

Packages needed

numpy, sklearn, Tensorflow, keras, h5py

Implementation details

The code implements a recommender system based on Multi – Layer Perceptron on MovieLens – small dataset using Tensorflow. Users and item information such as the User_id and the movie_id and movie_ratings are retrieved from the respective data files.

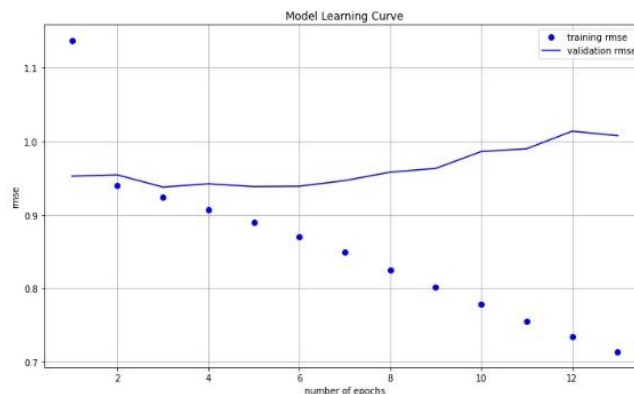
Individual embeddings of the user and item latent features are determined using keras.Embedding (). Input to the embedding are

- Dimension of the input - no. of users / no. of items
- Dimension of the output -half of no. of neurons in layer1 as we concatenate the output of the two embeddings
- Kernel initializer – uniform initializer
- Embedding regularizer – L2 regularizer
- Input length – length of the input sequence to the regularizer, set as 1

The embeddings are concatenated and fed to the multi layer perceptron where the output layer with a single dense neuron predicts the ratings.

The neural architecture input with the combined latent features of users and items promise higher flexibility.

Validation is done across 25% of the data and RMSE is used as the evaluation metric and the graph for a single MLP model looks as follows.



Model configurations:

The recommender system is implemented using different model configurations – size and depth of the MLP layers and the activation functions used in them.

The no. of layers and their size can be modified in the function call `get_MLP_model ()`.

Any non- linear function can be used for activation within the `get_MLP_model ()` by modifying the *activation* parameter to the `Dense ()` function.

Recommendation:

Given an input `user_id`, the list of unwatched movies by the user is determined and the ratings for all unwatched movies are predicted using the generated MLP model. Movies with the highest ratings are recommended to the user.

Base code

https://github.com/KevinLiao159/MyDataSciencePortfolio/blob/master/movie_recommender/movie_recommendation_using_NeuMF.ipynb