Using neural networks with reinforcement in the tasks of forming user recommendations

R S Kulshin, A A Sidorov and P V Senchenko

Tomsk State University of Control Systems and Radioelectronics, 40, Lenina Ave., Tomsk, 634050, Russia

E-mail: kulshinroman@mail.ru

Abstract. The article discusses the design of an algorithm for the formation of personalized user recommendations. The basis of the algorithm is a recurrent neural network with reinforcement. The use of neural networks in the recommendation algorithm will help to increase persistence as it becomes possible to identify weak relationships of implicit factors.

1. Introduction

In the modern world, more and more people are faced with a large number of products on the Internet. A huge number of companies from different industries around the world are striving to transfer their services from the real world to the digital one, reaching an increasing audience and, accordingly, selling more and more goods on the Internet. A variety of products pose the user with the problem of choosing a particular product.

In this situation, information technologies come to the user's aid. Huge corporations and small companies are starting to implement recommendation systems everywhere. Recommendation systems are used to improve the quality of products offered to the user, thereby increasing the company's profit. The most high-quality recommendation systems form personalized recommendations that are focused on each individual user.

The high demand of companies for recommendation systems has led to rapid development and great interest in algorithms for generating user recommendations.

The basics of recommendation generation technologies appeared quite a long time ago, at the time of the advent of machine learning. It was then that the basic mathematical principles and algorithms were laid down. Since then, a lot of changes have taken place, a large number of hybrid approaches to the formation of recommendations have appeared.

The purpose of this article is a theoretical justification of the possibility of using neural networks to solve the problem of forming personalized recommendations.

Next, the main methods and algorithms for forming recommendations will be considered, their main disadvantages are found and their own solution based on the use of neural networks with reinforcement is proposed.

2. Overview of methods for making recommendations

Currently, there are three main approaches and their hybrids for the formation of user recommendations: singular matrix decomposition (SVD), content filtering method (CBF) and collaborative filtering method (CF).
Singular matrix decomposition is a fairly simple but very effective tool. The principle of its operation is based on reducing the number of parameters of a sparse matrix, which consists of ratings that users have assigned to a particular product in the system [1]. In the tasks of forming recommendations, the matrix is always considered to be discharged, since one user cannot evaluate a significant proportion of products. There also cannot be many products that a significant proportion of users will be able to evaluate. The main advantages of this method are: reducing the amount of stored data (this problem is one of the main ones in the algorithms of recommendation systems) and getting rid of redundant and unused parameters in the processing process.

The content filtering method is one of the very first algorithms for generating recommendations. The principle of its operation is to identify products with similar characteristics in relation to those that have already interested the user. When forming recommendations with this algorithm, products with similar characteristics are highlighted [2]. For the effective operation of the content filtering method, a detailed description of the characteristics of the product, as well as information about a specific user, is necessary. The main advantage of this algorithm is a partial solution to the cold start problem, for a new user, the method quickly enough determines a set of products that will be of interest to him.

The collaborative filtering method, as well as its hybrids and modifications, is most common in the field of personalized recommendations formation. The principle of operation of the method is based on the assumption that its preferences are conservative (if users have evaluated a certain product equally, then it is quite likely that users will similarly evaluate other products with similar characteristics). The collaborative filtering method generates personalized recommendations based on the behaviors of users similar to it [3]. Two types of collaborative filtering methods are most common: collaborative filtering by analyzing the preferences of groups of users with similar characteristics (User-User CF) and collaborative filtering by identifying relationships between products (Item – Item CF). The main advantage of this method is the high theoretical accuracy of user recommendations, which leads to the achievement of high persistence.

3. Identification of shortcomings of existing algorithms and methods

During the study and analysis of the methods and algorithms described above for the formation of user recommendations, their main disadvantages were identified: sparsity of data, the problem of cold start, insufficient diversity, insufficient persistence, scalability [4].

The problem of sparsity of data is the large number of users in the system. As a result, the "subject-user” matrix turns out to be very large, and its fullness is insufficient for accurate forecasting.

The problem of cold start is the most difficult in terms of its solution. Its essence lies in the fact that new products have not yet been allocated a sufficient number of features to compare with a certain group of products, but for algorithms based on the context of the product, this problem is less significant. But for new users, a sufficiently optimal solution has not yet been found in the system.

The problem of insufficient diversity is due to the fact that the most popular algorithms will continue to become even more popular. Many algorithms create a complex environment for the promotion of insufficiently popular or newly appeared products.

Insufficient persistence is mainly manifested due to the fact that many algorithms are based on explicit factors (usually purchases and product ratings), but do not take into account a huge number of implicit factors (for example, time spent on the product page, viewed product photos, and so on). The inability to process implicit factors is due to the fact that the user's behavioral model is not supported by the main algorithm in any way, thereby missing a large amount of useful data.

With the increase in the number of users in the system, there is a scalability problem. Recommendation systems should give almost instant results to a request for the formation of a selection of recommended products, but a huge number of users and products leads to a large increase in the required computing power, and, accordingly, time.
4. **Formation of recommendations using a neural network with reinforcement**

In 2017, an article was published describing the use of neural networks to improve the accuracy of the collaborative filtering method, this solution was called neural collaborative filtering (NCF) [5]. The basic principle of operation of this algorithm is to replace matrix factorization with a multilayer perceptron to reduce the linearity of modeling.

Having analyzed the principles of the algorithms for the formation of recommendations and their main problems, a proprietary solution was proposed.

Based on the experience of using neural networks in the collaborative filtering method, we have a clear idea that the use of neural networks gives an increase in persistence. But the existing solution is based only on obvious factors.

Our proposed solution is to use neural networks with reinforcement. The target action of the user in the Internet system is used as reinforcement. The use of reinforcement in neural network training will allow us to take into account a huge number of implicit factors, which should lead to a strong increase in persistence [6].

The recurrent neural network (RNN) was chosen as the type of neural network. RNN is well suited for processing incoming sequences. The RNN type used is many to one. The architecture of the neural network is the Jordan network. In this architecture, the input layer is connected to auxiliary blocks. Auxiliary blocks retain the previous values, so this network can be used in sequence processing [7]. This approach goes beyond the possibility of a multilayer perceptron.

The input layer of the neural network designed by us consists of two subnets. The first subnet is a recurrent layer that accepts product characteristics. The second subnet is a recurrent neural layer that accepts many factors of user interaction with a certain product. The use of recurrent layers allows the neural network to accept more than one set of factors of user interaction with the product. In addition to the input layer itself, two contextual blocks are used to enable the operation of a recurrent neural network.

After the input layer comes the embedding layer. It combines two subnet vectors into a dense vector for further processing.

Next comes a sequence of hidden layers, in which the signs of the most suitable product for the recommendation to the user will be highlighted. Sigmoid was chosen as the activation function. This activation function is nonlinear, and the combination of such functions makes it possible to achieve even greater nonlinearity [8]. The sigmoid is characterized by a smooth gradient, since it is analog, and not binary as a step function. The combination of sigmoid activation functions will allow the neural network to identify even the most insignificant and hidden relationships in the input parameter.

The neurons of the last hidden layer are reduced to the last output layer with one neuron, because we need to get a single value, the probability of recommending a specific product to a specific user. The linear rectifier (ReLU) acts as an activation function on the output layer [9]. Therefore, at the output we will get values from 0 to 1, which corresponds to probability.

A modification of the Adagard – AdaDelta method is used as an optimization method. The AdaDelta method uses different learning rates for different parameters, while using past gradients, in a fixed-size window calculated for each specific parameter [10].

Adaptive boosting algorithm (AdaBoost) was chosen as a loss function to provide neural network training. This algorithm is adaptive, because each iteration of training is based on incorrectly classified in the past times [11]. It is less susceptible to retraining compared to analogues.

The trigger for training is the target action of the user regarding the product recommended to him. For example, buying an item or adding it to the cart. After performing a target action or a prolonged absence of action, the loss function receives a signal, and the neural network begins its calibration.

The neural network diagram is shown in Figure 1.
In the figure shown above, the values $a_1$, $a_2$, $a_i$ mean a set of parameters of a particular user that enter the neural network. Under the values $b_1$, $b_2$, $b_i$ there are many parameters of a specific product. At the output of the neural network, we get the $y$ value, which is the probability of a product recommendation.

Neural network training will be performed on the prepared dataset "Retailrocket recommender system dataset" which contains more than 2.5 million user events in a real online store and more than 1600 products. The data in this dataset has all the necessary characteristics for the subsequent training of a neural network.

A partial solution to the cold start problem is supposed to be performed by revealing hidden dependencies by the neural network. The detection of hidden dependencies should not lead to a complete solution to the cold start problem, but should accelerate the process of correct processing of user data that has recently registered in the system.

Figure 1. Neural network diagram
5. Conclusion
Thus, the designed algorithm for generating user recommendations based on the use of neural networks by reinforcement should improve the quality of the generated recommendations. The target action of the user in the Internet system is used as reinforcement. This gives the algorithm the ability to identify many relationships of implicit factors, which leads to increased persistence. Also, due to the use of neural networks, the sparsity of data is reduced.

Based on the results of the algorithm development and its testing, the optimal number of hidden layers and neurons contained in them will be formed. It is also possible to change the topology and internal structure of the neural network.

Acknowledgements
This paper is designed as part of the state assignment of the Ministry of Science and Higher Education; project FEWM-2020-0036.

References
[1] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2018 2.6 Singular Value Decomposition. Numerical Recipes in C. 2nd edition (Cambridge: Cambridge University Press)
[2] Kant V, Bharadwaj K K 2012 Enhancing Recommendation Quality of Content-based Filtering through Collaborative Predictions and Fuzzy Similarity Measures. Procedia Engineering 38 (2012) 939-944
[3] Su X and Khoshgoftaar T M 2009 A Survey of Collaborative Filtering Techniques A Survey of Collaborative Filtering Techniques, Advances in Artificial Intelligence Archive, 1-19
[4] Sammut C, Webb J 2010 Encyclopedia of Machine Learning. (NY, USA: IBM T J Watson Research Center) p 1031
[5] He X, Liao L, Zhang H, Nie L, Hu X and Chua T-S 2017 Neural Collaborative Filtering, Proceedings of the 26th International Conference on World Wide Web, pp 173–182
[6] Kaelbling L P, Littman M L, Moore A W 1996 Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research. 4 237–285
[7] Jordan M I 1997 Serial Order: A Parallel Distributed Processing Approach. Neural-Network Models of Cognition - Biobehavioral Foundations. Advances in Psychology. 121 471–495
[8] Han J and Moraga C 1995 The influence of the sigmoid function parameters on the speed of backpropagation learning. From Natural to Artificial Neural Computation. Lecture Notes in Computer Science. 930 195–201
[9] Behnke S 2003 Hierarchical Neural Networks for Image Interpretation. Lecture Notes in Computer Science. 2766
[10] Zeiler M D 2012 ADADELTA: An adaptive learning rate method. arXiv:1212.5701
[11] Schapire R and Singer Y 1999 Improved Boosting Algorithms Using Confidence-rated Predictions 80–91