International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Deep Neural Networks for Recommender Systems

Bhakti Ahirwadkar, Sachin N. Deshmukh

Abstract. ***The data available online, helps users to get information about anything of his/her interest. But since the data is huge and complex it is difficult to get useful information from it. Recommender Systems are effective software techniques to overcome this problem. Based on the user's and item's information available, these techniques provide recommendations to users in their area of interest. Recommender systems have wide applications like providing suggestive list of items to customers for online shopping, recommending articles or books for online reading, movie or music recommendations, news recommendations etc. In this paper we provide a study of Deep Neural Networks (DNN) approaches that can be used for recommender systems. They have been used widely in last decade in many fields like image processing, video streaming, Natural Language Processing etc. including recommendations to overcome the drawbacks of traditional systems. The paper also provides performance of Denoising AutoEncoders (DAE) which are feed forward neural networks and its comparison with traditional systems. Denoising Autoencoders are a type of autoencoders wherein some part of input is corrupted, i.e., noise is added to the input. While learning to remove noise from input, the DAE also learns to predict unknown values. This property of Denoising Autoencoders can help in recommendation systems to predict unknown values before recommending new items. Experimentation has shown improvement in the performance of recommendation systems with denoising autoencoders. The evaluation is performed on MovieLens-1M dataset with and without additional features of users (age and gender) and items (movie genres) provided in the dataset.***

Keywords: Recommender Systems, Collaborative Filtering, Deep Neural Network, Autoencoders, Convolutional Neural Networks, Multi-Layer Perceptron, Denoising AutoEncoders.

I. INTRODUCTION

The ever increasing use of internet, is leading to volumes of data available online which can be used by users to get information about things of interest. Filtering information from such voluminous and complicated data is getting difficult day by day. This is where Recommender Systems (RS) play role to help users to filter information through such data. RS provide suggestions about various interesting items to the users. They can be used in many areas like entertainment, E-Commerce and service industry.

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author
Bhakti Ahirwadkar*, Department of Computer Science and Engineering, Marathwada Institute of Technology, Aurangabad, Maharashtra, INDIA. Email: bhaktipr@gmail.com
Sachin N. Deshmukh, Department of Computer Science and IT, Dr Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, INDIA. Email: sndeshmukh@yahoo.com

Traditional Recommender Systems are software that use existing ratings provided by the users to various items to predict ratings of unrated items and suggest new items to users. They are basically classified into Content Based Recommender Systems, Collaborative Filtering Recommender Systems and Hybrid Systems.

Content based RS uses rating history of the user and features of items for predicting ratings and providing recommendations [1]. Collaborative filtering systems on the other hand use ratings of other users with similar likings as that of the current user. They are further classified into: neighborhood approaches and model based approaches.

Neighborhood based techniques are User Based CF, that computes similarity between users and Item Based CF, that computes similarity between items [1], [2]. The ratings of similar users or items are then users for predicting missing ratings of current user by using weighted sum. Neighborhood based techniques may not perform well as the number of users increases. It also cannot provide recommendations for new users or suggest new items, with no ratings, to the users. This is called cold start problem.

Model based techniques address the scalability and cold start problems of neighborhood based techniques. These techniques are Bayesian models, dependency networks clustering [4], [5], [6] and latent factor models that can be used to learn complex relations from the data. Latent factor models based on matrix factorization [3] have been used successfully for CF and have shown improvements in results [7], [8], [9]. These models and linear and fail learn important relations. In recent years deep neural networks have shown promising results in the field of recommender systems where non-linear functions are used for learning complex relations between the data. In this paper we attempt to give insights of Deep Neural Network approaches for RS and evaluate the performance of Denoising Autoencoders for MovieLens-1M dataset by providing additional features to the system.

II. DEEP LEARNING TECHNIQUES

Deep Learning is a sub field of Machine Learning which can be trained by supervised, semi-supervised or unsupervised approaches. They are also called as hierarchical learning since the learning process consists of several layers of non-linear processing units that try to learn complex relations from the input data. In recent years they have shown promising results in many areas like image processing, voice recognition, natural language processing, recommendation systems etc. Deep neural based recommender systems can be broadly classified into [10]:

- Recommender systems that generate predictions using only deep learning approaches.

- Recommender systems that generate predictions by integrating traditional recommenders with deep learning approaches.

A. Deep Learning Approaches for Recommender Systems:

Mostly commonly used deep learning approaches for recommender systems are:

Convolutional Neural Network (CNN)

This feed forward neural network [11] which is more commonly used in computer vision and natural language processing uses a convolution layer that generates local features and a pooling layer for concise representation. It has been used for recommender systems in recent years.

Multi-Layer Perceptron (MLP)

It is a feed forward neural network consisting of input layer, hidden layer and output layer. All nodes except input nodes use non-linear activation function and uses back propagation for training.

Autoencoders (AE)

Autoencoders are unsupervised neural networks that learn to reconstruct the input fed to the network. There are basically three layers: input layer, bottle neck layer and the output layer. In recent years AE have been used widely in recommender systems.

Recurrent Neural Networks (RNN)

RNNs are used for learning sequential data [15]. Many implementations show use of RNN in natural language processing and speech recognition. RNN are feed forward neural networks that can remember useful information of a sequence through time. This original model is Long Short Term Memory (LSTM).

III. RELATED WORK

A visual recommendation system [12] obtains style features of items from CNN through images of items. These features are then used with Bayesian Personalized Ranking framework for generating recommendations. Performance is measured using AUC on Amazon dataset. A context aware recommender system proposed [11] tightly couples CNN with Probabilistic Matrix Factorization to capture contextual information of documents for improving rating prediction accuracy. The model performance is measured on MovieLens and Amazon datasets. Convolutional Neural Networks without integration with traditional systems [13] for hashtag recommendations show improvement in results as compared to existing systems. Article recommendation [14] uses Dynamic Attention Deep Model and CNN to learn article representation to catch selection behavior of editors. Component level and item level attention neural networks with BPR and SVD++ [16] are implemented for multimedia recommendations. Item Based and User Based Recommender with AutoEncoder [17] show improvement in performance as compared to traditional techniques. Denoising Autoencoders (DAE) implemented for movie recommendations [18] are a type of Autoencoders to which corrupted input is given and the model is trained to reconstruct the input by reducing the noise. Stacking of such networks, show improvement in results. Additional features are used for addressing the cold start problem. Other implementations [19], [20] use AEs for deep representation learning ratings with additional information and Matrix Factorization techniques for collaborative filtering. Harald Soh et. al. [21] propose an architecture for personalized AUI that use deep learning, partially Gated Recurrent Units to learn user interaction patterns and collaborative filtering techniques that enable sharing of data among users.

IV. DENOISING AUTOENCODERS

Autoencoders are unsupervised neural networks that are trained to compress data received from input layer into a short code, and then uncompress that code into data as close as possible to the input. A simple Autoencoder is a two layer neural network [23]. Given an input $x \in R^D$, it first encodes by mapping it to hidden representation:

$$h = a_1(W_1 x + b_1) \quad (1)$$

where $W_1 \in R^{kxD}$ is the weight matrix and $b1 \in R^k$ is the bias and a_1 is non-linear activation function. The hidden layer also called the bottleneck usually has dimension less than the input. The encodings are then decoded to reconstruct the input at the output layer.

$$y = a_2(W_2 h + b_2) \quad (2)$$

where $W_2 \in R^{Dxk}$ is the weight matrix, $b_2 \in R^D$ is the bias vector and a_2 is the non-linear activation function for the output layer. The Autoencoders are then trained with back-propagation with the aim to minimize the reconstruction loss. For this the squared loss is back-propagated:

$$\mathscr{L}(x, y) = ||(x - y)||^2 \quad (3)$$

For optimization, algorithms like Stochastic Gradient Descent, Adam's, RMSProp and AdaGrad can be used.

A type of Autoencoders is Denoising Autoencoders (DAE) where noise is added to some fraction of input. The model is trained to remove noise and reconstruct the input. This is done so that the network does not turn into an identity network. A portion of input ratings can be zeroed for corrupting the input. The Denoising Autoencoder then can be trained to impute the missing values.

V. METHODOLOGY

The system uses Denoising Autoencoder (DAE) for evaluating the performance on MovieLens-1M dataset and comparing the results with traditional recommender systems. The input is corrupted by zeroing out some percentage of the input. The Autoencoder uses 2 encoder layers and 2 decoder layers. L_2 regularizer is used at these layers for regularizing the weights. We use two approaches: user based that uses user vectors as input and additional features of users and second that uses item vectors as input and additional item features. The objective is to reduce the loss:

$$\sqrt{||x - a_2(a_1(Wx' + b_1) + b_2)||^2 + \lambda/2||W||^2} \quad (4)$$

where a_1, and a_2 are the activation functions (ReLu in this experimentation), b_1 and b_2 are the bias. λ is the regularization strength for L_2 regularizer. The loss function id the RMSE of the output of the DAE, $a_2(a_1(Wx' + b_1) + b_2)$ and the original value, x. The L_2 regularizer adds a penalty, $\lambda/2||W||^2$, to the loss function. This reduces overfitting.

VI. EXPERIMENTAL RESULTS

A. Dataset

GroupLens Research provides many dataset which have been collected over different periods of time. MovieLens is a dataset of ratings [22]. MovieLens-1M dataset made available through MovieLens website is used for evaluating the results. This dataset contains approximately 3900 movies which have been rated by 6040 users and each user has rated at least 20 movies.

B. Results

For the DAE model proposed in the paper various hyper parameter values have been tested and the performance has been evaluated for the MovieLens-1M dataset. The dataset has been split into train test and validation splits. Experimentation has been conducted for train test and validation split of 80%-10%-10% and 60%-20%-20%. The percentage of corruption for input is set to 40%. The activation function used for the encoders as well as the decoders is Rectified Linear Units (ReLu). The system uses additional features: age and gender for users. For movies, it is the genres of the movies available in the dataset. After experimenting for different values of number of nodes at each hidden layer, the nodes for first hidden layer is set to 512, for second layer it is 256. The regularization strength for L_2 regularizer was tested for λ = 1, 0.5, 0.2, 0.3, 0.1, 0.01, 0.001 and was set to 0.001 for weights regularization.
Table 1 shows experimental results (RMSE on test set) on MovieLensLens-1M dataset for: traditional collaborative filtering algorithms and Denoising AutoEncoder (DAE) with above mentioned settings on test set and 80%-10%-10% dataset split (First the dataset is split into 90%-10% train-test data and the train set is further divided into 90%-10% train-validation split for performance tuning). For DAE, the RMSE is evaluated with and without additional features. Table 1 also shows experimental results for 60%-20%-20% dataset split.

The deep neural network model approach with DAE outperforms the traditional approaches. DAE for user-based CF with user features performs better than DAE without features and without noise. For item-based CF, performance of model was better with item features looking at the training graph and the root-mean squared error. The training graphs for both UBCF and IBCF without adding noise show underfitting.

The performance of the model was evaluated with and without L_2 regularizer. Figure 1 shows performance of the model without regularization. After few epochs the model over fits as the training loss decreases but the validation loss increases.

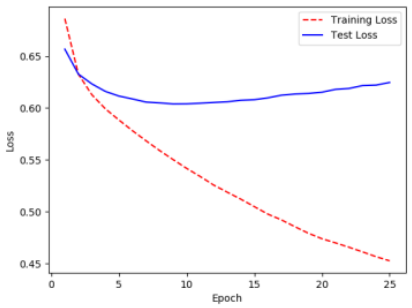

Figure 1: Performance of DAE without regularization

To reduce the overfitting of the model, L_2 regularizer was used. Figure 2 and Figure 3 show performance of DAE with L_2 regularizer for user-based CF with and without user features respectively using 80%-10%-10% dataset split. Figure 4 shows the performance using 60%-20%-20% split for user-based CF with features.

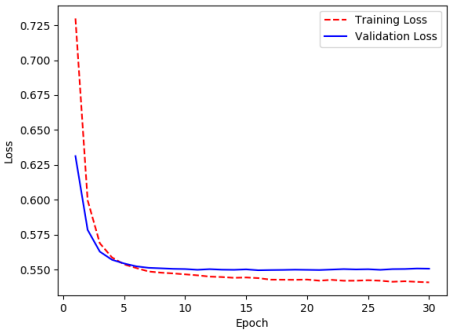

Figure 2: Performance of DAE for UBCF With User Features for 80%-10%-10% Split

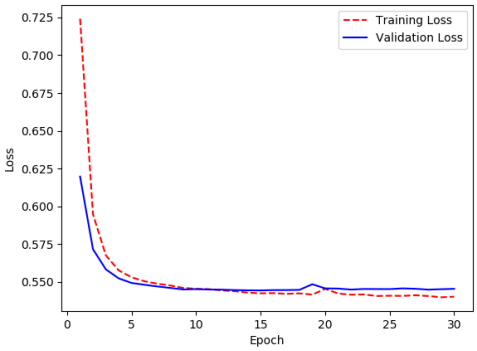

Figure 3: Performance of DAE for UBCF Without User Features for 80%-10%-10% Split

Table 1: RMSE for Traditional Algorithms and Denoising Autoencoder with L_2 Regularizer and 40% noise for 80%-10%-10% and 60%-20%-20% Dataset Split

Algorithm	MovieLens-1M (80%-10%-10% dataset split)	MovieLens-1M (60%-20%-20% dataset split)
Collaborative Filtering - Item Based	0.95	0.99
Collaborative Filtering – User Based	0.97	0.98
Matrix Factorization with SVD	0.85	0.85
DAE for UBCF with user features & 40% noise	0.527	0.532
DAE for UBCF without user features & 40% noise	0.537	0.539
DAE for UBCF without user features & without noise	0.585	0.604
DAE for IBCF with item features & 40% noise	0.515	0.557
DAE for IBCF without item features & 40% noise	0.478	0.522
DAE for IBCF without item features & without noise	0.517	0.632

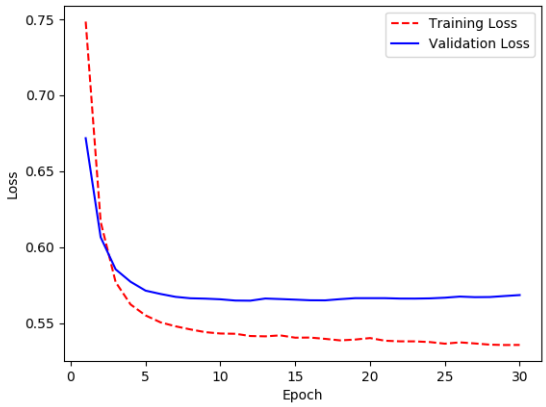

Figure 4: Performance of DAE for UBCF with User Features for 60%-20%-20% Split

Initially the training loss is much more than the validation loss. After training the model for certain epochs the difference reduced. The learning was stopped after 30 epochs since the validation loss started to increase gradually from this point. For item-based CF, Figure 5 and Figure 6 show performance of DAE with and without item features using 80%-10%-10% split.

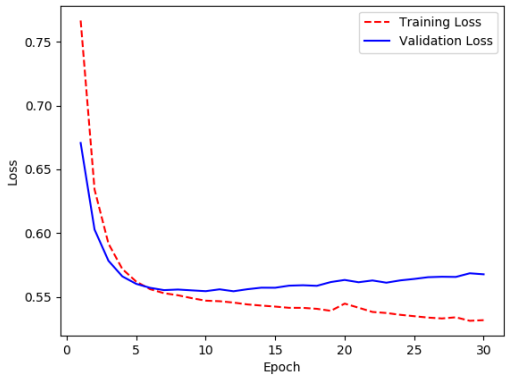

Figure 5: Performance of DAE for IBCF With Item Features for 80%-10%-10% Split

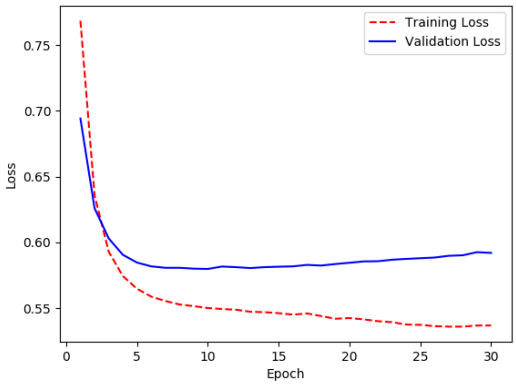

Figure 6: Performance of DAE for IBCF Without Item Features for 80%-10%-10% Split

For improvement in performance, experimentation on various train-test-validation splits of dataset should be carried out. If the model overfits, it should be regularized with any of the regularization techniques like L_1, L_2, dropout and data augmentation. Early stopping can also be used where the training is stopped at the point where the validation loss starts increasing. L_1 and L_2 regularizers are used commonly. The regularization strength is a hyper parameter which can be set after testing for various values on the dataset. We have set it to 0.001 after experimentation. The data split ratio, regularization technique and hyper parameters like the learning rate, regularization strength, size and number of hidden units should be set by evaluating the performances for various values and set to the values that best suit the model on considered dataset. The performance can be checked by plotting the train and validation loss against epochs and the average training/validation loss. Too much of difference between train and validation loss is not expected. If the train loss is very less than the validation loss then the model overfits. If the validation loss is very less than the train loss then the model under fits. In such cases regularization helps for stabilizing the model.

VII. CONCLUSIONS

Deep learning techniques can be used for improving the performance of recommender systems and for overcoming the drawbacks of traditional systems.

Denoising autoencoders can perform better as adding noise can make the algorithm predict unknown ratings. Though there is not much performance gain with and without features, use of different set of side information, hyper parameters and regularization techniques can be thought of for addressing the cold start problem and improving the performance of the model. This can reduce the test error and stabilize the model further by improving the training; reducing the difference between training and validation loss which can be monitored through the training graph.

REFERENCES

1. Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, "Collaborative Filtering Recommender Systems", Foundations and Trends in Human–Computer Interaction Vol. 4, No. 2 (2010) 81–173
2. Yongli Ren, Gang Li, Jun Zhang and Wanlei Zhou, "The Efficient Imputation Method for Neighborhood-based Collaborative Filtering", CIKM'12, October 29–November 2, 2012, ACM
3. Dheeraj Bokde, Sheetal Girase and Debajyoti Mukhopadhyay, "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey", Procedia Computer Science 49(2015) 136-146
4. Nitin Pradeep Kumar and Zhenzhen Fan "Hybrid User-Item Based Collaborative Filtering", Procedia Computer Science, 2015
5. Yao Wu, Xudong Liu, Min Xie, Martin Ester and Qing Yang, "CCCF: Improving Collaborative Filtering via Scalable User-Item Co-Clustering", WSDM'16
6. Li Zhang, Tao Qin and PiQiang Teng, "An Improved Collaborative Filtering Algorithm Based on User Interest", JOURNAL OF SOFTWARE, VOL. 9, NO. 4, APRIL 2014
7. Ruslan Salakhutdinov Andriy Mnih, "Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo", International Conference on Machine Learning, 2008
8. Daniel D. Lee and H. Sebastian Seung, "Algorithms for Non-negative Matrix Factorization", NIPS 2000
9. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", CSCW 94- 10/94
10. Shuai Zhang, Lina Yao and Aixin Sun, "Deep Learning Based Recommender System: A Survey and New Perspectives", ACM J. Comput. Cult. Herit., Vol. 1, No. 1, Article 35
11. Dongghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee and Hwanjo Yu, "Convolutional Matrix Factorization for Document Context- Aware Recommendation", Recsys'16, September 15-19, 2016, ACM
12. Qiang Liu, Shu Wu and Liang Wang, "DeepStyle: Learning User Preferences for Visual Recommendations", SIGIR, ACM 2017
13. Yuyun Gong and Qi Zhang, "Hashtag Recommendation Using Attention-Based Convolutional Neural Networks", Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI-16)
14. Xuejian Wang, Lantao Yu, Kan Ren, Guanya Tao, Weinan Zhang, Yong Yu and Jun Wang, "Dynamic Attention Deep Model for Article Recommendation by Learning Human Editor's Demonstration", KDD 2017, ACM
15. Robin Devooght and Hugues Bersini, "Collaborative Filtering with Recurrent Neural Networks", arXiv, 3rd Jan 2017
16. Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu and Tat-Seng Chua, "Attentive Collaborative Filtering: Multimedia Recommendation with Item and Component – Level Attention", SIGIR 2017, ACM
17. Suhas Sedhain, Aditya Krishna Menon, Scott Sanner and Lexing Xie, "AutoRec: AutoEncoders meet collaborative filtering", WWW 2015 ACM
18. Shuai Zhang, Lina Yao, Xiwei Xu, Sen Wang and Liming Zhu, "Hybrid Collaborative Recommendation via Semi-AutoEncoder", 4th International Conference, ICONIP 2017
19. Xin Dong, Lei, Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan and Fangxi Zhang, "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems", Thirty first AAAI Conference on Artificial Intelligence, 2017
20. Sheng Li, Jaya Kawale and Yun Fu, "Deep Collaborative Filtering via Marginalized Denoising Auto-encoder", CIKM 15, ACM
21. Harald Soh, Scott Sanner, Madeleine White and Greg Jamieson, "Deep Sequential Recommendation for Personalized Adaptive User Interfaces", IUI 2017, ACM
22. https://grouplens.org/datasets/movielens/
23. Yao Wu, C hristopher DuBois, Alice X. Zheng, Martin Ester, "Collaborative Denoising Auto-Encoders for Top-N Recommender Systems", WSDM'16 ACM

AUTHORS PROFILE

Bhakti Ahirwadkar is currently working as Associate Professor in the Department of Computer Science and Engineering, Marathwada Institute of Technology, Aurangabad, Maharashtra and is pursuing Ph.D. in the Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra. She is having experience of around 18 years in teaching graduate and post graduate courses. Her area of research is Machine Learning and Artificial Intelligence.

Sachin N Deshmukh is currently working as Professor in Department of Computer Science and IT, Dr Babasaheb Ambedkar Marathwada University, Aurangabad and having experience of around twenty four years in teaching for post graduate and graduate courses. He has published more than 80 Research papers in National and International reputed journals and conferences. His area of research is Text Mining, Social Media Data Analytics, Sentiment Analysis, Opinion Mining and Intension Mining. He is also involved in the approval and accreditation of higher education Institute.