Hindawi
Scientific Programming
Volume 2022, Article ID 3437139, 10 pages
https://doi.org/10.1155/2022/3437139

Research Article

Career Recommendation for College Students Based on Deep Learning and Machine Learning

Qing Wan and Lin Ye

Shaanxi Institute of International Trade&Commerce, Shaanxi 712046, China

Correspondence should be addressed to Qing Wan; 368009050@qq.com

Received 4 November 2021; Revised 25 November 2021; Accepted 16 December 2021; Published 22 February 2022

Academic Editor: Bai Yuan Ding

With the popularization of higher education, China's higher education has moved from the stage of elite education to the stage of universal education, and the number of graduates is also increasing. At present, college students are facing tremendous pressure on employment. One is the huge number of jobs, the other is the difference in professional needs, and the third is the spread of job information, which makes it difficult for college students to find a job that suits them. In order to solve this dilemma, this paper analyzes the related technologies and in-depth basic theories of data mining. After introducing several traditional recommendation algorithms, the traditional convolutional network method is improved from three aspects: activation function, pooling strategy, and loss function. Finally, using the hybrid convolutional neural network, a career recommendation model for college students based on deep learning and machine learning is proposed, and simulation experiments are carried out on it. The main research work is as follows: (1) a hybrid convolutional neural network is proposed, which uses convolution operation to learn high-level features to achieve personalized employment recommendation; (2) the training optimization strategy of the hybrid convolutional neural network is studied, aiming at the activation function, pooling processing, and loss function, and the feasibility of the optimization method is verified through simulation experiments; (3) finally, according to the evaluation index of the recommendation algorithm (recall rate and $F1$-Score), the recall rate of the algorithm in this paper is nearly 15% higher than that of the DNN model. The experiment is compared with the traditional commonly used recommendation algorithm, and the comparative analysis of the experimental results proves the effectiveness of the algorithm for the employment recommendation of college students.

1. Introduction

As the enrollment scale of colleges and universities continues to expand, as of 2020, the gross enrollment rate of higher education in China has exceeded 50%, and higher education has entered the stage of universal education from the stage of elite education [1]. The employment demand of graduates is increasing year by year [2], and enterprises have higher and higher requirements for the professional skills and comprehensive quality of graduates. The traditional employment guidance model in the past can no longer adapt to the new employment supply and demand relationship. How to provide graduates with personalized and precise employment services has become the focus of the work of colleges and universities. Data show that in 2020, the number of college graduates nationwide will reach 8.74 million [3]. However, with the impact of the new crown epidemic and the intensified world trade protection, China's economy is under great downward pressure, which has gradually led to the emergence of college students" employment problems mainly realizing the following aspects:

(1) The number of employed people is huge, the employment peak lasts for a long time, and the situation is severe. The employment pressure of fresh graduates will not weaken in a relatively long period of time in the future.
(2) In the context of the rapid development of China economy, the demand for talents is constantly expanding, but some majors have not matched the

needs of China industry, resulting in structural unemployment for college students.

(3) Under the massive data situation of ‘information overload’ faced by college students when choosing employment, a wide range of job information makes it easy for college students to fall into information scrutiny fatigue and unable to clarify their job search needs, which leads to the need to spend more energy to find what is really suitable for them. The post information caused employment difficulties in the talent market [4]. The lack of a unified platform support and the inability to deeply match their own characteristics with the needs of enterprises result in a lack of scientific advice on big data employment guidance for college graduates.

In recent years, scholars from various universities and research institutions in China have also begun to explore related issues such as job matching and job matching. Kumar and Gupta and others put forward the job matching system theory [5]. Sajjadi et al. proposed the method basis of job matching in 2002 [6], and Figueiredo et al. took a different approach [7] and proposed the use of dynamic research methods to study the matching relationship between candidates and organizations. In 2006, under the leadership of Hiriyannaiah et al. and others, the dynamic person-post matching model became more and more mature, and dynamic fitting was proposed [8]. Koehn and Da’u, respectively, used fuzzy mathematics theory and grey system theory in their research and proposed a new matching model [9, 10]. With the advent of the data age, the simple theory of job matching with applicants can no longer provide accurate advice for college students’ employment choices. A large amount of data has made the inherent theoretical avatars useless. Many scholars have begun to introduce recommendation algorithms from other fields and analyze big data. Applied to job recommendation, traditional recommendation algorithms mainly include three types: collaborative filtering recommendation algorithm, content-based recommendation algorithm, and hybrid recommendation algorithm. The collaborative filtering recommendation algorithm can use existing historical activity interaction information and a series of user comparison information with the same behavior to make collaborative similarity matching recommendation without relying on user or item content information [11]. The content-based recommendation algorithm is an algorithm scheme that uses the user’s profile or project description information to recommend an algorithm scheme [12], which combines the attributes and characteristics of the project by means of data mining or information retrieval to build the user’s profile model. The hybrid recommendation algorithm aims to obtain better prediction or recommendation performance by combining the content-based recommendation algorithm and collaborative filtering recommendation algorithm, taking the advantages of the two. Traditional recommendation algorithms need to perform a lot of experiments on the selection of combination methods and the sequence of combinations to find a better combination method, and the weight distribution of the recommendation results obtained by different methods also needs to be experimentally tested and analyzed in order to make up for this. For this kind of defect, scholars introduced the concept of deep learning and started the research of the recommendation algorithm based on deep learning. Covington et al. proposed a deep neural network for YouTube video recommendation [13]. Cheng et al. proposed a wide and deep model to provide application recommendations for Google’s application mall [14]. Okura et al. proposed using the RNN model to recommend news information for Yahoo News [15].

This paper draws on the research results of Zhu Wei’s flight delay prediction algorithm combined with convolution processing [14], hoping to use the high-level feature learning ability of convolution processing to improve the quality of employment recommendation of college students and solve the “data trap” problem faced by college students in graduation selection.

2. Related Work

2.1. Hybrid Convolutional Neural Network. Convolutional neural network can be understood as a hierarchical model. The original data are used as the input of the model. Through a series of operation layers such as convolution operation, pooling operation, and nonlinear activation function mapping, the high-level abstract information is removed from the original extracted from the data input layer. Among them, the level of abstract operation is the “feedforward operation” in the convolutional neural network. The specific mathematical expression is as follows:

$$\begin{aligned} &x^1 \longrightarrow \omega^1 \longrightarrow x^2 \longrightarrow \cdots \longrightarrow x^{L-1} \longrightarrow \omega^{L-1} \longrightarrow x^L \longrightarrow \omega^L \longrightarrow z, \\ &z = f\left(x^L, y\right), \end{aligned} \tag{1}$$

where x^L represents the data input of the Lth layer, ω^L represents the weights related to the Lth layer, z represents the loss function of this calculation process, y represents the final true classification mark, and the function f uses ω^L as the calculation parameter.

2.1.1. Receptive Domain. In a convolutional neural network, the receptive field is expressed as the size of the region where the output features of each layer in the network are mapped to the input space. The receptive field represented by a certain feature can be described by the center position of the receptive field and its size. Figure 1 shows the receptive field performance of a convolution operation with a size of 3×3 and a step size of 1. The convolution kernel can obtain the same scale of receptive field as the large convolution kernel through multilayer superposition. The use of the small convolution kernel can deepen the network depth to enhance the capacity and complexity of the network and also reduce the number of training parameters.

2.1.2. Distributed Representation. In deep learning, convolutional neural networks will exhibit important characteristics of distributed representation. The characteristics of distributed representation can indicate that the convolutional neural network can form different representation modes by different convolution kernels, and different responses can be generated from different modes, and different feature representations can be abstracted according to different convolution kernels. Therefore, this feature can be used to implement different abstract feature representations using different convolution kernels to obtain more property feedback related to the input data.

2.1.3. Local Connection and Weight Sharing. The local connection method and the weight sharing method are both based on the local receptive domain, and the feature input satisfies the sparse but locally dense nature and is used to model the input data [17]. The local connection completes the feedback of various properties of the input data characteristics through the local information, and it is also the theoretical support for the distributed representation of the convolutional neural network. Each neuron in the convolutional neural network model only needs to be connected to a part of the forward layer. The local connection can not only obtain the different properties of the input data, but also reduce the risk of overfitting. The operation diagram of the local connection is shown in Figure 2.

The proposed weight sharing is used for the reduction of model training parameters, and the number of training parameters required for model training is reduced by sharing the convolution kernel [18]. Weight sharing uses convolution kernels with the same internal weights, and by using a common mode for feature detection, it can be understood that weight sharing is to use a template to abstract a certain feature of the data. Although the weight sharing mode can effectively reduce the scale of parameters required for model training, weight sharing can only extract a common feature property feedback, which is not as rich as the locally connected multiproperty feedback, and cannot mine more latent association representations at one time. The operation diagram of weight sharing is shown in Figure 3.

2.2. Traditional Recommendation Algorithm. The use of recommendation algorithms is currently the main effective measure to deal with the problem of information overload. It can provide users with appropriate personalized options in the case of massive data, avoid users from getting into information review fatigue, and save a lot of time and energy.

2.2.1. User-Based Collaborative Filtering Recommendation Algorithm. The user-based collaborative filtering recommendation algorithm mainly implements similar recommendations for users to find other users with similar behaviors or constructs a user's own preference model and performs a predictive score based on their own preference model to complete personalized recommendation. The core of the user-based collaborative filtering recommendation algorithm is similarity calculation. The main function of similarity calculation is to measure the similarity $S_{u,v}$ between user u and user v. The mathematical expression of Pearson similarity calculation is shown as follows:

$$S_{u,v} = \frac{\sum\left(z_{u,i} - \overline{Z_u}\right)\left(z_{v,i} - \overline{Z_v}\right)}{\sqrt{\sum\left(z_{u,i} - \overline{Z_u}\right)^2}\sqrt{\sum\left(z_{v,i} - \overline{Z_v}\right)^2}}. \tag{2}$$

$z_{u,i}$ and $z_{v,i}$ represent the actual scores of user u and user v on item i and $\overline{Z_u}$ and $\overline{Z_v}$ represent the average of all item scores of user u and user v, respectively.

2.2.2. Content-Based Recommendation Algorithm. The content-based recommendation algorithm is to use the content that users are interested in to calculate the similarity to achieve relevant recommendations. By means of data mining or information retrieval, a data file model belonging to the user is constructed by combining the attributes and characteristics of the project. The recommendation algorithm adopted by this algorithm is to use the content that the user is interested in to calculate the similarity to realize the recommendation. The algorithm uses x to represent the similarity evaluation of item s recommended to user c, where x is based on s, indicating that the recommendation of item c is obtained by quantitative calculation of other items with similar meta-information to item s.

2.2.3. Model-Based Collaborative Filtering Recommendation Algorithm. PMF is a model-based collaborative filtering recommendation algorithm. It reduces the dimension of the high-dimensional evaluation matrix to obtain the user hidden feature matrix and the item hidden feature matrix and further predicts the missing value of the evaluation matrix by calculating the inner product of the two. Its mathematical expression is as follows.

Given an $m \times n$ evaluation matrix R, two low-dimensional matrices U and V with rank d are used for fitting, and the fitting formula is as follows:

$$R \approx UV^T. \tag{3}$$

$U \in m \times d$; $V \in d \times n$, $d \leq \min\{m, n\}$; U represents the tendency of each user to d features; V represents the existence of d features in each item.

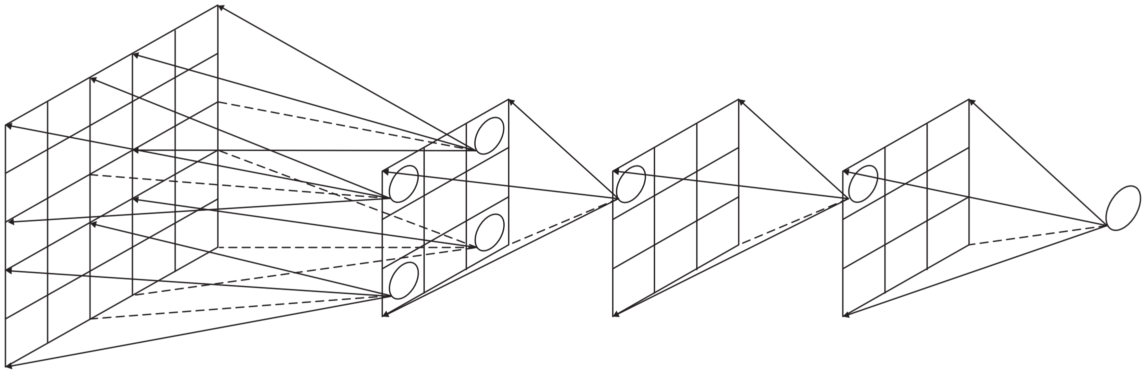

Figure 1: The receptive field with a size of 3×3 and a step length of 1 [16].

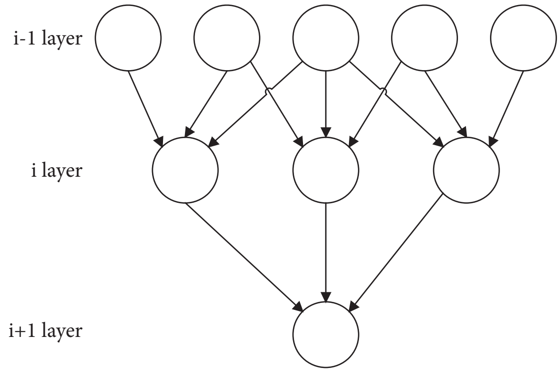

Figure 2: Schematic diagram of partial connection operation.

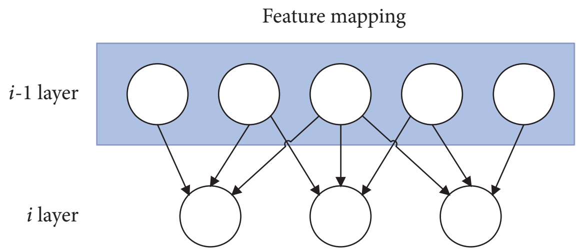

Figure 3: Schematic diagram of convolution weight sharing operation.

2.3. Data Feature Extraction. In this paper, gradient boosting tree is used for data feature processing, and the input features are filtered and processed in an integrated manner to eliminate the influence of useless features. The specific feature conversion flowchart is shown in Figure 4.

Before using the gradient boosting tree for feature conversion processing, it is necessary to perform feature dimensionality reduction processing on the type attributes of the one-hot encoding format, use the function of the Embedding operation to map low-level features to different feature spaces, and use high-level features as the mapping representation. The dimensionality reduction processing usually uses a neural network for training, and its standard algorithm expression is shown in the following formula:

$$y_z = \mathscr{F}\left(W_z x_z + b_z\right). \tag{4}$$

Among them, z is used to represent the index value of the feature x_z that needs dimensionality reduction in the input features, W_z represents the matrix of $m_z \times n_z$, b_z represents the deviation term of the vector, y_z is the vector obtained by dimensionality reduction, and $\mathscr{F}(\cdot)$ represents nonlinearity activation function.

3. Employment Model for College Students Based on Deep Learning and Machine Learning

This paper draws on the research results of Zhu Wei's flight delay prediction algorithm combined with convolution processing [18], hoping to improve the recommendation quality with the help of the high-level feature learning ability of convolution processing. The hybrid convolutional neural network model is selected, and the transformed features are used as input to train the hybrid convolutional neural network model through model stacking integration to generate recommended results.

3.1. Model Structure Design. The composition of the hybrid convolutional neural network model is mainly divided into a multichannel hybrid convolution submodel and a convolution and local connection hybrid submodel. The overall model structure diagram is shown in Figure 5.

Among them, Σ is represented as a data combination and splicing operation, which is used to combine the processing and generation results of multichannel convolution and is used for subsequent convolution and local connection mixing operations. The loss function used in the training process of the hybrid convolutional neural network is the cross-entropy loss function, which is used to determine how close the current actual output is to the expected output. Taking the binary classification task as an example, the mathematical expression of the cross-entropy loss function is shown in formula (5). Among them, y represents the true category label of the sample and p represents the predicted probability of the sample.

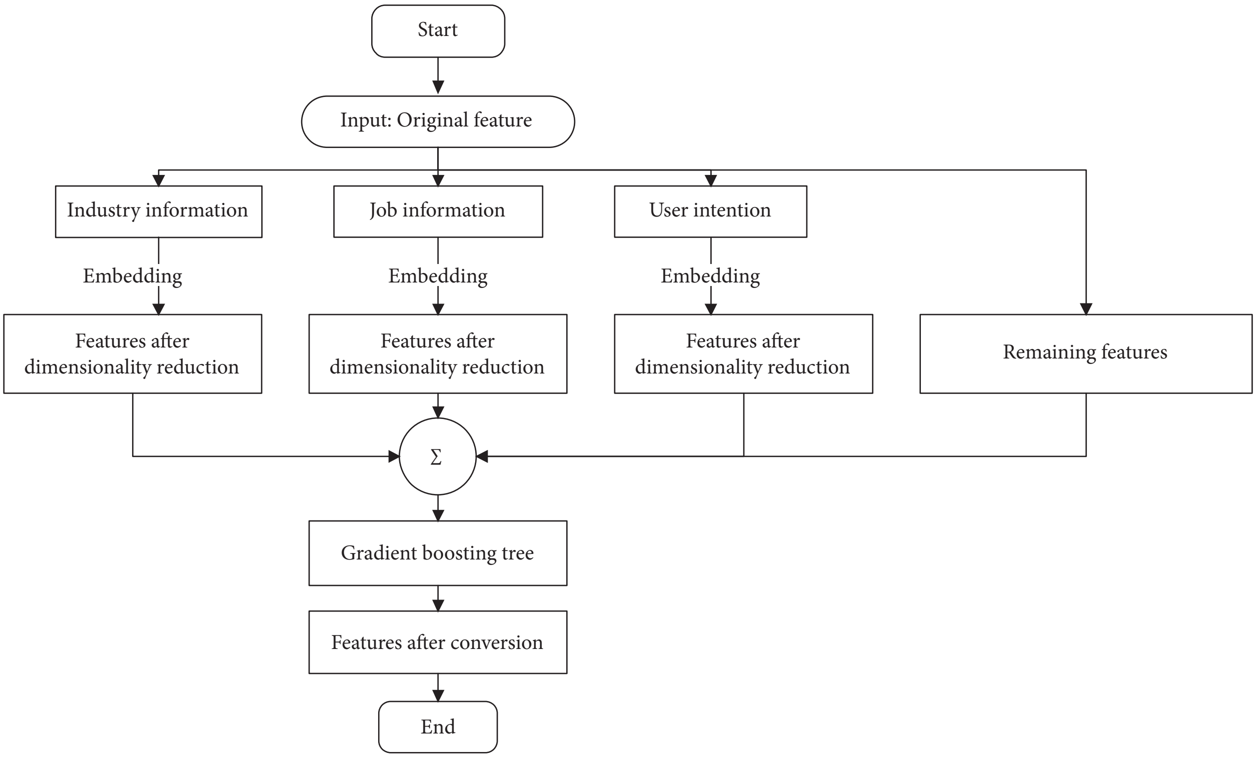

Figure 4: Feature extraction model.

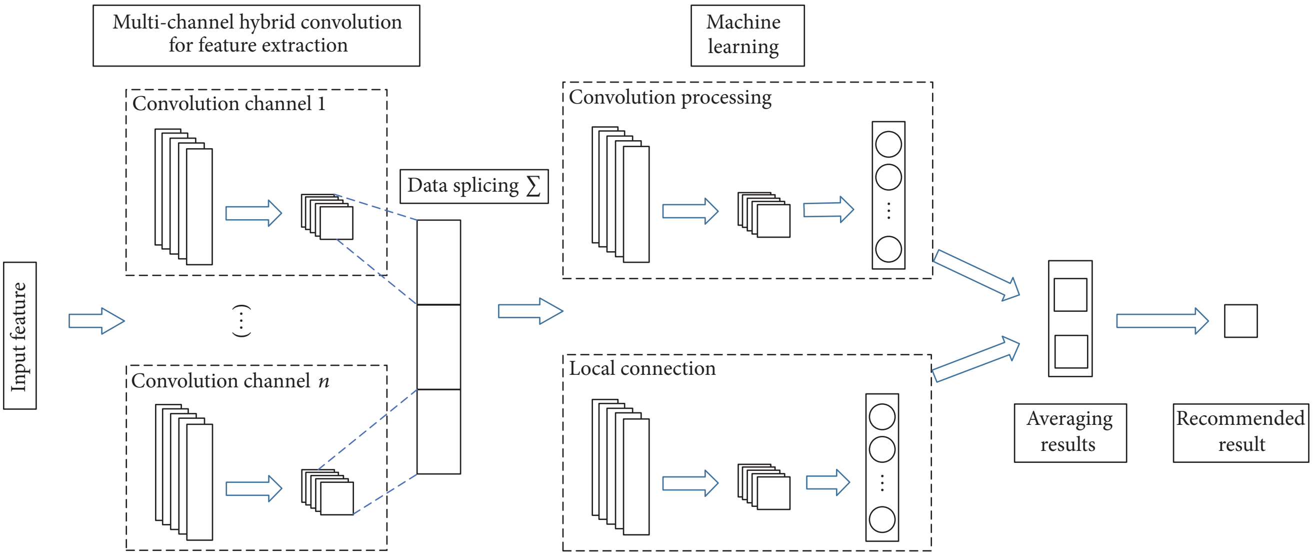

Figure 5: Model diagram of hybrid convolutional neural network.

$$\mathscr{L} = -\sum_{n} (y \log p + (1-y)\log(1-p)). \tag{5}$$

The description of the training algorithm of the hybrid convolutional neural network model is shown in Algorithm 1:

3.2. Model Training Optimization. The abovementioned hybrid convolutional neural network model is the benchmark model structure of the recommendation algorithm in this article, which can be used to recommend employment for college students, but it is found that there is still room for improvement in the process of model training.

3.2.1. Activation Function Optimization. Currently, commonly used nonlinear activation functions mainly include three functions: Sigmoid, Tanh, and ReLU. In the convolutional neural network model, the activation function of

```
Input: input feature D
Output: hybrid convolutional neural network model
Initialization parameters: the number of iterations $t$, the number of convolution channels $n$, the learning rate $\ell$ of the Adam
    algorithm, and the hyperparameters $\beta_1, \beta_2$, and $\varepsilon$
for $i = 1$ to $t$ do
    for $j = 1$ to $n$ do//$n$ represents the number of convolution channels
        Calculate the extracted features of the j-th CNN channel model, and the output is $F^{\omega}$
    end for
Integrate the extracted features of multichannel convolution $\{F^{(1)}, F^{(2)}, \ldots, F^{(n)}\}$, get F
Input the F input convolution and local connection hybrid processing model to obtain the recommended prediction probabilities
    $p_{CNN}$ and $p_{LC}$, respectively
Averaging $p_{CNN}$ and $p_{LC}$ to get the current prediction result
Calculate the error $\ell^i$ between the predicted result and the actual recommended result according to formula (4)
Calculate the reciprocal of model parameters W and $b$ according to back propagation and chain rule
```

Algorithm 1: Description of hybrid convolutional neural network training algorithm.

ReLU is optimized, and the main function is to alleviate the defect of network convergence caused by the death of neurons with input signal $x \leq 0$. The activation optimization used in this paper is the deformation optimization of ReLU to realize ELU. When dealing with negative input, the exponential term calculation method is used to alleviate the death problem caused by neuron death [19]. The specific algorithm expression of ELU is as follows:

$$ELU(x) = \begin{cases} x, & x > 0, \\ \alpha(\exp(x) - 1), & x \leq 0. \end{cases} \tag{6}$$

Using the activation function method of ELU for optimization processing can better obtain the representation information of the features, and compared with other ReLU improvements, ELU is more robust to input changes or noise.

3.2.2. Pooling Strategy Optimization. In the hybrid convolutional neural network benchmark model in this article, the pooling strategy adopted by the convolutional hierarchical model is the maximum pooling strategy of uniform size, which is a single pooling strategy. In this paper, hybrid pooling is used to make up for the shortcomings of feature loss caused by single pooling and to alleviate the model loss caused by feature loss. Integrating the advantages of the maximum pooling strategy and the intermediate value pooling strategy, on the basis of retaining the most salient feature representation in the pooling area, the pooling process of the hybrid convolutional neural network model is optimized through the intermediate value pooling strategy. The specific hybrid pooling integration implementation structure diagram is as shown in Figure 6. $\overline{y}$ represents the averaging process and $\sum$ represents the combined process, used to splice the maximum pooling result and the intermediate value pooling result, as the output of the mixed pooling operation.

3.2.3. Loss Function Improvement. The model training loss function used in this paper is the cross-entropy loss function, and the mathematical expression is shown in (4), which is used to determine the closeness of the current actual output to the expected output. After analyzing the prediction result data, He et al. put forward the drawback of "misleading loss" for the situation that the number of samples that are easy to determine is too large [19]. If the number of samples that are easy to determine is too large, it may conceal the effects of other types of samples and make a major contribution to the decrease of the loss value, which may dominate the overall gradient update direction of the model, leading to poor model convergence results. In response to this problem, He et al. put forward the idea that samples are difficult and easy to distinguish, by improving the loss function to reduce the loss contribution of easy-to-classify samples and alleviate the problem of loss misleading. This paper draws on the idea of difficult-to-differentiation of the algorithm and uses the threshold to distinguish the classification difficulty of samples, so that the model pays more attention to the learning of difficult-to-differentiate samples [20]. In order to better classify sample categories, penalty weights are introduced to deal with the problem of imbalanced sample categories. The improved loss function expression is shown in the following equation:

$$\mathscr{L} = -\sum_{n} \left(\beta_{+}\theta(p - m)y \log p + \beta\_\theta(1 - p - m)(1 - y)\log(1 - p)\right). \tag{7}$$

Y is the true classification category of the sample, p is the category prediction rate of the sample, β_+ and β_- represent the penalty weights of positive and negative samples, respectively, m represents the set training

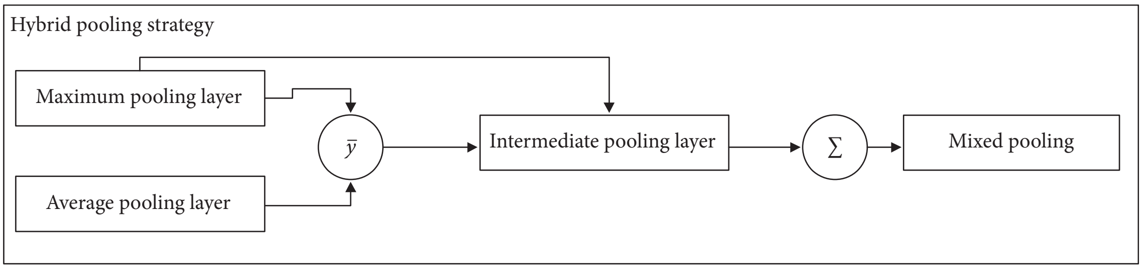

Figure 6: Implementation diagram of hybrid pooling strategy.

threshold, and $\theta(\bullet)$ represents the processing function for comparing the predicted probability with the threshold. The expression is shown in as follows:

$$\theta(x) = \begin{cases} 0, & x > 0, \\ 1, & x \leq 0. \end{cases} \tag{8}$$

4. Simulation Experiment

4.1. Recommended Quality Evaluation Standards. In the recommendation field, the commonly used recommendation quality evaluation standards include Precision, Recall, $F1$-Score, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). Among them, MAE and RMSE are generally applied to the recommendation method of the scoring type, and the quality of the recommendation is judged by calculating the error between the predicted recommendation score and the actual score. In this paper, recall rate and $F1$-Score are selected to evaluate the quality of algorithm recommendation. The calculation formula of recall rate is shown in formula (9), and the calculation formula of $F1$-Score is shown in formula (10).

$$\text{Recall} = \frac{TR}{TR + FNR}, \tag{9}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

TR indicates the number of positions that are correctly recommended to college students who are really interested, FNR indicates the number of positions that are truly interested in incorrectly predicted as nonrecommended categories, and FR indicates that positions that are not of interest to college students incorrectly predicted as recommended categories.

4.2. Experimental Data. The experimental data set used in this article is collected from the employment guidance platform for college students. The data set used in the experiment mainly contains information about college students themselves (gender, grades, interests, and so on), job requirements information, and user-post interaction behavior information. Since this article converts the recommendation question into a “recommendation/nonrecommendation” two-category question, the collected data are divided into two types such as positive samples and negative samples based on behavioral information. The sample is marked as a negative sample, and the sample that has undergone job browsing, job collection, and job application is marked as a positive sample. The total number of job seekers is 7547, the total number of jobs is 15023, the number of recommended positive samples is 9870, and the number of recommended negative samples is 5153.

4.3. Experimental Results and Analysis. This paper divides the collected employment data set of college students in proportion, divides 90% of the data set into the algorithm training data set, and uses the remaining 10% as the algorithm test data set. The experiment will verify the model optimization strategy and the performance comparison between the algorithm in this paper and the existing recommended algorithm. The experimental results will be analyzed, and the experimental conclusions will be drawn.

4.3.1. Activation Function Optimization Verification. The benchmark model uses ReLU as the activation function. Aiming at the dead zone problem caused by the ReLU activation function on negative inputs, this paper proposes an ELU optimization method based on ReLU and uses exponential terms to alleviate the neuron suppression phenomenon of negative inputs. This article will compare ReLU (benchmark model) and ELU on the basis of the recommended length of the post of 70 (regardless of the changes in the learning rate in the model, the loss function starts to stabilize when the number of iterations is 70). Among them, the recommended index for the benchmark model using ReLU is marked as X, and the recommended index for other activation methods is marked as Y. All models adopt the maximum single pooling strategy. The results of the comparative experiment are shown in Table 1.

From the comparison of the experimental results in Table 1, the recommended recall rate and $F1$-Score obtained by all models with optimized activation methods are better than ReLU, which proves the necessity of alleviating the neuron suppression problem that exists during model training. Optimizing the defect of the zero-value output of the negative input can effectively help the prediction of the model recommendation.

4.3.2. Pooling Strategy Optimization Verification. This section mainly conducts experimental comparisons for different pooling strategies. All pooling comparison

Table 1: Recommendation quality evaluation under different activation function modes.

Activation function type		Recommended quality evaluation index			
				Recall	F1-Score
ReLU	X	0.7305	—		
ELU (alpha = 0.01)	Y	0.7356	+1.88%		

Table 2: Comparison of experimental results of different pooling strategies.

Pooling strategy	Recommended quality evaluation index	
	Recall	F1-Score
Maximum single pooling strategy	0.7964	0.7356
Average single pooling strategy	0.7854	0.7334
A single pooling strategy for the median value	0.7915	0.7341
Maximum-average hybrid pooling strategy	0.7972	0.7363
Average-median mixed pooling strategy	0.8054	0.7376
Maximum-medium mixed pooling strategy	0.8104	0.7414
Max-median-mean hybrid pooling strategy	0.8031	0.7368

experiments are based on the used ELU activation function model. Based on the maximum pooling, average pooling, and intermediate value pooling, this article mainly divides the pooling strategy into two methods: single pooling strategy and mixed pooling strategy. The specific pooling strategy is divided into the following seven kinds:

(1) Maximum single pooling strategy
(2) Average single pooling strategy
(3) A single pooling strategy for the median value
(4) Maximum-average hybrid pooling strategy
(5) Average-median mixed pooling strategy
(6) Maximum-medium mixed pooling strategy
(7) Max-median-mean hybrid pooling strategy

The verification experiment will compare the Recall rate and $F1$-Score of the above seven pooling strategies based on the recommended length of the post of 70. The specific experimental results are compared as shown in Table 2.

From the comparison of the experimental results in Table 2, it can be concluded that on the whole, the recall rate and $F1$-Score of the hybrid pooling strategy are higher than those of the single pooling strategy. The recall rate obtained by the max pooling strategy is higher than that of average pooling and median pooling, and the recall rate of median pooling is also higher than that of average pooling, which proves that maximum pooling and median pooling are relative to average pooling. It can have a better extraction effect on feature information. By comparing the $F1$-Score of each pooling strategy, it can be obtained that the $F1$-Score of the maximum-median mixed pooling is higher than other pooling strategies. Therefore, this paper adopts the maximum-median hybrid pooling strategy that performs better in terms of performance and recommendation quality as the model pooling strategy.

4.3.3. Loss Function Improvement Verification. In this paper, the cross-entropy loss function is improved, the classification difficulty of samples is distinguished by the way of threshold setting, and the learning of difficult to distinguish samples is paid more attention to improve the quality of recommendation. This section compares the recall rate and $F1$-Score of the algorithm before and after the loss function improvement to verify the feasibility of the loss function improvement. The comparison experiment is based on the ELU activation method and the maximum-median hybrid pooling strategy. The threshold is set to 0.65. The specific experimental results are shown in Figures 7and 8.

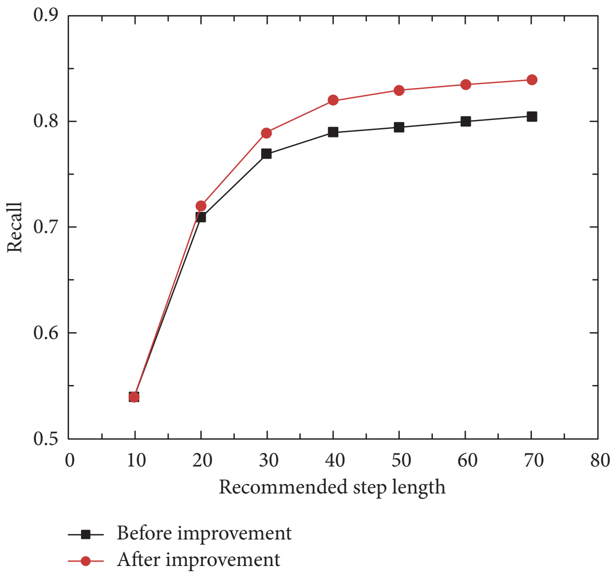

Figure 7: Recall experiment comparison chart before and after improvement.

The optimal values of recall rate and $F1$-Score before loss function improvement are 0.8104 and 0.7414. After using the improved loss function, the optimal values of recall rate and $F1$-Score are 0.8198 and 0.7432, which are increased by 1.159% and 0.243%, respectively. And the corresponding algorithm training time is reduced by about 9.6%. The comparison results prove that the model using the improved loss function can pay more attention to the learning of indistinguishable samples, which is beneficial to improve the quality of recommendations, and verifies the effectiveness of the improved loss function.

4.3.4. Algorithm Comparison Results. The selected comparative recommendation algorithms include user-based collaborative filtering recommendation algorithm (UserCF), content-based recommendation algorithm (CBF), probability-based matrix factorization (PMF), and four-layer deep neural network (DNN). The specific experimental comparison is shown in Figures 9 and 10.

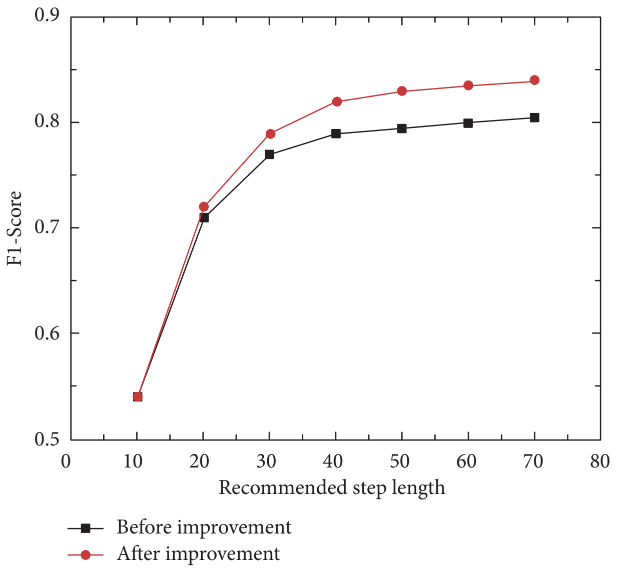

Figure 8: *F*1-Score experiment comparison chart before and after improvement.

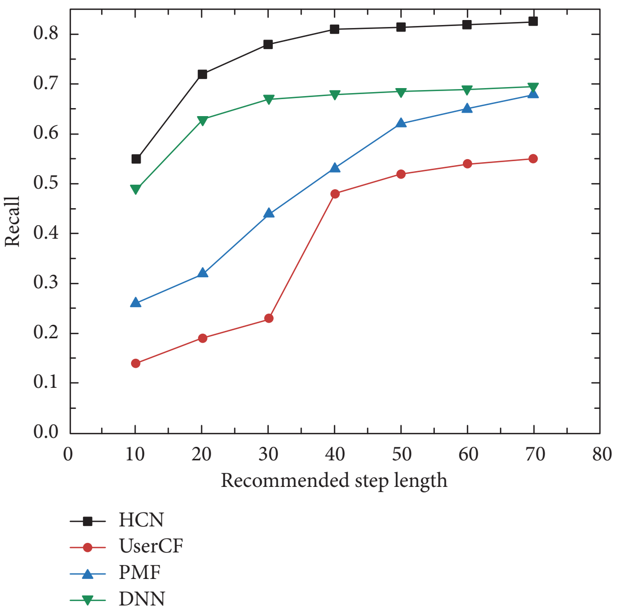

Figure 9: Experimental comparison of recall rates of various algorithms.

In a stand-alone environment, the recommendation time based on the test set, the algorithm in this paper, takes 36.33 seconds, and the DNN takes 16.67 seconds due to less calculation parameters, but the recommendation effect is not as good as the algorithm in this paper, and the recall rate of the algorithm in this paper increases by nearly 15%. For the calculation of similarity between a large number of college students and their positions, the recommended prediction time consumed is hour-level; for space occupation, the space consumption of the algorithm and DNN in this paper is mainly based on the network-level hyperparameters set by the model, while other algorithms require complementing the data matrix for similarity calculation depending on the number of college students and jobs in the school. The larger the corresponding data scale, the more space it will take up. Therefore, the algorithm in this paper has a certain improvement in the time and space consumption of prediction and can optimize the recommended prediction.

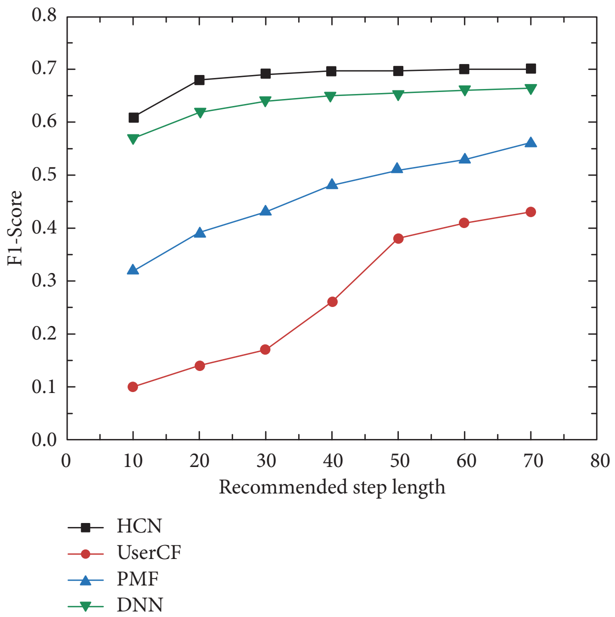

Figure 10: *F*1-Score experimental comparison of each algorithm.

From the experimental results in Figures 9 and 10, it can be seen that the algorithm in this paper has achieved better results in recall rate and *F*1-Score compared with the commonly used recommendation algorithms, which proves that the gradient boosting tree model and the feasibility of combining convolutional neural network models. With the help of the gradient boosting tree model for feature conversion and rich overall feature abstraction through multichannel convolution, combined with convolution processing and local connection processing, it can better carry out a deeper correlation between the information of college students and job information mining, and learning high-level abstract feature information can improve the recommendation quality of the model on human resources, which proves the effectiveness of the algorithm in this paper to improve the quality of human resources recommendation.

5. Conclusion

This paper establishes a college student employment data set and uses distributed data processing to extract data features through the analysis of college students' personal analysis and job information mining, so as to avoid the problem of "information flooding." At the same time, a hybrid convolutional neural network model is proposed for employment recommendation of college students. By improving the activation function, pooling strategy, and loss function in the algorithm, the quality of model prediction is greatly improved. Compared with the traditional recommendation algorithm, the recall rate and $F1$-Score of the algorithm in this paper are both ahead of traditional recommendation algorithms and ordinary convolutional neural networks. The recall rate of the algorithm in this paper is nearly 15% higher than that of the DNN model. The experiment is compared with the traditional commonly used recommendation algorithm, and the comparative analysis of the experimental results proves the effectiveness of the algorithm for the employment recommendation of college students. Multichannel convolution is used to enrich the overall feature abstraction, and combined with convolution processing and local connection processing, it can better conduct deeper correlation mining between college student information and job information, and learn high-level abstract feature information, thereby improving the recommendation quality of the model which proves the effectiveness of this algorithm in improving the quality of human resources recommendation.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Su, *Problems and Countermeasures in Career Planning of College Students*, Asia Pacific Education, no. 21, , pp. 267-268, Chennai, India, 2016.

[2] Q. Zhong, “Review and prospect of college graduates’ view on employment since the enrollment expansion%. review and prospect of college graduates’ view on employment since the enrollment expansion%,” *University Education Science*, vol. 43, no. 4, pp. 32–36, 2015.

[3] M. Gupta and P. Kumar, “Recommendation generation using personalized weight of meta-paths in heterogeneous information networks,” *European Journal of Operational Research*, vol. 284, no. 2, pp. 660–674, 2020.

[4] G. Nicolas, A. Tassadit, C. Olivier, and C. Fabien, “Gorthaur-EXP3: Bandit-based selection from a portfolio of recommendation algorithms balancing the accuracy-diversity dilemma,” *Information Sciences*, vol. 546, pp. 378–396, 2021.

[5] S. Kumar and M. D. Gupta, “c+GAN: complementary fashion item recommendation,” 2019, https://arxiv.org/abs/1906.05596.

[6] M. S. M. Sajjadi, G. Parascandolo, A. Mehrjou, and B. Schölkopf, “Tempered adversarial networks,” in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 4448–4456, Stockholm, Sweden, July 2018.

[7] R. Figueiredo, L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, “Struc2vec: learning node representations from structural identity,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 385–394, Halifax, Canada, August 2017.

[8] S. Hiriyannaiah, G. M. Siddesh, and K. G. Srinivasa, “Deep visual ensemble similarity (DVESM) approach for visually aware recommendation and search in smart community,” *Journal of King Saud University - Computer and Information Sciences*, pp. 1–13, 2020, Inpress.

[9] D. Koehn, S. Lessmann, and M. Schaal, “Predicting online shopping behaviour from clickstream data using deep learning,” *Expert Systems with Applications*, vol. 150, Article ID 113342, 2020.

[10] A. Da’u, N. Salim, I. Rabiu, and A. Osman, “Recommendation system exploiting aspect-based opinion mining with deep learning method,” *Information Sciences*, vol. 512, pp. 1279–1292, 2020.

[11] H. Sheil and O. Rana, “Classifying and recommending using gradient boosted machines and vector space models,” in *Proceedings of the UK Workshop on Computational Intelligence*, pp. 214–221, Springer, September 2017.

[12] E. Çano and M. Morisio, “Hybrid recommender systems: a systematic literature review,” *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, 2017.

[13] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youTube recommendations,” in *Proceedings of the ACM Conference on Recommender Systems*, pp. 191–198, ACM, Boston, MA, USA, September 2016.

[14] H. T. Cheng, L. Koc, J. Harmsen et al., “Wide & deep learning for recommender systems,” in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 7–10, ACM, Boston, MA, USA, September 2016.

[15] S. Okura, Y. Tagami, A. Tajima, and O. Shingo, “Embedding-based news recommendation for millions of users,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1933–1942, ACM, Halifax, Canada, August 2017.

[16] A. Karatzoglou, B. Hidasi, D. Tikk et al., “RecSys’16 workshop on deep learning for recommender systems (DLRS),” in *Proceedings of the ACM Conference on Recommender Systems*, pp. 415-416, ACM, Boston, MA, USA, September 2016.

[17] M. Balabanović and Y. Shoham, “Fab: content-based, collaborative recommendation,” *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[18] T. Mahara and S. Kant, “A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment,” *Procedia Computer Science*, vol. 89, pp. 450–456, 2016.

[19] D. Kim, C. Park, J. Oh, S. Lee, and F. Yu, “Convolutional matrix factorization for document x`context-aware recommendation,” in *Proceedings of the Acm Conference*, Boston, MA, USA, September 2016.

[20] T. Doleck, D. Lemay, R. Basnet, and B. Paul, “Predictive analytics in education: acomparison of deep learning frameworks,” *Education and Information Technologies*, vol. 25, no. 10, pp. 1951–1963, 2020.