly reviews the history of CNN application in recommendation models, as well as relevant literature on sequential recommendation and previous research on the next recommendation system, including the main characteristics of each method. Chapter 3 introduces the core mechanism, computational principles, deployment of network structure, and related technical concepts of the NextItNet model of convolutional generative networks recommended for the next item. Chapter 4 presents the experimental results and comparative analysis. Chapter 5 discusses the management, professional issues, and risk issues of this project. Finally, Chapter 6 will discuss the limitations of this project and future research directions.

## **Aim**

This project is an analysis and research on the next recommendation system based on convolutional generative networks. The system introduces convolutional generative networks to enable the model to learn more complex user behavior and preference representations, and better capture the features of the data. This recommendation system can surpass existing recommendation systems and achieve better recommendation results. This article will implement this recommendation system and test it on various types of datasets, and compare and analyze the results to identify the key factors that affect the effectiveness of recommendations.

## **Objectives**

The objectives are depicted as follows:

1. Conduct comprehensive literature review on the next recommendation and sequence recommendation techniques, as well as the application of convolutional generative networks in recommendation systems.
2. Collect a dataset suitable for sequence recommendation, check its applicability, and preprocess it.
3. Implement a usable sequence recommendation model based on convolutional generative networks.
4. Train recommendation models on different types of datasets, validate and test their performance under different conditions.
5. Use different statistical metrics to evaluate and optimize recommendation models.
6. Discuss and summarize the performance of recommendation models on different datasets and metrics, analyze the impact of datasets and evaluation parameters on model recommendation performance.

## **Project Overview**

### **Scope**

The main purpose of this project is to use the next recommendation model (NextItNet) based on the convolutional generation network to achieve recommendations for users on different types of data, including movies, music, purchase records, games, clothing and other fields. The introduction of convolutional generative networks and residual learning has effectively made recommendations. Then, by comparing the test results of the model on 5 different types of datasets, the best performing dataset is identified. By adjusting the relevant parameters for testing, the relationship between the model and the dataset is analyzed, and the key factors affecting the recommendation effect are identified. In this way, the most suitable dataset type is found for the model to maximize its recommendation effect.

### **Audience**

Users and producers of products that require real-time recommendations will be one of the main beneficiaries, such as applications in music streaming, video on demand, e-commerce, online games, and other fields. For their product users, the Next Item Recommendation System will provide more personalized and real-time recommendations that users are interested in, making it easier for product users to find products, services, or information of interest, This saves time and effort, and product developers gain more users as a result. In addition, some companies or advertisers are the main beneficiaries. By increasing transaction volume, businesses can benefit, and advertisers can place more targeted advertisements to improve advertising efficiency. Technical researchers will also benefit from new technologies and methods.

# **Background Review**

The earliest work and ideas for sequence recommendation mainly relied on Markov chains [12] and feature based matrix decomposition [13] methods. Markov chains are a mathematical model in which the occurrence of an event only depends on the state of the previous event, and is independent of the earlier state. But it also has some shortcomings, especially when dealing with complex sequence data, its ability to model complex nonlinear relationships and patterns in sequence data is limited and lacks long-term memory. Afterwards, deep learning models gradually began to demonstrate advanced recommendation accuracy. In 2016, Hidasi et al. [14] proposed a deep learning based SBR system, commonly known as GRU4Rec. This is the first model to use RNN, which introduces session parallel small batch, output sampling based on small batch, and sorting loss function, resulting in significant results due to popular baselines. Afterward, Tan et al. optimized the GRU4Rec model by proposing two improvement methods: data augmentation and modifying the input data distribution [15]. This ultimately led to a significant improvement in the model's recall rate and Mean Reciprocal Rank. Additionally, Quadrana et al. introduced a hierarchical RNN model for cross-session information propagation [16]. This model utilizes a novel algorithm to handle two different types of sessions, with all metrics significantly outperforming traditional session-based RNN models. In 2018, Tang and Wang [17] proposed a new sequence recommendation called Caser. They abandoned the RNN structure and proposed a convolutional sequence embedding model, demonstrating that this CNN based recommendation can achieve similar or superior performance in the popular RNN model's top-N sequence recommendation. Not long after the same year, Yuan et al. [1] proposed a simple, efficient, and efficient convolutional generation model for session based top-N project recommendations. This model is suitable for short-term and long-term project dependencies and simplifies deeper network optimization. Ultimately, the model's recommendation accuracy and effectiveness are significantly better than existing technologies at the time. In 2021, Song et al. [18] designed an effective SBRS called Intersessional Collaborative Recommendation Network (Insert) to recommend the next project in short sessions, and designed a Session Retrieval Network (SSRN) to identify sessions similar to the current short session from the historical sessions of the current user and other users, resulting in better recommendation performance than the most advanced series of recommendations at the time. In the same year, Chai et al. [19] proposed a secure matrix decomposition framework under federated learning settings, called FedMF, which to some extent prevents users' raw data leakage and ensures user data privacy, but has not been applied in recommendation systems. In 2023, Li et al. [20] proposed Federated Recommendation with Additive Personalization (FedRAP), which introduced federated learning. This recommendation system effectively avoids user information leakage, reduces communication costs, and solves the problem of poor personalization in other federated recommendations. Afterwards, Kumar et al. [21] proposed a Horizontal Vertical Convolutional Neural Network (HV-CNN) embedded with Word2Vec technology, which outperformed state-of-the-art methods on 30 publicly available music datasets.

# **Methodology**

This section discusses the Recommendation model used in the project, introduces the probability calculation method and related mathematical concepts of the model, network structure, masked convolutional residual network technology, dropout mask technique, network generation, and product recommendation. In addition, methods and tools for managing and implementing the project were also introduced.

## **Approach**

## **3.1.1 A Convolutional Generation Model for Item Recommendation Based on Conversation**

To avoid the problem that traditional models perform poorly in modeling long-range dependencies in the item sequence, this paper uses the NextItNet model, a probabilistic generative model. It consists of a stack of one-dimensional convolutional layers and has a deeper structure that does not depend on pooling layers and introduces the technique of masked convolutional residual networks. The model is mainly applied to sequential recommendation. It makes recommendations for items or content that may be of interest to the user in the future by making recommendations based on the historical data of the sequence of interaction items.

## **3.1.2 Calculation of the probability of the distribution of items in an interaction sequence**

NextItNet model estimates and computes the distribution of the original item interaction sequences by directly accessing sequences of items that have been interacted with in order to reliably compute the probability of the items and to recommend future items that users may like to interact with. The model introduces a probability distribution p (x), which represents the joint distribution of item sequence x0 to xt. Through the chain rule in probability theory, the product of conditional probability density functions can be used to represent the joint probability density function of multiple random variables to compute the probability distribution of the item p(x), as shown in Formula 1 [1]. In the probability distribution formula, t denotes the length of the sequence x, i.e., the number of elements in the sequence. “p(xi |x0:i-1, θ )” denotes the conditional probability density function that generates the i-th element xi given the first i-1 elements and the model parameter θ. p(x0 ) denotes the probability that the first element x0 is generated without any prior information.

Regarding the approach of modeling conditional distributions of user-item interactions using neural networks, in contrast to traditional CNN sequential recommendation models such as Caser [17], GRURec [14], Neural Attention Recommendation Machine (NARM) [22] and Hierarchical RNN Model (HRNN) [16], the model used in this paper models all conditional probabilities instead of a single conditional distribution. In these traditional models, only the probability distribution of the next item (x15) is estimated (as shown in Figure 1), whereas the generation method of the model used in this paper estimates the probability distribution of all individual items from x1 to x15 of all individual items in the distribution. This is the key to capturing the relationship characteristics of various length types of sequences.

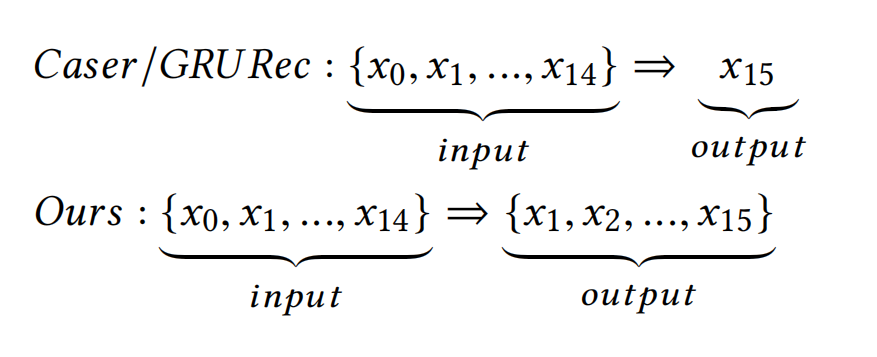


Figure 1: Conditional probability of an item (Yuan et al, 2018)

## **3.1.3 Neural network structure of the model**

Embedding the lookup layer. The role of this layer is to transform a given sequence of items (from x0 to xt) into an embedding matrix, which retrieves t items by a checklist through the model and stacks the embedding vectors of these items in order, if the embedding vectors have dimension 2k, where k is the number of internal channels which is the number of convolution kernels, the matrix is of size t × 2k. Unlike the traditional CNN recommendation model Caser, the filter uses a one-dimensional convolutional filter rather than treating the data as a two-dimensional image process.

Expansion Layer. Figure 2 (a) shows the standard filter, which has a size of 1\*3 and can only perform current convolution through a limited number of receptive fields and network depths. This is detrimental to the processing of random sequences. The model used in this paper uses dilation convolution, which is based on the principle of inserting 0 values at intervals in the convolution kernel, which is also known as the filter, as shown in Figure 2 (b), so that when the convolution operation is carried out, the elements in the convolution kernel are not applied sequentially according to the neighboring positions, but are applied at certain intervals. If the dilation rate is 1, then it is a standard convolution operation; if the dilation rate is 2, then a zero value is inserted between each element of the convolution kernel, causing the convolution kernel to dilate by one unit in the horizontal and vertical directions, respectively, and thus expanding the sense field. This operation allows the convolutional kernel to perceive information over a wider area, thus enabling a larger range of contextual information to be captured, which helps in tasks such as processing sequence data or image data with long correlations.

图片包含 图表

描述已自动生成

Figure 2: Conditional distribution prediction (Yuan et al, 2018)

Figure 2 shows a network comparison between standard and dilated convolution in a sequential generation model, where part (b) shows the effect of different dilation rates (1, 2, 4, 8). To describe the network structure, we denote the size of the convolutional kernel by the acceptance domain, the size of the feature map of the jth convolutional layer by Fj, the number of channels by C, and the number of convolutional layers by l. The convolutional filters are set to the size of the convolutional kernel. By setting the width f3 of the convolutional filter, we can see that the dilated convolution allows the size of the receptive domain to grow exponentially (r = 2^j+1-1), whereas the size of the receptive domain of the standard convolution grows only linearly (r = 2j + 1). The filtering window for position i is shown in formula (2) [1]. Formula (3) shows the new value of element h after applying the dilated convolution operation [1], where \*l is the one-dimensional dilated convolution operator, g is the filter function, f is the filter width, and g(i) denotes the value of the filter function at position i.

## **3.1.4 One-dimensional transformations and reshaping operations**

The network architecture of the NextItNet model used in this paper actually consists of all one-dimensional convolutional layers. This is achieved mainly through a reshaping operation: a 2D matrix of size t × 2k is reshaped into a 3D vector of size t × 2k., where 2k, unlike the traditional CNN recommendation model Caser, is represented as an image channel rather than a filter width. Figure 3 illustrates the conversion process from a 2D filter (C=1) (left) to a 1D 2-dilatation filter (C=2k) (right), with a default step size of 1 and black arrows showing the direction of filter sliding.

图片包含 白板

描述已自动生成

Figure 3: Reshaping operations (Yuan et al, 2018)

## **Masked Convolutional Residual Network Techniques**

The NextItNet model has a deeper network structure, and in order to solve the accompanying gradient descent problem, the model also introduces a residual learning technique. The authors of this model are also the first to refer to residual learning in the field of recommender systems. Residual learning is a technique used to train deep neural networks and the basic idea is by introducing residual connections or known as jump connections. Such connections allow information to flow through the network in a more direct manner. In a residual connection, the input signal is not only passed on to the subsequent layers for processing, but is also passed directly to the output layer through a jump connection, thus constituting a "shortcut path". Since the goal of the modeled network is to learn to map the residuals between inputs and outputs around zero. Therefore, the output of the network is modeled as the sum of the inputs and the learned residuals, rather than directly as the original target output. In this way, the network only needs to learn to convert the input signal to a residual signal, instead of having to learn the complex mapping relationship directly. Residual learning helps to alleviate the problems of gradient vanishing and gradient explosion, and also accelerates convergence and improves the generalization ability of the model. For the NextItNet model, which applies CNNs to recommender systems, residual learning can improve the recommendation effect to some extent. As shown in Figure 7 (a) and (b), the model introduces two residual modules. In Figure (a), each dilated convolutional layer is wrapped by one residual block, while in Figure (b), each dilated convolutional layer employs another different residual block. For Figure (b), the model connects the input layer and the second convolutional layer via jump connections (indicated by blue lines in Figure 2). Specifically, each residual block consists of a series of operations including normalization, activation function (ReLU), convolutional layer, and jump connections in a specific order. In addition to this, a layer normalization technique that is more suitable for sequence processing is used before each activation function; layer normalization is performed in a single hidden layer for each sample, rather than in the entire batch. This makes it more suitable for sequential data or online learning scenarios, as it does not require statistical computations over the entire batch. The formula for layer normalization is:

Where x is the input eigenvector, μ is the mean of the eigenvector σ is the standard deviation of the eigenvector, γ and β are the scaling factor and the translation factor of the learning, and ϵ is a very small number used to stabilize the computation from the case where the standard deviation is zero.

图示

描述已自动生成

Figure 4: Dilated residual blocks (Yuan et al, 2018)

Regarding the construction and parameter computation of the residual block, in the residual block (a), three convolutional filters are included: an expansion filter of size 1 × 3 and two standard filters of size 1 × 1. The 1 × 1 filters were introduced to vary the number of channels in order to reduce the number of parameters in the 1 × 3 convolution kernel. The first 1 × 1 filter (close to the input E) is used to reduce the number of channels from 2k to k, whereas the second 1 × 1 filter is used, in contrast, to maintain the spatial dimension of the next stacking operation. The effectiveness of the 1 × 1 filter in (a) can be verified by calculating the number of parameters in (a) and (b). under the condition that the activation and normalization layers are omitted. It can be found that the number of parameters of the 1 × 3 filter is 1 × 3 × 2k × 2k = 12k^2 without considering the 1 × 1 filter (i.e., in (b)). In contrast, in (a), the number of parameters to be learned is 1 × 1 × 2k × k + 1 × 3 × k × k + 1 × 1 × k × 2k = 7k^2. The residual mapping F(E, {Wi}) takes the form shown in (a) and (b) as follows [1]:

where σ and ψ denote the ReLU and the layer normalization, W1 and W3 denote the weight functions of the standard 1 × 1 convolution, and W2, W2', and W4' denote the weight functions of l-expansion convolution filters of size 1 × 3 and the bias term is omitted.

## **Dropout mask technique**

In terms of information security, it is important to prevent the network from "seeing" future information during prediction. Specifically, when we want to predict the probability of an item appearing, we do not want the convolutional layer to use information from future time steps, as shown in Figures 5 (b) and (c). They use information from future items, i.e. future time steps, when predicting "1" and "3". In the real world, we cannot obtain future information when making predictions, so we need a method to ensure that the network can only use current time steps and past information for prediction. The authors of the NextItNet model, Yuan et al. [1], explained that there are two methods to implement this dropout mask operation. One method is to add some additional information (padding) to the input sequence before performing the convolution operation, to ensure that the network cannot "see" future information. As shown in Figure 5 (d), when predicting "1", the items of {1, 2, 3, 4} are masked, and the actual padding items are used. Another method is to move future information back a few time steps in the output sequence, so that the network cannot use this future information for prediction. However, for shorter sequences, the method of moving the output sequence may result in information loss. Therefore, the NextItNet model uses a strategy of filling the input sequence and filling it with a specific size to ensure that future information is not used in project prediction.

图表

描述已自动生成

Figure 5: Dropout masks in padding way (Yuan et al, 2018)

## **The last layer of the network and item prediction**

Regarding the final layer of the network, the output should be a tensor or matrix that contains the probability distributions of all output sequences x1 to xt. Therefore, an additional convolutional layer with a filter size of 1 \* 1 \* 2k \* n is used on the last convolutional layer in Figure 2, where n is the number of items. The purpose of this convolutional layer is to convert the feature representation of the last layer into the probability distribution of each item. According to the one-dimensional transformation process in Figure 3, the expected output matrix Ep ∈ Rt × n was ultimately obtained. After softmax operation, each row vector represents the class distribution of xi (0<i ≤ t), that is, a probability distribution corresponding to the item was generated at each time step.

According to Yuan et al. [1], the NextItNet model only predicts the next project during the evaluation phase, but the model can continuously generate project sequences by feeding a predicted project (or sequence) into the network. This means that the predictions during the generation phase are sequential, which is also in line with real-world recommendation scenarios. In addition, during the training and evaluation stages, since a complete input sequence is already available, conditional predictions for all time steps can be performed in parallel.

## **Technology**

Hardware includes: computer (GPU: NVIDIA RTX2070S, RAM: 16G), software includes: deep learning framework: recbole, pytorch, TensorFlow, programming language: Python 3.9, editor: PyCharm3.3, data preprocessing tool: recbole.

The techniques used in this project are mainly based on the recbole framework tool and python. recbole is a recommender system framework that contains models and datasets for recommender systems, and by using the recbole framework tool it is easier to train and implement recommendation models.

In addition, a pytorch virtual environment was created for the project using python to facilitate efficient management of dependencies and libraries by the developers, and the recbole framework was required to run under the pytorch virtual environment as well. The training of the model was done on my own laptop using an RTX 2070-based GPU, and debugging regarding model parameters, training parameters, and evaluation parameters was done by writing configuration files.

## **Project Version Management**

Use the Git repository and Feishu to manage project code or multiple versions of models that have been developed. The code will continue to be updated on Github's personal homepage, and backup management will also be carried out every time it is submitted through Flybook to facilitate subsequent changes.

# **Implementation and Results**

In this section, we will describe the dataset and evaluation metrics, and then compare the performance of the model on various parameters in different datasets and analyze the reasons. In addition, methods were proposed to achieve maximum performance and improve recommendation accuracy.

## **Datasets and Experiment Setup**

In this paper, five different types of datasets are used to train the NextItNet model, namely: the Yoochoose-buys dataset from RecSys Challenge 2015, two different sizes of MovieLens datasets: MovieLens-100k and MovieLens-1m, the Amazon Video Games dataset from the Amazon website, and the public dataset from fashion company Rent the Runway, which is referred to as Rent the Runway in this article.

The yoochoose-buys dataset comprising the information of the buy events. Each record in the file has the following fields: session id: the unique identifier of the session, which is of type token; item id: unique identifier of the item, type token; count: number of items purchased in the session, type float; timestamp: timestamp of when the purchase occurred, type float. The number of users in this dataset is 509697, the average number of user behaviors is 2.16, the number of items (merchandise) is 19950, the number of records in the dataset is 1102955, and the dataset sparsity is 99.98%.

The MovieLens-100k dataset is a dataset about the interaction between users and movie ratings, derived from MovieLens. Each record in the dataset has the following field: user id: the id of the users and its type is token; Item id: The id of the movies and its type is token; Rating: The rating of the users over the movies, and its type is float; Timestamp: The UNIX timestamp of the rating, and its type is float. The number of users in this dataset is 944, the average number of user behaviors is 106.04, the number of items (products) is 1683, the number of records in the dataset is 100000, and the sparsity of the dataset is 93.705%.

The MovieLens-1m dataset is also a dataset about the interaction between users and movie ratings, used for comparison with the MovieLens-100k dataset. It comes from MovieLens, where each record in the dataset has the following field: user id: the id of the users and its type is token; Item id: The id of the movies and its type is token; Rating: The rating of the users over the movies, and its type is float; Timestamp: The UNIX timestamp of the rating, and its type is float. The number of users in this dataset is 6041, the average number of user behaviors is 165.59, the number of items (products) is 3707, the number of records in the dataset is 1000209, and the sparsity of the dataset is 95.53%.

The Amazon Video Games dataset is a dataset about the interaction between users and game ratings, sourced from the Amazon website. Each record in the dataset has the following field: user id: the id of the users and its type is token; Item id: The id of the items and its type is token; Rating: The rating of the users over the item, and its type is float; Timestamp: The UNIX time of the interaction, and its type is float. The number of users in this dataset is 1540619, with an average of 1.66 user actions, 71983 items (products), 2565349 records, and 99.99% sparsity.

The Rent the Runway is a dataset about the interaction between users and clothing ratings. Each record in the dataset has the following field: user id: the id of the users and its type is token; Item id: The id of the items and its type is token; Rating: The rating of the users over the item, and its type is float; Timestamp: The UNIX time of the interaction, and its type is float; Fit: The user's evaluation of the fitting degree of the clothing, with the field type being token; Rented for: The purpose of clothing, with a field type of token. For the 6 entries in this dataset, only the first 4 entries need to be used this time, and the data preprocessing operation is completed by writing a configuration file suitable for recall. The number of users in this dataset is 105572, with an average of 1.82 user actions and 5851 items (products). The number of records in the dataset is 192544, and the sparsity of the dataset is 99.96%.

## **Evaluation Metrics**

Recall: Recall is the ability of a model to find all relevant items. For sequential recommendation models, the recall rate indicates how many relevant items in the recommendation list have been successfully found and presented to users. In addition, the improvement of recall rate means that the model better captures user interests and provides more relevant recommendation items.

MRR (Mean Reciprocal Rank): MRR measures the effectiveness of a model in presenting relevant items for the first time in a recommendation list. For sequential recommendation models, MRR can tell us the average position where users find relevant items in the recommendation list. The increase in MRR means that the model places relevant items at the top of the recommendation list more quickly.

NDCG (Normalized Discounted Cumulative Gain): NDCG comprehensively considers the position and relevance of relevant items in the recommendation list. For sequential recommendation models, NDCG can measure the importance of relevant items at different positions in the recommendation list, and the increase of NDCG indicates that the model better optimizes the sorting of relevant items in the recommendation list.

Hit: Hit measures the proportion of items that a model can find that users like, without considering the ranking of the items. For sequential recommendation models, Hit can tell us how many items users like the model can find, without worrying about their position in the recommendation list. The increase in Hit indicates that the model can recommend items that users like more frequently.

Precision: Precision measures how many of the items recommended by the model are truly of interest to the user. For sequential recommendation models, accuracy can tell us how many items in the model's recommendation list are actually what the user wants.

The calculation formulas for the above indicators are:

Where TP represents the true case (the number of correctly recommended related items by the model), and FN represents the false negative case (the number of not recommended but actually related items). N represents the number of recommendations or queries, pi represents the position of the item that the user actually visited in the recommendation list. If it is not in the recommendation sequence, p is infinite, and 1/p is 0. Reli indicates whether the item in the i-th position is liked by the user. Likes it as 1, otherwise it is 0. REL represents the set of original recall sets R sorted in descending order of scores.

## **Hyperparameter Optimization**

## **Parameter Settings**

Dataset parameters: As the datasets of the sequential recommendation model are all text data or user item interaction sequences, (\t) is used as the separator between fields, and spaces are used as intervals. For the five datasets used in this article, four corresponding field names are selected to match the user ID field, item ID field, rating field, and time field. For example, for the yoochoose buy dataset, these four fields are session id, item id, count, and timestamp, respectively.

Training parameters: The dimension size of the embedding layer in the rectangle of the NextItNet model defaults to 64, which means that each item or user will be represented as a 64 dimensional vector. Set the training batch size to 2048 and use the Adam optimizer with a learning rate of 0.001. The size of the training batch is related to the training time. Considering the time relationship and hardware performance, setting it to 2048 is for faster training. For CNN recommendation models, large-scale training can usually improve training speed and model stability. The Adam optimizer combines the advantages of AdaGrad and RMSProp, and typically performs well for non convex optimization problems in recommendation systems. In addition, the number of negative samples is 0, and the negative sampling parameter in the CE loss function is null.

Evaluation parameters: In the evaluation settings, this article chooses to sort the data by time, uses the left one method to partition the dataset, and uses full sorting for evaluation. This setting can better simulate practical application scenarios and ensure the generalization ability of the model. The evaluation indicators include Recall, MRR, NDCG, Hit, and Precision, which can comprehensively evaluate the performance of the model in recommendation tasks. This article also sets the Top K values to 5, 10, 20, 50, and 100, which means comparing the performance of the model under different recommendation lengths. Finally, choose MRR@10 It is reasonable to use MRR as a criterion for stopping training early, as it can evaluate the ranking performance of the model under a given recommendation list length.

## **Performance Comparison**

This section will compare horizontally from the perspective of each indicator: under each indicator, the results of different datasets will be compared to observe the differences between them. And vertical comparison: comparing the performance of the same dataset under different indicators to observe its internal consistency and changes.

For metric recall@K, the Yoochoose-buys dataset performs the best, Recall@100 Reached 67.25%, followed by MovieLens-1m, then Amazon Video Games and MovieLens-100k, and finally the Rent The Runway, recall rate for 100 recommended items is only 12.15%. This may be because the Yoochoose-buys and MovieLens-1m datasets have a higher number of user behaviors and lower data sparsity, making it easier for the model to capture user preferences. In addition, for Recall@k, As k increases from 5 to 100, the recall of all five datasets increases. This is because more items are considered, giving the system more opportunities to find items that users may like. A larger recommendation list provides more selection space, thus having more opportunities to cover user interests.

MRR (Mean Relational Rank) is an indicator used to evaluate the ranking quality of recommended items, and a higher value indicates that the model is able to rank relevant items at the top. The Yoochoose-buys dataset is available in MRR@k The best performance is on, MRR@100 Reached 0.1739, while RentTheRunway performed the worst among all datasets, MRR@100 It is 0.0101. Yoochoose-buys MRR@k The higher value may be due to the higher user behavior in the dataset, making it easier for the model to find relevant items and rank them at the top. Similarly, as k increases from 5 to 100, the MRR indicators of all five datasets improve. This is because a larger k value means that more recommendations are taken into account, and the system may rank recommendations more accurately, making it more likely that relevant results will appear at the forefront, thereby improving MRR. Another reason may be that recommendation systems may perform better at larger recommendation list lengths, as there are more recommendations to consider, thereby improving MRR.

NDCG (Normalized Discounted Cumulative Gain) considers the relevance and ranking order of recommended items. The Yoochoose-buys dataset once again performed the best on this metric, NDCG@100 The value of is 0.2742, while RentTheRunway is the worst, NDCG@100 =0.0294. The main reason for this phenomenon is that it has a higher number of user behaviors, while RentTheRunway has the opposite. Similarly, as k increased from 5 to 100, the NDCG indicators of all five datasets improved. The reason for this phenomenon is similar to MRR, as both are indicators for evaluating the quality of ranking.

Hit@k Indicate whether there are any items related to the user's true interests in the first k recommendations. The Yoochoose-buy dataset once again performed the best, while RentTheRunway performed the worst, Hit@100 The values are 0.6725 and 0.1215, respectively. This may be because the Yoochoose-buy dataset has richer user behavior, making it more likely to include items of interest to users in recommendations. Similarly, as k increases from 5 to 100, the Hit metrics of all five datasets improve. The main reason is that as the length of the recommendation list increases, the system has more opportunities to provide items that users like.

Precision@k How many of the top k recommendations are items that users are truly interested in. The Yoochoose-buys dataset still performs the best on this metric, while RentTheRunway performs the worst, Precision@100 The values of the indicators are 0.0067 and 0.0012, respectively. This may also be because there are more user behaviors in the Yoochoose-buys dataset, making it easier for the model to find items that users are interested in. Unlike other indicators, as k increases, the Precision indicator of all 5 datasets shows a downward trend. This is because the calculation formula for Precision is to correctly predict the number of positive samples divided by the number of recommended items. An increase in k means that the denominator directly increases, leading to a decrease in the value of this indicator.

For the MovieLens-100k and MovieLens-1m datasets, which are of the same type, the only difference is the size of the dataset. The MovieLens-1m dataset has more data. By controlling variables, it can be found that the MovieLens-1m dataset with larger data volume performs better on all five indicators, indicating a positive correlation between these indicators and data volume.

Overall, the performance of the NextItNet model varies across different datasets, with the Yoochoose-buys dataset performing the best and the RentTheRunway dataset performing the worst. As the recommendation length increases, the performance of the model usually improves, but the magnitude and trend of the improvement vary on different datasets. In addition, the characteristics and data distribution of different datasets may also affect the performance of the model. Through horizontal comparison, the performance differences of recommendation systems can be found in different dataset scenarios. More user behavior data and lower sparsity will bring better recommendation results. Through vertical comparison between indicators, it can be found that more recommendations will bring some performance improvement to most indicators, but for some indicators that are limited by calculation methods, such as Precision, the indicator values will actually decrease. In addition, through controlled variable control analysis, it can be found that a larger amount of data may increase the performance of recommendation systems. The above data results are as follows.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DATA | Recall@5 | MRR@5 | NDCG@5 | Hit@5 | Precision@5 |
| MovieLens-100k | 0.0604 | 0.0294 | 0.0369 | 0.0604 | 0.0121 |
| Yoochoose-buys | 0.2687 | 0.1469 | 0.177 | 0.2687 | 0.0537 |
| MovieLens-1m | 0.1674 | 0.0931 | 0.1114 | 0.1674 | 0.0335 |
| Amazon Video Games | 0.0811 | 0.0594 | 0.0648 | 0.0811 | 0.0162 |
| Rent the Runway | 0.0117 | 0.0059 | 0.0073 | 0.0117 | 0.0023 |

Table 1: The result of Recall@5, MRR@5, NDCG@5, Hit@5, Precision@5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DATA | Recall@10 | MRR@10 | NDCG@10 | Hit@10 | Precision@10 |
| MovieLens-100k | 0.1166 | 0.0366 | 0.0548 | 0.1166 | 0.0117 |
| Yoochoose-buys | 0.3824 | 0.1622 | 0.2139 | 0.3824 | 0.0382 |
| MovieLens-1m | 0.2487 | 0.1037 | 0.1375 | 0.2487 | 0.0249 |
| Amazon Video Games | 0.1057 | 0.0627 | 0.0727 | 0.1057 | 0.0106 |
| Rent the Runway | 0.0219 | 0.0072 | 0.0106 | 0.0219 | 0.0022 |

Table 2: The result of Recall@10, MRR@10, NDCG@10, Hit@10, Precision@10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DATA | Recall@20 | MRR@20 | NDCG@20 | Hit@20 | Precision@20 |
| MovieLens-100k | 0.1909 | 0.0416 | 0.0734 | 0.1909 | 0.0095 |
| Yoochoose-buys | 0.4809 | 0.1691 | 0.2388 | 0.4809 | 0.024 |
| MovieLens-1m | 0.3558 | 0.111 | 0.1644 | 0.3558 | 0.0178 |
| Amazon Video Games | 0.1406 | 0.0651 | 0.0815 | 0.1406 | 0.007 |
| Rent the Runway | 0.039 | 0.0084 | 0.0149 | 0.039 | 0.002 |

Table 3: The result of Recall@20, MRR@20, NDCG@20, Hit@20, Precision@20

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DATA | Recall@50 | MRR@50 | NDCG@50 | Hit@50 | Precision@50 |
| MovieLens-100k | 0.368 | 0.0471 | 0.1083 | 0.368 | 0.0074 |
| Yoochoose-buys | 0.5957 | 0.1728 | 0.2617 | 0.5957 | 0.0119 |
| MovieLens-1m | 0.5129 | 0.1161 | 0.1957 | 0.5129 | 0.0103 |
| Amazon Video Games | 0.204 | 0.0671 | 0.094 | 0.204 | 0.0041 |
| Rent the Runway | 0.0751 | 0.0095 | 0.0219 | 0.0751 | 0.0015 |

Table 4: The result of Recall@50, MRR@50, NDCG@50, Hit@50, Precision@50

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DATA | Recall@100 | MRR@100 | NDCG@100 | Hit@100 | Precision  @100 |
| MovieLens-100k | 0.5292 | 0.0493 | 0.1343 | 0.5292 | 0.0053 |
| Yoochoose-buys | 0.6725 | 0.1739 | 0.2742 | 0.6725 | 0.0067 |
| MovieLens-1m | 0.6272 | 0.1177 | 0.2143 | 0.6272 | 0.0063 |
| Amazon Video Games | 0.2667 | 0.0679 | 0.1042 | 0.2667 | 0.0027 |
| Rent the Runway | 0.1215 | 0.0101 | 0.0294 | 0.1215 | 0.0012 |

Table 5: The result of Recall@100, MRR@100, NDCG@100, Hit@100, Precision@100

In addition to the five evaluation parameters mentioned above, this article also utilizes the tensorboard function of Rebole to visually display the loss and fail score indicators during model training and evaluation. For loss, the RentTheRunway and MovieLens-100k datasets have lower losses, while the MovieLens-1m dataset has the highest average loss and the longest training time, taking 5.56 hours. For the failed score, the dataset Yoochoose-buys performed the best with a score of 0.14, which also matches the performance of the dataset in the five indicators mentioned above. The RentTheRunway dataset still performed the worst, with a score of only 0.0068.

图表, 折线图

描述已自动生成

Figure 6: Loss performance of five datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | Smoothed value | Final value | Step | Time |
| Amazon Video Games | 2103 | 2066 | 19 | 2.66 hr |
| Rent the Runway | 115.1 | 114.3 | 26 | 17.26 min |
| MovieLens-100k | 254.7 | 251.6 | 20 | 19.88 min |
| MovieLens-1m | 2353 | 2348 | 33 | 5.56 hr |
| Yoochoose-buys | 766.9 | 761.5 | 30 | 2.12 hr |

Table 6: Loss performance of five datasets

图表, 折线图

描述已自动生成

Figure 7: Vaild score performance of five datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | Score(Smoothed) | Score(Final) | Step | Time |
| Amazon Video Games | 0.05975 | 0.059 | 19 | 2.66 hr |
| Rent the Runway | 0.006706 | 0.0068 | 26 | 17.26 min |
| MovieLens-100k | 0.03477 | 0.0365 | 20 | 19.88 min |
| MovieLens-1m | 0.1061 | 0.1052 | 33 | 5.56 hr |
| Yoochoose-buys | 0.141 | 0.1407 | 30 | 2.12 hr |

Table 7: Loss performance of five datasets

# **Professional Issues**

This section first discusses project scheduling and time management, as well as data and literature management, including an activity table, a task list with a progress chart, a time plan, and also introduces how to use software such as Zotero and Github for related management. In addition, this section also conducted a project risk assessment. Finally, the considerations of the project in terms of legal, social, ethical, and environmental issues were also discussed.

## **Project Management**

### **Activities**

|  |  |
| --- | --- |
| **Objectives** | **Tasks** |
| Collection of relevant literature | Read at least 20 articles in relevant fields, select and take notes. |
| Project Proposal | Complete the proposal with clear structure and logic, including a reference list. |
| Understanding models and mathematical methods | Analyze and compare the differences between models and interpret relevant methods. |
| Model implementation | Using framework tools such as Rebole, conduct in-depth research on network architecture, implement a sequence recommendation model based on convolutional generative networks, and debug it appropriately. |
| Experimental data processing | According to relevant research fields, the data set was searched and preprocessed |
| Experiment and Test | The model was implemented by code and applied to the dataset, trial and error |
| Summary | Analyze and summarize the experimental results to reach a conclusion, and complete the remaining writing |
| Paper Modify | Revise the format and improve the article |
| Presentation Prepare | Prepare PPT and review research work |

Table 8: Activities

### **Schedule**

图表, 漏斗图

描述已自动生成

Figure 8: Gantt chart of thesis plan (Semester 1)

图表, 瀑布图

描述已自动生成

Figure 9: Gantt chart of thesis plan (Semester 2)

### **Project Data Management**

This article uses Zotero to manage relevant literature, including storing the literature in the Zotero manager and annotating and taking notes on each literature, making it easy to quickly search for the desired literature. For code files and dataset files, the code files will be uploaded to a personal account through Github for backup, and each version and date will be classified. Finally, regarding report file management, literature will be stored in cloud documents through the use of WPS office.

|  |  |
| --- | --- |
| Literature management | Zotero |
| Code management | Github |
| Report management | WPS office |

Table 9: Management

### **Project Deliverables**

|  |  |
| --- | --- |
| Project Resources | Deadline |
| Weekly reports | Every week from the fourth week of the first semester |
| Project proposal, ethic form, plagiarism report | November 3, 2023 |
| Progress Report | December 22, 2023 |
| Final report | April 12, 2024 |
| Presentation poster, video, and demo | May 29th to 31st, 2024 |
| Tech Show | June 6th -7th, 2024 |

Table 10: Resources and Submission Date

## **Risk Analysis**

In order to identify potential risks in the early stages of the process, risk identification was carried out in this project. Risk identification is a management plan that allows project managers to prioritize risks by analyzing them and assessing their level. In addition, it also provides measures to address risks. The risk identification is shown in Tables 11 and 12, where the risk ID corresponds to the potential risk, and the Cause ID represents the number of the cause or event that caused the risk, corresponding to the potential cause. Severity indicates the severity of the event, which includes 5 levels from 1 to 5. A value of 1 for Severity indicates that the event is relatively minor, while a value of 5 indicates that the event is severe. Likelihood represents the probability of an event occurring, including 5 levels from 1 to 5, where a value of 1 indicates a low probability of the event occurring and a value of 5 indicates a very high probability of the event occurring. And Risk represents the degree of risk, which is calculated by multiplying the value of Severity with the value of Likelihood. This means that there will be a total of 5 \* 5=25 risk levels, with the highest risk level being 25. In the table, green represents low risk, red represents high risk, and yellow represents moderate risk. The Mitigation ID corresponds one-to-one with the Cause ID, and the subsequent Mitigation represents the response measures after a risk event occurs.

Next is a detailed explanation of the above risks. In the previous progress, there was a risk of missing the deadline, which occurred on December 18, 2023. Due to tight schedules in other disciplines, students reported to the supervisor in a unified manner. The supervisor decided to postpone the deadline for the progress report to December 27. This risk was resolved by reporting and communicating with the supervisor in advance. Another risk is environmental configuration and tool installation issues. I encountered many problems while installing Rebole, including downloading errors in the PyTorch version, mismatch between the PyTorch version and the Cuda version, incompatibility between PyTorch and the Cuda and Python versions, and very slow installation speed of Rebole. The first few problems were solved by searching a large amount of information on websites such as CSDN and spending a lot of time, The latter's network and device issues are resolved under the patient guidance of the supervisor. For future risks that may involve data loss, regular use of the Git tool for backup will be adopted.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Potential Risk** | **Cause ID** | **Potential Causes** | **Severity** | **Likelihood** | **Risk** |
| R1.1 | Missed deadline | C1.1.1 | Illness | 1 | 3 | 3 |
| C1.1.2 | Cannot choose topic | 1 | 1 | 1 |
| C1.1.3 | Poor time management | 4 | 3 | 12 |
| R1.2 | Feature creep | C.1.2.1 | Over-ambitious project spec. | 3 | 2 | 6 |
| R1.3 | Environmental configuration & installation issues | C1.3.1 | Careless | 1 | 3 | 3 |
| C1.3.2 | Computer equipment or network issues | 4 | 3 | 12 |
| R1.4 | Loss of data | C1.4.1 | Poor version control | 4 | 4 | 16 |

Table 11: Potential risks

|  |  |
| --- | --- |
| **Mitigation ID** | **Mitigation** |
| M1.1.1 | Explain the situation to the supervisor in advance. |
| M1.1.2 | Conduct research early and meet supervisor. |
| M1.1.3 | Make a Gantt plan early and reserve time for unexpected situations. |
| M1.2.1 | Discuss plan with supervisor early. Create basic (must-have) goals and enhancements (nice-to-have). |
| M1.3.1 | Carefully and patiently follow the installation tutorial step by step or search for relevant resources on CSDN website. |
| M1.3.2 | Refer to online materials pr contact the supervisor in a timely manner, communicate and discuss countermeasures. |
| M1.4.1 | Implement version control strategy at start(use git tool). |

Table 12: Potential risks

## **Professional Issues**

## **5.3.1 Legal issues**

Due to the involvement of various fields of data in this project, such as user transaction records, music preference data, game preference data, movie rating data, clothing rating data, etc., there may be a risk of a large amount of user privacy data being violated and stolen during the dataset collection stage. Therefore, it is necessary to ensure compliance with privacy regulations such as GDPR (European General Data Protection Regulations) or similar regulations in other countries or regions. BCS's Code of Computer Conduct emphasizes the principles of protecting privacy and legitimate use of information. In these regulations, it is necessary to clarify that the datasets used only contain necessary text and fields. In addition, adhering to the principle of transparent use of data, it is also necessary to clearly explain the purpose and data source of using these datasets. On the other hand, regarding intellectual property issues, it is necessary to ensure compliance with relevant intellectual property regulations, such as ACM's Code of Ethics, which involves the principles of legitimate use of computing resources and respect for privacy, as the project uses publicly available datasets from others and references the algorithm framework provided by Recoole. When using these resources, it is also necessary to clearly cite them to avoid issues such as intellectual property infringement and duplicate development.

## **5.3.2 Social issues**

Due to the uncertainty of user behavior, the recommendation results generated by recommendation systems based on convolutional generative networks may not be necessarily accurate. These recommendation results may affect people's judgment to some extent, and they may make inappropriate decisions based on incorrect recommendation results. In addition, the recommendation system used in this article may affect the sales of merchants, as the recommended products are more likely to be purchased. This may lead to some small businesses losing competitiveness due to a lack of exposure, while large enterprises will benefit more. Referring to the ACM Code of Professional Ethics, it is necessary to be aware of the widespread impact of computer science on society and consider the long-term interests of society when designing recommendation systems. It is necessary to ensure that the design of the system does not cause undue harm to specific individuals or groups. In practical applications, users should also be provided with the right to choose whether to be recommended, in order to minimize the negative impact of recommendation systems on the public.

## **5.3.3 Ethical issues**

Due to the rarity of publicly available recommendation systems, there are few opportunities for the public to learn about recommendation systems and models. The Computer Code of Conduct based on BCS involves ethical principles, including transparency and respect for user choices. It is necessary to consider improving the interpretability of recommendation system models, maximizing public awareness of the issues that may be involved in recommendation systems, and safeguarding the public's right to information. In addition, some applications that use the recommendation system mentioned in this article may force users to make recommendations, and it is necessary to explain the reasons for the recommendation or the need for user ratings to the users at the same time as the recommendation.

## **5.3.4 Environmental issues**

Due to the high computational resources required for the model in this article, especially in large-scale datasets, the training and inference process of the model may have negative impacts on the environment, such as increasing carbon footprint and energy consumption. The BCS (British Computer Society) Code of Conduct emphasizes the responsibility of using resources. In this guideline, it not only requires computer professionals to consider the rational utilization of resources when designing, developing, and implementing systems, but also emphasizes that they should be responsible for the consequences of resource consumption. This means that when choosing this recommendation model, not only should its performance and effectiveness be considered, but also its energy consumption and potential impact on the environment. In addition, resource consumption can be reduced to some extent by using algorithms that optimize resources, such as distributed and parallel computing algorithms, dynamic computing resource allocation algorithms, etc.

# **Conclusion**

## **6.1 Reflection and Conclusion**

In this report, the NextItNet model was implemented to apply convolutional generative networks to recommendation systems. The model was tested using different statistical metrics and five different datasets: Yoochoose-buys, MovieLens-100k, MovieLens-1m, Amazon Video Games, Rent The Runway. While implementing recommendations, the relationship between the model, dataset, and related parameters was analyzed to identify factors that affect recommendation performance.

The experimental results show that the yoochoose buy dataset performs the best among 25 metrics, especially Recall and Hit, reaching 67.25% simultaneously. Due to the fact that the yoochoose buy dataset has more user interaction data and lower sparsity, it can be found that the NextItNet model has a certain requirement for the size of the dataset, making it more suitable for complex datasets with more data volume. This can also be demonstrated through parallel controlled experiments on movieLens datasets of the same type but with different data volumes. The MovieLens-1m dataset with larger data interaction has significantly higher indicators than the MovieLens-100k dataset. In addition, according to the test results of adjusting different statistical metrics, the higher the k value (the number of recommended items per time), the better the recommendation effect of the model will be. However, if users pursue a higher Precision value, due to the calculation of the formula itself, as the k value increases, the metric will decrease.

Overall, this project has successfully implemented a sequence recommendation model based on convolutional generative networks. The model uses deep learning techniques to better capture hidden features in user project interaction data, and based on residual learning, supports sequence recommendations of different lengths. Then, the model was applied to different types of datasets, and the deployment results revealed the feasibility and effectiveness of the model in different recommendation scenarios. The requirements of the model for the dataset and the main factors affecting recommendation performance were pointed out.

## **6.2 Potential Future Work**

This study has encountered some limitations, mainly including high training and time costs, limited equipment performance, inability to handle large datasets, and the use of smaller datasets. Below, we will explore the expansion of this project and future research directions from different perspectives.

Hardware and Datasets: Based on the limitations of this device, we will consider using graphics card devices with stronger computing performance and higher running memory to support model training on large datasets in the future. In addition, updated data preprocessing techniques will be used to simplify and shrink the dataset without affecting recommendation performance, in order to meet device requirements, or phased training can be adopted for the dataset.

Model tuning and improvement: This time, the NextItNet model has been applied to 5 datasets. In the future, more recommendation models based on convolutional generative networks will be used, and comparisons will be made between the models. In addition, adjustments will be made to the model parameters and structure of the NextItNet model, aiming to find more suitable neural network types for application in recommendation models.

Data security: Based on this study, it was found that recommendation models require users to transfer data to a data center when making recommendations, which then returns the results to the user. In the future, the author will consider introducing federated learning into the recommendation model, so that it can directly make recommendations to user local data, thereby maintaining user data security to a certain extent.

# **References**

[1] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, ‘A Simple Convolutional Generative Network for Next Item Recommendation’. arXiv, Nov. 28, 2018. Accessed: Oct. 27, 2023. [Online]. Available: http://arxiv.org/abs/1808.05163

[2] M. Xu, F. Liu, and W. Xu, ‘A Survey on Sequential Recommendation’, in *2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, Dec. 2019, pp. 106–111. doi: 10.1109/ICISCE48695.2019.00031.

[3] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, ‘Evaluating Collaborative Filtering Recommender Algorithms: A Survey’, *IEEE Access*, vol. 6, pp. 74003–74024, 2018, doi: 10.1109/ACCESS.2018.2883742.

[4] R. K. Nath and T. Ahmad, ‘Content Based Recommender System: Methodologies, Performance, Evaluation and Application’, in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Dec. 2022, pp. 423–428. doi: 10.1109/ICAC3N56670.2022.10074332.

[5] S. Zhang, L. Yao, A. Sun, and Y. Tay, ‘Deep Learning Based Recommender System: A Survey and New Perspectives’, *ACM Comput. Surv.*, vol. 52, no. 1, p. 5:1-5:38, Feb. 2019, doi: 10.1145/3285029.

[6] Z. Cheng, J. Shen, L. Zhu, M. Kankanhalli, and L. Nie, ‘Exploiting music play sequence for music recommendation’, in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, in IJCAI’17. Melbourne, Australia: AAAI Press, Aug. 2017, pp. 3654–3660.

[7] A. Beutel *et al.*, ‘Latent Cross: Making Use of Context in Recurrent Recommender Systems’, in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, in WSDM ’18. New York, NY, USA: Association for Computing Machinery, Feb. 2018, pp. 46–54. doi: 10.1145/3159652.3159727.

[8] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, ‘Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations’, in *Proceedings of the 10th ACM Conference on Recommender Systems*, in RecSys ’16. New York, NY, USA: Association for Computing Machinery, Sep. 2016, pp. 241–248. doi: 10.1145/2959100.2959167.

[9] A. Dallmann, A. Grimm, C. Pölitz, D. Zoller, and A. Hotho, ‘Improving Session Recommendation with Recurrent Neural Networks by Exploiting Dwell Time’. arXiv, Jun. 30, 2017. doi: 10.48550/arXiv.1706.10231.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ‘Gradient-based learning applied to document recognition’, *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘ImageNet classification with deep convolutional neural networks’, *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[12] C. Cheng, H. Yang, M. R. Lyu, and I. King, ‘Where You Like to Go Next: Successive Point-of-Interest Recommendation’.

[13] X. He and T.-S. Chua, ‘Neural Factorization Machines for Sparse Predictive Analytics’. arXiv, Aug. 16, 2017. Accessed: Oct. 27, 2023. [Online]. Available: http://arxiv.org/abs/1708.05027

[14] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, ‘Session-based Recommendations with Recurrent Neural Networks’. arXiv, Mar. 29, 2016. Accessed: Oct. 27, 2023. [Online]. Available: http://arxiv.org/abs/1511.06939

[15] Y. K. Tan, X. Xu, and Y. Liu, ‘Improved Recurrent Neural Networks for Session-based Recommendations’. arXiv, Sep. 16, 2016. doi: 10.48550/arXiv.1606.08117.

[16] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, ‘Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks’, in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, Aug. 2017, pp. 130–137. doi: 10.1145/3109859.3109896.

[17] J. Tang and K. Wang, ‘Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding’. arXiv, Sep. 19, 2018. Accessed: Oct. 27, 2023. [Online]. Available: http://arxiv.org/abs/1809.07426

[18] W. Song, S. Wang, Y. Wang, and S. Wang, ‘Next-item Recommendations in Short Sessions’. arXiv, Jul. 20, 2021. doi: 10.48550/arXiv.2107.07453.

[19] D. Chai, L. Wang, K. Chen, and Q. Yang, ‘Secure Federated Matrix Factorization’, *IEEE Intell. Syst.*, vol. 36, no. 5, pp. 11–20, Sep. 2021, doi: 10.1109/MIS.2020.3014880.

[20] Z. Li, G. Long, and T. Zhou, ‘Federated Recommendation with Additive Personalization’. arXiv, May 17, 2023. doi: 10.48550/arXiv.2301.09109.

[21] C. Kumar and M. Kumar, ‘Next-item recommendation within a short session using the combined features of horizontal and vertical convolutional neural network’, *Multimed. Tools Appl.*, 2023, doi: 10.1007/s11042-023-17201-z.

[22] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, ‘Neural Attentive Session-based Recommendation’, in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, in CIKM ’17. New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1419–1428. doi: 10.1145/3132847.3132926.

# **Appendices**