ISSN: 2688-8645

Personalized Movie Recommendation based on Convolutional Neural Network

Haocheng Zhang[1], Yanxia Zhao[1,2,*], Xinyang Pan[1], Jiale Fu[1], Xuanyin Yao[1]

[1]College of International Business, Zhejiang Yuexiu University, Shaoxing, 312069, China

[2]School of Public Administration, Zhejiang Gongshang University, 310018, China

*Corresponding Author

Abstract

In pace with the development of the economy, the spiritual entertainment brought by movies is increasingly valued by people, and the problem of how to recommend the most suitable movie for users among the numerous movies also arises. Based on this, experiments were conducted using convolutional neural networks in the field of deep learning for movie recommendation. The convolutional neural network was trained using user information, movie information, and user movie rating data from the Douban Movie Network. In data preprocessing, instead of converting category fields to one hot encoding, they are converted into numbers and used as indexes for embedding matrices. After extracting features from the embedding layer, input these two features again in the fully connected layer, output a value, and then regress this value to the score to obtain MSE loss, and use the recall rate and NDCG indicator evaluation model. By comparing the accuracy with the other four recommendation algorithms on the same dataset, it was found that the movie recommendation algorithm obtained through the convolutional neural network approach studied in this paper has shown better results on the same evaluation indicators.

Keywords

Convolutional Neural Networks; Recommendation; Movies.

1. Introduction

The film industry in Chinese mainland (hereinafter referred to as China) is known for its good status as the world's largest film industrygrowth and attention. [1]. Movie recommendation is an extremely important field in today's digital entertainment industry. Big data brings the expansion of information quantity and qualitative change[2]. In order to meet the needs of consumers in choosing movies, there is a lot of room for improvement in the accuracy and user experience of traditional conventional recommendation methods.Traditional movie recommendation systems are increasingly falling short in the contemporary landscape of abundant information and evolving user behaviors [3]. However, with the changing times and technological advances, there is a lot of room for improvement in the accuracy and user experience of this traditional recommendation method.

In this paper, we adopt the convolutional neural network (CNN) in the field of deep learning to do the experimental study of movie recommendation, and train the convolutional neural network through the user information obtained from Douban Movie Website, the name of the movie, and the rating data of the user after watching the movie. Traditionally to realize collaborative filtering algorithms based on users and items, two problems need to be solved: how to calculate the similarity of the data in order to find people who have similar hobbies with you. Convolutional neural network is a deep learning algorithm that can effectively perform processing and analysis in the fields of image, speech, and text. The excellent performance of

ISSN: 2688-8645

CNN in image recognition has made it gradually become a key technology in the field of movie recommendation as well. Convolutional neural networks can automatically extract features from information such as users' historical viewing records and ratings as a way to recommend suitable movies for users. In this context, processing large-scale data for recommendation and improving the recommendation effect through convolutional neural network has been a hot research area, and movie recommendation is gradually taking the recommendation algorithm of deep learning technology as the mainstream direction. Among them, convolutional neural network, as an effective model induction algorithm, has been widely used in movie recommendation, and its excellent performance has been practiced by many movie recommendation commercial websites and gradually growing market share.

2. Related Work

Pasquale Lops [4] proposed a recommendation algorithm based on item content, which uses the content description of the item, abstracts it into the features of the item, uses the user's interest penchants, calculates the similarity between the user's predilections and the item's features, and then recommends the item preferred by the user.Zhang et al. [5] and Liu et al. [6] considered the timeliness of the item in the recommender system , which can greatly improve the accuracy of recommendation by merging the rating information of users and items with the timeliness of items. However, the above research is still based on collaborative filtering and the proposed algorithm still has large problems such as cold start and data sparsity.Lilinze et al [7] put forward a profound learning based recommendation algorithm, which utilizes neural network instead of user rating matrix to extract the features of the user and the movie, in order to enhance the suggestion accuracy and to solve the sparsity and cold start problems. The algorithm considers the accuracy and computation time of the recommendation results. On this basis Alexandros Karatzoglou et al [8] elaborated a deep learning based content recommendation and collaborative filtering recommendation method, which also provided ideas for the later application of various forms of deep learning models based on multilayer perceptron machines, convolutional neural networks, and recurrent neural networks in recommender systems.Pong Eksombatchai [9] proposed a base hand generative adversarial network for application in recommender systems. However, the algorithm in the number of users of the size of a large number of users, the likelihood matrix calculation of the user phase generated by the computational cost will be too high. Whereas, when the size of the number of users is small it leads to sparsity, which leads to an increase in the error of the similarity calculation.

The expandability of user-based collaborative filtering algorithms is related to the scale of the number of users; Qinyong Wang;James A. Esquivel uses deep deterministic strategy gradient deep reinforcement learning algorithm to build a more pinpoint and personalized movie endorsement system[10]. K.N. Asha;R. Rajkumar proposed a deep learning model using a deep neural network mechanism and a multiplicative Bi LSTM model. This scheme uses embedding, weight update, and preference learning processes to improve the performance of recommender systems[11]. Airen Sonu;Agrawal Jitendra uses the partition-weighted federated clustering of the movie recommendation system, and fine-tunes the parameters of users and movie neighborhoods by setting different values for the number of row clusters and the number of column clusters of federated clustering. [12], Mu Yongheng;Wu Yun proposed a personalized multimodal movie recommendation system based on multimodal data analysis and deep learning. A real MovieLens dataset was selected to test the effectiveness of our new recommendation algorithm. Combined with the input information, deep learning is used to mine the hidden features of movies and users, and a deep learning network algorithm model is constructed for training[13] .

ISSN: 2688-8645

The personalized recommendation system based on convolutional neural network proposed in this thesis is based on the information of the video platform users, movie information and the rating data of the users after watching the movie to train the model and then make the movie recommendation. It can find out the most matching movie with the user from the huge amount of movie information and recommend it to the user, which effectively reduces the time of the user to find a movie; for the platform, it can effectively reduce the waste of information resources and save the actual cost and improve the efficiency; for the society as a whole, it can reduce the waste of social resources in this era of information overload.

3. Related Recommendation Algorithms

3.1. Recommendation Algorithm

The movie recommendation algorithm is an algorithm that uses data mining and machine learning techniques to predict user preferences for movies and provide recommended movies. Based on the user's historical viewing records, ratings, search records, and other data, the algorithm will analyze, learn, and identify the user's preference features through various mathematical models, and recommend movies that the user may be interested in based on these features. The movie recommendation algorithm can provide users with movie recommendations that match their personal interests and tastes, improving their movie viewing experience.

3.2. User based Collaborative Filtering

User based collaborative filtering recommendation algorithm is a fundamental recommendation algorithm that has always held an important position in the field of recommendation algorithms. This algorithm is mainly user centered, with the basic idea that similar users are more likely to be interested in the same items.

This algorithm needs to first find a number of users in the similitude of the target user, and analyze the similarity between users through operations. Then, recommend these similar user sets that are already of interest to the target user but have not yet been purchased or have already been tried.

In practice, user based recommendation algorithms rely on a similarity matrix. When it is necessary to recommend products to the target user, the algorithm will use a similarity matrix to compare and find the group of users who are most similar to the target user, and use their purchasing preferences to predict the products that the target user may purchase. When calculating the similarity matrix, Pearson correlation coefficient and cosine similarity are usually used. Taking cosine similarity as an example, let u and v be two users, and N (u) and N (v) be the set of item names evaluated by user u and user v, respectively. The cosine similarity calculation formula between user u and user v is shown in (1),

$$\mathrm{sim}(u,v)=\frac{|N_u\cap N_v|}{\sqrt{|N_u||N_v|}} \tag{1}$$

If the data is in the form of user ratings for movies, the cosine value calculation formula is shown in (2),

$$\cos(\beta)=\frac{X_u\bullet X_v}{|X_u||X_v|} \tag{2}$$

Where Xu and Xv are the rating vectors for user u and user v, respectively.

It should be noted that user-based recommendation algorithms also have their limitations. For example, algorithms require sufficient user data to ensure the prepared recommendation results, and new users may encounter cold start issues after joining the system.

ISSN: 2688-8645

3.3. Collaborative Filtering based on Products

Collaborative filtering based on products is built on the similarity of items between users. This algorithm is used to calculate the similarity between products and find a set of products that are similar to the target product by calculating the similarity. Collaborative filtering based on products can also use various types of item similarity measurement methods to calculate the similarity between products. The similarity calculation method is the same as collaborative filtering. If you want to recommend movie A to user u, you need to determine the similarity between movie A and the movies that user u has watched, find the rating of the movie with the highest similarity between movie A and the movies that user u has watched, and set a threshold, If user u's predicted rating for movie A exceeds this threshold, it is like user u recommending movie A.

Finally, it should be noted that collaborative filtering based on products also has certain limitations. For example, when a product exceeds the activity range, the algorithm may not be able to predict a user's preferences well. In addition, data sparsity can also lead to poor algorithm performance, lacking common ratings or purchase history between items, making it difficult to accurately evaluate their similarity measures.

3.4. Collaborative Filtering of Matrix Factorization

Matrix decomposition is a commonly used collaborative filtering algorithm method that can decompose user rating records into user feature matrices and item feature matrices, and use this information to forecast the ratings of user unrated items, thereby achieving recommendation functionality. The collaborative filtering algorithm of matrix factorization has the following advantages:

- (1) It can solve the problem of data sparsity. When the user rating matrix is very sparse, matrix factorization can fill the gap in ratings by learning hidden layer features, improving recommendation performance.
- (2) Can adapt to new items and new users. When new items or users join, matrix factorization can adapt to changes by learning new item or user features, without the need for retraining the model.
- (3) Can handle multiple types of data. Matrix decomposition can not only handle rating data, but also other types of data, such as user behavior data, text data, etc.

However, matrix factorization also has some drawbacks:

- (1) Difficult to handle cold start issues. When new users or items do not have any rating data, matrix factorization cannot recommend items for them.
- (2) Easy to be disturbed by noisy data. When there are abnormal or incorrect rating data in the data, matrix factorization may be interfered with, affecting the recommendation effect.
- (3) The learning process involves a quantity of computing resources. Training matrix factorization models requires a significant large number of computational resources and time, especially when dealing with large-scale data.

3.5. Convolutional Neural Network Model

Convolutional neural networks are currently a fundamental model in the field of deep learning. Due to their excellent performance in image and audio recognition, they have been widely studied and used, and have also become the basic architecture for many subsequent extended models [14].

Convolutional neural networks are a sort of neural network model widely used in fields such as images, videos, and speech. The neuron model is shown in Figure 1, where X1, X2, and X3 are three input values, and w1, w2, and w3 are the weights corresponding to the three input values.

ISSN: 2688-8645

Sum and activation functions are the calculation functions of two neural networks, and frequently-used activation functions include sigmoid, tanh, and relu functions.

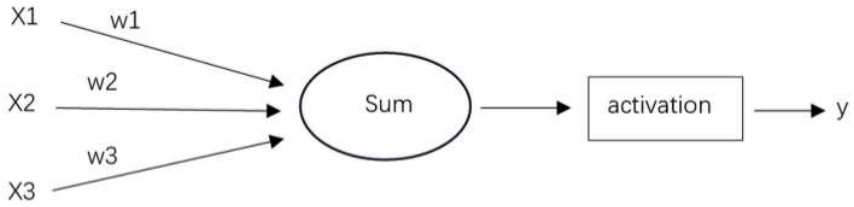

Figure 1. Neuron model

In terms of movie recommendation, CNN is used to model user viewing records and recommend new movies that meet their preferences based on their viewing preferences. The basic principle of CNN is to extract the features of input data using a series of convolutional and pooling layers, and then map these features to the output using a fully connected layer. In movie recommendation scenarios, input the user's historical viewing history or output a list of recommended movies.

Assuming a user u has watched n movies in the past, the feature vectors of these movies can be represented by x1, x2,..., xn. Firstly, the feature vectors of these movies are combined to form a matrix X=[x1, x2,..., xn], and the input to the CNN model is this matrix X. Next, extract the features of X using a series of convolutional and pooling layers. Specifically, extracting features from different levels using multiple convolutional and pooling layers, and then concatenating these features to form a feature vector f. Then, the feature vector f is input into a fully connected layer, which maps f to the score y of the recommended movie. The score y can be counted by using the following formula (3):

$$Y = b + w1 * f1 + w2 * f2 + \cdots + wn * fn \quad (3)$$

Among them, b is the bias term, w1, w2,..., wn are the weight parameters, f1, f2,..., fn are each element of the feature vector f. Finally, recommend movies based on the value of y, and movies with higher scores are more likely to be liked by users.

The structure of convolutional neural networks is divided into feature extraction layer and feature mapping layer. In the first layer, the input of neurons is combined with the input value of the previous layer, and then the features in the previous layer are extracted. As long as this feature is extracted, the relationship between this feature and its layer is determined.

The other layer is the feature mapping layer used to calculate feature values, where the weights of neurons on this layer are equal, and the activation function is done using the sigmoid function. The weights of the same neuron are shared by neurons located on the same map, reducing parameters and reducing the resolution of features.

4. Data Statistics and Visualization Analysis

4.1. Dataset Introduction

The dataset used in this study is from a publicly available dataset on the CSDN blog. The following table shows an example of the data content of the three tables used in this study. The movies table stores a total of 140000 pieces of movie information, the ratings table stores 1.04 million pieces of user comment data, and the users table stores a total of 5200 pieces of user data.

ISSN: 2688-8645

Table 1. Movies Table

Movieid	Title	Genres
26670818	Qingding Hezhou	Plot | Love
25815002	I'm not Bruce Lee	Action | Love
26392287	Manhattan Chinese Girl	Plot
26695995	Green haired water monster	Love
26392292	For the sake of our motherland	Plot | History

Table 2. Ratings Table

RatingID	UserID	MovieID	Ratings	timestamps	time
1359352573	1	5113101	2	2018/9/5 19:42	2018
1598245094	1	5113101	1	2019/7/9 14:52	2019
311937819	1	3718526	3	2010/11/5 22:15	2010
457663846	1	3718526	4	2011/11/14 22:31	2011
1359352573	1	5113101	2	2018/9/5 19:42	2018

Table 3. Users Table

UserID	Gender	Age	JobID
1	F	1	10
2	M	56	16
3	M	25	15
4	M	45	7

Table 4. Description of Each Field

field	meaning
Movieid	The unique ID of the movie
Title	The unique name of the movie
Genres	Keywords or genres of movies
Time	Film release year
Age	User age
Userid	User's Id
Ratings	User ratings of movies
Timestamp	Comment time
JobID	User occupation
Gender	User Gender
RatingID	Comment ID

ISSN: 2688-8645

Table 5. Statistics of Data Volume for Each Table

Table	Movies	Ratings	Users
Count	140000	1040000	5200

4.2. Trend and Total Number of Film Genres Over Time in Recent Years

There are 3 datasets used in this experiment, namely rating data, user data, and movie data. After reading in the three datasets, merge them together using the movie ID. Extract the year from the movie title and calculate the number of reviews and average rating for each movie. Finally merged into a new dataset.

Extract a unique movie type from the "Genres" column of the new dataset, create a new column for each type, and fill the column based on whether the movie belongs to that type. Fill the column with 1 for movies belonging to that type and 0 for movies not belonging to that type. Then add the values of the same type in the column to obtain the total number of movies of that type. Then, group the movies by year and type, and count the number of movies of each type in each year. Finally, use a horizontal bar chart to display the top 10 movie types with the most movies, as shown in Figure 2.

After calculating the number of each category, group the dataset by year and calculate the total number of movies of each type in each year. The resulting data box displays the total number of movies for each type of movie in each year. Finally, use Matplotlib to create a line chart that displays the trend of each type over time. The X-axis represents the years from 2000 to 2022, and the Y-axis represents the number of movies of each type.

As shown in Figure 3, the most common types of movies are plot movies, followed by movies with comedy. These two types, along with action and horror movies, account for approximately 50% of the total number of movies.

The film industry has shown a growth trend in all genres since 2005, especially with a rapid increase in the number of films. Among them, plot, comedy, and sports genres have grown the fastest, and to this day, these genres are still the ones with the highest number of shots. The number of different types of movies in the graph shows a roughly bimodal trend over time, with the first peak in 2015 and the second peak around 2018.

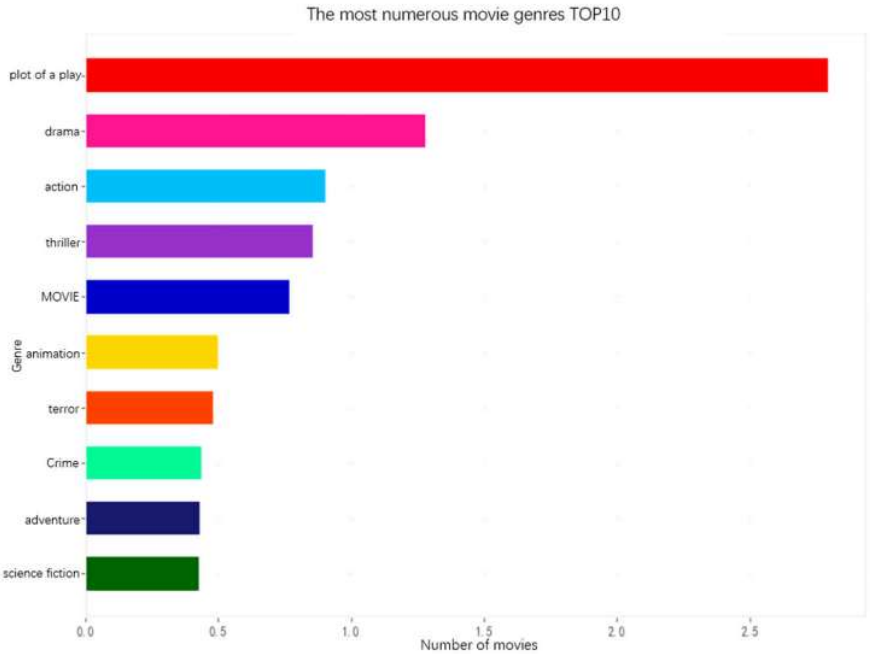

Figure 2. The movie types with the highest number of types

ISSN: 2688-8645

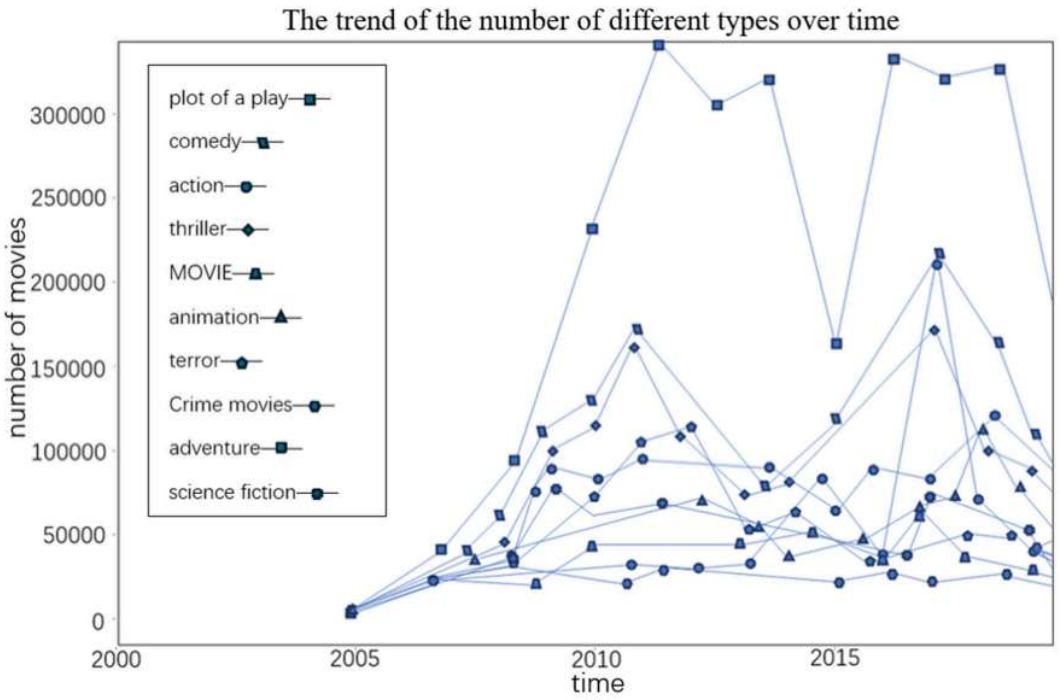

Figure 3. The movie types with the highest number of types

4.3. Score Distribution

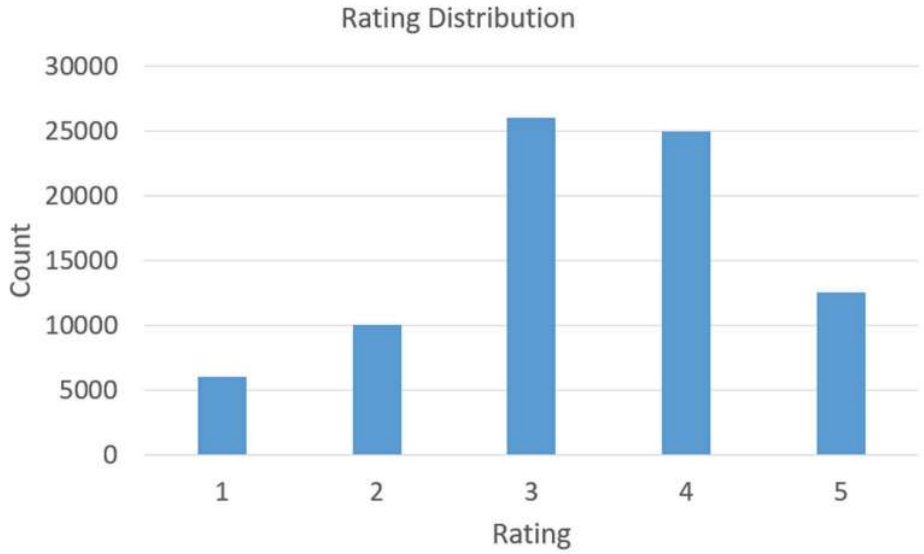

Figure 4. Statistical Chart of Score Distribution

The rating is the most reflective of the user's real viewing experience. In this experiment, the data was collected on a merged new dataset and sorted in descending order using value_ The counts() function evaluates the score and displays the results in the form of a bar chart. Next, add a title to the chart using the title() function, xlabel(), and ylabel() functions to set the labels on the horizontal and vertical axes, where the horizontal axis represents the score and the vertical axis represents the quantity calculated by that score. In order to improve the efficiency of the experiment, only 80000 comment data were randomly selected from the dataset for visual analysis of rating data. As shown in Figure 4, out of 80000 rating data, the highest rating

ISSN: 2688-8645

given by the audience was 3 out of 5, totaling approximately 26000, followed by 4 out of 25000, with nearly 15000 out of 5.0. It can be seen that the ratings given by the audience are generally high, with scores of 3.0 or above accounting for more than half of the total, and low ratings being relatively low. But judging from the high and low ratings, the quality of the movies made at this stage is still good.

4.4. The Relationship and Distribution between Average Scores and Years

This experiment extracts two columns of required movie duration and average rating from the dataset, and stores them as two new variables. Next, create a graph with a size of (9,6) pixels using the matplotlib library, and use the scatter() method to draw a scatter plot. The horizontal axis represents the year (i.e. movie duration), the vertical axis represents the average rating, and the color of the set points is black. The experimental results are shown in Figure 5. After obtaining the relationship between score distribution and time, a histogram is generated using the distplot function in the Seaborn visualization library to display the distribution of average scores in the dataset. This histogram divides the data into 10 bins and sets the image size to 10 times 5. A histogram is used to display the distribution of continuous variables. In this experiment, the x-axis is divided into several range intervals (i.e. bin), and the y-axis represents the frequency or density of values within that interval, thereby displaying the overall distribution of the variable. The seaborne. distplot() function can also plot fitting curves and kernel density estimates on histograms to better describe data distribution.

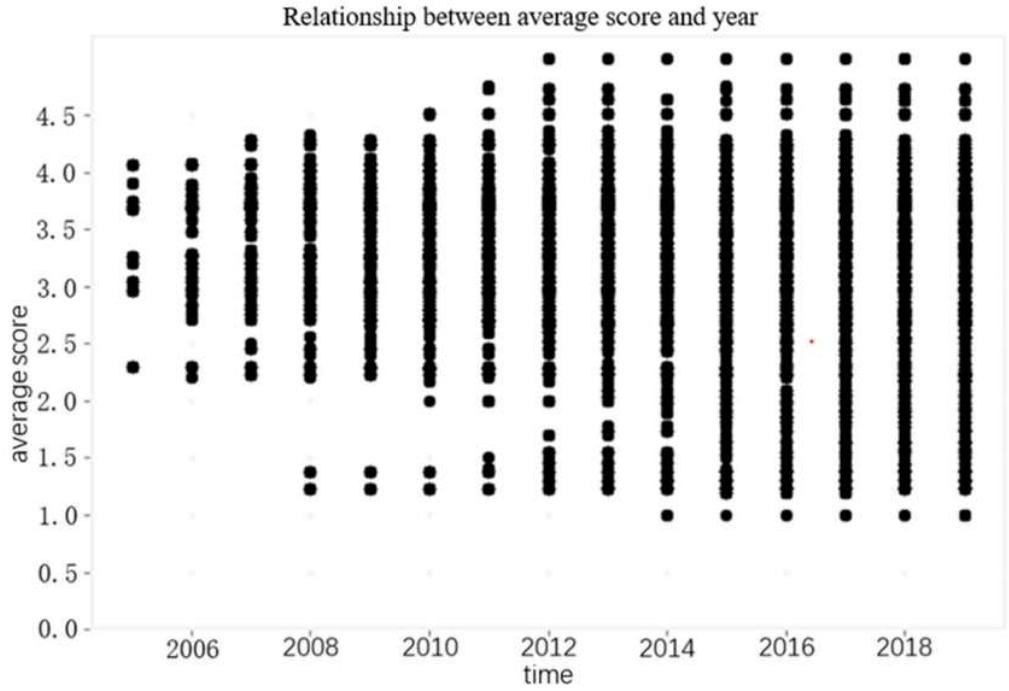

Figure 5. Relationship between average score and year

The experimental results are shown in Figure 6, where the average rating of the movie is roughly in the range of 2.0-4.5, with the highest being in the range of 3.0-4.5. From the graph of changes in ratings over years, it can be seen that as years increase, the number of movies that receive negative reviews decreases. After 2010, there are still many movies with an average rating between 1.0 and 2.5. With the development of the times and film technology, the quality of the movies produced is also increasing, and the number of positive reviews received is also increasing.

ISSN: 2688-8645

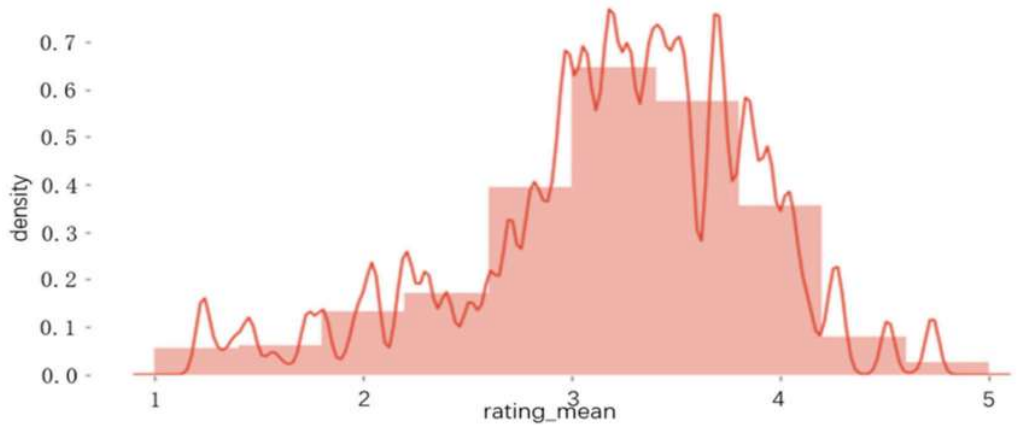

Figure 6. Relationship between average score and year

4.5. Number of Comments Per Year

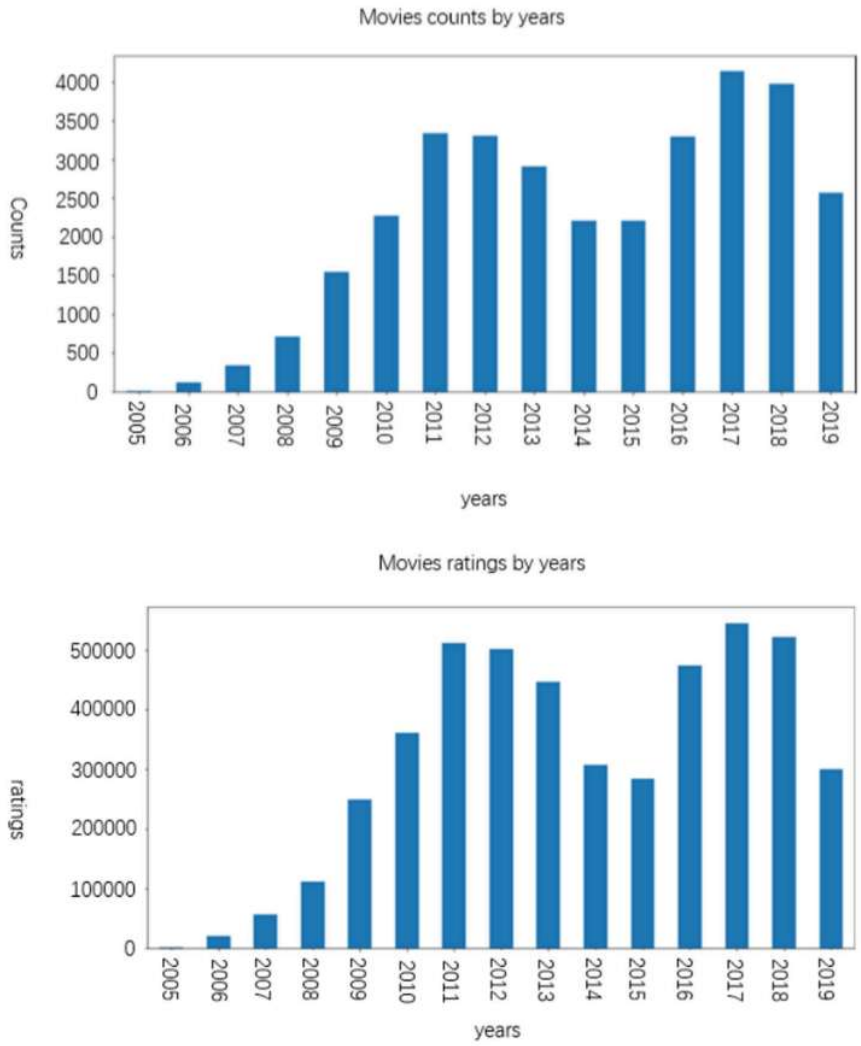

Figure 7. Number of reviews by year

The experimental result in Figure 7 visualizes the number of movies per year and the number of movies being reviewed each year using a bar chart. Firstly, the data is grouped by year using

ISSN: 2688-8645

the groupby function, and then the number of movies and reviews is calculated using the count and sum functions, and two subgraphs are drawn. In the experiment, the canvas size was set to 16x16, and two subgraphs were created to display the number of movies and movie reviews, respectively. The X-axis represents the year, and the Y-axis represents the total number of reviews for that year.

From the experimental results, it can be seen that the number of reviews for movies in 2005 was almost zero, which is closely related to the historical background of that time. At that time, China's internet technology was not yet developed, and people had fewer opportunities to watch movies online. At the same time, the number of movies produced by the producers was also limited. Everyone was solving the problem of food and clothing, let alone going to watch movies to relax. The peak of movie reviews occurred in 2017, and the number of movies has remained above 2500 per year since 2016. In 2019, the number was relatively small due to the impact of the pandemic.

5. Discussion and Analysis of Experimental Results

5.1. Model Training Results

When training the model, this article divides the dataset into training and testing sets in a 4:1 ratio, and then randomly selects 20% of the data from the training set as the validation set. After training the model, the training loss and testing loss can be obtained, as shown in Figures 8 and 9. The loss value of the training set reached around 10 at the beginning of the model's training iteration. As the number of model iterations increased, the loss value gradually decreased. The loss value reached a convergence state when the model was iterated 250 times, and the final loss value of the training set was around 1.05. The loss value of the test set during the initial training iteration of the model is around 1.5. As the number of model iterations increases, the loss value fluctuates up and down, but overall shows a decreasing trend. The loss value reaches convergence state when the model is iterated 400 times. The loss value of the test set is also similar to that of the training set, and the final result is around 1.05, with the lowest iteration being around 0.9. If the number of iterations is increased, the error of the experiment may also decrease. The specific results are shown in the following figure:

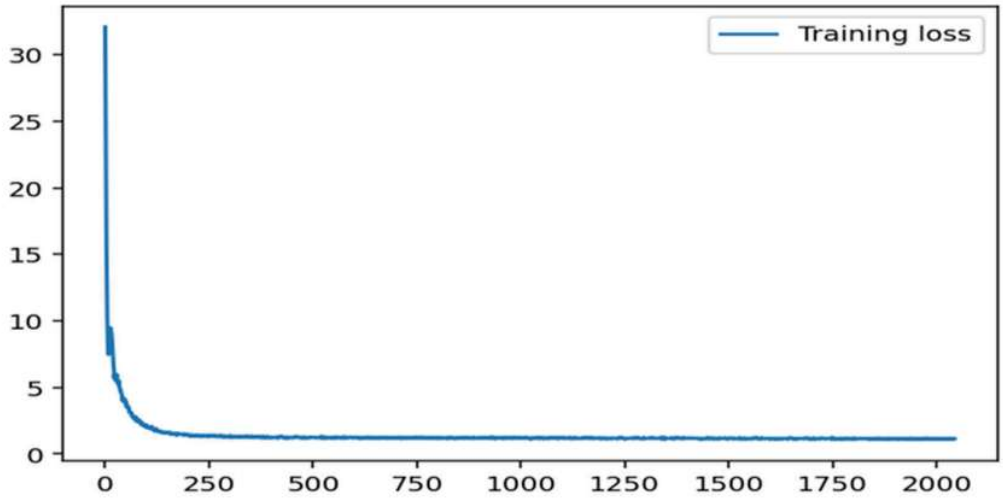

Figure 8. Training set loss

ISSN: 2688-8645

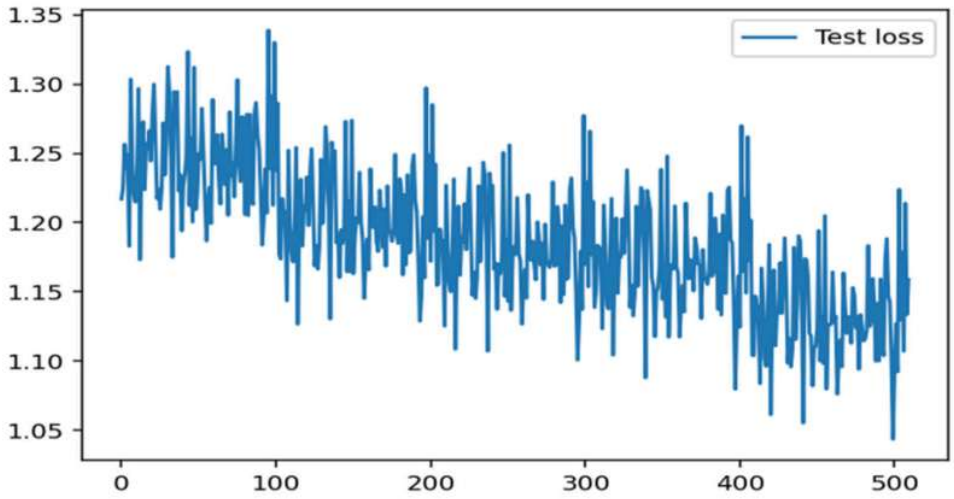

Figure 9. Test set loss

5.2. Movie Recommendation List

The result of model training is to obtain a recommendation list. Based on the model training results, this article mainly indexes movie feature similarity, movie rating distribution, and user similarity to obtain three movie recommendation lists.

This experiment first uses the API function provided by TensorFlow to get_ Tensor_ By_ Name, loaded in the already loaded graph model_ Retrieve the Tensor object with the specified name from the graph. Among them, the name of the tensor has been determined when defining the graph model. These tensors can be used for forward propagation of computational graph models to achieve specific functional tasks. Finally, the function returns these tensors (user account, occupation, gender, age, gender, movie title, category, etc.) as tuples.

After obtaining tuples composed of these tensors, predict the user's rating for a specific movie based on user information, as well as the category and title of the movie. Accept user_ ID_ Val and movie_ ID_ Two parameters, val, are the IDs of the user and movie. Load a pre trained convolutional neural network model from the saved directory, obtain necessary tensors from the loaded model, and then input user and movie data for prediction. Predictive rating is returned as output.

The neural network model of this experiment has been trained and optimized through a large amount of historical data, and can accurately predict user ratings for movies based on input user and movie data. In the prediction process, forward propagation algorithm is used, which starts from the input layer and calculates the output of each layer layer layer by layer until the final prediction result is obtained, in order to achieve the prediction of the user's movie rating.

5.2.1. Recommendation based on Movie Feature Similarity

This experiment first loads the previously trained and saved model, and uses the movie in it_ The metrics variable is used to represent the movie feature vector and normalized. Next, based on the given movie ID, search for its corresponding index in movieid2idx and retrieve it from the movie_ Obtain the feature vector of the movie from the metrics variable, and then use cosine similarity to calculate the similarity score between the movie and all other movies. After calculation, the obtained array will be dimensionally reduced, and the np. random. choice() function will be used for random sampling based on the similarity score, and a set will be used to avoid recommending duplicate movies. The recommended results obtained from this experiment are listed in Table 6.

ISSN: 2688-8645

Table 6. Recommended list of movie feature similarity

The movie you watched is: [6389523 'Heaven and Earth are useless' and' Plot | Love ']
[6389523 'Heaven and earth are useless'' Plot | Love ']
[5062536 'Where She Gores'' Plot ']
[21340764 'Who Moved My Dreams'' Plot | Comedy | Love ']
[25921901 'Moving Painting' and 'Animation']

5.2.2. Recommendation List based on Movie Rating Distribution

This experiment also first loads the previously trained and saved model, then finds its corresponding embedding vector based on the user ID, calculates the similarity score between this vector and all movie embedding vectors, sorts all similarity scores, and selects the top k movies with the highest scores as the recommendation result. The selected movie ID, name, and category are output, and a list composed of these IDs is returned. Among them, 5 movies were randomly selected, not necessarily the top 5 movies with the highest scores. The results are shown in Table 7.

Table 7. Recommended List of Top Five Movies for Distribution of Related Movie Scores

The movie ID you watched is 234
[1938951 'Electric Light Firestone', 'Plot | Suspense | Horror | Crime']
[3221804 'Battle of Tiger and Mouse' 'Action | Crime']
[3212017 'Dancer' and 'Plot']
[5071955 'Broadway Musketeers'' Plot | Music ']
[5152995 'Massacre' and 'Terror']
{30270,32811,34320,86776,129773}

5.2.3. User Based Recommendation List

Table 8. List of users who like the same movie

People who enjoy watching movies [6389523 'Useless World' and 'Plot | Love'] are
[4091 'M' 25 17]
[1684 'M' 18 3]
[691 'M' 35 11]
[2757 'M' 25 2]
[3111 'M' 18 4]
[2542 'F' 45 13]
[2028 'M' 45 6]
[3048 'M' 18 2]
[4443 'M' 56 2]
[1395 'M' 25 2]
[1460 'F' 18 4]
[5200 'F' 35 15]
[1637 'M' 56 0]
[1454 'M' 18 4]
[2252 'F' 18 1]
[3153 'F' 25 4]
[4908 'M' 35 15]
[1630'M '45 17]
[4006 'M' 1 7]
[3595 'M' 25 0]

ISSN: 2688-8645

Similar to the previous two experiments, the one sample experiment also first loads the previously trained and saved model, and then calculates the similarity between the input movie and the user's favorite movie. It is achieved by multiplying the embedding of the input movie by the transpose embedded by the user. The obtained array is sorted in descending order and returns the index of the top k values with the highest similarity. Then output the name of the movie and the users who like it. Finally, randomly select five movies with high similarity and return their indexes as a set. The specific results are shown in Tables 8 and 9.

Table 9. Top 5 movie recommendation lists for user similarity

People who enjoy watching movies [6389523 'Useless World' and 'Plot | Love'] also enjoy watching them
[1301828 'Legend of War', 'Love | Western | Adventure']
[25911941 'Il barbiere di Siviglia' 'Music']
[3212017 'Dancer' and 'Plot']
[3192680 'Blonde Girl in Black Leather' 'Plot | Comedy | Thriller']
[5152995 'Massacre' and 'Terror']
{30270,34320,40872,57795,131163}

The results shown in the above table are hyperparameters num_ Epochs=5, batch_ Size=2048, dropout_ Keep=0.5, learning_ Rate=0.001, show_ Every_ N_ The recommended list obtained after training the model with batches=20.

5.3. Model Comparison

Recommendation algorithms have broad application value in practical applications, and the performance of different algorithms also varies. As indicators for evaluating algorithm performance, NDCG and Recall are widely used in recommendation algorithms. This article will introduce these two indicators and draw conclusions after analyzing the MSE, Recall, and NDCG indicators of the five recommendation algorithms: CNN, ENMF, BPR, FISM, and DIN.

5.3.1. MSE Indicators

MSE is an indicator used to measure prediction error and is typically used to evaluate the magnitude of prediction error. The calculation method is to sum the square of the difference between the predicted value and the actual value of the model and divide it by the number of samples. The smaller the MSE, the smaller the prediction error and the better the fitting effect of the model. MSE is commonly used in machine learning, especially in predicting continuous variables. The smaller the value of MSE, the better the model. The calculation formula is shown in (4),

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad (4)$$

Where represents real data and represents predicted data.

5.3.2. NDCG Indicators

The full name of NDCG is "Normalized Loss Accumulated Gain". NDCG is an indicator used to evaluate the quality of recommendations in a recommendation system, measuring the system's ability to rank items that users are truly interested in among multiple recommendation results. NDCG considers the ranking effect of recommendation results in a discounted manner. When evaluating, the system will provide a score for the first K items recommended, and then calculate the NDCG value based on the scores of these items. The closer the value of NDCG is to 1, the better the ranking effect of the recommendation results. The calculation formula is shown in (5),

ISSN: 2688-8645

$$NDCG_u = \frac{1}{|u|}\sum_{u\in U}\frac{DCG_u}{IDCG_u} \quad (5)$$

$$DCG_u = \sum_{k=1}^{n}\frac{\delta(\hat{\Im}_{u,N}\in\Im_u)}{log_2(k+1)} \quad (6)$$

$$IDCG_u = \sum_{k=1}^{|\Im_u|}\frac{1}{log_2(k+1)} \quad (7)$$

5.3.3. Recall Metrics

Recall is an important performance metric in classification problems, commonly referred to as recall, which refers to one of the precision and recall rates. Recall is defined as the percentage of movies recommended by the system that the user actually likes. The higher the Recall indicator, the better the system can recommend movies that users are interested in. The calculation formula is shown in (8),

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

TP represents the proportion of positive cases made by the model to all positive cases in the dataset, while FN represents the data that the model considers to be negative but actually positive.

5.3.4. Comparison of Algorithm Accuracy

This study employed various algorithms from different fields to perform movie recommendations on the same dataset and compared their accuracy. Among them, convolutional neural network algorithm is used for movie recommendation, while EnMF, BPR, FISM and other algorithms in the field of recommendation algorithms are used to seek more accurate recommendation results. After experimental testing, it was found that CNN algorithm and ENMF algorithm perform well in three indicators: mse, recall, and NDCG. They can more accurately predict user ratings for movies, and also better match user interests with movies, improving the recommendation quality of recommendation systems. The specific experimental results are shown in Table 10.

Table 10. Accuracy table of each recommendation algorithm

algorithm	Mse	Recall	Ndcg
CNN	0.9459	0.9338	0.9785
ENMF	2.9438	0.8473	0.8827
BPR	1.6036	0.8204	0.8730
FISM	41.2579	0.6187	0.5877
DIN	3.6411	0.7911	0.7231

Firstly, looking at the performance of MSE, it can be seen that the CNN algorithm has the smallest MSE value, the ENMF algorithm has a relatively small MSE ', and the FISM algorithm has the largest MSE value. This indicates that CNN algorithm and ENMF algorithm can more accurately predict user ratings for movies, while FISM algorithm has poor prediction performance.

Moving on to the Recall metric, it can be seen that the CNN algorithm has the highest Recall value, while the ENMF and DIN algorithms have good Recall values, while the FISM algorithm has the lowest Recall value. From this, it can be concluded that CNN algorithm, ENMF algorithm, and DIN algorithm can better recommend movies that users are truly interested in. Finally, looking at the NDCG index, the CNN algorithm has the highest NDCG value, the ENMF algorithm has the second highest NDCG value, while the BPR algorithm and FISM algorithm perform relatively poorly in NDCG values. This indicates that CNN algorithm and ENMF algorithm can

better rank the movies that users are truly interested in at the top, improving the recommendation quality of the recommendation system.

In summary, both CNN algorithm and ENMF algorithm have shown excellent performance indicators in recommendation systems, and can better recommend movies that users are truly interested in. However, overall, the CNN algorithm still performs better than other algorithms on the three-evaluation metrics used in this article, while the BPR algorithm and FISM algorithm perform relatively poorly, and their algorithms need to be improved to enhance recommendation performance.

6. Conclusion

In terms of movie recommendation, the most commonly used recommendation algorithm is based on collaborative filtering. However, in today's booming big data era, traditional movie recommendations based on collaborative filtering algorithms are no longer able to meet people's viewing needs. The convolutional neural network used in this article can effectively avoid the cold start problem caused by data sparsity in movie recommendation. We used Douban's movie dataset for recommendation tasks, which includes data preprocessing, exploratory data analysis, building movie and user features, constructing neural networks, and obtaining recommendation lists. We evaluated the model using MSE, NDCG, and Recall. After conducting movie recommendation using convolutional neural networks, in order to verify the feasibility of convolutional neural network algorithms in movie recommendation algorithms, this paper also conducted comparative experiments on the same dataset using commonly used ENMF, BPR, FISM, and DIN algorithms in the field of recommendation algorithms. Through the comparison of five models on the evaluation indicators commonly used in three recommendation algorithms, it was found that convolutional neural networks have better accuracy and can effectively avoid the problem of cold start.

Acknowledgments

Natural Science Foundation.

References

[1] Kai Ruo Soh; What happens behind the ‘transnational’ of transnational films? Examining China and South Korea’s co-productions [J]. Creative Industries Journal .2023(04):46-50.

[2] Qinyong Wang; James A. Esquivel. Film recommendation systems have gained widespread application in the information age [J]. The Frontiers of Society, Science and Technology. 2023 (01): 72-77.

[3] Chin Yi Chen; Jih Jeng Huang. Temporal-Guided Knowledge Graph-Enhanced Graph Convolutional Network for Personalized Movie Recommendation Systems [J]. Future Internet.2020(27):41-42.

[4] Pasquale Lops;;Dietmar Jannach;;Cataldo Musto;;Toine Bogers;;Marijn Koolen. Trends in content-based recommendation[J].User Modeling and User-Adapted Interaction,2019,29(2):239-249.

[5] Fuguo Zhang; Qihua Liu;;An Zeng.Timeliness in recommender systems[J].Expert Systems With Applications, 2017,85:270-278.

[6] Yue Liu; Fei Cai; Pengfei Ren; Zhizhou Gu.Item life cycle based collaborative filtering.[J].Journal of Intelligent and Fuzzy Systems,2019,36(3):2743-2755.

[7] Li Linze. Movie recommendation algorithm based on Deep Learning,2020,14.

[8] Shuai Zhang 0007; Lina Yao; Aixin Sun;Yi Tay. Deep Learning Based Recommender System: A Survey and New Perspectives.[J].ACM Comput. Surv.,2019,52(1):5:1-5:38.

[9] Rex Ying; Ruining He;;Kaifeng Chen;;Pong Eksombatchai; William L. Hamilton;;Jure Leskovec.Graph Convolutional Neural Networks for Web-Scale Recommender Systems,2018:974-983.

ISSN: 2688-8645

[10] Huali Pan; Jingbo Wang; Zhijun Zhang. A movie recommendation model combining time information and probability matrix factorisation [J]. International Journal of Embedded Systems, 2021, 14(3): 239-247.

[11] Li Linze. Movie recommendation algorithm based on Deep Learning,2020,14.

[12] Qinyong Wang; James A. Esquivel. Personalized Movie Recommendation System Based on DDPG: Application and Analysis of Reinforcement Learning in User Preferences [J]. The Frontiers of Society, Science and Technology, 2022, 30(19),60-64.

[13] K.N. Asha; R. Rajkumar. DCF-MLSTM: a deep security content-based filtering scheme using multiplicative BiLSTM for movie recommendation system [J]. International Journal of System of Systems Engineering.2020(03):483-493.

[14] Airen Sonu; Agrawal Jitendra. Movie Recommender System Using Parameter Tuning of User and Movie Neighbourhood via Co-Clustering [J]. Procedia Computer Science.2019(01).

[15] Mu Yongheng; Wu Yun. Multimodal Movie Recommendation System Using Deep Learning. Mathematics.