ISSN: 2688-9323

Research on Personalized Recommedation Algorithm based on Neural Network

Luyao Jiang, Xingwang Yang and Dong Liu

School of Economics and Management, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract

With the development of Internet technology, people's ability to create information has also increased, which has caused the explosion of information, and how to find useful and interesting information from massive amounts of information has become a problem.At present, there are mainly three methods for solving the information overload problem:search engine, portal website and recommendation system.Among the three methods, only the recommendation system does not require the provided information displayed by the user, which is more user-friendly.Although the recommendation system has been able to solve the problem initially, the recommendation algorithm still faces two major problems:data noise and data sparseness.In recent years, deep learning technology has been well applied in fields such as image recognition and natural language processing, and has achieved results far exceeding traditional machine learning.The research on the combination of deep learning and recommendation system has also made a big breakthrough, which has more advantages than traditional recommendation methods.In view of this, this paper combines generative adversarial neural networks with recommendations, and proposes a generative adversarial collaborative filtering algorithm based on a trainable activation function.In this paper,we use the classic Movielens data set to conduct a systematic experiment on the generation of an anti-cooperative filtering algorithm based on a trainable activation function,using a fully connected neural network structure with different structures,learning steps and types of activation functions.Finally,this paper conducted comparative and corresponding tests on differentmodels,using recall rate,accuracy rate and F1value as the quality evaluation standard of the recommended algorithm.The experimental results show that the proposed algorithm based on trainable activation function has certain feasibility and effectiveness.

Keywords

recommendation algorithm, neural network, deep learning.

1. Introduction

In the information-based Internet world, the amount of information is growing rapidly, and the speed of human processing information can not keep up with the speed of information birth, which leads to the problem of information overload. In the past period, people pursue timely satisfaction, expect novelty experience, but now, in the face of massive information push, people feel overwhelmed by it, it is difficult to quickly find interested information from a large number of information. People are eager to block the troubles caused by information explosions, and businesses need to know how to create more value on existing data. Today, the main ways to ease information overload are portals, search engines and referral systems. Portal and search engines need users to enter keywords or click operations to actively search for relevant information, so as to achieve the function of filtering information. And the recommendation system is based on people's historical interactive data, so as to infer and predict the information

ISSN: 2688-9323

that users pay attention to and present it in front of people, and do not need to filter the information displayed by users.

The recommendation system already has many applications in various fields, such as NetEase cloud music use recommendation system for personalized music recommendation; Taobao use recommendation system to achieve accurate product recommendation; Netflix use recommendation system to recommend movies similar to those they once liked; Google's Adsense also use recommendation system for personalized advertising. It can be seen that the recommendation system already involves all aspects of people's life, and can provide more interesting and useful information for people more conveniently and quickly.

According to extensive literature and Internet statistics, collaborative filtering algorithms are the most influential and used in industry and academia[1]. most of the current recommended algorithms in industry are collaborative filtering algorithms based on deep learning techniques, such as: DeepFM[2], proposed by google in 2016 in 2017 and ESMM model [3] proposed by alimama in 2018. The application of deep learning in recommendation field is more and more, and it is also the current mainstream research direction.

although the present recommendation model already has good performance, the recommendation algorithm still has two challenges, one is data sparsity, the other is data noise [4]. At present, most of the collaborative filtering recommendation algorithms based on deep learning are modeling the feature interaction between users and projects, while the processing of data itself is less research in mitigating data sparsity is still insufficient.

For individuals, the recommendation system is able to analyze the information of personal interest and provide effective information without showing to inform the system of personal preferences; for enterprises, to increase the exposure of their products and improve their performance, the use of a well-performing recommendation system is the best choice at present.

In this paper, we improve the cooperative filtering algorithm based on generative countermeasure network, let the activation function of neural network do adaptive training learning, so as to improve the learning ability based on generative countermeasure network and make it have stronger data fitting ability, thus improve the quality of generative network generative data. the proposed collaborative filtering algorithm based on generating adversarial networks uses noise data to mitigate data sparsity and makes project prediction recommendations based on the completed data (generating network generated data). On this basis, the following research work is carried out:

Transform and generate adversarial networks so that their activation functions can also participate in training to enhance model learning.

Analyze the effect of the number of final recommended items on the results.

According to the experimental results, the proposed adaptive activation function-based generation adversarial network collaborative filtering algorithm has better results in most cases. The precision index can increase by about 4 percentage points, the recall index by about 3 percentage points and the F1 index by about 3 percentage points.

2. Organization of the Text

2.1. Relevant Work and Theoretical Basis

2.1.1. Neural Network Algorithm

Neural network algorithm, artificial neural network algorithm, is an end-to-end computing model, which belongs to deep learning. Artificial intelligence contains machine learning, machine learning contains deep learning, artificial intelligence is a general term. The main difference between machine learning and deep learning is the processing of features. The algorithm is based on the original data and directly calculate, that is, manually write rules, to

ISSN: 2688-9323

obtain the final prediction results; machine learning needs to process on the basis of the original data, extract the original data feature vector, and then take the feature vector as the input of the machine learning algorithm, in which the processing process of extracting features is done manually; Deep learning does not need to display the extraction of features, the work of feature extraction in the depth learning algorithm, can be through the internal algorithm to extract more complex, represent more information rich combination features and according to this feature to predict the final results, is an end-to-end algorithm [5].

2.1.2. Activation Function

The activation function is a nonlinear function. the function of the activation function is to introduce nonlinear factors into the neural network algorithm, which makes the neural network change from a linear model to a nonlinear model. a simple and appropriate activation function gives the neural network a stronger fitting ability. the common activation functions currently used are relu, softmax, and tanh functions.

2.1.3. Neurons and Layers

Neurons are the constituent nodes in neural network algorithms.node represents a number. The output value of the activation function is the output value represented by the neuron node. The neural layer is divided into input layer, output layer and hidden layer according to the way of processing data. The input layer is the data vector input by the algorithm, the hidden layer is the process of data processing, and the output layer is the output result of the algorithm. For the regression problem, there is only one node in the output layer; for the classification problem, the number of nodes in the output layer is generally the number of categories.

2.1.4. Collaborative Filtering Algorithm

The idea of the recommendation algorithm based on collaborative filtering is to help users make decisions and recommendations through group behavior. The user-based collaborative filtering method embodies the idea of "people by people ", but the complexity of this method will rise with the increase of the number of users, it is difficult to achieve real-time recommendation. In order to solve this problem, the model-based method is proposed, and the hidden factor matrix decomposition model is a typical representative of the model-based collaborative filtering method.

2.1.5. Movielens Data Sets

Movielens dataset is the data collected by Grouplens research group on movielens website (http://movielens.org). This data set mainly describes the rating data of multiple users for multiple films, and also includes the metadata information of the film itself and the information of the user himself. The largest data set has 140,000 users,

covering 27000 films.

The most important data in the Movielens is the rating file, which contains the user id, the item id, the user's time stamp when grading the item

2.2. Model Solution

2.2.1. Generating Adversarial Networks

The generative adversarial network (Generative Adversarial Network, referred to as GAN) is an unsupervised machine learning method based on neural network. this algorithm consists of a generative network and a discriminative network, and two neural networks learn and [6,7] by mutual game.

the essence of the generating network is a neural network. the input of the generating network is a given noise data. the synthetic data is generated according to the noise data. the purpose is to make the synthetic data distribution closer to the real distribution.

ISSN: 2688-9323

Discriminating the nature of the network is also a neural network. Counteracting network input is a mixture of synthetic data and real data.

2.2.2. Collaborative Filtering Algorithm based on Generating Adversarial Networks

The input of the algorithm is a vector of 1682 dimensions, which is represented as 1682 nodes in the input layer. there are 4 hidden layers in the generation model of this paper. each hidden layer contains 400 neurons, in which the activation functions used in the hidden layer are all relu functions. the output layer is a neural layer containing 1682 neurons. the activation function used is the tanh function.

2.2.3. Model Framework

The main steps of solving the model include:

1. Collection of user-project interaction data
2. Construction of scoring matrix
3. Get the user purchase vector from the scoring matrix
4. Using User Purchase Vector Data to Train Generation Networks and Confrontation Networks
5. Predict a user's complete purchase vector using a trained generation network
6. Sort items according to the values in the purchase vector
7. Advice TopN items to users

2.2.4. Data Sampling Method

The implementation of negative sampling is to sample the incomplete real purchase vector of the user.

2.2.5. Forecast

After getting a trained generation model (generating network), we make a complete purchase vector prediction for each user, then sort the items and recommend K items with the top score to the user.

2.3. Common Activation Function

2.3.1. Relu Function

The relu function is the most frequently used activation function in the current artificial neural network. the relu function solves the problem that the class gradient disappears in the positive interval range, and its computation speed is fast and the computation resource consumption is less, and the relu function makes the convergence speed of the neural network much faster than that of the sigmoid function and the tanh function.

Although relu has many benefits, there are some problems. the output of the relu function is not a zero-mean distribution; there is a problem Relu death, that is, if the signal received by the neuron is a negative interval of the input of the relu function, then this neuron can never be activated, resulting in the corresponding parameters can never be updated.

2.3.2. Elu Function

The elu function is proposed to solve the problem of relu activation function. elu has the basic advantages. the difference between elu function and relu function lies in the value in the range of negative interval. There is no death Relu problem in elu function, which leads to the problem that the gradient of some neurons is zero and can not be updated, and the output value of the function is close to the zero mean distribution.

elu activation function has power operation in the range of negative interval, the calculation amount is slightly larger than the relu activation function, and the resources consumed are more.

ISSN: 2688-9323

2.3.3. An Adaptive Activation Function-based Generation Adversarial Network

An adaptive activation function-based generation network architecture layer is a 1682 dimension

The vector contains four hidden layers, each containing 400 neurons, where the hidden layer makes

The activation functions used are all adaptive activation functions (AAF), the output layer is a neural layer containing 1682 neurons, and the activation function used is tanh function

2.4. Experimental Analysis

2.4.1. Data Distribution

A dataset of movielens 100k users and 943 users was used in this experiment

Of the 1682 films, each user rated at least 20 films, with sparse data 93.69%.

2.4.2. Analysis of Experimental Results

2.4.3. Analysis of the Effect of Different Recommended List Lengths on Accuracy

The following figure, the horizontal axis represents the list length of the recommended items, the vertical axis represents the accuracy of the recommended results, CFGAN represents the collaborative filtering recommendation algorithm based on the generated adversarial network network, CFGAN_AAF represents the generation adversarial collaborative filtering recommendation algorithm based on the adaptive activation function. With the increase of recommendation list length, the accuracy of both methods decreases gradually, that is, the more items recommended, the more items recommended will decrease. CFGAN_AAF except when the length of the list is equal to 15, the accuracy is slightly lower CFGAN, other cases are higher than that. Thus, the adaptive activation function improves the learning ability of generating network and discriminating network.

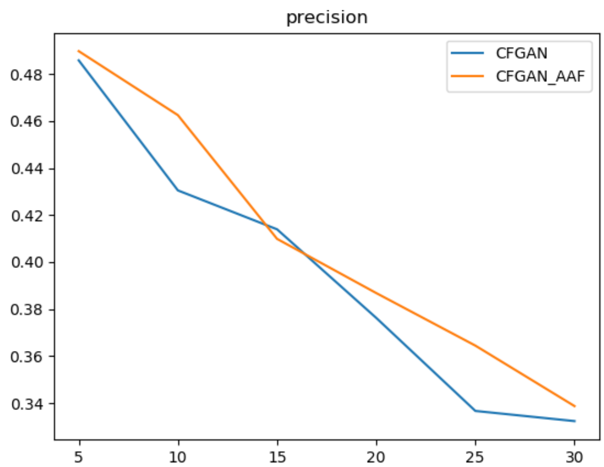

Fig 1. The CFGAN and CFGAN_AAF of Fig.

2.4.4. Analysis on the Effect of Different Recommended List Length on Recall Rate

As shown below, the horizontal axis also represents the length of the final list of recommended items and the vertical axis represents the recall rate. It can be seen from the figure that with the increase of the number of recommended items, the value of recall rate is also increasing. the recall rate obtained by CFGAN_AAP method is slightly higher than that of CFGAN algorithm.

ISSN: 2688-9323

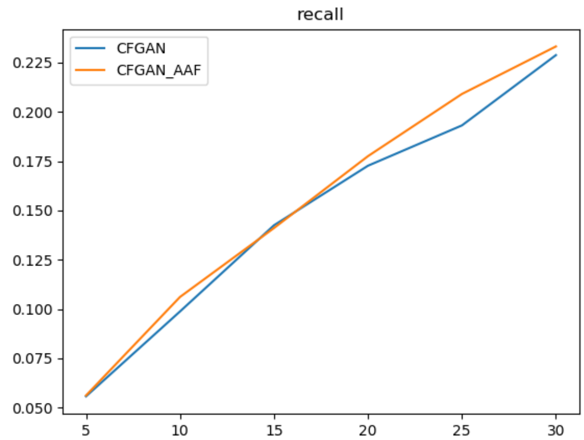

Fig 2. The CFGAN and CFGAN_AAF of Fig

Combined with the accuracy rate and recall rate, the accuracy rate will decrease with the increase of the number of recommended items, and the recall rate will increase with the increase of the number of recommended items, that is, the accuracy rate will increase with the decrease of the accuracy rate, and the recall rate will increase.

2.4.5. Analysis of the Effect of Different Recommended List Lengths on F Values

As shown below, the horizontal axis represents the number of recommended items and the vertical axis represents the size of the F value. overall, the number of recommended items increases, and the F value of the CFGAN_AAF is also higher than that of the CFGAN method. this also shows that the generation-based adversarial network collaborative filtering recommendation based on adaptive activation function is better than the generation-based adversarial network collaborative filtering recommendation algorithm. the model trained using the former has stronger prediction ability.

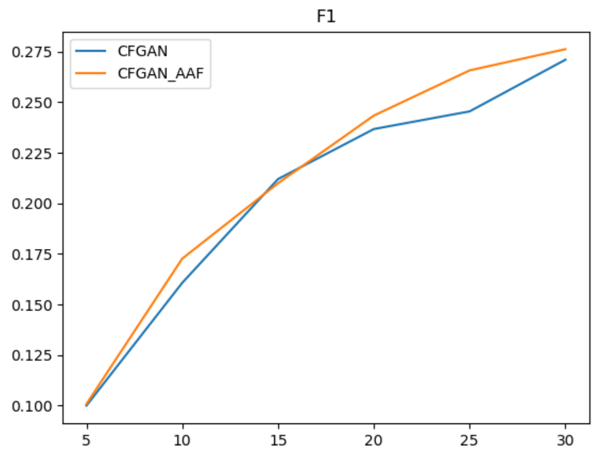

Fig 3. CFGAN and CFGAN_AAF change with N F1 values

3. Literature References

In rating-based collaborative filtering, we have m users, n items, and a partially observed user-item rating matrix R $\in \mathbb{R}^{m\times n}$. Each user u $\in$ U = {1 . . . m} can be represented by a partially observed vector r(u) = (Ru1, . . . Run) $\in \mathbb{R}^n$. Similarly, each item i $\in$ I = {1 . . . n} can be represented by a partially observed vector r(i) = (R1i, . . . Rmi) $\in \mathbb{R}^m$. Our aim in this work is to design an item-based (user-based) autoencoder which can take as input each partially observed r(i) (r(u)), project it into a low-dimensional latent (hidden) space, and then reconstruct r(i) (r(u)) in the output space to predict missing ratings for purposes of recommendation. Formally, given a set S of vectors in R d , and some k ∈ N+, an autoencoder solves

$$\min_{\theta} \sum_{\mathbf{r} \in \mathbf{S}} ||\mathbf{r} - h(\mathbf{r}; \theta)||_2^2, \tag{1}$$

where h(r; θ) is the reconstruction of input r $\in \mathbb{R}d$,

$$h(\mathbf{r}; \theta) = f\left(\mathbf{W} \cdot g(\mathbf{V}\mathbf{r} + \boldsymbol{\mu}) + \mathbf{b}\right)$$

for activation functions f(·), g(·). Here, θ = {W, V, μ, b} for transformations W $\in \mathbb{R}d \times k$, V $\in \mathbb{R}k \times d$, and biases μ $\in \mathbb{R}k$, b $\in \mathbb{R}d$. This objective corresponds to an auto- associative neural network with a single, k-dimensional hidden layer. The parameters θ are learned using backpropagation. The item-based AutoRec model, shown in Figure 4,
applies an autoencoder as per Equation 1 to the set of vectors

$$\left\{r^{(i)}\right\}_{i=1}^{n},$$

with two important changes. First, we account for the fact that each r(i) is partially observed by only updating during backpropagation those weights that are associated with observed inputs, as is common in matrix factorisation and RBM approaches. Second, we regularise the learned parameters so as to prevent overfitting on the observed ratings. Formally, the objective function for the Item-based AutoRec (I-AutoRec) model is, for regularisation strength λ > 0,

$$\min_{\theta} \sum_{i=1}^{n} ||\mathbf{r}^{(i)} - h(\mathbf{r}^{(i)}; \theta))||_{\mathcal{O}}^2 + \frac{\lambda}{2} \cdot (||\mathbf{W}||_F^2 + ||\mathbf{V}||_F^2) \tag{2}$$

where || ·||2 O means that we only consider the contribution of observed ratings. User - based AutoRec (U-AutoRec) is derived by working with {r(u) } m u=1. In total, I- AutoRec requires the estimation of 2mk + m + k parameters. Given learned parameters θ, I-AutoRec's predicted rating for user u and item i is

$$\hat{R}_{ui} = (h(\mathbf{r}^{(i)}; \hat{\theta}))_u. \tag{3}$$

Figure 4 illustrates the model, with shaded nodes corresponding to observed ratings, and solid connections corresponding to weights that are updated for the input r(i) .

ISSN: 2688-9323

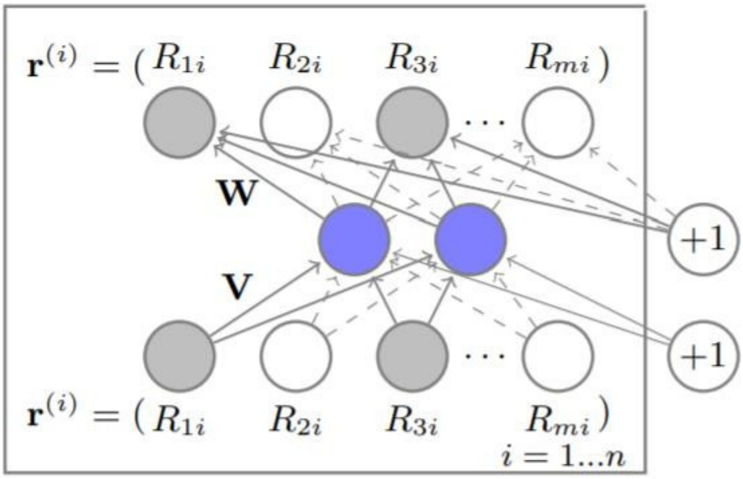

Fig 4. Item-based AutoRec model. We use plate notation to indicate that there are n copies of the neural network (one for each item), where W and V are tied across all copies.

RMSE for I-AutoRec with choices of linear and nonlinear activation functions, Movielens 1M dataset.

Comparison of I-AutoRec with baselines on MovieLens and Netflix datasets.

We remark that I-RBM did not converge after one week of training. LLORMA's performance is taken from [2].

Table 1. (a) Comparison of the RMSE of I/U-AutoRec and RBM models

	ML-1M	ML-10M
U-RBM	0.881	0.823
I-RBM	0.854	0.825
U-AutoRec	0.874	0.867
I-AutoRec	**0.831**	**0.782**

(a)

$f(\cdot)$	$g(\cdot)$	RMSE
Identity	Identity	0.872
Sigmoid	Identity	0.852
Identity	Sigmoid	**0.831**
Sigmoid	Sigmoid	0.836

(b)

	ML-1M	ML-10M	Netflix
BiasedMF	0.845	0.803	0.844
I-RBM	0.854	0.825	-
U-RBM	0.881	0.823	0.845
LLORMA	0.833	**0.782**	0.834
I-AutoRec	**0.831**	**0.782**	**0.823**

(c)

AutoRec is distinct to existing CF approaches. Compared to the RBM-based CF model (RBM-CF) [4], there are several differences. First, RBM-CF proposes a generative, probabilistic model based on restricted Boltzmann machines, while AutoRec is a discriminative model based on autoencoders. Second, RBM-CF estimates parameters by maximising log likelihood, while AutoRec directly minimises RMSE, the canonical performance in rating prediction tasks. Third, training RBM-CF requires the use of contrastive divergence, whereas training AutoRec requires the comparatively faster gradient-based backpropagation. Finally, RBM-CF is only applicable for discrete ratings, and estimates a separate set of parameters for each rating value. For r possible ratings,

this implies nkr or (mkr) parameters for user- (item-) based RBM. AutoRec is agnostic to r and hence requires fewer parameters. Fewer parameters enables AutoRec to have less memory footprint and less prone to overfitting. Compared to matrix factorisation (MF) approaches, which embed both users and items into a shared latent space, the item-based AutoRec model only embeds items into latent space. Further, while MF learns a linear latent representation, AutoRec can learn a nonlinear latent representation through activation function g(·).

In this section, we evaluate and compare AutoRec with RBM-CF [4], Biased Matrix Factorisation [1] (BiasedMF), and Local Low-Rank Matrix Factorisation (LLORMA) [2] on the Movielens 1M, 10M and Netflix datasets. Following [2], we use a default rating of 3 for test users or items without training observations. We split the data into random 90%–10% train-test sets, and hold out 10% of the training set for hyperparamater tuning. We repeat this splitting procedure

ISSN: 2688-9323

5 times and report average RMSE. 95% confidence intervals on RMSE were ±0.003 or less in each experiment. For all baselines, we tuned the regularisation strength $\lambda \in \{0.001, 0.01, 0.1, 1, 100, 1000\}$ and the appropriate

latent dimension $k \in \{10, 20, 40, 80, 100, 200, 300, 400, 500\}$. A challenge training autoencoders is non-convexity of the objective. We found resilient propagation (RProp)

[3] to give comparable performance to L-BFGS, while being much faster. Thus, we use RProp for all subsequent experiments: Which is better, item- or user-based autoencoding with RBMs or AutoRec? Table 1a shows item-based (I-) methods for RBM and AutoRec generally perform better; this is likely since the average number of ratings per item is much more than those per user; high variance in the number of user ratings leads to less reliable prediction for user-based methods. I-AutoRec outperforms all RBM variants. How does AutoRec performance vary with linear and nonlinear activation functions $f(\cdot)$, $g(\cdot)$? Table 1b indicates that nonlinearity in the hidden layer (via $g(\cdot)$) is critical for good performance of I-AutoRec, indicating its potential advantage over MF methods. Replacing sigmoids with Rectified Linear Units (ReLU) performed worse. All other AutoRec experiments use identity $f(\cdot)$ and sigmoid $g(\cdot)$ functions.

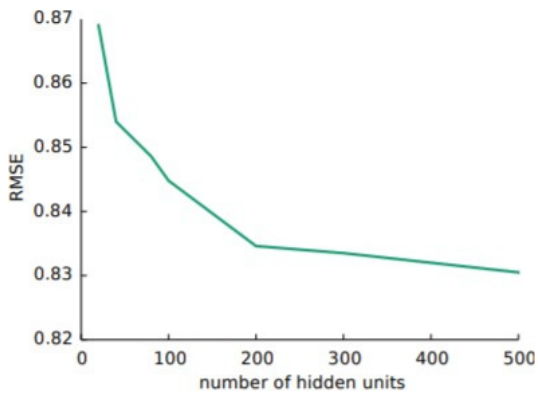

Fig 5. RMSE of I-AutoRec on Movielens 1M as the number of hidden units k varies.

How does performance of AutoRec vary with the number of hidden units? In Figure 5, we evaluate the performance of AutoRec model as the number of hidden units varies.We note that performance steadily increases with the number of hidden units, but with diminishing returns. All other AutoRec experiments use $k = 500$. How does AutoRec perform against all baselines? Table 1c shows that AutoRec consistently outperforms all baselines, except for comparable results with LLORMA on Movielens 10M. Competitive performance with LLORMA is of interest, as the latter involves weighting 50 different local matrix factorization models, whereas AutoRec only uses a single latent representation via a neural net autoencoder. Do deep extensions of AutoRec help? We developed a deep version of I-AutoRec with three hidden layers of (500, 250, 500) units, each with a sigmoid activation. We used greedy pretraining and then fine-tuned by gradient descent. On Movielens 1M, RMSE reduces from 0.831 to 0.827 indicating potential for further improvement via deep AutoRec. Acknowledgments NICTA is funded by the Australian Government as represented by the Dept. of Communications and the ARC through the ICT Centre of Excellence program. This research was supported in part by ARC DP140102185.

4. Conclusion

In this paper, a new generation adversarial network algorithm based on trainable activation function is proposed. Firstly, the neural network algorithm is introduced in detail. Secondly, the two most classical algorithms are introduced: user-based cooperative filtering algorithm and matrix decomposition algorithm based on hidden factors. Then, the problem is analyzed and

ISSN: 2688-9323

expounded in depth. Then, the cooperative filtering algorithm based on generative adversarial network is proposed, and then improved. a Movielens data set is used in this experiment. after a large number of experiments, the proposed generative adversarial network based on trainable activation function can improve the prediction ability of the model to some extent.

In the experiment of this paper, different neural network structures are tried, mainly:

(1) Different layers of nerve

(2) Number of different neurons in the nerve layer

At the same time, a variety of combinations of different hyperparameters have been tried, mainly:

(1) Batch size of data input

(2) Total number of training sessions

(3) different gradient descent algorithms, such as stochastic gradient descent, RMSprop, Adam, etc.

(4) Learning rate in gradient descent algorithm of neural network

In general, in order to alleviate the sparse problem in the recommendation system, this paper proposes a collaborative filtering algorithm using generative adversarial network structure and trainable activation function. the traditional generative adversarial network learning ability is limited. the use of trainable activation function can enhance the learning ability of generative adversarial network model. so that the collaborative filtering algorithm based on generative adversarial network can better learn the interactive data distribution between users and projects, make the generated network data closer to the distribution of real data, and the generated network generated data can alleviate the problem of data sparsity to some extent by experimental tests.

Acknowledgements

First and foremost, I appreciate my college which gives me a comfortable learning atmosphere. Second, I would like to show my deepest gratitude to my supervisor, Mr Wang, who has walked me through all the stages of the writing of this thesis. Without his illuminating instruction and patience, this thesis could not have reached its present form. I am also greatly indebted to all my teachers who have helped me to develop the fundamental and essential academic competence. My sincere appreciation also goes to all my classmates, who are my proud of my life. Last but not least, I want to thank all my friends, especially my lovely roommates, for their encouragement and support.

References

[1] Cheng H T, Koc L,Harmsen J,et al. Wide &deep learning for recommender systems[C]//Proceedings of the 1st workshop on deep learning for recommender systems.2016 : 7-10.

[2] Guo H,Tang R,Ye Y,et al.DeepFM:a factorization-machine based neural network for CTR prediction [J]. arXiv preprint arXiv:1703.04247,2017.

[3] Ma X,Zhao L, Huang G,et al.Entire space multi-task model; An effective approach for estimating post-click conversion rate [C] //The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.2018 : 1137-1140.

[4] Karatzoglou A, Hidasi B. Deep learning for recommender systems[C]//Proceedings of the eleventh ACM conference on recommender systems.2017:396-397.

[5] Hidasi B, Quadrana M,Karatzoglou A, et al. Parallel recurrent neural network architectures for feature-rich session-based recommendations[C]//Proceedings of the 10th ACM Conference on Recommender Systems.2016 : 241-248.

ISSN: 2688-9323

[6] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42(8) : 30-37.

[7] Xue H J,Dai X,Zhang J,et al.Deep Matrix Factorization Models for Recommender Systems [C]// IJCAI. 2017: 3203-3209.