Design and Application of Online Courses under the Threshold of Smart Innovation Education

Qin Wang[1*], Anya Xiong[2], Huirong Zhu[3]
Chongqing University of Posts and Telecommunications, Nan An Chongqing 400065 China[1, 2]
ChongQing JiaoTong University, Nan An Cahongqing 400074 China[3]

***Abstract*—With the rapid development of the Internet and the growing demand for education, a new online teaching mode, massive open online courses (MOOC), emerged in 2012. To address the problems of sparse data and poor recommendation effect in online course recommendation, this paper introduces deep learning into course recommendation and proposes an auxiliary information-based neural network model (IUNeu), on the basis of which a collaborative neural network filtering model (FIUNeu) is obtained by improving it. Firstly, the principles and technical details of the deep learning base model are studied in depth to provide technical support for course recommendation models and online learning recommendation systems. In this paper, based on the existing neural matrix decomposition model (NeuMF), we combine user information and course information and consider the interaction relationship between them to improve the accuracy of the model to represent users and courses. The neural network model of auxiliary information (IUNeu) is incorporated into the online learning platform, and the system development is completed with the design of front and back-end separation, realizing the functions of the online learning module, course collection module, course recommendation module, and resource download module. Finally, the experimental results are analyzed: under the same experimental conditions, the test experiments are repeated 10 times, and the RMSE calculation results are averaged. The RMSE value of the neural network collaborative filtering model (FIUNeu) proposed in this paper based on deep learning is 0.85517, which is the best performance and has a high accuracy rate of rating prediction, and is useful for alleviating the data sparsity problem.**

Keywords*—Massive open online courses (MOOC); deep learning; collaborative neural network filtering model (FIONeu); course recommendation; online learning recommendation system

I. INTRODUCTION

In latest years, MOOC structures such as Coursera, edX, and Scholastic Online have flourished and swiftly attracted a giant variety of learners. The ease with which customers can get entry to a giant variety of magnificent mastering assets in moot classification systems has significantly facilitated expertise sharing and online learning and supplied many possibilities for character newbies to get hold of training [1]. However, customers are regularly pressured via the "information overload" of the exploding quantity of publications on the Mootools platform. How to assist customers discover the getting to know assets they want and make correct tips has grown to be one of the most urgent issues [2].

Currently, Li Xiaojun et al. crawled the getting-to-know records on MUOC online for experiments, and the effects confirmed that the overall performance of the IUNeu mannequin grows quicker than the NeuMF mannequin as the vector size and the number of endorsed publications expand [3]; Xiaoyan ZD et al. proposed a bipartite diagram context collaborative filtering algorithm primarily based on the traits of MOOC platforms, which improves the advice fine of the algorithm with the aid of preprocessing the dataset and setting up distinct linkage quantification values [4]; Cheng Yan proposed a prolonged ant colony algorithm for fixing the suggestion trouble of studying paths, which takes into account the comparison of getting to know paths via the crew of newbies and the traits of goal customers in phrases of know-how stage and getting to know fashion when making advice selections [5]. Zhang H et al. blended with the traits of the MOOC platform, proposed MCRS to make splendid upgrades in path suggestion mannequin and suggestion algorithm, and the direction suggestion records are transferred to MySQL via Sqoop to obtain well-timed comments and enhance the path retrieval effectivity of customers [6]; Sun X et al. designed a customized suggestion device mannequin for on-line gaining knowledge of assets primarily based on the traits of learners' mastering conduct and on-line getting to know assets in the getting to know the process, mixed with collaborative filtering algorithm [7]; Yin H used the evaluate matrix technique to set up the hobby contrast matrix and consumer activity comments model, and optimized the suggestion algorithm the use of the hobby remarks model, accordingly realizing the suggestion of distance getting to know sources in MOOC training mode [8].

With the quickly flourish of the Internet and the growing demand for education, a new online teaching model, large-scale online open courses, emerged in 2012. To tackle the troubles of sparse facts and negative advice impact in online path advice, this paper introduces deep learning into course recommendation and proposes an auxiliary information-based neural network model (IUNeu). First of all, the principle and technical details of the basic model of deep learning are deeply studied to provide technical support for the course recommendation model and online learning recommendation system. Based on the existing Neural matrix decomposition model (NeuMF), this paper combines user information and course information, and considers the interaction between them to improve the accuracy of the model to represent users and courses. The auxiliary information neural network model (IUNeu) is integrated into the online learning platform, and the system is developed by using the front-end and back-end separation design, and the online learning module, course collection module, course recommendation module, resource download module and other functions are realized. Finally, the experimental results were analyzed: under the same experimental conditions, the test experiment was repeated for 10 times, and the RMSE calculation results were averaged. The RMSE value of the neural network model proposed in this paper based on deep learning was 0.85517, with the best performance and high score prediction accuracy, which played a great role in alleviating the problem of data sparsity. This paper innovatively proposes a neural network model based on auxiliary information, which can accurately recommend courses for users and effectively solve the problems of sparse data and poor recommendation effect.

II. Deep Learning Techniques in MOOC

Deep learning is based on an artificial neural network (ANN), which is a mathematical model that can learn by itself and constantly adjust to bring the results closer to reality. As early as 1943, mathematical logician W Pitts and psychologist W S McCulloch proposed the concept of neuron, the core structure of neural network, which gradually developed into a research hotspot in the field of artificial intelligence [9]. With the development of artificial neural networks, the structure of the network becomes more and more complex, and people call artificial neural networks "deep learning". In current years, deep studying methods have been extensively used in the fields of information, medicine, economics, control, transportation, psychology, etc., due to their excellent processing and processing capabilities in image processing, prediction and classification, natural language processing, intelligent robotics, signal processing, and optimal combination [10]. The following is a detailed description of the neural network structures used in the study.

A. Convolutional Neural Networks

Convolutional neural networks, a typical model of deep learning models, do not use conventional matrix operations but use convolutional operations instead, and have one or more layers of convolutional layers. The structure of a convolutional neural network is made of an entry layer, a hidden layer, and an exit layer like the structure of a normal neural network [11]. Among them, the implicit layer usually consists of convolution, pooling, and full connectivity, while the unique aspect of convolutional neural networks is the convolutional operation and pooling operation, this is the core of convolutional neural networks.

The core building blocks of the convolutional layer are convolutional kernels, each of which consists of one or more elements. Like the neurons of the feedforward network, the elements constituting the convolutional kernel have their own amount of bias and weight coefficients, respectively. The convolution kernel has three parameters, which are the convolution kernel size, the convolution kernel cross step, and the facet padding of the entry data. The convolution kernel dimension is the measurement of the shift matrix used for the convolution operation, and the convolution kernel dimension is any fee smaller than the dimension of the matrix being convolved; the large the dimension of the convolution kernel, the richer function records can be realized from the convolved photo [12]. The convolution step, on the different hand, defines the dimension of the measurement of the convolution kernel shifted by way of the subsequent convolution operation with recognition of the preceding convolution operation. The fill is a proposed approach to make bigger the dimension of the function map earlier than the convolution operation in order to counteract the shrinking measurement of the characteristic map in the computation and is most frequently utilized via the usage of zero to fill the boundary or with the aid of repeating the boundary facts to gain the purpose of filling [13]. Based on these three parameters being able to calculate the size of the feature map obtained after convolution, assuming that the convolution input of layer l+1 is Z^t with size equal to L_l, the convolution output Z^{t+1} of layer l+1 is:

$$Z^{l+1}(i,j)=\sum_{k=1}^{K_l}\sum_{x=1}^{f_w}\sum_{y=1}^{f_h}\left[Z_k^l\left(s_0 i+x, s_0 j+y\right)w_k^{l+1}(x,y)\right]+b \tag{1}$$

Here, assuming that the length and width of the feature map are equal, the output feature map size is $L_{l+1}=\frac{L_l+2p-f}{s_0}+1$
where b is the amount of deviation, $Z(i,j)$ corresponds to the pixel with (i,j) coordinates of the feature map, $(i,j) \in \{0,1,\ldots Ll+1\}$, K is the number of channels of the feature map, and the three parameters of the convolution kernel: the convolution step size is so, the convolution kernel size is f, and the number of fill layers is p.

Neural networks possess a high degree of nonlinearity, and convolutional neural networks are no exception to this rule, and the activation function is a part of the process that produces this important effect [14]. The commonly used activation functions are linear rectifier function, hyperbolic tangent activation function, and sigmoid() function, while convolutional neural networks usually use linear rectifier function as activation function.

For matrix data, a pooling operation is proposed to extract the significant features representing the matrix data, i.e., to select a maximum or average number as the features of a region of data by means of aggregation statistics. On the one hand, the pooling layer enables the down sampling of data, thus reducing the number of model parameters, and on the other hand, the pooling layer is a kind of data dimensionality reduction, enabling the compression of data features, effectively reducing the redundant information contained in the data, streamlining the complexity of the model, and reducing the unnecessary computation and memory consumption of the model [15]. In addition, the pooling layer enables nonlinearity and invariance of data transformations, which can expand the perceptual field while preserving data characteristics. Pooling operations can be classified into three categories: Lp pooling, random/hybrid pooling, and spectral pooling. Mean pooling and maximum pooling in Lp pooling are the most frequently used pooling methods for pooling layers in convolutional neural networks, and the general representation of Lp pooling is:

$$A_k^l(i,j) = \left[\sum_{x=1}^{f}\sum_{y=1}^{f} A_k^l\left(s_0 i + x, s_0 j + y\right)^p\right]^{\frac{1}{p}} \tag{2}$$

Same as the convolutional layer, here so denotes the pooling step, (i, j) denotes the pixel points. *P* denotes the pre-specified parameter, when *P*=1, its capacity that the implied fee is taken as the pooling end result in the pooling region, i.e., mean pooling; when *P*=∞, it means that the great value is taken as the pooling result in the pooling region, i.e., maximum pooling. However, both methods lose some information about the image.

B. TextCNN

TextCNN has great overall performance in textual content classification problems. TextCNN has superb overall performance in textual content classification problems. In particular, TextCNN can effectively extract the shallow feature information of natural utterances of short texts, and it is extensively used in the area of brief texts with its benefit of true outcomes and speedy pace.

Intuitively understood, the convolutional operation of TextCNN is performed based on a one-dimensional convolutional kernel to obtain an n-dimensional feature representation of the text [16]. TextCNN uses pre-trained word vectors word embeddings as input, i.e., a sentence consisting of n words is represented by an $n \times k$ matrix, where k is the dimension of the word vector corresponding to each word, here the k-dimensional word embeddings of the ith word in the sentence are represented using $x_i \in \Re^k$. First, a convolution operation $w \in \Re^{hk}$ is performed to obtain a feature $c_i : c_i = f\left(w \cdot x_{i,i+h-1} + b\right)$ by using a convolution kernel on $x_{i,i+h-1}$. Where $x_{i,i+h-1}$ denotes the vector feature matrix of the i-th to $i+h$-1th word in the input matrix, w is the weight matrix of dimension $h \times k$, b is a functional deviation and $f()$ is a non-linear function. i starts from 1 and is added 1 each time until i equals n-h+1, and the c_i obtained each time is stitched to get the vector $c=\left[c_1,c_2,\ldots,c_{n-h+1}\right]$ after one convolution. Multiple convolution kernels are set up by defining different h, in order to extract different feature carriers and use them together as the convolution layer's output. Then, a maximum pooling operation is performed, i.e., a fixed-length vector representation is obtained by selecting the largest feature from the c vectors obtained from different convolution kernels and stitching them together to form a new vector [17]. Last, the globally connected layer activated using softmax() is accessed to output the probability values corresponding to each category.

C. Gated Circulation Unit

The gated recurrent unit (GRU), like the long- and non-permanent reminiscence network, is an enchantment on the trendy recurrent neural network, whose mannequin shape is proven in Fig. 1.

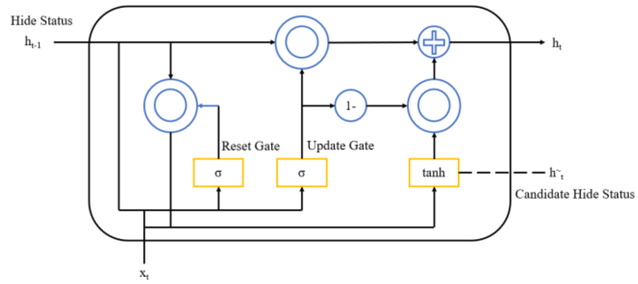

Fig. 1. GRU model structure.

The advantage of GRU over LSTM is that it uses a gating mechanism, while significantly reducing the problem of vanishing gradients. The GRU network structure contains two gating structures, one for update gating and one for reset gating. As a variant of recurrent neural networks, the gating mechanism controls which information can be transmitted through the network model to the output layer. In addition, these two gating structures do not discard data because of the length of the depth of the operation or the uselessness of some data for the resultant output but are able to preserve various types of information in the data sequence over time.

D. Self-attentive Mechanism

Self-attention mechanisms were first proposed as an important component of Transformer (ML architecture that has been successfully applied to various NLP tasks, especially sequence-to-sequence tasks, such as machine translation and text generation). The attention mechanism learns as the model fits the data results, and instead of just performing a comprehensive analysis of the full information, the work is focused on discovering important local information and analyzing it [18]. The self-attention mechanism is used to discover the parts of the input data that deserve more attention by interacting between each input two-by-two. The result of the interaction and aggregation of the input data is the output of the self-attentive mechanism, i.e., the attention score.

The inputs and outputs of the self-attentive mechanism are sequences, and assuming xi is the input sequence, the detailed derivation of the procedure is as below.

1) Each input is obtained by multiplying by a matrix to get a_i, and then multiplying by 3 different vectors W^q, W^k, W^v to get q^i, k^i, v^i respectively.

2) Calculate the weight of q and k by the similarity calculation method, for example, take the first input as an example, $\alpha_{1,i} = q^1 \cdot k^i / \sqrt{d}$, and then $\hat{\alpha}_{1,i} = \exp\left(\alpha_{1,i}\right) / \sum \exp\left(\alpha_{1,i}\right)$ need to be normalized by softmax.

3) Multiply v_i with the corresponding $a_{1,i}$ and then add them to get the output vector b_1 of the first input.

4) Repeat the above steps to obtain the corresponding output vector of the input sequence.

III. Deep Learning-based Course Recommendation Model

A. *Neural Matrix Decomposition Model*

The NeuMF model is made up of two parts: the GMF (generalized matrix factorization) model and the MLP (multilayer perceptron) model. In the GMF model, a linear kernel is applied to simulate the interaction between features; in the MLP model, a nonlinear kernel is applied to learn the interaction function [19]. Thus, the NeuMF model has both linear and nonlinear modeling capabilities, and the specific application structure of the model in course recommendation is shown in Fig. 2. In the figure, the left side of the model is the GMF part and the right side is the MLP part. The model is bottom-up, starting with the input layer, which contains one-hot codes for user and item IDs, and passes these codes to the embedded layer to represent the corresponding users and items. The output of the two-part molecular model is connected, and the predicted values $\hat{y}$ are obtained in the output layer by the Sigmoid activation function, which $\hat{y} \in \{0,1\}$ indicates the user's preference for the item (1 for like and 0 for dislike). The weights of each layer are trained by backpropagation.

The model-specific expressions are:

$$\hat{y}_{u,i} = f_{out}\left(W_{out}\left[X^{\mathrm{GMF}}, X^{\mathrm{MLP}}\right]\right) \tag{3}$$

$$X^{\mathrm{GMF}} = p_{\mathrm{u}}^{\mathrm{GMF}} \cdot q_{\mathrm{i}}^{\mathrm{GMF}} \tag{4}$$

$$X^{\mathrm{MLP}} = f_L\left(W_L\left(\cdots f_1\left(W_1\left[p_{\mathrm{u}}^{\mathrm{MLP}}, q_{\mathrm{i}}^{\mathrm{MLP}}\right]+b_1\right)\cdots\right)+b_L\right) \tag{5}$$

where: X^{GMF} and $X^{MLP}$are the potential feature vectors of the *GMF* and *MLP* model parts, respectively, $\hat{y}_{\mathrm{u,i}}$ is the predicted value, $f_{out,}$ and W_{out} are the activating features weights of the export layer, respectively, p_u^{GMF} is the user potential feature vector in X^{GMF}, q_i^{GMF} is the item potential feature vector in X^{GMF}, p_u^{MLP} is the user potential feature vector in X^{MLP}, q_i^{MLP} is the item potential feature vector in X^{MLP}, [] denotes the internal element connection, f_L and W_L are the activation function weights of the Lth layer, separately, and b_1 and b_L are the biases of the 1st and Lth layers, separately.

The detailed training procedure of the NeuMF model is as following: 1) Infant layer: extract the ID information of users and courses in the training set and encode them using one-hot. 2) Embedded layer: use the infant layer as the infant, independent embedded layers are used in the model, the GMF model on the left side of the model and the MLP model on the right side, each layer selects the linear rectification function as the activation function and outputs a $1\times n$ dimensional matrix. 3) Output layer: The outputs of the GMF and MLP models are linked first and last, and the results are mapped to [0, 1.0] by the Sigmoid activation function to obtain the prediction results $\hat{y}$.

The NeuMF model is applied to analyze the data of MUCN, and compared with the traditional user-based k-nearest neighbor algorithm, and MFALS algorithm, and the hit rate is used as the index for performance evaluation in the experiments by the leave-one-out method. It is found that the hit rate of the UserKNN algorithm is 0.075 because it does not involve the underlying feature vector length *n*; MFALS involves the construction of the user matrix and item matrix, and its hit rate varies with the increase of n and finally stabilizes at about 0.110; the hit rate of NeuMF model is higher than that of UserKNN algorithm and MFALS algorithm, and its hit rate is greater than 0.500 [20]. The reason for this is that the UserKNN algorithm uses similar historical behavior data of the target user and k neighbors to predict the course resources that users may like in the future, while the MFALS algorithm recommends course resources to users based on the high or low item history ratings, and both algorithms cannot better meet the actual demand of online course resource recommendation. Therefore, this study considers improving the NeuMF model by transforming the recommendation problem into a classification problem. A more efficient neural network model based on auxiliary information (item and user information) is proposed. The model is constructed by analyzing users' historical learning records and implementing a classification function with users and courses as the minimum unit, with positive classes indicating users' likes and negative classes indicating users' dislikes. The problems of the above UserKNN and MFALS algorithms for course recommendation are improved in the model to improve the recommendation effect.

B. *IUNeu Model*

In the IUNeu model, the input information in the Input layer is divided into two parts: 1) user-related information, such as user ID, user's gender, occupation, etc., where the information other than user ID is called user auxiliary information [21]; 2) course-related information, such as course ID, course's label, course category, etc., where the information other than course ID is called course auxiliary information. The form of the input information is [User ID, Male, Front-end Engineer, ...], [Course ID, Front-end Technology, JS Basic Design, ...]. The specific model architecture is illustrated in Fig 3. It can be observed that the user-course auxiliary information (hereafter collectively referred to as auxiliary information) is involved in all stages of the overall model. During training, the auxiliary information is used as input along with the user ID and course ID; in the GMF (MLP) composition model, the user ID, course ID, and user-course auxiliary information are fused to participate in linear modeling (nonlinear modeling); and in the Sigmoid activation function output, the auxiliary information is also fused.

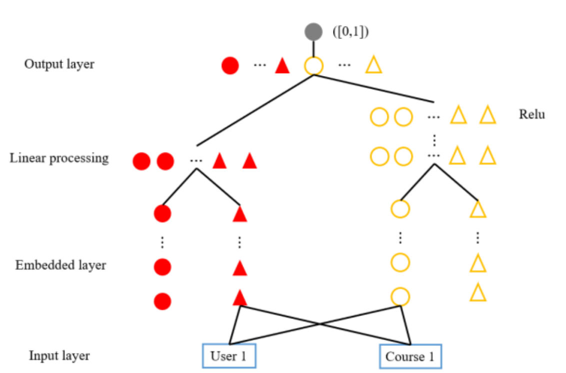

Fig. 2. Structure of the NeuMF model.

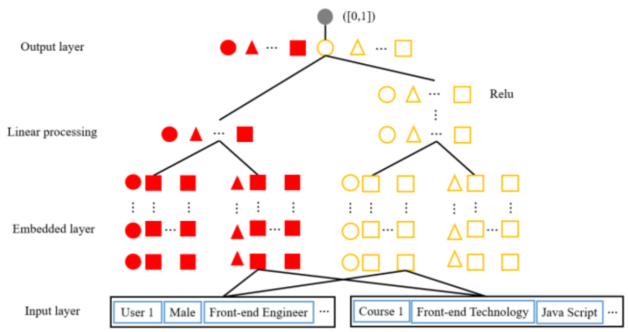

Fig. 3. IUNeu model structure.

The X^{GMF}, and X^{MLP} expressions in the IUNeu model structure improvement equation (6) are:

$$X^{\mathrm{GMF}} = p_{u}^{\mathrm{GMF}} \cdot q_{i}^{\mathrm{GMF}} = \left(u_0^{\mathrm{GMF}} \oplus u_1^{\mathrm{GMF}} \oplus \cdots \oplus u_m^{\mathrm{GMF}}\right) \cdot \left(c_0^{\mathrm{GMF}} \oplus c_1^{\mathrm{GMF}} \oplus \cdots \oplus c_m^{\mathrm{GMF}}\right) \tag{6}$$

$$\begin{aligned} X^{\mathrm{MLP}} &= f_L\left(W_L\left(\cdots f_1\left(W_1\left[p_u^{\mathrm{MLP}}, q_i^{\mathrm{MLP}}\right] + b_1\right)\cdots\right) + b_L\right) \\ &= f_L\left(W_L\left(\cdots f_1\left(W_1\begin{bmatrix} u_0^{\mathrm{MLP}} \oplus u_1^{\mathrm{MLP}} \oplus \cdots \\ \oplus u_m^{\mathrm{MLP}}, \end{bmatrix} c_0^{\mathrm{MLP}} \oplus c_1^{\mathrm{MLP}} \oplus \cdots \oplus c_m^{\mathrm{MLP}}\right] + b_1\right)\cdots\right) + b_L\right) \end{aligned} \tag{7}$$

where: $u_x^{\mathrm{GMF}}\left(u_x^{\mathrm{MLP}}\right)$ is the user word vector, x=0 denotes user ID, $x \neq 0$ denotes user auxiliary information, such as gender, occupation, etc.; $c_x^{\mathrm{GMF}}\left(c_x^{\mathrm{MLP}}\right)$ is the course word vector, x=0 denotes course ID, $x \neq 0$ denotes course auxiliary information, such as data mining, web front-end, etc.; $\oplus$ denotes the expansion of the original fixed $1 \times 1 \times n$ tensor into a $1 \times m \times n$ tensor, i.e., using the user (course) vector and the user auxiliary information (course auxiliary information) vectors to extend the original $1 \times n$-dimensional matrix, where n is the potential feature vector length and m is the infeed information length. Therefore, the outputs of GMF linear kernel modeling and MLP nonlinear kernel modeling are not only influenced by the IDs of the users and courses themselves but also determined by the content attributes of the users and courses themselves.

The detailed training procedure of the model is as following: 1) Infant layer: extract the information related to users and courses in the training set, including user gender, user occupation, course major category, course minor category, etc., and encode them with one-hot. 2) Embedding layer: use the infant layer as the infant, and output a $1 \times m \times n$ dimensional matrix. By flattening the operation, the length is extended to mn by adding auxiliary information to the original NeuMF model of length n, where $(m-1)n$ is the length of the auxiliary information vector. 3) Output layer: The GMF sub-model multiplies the flattened two results point by point as the linear output result, and the MLP sub-model connects the flattened results first and last to form a new vector and serve as the loser of the neural network (its activation function adopts the Relu activation function). Then the outputs of the GMF and MLP models are connected first and last and mapped to [0,1.0] by the Sigmoid activation function as the prediction results.

C. *Collaborative Filtering Model for Neural Networks with Fused Attention Relations*

Along with the development of the Internet, social networking has become an essential element of Internet applications. In addition to social networking platforms (e.g., Weibo, Zhihu, and Twitter) that are socially focused, many current Internet applications contain social elements, and many MOOC platforms, as one of the Internet applications, have social features [22]. In the AIMOC platform each user can follow their favorite users, which may include leaders in their own learning field, and become their fans, also called followers these followed people are called followers.

As a typical social network platform, Weibo can create a celebrity effect through the influence of well-known users who have a large number of followers. Unlike typical social networking platforms, MOOC platforms with social networking elements also have a celebrity effect because, as open e-learning platforms, the famous users here are more likely to be leaders in a certain learning field. Therefore, in MOOC platforms, the current interest preferences of these followers in the course are likely to influence their followers' choice of course.

Attention relationship as a specific expression in social networks, a recommendation algorithm AttentionRank+ is proposed, which considers the influence of the behavior of the followed on the behavior of the followers and improves the recommendation algorithm performance by fusing attention relations. In this article, we will also build a collaborative neural network filtering model (FIUNeu) based on the IUNeu model with fused attention relations to enhance the recommender effectiveness of the course recommender system.

The NeuMF model is a combination of models in the horizontal direction and is a fusion of recommendation algorithms. In this paper, the construction of the FIUNeu model is a fusion in the vertical direction, which is the fusion of recommendation results. The main idea of the FIUNeu model is to give the corresponding weight value to each course in the TopN courses recommended to the current user based on the IUNeu model by calculation. The FIUNeu model is described as follows:

Let the current user be u, the number of users he follows is p, and the maximum number of users he follows be M. First, the TopN courses $(c_1, c_2, \cdots, c_{n-1}, c_n)$ are recommended to user u by the IUNeu model, and the initial weights of these K courses are set to $(n, n\text{-}1, \cdots, 2, 1)$. For the ith recommended course c_i of the current user u, where $i \in (1, 2, \cdots, n\text{-}1, n)$, and q users among all users followed by this user choose this course, the weight formula for this course c_i is

$$f(i) = \frac{w_1(n-i+1) + w_2\left(\alpha \frac{q}{p} + \beta \frac{p}{M}\right)n}{2n} \left(w_1 + w_2 = 1, \alpha + \beta = 1\right) \tag{8}$$

Where w_1 and w_2 denote the proportion of the influence of the IUNeu model and attention relationship on the value of course c_i weight, α and β denote the proportion of attention and the proportion of the number of attentions on the influence of attention relationship, in this paper FIUNeu model for the above parameters, are selected as 0.5 through the above formula for this TopN course weights are recalculated, then according to the weights of this TopN course re-ranking Finally, the TopK courses are recommended to the current users.

The model framework of FIUNeu is illustrated in Fig. 4:

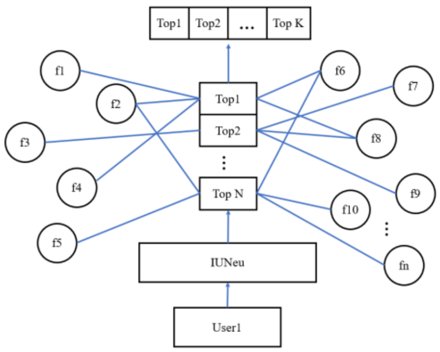

Fig. 4. FIUNeu model.

IV. Design and Implementation of an Online Learning Recommendation System

A. Overall System Design

1) System logical architecture design: The logical architecture of the system consists of five levels: the base device level, the data level, the middle level, the application level, and the user level. The device layer provides the basic hardware configuration such as network, server, operating system, and other hardware, and is the basic link of the logical architecture. The data layer is responsible for data storage and management, like user data, course data, log data, etc. The middle level is responsible for handling the business logic, including the recommendation algorithm. The application level is primarily concerned with invoking the algorithms of the middle layer according to the user data and processing the commands sent by the users [23]. The user level is concerned with user interaction, sending user requests, and other operations.

2) System technical architecture design: The system adopts the most commonly used three-layer technical architecture: front-end display layer, business logic layer, and data storage layer. Users first send request instructions in the front-end display layer. The business logic layer receives the instructions and then calls each functional component and gets the required data from the data storage layer for business logic processing and finally returns the data to the front end [24]. The layered technology architecture can solve the problem of system coupling and adopt the development method of separating front and back ends, which can improve the scalability of the system and the speed of development.

Front-end display layer: this layer mainly interacts with users and sends requests, which are processed by the business logic layer and returned to this layer, and then rendered and presented to users, and user behavior data can be collected through the front-end display layer. The front-end pages in the system are developed based on the Bootstrap framework, JavaScript, CSS3, and other technologies. The simple and friendly system interface is more user-friendly and can enhance user stickiness.

Business logic layer: This layer contains a large number of functional components, which are charged with the logical processing of business. The system adopts the Django framework and divides the business logic layer into three layers through Django's MTV pattern: the Model layer is responsible for reading, writing, and updating data from the database, the Template layer has rich template functions and is responsible for encapsulating and constructing HTML, and the View layer performs business processing and is the bridge between the Model and Template layers. Different roles have different business logic.

Data Store Layer: The data store layer is primarily responsible for storing user behavior data, log data, and other source data and a relational database is more convenient for developers and backend managers to maintain and manage. Therefore, the data storage layer uses the MySQL database to store and manage data and adopts md5 encryption to ensure the security of user data and avoid security risks caused by data leakage.

B. System Functional Module Design

The online mastering suggestion device is divided into three roles: pupil user, instructor user, and administrator. Student customers can log in to the gadget to learn about the course, consider the course, whole path assignments, download route sources, and different operations. Teachers can log in to the gadget to keep and manipulate route resources, pupil records, and private information, and directors can view and alter trainer information, scholar statistics, and route information.

The practical modules of the gadget are basically divided into 4 core modules: consumer registration and login module, online mastering module, route advice module, and historical past administration module.

1) User registration and login module If a new user first registers, fill in the registration page with standard information including a verification code, an account password, and other information, and then jump to the system home page after the background verification and storage of data, the registration page is concise and more user-friendly [25]. If the user is registered in the database, the user is prompted to log in directly. The cell phone verification code verification function uses the Ronglian communication cloud server to achieve the verification code acquisition. Enter the personal home page to improve the basic information, such as a nickname, birthday, gender, address, and other information.

The main classes involved in the user login module are: LoginFormn(), DynamicLoginView(), SendSmsView(), LoginView(), and ChangePwdForm(), which correspond to the functions of getting login form content, logging in after registration, verifying cell phone verification code, login judgment, and changing password, respectively.

2) The online mastering module is one of the core features of the online studying advice system, which makes hints via the behavioral information generated by way of the customers of this module. Users can find out about the course, and whole route assignments, download direction resources, and consider the direction and different features in this module. After logging into the system, customers can pick out the direction they are involved in from the direction list on the domestic web page and begin getting to know the course or proceed to get to know the taking part guides in the private middle My Courses page. After getting into the route important points page, customers can view the route introduction, trainer and organization profiles, path details, etc. They can pick to collect, begin learning, etc. After clicking begin learning, they can choose the corresponding chapter for learning, commenting, and scoring. In addition to learning the course, you can also download the materials and view the course assignments on the course page.

The main classes involved in the online learning module are: CourseListView(), CourseLessonView(), CourseDetailView(), CourseCommentsView(), VedioView(), and HomeworkView(), which are corresponding to get course list, get course chapter list, course comments, video learning, homework completion, and other functions.

3) The course recommender mode offers users a more precise course push, which effectively enhances users' learning productivity. The system uses different recommendation algorithms for users with different login statuses. For users who are not logged in and users who have not studied any courses, the popularity-based recommendation algorithm is used to make recommendations based on the number of course learners, and for users who are logged in, the hierarchical attention recommendation algorithm DHRAA algorithm, which integrates auxiliary information, is proposed in this paper to make recommendations.

The popularity-based recommendation algorithm is to sort the courses in reverse order according to the number of course learners, exclude the courses currently taken by users and select the top five courses for a recommendation, which can effectively tackle the user cold start issue and improve the recommendation efficiency.

This paper proposes a deep recommender algorithm according to auxiliary information that extracts the user id, user occupation, teacher institution, course id, course title, course review, course rating, and other information from the local database into the trained model, and calculates the list of course recommendations that match the user's characteristics.

The main classes involved in the course recommendation module are: CourseListView, CourseCommentsView, and CourseRecommView, which correspond to course list, course evaluation, course recommendation, and other functions respectively.

4) The backend administration mode contains two types of roles: teachers and administrators, mainly for user profile management, course material management, and permission management. User information management refers to the management of information of all roles on the platform and the addition, deletion, and checking of information of teachers' organizations. Course resource management refers to viewing and modifying course videos, course materials, course assignments, and other resources. When reviewing the content of comments submitted by users, administrators can query and delete user comments if they are found to be non-compliant.

C. System Database Design

The device makes use of MySQL database for records storage and management, and the statistics tables of every module include important information tables such as consumer statistics table, teacher records table, direction data table, route assessment information table, route project facts table, and direct aid facts table. The ER model is a conceptual design used to describe the relationships between entities, and it accurately describes various relevant data characteristics and their mutual restrictions. The EER model includes all the modeling concepts introduced by the ER model [26]. In addition, it includes the concepts of subclasses and super classes, as well as the concept of association types or categories, which are used to represent the association of objects of different entity types.

The user message table includes information about the user attributes needed in the recommender algorithm and stores the registration forms filled in by the user. The teacher information table stores the personal information of teachers, such as teacher id, teacher organization, teacher title, years of work, and other personal information, which is associated with the course organization table through the foreign key org_id. The path records desk includes the direction attribute facts required in the advice algorithm, which primarily data the small print of every difficulty route such as route description information, route situation level, quantity of path rookies, and different information. The path contrast records desk shops the person comparison and ranking facts of the course, which files the core statistics of the suggestion algorithm, and the route identification is set as the principal key in order to facilitate the advice algorithm to precisely stumble on the precise course. The direction venture information desk shops special facts about the venture together with the challenge title, theme options, etc. A route corresponds to more than one assignment. The course resources data table is used to store course-related resources, which are shown on the course details page, including software and documents, etc.

V. Experimental Tests and Results Analysis

A. Experimental Test Protocol

The experimental test computer hardware environment is as follows: processor Intel Corei9-9900K, graphics card GeForce RTX 2080T, memory 32GB The computer software environment used to run all experiments in the Python environment, using the Caffe tool to train and test the AlexNet network model. The data from the online learning platform of the China University MOOC National Excellence Course was used as the experimental data set.

To validate the proposed algorithm in this paper, the algorithm was objectively evaluated by calculating the root mean square error, which was calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,v}\left(R_{uv} - \hat{R}_{uv}\right)^2}{|R_t|}} \tag{9}$$

Where: R_{uv} denotes the true rating of course v by user u, $\hat{R}_{uv}$ denotes the predicted rating, and $|R_t|$ denotes the number of ratings in the test set. It can be seen that the lower the RMSE value, the higher the accuracy of the rating prediction and the better the performance of the recommendation system.

B. Analysis of Experimental Results

In order to verify the performance of the proposed algorithm, it is compared with other recommended algorithms, including Random method, UserAvg method, cooperative filtering method (CF) and non-negative matrix method (NMF). In order to ensure the stability of the algorithm, RMSE calculation results were averaged after repeated test for 10 times under the same experimental conditions. The comparison of results of different algorithms is shown in Fig. 5.

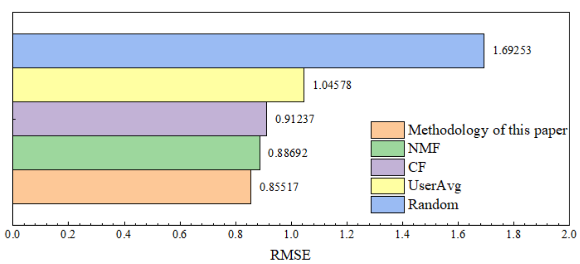

Fig. 5. Comparison of the results of different algorithms.

As can be seen from Fig. 5, RMSE value of Random method is 1.6925, RMSE value of UserAvg method is 1.0458, RMSE value of collaborative filtering method is 0.9124, and RMSE value of non-negative matrix method is 0.8869. It can be shown that the algorithm proposed in this paper performs optimally on the evaluation index of RMSE, with higher accuracy of rating prediction, which is useful for alleviating the problem of sparse rating data.

Next, in anticipation of verifying the influence of different experimental factors on the experimental results, this paper will verify the performance of the FIUNeu model, IUNeu model, and NeuMF model under different lengths of potential feature vectors and different TopNs based on different candidate course sets recommended by FIUNeu model

1) The candidate course set size is 30, and the performance of the three models is compared under different potential feature vector lengths and different TopN.

The FIUNeu model recommends the Top 10 courses twice on the basis of the 30-candidate course set recommended by the IUNeu model, while the IUNeu model and the NeuMF model directly select the Top 10, and verify the influence of different potential feature vectors on these three models on this basis. The vector lengths were selected as (4,8,12,16,20), and the experimental results are shown in Table I below:

TABLE I. WHEN THE COURSE SET IS 30, THE INFLUENCE OF DIFFERENT POTENTIAL FEATURE VECTOR LENGTHS ON THE MODEL

	NeuMF	IUNeu	FIUNeu
HR	0.5751	0.5939	0.5971
	0.5901	0.6081	0.6132
	0.5931	0.6064	0.6092
	0.5927	0.6042	0.6073
	0.5915	0.6026	0.6064
NDCG	0.3362	0.3632	0.3739
	0.3633	0.3775	0.3931
	0.3663	0.3782	0.3945
	0.3663	0.3768	0.3928
	0.3661	0.3764	0.3931

The FIUNeu model is based on the set of 30 candidate courses recommended by the IUNeu model, and the effect of different TopN on the three models is verified by the feature vector length selected as 8 and TopN selected as (5,10,15,20), as illustrated in Fig. 6.

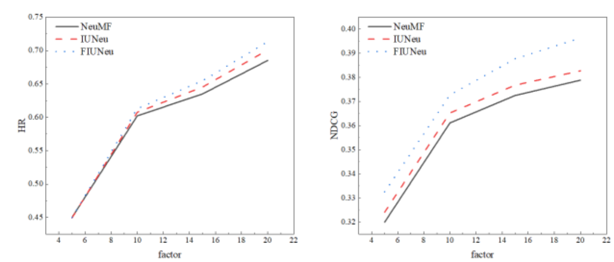

Fig. 6. Effect of different TopN on the model.

2) Here the candidate course set size is chosen to be 50, and the above two sets of experiments are conducted again and with the same experimental settings. As illustrated in Table II and Fig. 7.

TABLE II. WHEN THE COURSE SET IS 50, THE INFLUENCE OF DIFFERENT POTENTIAL FEATURE VECTOR LENGTHS ON THE MODEL

	NeuMF	IUNeu	FIUNeu
HR	0.575	0.594	0.597
	0.59	0.608	0.6142
	0.5929	0.6064	0.6092
	0.5927	0.6042	0.6073
	0.5914	0.6026	0.6067
NDCG	0.3356	0.3632	0.3741
	0.3628	0.3775	0.3931
	0.3661	0.3783	0.3938
	0.3656	0.3771	0.3945
	0.3659	0.3768	0.3928

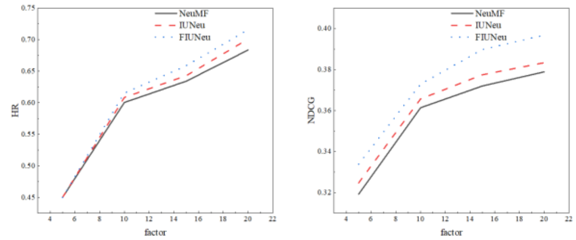

Fig. 7. Effect of different TopN on the model.

This paper only lists the experimental comparison between the case of 30 candidate course sets and 50 selected course sets recommended by IUNeu, and the experimental results based on other candidate course sets are roughly the same. From the above experimental results based on different lengths of potential feature vectors and different TopN settings, we can see that FIUNeu can indeed have better recommendation performance. From the above sets of experiments, we can find that FIUNeu, as an extension of the IUNeu model, is dependent on the IUNeu model for its performance, i.e., FIUNeu can make the IUNeu model perform better.

VI. Conclusion

In this paper, we study the course recommendation algorithm in the MU environment. Firstly, a neural network model based on auxiliary information (IUNeu) is proposed, and an improvement is made to the IUNeu model by fusing attention relations into the IUNeu model to obtain the FIUNeu model. The specific conclusions are as follows:

1) Considering the influence of the user and the content in the course on the model, a neural network model that fuses user and course information is constructed using the personal attributes of the household as well as the course information. On the basis of the original model features (user and course codes), the features such as the user's personal information and course categories are further fused to make the model's characterization of user and course features more accurate.

2) In this paper, the construction idea and the concrete implementation of the improved FIUNeu model based on the IUNeu model are presented, and the improved model FIUNeu is experimented on real data sets with the IUNeu model and NeuMEF model through MapReduce, so as to verify the performance of the three.

3) Under the same experimental conditions, the test experiments were repeated 10 times, the RMSE calculation results were averaged, and the RMSE value of the proposed neural network collaborative filtering model (FIUNeu) based on deep learning in this paper was 0.85517. The experimental results based on different potential feature vector lengths and different TopN settings show that FIUNeu can indeed have better recommendation performance. FIUNeu is an extension of the IUNeu model, its performance is dependent on the IUNeu model, i.e., FIUNeu can make the IUNeu model perform better.

The neural network model of auxiliary information proposed in this paper has some limitations. The cold start problem of the recommendation system, because it only relies on the user's feedback on the item, the newly added users and items need a period of data accumulation to reflect the recommendation effect. This will affect the new user experience. The comparison of algorithms in this paper are all offline tests, which cannot fully reflect the experience of real users. Due to the lack of real user feedback, online A/B testing is not possible. Therefore, in future work, researchers should work with companies to deploy algorithms into real production environments and optimize algorithms and system designs based on feedback from real users.

References

[1] Y. Li, D. Chen, Z. Zhan, “Research on personalized recommendation of MOOC resources based on ontology,” Interactive Technology and Smart Education, vol. 2022.

[2] Z. Wei, “Recommended methods for teaching resources in public English MOOC based on data chunking,” International Journal of Continuing Engineering Education and Life Long Learning, vol. 33, pp. 192-202, 2023.

[3] S.J. Li, H. Liu, H.X. Shi, et al. “Deep learning-based course recommendation model,” Journal of Zhejiang University: Engineering Edition, vol. 53, pp. 2139-2145, 2019.

[4] Z. Xiaoyan, B. Jie, “Research on MOOC system based on bipartite graph context collaborative filtering algorithm,” Proceedings of the 2019 8th International Conference on Software and Computer Applications, vol. 2019, pp. 154-158, 2019.

[5] Y. Cheng, “A group intelligence-based learning path recommendation method in online learning,” Journal of Systems Management, vol. 20, pp. 232-237, 2011.

[6] H. Zhang, T. Huang, Z. Lv, et al., “MCRS: A course recommendation system for MOOCs,” Multimedia Tools and Applications, vol. 77, pp. 7051-7069, 2018.

[7] X. Sun, Y. G. Wang, F. Qiu, “Research on personalized recommendation system of online learning resources based on collaborative filtering technology,” China Distance Education, vol. 15, pp. 78-82, 2012.

[8] H. Yin, “The recommendation method for distance learning resources of college English under the MOOC education mode,” International Journal of Continuing Engineering Education and Life Long Learning, vol. 32, pp. 265-278, 2022.

[9] F. He, H. Xue, R. Wang, “MOOC recommendation algorithm based on learning process sequence modeling and quantitative analysis,” International Conference on Neural Networks, Information, and Communication Engineering (NNICE). SPIE, vol. 12258, pp. 231-236, 2022.

[10] H. Zhang, T. Huang, Z. Lv, et al., “MCRS: A course recommendation system for MOOCs,” Multimedia Tools and Applications, vol. 77, pp. 7051-7069, 2018.

[11] H. Bao, Y. Li, Y. Zheng, “Social recommendation models and methods for large-scale online learning,” Modern Distance Education Research, vol. 3, pp. 94-103, 2018.

[12] X. He, P. Liu, W. Zhang, “Design and implementation of a unified MOOC recommendation system for social work major: Experiences and lessons,” 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). IEEE, vol. 1, pp. 219-223, 2017.

[13] S. X. Zhang, L. G. Zhang, “Construction of personalized recommendation service model for online learning resources,” China Medical Education Technology, vol. 31, pp. 172-176, 2017.

[14] L. Wang, “Collaborative filtering recommendation of music MOOC resources based on spark architecture,” Computational Intelligence and Neuroscience, vol. 2022, 2022.

[15] W. He, “Assessing the Impact of the MOOC Learning Platform on the Comprehensive Development of English Teachers at College Level under “Double First-Rate” by Utilization of the SWOT Analysis in Hunan Province, China,” Journal of Advanced Transportation, vol. 2022, 2022.

[16] H. Li, Z. Zhang, H. Guo, et al., “A personalized learning resource recommendation method from the perspective of deep learning,” Modern Distance Education Research, vol. 31, pp. 94-103, 2019.

[17] H. Yin, “The recommendation method for distance learning resources of college English under the MOOC education mode,” International Journal of Continuing Engineering Education and Life Long Learning, vol. 32, pp. 265-278, 2022.

[18] L. Yang, Y. H. Wei, K. X. Xiao, et al., “Research on personalized learning service mechanism driven by big data in education,” Research on e-learning, vol. 9, pp. 68-74, 2020.

[19] S. Assami, N. Daoudi, R. Ajhoun, “Implementation of a Machine Learning-Based MOOC Recommender System Using Learner Motivation Prediction,” International Journal of Engineering Pedagogy, vol. 12(5), 2022.

[20] Y. F. Dong, Y. A. Wang, Y. Dong, et al., “A review of online learning resource recommendation,” Computer Applications, vol. 2022: 0.

[21] D. Alahmadi, F. Alruwaili, “Deep Learning for MOOCs Course Recommendation Systems: State of the Art Survey,” International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies, vol. 12, pp. 1-9, 2021.

[22] H. Feng, H. Yi, X. Li, et al., “A review of privacy-preserving research on recommendation systems,” Journal of Frontiers of Computer Science and Technology, vol. 1, 2023.

[23] Y. Tian, Y. Sun, L. Zhang, et al., “Research on MOOC teaching mode in higher education based on deep learning,” Computational intelligence and neuroscience, vol. 2022, 2022.

[24] J. Dai, Q. Li, H. Chu, et al., “Breakthrough in smart education: a graph learning-based course recommendation system,” Journal of Software, vol. 33, pp. 3656-3672, 2022.

[25] Y. Yun, H. Dai, Y. Zhang, et al., “A review of personalized learning path recommendation,” Journal of Software, vol. 33, pp. 4590-4615, 2021.

[26] Y. Wang, B. Liang, W. Ji, et al., “A weighted multi-attribute method for personalized recommendation in MOOCs,” Proceedings of the 2nd International Conference on Crowd Science and Engineering, vol. 2017, pp. 44-49, 2022.