RecDNNing: a recommender system using deep neural network with user and item embeddings

Hafed Zarzour[1], Ziad A. Al-Sharif[2], and Yaser Jararweh[3]

Dept. of Computer Science, University of Souk Ahras, Souk Ahras, Algeria [1]
hafed.zarzour@gmail.com [1]
Dept. of Software Engineering, Jordan University of Science and Technology, Irbid, Jordan [2]
Dept. of Computer Science, Jordan University of Science and Technology, Irbid, Jordan [3]
{zasharif[2], yijararweh[3]}@just.edu.jo

***Abstract*—The success of applying deep learning to many domains has gained strong interest in developing new revolutionary recommender systems. However, there are little works studying these systems that employ deep learning; additionally, there is no study showing how to combine the users and items embedding with deep learning to enhance the effectiveness of the recommender systems. Therefore, this paper proposes a novel approach called RecDNNing with a combination of embedded users and items combined with deep neural network. The proposed recommendation approach consists of two phases. In the first phase, we create a dens numeric representation for each user and item, called user embedding and item embedding, respectively. Following that, the items and users embedding are averaged and then concatenated before being fed into the deep neural network. In the second phase, we use the model of the deep neural network to take the concatenated users and items embedding as the inputs in order to predict the scores of rating by applying the forward propagation algorithm. The experimental results on MovieLens show that the proposed RecDNNing outperforms state-of-the-art algorithms.**

***Index Terms*—recommender system, deep learning, deep neural network, user embedding, item embedding.**

I. Introduction

A recommender system is an important and attractive domain in information filtering systems that aims at predicting the preference or the rating of active users of a given item. A recommender system can be a powerful engine employing various techniques to recommend the most relevant items for a set of users based on their past behaviors and tastes. In past decades, each of the online-based companies specializing in Internet-related services and/or products integrate a recommender system with a specific recommendation strategy. For examples, Netflix makes recommendations for a user based on the habits of watching and searching of similar users. A social network such as LinkedIn and Facebook recommends (suggests) new friends by comparing the networks of connections between different users and their friends.

However, two main methods of recommender systems can be distinguished: collaborative filtering and content-based filtering methods. Collaborative filtering technique is the most commonly used in practice due to its ease of use and feasibility. These techniques can be divided into two categories: memory-based and model-based strategies [1], [2]. While the memory-based methods use the entire similarities between items or users to produce predictions, the model-based methods employ the ratings in training the model, which is then used to produce predictions for users' ratings of an unrated item or items. For instance, Zarzour et al. [3] proposed a new effective model-based recommender system for TED talks, in which their approach first groups users according to their preferences and then provides a powerful mechanism to improve the quality of recommendations for users. In addition, the authors in [4] presented two versions of algorithms for designing an effective recommender system. In the first version, they use only the improved k-means clustering method, whereas in the second version, they combine the improved k-means clustering method with the principal component analysis to improve the recommendation accuracy in big data context. Furthermore, the authors in [5] develop a new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques, which focuses on k-means clustering algorithm to create groups of users that are sharing the same interests and SVD to reduce the dimensionality.

On the other hand, content-based filtering techniques use the user's preferences profiles as well as the descrip-

978-1-7281-0045-6/19/$31.00 ©2019 IEEE

tion of item to make recommendation about items, which are most similar to the items that are highly rated in the past. However, most of these classical recommender systems suffer from several problems such as sparsity, scalability, and cold-start.

The sparsity problem happens due the fact that only some few items get rated by users in the user-items ratings matrix, where such items are considered to be the most popular elements. For instance, only one percent of videos from Netflix dataset, which contains 18000 rated by over 480000 users generating about 100 million ratings, has ratings [6]. The cold-start problem comes from the fact that the recommender system cannot make any recommendations for new items, new communities, new users, or all because it has not yet collected sufficient information about them. The scalability problem is caused when the recommender system works only for small datasets and cannot work for large datasets.

Recently, the success of applying deep learning [7] to many fields, such as computer vision and data analysis, has gained strong interest in developing new revolutionary recommender systems with a significant improvement in their performance. However, there are little works studying the recommendation systems using these techniques; additionally there is no study showing how to combine the users and items embedding with deep learning to enhance the effectiveness of the recommender systems.

Therefore, this paper proposes a novel approach called RecDNNing that it can be used as a recommendation framework with a combination of embedded users and items combined with deep neural network to improve the quality of recommendation. RecDNNing is composed of two stages. First, we create the input vectors for users and items embeddings. Then, we make recommendations using the deep neural network. An experimentation is conducted on MovieLens dataset [1], and the experimental results show that RecDNNing outperforms some of the state-of-the-art algorithms.

The rest of this paper is organized as follows: Section II presents related works. Section III describes our proposed RecDNNing approach. Section IV illustrates our experimentation and evaluates our results. Conclusions and future work are described in Section V.

II. Related Work

Deep learning is considered a sub-domain of Artificial Intelligence (AI), and more precisely the one based on Machine Learning (ML). Because of their capability in solving a huge number of complex problems, the deep learning techniques have been successfully applied to a wide range of fields such as speech recognition [8] and computer vision [9]. With the recent important development of these techniques, many types of powerful recommender systems based on deep leaning have gained significant attention, including news-based recommendation [10], Apps-based recommendation [11] and video-based recommendation [12].

The authors in [13] proposed a neural autoregressive method for collaborative filtering recommendation, which adapts the Restricted Boltzmann Machine technique for making recommendation based on Neural Autoregressive Distribution Estimator (NADE). The recommendation performance is enhanced by sharing some parameters between ratings made by users. They show that their method with one hidden layer outperforms all previous state-of-the-art methods on several datasets.

In [14], the authors used an outer product for explicitly modeling the pairwise correlations between the embedding space dimensions by employing the convolutional neural network. Their experimental results show that their solution beats the well-known methods in top-k recommendation.

Recently, many deep learning techniques have been proposed in the literature with the aim at addressing traditional problems of collaborative filtering methods such as the sparsity and cold-start issues. For example, Collaborative Deep Learning (CDL) [15] is recently investigated by performing deep representation learning for content information coupled with collaborative filtering for the ratings matrix. CDL uses a deep learning model based on Bayesian formulation to enable two-way interactions between the content information and the feedback from the ratings matrix. The experimentations of CDL on real-world datasets from Netflix [2] and CiteULike [3] show its significant encroachment over existing methods.

In [16], the authors proposed a novel deep learning based hybrid service recommendation approach for Web service recommendation by combining collaborative filtering and textual content. The deep neural network is used to make characterization of the complex relations between services and mashups with the use a sparse interaction matrix. The experimental results on a real-world Web service dataset called ProgrammableWeb

[1]The MovieLens Dataset, https://movielens.org/

[2]Netflix, https://www.kaggle.com/netflix-inc/netflix-prize-data

[3]CiteULike, http://konect.cc/networks/citeulike-ut/

demonstrate the effectiveness of this method in service recommendation.

In [17], the authors developed a model of Neural Network called Deep Cooperative Neural Networks enabling to create a model for items and users based on review text for rating prediction problems. They used two coupled neural networks in order to ensure that the model learns the features of each user and item. Theses learned features are then used to the corresponding rating on top of both networks using the matrix factorization algorithms.

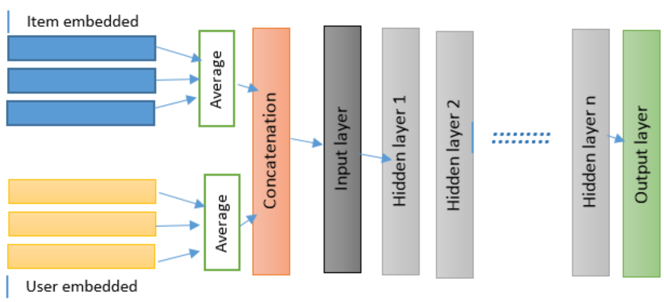

Fig. 1. The architecture of proposed RecDNNing.

III. The proposed Rec DNN*ing* approach

The main idea of the proposed approach, which is called RecDNNing, is to use the history of both users and items from the user-item rating matrix to predict the rating scores. Figure 1 illustrates the general architecture of the RecDNNing with a combination of embedded users and items combined with deep neural network. The proposed recommendation approach consists of two phases. In the first phase, we create a dens numeric representation for each user and item, called user embedding and item embedding, respectively. After that, the items and users embeddings are averaged and then concatenated before being fed into the deep neural network. In the second phase, we use the model of deep neural network to take the concatenated users and items embeddings as the inputs in order to predict the scores of rating by applying the forward propagation algorithm.

A. Creating the input vectors for users and items embeddings

The embedding mechanism aims at transforming the categorical or discrete variables, which are sparse entities, into the continuous vectors in order to obtain dense representations. In our case, we distinguish between two types of embeddings: user and item embeddings. The user embedding is a dense vector representation that contains the users' history obtained after mapping the sparse data to the dense data. In the same way, the item embedding is a dense vector representation that contains the items' history obtained after mapping the sparse data to the dense data. To construct the input vector, we average the obtained item and user embeddings before concatenating them. The concatenation function is defined in Equation 1.

$$InVec = Conc(U, I) \tag{1}$$

The main steps for creating the input vector for user and item embeddings are as follows:

Step 1: Creating users embeddings from the user-item ratings matrix using Skip-gram technique to get dens users vectors.
Step 2: Creating items embeddings from the user-item ratings matrix using Skip-gram model to get dense items vectors.
Step 3: Averaging the users embeddings.
Step 4: Averaging the items embeddings.
Step 5: Generating the input vectors by concatenating users' vectors and items' vectors using Equation 1; the users' history and items' history are both embedded, averaged and concatenated to be used as input into the deep neural network.

B. Making recommendations using deep neural network

In this study, the proposed recommendation model focuses on the deep neural network that consists of the input layer, a set of hidden layers and the output layer. We employed the ReLU as the activation function, because of its efficiency [18] and simplicity in the optimization process [19]. Moreover, it is one of the most widely used in deep learning and Convolutional Neural Networks (CNN, or ConvNet). The ReLU activation function is defined in Equation 2, where V_i is the output vector at the i^{th} layer, W_i is the matrix of weight between two neighboring layers, A_i is the vector of bias at the layer i.

$$V_i = ReLU(W_i \cdot V_{i-1} + A_i) \tag{2}$$

The main steps for training the deep neural network and predicting the active users' rating score are described as follows:

Step 1: Bringing the input vector ($InVec$ obtained from Equation 1) into the first layer, which is the input layer of the deep neural network.
Step 2: Getting the output vector for each hidden layer using Equation 2.

Step 3: Applying SOFTMAX function with the following parameters: the bias and weight matrix of the output layer and the output vector of the last hidden layer (V_{i-1}).
Step 4: Assessing the difference between the supervised and predicted values using the cross entropy method.
Step 5: Predicting the rating score for an active user by calculating the max of the predicted result.

IV. EXPERIMENTATION

To evaluate the effectiveness of the proposed method, experimentations are conducted on the MovieLens dataset [20]. MovieLens is a dataset built from the MovieLens website, by the GroupLens Research Project [4] at the University of Minnesota. There are several versions of this dataset, among which we used the $MovieLens-100k$. It includes 1682 videos rated by 943 users, all of which generated nearly 100000 rating scores. All rating records are not negative and ranging from 1 to 5 stars. To assess the performance of the proposed method in comparison with the state-of-the-art baselines, the Root Mean Square Error (RMSE) is employed as a metric that measures the predictive accuracy. Typically, this metric gives the difference between the observed and predicted values about a user that selects a given item. A smaller value of RMSE suggests better performance. RMSE is described in Equation 3, where $p_{u,i}$ is the predicted rating for user u on item i, and $r_{u,i}$ is the actual rating for user u on item i, finally, N is the total number of ratings on the set of items.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}(p_{u,i} - r_{u,i})^2} \tag{3}$$

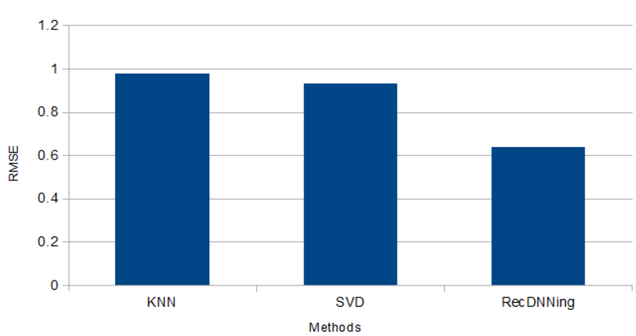

Fig. 2. The RMSE results on the MovieLens100k dataset with KNN, SVD and the proposed RecDNNing method.

[4]GroupLens Project, https://grouplens.org/datasets/movielens

Additionally, the SVD [21] and KNN [22] are selected as well-known recommendation methods in order to compare their performance with the performance of our proposed RecDNNing. In the context of recommender systems, SVD is considered a collaborative filtering method, which aims not only to reduce the features sizes for a given dataset using space dimensions reduction from A to B, where $B < A$, but also to cluster similar items in the singular value decomposition matrix. In other words, the SVD is a compact description of the whole user-item ratings matrix by taking into account the estimation of the messing entries values [23] . Furthermore, KNN is considered as one of the simplest algorithms used to solve recommendation problems while presenting significant advantages compared with many other methods developed for enhancing the performance of recommender systems. The idea behind it is to find a list containing the k nearest neighbors from the records of training [24]. To make recommendations for a given user, KNN runs the following:

1) Find the k neighbors of users regarding to the active user.
2) Perform an aggregation for unrated items.
3) Obtain the predictions from (2) and then return the list including the top k recommendations.

The MovieLens $100k$ dataset is split into training and testing sets with a percentage of 80% and 20%, respectively. Figure 2 shows the results of RMSE on the MovieLens $100k$ dataset with KNN, SVD and the proposed RecDNNing method. As we can see in this figure, the performance results of KNN are the worst, although it is independent from any assumptions on the distribution of data. This may be due to the similarity between item's feature or due to the dimensionality curse; specifically when using the Euclidean distance. The SVD method preforms well on the experimental dataset, but it is not the best. This is maybe because the missing values need to be replaced with modes or averages. The experimental results show that the proposed RecDNNing outperforms both the SVD and KNN methods on the MovieLens $100k$ dataset and obtains the best performance in terms of RMSE. The better prediction accuracy results can be explained by the fact that the use of users' and items' embeddings with deep neural network is very efficient.

V. CONCLUSION

In this paper, we have presented a novel approach called RecDNNing to enhance the quality of recommendation using the history of users and items coupled with deep neural network to predict the rating scores. RecDNNing is composed of two phases. First, the users and items embeddings are created, averaged, and concatenated. Then, a deep neural network took the obtained vectors as the inputs in order to predict the scores of rating using the forward propagation algorithm. The experimentations are conducted on MovieLens $100k$ dataset to evaluate the effectiveness of the proposed approach. The experimental results show that RecDNNing achieves higher prediction accuracy in terms of RMSE compared to the state-of-the-art algorithms, which confirms the importance of combining users and items embeddings with deep neural network for recommender systems. In future, we will explore more advanced deep learning methods to further enhance the quality of recommendation. Additionally, we will conduct more experimentations on additional datasets to evaluate the performance with other metrics such as the novelty and scalability.

REFERENCES

[1] T. Li, A. Liu, and C. Huang, "A similarity scenario-based recommendation model with small disturbances for unknown items in social networks," *IEEE Access*, vol. 4, pp. 9251–9272, 2016.

[2] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, "A hybrid recommender system using artificial neural networks," *Expert Systems with Applications*, vol. 83, pp. 300–313, 2017.

[3] H. Zarzour and Y. Jararweh, "An effective recommender system based on clustering technique for ted talks," *International Journal of Information Technology and Web Engineering*, vol. 15, 2020.

[4] H. Zarzour, F. Maazouzi, M. Soltani, and C. Chemam, "An improved collaborative filtering recommendation algorithm for big data," in *Computational Intelligence and Its Applications: 6th IFIP TC 5 International Conference, CIIA 2018, Oran, Algeria, May 8-10, 2018, Proceedings 6*. Springer, 2018, pp. 660–668.

[5] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub, and Y. Jararweh, "A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques," in *2018 9th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2018, pp. 102–106.

[6] J. Bennett, S. Lanning *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007. New York, NY, USA., 2007, p. 35.

[7] J. Liu, P. Zhou, Z. Yang, X. Liu, and J. Grundy, "Fasttagrec: fast tag recommendation for software information sites," *Automated Software Engineering*, vol. 25, no. 4, pp. 675–701, 2018.

[8] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[10] S. Okura, Y. Tagami, S. Ono, and A. Tajima, "Embedding-based news recommendation for millions of users," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1933–1942.

[11] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 2016, pp. 7–10.

[12] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*. ACM, 2016, pp. 191–198.

[13] Y. Zheng, B. Tang, W. Ding, and H. Zhou, "A neural autoregressive approach to collaborative filtering," *arXiv preprint arXiv:1605.09477*, 2016.

[14] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, "Outer product-based neural collaborative filtering," *arXiv preprint arXiv:1808.03912*, 2018.

[15] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2015, pp. 1235–1244.

[16] R. Xiong, J. Wang, N. Zhang, and Y. Ma, "Deep hybrid collaborative filtering for web service recommendation," *Expert Systems with Applications*, vol. 110, pp. 191–205, 2018.

[17] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 425–434.

[18] K. Jarrett, K. Kavukcuoglu, Y. LeCun *et al.*, "What is the best multi-stage architecture for object recognition?" in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2146–2153.

[19] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[20] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, p. 19, 2016.

[21] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artificial intelligence review*, vol. 50, no. 1, pp. 21–48, 2018.

[22] D. Adeniyi, Z. Wei, and Y. Yongquan, "Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method," *Applied Computing and Informatics*, vol. 12, no. 1, pp. 90–108, 2016.

[23] X. Guan, C.-T. Li, and Y. Guan, "Matrix factorization with rating completion: An enhanced svd model for collaborative filtering recommender systems," *IEEE access*, vol. 5, pp. 27 668–27 678, 2017.

[24] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 2017, pp. 306–310.