



Multi-Feature Fusion Book Recommendation Model Based on Deep Neural Network

Zhaomin Liang and Tingting Liang*

College of Artificial Intelligence, Nanning University, Nanning, Guangxi, 530200, China

*Corresponding Author: Tingting Liang. Email: whnmyt@qq.com

Received: 24 October 2022; Accepted: 17 February 2023; Published: 26 May 2023

Abstract: The traditional recommendation algorithm represented by the collaborative filtering algorithm is the most classical and widely recommended algorithm in the practical industry. Most book recommendation systems also use this algorithm. However, the traditional recommendation algorithm represented by the collaborative filtering algorithm cannot deal with the data sparsity well. This algorithm only uses the shallow feature design of the interaction between readers and books, so it fails to achieve the high-level abstract learning of the relevant attribute features of readers and books, leading to a decline in recommendation performance. Given the above problems, this study uses deep learning technology to model readers' book borrowing probability. It builds a recommendation system model through the multi-layer neural network and inputs the features extracted from readers and books into the network, and then profoundly integrates the features of readers and books through the multi-layer neural network. The hidden deep interaction between readers and books is explored accordingly. Thus, the quality of book recommendation performance will be significantly improved. In the experiment, the evaluation indexes of HR@10, MRR, and NDCG of the deep neural network recommendation model constructed in this paper are higher than those of the traditional recommendation algorithm, which verifies the effectiveness of the model in the book recommendation.

Keywords: Book recommendation; deep learning; neural network; multi-feature fusion personalized prediction

1 Introduction

With the application of new information technology represented by new technologies such as big data and artificial intelligence, information technology has promoted the continuous development of business informatization. Most modern libraries say goodbye to the traditional means of manual registration, borrowing, and returning. They have applied the book information system to carry out book borrowing and returning related business so that readers can more conveniently check and borrow books [1]. However, with the continuous expansion of the number of books in the library, readers may face massive book data and cannot effectively find the books they need. There is an



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

urgent need for a means to establish an efficient connection between readers and books [2]. The recommendation system is effective compared to Internet navigation or search engine methods.

The recommendation system first collects readers' historical borrowing data and then recommends books that readers may be interested in after analysis. The traditional recommendation algorithm represented by the collaborative filtering algorithm is the most classic and widely used in practical industrial applications. Most book recommendation systems also use this algorithm [3]. However, the traditional recommendation algorithms represented by the collaborative filtering algorithm cannot deal with the recommendation performance degradation caused by data sparsity. Moreover, it only uses the shallow feature design of the interactive behavior of readers and books. It cannot achieve high-level abstract learning of the related attributes of readers and books.

Given the above problems, this study first converts the embedded representation of user information and book information to be used as the neural network input. A text convolution neural network extracts text information for book titles, and deep learning technology is used to build a network recommendation model for predicting recommendations. This model solves the traditional recommendation problem and improves book recommendation quality. The subsequent chapters of this paper will be arranged as follows: Chapter 2 will introduce the review of relevant studies, Chapter 3 will introduce the specific construction methods of the model, Chapter 4 will introduce the experiments conducted to verify the model's effectiveness, and Chapter 5 will summarize.

2 Review of the Related Studies

The recommendation system is a helpful method of dealing with information overload, providing an efficient way for users to find their favorite items. In addition to being widely used in e-commerce, the recommendation system is also valued and applied by libraries. In 1998, the Cornell University Library developed a recommendation system called Mylibrary [4], which could recommend books or journals to readers according to particular rules. In China, at the end of 1999, the "China Digital Library Demonstration System" project hosted by the intelligent computer expert group of the National 863 Program carried out an exposition and research on the problem of personalized services in digital libraries [5]. It marks the beginning of the personalized service of digital libraries in our country.

Traditional recommendation system algorithms include collaborative filtering, content-based, and hybrid recommendation algorithms. The collaborative filtering algorithm is the most widely used in all areas. It can be seen in personalized recommendation service providers of various e-commerce websites. Still, the collaborative filtering algorithm also shows problems, such as data sparsity and cold-start. In addition, the traditional collaborative filtering algorithm cannot learn the in-depth features of users and items. Although the academic community has proposed some methods to alleviate these difficulties [6–8], the problems of the recommendation system still bring considerable challenges to the recommender system.

Because of its revolutionary progress, deep learning has received extensive attention in many fields in recent years [9]. It also has excellent potential in the application of recommendation systems. In terms of the recommendation system, deep learning can automatically extract features from massive data of users and items and can learn in-depth features of them. Deep learning can map data to the same feature space through automatic feature learning from data with different structures and contents, which can uniformly represent data. To a certain extent, it can deal with the data sparsity and cold-start problems of recommendation systems [10]. Cheng and colleagues proposed a Wide & Deep Learning model [11], which uses multi-source heterogeneous data such as user, context, and

item characteristics as the basis for mobile APP recommendation. The model has both high memory and generalization ability. The object embedding representation based on the shallow neural network model has been widely used in recent years. The embedding representation method was initially applied in natural language processing. It is a by-product of Embedding Models [12]. While gradually used to express the word vector representation in natural language, the embedding model is generally simple and effective. It is quickly used to construct the user behavior sequence pattern in the recommendation system. Barkan and colleagues regard the item as a word and users' behavior sequences as a set [13]. Through shallow neural network training, the embedding representation of items is obtained, and the similarity between items is calculated to achieve a recommendation.

3 Design of the Model

This study constructs a deep learning book recommendation model using a deep neural network to solve the traditional recommendation algorithm's problem of data sparsity and cold-start. Based on the library's lending data, this model uses reader feature attributes and book feature attributes to learn the hidden features between readers and books and deeply explore the interaction between readers and books to improve recommendation performance.

3.1 *The Overall Framework and Ideas of the Model*

After data preprocessing, according to the characteristics of book borrowing data, this study builds a deep learning book recommendation network based on various features. The overall framework of the model is shown in Fig. 1. The main purpose of this book recommendation model is to use deep learning to automatically extract and combine features of readers and books. Then abstract low-level attribute features of readers and books into high-level attribute features, and finally use multi-layer neural networks to non-linearly fit the features, to explore the deep relationship between readers and books. The flow of the model is as follows:

- (1) First, the reader attribute data and the book attribute data are input into the embedding layer to obtain the embedded vector of the reader attribute and the book attribute. Embedding is a successful application of deep learning, which represents discrete variables as continuous vectors. Embedding representation uses a low-dimensional vector to describe an object, which can be a word, a commodity, a book, etc. The property of this embedded vector is to make objects corresponding to vectors with similar distances have similar meanings.
- (2) Second, the embedded vector representations of reader and book attributes are spliced and fused, respectively. Among these, the text information features of book titles and other embedded features of book attributes obtained after being processed by a convolutional neural network [14] are also spliced and fused. The fused reader feature vectors and the book feature vectors are then fully connected to obtain the feature representations of the reader and the book.
- (3) Third, the fused reader features and the book features are input into a multi-layer neural network to predict the probability of readers borrowing books. The cross-entropy loss function trains the model and adjusts the parameters. Also, the Adam function is used for optimization.
- (4) Last, sort the obtained predicted probability values, and recommend books to readers according to the probability.

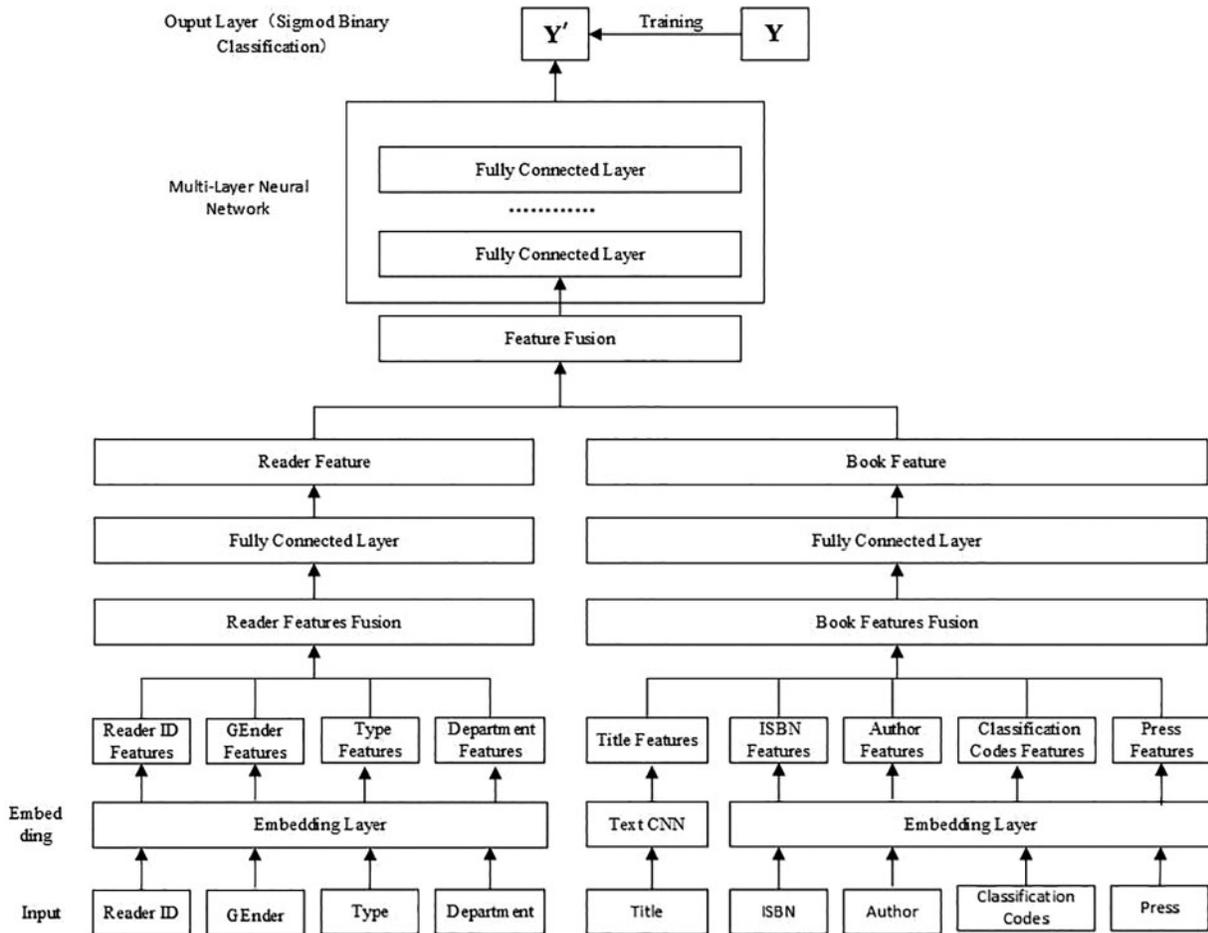


Figure 1: Framework of deep neural network book recommendation model

3.2 Input Feature Fusion into the Multi-Layer Neural Network

Fig. 1 shows that the embedded vector can be obtained after the reader and book feature information pass through the embedding layer. The book title also gets its vector representation after passing through the convolutional neural network. Then, the feature fusion will be performed through the network. We suppose that the attribute feature of the reader after passing through the embedding layer is $x: \{x_1, x_2, \dots, x_n\}$. The represents the vector of the attribute feature after the embedded representation of a reader. The attribute feature of the book attributes after passing through the embedding layer and the text neural network layer is $y: \{y_1, y_2, \dots, y_n\}$, where y_i represents a vector of book embedded representation or text extraction.

Integrate various attributes of users to obtain the features of readers u_i :

$$u_i = \text{concatenate}(x) \quad (1)$$

Integrate various attributes of books to obtain the features of books b_i :

$$b_i = \text{concatenate}(y) \quad (2)$$

Among them, u_i represents the user attribute feature obtained after feature fusion, b_i represents the book attribute feature obtained after feature fusion, and $concatenate()$ represents the fusion of various attributes of readers or books.

After acquiring the features of readers and books from the above steps, the features of readers and books are fused and input into a multi-layer neural network to predict whether readers are going to borrow books. According to the model framework shown in Fig. 1, before the reader features and book features enter the multi-layer neural network, they must be fused into input vectors. The formula is shown in (3):

$$x_0 = concatenate(u_i, b_j) \quad (3)$$

The function $concatenate()$ means to splice reader features and book features. x_0 is input to the multi-layer neural network, the schematic topology of the multi-layer neural network is shown in Fig. 2

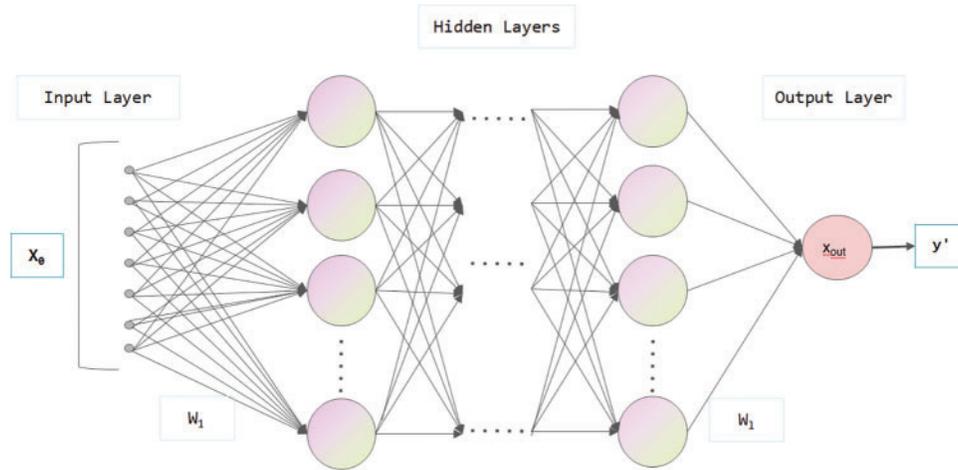


Figure 2: Schematic topology of the multi-layer neural network

The first hidden output value expression is:

$$x_1 = f(W_1 x_0 + b'_1) \quad (4)$$

Among them f represents activating the function. Here we use the Relu function as the activation function of the hidden layer. W_1 represents the weight matrix between the input layer and the first hidden layer, and b'_1 represents the bias vector. The function formula of the output layer is:

$$y' = x_{out} = f(W_l x_{l-1} + b'_l) \quad (5)$$

The activation function of the output layer is the sigmoid function.

The objective function is the cross-entropy loss function:

$$L = y \log y' + (1 - y) \log (1 - y') \quad (6)$$

To train the model and recommend books to readers according to the probability of borrowing predicted by the model.

4 Experimental Design and Results Analysis

4.1 Introduction to Experimental Datasets

The dataset used in this study is the borrowing data of the readers of Nanning University Library. After processing and exporting from the library information system, the data information shown in Fig. 3 is obtained. It mainly includes two parts. One part is the reader's information, such as the reader ID, the reader's name, the reader's type, and the reader's unit. Seldom off-campus readers and secondary vocational student readers come to borrow books. Thus, these two kinds of reader types are ignored in this paper. We only remain two types of teachers and students. Regarding reader units, the current study classifies all administrative departments, such as the Academic Affairs Office and the Personnel Office, except for secondary colleges, into executive departments. The other part is the book information, including the title, author, ISBN, publisher, and book classification number. When the system exports the information, a reader's book renewal is also counted as a borrowing record, so there will be duplicate borrowing data in the data set. We only retain one reader's borrowing record of a book through deduplication. After exporting and data cleaning, the borrowing data includes 2,487 readers and 79,551 books, totaling 169,427 borrowing data.

ReaderID	Reader Gender	Reader Type	Reader Department	Title	Author	ISBN	Classification Codes	Press
20162231	0	2	College of Mechanical and Electrical Engineering	Rainy Season Won't Come Again	San Mao	978-7-5302-0966-0	I267	Beijing October Literature and Art Press
20162231	0	2	College of Mechanical and Electrical Engineering	Genius and Psychotic	Ming Gao	978-7-307-07542-9	I253	Wuhan University Press
20162231	0	2	College of Mechanical and Electrical Engineering	Tatsukawa Tokio	Higashino Keigo(Japanese)	978-7-5442-4585-2	I313.45	Nan Hai Publishing Co.
20162231	0	2	College of Mechanical and Electrical Engineering	Chicken Soup for the Soul	Huling Hu	978-7-80723-567-5	I16	Distant Publishing House
20162231	0	2	College of Mechanical and Electrical Engineering	Grimm's Fairy Tales	Die Gebrüder Grimm(German)	7-80154-849-3	I18	Peace Press
20162231	0	2	College of Mechanical and Electrical Engineering	Running snail	Die Zhi Ling	978-7-5059-6581-2	I247.57	China Federation of Literary and Art Circles Press
20162231	0	2	College of Mechanical and Electrical Engineering	Almost a Hero	Jiajia Zhang	978-7-80148-866-4	I247.57	Chaohua Press
20162231	0	2	College of Mechanical and Electrical Engineering	No Double	Jiao Jiao	978-7-5054-1812-7	I247.5	Chaohua Press
20162231	0	2	College of Mechanical and Electrical Engineering	Transform into men and women	Zi Yue	978-7-80225-964-5	I247.59	Nova Press
20162231	0	2	College of Mechanical and Electrical Engineering	Farewell to Vienne	qin Shan	978-7-5470-2659-5	I247.7	Wanjuan Publishing Company

Figure 3: Borrowing data set of the library

To advance the experiment, this study uses the “leave-one-out” evaluation method, which has been widely used in several algorithm verification experiments [15–17]. In the borrowing dataset, for each reader, we keep his/her latest borrowing records in the test set and the remaining borrowing records in the training set. We follow the common method of experiments [18,19], randomly sample 99 books that the reader does not borrow as negative samples, and form 100 test samples together with his/her last borrowed book. Then, we rank the 100 samples and use the corresponding evaluation indicators to judge the performance of the sorted list.

4.2 Data Preprocessing

We need to tidy the data without abnormality to guarantee the recommender system's recommendation results quality. For this reason, we clean the borrowing data of readers and remove the out-of-spec and invalid data. In data cleaning, we follow the steps below:

- (1) First, we sort and filter the data with incomplete information in the borrowing data of readers. In particular, we pay attention to the completeness of the user's number and the ISBN attribute dictionary of the book.
- (2) Second, filtering the sample data of readers who seldom borrow books. It is difficult to capture readers' behavioral preferences and interests with too few borrowings. Therefore, after the overall analysis of the borrowing data, samples of readers with small borrowings are filtered and removed.
- (3) Third, sorting the classification numbers. The book classification numbers reflect the classification types of books and play an essential role in representing the features of books. Generally, books are classified according to the “Chinese Library Classification” [20]. According to

the “Chinese Library Classification,” books are divided into 22 categories, as shown in Table 1. Due to the finer classification, the sample data under this classification will be sparse. According to the actual situation of the sample, we only retain the sample data of the large category of letters.

- (4) Fourth, since some text data contains verbs connected with the author’s name, such as ‘Author’ and ‘Editor’ before or after the author’s name, it is necessary first to filter the data of authors by regularization rules. Then, we remove other symbols and words except for the author’s name before and after so that it only remains the author’s name. The processing effect is shown in Table 2. Since several authors may write a book, this paper only intercepts the first author as the input data of the recommendation model.
- (5) Fifth, since the text of the book title contains rich information, it is necessary to preprocess the text information of the book title as follows and then use it as the input of the neural network to extract the rich features of the content of the book. When preprocessing the text information of the book title, we first use the jieba participle to segment the book title text. After passing the jieba participle, the book title text is processed to remove the stop words. It aims to remove useless or meaningless words in the book title, such as auxiliary words, modal particles, etc. After removing the stop words, it is used as the input data of the neural network.

Table 1: Contents of Chinese library classification

A	Marxism, Leninism, Mao Zedong Thought, Deng Xiaoping Theory	B	Philosophy, Religion
C	Social Sciences	D	Politics, Law
E	Military Science	F	Economy
G	Culture, Science, Education, and Physical Education	H	Languages
I	Culture	J	Art
K	History, Geography	N	Natural Science
O	Mathematical Sciences, Chemistry	P	Astronomy, Earth Science
Q	Biology	R	Medicine, Health
S	Agricultural Science	T	Industrial Technology
U	Transportation	V	Aerospace
X	Environmental Science, Safety Science	Z	Comprehensive Books

Table 2: Examples of contrastive effects of book authors before and after processing

Before processing	After processing
Author Zhang Lijun	Zhang Lijun
Editors Wang Hailin, Zhang Yuxiang, et al.	Wang Hailin, Zhang Yuxiang
Editor An Hongzhang	An Hongzhang
Authors Liu Jicai, Tang Sisi, Liu Yongsheng	Liu Jicai, Tang Sisi, Liu Yongsheng
Author Charles Dickens (England)	Charles Dickens (England)
Editor Mirror Photography	Mirror Photography
Editor Chen Weizhen, Wei Yuping	Chen Weizhen, Wei Yuping

(Continued)

Table 2: Continued

Before processing	After processing
Editor Huang Daqing, Yin Xueyun	Huang Daqing, Yin Xueyun
Author Osamu Dazai (Japan)	Osamu Dazai (Japan)
Editors Zhan Xuejun et al.	Zhan Xuejun et al.

4.3 The Experiment Setting

(1) The hardware

A single laptop: CPU: i7-7700HQ; Graphics card: GTX 1050TI; Memory 8G; Hard disk 1TB.

(2) The software

Python version: Python 3.6.3

jieba version: jieba 0.39

Anaconda version: Anaconda 5.0.1

Windows version: Windows 10 64-bit

Tensorflow Version: Tensorflow 1.9.0 GPU

The construction and training of multi-layer neural networks use the Tensorflow framework, one of the most popular machine learning and deep learning frameworks [21,22]. Google develops Tensorflow, and many of Google's artificial intelligence businesses are implemented by Tensorflow [23].

4.4 Algorithm Comparison

To verify the validity and superiority of the model proposed in this study, we compare the following typical traditional recommendation algorithms widely used in developing recommendation systems.

- (1) The first one is the popularity-based algorithm, which is used in the early stage of the development of the recommendation system. The popularity-based algorithm is straightforward and rude, similar to major news. It recommends items to users based on their popularity, such as hot news, trending topics on Weibo, etc. [24]. This study uses the popularity-based algorithm to recommend the most popular books with the highest borrowing rate.
- (2) The other one is an Item-based collaborative filtering algorithm. This algorithm is currently the most widely used in practice. The algorithm utilizes the co-occurrence law of items in user behavior, analyzes user behavior, and recommends items similar to the items he/she liked before [25].
- (3) To verify the effectiveness of the auxiliary information for the recommendation of the deep neural network model, this study uses the deep neural network recommendation model that only contains behavioral interaction between users and items. No auxiliary information, such as attribute features of users or items, is added to the model [26].

4.5 Evaluation Indicators

In terms of selecting evaluation indicators, this study chooses to use HR (Hit Ratio), MRR (Mean reciprocal rank), and NDCG (Normalized Discounted Cumulative Gain). HR reflects the recommendation accuracy of the recommendation list. MRR and NDCG reflect the recommendation

quality of the recommendation list. These two indicators suggest that people hope that the positions of the items they are interested in in the recommendation list are always at the front.

(1) Hit Ratio. This indicator is commonly used to measure the recall rate in top-K recommendations [27]. The calculation formula is as follows:

$$HR@K = \frac{\text{NumberofHits@K}}{|\text{GT}|} \quad (7)$$

The denominator GT is the test of all sets, and the numerator is the sum of the number of items that belong to the test set in the top K items recommended by each user. This indicator aims to measure the recall rate.

(2) Mean reciprocal rank. Its core idea is that the quality of the recommendation list is related to the position of the first item that correctly matches the user's interest [28]. On the first recommendation list, the higher the item the user is interested in, the better the result.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (8)$$

Q is the set of users in the sample, and $|Q|$ is the number of users in the sample set. $rank_i$ represents the rank of the first item that correctly matches the user's interest preference.

(3) Normalized Discounted Cumulative Gain. It is an evaluation indicator to measure the quality of sorting, which considers the correlation of all elements [29,30]. The formula is as follows:

$$NDCG = Z_K \sum_{i=1}^K \frac{2^{r_i} - 1}{\log 2^{(i+1)}} \quad (9)$$

(3) Normalized Discounted Cumulative Gain. NDCG is an evaluation indicator to measure the quality of sorting, which considers the correlation of all elements. The formula is as follows:

r_i represents the level correlation at the i -th position, which can generally be handled with 0/1. If the item at this position is in the test set, then $r_i = 1$; Otherwise, $r_i = 0$. In addition, Z_K is the normalization coefficient, representing the reciprocal of the sum in the best case of the following cumulative sum formula: the sum of the following formulas when $r_i = 1$ is satisfied. It aims to calculate the value of i by NDCG within 0–1.

4.6 Result Analysis

The data set used in this experiment belongs to implicit feedback. In the model, the samples borrowed by readers are positive samples and the books that readers do not borrow need to be selected as negative samples. According to the borrowing amount of each reader, we randomly choose the books that the reader has not borrowed as a negative sample and iterate them through 15 rounds. The hidden layer parameters are configured with 32RuLU + 16ReLU + 8ReLU, and the ratio of positive and negative samples of readers is 1:1, 1:2, ..., 1:10, etc. We observe the impact of different ratios of negative sample sampling on the model's performance with a recommended length of 10. Experiments were carried out on the dataset, and the HR@10 values of the experiments are shown in Fig. 4 below.

The experimental results suggest that taking only one negative sample per positive sample under this dataset is not enough to achieve the best performance. It can be identified that more negative sampling is helpful to the improvement of system performance. However, after the negative sampling is 6, the recommendation effect of the model gradually flattens, and the increasing effect is not apparent anymore. This shows that excessive negative sampling will not bring about a significant increase in the

recommendation effect and may even have a bad impact on the recommendation effect of the model. In addition, excessive negative sampling will increase the number of training samples in the training set and the training time cost. Considering the effect improvement brought by the increase in negative samples and the cost of system calculation, the ratio of positive and negative samples to 1:6 is very cost-effective.

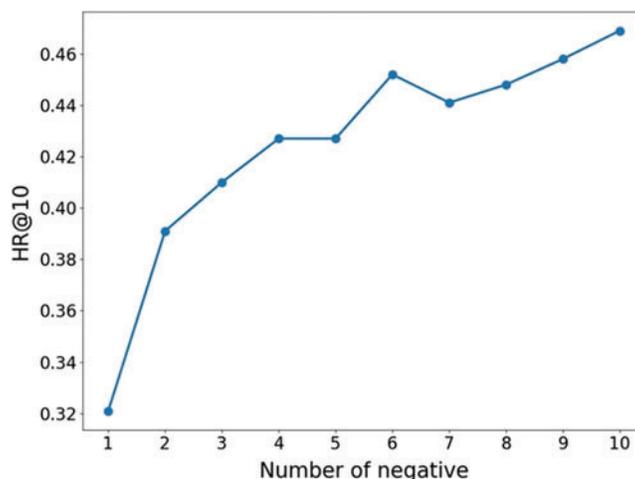


Figure 4: HR@10 curves with different negative samples

In analyzing the influence of the hidden layer in the multi-layer neural network on the model recommendation effect, we use different hidden schemes for analysis and comparison. The parameter configuration is that the negative sample parameter is set at a ratio of 1:6, and the number of iterations is 15 times. The results are shown in Table 3. It suggests that the recommendation effect is the worst when no hidden layer exists. Simply interacting with the reader and book feature vectors cannot get a good recommendation effect. From the table, we can see that increasing the width and depth of the hidden layer helps improve the model effect. It shows that increasing the hidden layer's width and depth can enable users and item features to interact more deeply, thereby improving the recommendation effect. Here, 32 ReLU+16 ReLU+8 ReLU recommends the best results.

Table 3: Effect of different hidden layers on recommendation performance

Hidden layer	HR@10
None	0.262
8 ReLU	0.405
16 ReLU	0.420
32 ReLU	0.425
16 ReLU + 8 ReLU	0.427
32 ReLU + 16 ReLU	0.430
32 ReLU + 16 ReLU + 8 ReLU	0.452
ReLU	

Epoch, the number of training iterations of the model, may have a specific impact on the model's performance. In this study, the iterative training is carried out on the premise that the ratio of positive and negative samples is 1:6. The impact of the number of iterations on the model effect is observed. The convergence of the loss function when the model is trained for 50 iterations is shown in Fig. 5 below:

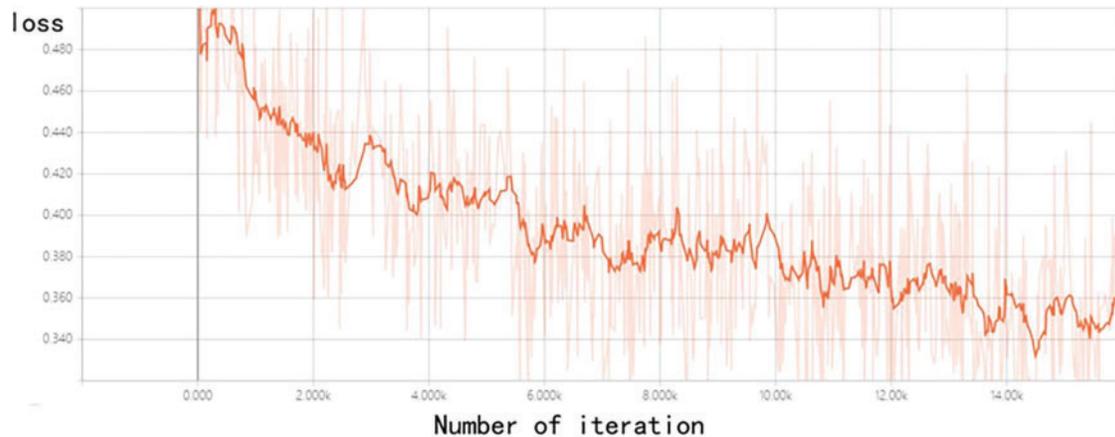


Figure 5: Loss curve of experimental training

According to the training results of the experiment, the squared error is about $0.6\times$ at the beginning. After 50 iterations, the error tends to be flat after the squared difference drops to about $0.3\times$. It suggests that after integrating reader features and book features, the feature representation of the data is complete and can be better fitted. It can be seen from Fig. 4 that the loss tends to converge as the iteration increases.

When observing the effect of different iterations of training on the performance of HR@10, the experimental values are shown in Fig. 6:

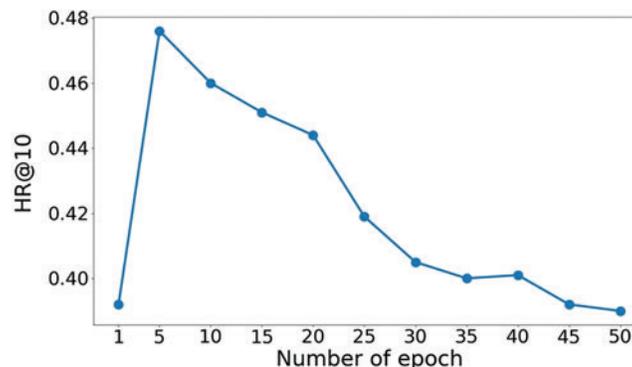


Figure 6: HR@10 curves with different epochs numbers

In the dataset, we can find that the number of iterations continues to increase, and the HR@10 of the dataset also continues to grow, indicating that the recommendation effect improves with the number of iterations. After 5 rounds of iterations, the HR@10 value decreases as the number of iterations increases. It means that too many iterations will cause overfitting, and the effect of model

recommendation will be weakened. Therefore, the model that chooses the number of iterations as 5 rounds is the best.

To verify the recommendation performance of the model proposed, this paper compares it with the traditional recommendation algorithm commonly used in the industry with a recommended length of 10. According to the previous experiments, we determine the parameter: the negative sampling ratio is 1:6, the loop iteration is 5 rounds, and the hidden layer uses the result of 32 RuLU + 16ReLU + 8ReLU. The popularity-based algorithm is referred to as most_TOP, and the item-based collaborative filtering algorithm is referred to as item_CF. The multi-layer neural network recommendation model proposed in this study is referred to as dnn_REM. To verify that integrating the auxiliary information of readers and books in this model can improve the effectiveness of the network model, we also set up a network recommendation model dnn_ONLY that only uses user and book interaction behavior as input compared with it. Of course, the parameter settings are the same during the comparison.

The experimental results are shown in [Table 4](#) below. The performance of the popularity-based recommendation algorithm is poorer than other recommendation algorithms in HR@10 and NDCG@10, reflecting the ranking quality. The popularity-based algorithm only recommends popular books to readers, and few personalized recommendation methods are used. It shows that using personalized recommendations is essential. The traditional item-based collaborative filtering algorithm only uses the linear interaction of the co-occurrence relationship between readers and books. It does not add any auxiliary information, so the effect is worse than the multi-layer neural network recommendation model proposed in this study.

Table 4: Performance of different algorithms on data set

Algorithm	HR@10	MRR@10	NDCG@10
most_TOP	0.274	0.113	0.150
item_CF	0.342	0.202	0.235
dnn_ONLY	0.449	0.304	0.338
dnn_REM	0.476	0.220	0.280

The multi-layer neural network recommendation model proposed in this study performs better in MRR@10 and NDCG@10 than in traditional personalized recommendation algorithms. The reason may be it adds multiple auxiliary attribute information of readers and books and fully non-linear fusion interaction of reader and book features through multi-layer neural networks. This reflects that the application of deep learning technology in book personalization is effective and superior. Under the same parameter configuration, dnn_ONLY without reader and book auxiliary information is compared to dnn_REM with reader and book auxiliary information. When the recommendation quality is roughly the same, the recommendation accuracy of the deep neural network model integrating reader and book auxiliary information is significantly higher than the deep neural network recommendation model using only the interaction behavior of readers and books. It shows that the deep neural network book recommendation model helps improve the recommendation effect after integrating the auxiliary information of auxiliary readers and books.

The length of the recommendation list also impacts the recommendation results. In this experiment, different recommendation lengths topN were set to 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. We observed the impact of different recommendation list lengths on the effect of recommendations. The experimental results are shown in [Table 5](#).

Table 5: Performance of evaluation indicators with different recommendation lengths

TopN evaluation indicators	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10
HR	0.118	0.187	0.247	0.293	0.335	0.364	0.395	0.423	0.447	0.476
MRR	0.118	0.153	0.173	0.184	0.193	0.197	0.202	0.205	0.208	0.220
NDCG	0.118	0.162	0.192	0.211	0.228	0.238	0.248	0.257	0.264	0.280

As seen from [Table 5](#), with the length of the recommendation list increasing, the HR, MRR, and NDCG values also increase, indicating the performance of the recommender system is affected by the length of the recommendation list. The increase in the length of the recommendation list is helpful for the performance of the recommendation system.

5 Conclusion

The current study uses deep learning to model the probability of readers borrowing books and constructs a multi-layer neural network recommendation model. After the information of readers and books is processed by the embedded table and text convolutional neural network, it is input into the multi-layer neural network to deeply explore the hidden deep related features between readers and books. The probability of readers borrowing books is predicted, the recommendation list is generated, and the personalized recommendations are completed. With the rigorous design of the experiment, the experimental results are compared and analyzed, which verifies the effectiveness and superiority of the deep learning multi-feature fusion recommendation model proposed in our study. It provides an effective way for a personalized recommendation for book borrowing.

However, this paper proposes that the model is only verified on the data in the actual work of the work unit, without using more extensive general recommended data. In the future, it can be considered to conduct research on the more general and extensive data of the model to improve its wider applicability.

Funding Statement: This work was partly supported by the Basic Ability Improvement Project for Young and Middle-aged Teachers in Guangxi Colleges and Universities (2021KY1800; 2021KY1804).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. Q. Li, Z. Y. Hao, L. Ding and X. Xu, "Research on the application of information technology of big data in Chinese digital library," *Library Management*, vol. 40, no. 8/9, pp. 518–531, 2019.
- [2] K. Anwar, J. Siddiqui and S. S. Sohail, "Machine learning-based book recommender system: A survey and new perspectives," *International Journal of Intelligent Information and Database Systems*, vol. 13, no. 2/3/4, pp. 231, 2020.
- [3] G. Ramakrishnan, V. Saicharan, K. Chandrasekaran, M. V. Rathnamma and V. V. Ramana, "Collaborative filtering for book recommendation system," in *Soft Computing for Problem Solving*. Singapore: Springer, pp. 325–338, 2020.

- [4] N. Pang, "A personalized recommendation algorithm for semantic classification of new nook recommendation services for university libraries," *Mathematical Problems in Engineering*, vol. 2022, no. 4, pp. 1–8, 2022.
- [5] H. H. Li, Z. Y. Ge, J. Zhang, F. Liu, C. H. Zhou *et al.*, "China digital library demonstration system," *Information and Documentation Services*, vol. 4, pp. 23–25, [In Chinese], 2001.
- [6] D. K. Chae, S. C. Lee, S. Y. Lee and S. W. Kim, "On identifying k-nearest neighbors in neighborhood models for efficient and effective collaborative filtering," *Neurocomputing*, vol. 278, no. 1, pp. 134–143, 2018.
- [7] H. Parvin, P. Moradi and S. Esmaeili, "TCFACO: Trust-aware collaborative filtering method based on ant colony optimization," *Expert Systems with Applications*, vol. 118, pp. 152–168, 2019.
- [8] G. Geetha, M. Safa, C. Fancy and D. Saranya, "A hybrid approach using collaborative filtering and content based filtering for recommender system," *Journal of Physics: Conference Series*, vol. 1000, no. 1, pp. 12101, IOP Publishing, 2018.
- [9] S. Saranya, M. Kumar, M. R. Ayyagari and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071–1092, 2020.
- [10] R. Xie, Z. Qiu, J. Rao, Y. Liu, B. Zhang *et al.*, "Internal and contextual attention network for cold-start multi-channel matching in recommendation," in *Int. Joint Conf. on Artificial Intelligence*, Yokohama, Japan, pp. 2732–2738, 2020.
- [11] S. Zhang, L. Yao, A. Sun and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.
- [12] Z. Y. Xiong, Q. Q. Shen, Y. J. Wang and C. Y. Zhu, "Paragraph vector representation based on word to vector and CNN learning," *CMC-Computers, Materials & Continua*, vol. 55, no. 2, pp. 213–227, 2018.
- [13] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo *et al.*, "Research commentary on recommendations with side information: A survey and research directions," *Electronic Commerce Research and Applications*, vol. 37, pp. 100879, 2019.
- [14] T. Y. Zhang and F. C. You, "Research on short text classification based on textcnn," *Journal of Physics: Conference Series*, vol. 1757, no. 1, Changsha, China, IOP Publishing, 2021.
- [15] I. Bayer, X. He, B. Kanagal and S. Rendle, "A generic coordinate descent framework for learning from implicit feedback," in *Proc. of the 26th Int. Conf. on World Wide Web*, Perth, Australia, pp. 1341–1350, 2017.
- [16] X. He, H. Zhang, M. Y. Kan and T. S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. of the 39th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Pisa, Italy, pp. 549–558, 2016.
- [17] J. F. Wang and P. F. Han, "Adversarial training-based mean bayesian personalized ranking for recommender system," *IEEE Access*, vol. 8, pp. 7958–7968, 2020.
- [18] S. J. Wang, L. B. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun *et al.*, "A survey on session-based recommender systems," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–38, 2021.
- [19] W. C. Kan and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE Int. Conf. on Data Mining (ICDM)*, Sentosa, Singapore, pp. 197–206, 2018.
- [20] Z. Hao, Y. Y. Xiao and Z. J. Bu, "Personalized book recommender system based on Chinese library classification," in *2017 14th Web Information Systems and Applications Conf. (WISA)*, Liuzhou, China, IEEE, pp. 127–131, 2017.
- [21] B. Pang, E. Nijkamp and Y. N. Wu, "Deep learning with tensorflow: A review," *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, pp. 227–248, 2020.

- [22] S. A. Sanchez, H. J. Romer and A. D. Morales, “A review: Comparison of performance metrics of pretrained models for object detection using the tensorflow framework,” *IOP Conference Series: Materials Science and Engineering*, vol. 844, no. 1, pp. 012024, IOP Publishing, 2020.
- [23] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries *et al.*, “Tensorflow lite micro: Embedded machine learning for tinyml systems,” *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, 2021.
- [24] H. Abdollahpouri, R. Burke and B. Mobasher, “Managing popularity bias in recommender systems with personalized re-ranking,” in *The Thirty-Second Int. Flairs Conf.*, Sarasota, USA, 2019.
- [25] C. Ajaegbu, “An optimized item-based collaborative filtering algorithm,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 12, pp. 10629–10636, 2021.
- [26] S. Rendle, W. Krichene, L. Zhang and J. Anderson, “Neural collaborative filtering vs. matrix factorization revisited,” in *Fourteenth ACM Conf. on Recommender Systems*, Virtual Event, Brazil, pp. 240–248, 2020.
- [27] A. Alsini, D. Q. Huynh and A. Datta, “Hit ratio: An evaluation metric for hashtag recommendation,” arXiv preprint arXiv:2010.01258, 2010.
- [28] R. Ye, X. Li, Y. Fang, H. Zang and M. Wang, “A vectorized relational graph convolutional network for multi-relational network alignment,” in *Int. Joint Conf. on Artificial Intelligence*, Macao, China, pp. 4135–4141, 2019.
- [29] L. Gienapp, B. Stein, M. Hagen and M. Potthast, “Estimating topic difficulty using normalized discounted cumulated gain,” in *Proc. of the 29th ACM Int. Conf. on Information & Knowledge Management*, Virtual Event, Ireland, pp. 2033–2036, 2020.
- [30] S. Lestari, T. B. Adji and A. E. Permasari, “NRF: Normalized rating frequency for collaborative filtering paper,” in *2018 Int. Conf. on Applied Information Technology and Innovation (ICAITI)*, Padang, Indonesia, pp. 19–25, 2018.