Research Article

Design and Implementation of a Machine Learning-Based English Intelligent Test System

Jing Guo

Wuchang Shouyi University, Wuhan, 430064 Hubei, China

Correspondence should be addressed to Jing Guo; 2004111015@wsyu.edu.cn

Received 25 March 2022; Revised 6 April 2022; Accepted 12 April 2022; Published 24 May 2022

Examination is a particularly important part of the students’ whole learning process. Therefore, this paper designs and develops an online examination system. This paper studies an examination recommendation algorithm based on a machine learning neural network model. The algorithm can recommend problems for students according to students’ error history, error prone, and knowledge poor problems. Experiments also show that the recommendation algorithm studied in this paper improves the recommendation effect compared with several other similar algorithms.

1. Introduction

The development of the Internet and the development of Internet plus strategy have led to the reform in the educational field, breaking through the traditional education mode in time and space and reducing the consumption of manpower and material resources. For example, examination is a particularly important part of students’ learning process, but the traditional paper-based examination involves the process of manually writing questions, printing test papers, students’ examination, manual scoring, and result statistics. The whole process may be very labor-intensive and cost-effective [2]. At present, network education is booming, and there are a large number of online examination systems. The emergence of a large number of online examination systems has well solved the disadvantages of traditional paper examination [3]. However, in the era of big data, the online education industry is also facing information overload, and a large number of resources are not fully utilized [4]. The problem is that information is overloaded and a large number of resources are not fully utilized. Some online platforms on the market and many universities provide practice and examination functions but provide the same access rights for all students. However, they provide the same problems to all students, ignoring their personal learning needs [5]. The existing examination system in the market is mainly profit oriented, paying more attention to the aesthetic design of the interface and the expansion of functions. In contrast, many university examination systems are designed to evaluate students’ recent learning, so they pay more attention to the core online examination function [6]. On the other hand, many university examination systems are designed to evaluate students’ recent learning, so they focus more on the relatively simple core online examination function. For these systems, with the accumulation of question bank, students can practice at random. The system is not designed to measure students’ recent learning, so it focuses more on the core online examination function, which is relatively simple [7]. Students are unable to find their own problems in the huge problem bank, and even a large number of problems have not been selected, which leads to the students’ half-hearted efforts [8].

The development of information technology and the popularity of the Internet have brought about a massive amount of information, and we have entered the era of big data [9]. While the popularity and growth of information has brought convenience to users, it has also brought the problem of information overload [10]. Faced with such a huge amount of data, both users and information service providers will encounter great challenges. It is difficult for
users to retrieve information that is useful or interesting to them [11]; it is also not easy for information service providers to make their information stand out [12].

Recommendation algorithms are the core component of recommendation systems. Traditional recommendation algorithms include content-based recommendation algorithms, collaborative filtering algorithms [13], and hybrid recommendations, among which collaborative filtering is the most widely used. With the development of deep learning and the increase of dataset size, traditional recommendation algorithms can no longer well solve the problems of data sparsity, cold start, and other problems, in addition to the accuracy of the learned model cannot meet the requirements. Therefore, the current academic and industrial research recommendation system algorithms are mostly based on deep learning [14].

Nowadays, personalized education is an important research topic. Based on consideration of the above issues, this paper’s recommendation algorithm is applied to the developed online examination system. The main focus is on the test practice module of the online examination system. The paper also focuses on the test practice module of the online examination system and researches a personalized recommendation algorithm that can recommend different questions for different students according to their history of wrong questions. The system can recommend different quality questions for different students based on their history of wrong questions, making the practice less blind and more focused, thus improving learning efficiency [15]. This helps both students and teachers to a certain extent. This will help both students and teachers to a certain extent, and will also contribute to the development of personalized education.

2. Related Work

Since 2006, when deep learning was proposed by Hinton et al., it has gradually been widely noticed and applied by the scientific community, and it has made outstanding achievements in image processing, natural language processing, etc. Especially in recent years, deep learning has been researched and applied quite extensively [12]. An earlier application of deep learning to recommendation algorithms used restricted Boltzmann bodies to build models, but the scale of connections between neural network layers was too large. Most of the current research in recommendation algorithms is built on deep learning. A recommendation algorithm based on multilayer neural networks was proposed by [13] and applied to YouTube video recommendation. [14] proposed a memory-based collaborative filtering algorithm for neural networks that exploits the interaction relationship to user items and combines neighboring users in a nonlinear way through the memory layer to make the representation of users more accurate with some improvement in effectiveness. However, only first-order connectivity is considered in the combination of near-neighboring users, without tapping into deeper user feature information. In order to solve the sparsity problem faced by traditional collaborative filtering, a hierarchical Bayesian model based on deep learning is proposed, adding certain auxiliary information to represent the content information in a neural network, taking into account complex nonlinear relationships and enhancing the vector representation performance. However, the form of inner product still used in the prediction layer is not sufficient to reveal the complex nonlinear relationships between user and item interactions [16]. In response to Wang’s research, [15] proposed an NCF model using a nonlinear neural network structure instead of inner product operations to enhance the interaction to capture the nonlinear feature interactions between users and items. [17] is an industrial application that employs multiple convolutional layers on the item graph for Pinterest image recommendation and thus captures relationships at the level of items rather than collective user behavior. [18] proposes a spectral convolution operation to discover all possible connectivities between users and items in the spectral domain, which allows the discovery of ri connections between user-item pairs through the feature decomposition of the graph adjacency matrix, but is complex, very time-consuming, and does not support large-scale recommendation schemes [19, 20].

3. Neural Graph Collaborative Filtering Algorithm

The neural graph collaborative filtering (NGCF) algorithm is an algorithm proposed in a paper published in 2019. Based on implicit feedback data, the core idea of the algorithm is to apply deep learning techniques to recommender system algorithms and exploit the higher-order connectivity of the user-item interaction graph to extract deeper features of users and items, which are used to enhance the embedding feature representation of users and items. This improves the performance of the algorithm by providing a more accurate representation of users and items than traditional collaborative filtering algorithms.

3.1. The Concept of Higher-Order Connectivity. In order to extract synergistic signals, this can be achieved using a higher-order connectivity graph structure for user-item interaction. The higher-order connectivity diagram is shown in Figure 1.
Figure 1 shows the $u_1$ expanded tree structure. Higher-order connectivity refers to paths that reach $u_1$ from any node and are greater than or equal to 1. The connectivity of this structure contains rich synergistic signals. For example, path $u_1 \leftarrow i_2 \leftarrow u_2$ contains behavioral similarities between users $u_1$ and $u_2$, $i_2$ as both interact with each other, while path $u_1 \leftarrow i_2 \leftarrow i_4 \leftarrow u_2$ suggests that $u_1$ has the potential to interact with $i_4$.

It can be seen that the higher-order connectivity structure contains a lot of valuable information. Therefore, it is of great importance to utilize it to integrate into the model representation.

3.2. Neural Graph Collaborative Filtering Model. The neurograph collaborative filtering model designs a neural network approach to propagation for recursion, thus enabling the integration of higher-order connectivity information into the embedding. Specifically, this is achieved by designing an embedding propagation layer that is primarily responsible for fusing interaction information into the user and item embedding, and then by adding multiple such propagation layers, the higher-order connected collaborative signals can be captured. In Figure 1, the behavioral similarities between layers can be captured by adding two additional layers. By adding three layers, it is possible to capture the potential interest in of interest.

The neural graph collaborative filtering model has three main components: an embedding layer, multiple embedding propagation layers, and a prediction layer. The overall structure of the model is shown in Figure 2.

(1) Embedding layer: the main purpose is to map the sparse feature vectors of users and items into a dense type and generate the initial user and item embedding. A matrix can be constructed as an embedding lookup table, as in

$$E = [e_{u_1}, \cdots, e_{u_N}, e_{i_1}, \cdots, e_{i_M}],$$

where $e_u \in R^d (e_i \in R^d)$ denotes the initial embedding of users and items and $d$ denotes the dimensionality. In the above set, the first $N$ items represent users and the last $M$ items represent items.

(2) Embedding propagation layer: each propagation layer consists of two phases: message construction and message synthesis.
(1) Message construction: for a user $u$, and an item $i$ with which an interaction has occurred, the composition is in terms of a user-item pair $(u, i)$, and the message from $i$ to $u$ is defined as

$$m_{u|i} = \frac{1}{\sqrt{N_u N_i}} (W_1 e_i + W_2 (e_i e_u)), \quad (2)$$

where $1/\sqrt{N_u N_i}$ is the Laplace parametrization, which represents the decay factor of the propagation process; $N_u$ and $N_i$ represent the number of items that user $u$ has interacted with and the number of users that have interacted with item $i$, respectively; $W_1$ and $W_2$ represent the two weight matrices, which are the parameters that the model needs to learn; $e_i$ and $e_u$ represent the initial imbedding; and $\cdot$ represents the element-by-element product of vector operator.

(2) Message synthesis: in order to obtain the embedding of user $u$ optimized after propagation, it is necessary to synthesize the messages from all items to $u$ that have interacted with user $u$. The synthesis function is defined in

$$e_u^{(1)} = \text{LeakyRelu} \left( m_i e_i + \sum_{i \in N_u \setminus u} W_2 (e_i e_u) \right), \quad (3)$$

where $e_u^{(1)}$ represents the embedding of user $u$ after one layer of propagation, LeakyRelu is the activation function, and $m_i e_i$ indicates that the self-connected signal retains the features $m_i = W_1 e$ before propagation.

Similar to obtaining the embedding of user $u$ above, the embedding representation of item $i$ can also be obtained after a layer of propagation $e_i^{(1)}$, also in the above two stages.

4. Intelligent Paper Test System

This system largely fulfills the design requirements described above (with the exception of support for automated marking machines, which is not included at this time to reduce costs). The subsystems in Figure 3 are briefly described below:

(1) Parameter input module

It can accept various control conditions input by the user and put various control conditions in the automatic grouping of papers for reasonable and flexible matching, which is an important guarantee of the system’s scientificity and flexibility. It is worth mentioning that this module can dynamically reflect the important information of the question bank in the system, such as various assessment points, the amount of questions of various assessment points, and generate question papers accordingly.

(2) Automated paper assembling module

The core part of the system, its efficiency is closely related to the structure of the question bank. A well-designed automatic paper-forming module should be able to interface well with the parameter input module and complete the function of randomly generating examination papers under strict control.

The output module of the system can output test papers directly to a printer or to Word for typesetting and printing. It is particularly worth mentioning that the organic
combination of the system and MS Word, borrowing Word’s powerful typesetting functions and spell checking and grammar checking tools, makes it possible for the user to have a complete and beautiful examination paper in front of him.

(3) System output module

The system output module can output test papers directly to the printer or to Word for typesetting and printing. It is particularly worth mentioning that the organic combination of the system and MS Word, using Word’s powerful typesetting functions and spell checking and grammar checking tools, makes it possible to present the user with a complete and beautiful examination paper [21, 22].

(4) System maintenance module

The maintenance module of the system can be divided into two submodules, one for the maintenance of test papers and the other for the maintenance of question banks.

The main purpose of the maintenance module is to modify test errors and save them back to the original question bank, which is in line with user habits, as users are generally reluctant to maintain the question bank directly, and there is some assurance of the security of the question bank.

The question bank maintenance module can add, delete, and change questions, and it has the ability to add new questions and new question types, so it can adapt to the changing standards of the National Level 4 and 6 examinations. Practice has proven that a dynamic and scalable system is a viable system.

(5) Automatic test module

This is where the user can conduct an on-machine test of the objective part of the set of test papers (including listening tests). The system here fully simulates the real examination room, e.g., listening can only be listened to once, other test questions can be referred to repeatedly; there is a clock to show the time, etc.

(6) Storage and loading module

The main purpose of this module is to allow users to use the system for on-board testing and paper maintenance. The web-based version can remotely load teacher-generated test papers for examinations [23].

(7) Information statistics module

Here, the user can access a variety of dynamic information about the test papers: the types of questions they contain, the number of questions in each type, the distribution of assessment priorities, the difficulty factor, etc. The user can even perform statistics such as pass rates and merit rates, in the simple result management system provided as an add-on to this module.

The flowchart for randomly selected questions is essentially a random selection of questions under a variety of conditions, see Figure 4.

The focus is on random selection, and the key is on condition control. Two types of question banks have been designed: special and general. The dedicated question bank contains a few typical question types already available in the system, such as reading comprehension. One of the difficulties of the system is that the parameter input module should allow the user to control the criteria for new question types.

About the interface between the system and Word in the output module, the system queries the registration information on the user’s machine to detect whether Word is installed and where it is located and then uses OLE Automation to communicate with it [24, 25].

Table 1: Format of data collected.

| Dataset       | Data collection format          |
|---------------|---------------------------------|
| Student error data | Student ID, title JD, whether it is wrong |
| Gowalla       | User ID, location ID, check in  |
| Amazon-book   | User ID, book ID, purchase or not|

Table 2: Statistical information on the dataset.

| Dataset    | Users | Items | Interactions | Density   |
|------------|-------|-------|--------------|-----------|
| Gowalla    | 27458 | 40685 | 1027424      | 0.000074  |
| Amazon-book| 54231 | 95478 | 2948712      | 0.00059   |

(1) Gowalla dataset: this dataset is a check-in dataset from the Gowalla website, which records the check-in time and location information of users on the social networking site, including user ID, check-in time, latitude, longitude, and location ID. Users can share their location through check-in.
In order to take advantage of the system’s on-board testing and paper maintenance capabilities, the system provides a dedicated storage and loading module in addition to text files to hold the papers. This storage can of course be done by storing a single paper in a single file, but this would result in a flood of files and inconvenience to manage, so the system uses a single file to store multiple papers.

5. Experiments and Data

5.1. Dataset Selection. In order to complete the test recommendation function, the first step is to obtain the data. The data can be obtained from the exam system database wrong question record table; you need to collect the student id, question id, whether the question is wrong (0 or 1), and all students’ wrong question record data. In this paper, the recommendation data interface is reserved for the recommendation exercise module of the examination system, which can obtain data from the database and call the recommendation algorithm. However, as the system prototype implemented in this paper is still in the testing stage, the data of real students’ question records are being collected and a large amount of data cannot be obtained. In order to ensure the accuracy and effectiveness of the recommendation algorithm, two publicly available datasets, Amazon-book and Gowalla, which can be mapped to the database structure of the prototype, were selected for experimentation to verify the effectiveness of the recommendation algorithm.

The descriptions of the dataset selection are as follows:

1. These two datasets are publicly available and have been widely used in the study of recommendation algorithms
2. In fact, the recommendation algorithm in this paper focuses on the implementation of the recommendation function and the accuracy of the algorithm’s recommendations and is not specific to a particular scenario, as shown in Table 1

The table above shows the format of the data recorded for real student errors in the system and the format of the records for the two datasets. For each dataset, in order to perform the algorithm experiments, data processing needs to be performed first.

The two datasets differ in size, sparsity, etc. and the statistics for both are shown in Table 2.

5.2. Comparative Analysis of Results. The model is improved based on NGCF as shown in Figure 5 for the examination evaluation of different students. The user and item embedding is first initialized and an attention mechanism is introduced in the propagation layer. Then, a combination of variable weighted averaging is applied to the interaction information, and after multiple layers of propagation, the embedding is obtained after optimization of each layer. Finally, these embeddings, together with the initial embedding, are connected through a cascade layer.
In addition, to validate the effectiveness of the algorithmic models, the model NGCF-Att from this paper was compared with the three algorithms BPRMF, NCF, and NGCF. For a fair comparison, all algorithms were optimized using the BPR loss function.

The recall@20 and ndcg@20 evaluation metrics for the four algorithms on the Gowalla and Amazon-book datasets are shown in Table 3.

The comparison of the results in the above table shows that the other three models based on deep learning are slightly better than the traditional matrix decomposition MF model, which reflects the optimization of the traditional MF to varying degrees.

Secondly, the index of NGCF-Att will be slightly higher than that of NCF and NGCF, because it combines the advantages of the two models and introduces an attention mechanism to improve network efficiency.

Therefore, we can draw the following conclusions:

(1) The research of recommendation algorithm based on deep learning is an important way to improve the performance of recommendation system

(2) Inner product has some limitations. Using neural network instead of inner product can capture the complex nonlinear interaction between users and items

(3) The attention mechanism assigns different weights to different items of each user, which is more in line with the realistic level of users’ preference for different items. Variable weight learning improves the performance of the model and makes it easier to explain

6. Conclusions

At present, the system has been used in different colleges and universities and is widely popular because of its powerful function suitable for English teaching and has strong promotion value. It is believed that the networking of the system will facilitate the use of Intranet for multimedia teaching and paperless examination process of institutions.

Data Availability

The raw data supporting the conclusions of this article will be made available by the author, without undue reservation.

Conflicts of Interest

The author declared that he/she has no conflicts of interest regarding this work.

References

[1] Z. Ghahramani, “Probabilistic machine learning and artificial intelligence,” Nature, vol. 521, no. 7553, pp. 452–459, 2015.

[2] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, “Deep learning for smart manufacturing: methods and applications,” Journal of Manufacturing Systems, vol. 48, pp. 144–156, 2018.

[3] S. Khan and T. Yairi, “A review on the application of deep learning in system health management,” Mechanical Systems and Signal Processing, vol. 107, pp. 241–265, 2018.

[4] K. H. Yu, A. L. Beam, and I. S. Kohane, “Artificial intelligence in healthcare,” Nature Biomedical Engineering, vol. 2, no. 10, pp. 719–731, 2018.

[5] R. Cioffi, M. Travaglioni, G. Piscitelli, A. Petrillo, and F. De Felice, “Artificial intelligence and machine learning applications in smart production: progress, trends, and directions,” Sustainability, vol. 12, no. 2, p. 492, 2020.

[6] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, “Machine learning for Internet of Things data analysis: a survey,” Digital Communications and Networks, vol. 4, no. 3, pp. 161–175, 2018.

[7] N. Hoic-Bozic, V. Mornar, and I. Boticki, “A blended learning approach to course design and implementation,” IEEE Transactions on Education, vol. 52, no. 1, pp. 19–30, 2009.

[8] J. A. Cruz and D. S. Wishart, “Applications of machine learning in cancer prediction and prognosis,” Cancer Informatics, vol. 2, p. 117693510600200, 2006.

[9] L. Wen, L. Gao, and X. Li, “A new deep transfer learning based on sparse auto-encoder for fault diagnosis,” IEEE Transactions on systems, man, and cybernetics: systems, vol. 49, no. 1, pp. 136–144, 2019.

[10] L. Buitinck, G. Louppe, M. Blondel et al., “API design for machine learning software: experiences from the scikit-learn project,” 2013, http://arxiv.org/abs/1309.0238.

[11] P. D. Sorlie, L. M. Avilés-Santa, S. Wassertheil-Smoller et al., “Design and implementation of the Hispanic community health study/study of Latinos,” Annals of Epidemiology, vol. 20, no. 8, pp. 629–641, 2010.

[12] A. Abbasi and H. Chen, “Applying authorship analysis to extremist-group web forum messages,” IEEE Intelligent Systems, vol. 20, no. 5, pp. 67–75, 2005.

[13] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: distributed machine learning for on-device intelligence,” 2016, http://arxiv.org/abs/1610.02527.

[14] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” ACM Computing Surveys (CSUR), vol. 54, no. 6, pp. 1–35, 2021.

[15] P. Zheng, Z. Sang, R. Y. Zhong et al., “Smart manufacturing systems for Industry 4.0: conceptual framework, scenarios, and future perspectives. Frontiers of,” Mechanical Engineering, vol. 13, no. 2, pp. 137–150, 2018.

[16] X. Lin, Y. Rivenson, N. T. Yardimci et al., “All-optical machine learning using diffractive deep neural networks,” Science, vol. 361, no. 6406, pp. 1004–1008, 2018.

[17] J. Li, Z. Zhou, J. Wu et al., “Decentralized on-demand energy supply for blockchain in internet of things: a microgrids approach,” IEEE Transactions on Computational Social Systems, vol. 6, no. 6, pp. 1395–1406, 2019.

[18] W. Duan, J. Gu, M. Wen, G. Zhang, Y. Ji, and S. Mumtaz, “Emerging technologies for 5G-IoV networks: applications, trends and opportunities,” IEEE Network, vol. 34, no. 5, pp. 283–289, 2020.

[19] Z. H. A. N. G. Zheng-wan, W. U. Di, and Z. H. A. N. G. Chun-jiong, “Study of cellular traffic prediction based on multi-channel sparse LSTM,” Computer Science, vol. 48, no. 6, pp. 296–300, 2021.
[20] P. An, Z. Wang, and C. Zhang, “Ensemble unsupervised auto-encoders and Gaussian mixture model for cyberattack detection,” Information Processing & Management, vol. 59, no. 2, article 102844, 2022.
[21] F. M. Abd Algalil and S. P. Zambare, “Effects of temperature on the development of Calliphorid fly of forensic importance Chrysomya megacephala (Fabricius, 1794),” Indian Journal of Applied Research, vol. 5, no. 2, pp. 767–769, 2015.
[22] H. Shi, M. Xu, and R. Li, “Deep learning for household load forecasting—a novel pooling deep RNN,” IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 5271–5280, 2018.
[23] E. Rosten, R. Porter, and T. Drummond, “Faster and better: a machine learning approach to corner detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 105–119, 2010.
[24] T. Wuest, D. Weimer, C. Irgens, and K. D. Thoben, “Machine learning in manufacturing: advantages, challenges, and applications,” Production & Manufacturing Research, vol. 4, no. 1, pp. 23–45, 2016.
[25] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Machine learning: a review of classification and combining techniques,” Artificial Intelligence Review, vol. 26, no. 3, pp. 159–190, 2006.