Neural Ranking Methods for Anime Recommendation

Alejandro Quiñones[1] and Genaro Laymuns[2]

***Abstract*— In this paper we explore different neural ranking methods that have been used in the last years for several recommender systems, including the state-of-art *Contextual Neural Personalized Ranking*. We show some results of the *Neural Collaborative Filtering* model applied to anime recommendation using a database and compare it to a traditional *Factorization Machines* model. Finally, we propose a model for future works based on *Contextual Neural Personalized Ranking*.**

I. INTRODUCTION

The progressively increasing earnings of the anime industry over the last years has induced a huge increase of the amount of animes produced every year [5]. Just in MyAnimeList, one of the biggest online anime communities, the amount of available animes has increased from $9K$ in 2015 to more than $20K$ in 2018.

The outline is as follows. In section 2, we describe the *Neural Collaborative Filtering* (NCF) method, which only uses information about the *id* of an user and item. In section 3 we show how we can build the current state-of-art *Contextual Neural Personalized Ranking* (cNPR) method from NCF. In section 4 we show the performance of NCF compared to a simple *Factorization Machines* (FM) model when predicting ratings that users assign to certain animes using a database from MyAnimeList published on Kaggle [4]. In section 5 we talk about future work defining the concept of *Double Contextual Neural Personalized Ranking* (2cNPR). Finally in section 6 we show some main conclusions about this work.

II. NCF MODEL

The NCF model [2] receives as input *one-hot* vectors representing certain item and user *id*. These *one-hot* vectors go through a fully-connected layer that transforms it into their own latent vector, which is a dense vector living in a lower dimensional space (the dimension of this resulting vectors is a hyperparameter of the NFC model).

The resulting latent vectors (user and item) are then taken as input for a Neural Network. This gives a certain expected rating for the given user u and item i, $\hat{r}_{ui}$, which is compared to the real rating r_{ui} to train the model. This structure is described in the Figure 1.

This model is therefore a generalization of a simple FM model, since FM directly calculates the rating as the inner product between the latent vectors, without going through a Neural Network.

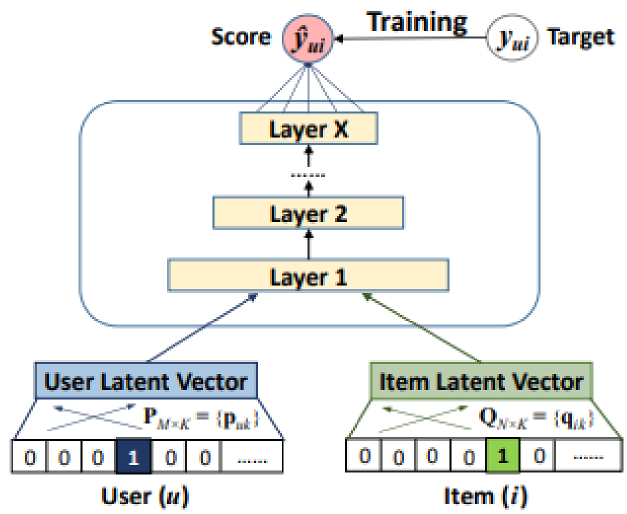

Fig. 1. Basic working scheme for NCF models [2]

In section 4 we show the performance of this method using only one layer in the Neural Network. We take as evaluation measures the *mean-square-error* (MSE), defined as

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in ratings}(\hat{r}_{ui} - r_{ui})^2}{\text{total ratings}}},$$

and the *mean-absolute-error* (MAE), defined as

$$MAE = \frac{\sum_{(u,i)\in ratings}|\hat{r}_{ui} - r_{ui}|}{\text{total ratings}}$$

III. NPR & cNPR MODELS

The NPR model [3] is similar to NCF with one layer, but it receives two items as input for each prediction. It's goal is to predict wether a certain user will prefer item A or B. Two NCF methods are ran simultaneously, and the final result is then substracted to predict the user's preference, i.e., $\hat{r}_{uA} - \hat{r}_{uB} > 0$ implies that the user will prefer item A over item B.

This method still limits to using only the item and user *id*. To take advantage of the extra information that we could have about the item, [3] proposes the cNPR method, which unlike NCF and NPR considers this contextual information (as for the context in which the item is presented). This contextual information is appended to the item latent vector and is taken

[1] Departament of Computational Science, Engineering School, Pontificia Universidad Católica de Chile
[2] Institute for Computational & Mathematical Engineering, Engineering School, Pontificia Universidad Católica de Chile

as input for the Neural Network. As the user latent vector needs to live in the same dimensional space as the item latent vector, the model needs to learn new parameters for the user, which represent the preferences a user may have for each of the item's features.

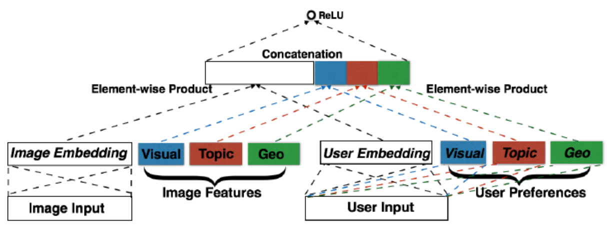

Fig. 2. cNPR working scheme [3]. The items are pictures taken from *Pinterest*, which have three kind of features: pixels (visual features), text (topic features) and location (geographic features). Item's features are taken as input for the model, whilst user preferences are part of the parameters the model adjusts.

IV. DATASET & RESULTS

The dataset we used contained 350056 ratings from 1000 users to 10383 items. That means it had a density of 3%.

Since the dataset contained only *ids* representing each entity, preprocessing was needed in order to obtain their one-hot embedding. Due to the size of the resulting dataset after the preprocessing, we had to train our model through mini-batches since the whole dataset didn't fit in memory. We used mini-batches of size 64.

We splitted our data in 70% training and 30% testing. We used Adam for optimizing and our hyperparameter, representing the dimensionality of our latent representation, was 100. Table I summarizes the results of our implementation of NCF compared with other simpler recommendation models.

Model	RMSE	MAE
NCF	4.6648	4.6680
UKNN	4.7256	3.4445
SVD100	4.530	3.303

TABLE I

ERROR MEASURES FOR DIFFERENT RECOMMENDATION MODELS.

V. FUTURE WORK

cNPR is the current state-of-art in the context of image recommendation. In [3], they limit to consider only extra information about the item, which they call "contextual information". We propose a method which we call *Double Contextual Neural Personalized Ranking* (2cNPR) as an improvement of cNPR. It consists of considering not only contextual information of each item, but also contextual information of each user, appending that information to the user latent vector and creating new learning parameters for the item latent vector related to the kind of users that like that kind of items. This model would need much more training than NCF or cNPR, as it requires to learn parameters both for the user and item.

VI. CONCLUSIONS

In this work we describes several Neural Ranking methods for item recommendation. The main conclusions obtained are:

- The structure of a Neural Network allows flexibility when the training function changes (rating prediction or item preference).
- Some models showed better results than NCF's. This might be due to a small dataset. NCF's first fully-connected layers might be having too little data per user and item in order to come up with a correct latent embedding.
- NCF's structure is easily modifiable to incorporate new contextual information, both for items and users.

REFERENCES

[1] G. O. Young, ÒSynthetic structure of industrial plastics (Book style with paper title and editor),Ó in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15Ð64.
[2] He X., Liao L., Zhang H. (2017). Neural Collaborative Filtering.
[3] Niu W., Caverlee J., Lu H. (2018). Neural Personalized Ranking for Image Recommendation.
[4] MyAnimeList Dataset, Kaggle. Disponible en: https://www.kaggle.com/azathoth42/myanimelist/version/9
[5] Data & Topics of Japanese Animation Industry (2017). Anime Industry Report 2017.