

Research Article

Network Course Recommendation System Based on Double-Layer Attention Mechanism

Qianyao Zhu 

Education Information Network Center, Guizhou Education University, Guiyang, Guizhou 550018, China

Correspondence should be addressed to Qianyao Zhu; zhuqianyao@gznc.edu.cn

Received 20 October 2021; Revised 3 November 2021; Accepted 9 November 2021; Published 16 December 2021

Academic Editor: Le Sun

Copyright © 2021 Qianyao Zhu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In view of the lack of accurate recommendation and selection of courses on the network teaching platform in the new form of higher education, a network course recommendation system based on the double-layer attention mechanism is proposed. First of all, the collected data are preprocessed, while the data of students and course information are normalized and classified. Then, the dual attention mechanism is introduced into the parallel neural network recommendation model so as to improve the model's ability to mine important features. TF-IDF (term frequency-inverse document frequency) based on the student score and course category is improved. The recommendation results are classified according to the weight of course categories, so as to construct different types of course groups and complete the recommendation. The experimental results show that the proposed algorithm can effectively improve the model recommendation accuracy compared with other algorithms.

1. Introduction

At present, online learning has become a common way of learning. Students do not have to be limited by the physical classroom; instead, they can break through the traditional time and space free choice of learning [1]. Online learning platforms emerge one after another, bringing great convenience to people. In the context of the new form of intelligent education [2, 3], the teaching platform that recommends online courses through the analysis of historical behavioral data has become the mainstream of the “Internet + education” model [4].

In the online learning environment, students need to spend tremendous time searching and browsing to confirm whether they are interested in the resources. Therefore, the problem of knowledge overload inevitably arises. An effective way to solve the problem of knowledge overload is to use a personalized recommendation algorithm [5]. After years of development, the performance of the recommendation system has greatly improved. Currently, recommendation algorithms are mainly divided into the collaborative filtering-based recommendation algorithm [6], content-based recommendation algorithm [7], and hybrid

recommendation algorithm [8]. The recommendation algorithm based on collaborative filtering finds similar users by acquiring their historical behavior and rating data and by capturing the likes and dislikes of similar users and recommending items that users have not seen before. Content-based recommendation algorithms are mainly based on the items or ratings that users have selected and also based on the user's historical behavior information to find similar projects to recommend. The hybrid recommendation algorithm will fuse different recommendation algorithms and then recommend to get better recommendation effect. As data grow, so do the types of data. Traditional recommendation algorithms cannot learn the deep features of users or projects. How to make full use of multisource heterogeneous data to improve the performance of recommendation system has become a hot topic of recommendation system research [9].

In recent years, deep learning has its own strong learning ability and has been widely applied in image recognition [10, 11], speech recognition [12, 13], natural language processing [14, 15], and other fields [16]. Deep learning is good at mining and learning the deep features of multisource heterogeneous data. By combining it with the

recommendation system, the hidden features of user and project attributes can be learned more efficiently [17]. Therefore, more and more researchers apply deep learning to the recommendation system. Although the application of neural network to the recommendation system can effectively improve the recommendation performance, not all feature interactions can contribute to the prediction results. For example, learning user or project features interacting with useless features may introduce noise and thus affect the performance of the recommendation system [18].

The teaching recommendation system algorithm is mainly based on content recommendation and collaborative filtering recommendation. For example, Yao et al. [19] introduced the application of three-dimensional biased weight tensor decomposition in teaching recommendation. Jie-Guang et al. [20] introduced a teaching optimization and research mode of rectangular neighborhood structure. Hou et al. [21] introduced WebGIS course teaching evaluation based on the dynamic adaptive teaching and learning optimization algorithm. Guan et al. [22] introduced the design and implementation of an open platform for dynamic mathematical and digital resources.

In view of the lack of accurate recommendation and selection of courses on online teaching platforms, the main work and innovations of this paper are as follows:

- (1) An online course recommendation system based on the double-layer attention mechanism is proposed. By introducing the double-layer attention mechanism into the parallel neural network recommendation model, the model's ability to mine important features is improved.
- (2) Reset the weight of the preprocessed course text information. The student feature vectors and course feature vectors learned by a multilayer fully connected neural network were input into the second-layer attention mechanism, and the multilayer perceptron was used to parameterize the attention score.

Finally, the curriculum attribute dataset is used to conduct curriculum recommendation experiments, and the accuracy of the proposed method in curriculum recommendation is compared and verified.

The structure of this paper is as follows. Section 2 focuses on the proposed algorithm model in this paper. Section 3 describes the experiment and analysis. Section 4 lists the conclusion.

2. The Proposed Model

A neural collaborative filtering (NCF) model uses a parallel neural network to learn user and project potential feature vectors, respectively [23]. In the prediction layer, the implicit vector is mapped to the predicted value by a multilayer perceptron (MLP). The NCF model uses the MLP to extract higher order feature information to improve the recommendation ability of the model. In the process of feature interaction, not all feature interactions can contribute to the prediction results, and the different effects of items in the historical interaction sequence on the current prediction are

ignored. Therefore, this paper introduces the attention mechanism into the neural network to assign personalized weight to the items of historical interaction sequence and improves the model.

On the basis of the NCF recommendation model, attribute information of students and online courses is taken as input data $u: \{u_1, u_2 \dots u_n\}$ and $v: \{v_1, v_2 \dots v_n\}$, respectively. The algorithm model architecture of this paper is shown in Figure 1.

A double-layer attention mechanism is introduced, and the first layer is used to combine with the convolutional neural network (CNN) to build a subnetwork so that the CNN can learn the key content in the online course text. The other layer takes student and online course feature vectors as input data and uses the attention mechanism to assign personalized weight to the students' history interactive online course. Then, the effect of different online courses on the current prediction preference can be obtained. The recommendation results are grouped and shown to students in the form of the online course group to enhance the order of the recommendation content.

2.1. Learning the Underlying Characteristics. To solve the problem of sparse data in the recommendation system, the attribute information of students and online courses is used to predict scores. After data preprocessing, the attribute information of students and online courses is input into the embedded layer to encode the attribute information. The embedding layer maps the input sparse vector to a dense low-dimensional embedding vector and obtains the embedding representations ρ_u and σ_v of student and online course attributes. At the beginning of the training, the embedding is simply selected randomly. As the training goes on, each embedded vector will be updated to help the neural network perform its task.

The embedding vectors ρ_u and σ_v of students and online courses are input into the parallel multilayer fully connected neural network to learn the potential feature vectors of nontext attributes of students and online courses, respectively:

$$\begin{aligned}\hat{\rho}_u &= f(w_{u2}f(w_{u1}\rho_u + d_{u1}) + d_{u2}), \\ \hat{\sigma}_v &= f(w_{v1}\rho_v + d_{v1}),\end{aligned}\quad (1)$$

where $f(\cdot)$ is the tanh activation function and w_v and d_u are the weight matrix and bias to be learned, respectively.

2.2. Convolutional Neural Network with Attentional Mechanism. The text information of the attributes of online courses includes the title of online courses. In order to enhance the network's ability to learn the key content in the text, the attention mechanism is combined with the CNN to form a subnetwork to extract text features. The text convolutional neural network is composed of an attention layer, a convolutional layer, a pooling layer, and a full connection layer, as shown in Figure 2.

The attention layer assigns attention weight to the word vector matrix of each online course text to obtain the

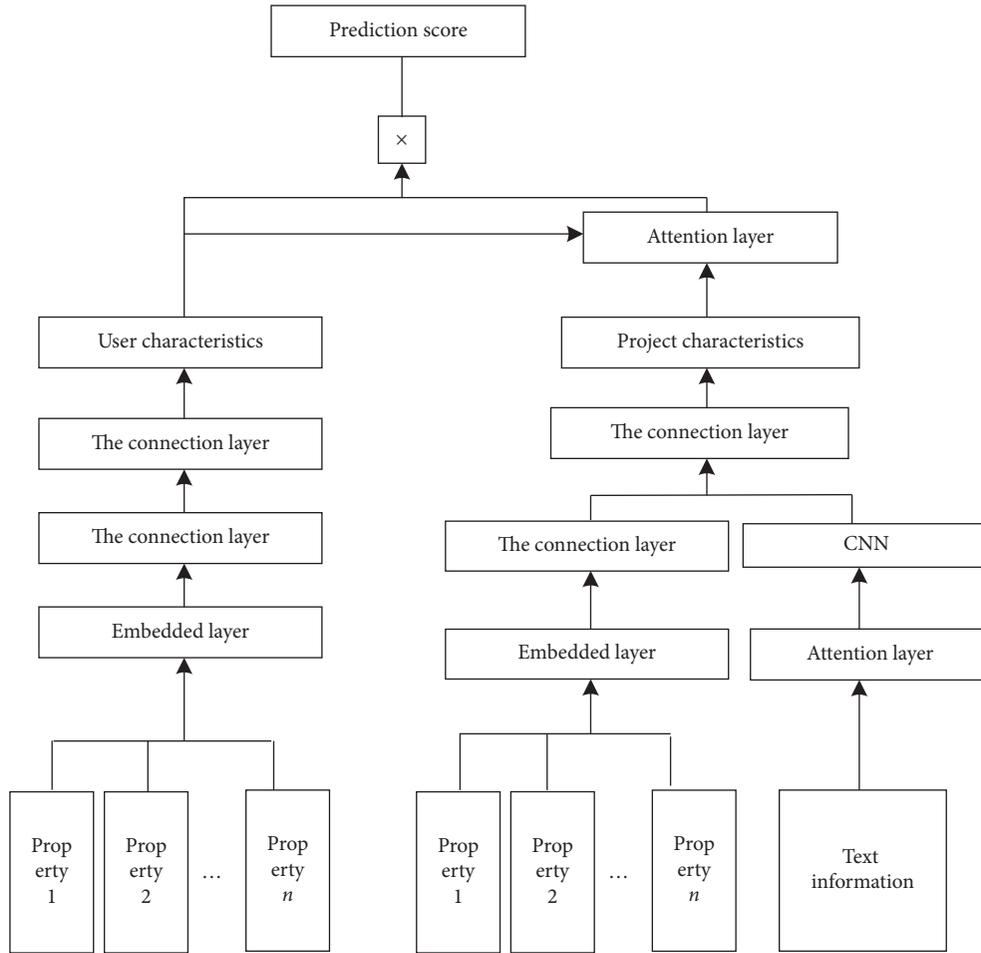


FIGURE 1: Architecture of the proposed algorithm in this paper.

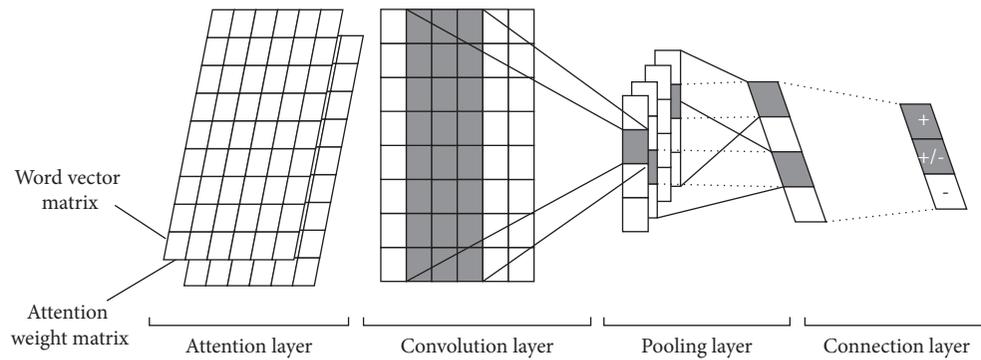


FIGURE 2: Text convolutional neural network.

updated word vector matrix. The word vector matrix $E \in \mathbb{R}^{n \times k}$ is obtained by embedding the text content of online course. K is the dimension of word vector, that is, every word is mapped to a K -dimensional vector $\alpha_x \in \mathbb{R}^k$. n is the number of words. $F \in \mathbb{R}^{n \times k}$ represents the word vector matrix of the text information carried by all the online courses browsed by the target student u_x . α_x is the word vector of the i th word indicating $\alpha_x \in \mathbb{R}^k$. The word vector representation α_x of each word in the word vector matrix F of target students and the attention score of each word α_x in

the word vector matrix E of all texts of online courses are calculated:

$$c(\alpha_x, \alpha_y) = v_c^T R(w_c [\alpha_x \oplus \alpha_y]), \quad (2)$$

where v_c^T and w_c are training parameters.

The attention score of $c(\alpha_x, \alpha_y)$ is normalized by Softmax function, and the attention weight a_{ij} corresponding to each word vector is obtained:

$$c_{xy} = s(c(\alpha_x, \alpha_y))$$

$$= \frac{\exp(c(\alpha_x, \alpha_y))}{\sum_{y=1}^n \exp(c(\alpha_x, \alpha_y))}, \quad (3)$$

where $c_{xy} \in C^{s \times k}$ is the attention weight value. The vector splicing operation of the attention weight matrix $C^{s \times k}$ and original word vector matrix F is carried out. The updated network course word vector matrix M_{att} will be obtained as the input matrix of the convolutional neural network:

$$M_{att} = C \odot F. \quad (4)$$

In the convolution layer, each neuron slides from the leftmost of the matrix F_{att} to the right along the direction of the sentence by the convolution kernel $F \in \mathbb{R}^{k \times m}$. The window size of convolution kernel F_y is set to m , and the feature representation of each word in the sentence is obtained after the convolution operation. Feature graphs are formed by activation functions. The y -th neuron produces features as shown in the following equation:

$$H_y = f(M^{s \times k} * F_y + d_y), \quad (5)$$

where $*$ is the convolution operation, d_y is the offset term, and f is the nonlinear activation function ReLU; the nonlinearity of the convolutional neural network is enhanced through the activation function f :

$$R(\alpha) = \max(0, \alpha). \quad (6)$$

Select maximum pooling to pool the output results of the convolution layer. The feature graph is divided into several rectangular regions, and the maximum value is output for each subregion. Maximum pooling removes unimportant or repetitive features in each subarea for the current task and retains information that can express text features. The pooling result of the y -th convolution kernel is shown in the following equation:

$$h'_y = \max\{h_1, h_2, h_3, \dots, h_y^{k-m+1}\}. \quad (7)$$

The pooled output is input to the full connection layer, multiplied by the weight matrix of the full connection layer, and added with a bias sum. Classification output is obtained after ReLU activation function; the hidden features of online course text information are as follows:

$$\sigma_{\text{text}} = R(w_y h'_y + d_y), \quad (8)$$

where w_y is the weight coefficient of the full connection layer and d_y is the offset term. Assume that the nontext attribute feature vector $\hat{\sigma}_v$ of online courses can be connected to the text feature vector σ_{text} to obtain the online course feature $\hat{\sigma}_y$:

$$\hat{\sigma}_y = h(\hat{\sigma}_v, \sigma_{\text{text}}). \quad (9)$$

2.3. Prediction Score. Traditional recommendation models typically perform the interaction between the implicit representation of $\hat{\rho}_u$ of student characteristics and $\hat{\sigma}_y$ of online

course characteristics to obtain the final predicted grade. Due to the lack of customized optimization of the recommendation task, the equal treatment of all the history online courses will limit the representation ability of the model. The traditional neural network recommendation model ignores that different online courses of student history play different roles in predicting the next online course, so the accuracy is low.

In the prediction layer of the algorithm model in this paper, a neural attention network is used to distinguish the importance of history network courses to overcome the limitations of the traditional neural network recommendation model. Learn the tacit representation of $\hat{\rho}_u$ of student characteristics and $\hat{\sigma}_y$ of online course characteristics as the input of the attention layer. The attention of target students to different online courses has different effects on predicting the next online course. The attention score of the student u_x to the online course v_y is shown in the following equation:

$$s_h(\hat{\rho}_u, \hat{\sigma}_y) = R_e(w_1(\hat{\rho}_u \odot \hat{\sigma}_y) + d_1), \quad (10)$$

where w_1 and d_1 are the weight matrix and bias term to be learned. The nonlinear relation of the current online course to predict the next online course is obtained by Re activation function. $s_h(\hat{\rho}_u, \hat{\sigma}_y)$ indicates that the student u_x pays more attention to the online course v_y . The online course v_y plays an important role in predicting the next online course. Softmax function was used to normalize the attention score \hat{c}_{xy} :

$$\hat{c}_{xy} = \frac{\exp(s_h(\hat{\rho}_u, \hat{\sigma}_y))}{\sum_{y \in R(u)} \exp(s_h(\hat{\rho}_u, \hat{\sigma}_y))}, \quad (11)$$

where \hat{c}_{xy} is the contribution degree of online course v_y to students' u_x preference curve and $R(u)$ indicates the historical interactive network course set of the student u_x . The weight of $\hat{\sigma}_y$ is implicitly reassigned to course characteristics as follows:

$$\hat{\sigma}_x = \sum_{y \in R(u)} \hat{c}_{xy} \hat{\sigma}_y. \quad (12)$$

As the inner product of student bearing $\hat{\rho}_u$ and online course bearing $\hat{\sigma}_x$ is calculated, the predicted score can be obtained as shown in the following equation:

$$\hat{\beta}_{ux} = \hat{\rho}_u^T \hat{\sigma}_x. \quad (13)$$

The mean square error (MSE) was used as a loss function to minimize the gap between the real score and the predicted score during the training of the model:

$$L_{\text{sqr}} \geq \sum_{(u,\lambda) \in R(u)} (\beta_{ux} - \hat{\beta}_{ux})^2, \quad (14)$$

where β_{ux} marks students' real online courses and $\hat{\beta}_{ux}$ estimates the scores students have for online courses. The objective function was optimized by using the stochastic gradient descent method to minimize the loss function, and the weight w_n and bias d_n of each layer were optimized by using the backpropagation algorithm. After the neural

network training is completed by the algorithms above, the model is used to predict students' grading of ungraded online courses. Recommendations are made to target students according to the predicted score size. The recommendation results are used for subsequent grouping to realize online course group recommendation.

2.4. Improved TF-IDF Method. TF-IDF is often used for text classification and information retrieval. TF-IDF generally considers only the number of documents and the frequency with which keywords appear in documents. When words have score data, they cannot make full use of score data to calculate TF-IDF values more accurately. The scoring data are introduced into the TF-IDF method to avoid losing words with high scores while evaluating the importance of words. The improved TF-IDF method is shown in the following equation:

$$S_{u_y, w_x} = \frac{\sum_{s=1}^n r_{u_y, v_s^x}}{\sum_{s=1}^n r_{u_y, v_s}} \text{lb} \frac{|K|}{|\{y: w_x \in k_y\}|}, \quad (15)$$

where S_{u_y, w_x} is the importance of w_x to student u_y , ranging from 0 to 1; $\sum_{s=1}^n r_{u_y, v_s^x}$ is the score of the file containing the words w_x ; $\sum_{s=1}^n r_{u_y, v_s}$ is the score sum of all files; $|K|$ is the total number of files in the database; and $|\{y: w_x \in k_y\}|$ is the number of all files. According to the dataset of the online course, scoring data are determined and obtained as r_{u_y, v_s} . S_{u_y, w_x} is calculated based on the score data of the history online courses of the designated students in the training set. The improved TF-IDF is used to calculate the TF-IDF value of the types of online courses contained in the recommendation results, and the group recommendation of online courses is realized by obtaining students' preference for different types.

The first part of equation (15) is term frequency calculation of TF-IDF. The predicted score was used to calculate the score of online courses containing file WI and its proportion in the total file score sum. When the number is large, it indicates that the score of the file containing w_x is higher, which reflects the importance of w_x to some extent. The second part is the inverse document frequency of TF-IDF. The more the number of files contain words w_x in the total number of database files, the more weak the importance of TF will be w_x . S_{u_y, w_x} is obtained by multiplying the two parts, and the w_x value indicates students' preference for the word.

Based on the improvement of the TF-IDF algorithm, this paper analyzes the types of online courses in the recommendation results, obtains the importance of different types of online courses to students, and realizes the automatic grouping of recommendation results. The same online courses from the top- N recommendation results were placed into the same group. Students' favorite type of online courses will be recommended first, so that students can quickly find content that matches their interests. The specific implementation steps of the improved TF-IDF algorithm are as follows:

Step 1: calculate the word frequency in the specific file k_y , including the word w_x score of the online course and the proportion of the score and sum of all online courses in the students' browsing history:

$$\text{TF}_{y,x} = \frac{\sum_{s=1}^n r_{u_y, v_s^x}}{\sum_{s=1}^n r_{u_y, v_s}}. \quad (16)$$

Step 2: calculate the reverse file frequency index. Divide the number of centralized online courses by the number of online courses containing the word w_x and then take the logarithm of the quotient:

$$\text{IDF}_x = \text{lb} \frac{|K|}{|\{y: t_x \in k_y\}|}. \quad (17)$$

Step 3: multiply word frequency and reverse file word frequency to obtain the TF-IDF value of w_x in the file S_{u_y, w_x} :

$$S_{u_y, w_x} = \text{TF}_{y,x} \times \text{IDT}_x. \quad (18)$$

The improved TF-IDF uses score data to reflect the proportion of word w_x in the file k_y when calculating word frequency.

2.5. Grouping the Recommended Results. The improved TF-IDF method was used to obtain students' preference for different types of online courses. Taking online course recommendation as an example, the CourseLens dataset is used to test the algorithm. The word frequency information of online course types is shown in Table 1. First of all, N online courses are recommended to students. Second, the online course information document $M = \{m_1, m_2, \dots, m_N\}$. Extract the network course types contained in the network course information documents and establish the network course type information documents $G = \{g_1, g_2, \dots, g_N\}$. Finally, the word frequency statistics of online course type documents are made to obtain the word frequency information of online course type in the recommended results.

When the recommended online course m_s contains the online course type, the value of R_{m_s, g_x} is 1. Otherwise, the value is 0. The improved TF-IDF is used to analyze the word frequency information of online course types and predict students' preference for different online course types, as shown in the following equation:

$$S_{u_y, g_x} = \frac{\sum_{s=1}^n r_{u_y, m_s} \times R_{m_s, g_x}}{\sum_{s=1}^n r_{u_y, m_s}} \text{lb} \frac{N}{\sum_{s=1}^n R_{m_s, g_x}}, \quad (19)$$

where r_{u_y, m_s} is the scores of the student u_y on the online course m_s , R_{m_s, g_x} indicates whether the online course m_s contains type g_x , N is the number of online courses with recommended results, and S_{u_y, g_x} is students' preference for different types of online courses.

The types of online courses are arranged in descending order according to their preference degree S_{u_y, g_x} , and the first

TABLE 1: Word frequency information of course types.

Course	Type			
	g_1	g_2	\dots	g_x
m_1	Rm_1, g_1	Rm_1, g_2	\dots	Rm_1, g_x
m_2	Rm_2, g_1	Rm_2, g_2	\dots	Rm_2, g_x
\dots	\dots	\dots	\dots	\dots
m_s	Rm_s, g_1	Rm_s, g_2	\dots	Rm_s, g_x

K types are taken as the group name of the online course group to be recommended, namely, $L = \{L_1, L_2, \dots, L_K\}$, where L represents the set of online course groups to be recommended. Add D online courses of the same type to each online course group. The online courses in L_K are derived from the recommendation results generated by the algorithm model in this paper. The online courses in the online course group are arranged in descending order according to the predicted score. Finally, K online course groups are formed, and each online course group contains online course group recommendation of the same type of online course in part D .

3. Experiment

3.1. Experimental Environment and Dataset. The experiment was carried out in PyCharm integrated development environment on a 64 bit Windows 10 system. Versions of Python and TensorFlow deep learning frameworks are 3.7.9 and 1.8.0. The CPU of the computer is 3.50 GHz Intel Core i7-7800, and the memory is 32 GB.

The experiment used data from an online learning platform. Through collection and collation, the dataset of 500 students, 700 online courses, and 35,000 scoring records was finally formed. As the collected data have both structured and unstructured data, it is necessary to preprocess the data. First of all, the data are classified, and different processing methods are selected for different classification using group student ID, online course ID, the number of studies, and grade value. As the types of students, grades, and categories of online courses are classified information, they are classified as one class. Then, the values of each field are digitized. Since the name and description information of online courses are both text information and cannot be quantified, they are classified as one class.

There are three stages to realize the recommendation: the first stage is data preprocessing, the second stage is the construction and training of deep neural networks, and the third stage is the generation of the recommendation list.

- (1) Data preprocessing. Since the data used are not all of digital form, such as the type of students and the category of online courses, these data cannot be directly input into the network; instead, they need to be transformed into vector representation.
- (2) Building and training the network model. First, a deep neural network is constructed. On this basis, the data processed in the previous step are divided into training sets and test sets. In this experiment,

80% of the datasets are used as training sets and 20% of the datasets are used as test sets.

- (3) Generating a recommendation list. The trained model was used to predict scores. Finally, the top- N online courses are recommended for students according to the rank of scoring value.

3.2. Evaluation Indicators. There are many evaluation indexes based on the top- N recommendation algorithm. The HR (hit ratio), NDCG (normalized discount cumulative gain), RMSE (root mean square error), and MAE (mean absolute error) will be used in this experiment.

The calculation equation of HR is as follows:

$$HR(N) = \frac{|\rho \in \text{testset} \& \rho \in R_N|}{|\text{testset}|}, \quad (20)$$

where P represents the online courses in the test set, $\rho \in \text{testset} \& \rho \in R_N$ indicates that online courses in the test set are also in the top- N recommended set, and testset indicates the number of test sets.

The NDCG is widely used in the evaluation tasks of recommendation ranking and information index ranking. It is a kind of evaluation index sensitive to ranking position. The more relevant the ranking is, the bigger the corresponding NDCG will be. First, the DCG is introduced, and its calculation equation is as follows:

$$DCG_n = \sum_{x=1}^n \frac{2^{\text{rel}_x - 1}}{\log_2(x + 1)}, \quad (21)$$

where rel_x is the relevance of the first online course to students. In this experiment, if the online course is a positive sample, the correlation degree is 1, while the negative sample has a correlation degree of 0. If more relevant online courses are ranked later, the overall DCG will be smaller.

Each student's recommendation is ranked by an ideal value, which is ranked from most relevant to least relevant. This is the minimum IDCG that each student can calculate. Thus, the NDCG can be calculated by the following equation:

$$NDCG_n = \frac{DCG_n}{IDCG_n}. \quad (22)$$

The NDCG has a value between 0 and 1, and the larger the value, the better the recommendation effect.

The RMSE and MAE are calculated as follows:

$$\begin{aligned} RMSE &= \sqrt{\frac{\sum_{u,x \in T} (r_{ux} - \hat{r}_{ux})^2}{|T|}}, \\ MAE &= \frac{\sum_{u,x \in T} |r_{ux} - \hat{r}_{ux}|}{|T|}, \end{aligned} \quad (23)$$

where T is the number of online courses with scoring records in the test set.

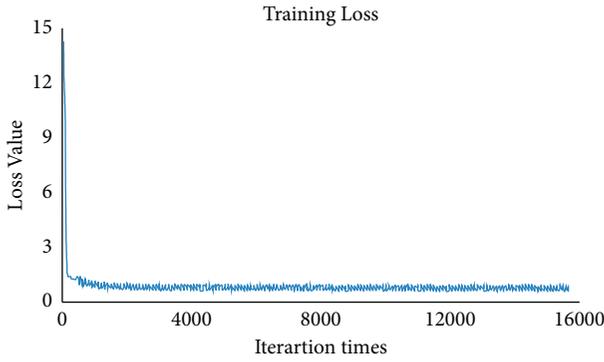


FIGURE 3: Loss value of the proposed method.

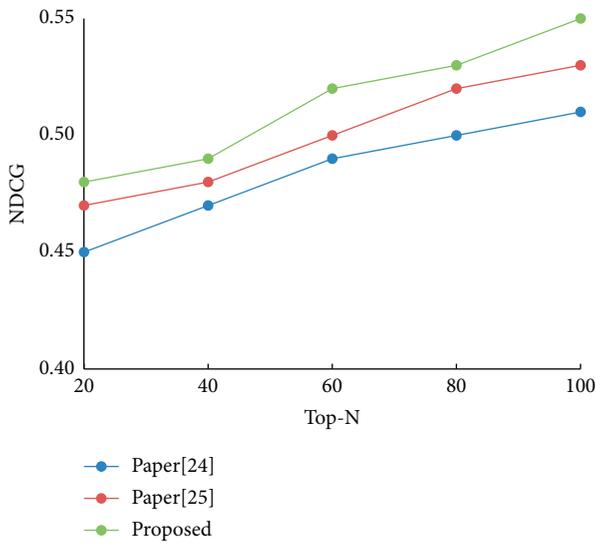


FIGURE 4: Comparison results of the NDCG index.

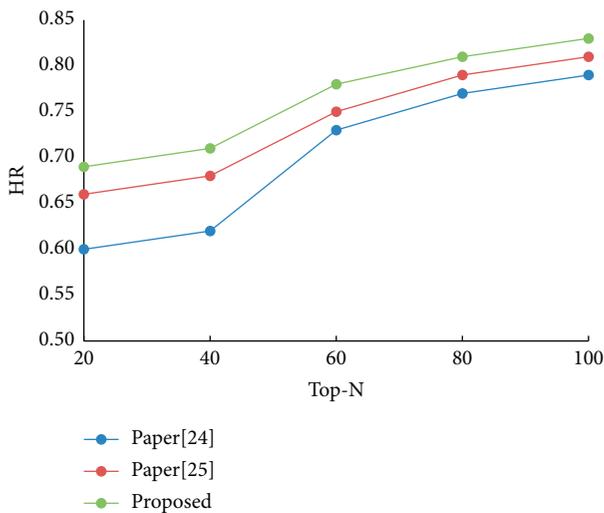


FIGURE 5: Comparison results of the HR index.

3.3. *Experimental Results and Analysis.* In order to verify the effectiveness of the algorithm in this paper, MAE function was used for training in the experiment. It can be seen from

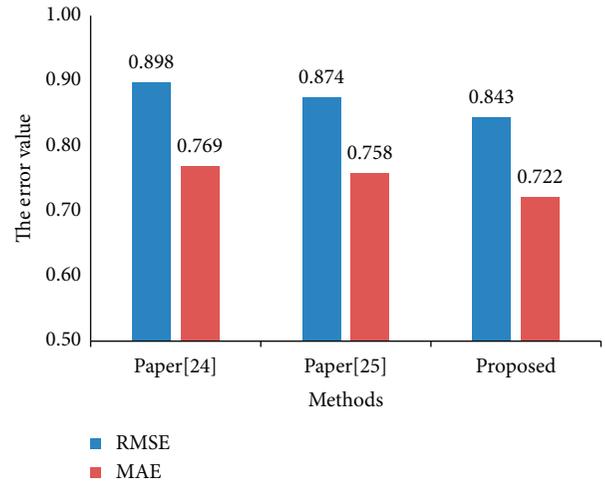


FIGURE 6: Comparison results of RMSE and MAE indexes.

the Figure 3 that, with the increase of iteration times, the MAE value gradually decreases to 0.69. The algorithm model in this paper, which combines student attributes, online course attributes, and content description information, has more complete data and tends to reduce errors. Therefore, the model in this paper has certain validity (Figure 3).

Next, the algorithm in this paper is analyzed and compared with other algorithms [24, 25]. In order to test the influence of top- N number on the recommendation effect, the values of N were set as 20, 40, 60, 80, and 100, respectively. The NDCG and HR were analyzed, respectively. The experimental results are shown in Figures 4 and 5. It can be seen that, with the increase of top- N number, NDCG and HR indicators of these three algorithm models are constantly improved. The performance indexes of the proposed algorithm are all higher than those of other algorithm models, indicating that the proposed algorithm model has a better recommendation effect.

The comparison of the RMSE and MAE between online course group recommendation and other recommendation algorithms is shown in Figure 6. The RMSE and MAE values of the proposed algorithm are higher than those of other algorithms. This further indicates that the proposed algorithm can improve the accuracy of the recommendation algorithm by extracting implied features of students and online courses after referring to the attention mechanism.

4. Conclusion

This paper proposes a kind of online course recommendation system based on the double-layer attention mechanism, aiming to solve the problem of lack of precise guided course selection in the existing network platform. The feature extraction ability of the convolutional neural network is improved by introducing the two-level attention mechanism into the convolutional neural network. Different preference weights are assigned to curriculum features to achieve a recommendation method more in line with students' preferences. In course recommendation to target students, score data and network course type data are

combined to complete the grouping of recommendation results. The experimental results show that the system achieves better performance in the NDCG, HR, RMSE, and MAE, which provides necessary theoretical support for the accurate selection of course guidance on the network platform. The analysis and optimization of algorithm efficiency is the direction of further research.

Data Availability

The labeled datasets used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This work was supported by the study on the Practice of Improving Information Technology Application Ability of College Teachers in the Context of Education Informatization 2.0 (no. 20190202047069) and in part by the study on the Measurement Model of Teaching Proximity in Online Courses in Higher Education (no. 19GZQN26).

References

- [1] L. Dunlop, L. Clarke, and V. McKelvey-Martin, "Free-choice learning in school science: a model for collaboration between formal and informal science educators," *International Journal of Science Education, Part B*, vol. 9, no. 1, pp. 13–28, 2019.
- [2] S. Ali, B. H. Payne, R. Williams, and C. Breazeal, "Constructionism, ethics, and creativity: developing primary and middle school artificial intelligence education," in *Proceedings of the International Workshop on Education in Artificial Intelligence K-12 (EDUAI'19)*, pp. 1–4, Austria, Hungary, 2019.
- [3] K. Kim, "An artificial intelligence education program development and application for elementary teachers," *Journal of the Korean Association of Information Education*, vol. 23, no. 6, pp. 629–637, 2019.
- [4] K. Ren and Y. Bai, "Research on innovative methods of college students' sports teaching based on internet education technology," *Educational Sciences: Theory and Practice*, vol. 18, no. 5, p. 2486, 2018.
- [5] X. Xiaoli, H. Hui, W. Mengmeng, L. Yu, Y. Ziheng, and W. Yingfeng, "Development of personalized learning resources recommendation system based on knowledge graph," *International Journal of Educational Technology and Learning*, vol. 10, no. 2, pp. 68–72, 2021.
- [6] L. Wu, X. He, X. Wang, and K. Zhang, "A survey on neural recommendation: from collaborative filtering to content and context enriched recommendation," 2021, <https://arxiv.org/abs/2104.13030>.
- [7] J. Shu, X. Shen, H. Liu, B. Yi, and Z. Zhang, "A content-based recommendation algorithm for learning resources," *Multi-media Systems*, vol. 24, no. 2, pp. 163–173, 2018.
- [8] Y. Tian, B. Zheng, Y. Wang, Y. Zhang, and Q. Wu, "College library personalized recommendation system based on hybrid recommendation algorithm," *Procedia CIRP*, vol. 83, pp. 490–494, 2019.
- [9] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 278–288, Florence, Italy, May 2015.
- [10] J. Huixian, "The analysis of plants image recognition based on deep learning and artificial neural network," *IEEE Access*, vol. 8, Article ID 68828, 2020.
- [11] T. Shen, C. Gao, and D. Xu, "The analysis of intelligent real-time image recognition technology based on mobile edge computing and deep learning," *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 1157–1166, 2021.
- [12] A. Santhanavijayan, D. Naresh Kumar, and G. Deepak, "A semantic-aware strategy for automatic speech recognition incorporating deep learning models," in *Advances in Intelligent Systems and Computing*, pp. 247–254, Springer, Berlin, Germany, 2021.
- [13] H. Veisi and A. Haji Mani, "Persian speech recognition using deep learning," *International Journal of Speech Technology*, vol. 23, no. 4, pp. 893–905, 2020.
- [14] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox, "Natural language processing advancements by deep learning: a survey," 2020, <https://arxiv.org/abs/2003.01200>.
- [15] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2020.
- [16] Q. Yuan, H. Shen, T. Li et al., "Deep learning in environmental remote sensing: achievements and challenges," *Remote Sensing of Environment*, vol. 241, Article ID 111716, 2020.
- [17] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning--based text classification," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–40, 2021.
- [18] H. J. Xue, X. Dai, and J. Zhang, "Deep matrix factorization models for recommender systems," in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI 2017)*, vol. 17, pp. 3203–3209, Melbourne, Australia, August 2017.
- [19] D. H. Yao, L. I. Shi-Jun, and H. U. Ya-Hui, "A three-dimensional partial weight tensor model for teaching recommendation," *Journal of University of Electronic Science and Technology of China*, vol. 46, no. (5), pp. 747–754, 2017, <https://doi.org/10.3969/j.issn.1001-0548.2017.05.018>.
- [20] H. E. Jie-Guang, Z. P. Peng, W. H. Lin, and D. L. Cui, "A teaching-learning-based optimization algorithm with rectangle neighborhood structure," *Acta Electronica Sinica*, vol. 47, no. 8, pp. 1768–1775, 2019.
- [21] J.-w. Hou, K.-l. Jia, and X.-j. Jiao, "Teaching evaluation on a WebGIS course based on dynamic self-adaptive teaching-learning-based optimization," *Journal of Central South University*, vol. 26, no. 3, pp. 640–653, 2019.
- [22] H. Guan, X. Qin, and Y. Rao, "Research and design of dynamic mathematical digital resources open platform," *Journal of Harbin Institute of Technology*, vol. 51, no. 5, 22 pages, 2019.

- [23] X. He, L. Liao, H. Zhang, and L. Nie, “Neural collaborative filtering,” in *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, Perth, Australia, April 2017.
- [24] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, “NAIS: neural attentive item similarity model for recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2354–2366, 2018.
- [25] Y. Zhu, H. Lu, P. Qiu, K. Shi, J. Chambua, and Z. Niu, “Heterogeneous teaching evaluation network based offline course recommendation with graph learning and tensor factorization,” *Neurocomputing*, vol. 415, pp. 84–95, 2020.