Movie Rating Prediction Using Deep Neural Networks

[1]Riyazahemed Jamadar,[2]Vishal Kore,[2]Ganesh Nage,[2]Kiran Alhat,[2]Ankita Kute

[1]Associate Professor,[2]Student,
*Information Technology,
*AISSMS Institute of Information Technology,Pune,India

*Abstract :*Recommender systems are used widely to recommend personalised products and services to customers. With recent development in Deep Learning, fields like Computer Vision, Natural Language Processing and Speech Recognition etc have shown great performance improvement. Use of Deep Learning in recommender systems also are showing great promises. In this paper we implemented collaborative filtering recommender systems based on Deep Neural Network. We make use of embedding layers to deal with sparse categorical variables. Results of our algorithm are comparable to state of art recommender system algorithms.

***Index Terms*–Recommendation Systems, Collaborative Filtering, Deep Learning, Movie Recommendation**

I. INTRODUCTION

Recommendation system helps user choose among overwhelming number of choices between product and services by offering relevant product and services. Recommender systems are widely used by commercial companies to give tailored experience for every user. Collaborative filtering and content-based filtering are main approaches used in recommender systems. Collaborative filtering recommends a product based on user’s previous data. Content-based recommender systems makes use of information about products and users to recommend products. Hybrid recommender systems combine two or more approaches. [[1], [2], [10]]

Neural network is architecture inspired from neurons in human brain. Neural networks use many machine learning algorithms to perform complex operations and learn tasks without being explicitly programmed. Deep neural networks are neural networks with more than one hidden layer. In recent years Deep learning has brought great improvement in Image processing, Natural language processing and speech recognition etc.

With impact of deep learning in fields like computer vision, natural language processing and speech recognition etc.it is also being used recommender systems. Deep learning-based recommender systems use Autoencoder,SVD and RBM for recommendation. In this paper we present a novel collaborative filtering algorithm for recommendation using deep neural network for movie rating prediction. [4], [5]

II. BACKGROUND

Gediminas Adomavicius,and Alexander Tuzhilin presented a survey of various approaches used for recommendation systems in collaborative filtering ,content-based filtering and hybrid methods. Also discussing limitation of current methodologies and hinting to what can be done to overcome such limitations[10].Liu Juntao and Wu Caihua released a study paper to explore deep learning methodologies for recommender systems and classified these approaches based on their input and output methodologies[1]. Li Deng and Dong Yu discussed deep neural networks and various architectures of deep neural network like Autoencoders and Restricted Boltzmann Machine. Following architectures, they discussed applications of deep neural networks in fields like speech recognition, audio processing, natural language processing, information retrieval, object detection and computer vision[4].

Hyeungill Lee and Jungwoo Lee proposed a collaborative filtering algorithm for movie rating predication using deep neural network. Using user normalized and item normalized vectors as input to neural network and batch normalization between hidden layers to prevent model from overfitting[11]. Heng-TzeCheng et al. proposed combining wide linear model with deep neural network model.Training wide liner models and deep neural networks jointly allows memorization and generalization to be combined to recommend more relevant products to the users[7].

Shuai Zhang et al. presented survey of current deep learning techniques used in recommendation systems. They also stated advantages of using deep learning in recommendation which include non-linear transformations, representational learning and sequence learning also discussing potential limitation of deep learning methods which include interpretability,data requirement and extensivehyperparameter tuning required by deep learning models. They also discussed state-of-the-art recommendation based on deep learning[2].Yang Shuo et al. proposed a hybrid recommendation technique for job recommendation. The used Statistical Relational Learning for combing content-based and collaborative filtering approach, which avoid immense feature engineering of user and item required by other hybrid methodologies[3].

Rim Fakhfakh et al. present a survey paper on recommendation approaches using deep learning. They discussed deep learningcontent-based and collaborative filtering based methodologies and current issues persisting in those methodologies like cold start problem, grey sheep and new user problem. Following this they discussed recommendation system architecture of YouTube, a major video recommendation platform by Google Inc.[9]. Lei zheng in his paper discussed recommendation systems and traditional techniques used in recommendation systems. After that he discussed state-of-the-art deep learning methods for recommendation systems and advantages and potential limitations of this methods[6].

Aaron van den oord et al. proposed a deep convolutional neural network based music recommendation system. Convolutional neural network is used to learn latent factors of audio signals. Using this approach, it is easy to recommend newly released songs to users and unlike many recommendation systems it uses deep convolutional neural networksfor recommendation[8].

III. MODEL ARCHITECTURE

Presented architecture makes use of embedding layers from Keras API to embed user id and movie id. This embeddings are vectors initialized to random number and represent latent factors of users and movies. The dot product of user embedding and movie embedding represents ratings given by the users to the movies. We train our deep neural network model to learn user embedding and user embedding which was initialized to random so that it can predict rating of movie which user has not yet seen.

These vectors are initialized to random numbers						-1.69	1.49	-0.14	1.95
Then we train model to optimize them						1.01	0.12	1.36	1.49
with loss function						0.82	1.48	0.02	0.53
						1.89	0.50	1.74	0.41
					movieId	2.39	1.13	1.15	-0.74
				userId		**27**	**49**	**57**	**72**
0.21	1.61	2.89	-1.26	0.82	**14**	3.25	5.10	0.98	3.23
1.55	0.75	0.22	1.62	1.26	**29**	4.40	4.98	5.08	3.99
1.50	1.17	0.22	1.08	1.49	**72**	4.43	4.94	4.98	4.13
0.47	0.89	1.32	1.13	0.77	**211**	5.16	4.21	4.01	2.83

Fig.1: Examples embedding vectors representing users and movies

The input to neural network is user id and movie id. Each individual input goes through embedding layer before it is feed to neural net. After obtaining embedding of each input it is concatenated and is given as input to neural network. Deep Neural Network is trained to optimize the value of embedding vector using backpropagation.

We used ReLU as activation function. ReLU overcomes the problem of vanishing gradient problem which is present in sigmoid and hyperbolic tangent function. To train Deep Neural Network with backpropagation we need a activation function which will act as linear function but will be a non-linear function. ReLU is activation function which returns zero if input value is negative and for any positive value it returns that value back. It can be written as $f(x) = \max(0, x)$

We used Adam optimizer from Tensorflow API. Adam optimizer is variation of gradient descent optimizer. Adam optimizer are memory efficient and works well with sparse data. Adam optimizer keeps track of separate learning rate for weight and decaying average of previous gradients.

We used RMSE (Root Mean Squared Error) as loss function. RMSE is known to penalize large errors in prediction. RMSE can defined as $\sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_p - y_T)^2}$ where Y_P represent predicted rating and $Y_T$represent true rating.

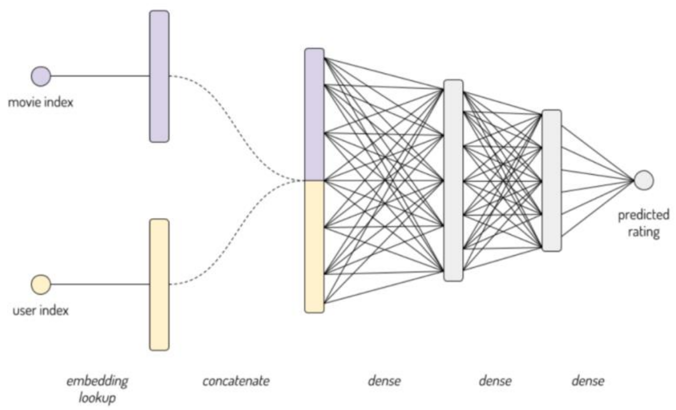


Fig.2: DNN Model architecture

IV. EXPERIMENT

3.1 Dataset

We trained our deep neural network on MovieLens dataset. MovieLens is run by GroupLens, a research lab at the University of Minnesota. Movielens dataset are provided by GroupLens research organisation which is collected from MovieLens website(https://movielens.org/). Dataset of various sizes are available including stable benchmark datasets. For better results users who have less than 20 ratings have been removed. We used MovieLens 1M and MovieLens 100K dataset for out model.TheMovieLens 1M dataset contains 1,000,000 rating given by 6040 users on 3706 movies and MovieLens 100K dataset contains 100,00 ratings given by 943 users on 1682 movies.

Dataset	Users	Movies	Ratings	Density	Year
MovieLens 1M	6040	3706	1,000,000	4.47%	2000-03
MovieLens 100K	943	1682	100,000	6.30%	1997-98

Table.1: MovieLens dataset statistics

3.2 Experimental results

We trained our Deep Neural Network Model with MovieLens 1M dataset and MovieLens 100K dataset. We varied many parameters like number of epochs, batch size and learning rate and tried out neural network with varying number nodes in hidden layer to observe performance of our model which discussed below.

We trained our deep neural network with varying numbers of nodes in hidden layers and found that moderate number nodes in hidden gives better results. Having lower or higher nodes in neural network lead to poor performance.

Shape	Training loss	Validation loss
(64,5,1)	0.8599	0.8806
(64,5,1)	0.8705	0.8864
(64,5,1)	0.8705	0.8807
(128,5,1)	0.8643	0.8815
(128,5,1)	0.8585	0.8803
(128,5,1)	0.8634	0.881
(256,5,1)	0.8624	0.8805
(256,5,1)	0.8582	0.879
(256,5,1)	0.857	0.8794
(512,5,1)	0.8582	0.8805
(512,5,1)	0.8559	0.879
(512,5,1)	0.859	0.8796
(1024,5,1)	0.8589	0.8788
(1024,5,1)	0.858	0.8786
(1024,5,1)	0.854	0.8796
(2048,5,1)	0.8546	0.8794
(2048,5,1)	0.858	0.8799
(2048,5,1)	0.859	0.8797
(4096,5,1)	0.8605	0.881
(4096,5,1)	0.8581	0.8811
(4096,5,1)	0.8554	0.879

Table.2: training and validation loss for different shapes of DNN model

We also tried various batch size and number of epochs for our neural network model. Increasing batch size resulted in lesser training time but also lead to slight increase in loss values. Increasing number of epochs results in better performance but the model starts to overfit as the performance improvement on training set is significantly larger than on validation set.

Batch size	epochs	Training loss	Validation loss	Difference
64	5	0.8601	0.8766	0.0165
64	10	0.8347	0.8745	0.0398
64	20	0.8027	0.8715	0.0688
128	5	0.8677	0.8794	0.0117
128	10	0.8401	0.8793	0.0392
128	20	0.8016	0.8774	0.0758
256	5	0.8691	0.8814	0.0123
256	10	0.8438	0.8785	0.0347
256	20	0.8073	0.8758	0.0685
512	5	0.8785	0.8878	0.0093
512	10	0.8506	0.8786	0.028
512	20	0.8254	0.8801	0.0547
1024	5	0.8859	0.896	0.0101
1024	10	0.8574	0.879	0.0216
1024	20	0.8235	0.8757	0.0522
2048	5	0.8883	0.8976	0.0093
2048	10	0.8661	0.8827	0.0166
2048	20	0.8411	0.8813	0.0402

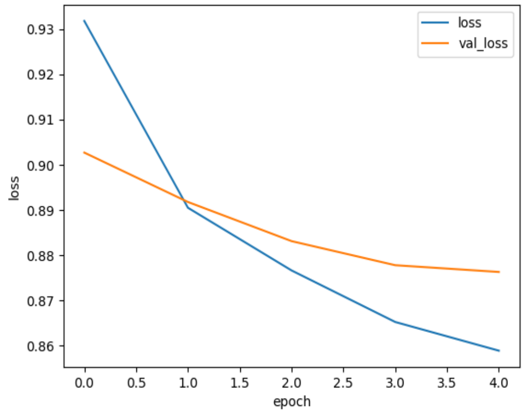
Table.3: training and validation loss for different epoch and batch size of DNN model

Learning rates decide how large steps should be taken in order to reach optimal value for loss function. If learning rate is too high it never reaches the optimal values and if it is too low it takes too long to reach optimal value. These problems are knowns as exploding gradient and vanishing gradient problems. By varying learning rate for our neural network model we observed that lower learning rate resulted in better performance but after lowering value further and further, loss values increased significantly. We found value of 0.0005 to work best for our model and dataset.

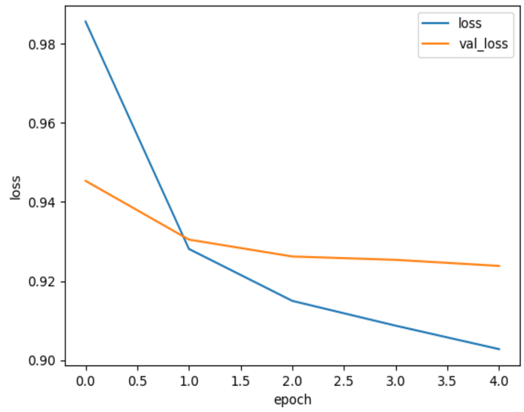
Learningrate	Trainingloss	Validationloss
0.005	0.8277	0.8755
0.005	0.8275	0.8765
0.005	0.8285	0.8763
0.001	0.8458	0.8785
0.001	0.8496	0.8796
0.001	0.8486	0.8797
0.0005	0.8559	0.8786
0.0005	0.858	0.8799
0.0005	0.8573	0.8796
0.0001	0.8927	0.9032
0.0001	0.894	0.9043
0.0001	0.8953	0.9053

Table.4: training and validation loss for different shapes of DNN model

Fig.3 shows plot of RMSE loss against epochs. After tuning our model to best parameters, we plotted the RMSE loss on MovieLens 1M dataset and MovieLens 100K dataset. RMSE is regression loss function which is negatively oriented meaning lower the value the better it is. We can see that model is performing well as the RMSE value is decreasing steadily with every epoch. Table.4 compares RMSE of our Deep Neural Network model with other methods. As seen Table.1 our Deep Neural Network outperforms traditional methodologies as neural networks more efficient with sparse data and can learn features about users and movies easily.



(a) MovieLens 1M dataset



(b)MovieLens 100K dataset

Fig.3: RMSE for MovieLens dataset

Method	MovieLens 100k	MovieLens 1M
User-Based CF	0.937	0.915
Item-Based CF	0.932	0.901
SVD	0.940	0.893
Biased-SVD	0.926	0.887
RBM	0.953	0.918
Autoencoder	0.939	0.892
Stacked Autoencoder	0.933	0.890
Deep Neural Network (Proposed)	0.923	0.876

Table.5: Comparison of RMSE values for MovieLens datasets

V. Conclusion

In this paper we presented a Deep Neural Network based Recommender System. We used dataset made available GroupLens. The MovieLens datasets contain ratings given by users to the movies. The user item matrix representing ratings given by users to the movies is sparse as most of the users have not rated great number of movies. To deal with sparse categorical variables in dataset we used embedding layer. Neural networks are great to deal with sparse dataset and learn hidden features of dataset. We experimented our model with MovieLens 1M dataset and MovieLens 100K dataset. We used RMSE loss function and Adam optimizer to train our model on MoviLens dataset. We tuned our model to optimize the value RMSE loss function. We used 80% of dataset to train our neural network model and 20% of data to test our model. We compared results of our model to existing methodologies and found that it outperforms traditional methodologies and is comparable with state of art algorithms. With results of our Deep Neural Network based recommender systems we conclude that use of deep learning in recommender systems is very promising and is potential topic for future research. With more data about users and movies scrapped from internet, neural network models can learn more about users and movies and thus can result in very sophisticated model that can recommend very relevant movies.

References

[1] Liu, Juntao& Wu, Caihua. (2017). Deep Learning Based Recommendation: A Survey. 451-458. 10.1007/978-981-10-4154-9_52.

[2] Zhang, S., Yao, L., & Sun, A. (2017). Deep learning basedrecommender system: A survey and new perspectives.arXiv preprint arXiv:1707.07435

[3] Yang, Shuo&Korayem, Mohammed &Aljadda, Khalifeh & Grainger, Trey &Natarajan,Sriraam. (2017). Combining Content-based and Collaborative Filtering for Job Recommendation System: A Cost-sensitive Statistical Relational Learning Approach. Knowledge-Based Systems. 10.1016/j.knosys.2017.08.017

[4] Li Deng, Dong Yu, et al. 2014. Deep learning : methods and applications. Foundations and Trends® in Signal Processing 7, 3–4 (2014), 197–387.

[5] Ian Goodfellow,YoshuaBengio, and Aaron Courville. 2016. Deep Learning. MIT Press. h. p://www.deeplearningbook.org

[6] Lei zheng. 2016. A survey and critique of deep learning on recommender systems. Department of computer science

[7] Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Anil, R. (2016, September). Wide & deep learning for recommender systems. InProceedings of the 1st Workshop on Deep Learning for Recommender Systems(pp. 7-10). ACM.

[8] Van den Oord, A., Dieleman, S., &Schrauwen, B. (2013). Deep content-based music recommendation. InAdvances in neural information processing systems(pp. 2643-2651)

[9] Fakhfakh R., Ben Ammar A. & Ben AmarC., Deep Learning-Based Recommendation: Cur-rent Issues and Challenges, International Jour-nal of Advanced Computer Science and Applica-tions(IJACSA), 8(12), 2017.

[10] Adomavicius, G., and Tuzhilin, A. "Recommendation Technologies: Survey of Current Methods and Possible Extensions," MISRC Working Paper 03-29, MIS Research Center, University of Minnesota, Minneapolis, MN, 2003.

[11] Hyeungill Lee, JungwooLee,Scalable deep learning-based recommendation systems,ICT Express,2018,,ISSN 2405-9595,https://doi.org/10.1016/j.icte.2018.05.003.