International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

An Experiment of Recommender System Based on Deep Learning

Le Nhat Binh[1], Giao N. Pham[2], Binh A. Nguyen[3], Hanh T. Pham[3], Ngoc T. Le[3], Truong V. Tran[3]*

[1]Faculty of Electrical and Electronics Engineering, Vietnam Aviation Academy, Vietnam

[2]Dept. of Computing Fundamentals, FPT University, Hanoi, Vietnam

[3]ICT Department, FPT University, Hanoi, Vietnam

E-mail: binhln@vaa.edu.vn, giaopn@fe.edu.vn, binhnase04865@fpt.edu.vn, hanhpthe130014@fpt.edu.vn, ngoclthe131028@fpt.edu.vn, truongtvhe130302@fpt.edu.vn

***C**orresponding author

ABSTRACT

Recently, deep learning has been widely applied in many domains in life. In recommender system, deep learning is used to replace the traditional techniques.This paper presents an experiment for a recommender system based on Collaborative Filtering. The Light-FM library is used to extract features for both of User-based and Item-based in Collaborative Filtering recommendation. Experiment is applied on Movie-Lens dataset

Keywords: Light-FM, Collaborative-Filtering, Movie Lens

INTRODUCTION

Collaborative Filtering, on the other hand, does not require any information about the items or the users themselves. It recommends items based on users' past behavior. There are two categories of Collaborative Filtering:

- User–Based: measure the similarity between target users and other users
- Item–Based: measure the similarity between the items that target users' rates

Both of these approaches try to compute the average expression of experienced users on a particular item based on similarity between users/items. In the past, many researchers made an effort to improve the method of finding similarities. Many of them seek to discover latent features.Many deep learning architectures [1] can be used to develop a recommender system such as: Multilayer Perceptron, Auto-Encoder, CNN, RNN, RBM and so on.

Light-FM is a Python library that provides some popular recommender algorithms. It allows incorporating the metadata into the matrix factorization algorithms by represents users and items as the sum of the latent representations of their features [2]. Light-FM provides a deep learning method to learns latent representations in a high-dimensional space for users and itemsin a way that encodes user preferences over items. These representations generate a score for each item for a given user when they multiplied together.

Loss functions are one of the essential parts of a machine learning algorithm; by telling the algorithm what it got right or wrong, they primarily define what it is learning. The Light-FM library use four loss functions:Logistic, Bayesian Personal Ranking (BPR) [3], Weighted Approximate-Rank Pairwise (WARP) [4], K-OS WARP (A modification of WARP).

In addition, we will discuss the effect of learning rate on the improve accuracy. In Light-Fm, two learning rate schedules are available: Adagrad [5] is an algorithm that adjusts the learning rate according to the gradient value of the independent variable in each dimension to eliminate problems caused when a unified learning rate has to adapt to all dimensions. Adagrad constantly adjusts the learning rate during iteration to give each element in the independent variable of the objective function its own learning rate. When using Adagrad, the learning rate of each element in the independent variable decreases (or remains unchanged) during iteration. Adadelta [6] is another common optimization algorithm that helps improve the chances of finding useful solutions at later stages of iteration, which is difficult to do when using the Adagrad algorithm for the same purpose. The interesting thing is that there is no learning rate hyperparameter in the Adadelta algorithm. Adadelta has no learning rate hyperparameter, it uses an EWMA on the squares of elements in the variation of the independent variable to replace the learning rate.

EXPERIMENT

We construct an interaction matrix between users and items with a rating score from source data. All null rating positions in the matrix will be replaced by zero. In the next step, we fit an interaction matrix to Light-FM Model and receive two matrix user embedding (user feature) and item embedding (item feature).We test Light-FM with all of four loss functions and two learning rate schedules. We evaluate the model with Root Mean Squared Error.

The embedding features then passed to traditional collaborative filtering with 10 similar neighbors. Fig. 1 illustrates the whole experimented procedure. After that, we summarize our experiment to show readers how the model works with different hyper-parameters and different datasets.

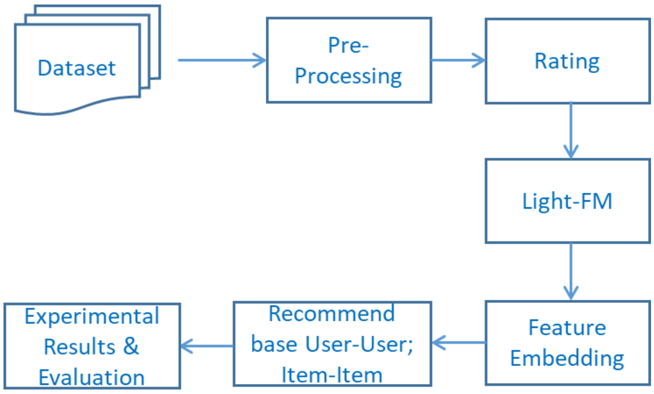

Figure 1: Basic flow of the experiments

We used MovieLens dataset to optimize the model first, which has 101068 records include the interaction of 611 users and 9724 movies to find the maximum performance of the Light-FM.

Firstly, we will build the base model with default parameters (loss = 'warp,' epochs = 100) to compare with the optimized model in the next part. We received the following score:

- Precision at 20: train 0.35, test 0.27.
- Recall at 20: train 0.15, test 0.17.
- AUC: train 0.99, test 0.96.

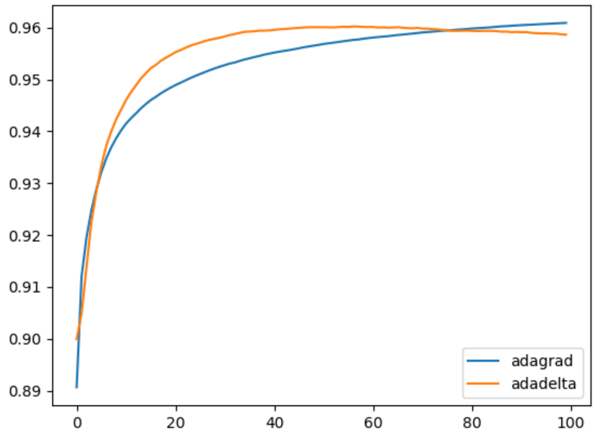

Figure 2: AUC score – Epochs with WARP

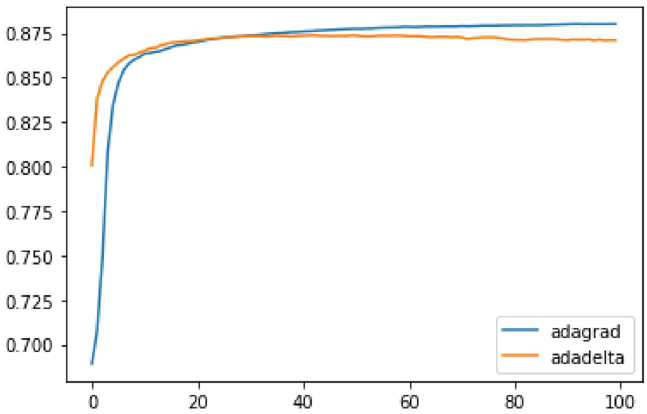

Figure 3: AUC score – Epochs with BPR

Secondly, we choose a learning schedule for the model (Fig. 2 and Fig. 3). Adadelta converges faster than Adagrad at the beginning. However, when we try with a high number of epochs Adagrad gives a much higher AUC score than Adadelta. So that we use the Adagrad learning schedule for our model.Thirdly, we optimize our model using forest minimize algorithm of scikit-learn to minimize test precision score with a space of four parameters : epoch [100 , 250] , learning rate [10^{-4} , 1.0] , number of components [20 , 200] , alpha [10^{-6} , 10^{-1}]. We try 250 times, its cost 3 hours and we received a better score than base model.

- Precision: train 0.65, test 0.35.
- Recall: train 0.23, test 0.20.
- AUC: train 0.99, test 0.92.

We take the model for extract feature for user-user and item-item based algorithms will be used in the next step.

CONCLUSION

This research provides an overview of Light-FM and compares the effectiveness of using the original way used Collaborative Filtering. In the experimentation's results, we can see that the Light-FM model can extract useful feature which improves RMSE score with original way used of Collaborative Filtering. We also build a recommender system with User-based and Item-Based Collaborative Filtering with input are two feature matrix received from the Light-FM.

Acknowledgement

This work is supported by FPT University, Hanoi,
Vietnam; and Faculty of Electrical and Electronics Engineering, Vietnam Aviation Academy, Vietnam.

REFERENCES

[1] S. Zhang, L. Yao, A. Sun, and T. Yi,"Deep Learning based Recommender System: A Survey and New Perspectives. ACM Computing Survey 1, Article 1, 2018.

[2] M. Saveski, and A. Mantrach, "Item cold-start recommendations", in Proc. of the 8th ACM Conference on Recommender Systems, 2014.

[3] Steffen et. al., "BPR: Bayesian personalized ranking from implicit feedback",in Proc. of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2009.

[4] Weston, Jason, Samy Bengio, and Nicolas Usunier, "Wsabie: Scaling up to large vocabulary image annotation." IJCAI. Vol. 11. 2011.

[5] Duchi, John, Elad Hazan, and Y. Singer. "Adaptive subgradient methods for online learning and stochastic optimization." The Journal of Machine Learning Research 12 (2011): 2121-2159.

[6] Zeiler, Matthew D. "ADADELTA: An adaptive learning rate method." arXiv preprint arXiv:1212.5701 (2012).