成都理工大學中英合作办学
CDUT Sino-British Collaborative Education

OXFORD BROOKES UNIVERSITY

UNDERGRADUATE PROJECT PROGRESS REPORT

Project Title:	**Book Recommendation Using Deep & Cross Network**
Surname:	**Cheng**
First Name:	**Arnold**
Student Number:	**201918020422**
Supervisor Name:	**Joojo Walker**
Module Code:	**CHC 6096**
Module Name:	**Project**
Date Submitted:	**2022/01/12**

Table of Contents

1 Introduction

1.1 Background

The rapid development of web applications, especially mobile applications, has made it easy for people to browse a large number of web information resources, and the problem of information overload is a result of the rapid expansion of the Internet in terms of coverage and scale. There is so much information available at one time that it is difficult for users to filter it, reducing the effectiveness of information use[1], and how to recommend resources (e.g., goods, movies, books, etc.) that meet users' needs from the vast amount of information resources has become one of the current concerns of researchers.

A particularly promising approach to solving the information overload problem is the use of recommendation systems (RS), which are important information filtering tools. In 1997, Resnick & Varian [2]gave the now widely cited definition of a recommendation system: "It is the use of e-commerce websites to provide customers with product information and recommendations to help users decide what products they should buy, simulating a salesperson helping customers through the buying process". Extracting appropriate information from a large amount of information and recommending it to users can solve the problem of information overload and create a smooth and comfortable network environment[3].

Customer needs are often ambiguous, so if merchants can recommend products that meet users' ambiguous needs, they can convert users' potential needs into real needs and thus achieve increased product sales. And for different users, recommendation systems are committed to providing a recommendation system with different results for everyone[4]. Amazon's book recommendations, Apple Music's recommendations, and Taobao's product movie news usage recommendations all use these methods and have achieved significant benefits[5].

Recommender systems are emerging as a new research hotspot and also face the data sparsity problem (too small number of user ratings for recommended items) and the cold start problem (no rating data for newly recommended items and new users). Deep Learning (DL), a machine learning algorithm with recognition, analysis, and computation, brings new opportunities to alleviate data sparsity and cold start problems.

According to Singhal A et al., most of deep learning methods are enhancing collaborative filtering methods and are a significant improvement over matrix decomposition methods[6].

Therefore, the research on book recommendation systems incorporating deep learning is profoundly promising, with the goal of taking book recommendation systems to a whole new level of performance, reducing the difficulty of accessing information, and increasing the effectiveness of information delivery in today's society. In my next work, I will use deep & crossing network (DCN) deep learning models to build a recommendation system that can recommend books more effectively. The rest of the report is structured as follows

1.2 Aim

The aim of my project is to build and train a book recommendation system using a deep learning model.

1.3 Objectives

My objectives are as follows:

A). completes the research, study, and research of the recommender system.

B). completes the learning and research of deep learning.

C). collects appropriate data for analysis and evaluation.

D). uses suitable deep learning models in combination with recommender systems.

E). implemented and tested.

1.4 Project Overview

1.4.1 Scope

The purpose is to use a model of deep learning called Deep & Cross Network (DCN) [7]to train and build a book recommendation system. Because the DCN can effectively capture the interaction of limited effective features, learn highly nonlinear interactions, does not require manual feature engineering or traversal search, and have low computational cost, and also can more efficiently processing of data sparsity, and cold start problems of traditional recommendation systems.

The importance of my experiments is to use deep learning algorithms to learn and deal with complex problems like humans, to analyze and compute linear or nonlinear feature sequences from multiple dimensions in the face of complex scale data, to automatically learn features that match user needs from massive data, and to build recommendation systems.

1.4.2 Audience

Users, merchants, and those who provide web services all benefit from this.

Because book recommendations solve the problem of users choosing a huge number of products quickly. And for a book site, generating more purchases is where the real money is. And recommending books that are more to the user's liking will undoubtedly increase user stickiness, improve user retention, and better attract users, and then the advertising revenue that comes with it will increase. This is also a win-win situation.

More efficient recommendations also save more reuse of resources, as well as a surge in search volume retrieval, maintaining the stable operation of the site.

The core needs of the Internet are growing, and recommendation systems are at the heart of that growth.

2 Background Review

A recommendation system is a new research field that combines data mining, prediction algorithm [8], machine learning, and other disciplines. The first definition of a recommendation system was given in the literature[2], which pointed out that in daily life, whether it is known or unknown events, people need to make decisions at all times. In the face of familiar things, people can often rely on past experience to make reasonable decisions. However, in the face of unknown things, people need other people's oral suggestions, book reviews, film reviews, recommendations, etc. to make judgments.

In literature[9], it is believed that the recommendation system is to match a large number of items for different users that meet their interests and preferences but are not observed by users. It is believed that the recommendation system is becoming an important business with significant economic impact. Essentially, recommendation system is a simulation of human behavior. It analyzes and processes specific data information through recommendation algorithm, and then recommends the processed results to users with relevant needs[10]. The recommendation algorithm is the core of the recommendation system. It can model according to the user's historical purchase needs, behavior records or similar preferences, to find the requirements that meet the user's preferences and recommend them to the user. The formal definition of recommendation system [11], [12] is as follows:

Let P represent the collection of all users, and C represent the collection of objects that users can recommend. In practical problems, P and C are very large collections. The function f indicates the preference of user p for c, that is, f: $P \times C \rightarrow R$, where R represents a finite sequence of non-negative real numbers, and the recommendation object $c' \in C$ that makes the function f obtains the maximum value is recommended to the user. As shown in formula:

$$\forall p \in P,\ c_p' = \arg\max_{c \in C} f(p,c) \tag{1}$$

c_p' represents the recommendation object that best meets the user's p preference.

Therefore, before selecting the most interesting object for users, the recommendation system must use known user recognition to complete the prediction of the unknown recommendation object recognition, which is the process of recommendation system extrapolation. In recent years, recommendation technologies have been classified from different perspectives, and different scholars have given different connotations to recommendation systems. At present, traditional recommendation systems are divided into three categories[13]: content-based recommendation (CB)[14], collaborative filtering recommendation (CF)[15], and hybrid recommendation[16].

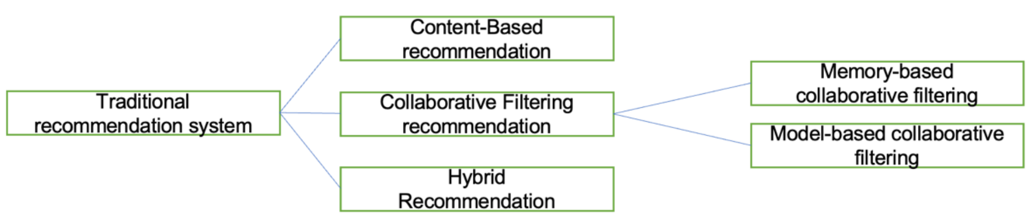

Figure1: traditional recommendation system

Recommendation system based on deep neural network

Deep Neural Network (DNN) is one of the deep learning models[17]–[20], which can also be called multi-layer neural network or multi-layer perceptron (MLP). At present, the trend of introducing deep neural network technology into personalized recommendation is increasingly obvious [18], [21], [22].

In the essay [20], the deep neural network model was first integrated into the field of video recommendation, and simulation experiments were conducted on YouTube video website. YouTube video website is characterized by many registered users, fast video updates, different video duration, and a large number of videos. It is difficult for traditional recommendation algorithms to recommend video content that meets users' preferences. The recommendation process is divided into two stages: candidate set generation and video sequencing. The candidate set generation stage can be regarded as a process of video filtering, that is, select a video set similar to the user's viewing history from the existing videos according to the user's viewing records as the next recommended candidate video. In the candidate set generation stage, the video recommendation problem is regarded as a multi-classification problem, and the user and video are modeled using a deep neural network. The simulation results show that the recommendation model proposed in the literature[20] has a high recall rate and efficiency and can train video data sets with a scale of millions However, the model still has the following shortcomings: 1) In the face of massive video data, the model only cleans the data simply; 2) Video websites often have malicious videos (such as advertisements).

A recommendation model combining DNN and matrix decomposition is proposed in reference[19], which can quickly establish the nonlinear model required for generating interaction functions for user items. Compared with the single matrix decomposition algorithm, the model further improves the accuracy of scoring prediction results and improves the recommendation performance; However, the model does not extract user preferences from multiple dimensions, and its generalization ability is poor. To solve this problem, a DNN-based deep hybrid recommendation model was proposed in[18]. The model inputs the user and project information into the improved machine learning model for training, and further studies the interaction between the user and the recommended project from multiple dimensions. The feature learning part of the model for users and projects is composed of two parallel DNNs. The purpose is to extract the potential features of static projects and dynamic users, to accurately predict user preferences and improve recommendation performance.

In [17], a Wide&Deep model is proposed to solve large-scale online recommendation problems. This model is a combination of a single-layer Wide part and a multi-layer Deep part.

The Wide&Deep model mainly uses the Wide part to learn the characteristics of target users, and the Deep part to generalize similar recommendation items. It can train 500 billion samples, effectively alleviate the problem of data sparsity, and can also be used for classification, regression, search, and other problems; Its shortcoming is that it needs artificial feature engineering.

Deep & cross network (DCN)[7] is to replace the wide part with the cross-layer network based on wide&deep to handle feature crossing.

3 Project Technical Progress

3.1 Methodology

3.1.1 Approach

A brief description of the principle of the Deep & Cross Network model

The Deep & Cross Network model we will refer to as the DCN model below:

A DCN model starts with an embedding and stacking layer, followed by a crossover network and a deep network parallel to it, followed by a final combinatorial layer that combines the outputs of both networks. The complete network model is shown in the figure:

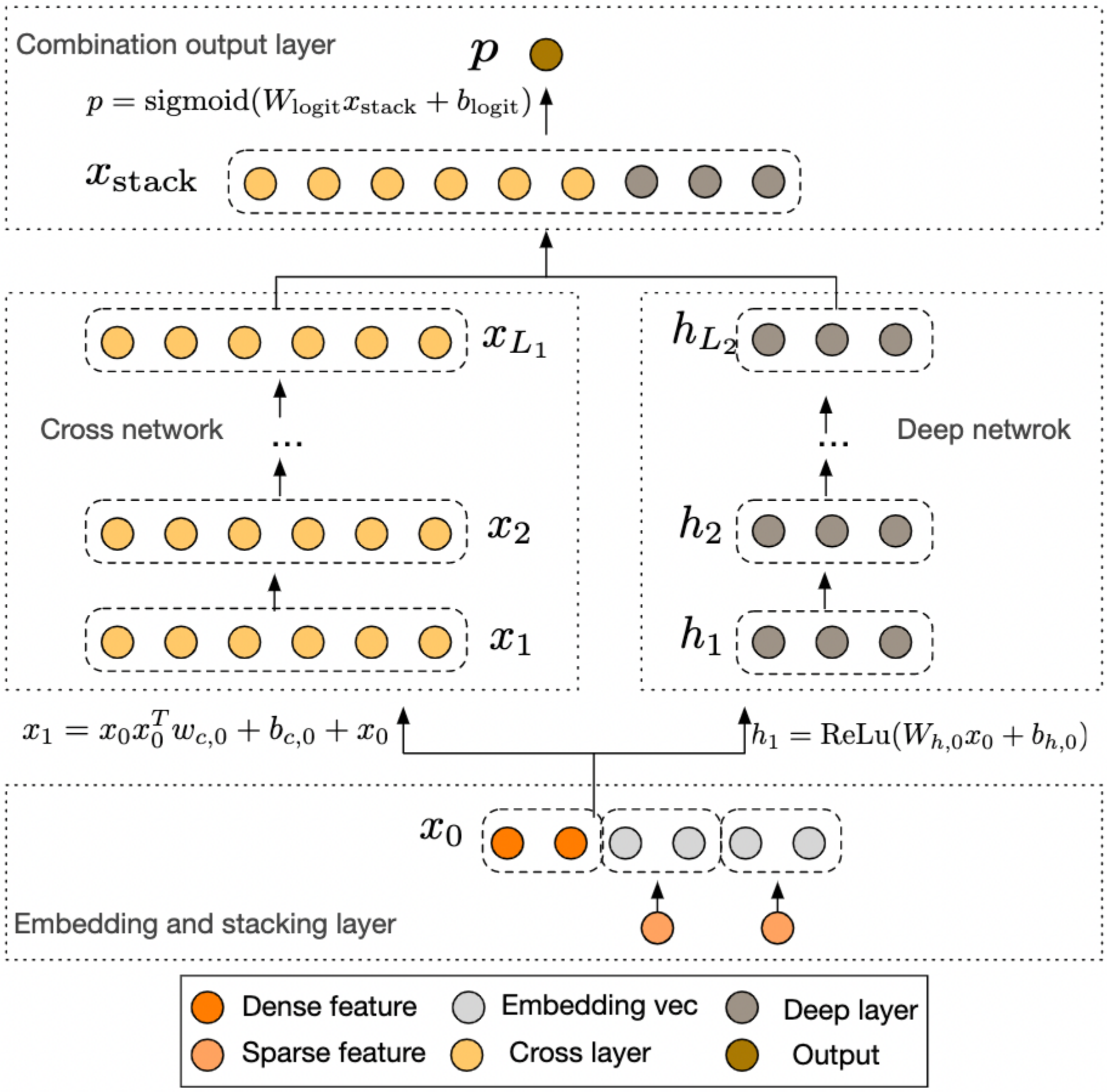

Figure 2: Network Model

Embed and stack layers

The authors consider input data with discrete and continuous characteristics. In network-scale recommendation systems, such as CTR prediction, the input is mainly categorical features, such as "country=USA". These features are usually encoded as a single heat vector such as "[0,1,0]"; However, this often leads to excessive high-dimensional feature space for large vocabulary.

To reduce the dimensionality, we use an embedding process to convert these discrete features into a dense vector of real values (often called an embedding vector):

$$x_{embed,i} = W_{embed,i}x_i \tag{2}$$

Then, we superimpose the embedding vector with the continuous eigenvector to form a vector:

[2] $$x_0 = \left[x_{embed,1}^T \cdots\cdots\cdots x_{embed,k}^T, x_{dense}^T\right] \tag{3}$$

The stitched vector X0 will serve as input to our Cross Network and Deep Network

Cross Network

The core idea of crossover networks is to apply explicit feature intersections in an efficient way. A crossover network consists of intersecting layers, each with the following formula:

$$x_{l+1} = x_0 x_l^T w_l + b_l + x_l = f(w_l, b_l, x_l) + x_l \tag{4}$$

A visualization of a cross-layer is shown in the figure:

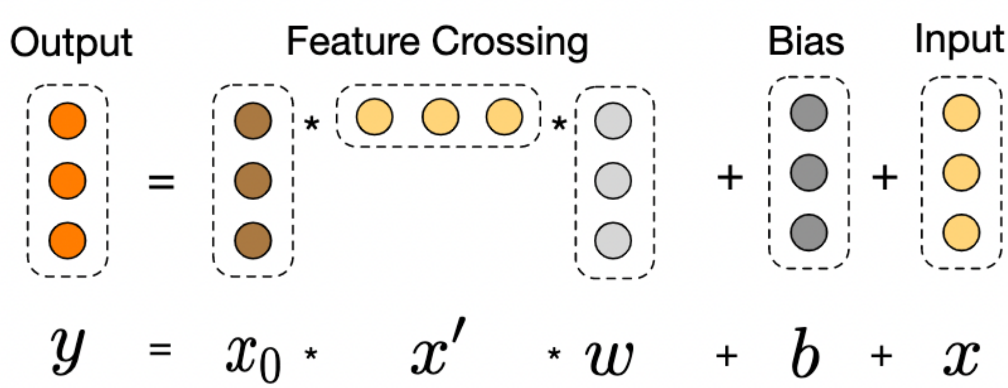

Figure 3: visualization of a cross-layer

The special structure of the intersection network makes the degree of intersection features increase with the increase of layer depth. The highest degree of polynomial (in terms of input X0) is the L-layer crossover network L+1. If Lc is used to represent the number of intersections, d is used to represent the input dimension. Then, the number of parameters involved across the network parameters is d*l_c*2 (w and b).

A few parameters of the cross-network limit the model capacity. To capture highly nonlinear interactions, the model introduces a deep network in parallel.

Deep Network

A deep network is a fully connected feedforward neural network, and each depth layer has the following formula:

$$h_{l+1} = f(W_l h_l + b_l) \tag{5}$$

Combination Layer

The link layer connects the outputs of the two parallel networks and passes through a full link layer to get the output:

$$\rho = \sigma([x_{L1}^T, x_{L2}^T]w_{logits}) \tag{6}$$

DCN can effectively capture the interaction of limited effective features, learn highly nonlinear interactions, do not require manual feature engineering or traversal search, and have low computational cost.

The main contributions of the paper include:

a) A new crossover network is proposed to explicitly apply feature intersection at each layer, effectively learning bounded predicted crossover features without manual feature engineering or exhaustive search.

b) Simple and effective across networks. By design, each layer has the highest polynomial series and is determined by the layer depth. The network consists of all the intersections, which have different coefficients.

c) Cross-network memory is efficient and easy to implement.

d) Experimental results show that the crossover network (DCN) has nearly an order of magnitude less parameters than DNN on LogLoss.

Then Google launched the DCN-V2 model[23]to optimize V1.

The main improvements of DCN-V2 over the previous version of the model are:

(1) Replace vectors with matrices in Wide side-Cross Network;

(2) Two model structures are proposed, the traditional Wide&Deep parallel + Wide&Deep serial.

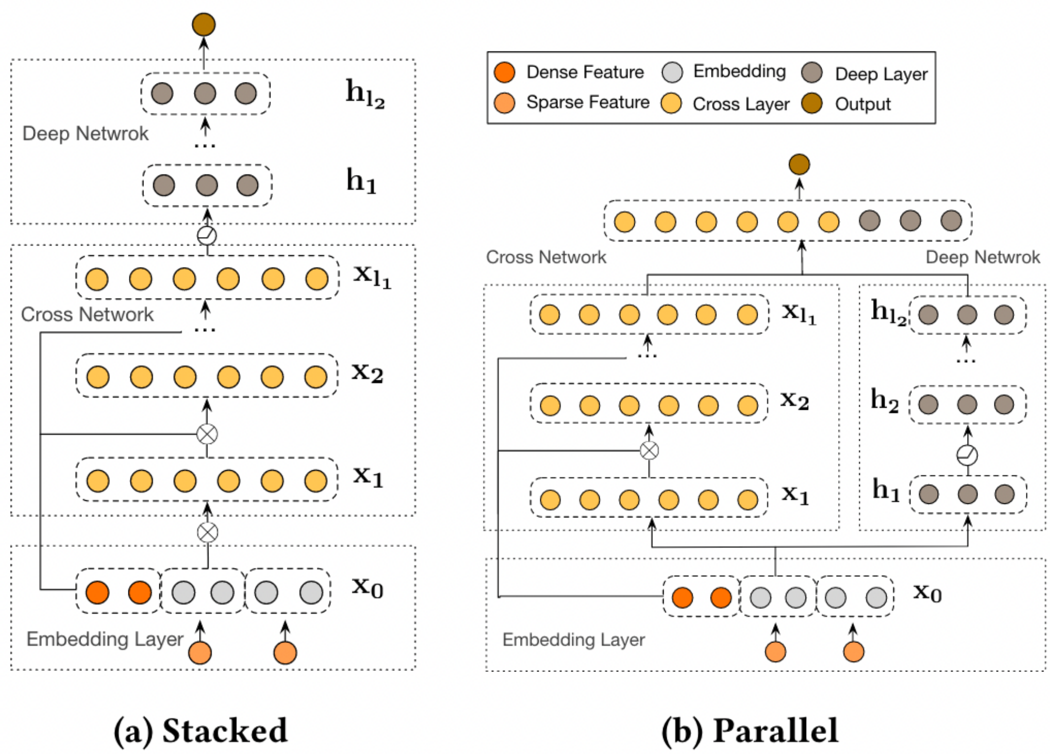

Figure 4: Different model structures

The right side (b) Parallel structure (Wide&Deep parallel) is the DCN-V1 structure, regardless of the cross network structure.

Then the first improvement is to serialize the Wide side and Deep side. Model Serial on both sides.

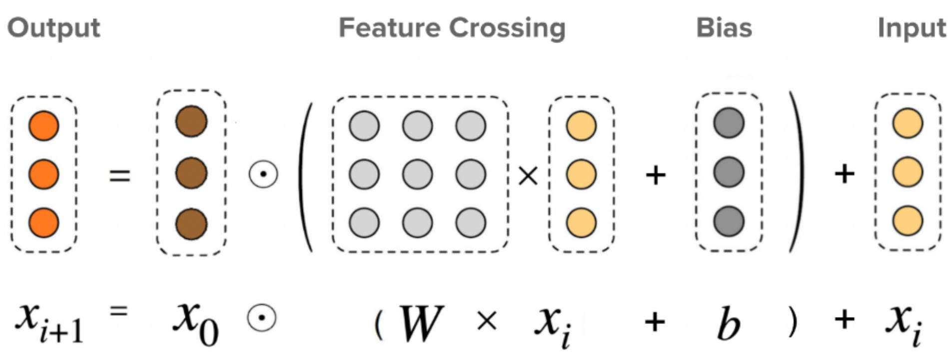

Figure 5: DCN-V2 cross network layer

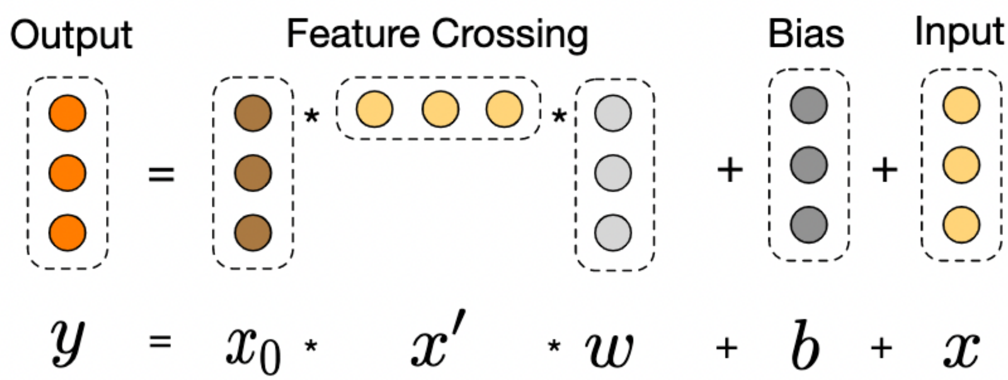

Figure 6: DCN-V1 cross network layer

The second difference between V2 and V1 is that in the Ross Network on the Wide side, a matrix is used instead of the vector in V1 to fit. At the same time, the order of calculation has been changed:

$$x_0 * x_i * w + b \Longrightarrow x_0 * (W * x_i + b) \tag{7}$$

The essence of a matrix is to transform vectors. In V 1, the parameter we use to fit is a vector, as shown in Fig. Shape(w) = [3,1]; In V2, our fitting parameters become a matrix shape (W) = [3,3];

For the same vector, the matrix of [3,3] can be transformed more strongly than that of [3,1], so the fitting ability is stronger.

3.1.2 Technology

These are the hardware and software that I'm going to use in the project

HARDWARE	SOFTWARE
MACBOOK PRO 16 INCHES SOC: M1 PRO	Colab Pytorch environment Google Drive Language: python Jupyter notebook

Table1 : tools

3.2 Testing and Evaluation

Firstly I do the pre-train testing

```python
##test DeepCross

x_num = torch.randn(2, 3)
x_cat = torch.randint(0, 2, (2, 3))

dcn_vec = DeepCross(d_numerical=3, categories=[4, 3, 2], d_embed_max=4,
                    n_cross=2, cross_type="vector",
                    mlp_layers=[20, 20], mlp_dropout=0.25,
                    stacked=False,
                    n_classes=1)

dcn_matrix = DeepCross(d_numerical=3, categories=[4, 3, 2], d_embed_max=4,
                       n_cross=2, cross_type="matrix",
                       mlp_layers=[20, 20], mlp_dropout=0.25,
                       stacked=True,
                       n_classes=1)

dcn_mix = DeepCross(d_numerical=3, categories=[4, 3, 2], d_embed_max=4,
                    n_cross=2, cross_type="mix", low_rank=32, n_experts=4,
                    mlp_layers=[20, 20], mlp_dropout=0.25,
                    stacked=False,
                    n_classes=1)

print(dcn_vec((x_num, x_cat)))
print(dcn_matrix((x_num, x_cat)))
print(dcn_mix((x_num, x_cat)))
```

Figure7: Pre-train Testing

Times	Type of cross-layer	result
1	vector	tensor([-0.1580, 0.6446], grad_fn=<SqueezeBackward1>)
2	matrix	tensor([-0.0622, -0.7697], grad_fn=<SqueezeBackward1>)
3	mix	tensor([0.1216, 0.5322], grad_fn=<SqueezeBackward1>)

Table 2: the result of Pre-train Testing

And successfully get the results.

For the next work, I will use the post-train testing, including:

Invariant tests - examine how the input data is changing without compromising the machine learning model's overall performance.

Directional tests - a kind of hypothesis testing in which the testing's direction is decided upon beforehand.

Minimum functional tests - are used to determine whether the software or model is operating in accordance with the required dataset.

And also need to consider Pipeline testing in the future.

3.3 Design and Implementation

3.3.1 Data processing

In data processing, first of all, I will start with http://www2.informatik.uni-freiburg.de/~Cziegler/BX/This website downloads the data set of csv. After decompressing, it is found that there are three files: BX-Books.csv, BX-Users.csv, and BX-Book-ratings.csv. The three files are loaded with python, and a part of the data is previewed. The first line of data that is not needed is deleted, the data type is converted, and the unreasonable data is deleted (for example, unreasonable user age, unreasonable book publishing year), Then check the blank value of the book and fill in the blank value.

Then the data is integrated, combining users and rankings through "user_id", and then combining the merged files with book data through "isbn" to form a complete data set.

3.3.2 model design

In terms of model, I first studied the paper of DCN model, and then learned and implemented the model according to the file of the model implemented by the pythory and api. Then I studied the paper of DCN-V2, and tried to improve the common DCN model cross-layer

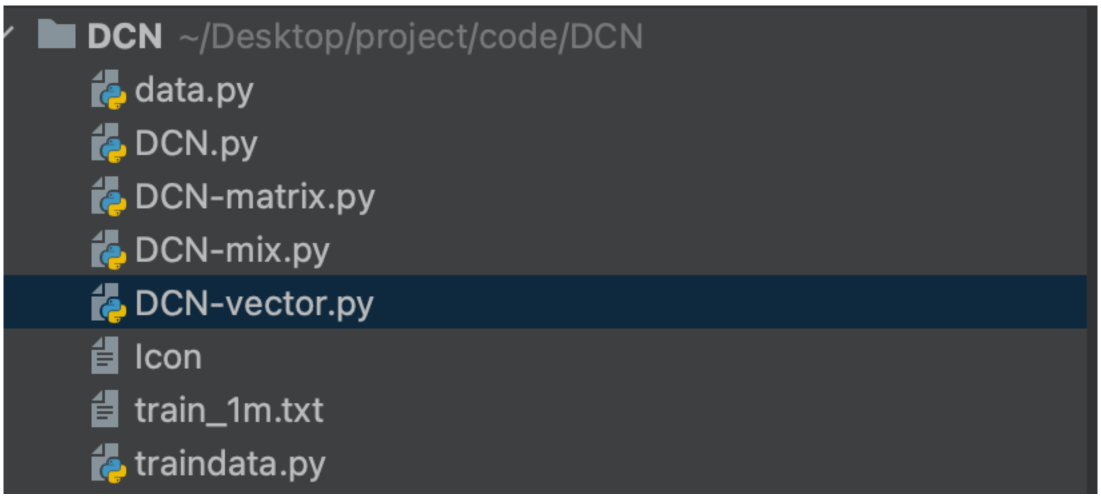

Figure8: model structure

4 Project Management

4.1 Activities

Here are the activities to accomplish each goal:

Object	Activities	complete	uncompleted
A) completes the research, study, and research of the recommender system.	a) Conduct a systematic search of similar software b) Create a feature comparison table c) Complete a literature search	ALL	NULL

	d) Perform a literature review e) Conduct a user survey		
B) completes the learning and research of deep learning.	a) learning online course of deep learning b) research of the model c) collection of the model implements d) learn the main theory and accomplish one by myself	ALL	NULL
C) collects appropriate data for analysis and evaluation.	a) find some articles about processing the data b) find the book dataset c) Analyze the data set, according to different characteristics d) Divide the dataset into training and testing sets	a), b), c)	d)
D) uses suitable deep learning models in combination with recommender systems.	a) create the Deep & Cross Network (DCN) model b) Use the training set for model training c) Test with the test set	a)	b), c)
E) implemented and tested.	a) Use the different dataset to test the whole system		a)

Table 3: Activities

4.2 Schedule

That is my time management and Gantt chart to show the activities and their deadlines:

Task	Start Date	End Date	Duration
Prepare Gantt	2022/10/30	2022/11/7	8
Registration Form and Ethics Form	2022/10/24	2022/10/31	7
Prepare Project Proposal	2022/10/26	2022/11/11	16
Literature Analysis	2022/10/18	2022/11/3	16
Requirements gathering	2022/10/31	2022/12/13	43
Requirements modeling	2022/11/15	2023/2/10	87
Literature Review	2022/10/18	2022/12/7	50
Finish the Progress Report	2022/11/12	2023/1/11	60
Testing & Evaluation	2023/1/1	2023/3/20	78
Write Final Report	2023/1/25	2023/3/20	54
Design and build	2022/11/15	2023/4/3	139
Implementation	2023/1/1	2023/4/10	99

Figure 9: time management

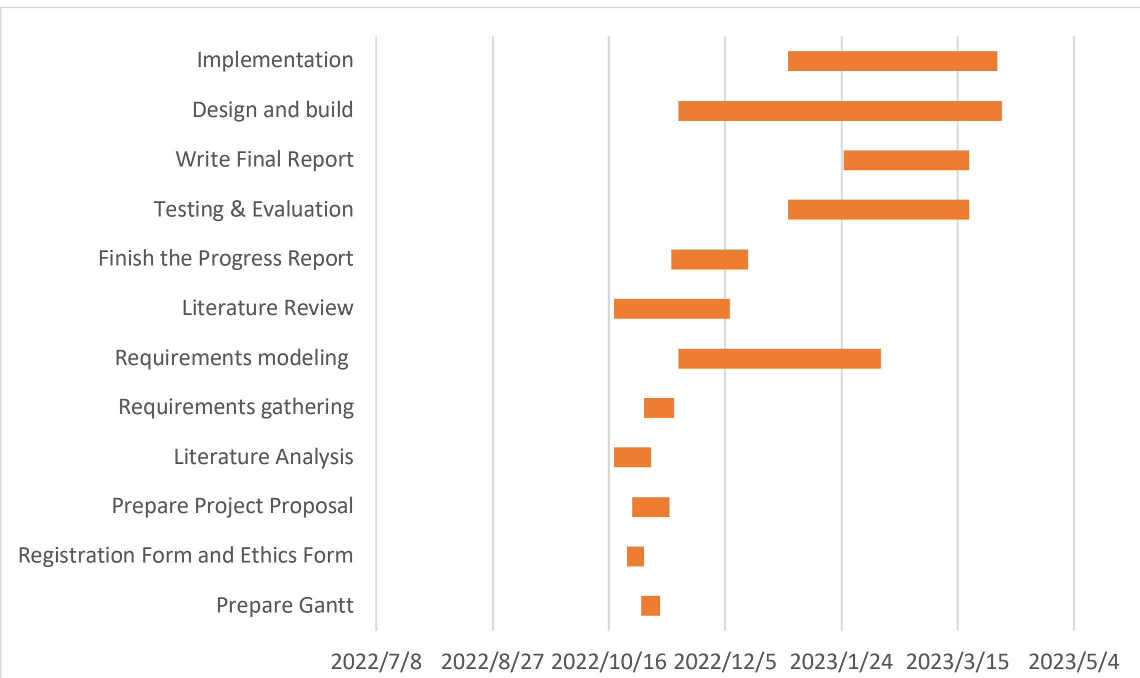

Figure 10: Gantt chart

4.3 Project Version Management

I use both local storage and Google Cloud to manage my projects, and the papers and data are stored locally before being uploaded to a folder in the cloud and synced. When I use colab, It will automatically upload the code to Google Cloud after completing the code in the cloud, and then back it up to a local file for saving.

Figure11: Upload the folder to the google drive

4.4 Project Data Management

Create a local folder and upload it through Google Cloud Service

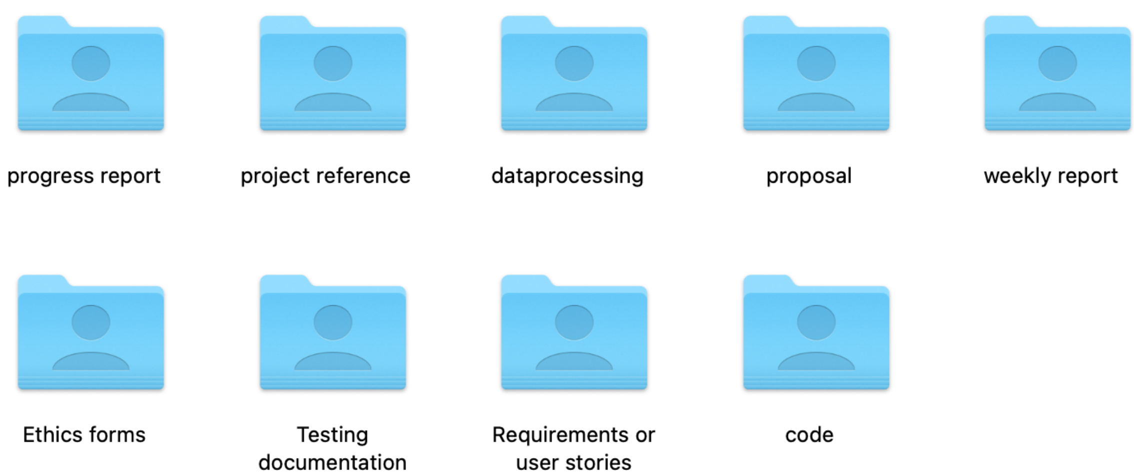

Figure 12: local folder

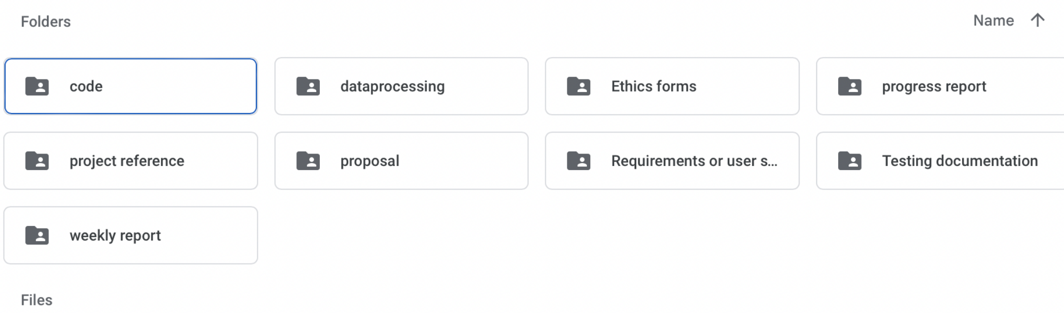

Figure 13: Cloud disk folder

4.5 Project Deliverables

Submitted: progress report, project reference, dataprocessing, proposal, weekly report, ethics forms, testing documentation

Not submitted: final report, project code, dataset, testing data, requirements, or user stories

5 Professional Issues and Risk:

5.1 Risk Analysis

That is my risk table to shou the risk analysis:

Risk ID	Potential Risk	Cause ID	Potential Causes	Severity	Likelihood	Risk	Mitigation ID	Mitigation
R1.1	Missed deadline	C1.1.1	Illness	1	3	3	M1.1.1	Register exceptional circumstances if ill.
		C1.1.2	Cannot choose topic	1	1	1	M1.1.2	Conduct research early and meet supervisor
		C1.1.3	Poor time management	4	3	12	M1.1.3	Make a Gantt plan early
R1.2	Feature creep	C.1.2.1	Over-ambitious project spec.	3	2	6	M1.2.1	Discuss plan with supervisor early. Create basic (must-have) goals and enhancements (nice-to-have).
R1.3	Software bugs	C1.3.1	Non-modular design	1	3	3	M1.3.1	Create highly modular desigh before implementation
		C1.3.2	Poor test plan	4	3	12	M1.3.2	Create test plan at start
R1.4	Loss of data	C1.4.1	Poor version control	4	4	16	M1.4.1	Implement version control strategy at start.
R1.5	Computer crashed	C1.4.2	Insufficient hardware, too large data set	3	4	12	M1.4.2	use cloud host
R1.6	Environmental error reporting	C1.4.3	package loss	4	1	4	M1.4.3	Implement the coding enironment at start.

Table4: risk

5.2 Professional Issues

5.2.1 Legal

Since this project uses personal data for commercial activities, all requirements of Privacy and Electronic Communications (PECR) Directive on data privacy must be met. Because this dataset contains only basic user information, these requirements are met. We need to take the utmost care in terms of privacy and data untraced ability.

In addition, if any information and data are leaked, the machine will bear legal responsibility.

5.2.2 Social

Because the project uses computers to collect and calculate user data, this brings a social problem, that is, whether the public is willing to accept this technology to collect public information in the background. Some users may feel uncomfortable with the recommendation made by the computer.

In order to safeguard the public interest, users must be able to decide whether they want to use this technology for their daily web browsing. They need to inform users and provide options that can be turned off.

5.2.3 Ethical

An ethical issue of this project is how to use these models if they are applied to book websites. This is important because the calculations generated by these models will read a lot of user information. This may cause privacy concerns, that is, whether the technology should only be used for book recommendations, or whether users need to test data acquisition when using the technology. If the user feels that the technology obtains too private information, this may also become a legal issue.

In addition, the use of patient data in the training of machine learning models may have privacy problems. Some users may object to recommending books to them in this way and may need the user's consent to train the model or use the model to help recommend.

5.2.4 Environmental

Training deep learning models may require a lot of resources and, if done regularly, may have an impact on the environment. Since the project only trains each model for each dataset type once, it minimizes resource use.

6 References

[1] L. E. Pilli and J. A. Mazzon, “Information overload, choice deferral, and moderating role of need for cognition: Empirical evidence,” *Revista de Administração*, vol. 51, no. 1, pp. 036–055, 2016, doi: 10.5700/rausp1222.

[2] P. Resnick and H. R. Varian, “Recommender Systems,” 1997. [Online]. Available: http://www.firefly.com

[3] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering,” *IEEE Internet Comput*, vol. 7, no. 1, pp. 76–80, 2003, doi: 10.1109/MIC.2003.1167344.

[4] B. Smith and G. Linden, "Two Decades of Recommender Systems at Amazon.com," in IEEE Internet Computing, vol. 21, no. 3, pp. 12-18, May-June 2017, doi: 10.1109/MIC.2017.72.

[5] P. Karthikeyan and C. Tejasvini, “Review of Movie Recommendation System,” in 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022, vol. 1. Accessed: Oct. 31, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9785014/

[6] A. Singhal, P. Sinha, and R. Pant, “Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works,” 2017.

[7] R. Wang, G. Fu, B. Fu, and M. Wang, “Deep & cross network for ad click predictions,” in *2017 AdKDD and TargetAd - In conjunction with the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2017*, Aug. 2017. doi: 10.1145/3124749.3124754.

[8] J. Scott Armstrong , “PRINCIPLES OF FORECASTING: A Handbook for Researchers and Practitioners.” Springer New York, NY, pp. 61-70, 2001, doi: https://doi.org/10.1007/978-0-306-47630-3

[9] M. Sun, G. Lebanon, and P. Kidwell, “Estimating Probabilities in Recommendation Systems,” 2011.

[10] H. Liu, J. He, T. Wang, W. Song, and X. Du, “Combining user preferences and user opinions for accurate recommendation,” *Electron Commer Res Appl*, vol. 12, no. 1, pp. 14–23, 2013, doi: 10.1016/j.elerap.2012.05.002.

[11] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," in IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, June 2005, doi: 10.1109/TKDE.2005.99.

[12] Sung Ho Ha, "Helping online customers decide through Web personalization," in IEEE Intelligent Systems, vol. 17, no. 6, pp. 34-43, Nov.-Dec. 2002, doi: 10.1109/MIS.2002.1134360.

[13] K. Verbert *et al.*, “Context-aware recommender systems for learning: A survey and future challenges,” *IEEE Transactions on Learning Technologies*, vol. 5, no. 4. pp. 318–335, 2012. doi: 10.1109/TLT.2012.11.

[14] R. J. Mooney, “Content-Based Book Recommending Using Learning for Text Categorization,” 2000.

[15] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” 1998.

[16] Balabanović, M. and Shoham, Y,” Fab: content-based, collaborative recommendation.” Communications of the ACM, 1997, pp.66-72.

[17] H. T. Cheng *et al.*, “Wide & deep learning for recommender systems,” in *ACM International Conference Proceeding Series*, Sep. 2016, vol. 15-September-2016, pp. 7–10. doi: 10.1145/2988450.2988454.

[18] L. Zhang, T. Luo, F. Zhang, and Y. Wu, “A Recommendation Model Based on Deep Neural Network,” *IEEE Access*, vol. 6, pp. 9454–9463, Jan. 2018, doi: 10.1109/ACCESS.2018.2789866.

[19] X. Xu, “Matrix factorization recommendation algorithm based on deep neural network,” in *2019 2nd International Conference on Information Systems and Computer Aided Education, ICISCAE 2019*, Sep. 2019, pp. 320–323. doi: 10.1109/ICISCAE48440.2019.221643.

[20] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems*, Sep. 2016, pp. 191–198. doi: 10.1145/2959100.2959190.

[21] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, “Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks,” Jan. 2017, [Online]. Available: http://arxiv.org/abs/1701.02720

[22] S. Okura, Y. Tagami, S. Ono, and A. Tajima, “Embedding-based news recommendation for millions of users,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2017, vol. Part F129685, pp. 1933–1942. doi: 10.1145/3097983.3098108.

[23] R. Wang *et al.*, “DCN V2: Improved deep & cross network and practical lessons for web-scale learning to rank systems,” in *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, Apr. 2021, pp. 1785–1797. doi: 10.1145/3442381.3450078.