IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 24-29

Hybrid Movie Recommender Using Deep Learning: A Review

Saswata Bhattacharya[1], Mitesh Vishwasrao[2], Abhijeet Sonar[3], Prof.Jyothi Arun[4]
[1, 2, 3, 4]*(Department, of Information Technology, Atharva College of Engineering Malad (W) Mumbai)*

Abstract: *A common dilemma of our routine life while relaxing is to decide what content to watch. Sitting in front of TV seems like a futile exercise with no control and no remembrance of content that we consumed. We prefer a smart platform which understands your mood and preferences and not just run on predefined content. We will develop recommender system that recommends movies similar to a given input movie. To create the hybrid model, we accumulated the results of an autoencoder which learns content-based movie embeddings from tag data, and a deep entity embedding neural network which learns collaborative-based movie embeddings from ratings data.*
Keywords: *Content Filtering, Collaborative Filtering, Deep Learning, Recommender System*

I. Introduction

At the advent of internet, the users were concentrated only in twelve countries but the scenario has completely changed now. Today's infographic shows that World Wide Web has spread globally over the past few decades [13]. This has led to rapid generation of huge amounts of data. Sifting through the vast collection of data is frustrating and time consuming. In addition the availability of wide variety of options confuses the user and in most of the cases, he is not satisfied with the content he receives in return of his search. With huge content available online it is essential to provide users with personalised content. This is where the need of recommender system comes into picture. Most existing recommender systems use machine learning techniques like matrix factorisation which are not efficient enough. Deep learning has lately been widely proposed for recommendation systems. Due to its high performance, it provides better understanding of user demands. We will be using deep learning approach for developing an efficient recommender system that provides a wide filtering criteria.

A. Content Filtering

Content filtering is built on a user's profile. A profile contains user information and their preferences. Preference is based on user rating for different items. In this process, the items that are already positively rated by the user are compared with those that are unrated and similarities are looked for. Items analogous to the positively rated items will be recommended to the user. Thus based on a user's preferences and behaviour a content filtering recommendation system can be developed by suggesting items appropriate to a user's liking. Such a framework is effective and personalized but suffers from the absence of diversity. The content-based recommendation engine will only suggest items related to categories that the user has interacted with and may never recommend anything in other categories as the user never viewed those before. This problem can be solved by using another form of recommendation algorithm termed as collaborative filtering [3].

B. Collaborative Filtering

In collaborative filtering, the focus is on finding users that share similar interests. If two users have given the same rating to many items, then they are considered to have similar preferences. Such users form a neighbourhood. Thus the system can recommend items that the user hasn't rated before but was positively rated by other users of his/her neighbourhood. Collaborative filtering has basically two approaches [3]:

1. **User Based Approach**

In this approach, the system recommends items that are positively rated by the neighbourhood of a user. Thus even if the user hasn't explored a certain category, he might get recommendations if some item of that category is highly rated by the user's neighbourhood. The user based approach has some drawbacks, the major one being that user's preferences change over time. To overcome the drawbacks, item based approach is used.

2. **Item Based Approach**

In item based filtering, similarities between items instead of between users is calculated as the former is always constant. Focusing on similarities between unchanging objects can produce better results than looking at similarities between people who may have liked something last week and something totally different this week.

Another advantage is that we usually have far fewer items to deal with than people. Thus, the similarity between items based on the ratings of users is used for making recommendations.

C. Deep Learning

Artificial Intelligence is the ability of a machine to think like humans and is achieved by understanding the problem solving mechanism of the human brain. Deep learning is a subset of machine learning and machine learning is a subset of artificial intelligence.Machine learning gives computers the ability to learn without having to be programmed minutely. It is not required to define clearly all the steps or conditions like any other programming application. Rather, the machine gets trained on a dataset and takes decisions based on its learning.Machine learning has certain drawbacks. It is inept in handling high dimensional data. High dimensional data is found in cases like image processing, NLP, image translation to name a few. The inability to effectively solve such cases led to the evolution of deep learning. Deep learning is capable of handling high dimensional data and is also efficient in automatic feature extraction.Deep Learning studies the basic unit of a brain called a brain cell or a neuron. Inspired from a neuron an artificial neuron or a perceptron was developed. If we focus on the structure of a biological neuron, it has dendrites which is used to receive inputs. These inputs are summed in the cell body and using the Axon it is passed on to the next biological neuron as shown in the above image. Similarly, a perceptron receives multiple inputs, applies various transformations and functions and provides an output. As our brain consists of multiple connected neurons called neural network, we can also have a network of artificial neurons called perceptrons to form a deep neural network [14]. The deep neural network consists of three layers:

1. **Input Nodes –** The input nodes provide information from the outside world to the neural network and are collectively called "Input Layer". Computation is not performed in the input nodes. They simply pass on the information to the hidden nodes.
2. **Hidden Nodes –** The hidden nodes are not connected with the outside world and hence are termed hidden. They perform computations and transfer information from the input nodes to the output nodes. A collection of hidden nodes forms a "Hidden Layer".
3. **Output Nodes –** The output nodes are collectively referred to as the "Output Layer". They perform computations and transfer information from the neural network to the outside world [12].

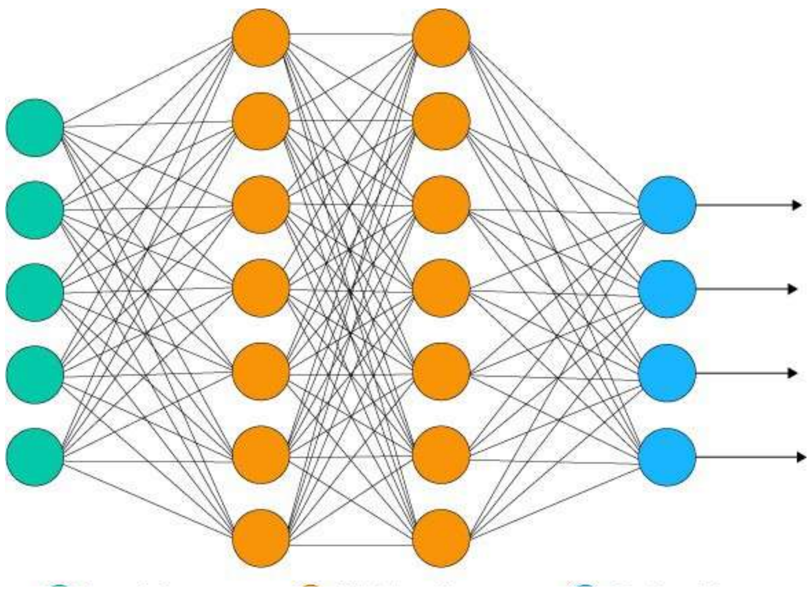

Figure 1: Deep Learning Architecture [15].

II. Literature Review

In paper [1] the author has tried to create a deep learning framework for a movie recommendation system. The authors have implemented collaborative filtering using autoencoders. Two N-Dimensional vectors are given as input to the autoencoder. The first vector describes the user profile. Each dimension of the vector is the rating given by the user to a movie and zero if the movie is unrated. The second vector is a one hot encoding of a particular movie. With this two vectors as an input, the neural network is able to predict the rating of a movie. The efficiency of the system was measured on the basis of root mean square error and the system performed better than existing systems. In order to minimise prediction errors, regularisation methods were applied. The system developed easily surpassed existing systems using traditional machine learning techniques, in efficiency.

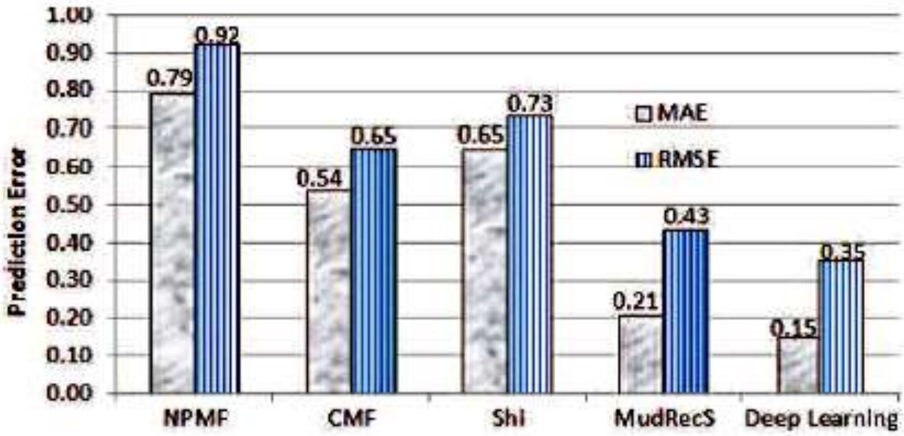

Figure 2: The MAE and RMSE scores for various movie recommendation systems based on the MovieLens dataset [1]

In paper [2] the authors have proposed a recommender system using traditional machine learning methods by combining the asymmetric method of calculating similarity with matrix factorization and Tyco (typicality-based collaborative filtering). After setting up the MovieLens dataset, user item typicality matrix, user-user similarity matrix and gradient descent matrix was calculated. Linear regression was applied on the above three matrices and ratings were predicted, thus enabling the system to recommend movies to users.

In paper [3] the authors have proposed an algorithm that takes into considerations the tags and genres specified in the dataset. Using a set matching comparator the common objects(tags and genres) between two movies are found. For every movie, the tags and genres are combined into a single set. The weight of each set for a movie is calculated. Once the weights are calculated for each set, they are then used to predict the ratings of the unrated movies using the rated movies. The length of the set thus formed is used to predict the ratings using the formula,

$R = M * (H_r / M')$ where, R is the rating for an active movie, M indicates the number of common objects, M' is the maximum number of matching objects between any two movies in the dataset and highest_rating (Hr) is the maximum rating that we can assign to a movie, which in our case is 5. If the rating is greater than 2.5(a threshold value which is equal to the average of the lowest and the highest rating possible), then collaborative filtering is applied.

In paper [4] the authors have proposed user-centred framework that incorporates the content attributes of rated movies (for each user) into a Dirichlet Process Mixture Model to deduce user preferences and build a recommendation model. The rating record fetched from the user item rating matrix is used to create a user profile which contains the storyline and genre of movies. The interests of a user are extracted from his profile and a preference vector is formed. The preference vector is used to calculate the probability of a new movie for its suitability in the user preference cluster. Thus the system recommends movies to users based on likelihood of a movie belonging to a cluster.

In paper [5] the authors have proposed a personalized meta-level hybrid recommendation method that combines collaborative and content-based filtering in a hierarchical manner to recommend movies to users. User preferences are gathered from Facebook and an attributed relational graph is created for users and movies, where the edges are vectors. Content based algorithm is first applied. The output from content based filtering is given as input to collaborative filtering algorithm and movies are recommended to users.

In paper [6] the authors have proposed a model that utilizes smartphone browsing history to personalize movie recommendations. The browsing history and movie plot summaries are used to calculate a similarity score. The calculated score is integrated into a latent factor model that computes latent user and item features. This model handles sparsity and cold-start scenarios using user browsing history and recommends movies that are similar to the ones the user liked.

In paper [7] the authors have proposed a framework that applies clustering algorithm on ratings dataset followed by collaborative filtering for developing a movie recommender system. The data set is partitioned on the basis of genre of the movies and it provides the platform for clustering. K-Means clustering is applied on the partitioned data set, to form cluster of movies with the same genre. The user chooses a movie from the clusters obtained and rates it. Based on the rating provided by the user, the slope one algorithm for collaborative filtering is applied on clusters. Thus system is able to recommend movies to users.

In paper [8] the authors have combined the features of item-item collaborative filtering as well as user-user collaborative filtering to make efficient group recommendation by making homogeneous groups. Most of the recommender systems are used by a group of users. In such cases, we need to aggregate each user's preferences in a group. The similarity of likeness among the users in the group need to be found to make the system effective. Authors have combined user-user filtering to make homogeneous groups and item-item filtering for predicting items that are common for most of the users in the group.

In paper [9] the authors have proposed a framework called DeepMovRS. Users and items are mapped to a common latent space. The latent representation is used for retrieving and ranking items. The model consists of two deep neural networks, one for generating top-Ncandidate subset of videos, and the second for ranking results from the output generated by the former. The two neural networks are combined to recommend movies to users.

In paper [10] the authors have proposed a framework consisting of two models Collaborative Filtering Deep Recommender architecture (CFDR) and Collaborative Filtering based Multistage Deep Neural Network architecture (CFMDNN) CFDR and CFMDNN model complex structures of user and item interactions using matrix factorization. The models were built using Keras and K-fold cross-validation for training and evaluation and the Mean Absolute Error (MAE) as the performance metrics. Dropout as a regularization technique was used to prevent overfitting. Densely fully linked feed forward neural network, and Adam optimization algorithm was applied.

Analysing the different systems proposed we have found some issues plaguing recommender systems built using machine learning methodology. Computing the similarity matrices can quickly get unfeasible. Efficient implementation of machine learning techniques depends on an offline phase where the similarity matrix is computed and an online phase where the prediction is made for a given user. The online phase requires less computation cycles but, for a matrix of m users * m items, the offline phase is at least $O(m^2)$ and $O(n^2)$ in space and memory for user-based and item-based methods respectively[11]. Scaling the system becomes impossible because it is computationally expensive to update the whole similarity matrix regularly.Sparsity is also an issue in machine learning techniques. Computing similarity values between users with very few co-rated items is difficult. Also, the cold start problem, where a new user/item with no ratings is added, is extremely difficult to deal with using these methods. These issues occur frequently when designing recommendation systems using traditional machine learning techniques.

III. Proposed Work

We propose to develop a hybrid movie recommender system using deep learning approach.We design a method for effectively suggesting movies for users by using movie rating information and movie metadata information with deep learning technology. The deep learning framework will be built in the following manner:

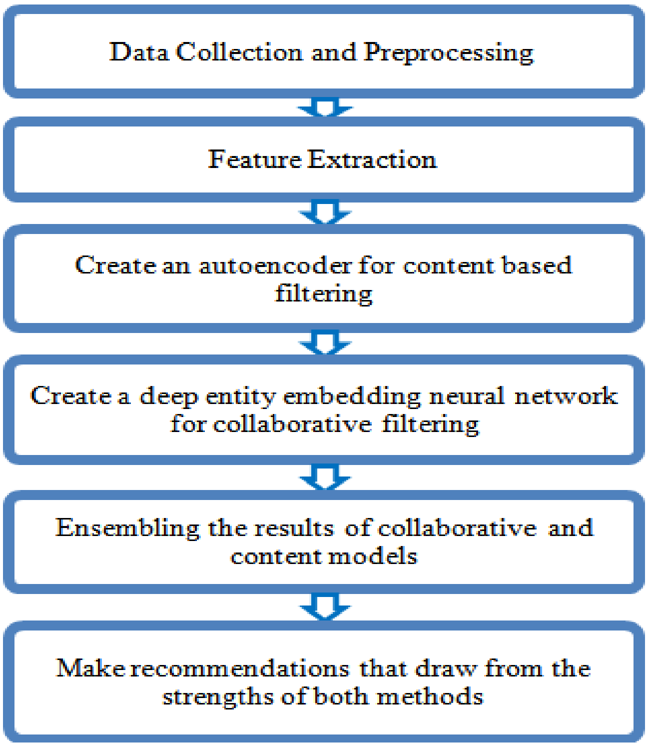

Figure 3: Flowchart of proposed system.

IV. Conclusion

In this paper we have reviewed 10 papers on movie recommendation systems and studied in detail various issues of recommendation system using machine learning techniques. We have analysed the existing systems and found that they are not efficient, scalable and have high RMSE(Root Mean Square Error) and MAE(Mean Absolute Error). The high performance of deep learning, its ability to better understand user preferences and its highly efficient automatic feature extraction mechanism removes the flaws of machine learning models. Also using a hybrid recommender combining content and collaborative filtering overcomes the drawbacks faced by using these filtering methods independently. Hence the hybrid content collaborative recommender using deep learning can ensure good scalability and strength of recommendation systems.

References

[1]. Lund, J., & Ng, Y. (2018). Movie Recommendations Using the Deep Learning Approach. 2018 IEEE International Conference On Information Reuse And Integration (IRI).

[2]. Rahul Katarya, Om Prakash Verma (2016).Effective collaborative movie recommender system using asymmetric user similarity and matrix factorization.International Conference on Computing, Communication and Automation (ICCCA2016).

[3]. Ashish Pal, Prateek Parhi and Manuj Aggarwal(2017), An Improved Content Based Collaborative Filtering Algorithm For Movie Recommendations, Proceedings of 2017 Tenth International Conference on Contemporary Computing (IC3), 10-12 August 2017, Noida, India.

[4]. Bagher Rahimpour Cami, Hamid Hassanpour, Hoda Mashayekhi(2017), A Content-based Movie Recommender System based on Temporal User Preferences, 2017 3rd Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS).

[5]. Ms. Neeharika Immaneni, Ms. Isha Padmanaban, Ms. Bagavathy Ramasubramanian, Mrs. Rajeswari Sridhar(2017), A Meta-Level Hybridization Approach to Personalized Movie Recommendation.

[6]. Dixon Prem Daniel R, Rangaraja P Sundarraj(2017), A Latent Factor Model based Movie Recommender using Smartphone Browsing History.

[7]. Namita Mittal, Richi Nayak, MC Govil, KC Jain(2010), Recommender System Framework using Clustering and Collaborative Filtering, Third International Conference on Emerging Trends in Engineering and Technology.

[8]. Abinash Pujahari, Vineet Padmanabhan(2015), Group Recommender Systems: Combining user-user and item-item Collaborative filtering techniques, 2015 14th International Conference on Information Technology.

[9]. S. M. Taheri, Iman Irajian(2018), DeepMovRS: A unified framework for deep learning-based movie recommender systems, 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS).

[10]. Maram Almaghrabi, Girija Chetty(2018), A Deep Learning Based Collaborative Neural Network Framework for Recommender System, 2018 International Conference on Machine Learning and Data Engineering (iCMLDE).

[11]. →, V. (2019). Recommender systems: Pros and cons of neighborhood-based methods. Retrieved from http://thmsrey-ds.com/2017/07/21/recommender-systems-pros-and-cons-of-neighborhood-based-methods-2/

[12]. A Quick Introduction to Neural Networks. (2019). Retrieved from https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/

[13]. Consent Form | Popular Science. (2019). Retrieved from https://www.popsci.com/science/article/2012-12/widening-world-web-internet-infographic

[14]. Deep Learning Tutorial | AI Using Deep Learning | Edureka. (2019). Retrieved from https://www.edureka.co/blog/deep-learning-tutorial
[15]. When not to use Neural Networks. (2019). Retrieved from https://medium.com/datadriveninvestor/when-not-to-use-neural-networks-89fb50622429