Name: Akansha Jajodia SID: 014489215

**Problem:** Build a recommendation system using deep learning.

Building a movie recommendation system. Given a past record(ratings/reviews) of the movies seen by user, recommend a movie which user will like.

**Environment:** Pytorch 1.3, Python 3.7

**Dataset:** Amazon review (<http://snap.stanford.edu/data/amazon/productGraph/>)

**Approach:** Preprocess the data, build a NCF model.

The data is downloaded from the above link. More specifically we are focusing on Movie and TV reviews subset. The data is available in json format. I have preprocessed the data and extracted the columns which will be required.

The rating column in the dataset has value from 1 to 5. To better train the model I have dropped the exact rating and converted into an implicit feedback as in any positive interaction by user is given a value of 1.

Since userid and itemid data was not consistent for training the model these columns are converted into integer values.

**Model:** Treating this problem as a binary classification model. Hence both positive and negative values are needed. I have used leave one method for implementing this.

**NCF:** Neural Collaborative Filtering is used as an algorithm. It is deep multi-layer perceptron.

 The input to the model is userID and itemID, which is fed into an embedding layer. Thus, each user and item is given an embedding. There are multiple dense layers afterward, followed by a single neuron with a sigmoid activation.

One of the main contributions is the idea that one can replace the matrix factorization with a Neural Network. That way, you can learn an arbitrary function that explains the interaction between users and items.

**Evaluation:** For evaluation metrics Hit ratio and Normalized Discounted Cumulative Gain is used.

The items are sorted in decreasing order of their score, and top 10 items are given as recommendation. If the test item (which is only one for each user) is present in this list, HR is one for this user, else it is zero. The final HR is reported after averaging for all users. A similar calculation is done for NDCG.

The no. of epochs were 30 starting from 0. The best epoch result was obtained at Epoch 28

HR = 0.6013 | NDCG = 0.3294

The results are good. These numbers are obtained from a very coarse hyper-parameter tuning. It might still be possible to extract more juice by hyper-parameter optimization.

A screenshot of text

Description automatically generated

Ref: 1) <https://arxiv.org/pdf/1708.05031.pdf>

2) <https://arxiv.org/pdf/1708.05031.pdf>

3) Prof. KaiKai Liu – Class Lectures and notes

Appendix:

To run the project unzip the folder.

Run python train\_test\_split.py for splitting the data into train and test data

Run python MLP.py to run the model.