

UNDERGRADUATE PROJECT PROPOSAL

|  |  |
| --- | --- |
| **Project Title:** | **Federated** **Convolutional Generative Network for Next Item Recommendation** |
| **Surname:** | **Long** |
| **First Name:** | **Yicheng(Jack)** |
| **Student Number:** | **202018010116** |
| **Supervisor Name:** | **Joojo Walker** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** |  |

**Table of Contents**

[1 Introduction3](#_Toc118788384)

[1.1 Background 3](#_Toc118788385)

[1.2 Aim 3](#_Toc118788386)

[1.3 Objectives 3](#_Toc118788387)

[1.4 Project Overview 4](#_Toc118788388)

[1.4.1 Scope 4](#_Toc118788389)

[1.4.2 Audience 4](#_Toc118788390)

[2 Background Review 5](#_Toc118788391)

[3 Methodology 6](#_Toc118788392)

[3.1 Approach 6](#_Toc118788393)

[3.2 Technology 9](#_Toc118788394)

[3.3 Version management plan 9](#_Toc118788395)

[4 Project Management 9](#_Toc118788396)

[4.1 Activities 9](#_Toc118788397)

[4.2 Schedule 11](#_Toc118788398)

[4.3 Data management plan 11](#_Toc118788399)

[4.4 Project Deliverables 11](#_Toc118788400)

[5 References 12](#_Toc118788401)

# Introduction

## Background

## In recent years, recommendation systems that utilize user project interaction sequences to improve real-world performance have become increasingly popular due to the fact that users typically watch or listen to a series of items consecutively, with strong correlations between these items. For example, users of Last. fm or Weishi typically enjoy a series of songs/videos over a period of time [5]. As a deep learning neural network architecture, CNN (Convolutional Neural Network) has excellent feature extraction capabilities and sequential data processing performance, making it very common to apply CNN to recommendation systems.

## Aim

In recent years, with the advancement of hardware and the development and application of deep learning technology, applying machine learning models to next recommendation has become increasingly popular. However, traditional CNN based recommendation systems still have some shortcomings. This paper proposes a federated convolutional generation network model for next recommendation to solve these problems and improve recommendation effectiveness.

## Objectives

Collection of relevant literature

,Project Proposal Draft

Understanding models and mathematical methods

Improved model

Experimental data processing

Experience and Test

Summary

Paper Modify

Presentation Prepare

## Project Overview

### Scope

The purpose of the study is to improve the next recommendation system by introducing a federated convolutional generation network model to overcome the shortcomings of traditional CNN recommendation systems. This model aims to better capture the features and sequential patterns in user project interaction sequences, thereby providing more personalized and accurate recommendations. Regarding importance, it has technological innovation and addresses the shortcomings of traditional methods. It also has significant importance for practical applications, improving user experience and having a positive impact in commercial applications.

### Audience

Users will be one of the main beneficiaries. The next improved recommendation system will provide more personalized and user interested recommendation content, thereby improving their user experience. Users will find products, services, or information of interest more easily, thereby saving time and effort. In addition, there are also businesses or advertisers who are the main beneficiaries. By increasing transaction volume, businesses can benefit, and advertisers can place more targeted advertisements to increase 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 [1] and feature based matrix decomposition [2] 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. [3] proposed a DL 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. In 2018, Tang and Wang [4] 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. [5] 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. [6] 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. Finally, in 2023, Kumar et al. [7] 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

## Approach

## 3.1.1 Sequential Recommendation

## Sequence refers to user item interaction sequence, which records the interaction history between the user and the item. This can be a user's interaction with an item in a session, such as clicking, buying, or scoring. Each item in the sequence is represented by Xi, usually the index of an item, which indicates which item the user interacted with in the sequence. There is a subsequence prefix item sequence in the user item interaction sequence, which is usually represented by x = {x0,..., xi}, where 0 ≤ I < t. This represents a part of the interaction history between users and items, which can be used to predict the items that users may be interested in in future interactions. The task of sequential recommendation is to generate a ranking or classification distribution. For a given prefix item sequence x, the goal of the model is to generate a ranking or classification distribution y for all candidate items. This distribution y contains the scores, probabilities or rankings of all candidate items, indicating their probability in the interaction sequence of future users. In practice, it is usually necessary to provide users with multiple recommendation results, not just a single one. To do this, you can select the top n highest scores or most likely items from the generated ranking or classification distribution, and then recommend them to users. These selected items constitute the so-called "top-N recommendation" to meet the personalized needs of users.

## 3.1.2 Recommended scheme of conversational neural networks: Caser

## Caser (convolutional sequence embedded recommender) is a model for sequential recommendation tasks, which is based on convolutional neural network (CNN) rather than recurrent neural network (RNN) structure. The core idea of caser is to embed the user's interaction sequence in the session into a two-dimensional matrix, similar to the image in image processing, and then use convolution operation to capture the local features in this "image" for recommendation.

## The following are the main principles and steps of the caser model:

## Input representation: the caser model accepts a user's interaction sequence in the session as input. This sequence is usually expressed as a matrix, where each row corresponds to a time step and each column corresponds to an item (or feature). The size of this matrix is usually t × k. Where t is the length of the interaction sequence and K is the embedding dimension.

## Convolution processing: the caser model uses convolution operations to process this matrix. Usually, CNN is applied to image recognition, that is, image data is often used as input data of CNN model, and the idea of this model is to integrate t × The embedding matrix of K is treated as an "image", where T represents the time step and K represents the embedded dimension. Convolution allows the model to capture local features in the input matrix, just as convolution layers in image processing can capture features of images.

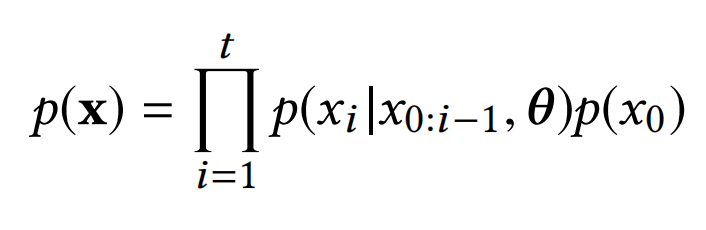
## Local feature extraction: the caser model regards the sequential patterns in the interaction sequence as the local features of this "image". This means that the model can capture the correlation between adjacent interactions without relying on recursive structures such as RNN.

## Maximum pooling operation: in order to increase the receptive field of the model to handle input sequences of different lengths, the caser model performs the maximum pooling operation. The maximum pooling operation only preserves the maximum value in the convolution layer, allowing the model to focus on a wider range of information without being limited by the length of the input sequence.

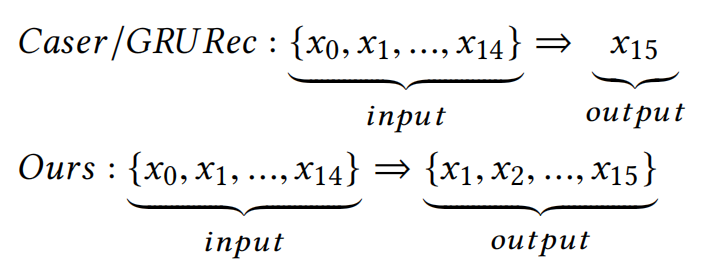
## Parallel computing: a key advantage is that unlike RNN, CNN can compute each element in parallel without relying on the previous time step, which helps improve the efficiency of training and inference.

## 3.1.3 A New Probability Generation Model

## The goal of this solution is to create a model that captures the distribution of item interaction sequences, which represents how users interact with items over time. The goal of this model is to estimate the likelihood of these interactions and generate predictions for future interactions. The model defines a joint distribution represented by p (x), covering the entire item sequence x={x0,..., xt}, where t represents the time step. This distribution captures the relationships and dependencies between items in the sequence. To model p (x), the joint distribution is decomposed using chain rules. This means that the probability of observing the entire sequence is decomposed into the product of the conditional probabilities of each item. As shown in Figure 1, p (x) represents the conditional probability p (xi | x0: i-1) of each item xi, θ) The product of. Each conditional probability p (xi | x0: i-1, θ) Indicates the probability of the i-th item appearing under the conditions of all previous items in sequence x0: i-1, where θ Represents model parameters. This scheme proposes the use of neural networks, specifically a set of 1D convolutional networks, to model these conditional distributions. This network takes sequence x0: t-1 as input and generates the probability distribution of future items x1: t. The model considers the order dependency in interactive data. For example, the distribution of item xt is affected by x0: t-1, while x14 is affected by x0:13. This means that the model captures how the previous item in the sequence affects the probability of observing the next item, while compared to the previously mentioned Caser scheme, GRU Rec, etc., typically only a conditional distribution p (xi | x0: i-1, θ)， Used to predict the next item (in this case x15). In contrast, as shown in Figure 2, the new model proposes a generation model that estimates the distribution of all individual items in the sequence from x1 to x15, providing a more comprehensive potential for user interaction. Overall, the improvement of the proposed model is mainly reflected in estimating the probability distribution of multiple future items simultaneously, capturing all possible interaction ranges.



**Figure 1: Conditional probability of an item**



**Figure 2: Conditional distribution prediction**

## Technology

Hardware includes: computer (GPU: NVIDIA RTX2070S, RAM: 500G), software includes: deep learning framework: TensorFlow, programming language: Python 3.9, editor: PyCharm3.3, data preprocessing tool: MySQL.

## Version management plan

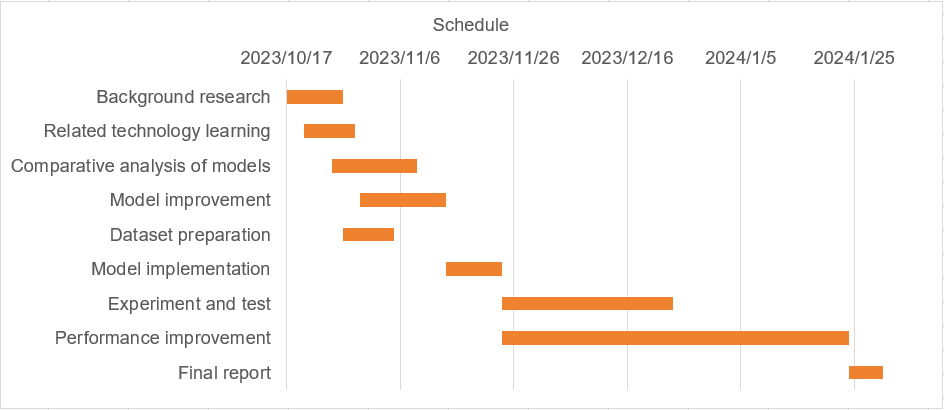
Use the Git repository and Flybook to manage project code or multiple versions of models that have been developed. The code will continue to be updated on Github's personal homepage, and backup management will also be carried out every time it is submitted through Flybook to facilitate subsequent changes.

# 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 |
| Improved model | According to the shortcomings of the existing model, search for possible improved technologies and methods through the literature |
| 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 |

## Schedule



**Table 1: Gantt chart of thesis plan**

## Data management plan

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

**Table 2: Management**

## Project Deliverables

Project proposal, weekly report, final report, project code/software, presentation PowerPoint.

# References

1. C. Cheng, H. Yang, M. R. Lyu, and I. King, ‘Where You Like to Go Next: Successive Point-of-Interest Recommendation’.
2. 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>
3. 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>
4. 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>
5. 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>
6. 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](https://doi.org/10.48550/arXiv.2107.07453).
7. C. Kumar and M. Kumar, ‘Next-item recommendation within a short session using the combined features of horizontal and vertical convolutional neural network’, Multimedia Tools and Applications, 2023, doi: [10.1007/s11042-023-17201-z](https://doi.org/10.1007/s11042-023-17201-z).