Report

Choosing a model, formatting and training it

Process of finding a model:

At the very beginning it was needed to check whether there are already existing and proved to be good in making predictions solutions. First and last Internet resource that was found as a source of such solutions is PapersWithCode and in particular one of the pages that is related to Recommendation Systems. There can be easily found models that were tested as solutions for recommending movies on curtain datasets. For MovieLens 100K dataset there are three models that have almost identical RMSEs (approximately 0.88): GHRS, GLocal-K, and MG-GAT. The last one uses extra training data and hence the first two models were considered as a basement for the solution.

Choice making:

It was needed to decide which model to choose as a main part of the solution. Firstly, the main research paper about the GHRS model was red and then the GitHub repository with partial implementation was checked. Secondly, the same process was performed with GLocal-K model(research paper and GitHub repo) and after that some pros and cons were extracted:

1) GHRS has more complex implementation which utilizes an extra data about the users(such as Zip codes and names) and looks like more interesting as a solution
2) GLocal-K has a simple idea that gives almost the same result as GHRS.

To use the time for creating a solution effectively it was decided to use GLocal-K as a main part of it. In addition, think about completing GHRS partial implementation which includes such steps as: reading the README.md and the source code, reading the main research paper, trying to aggregate already existing solutions as absent parts with the partial implementation to get the working model.

Code formatting(refactoring):

After cloning the GitHub repo of GLocal-K adapted for PyTorch it was needed to check the code to ensure that it is readable, well structured, runnable, and in overall gives the same results as in the leaderboard from which it was chosen.

After some time of reading the code it was found that:

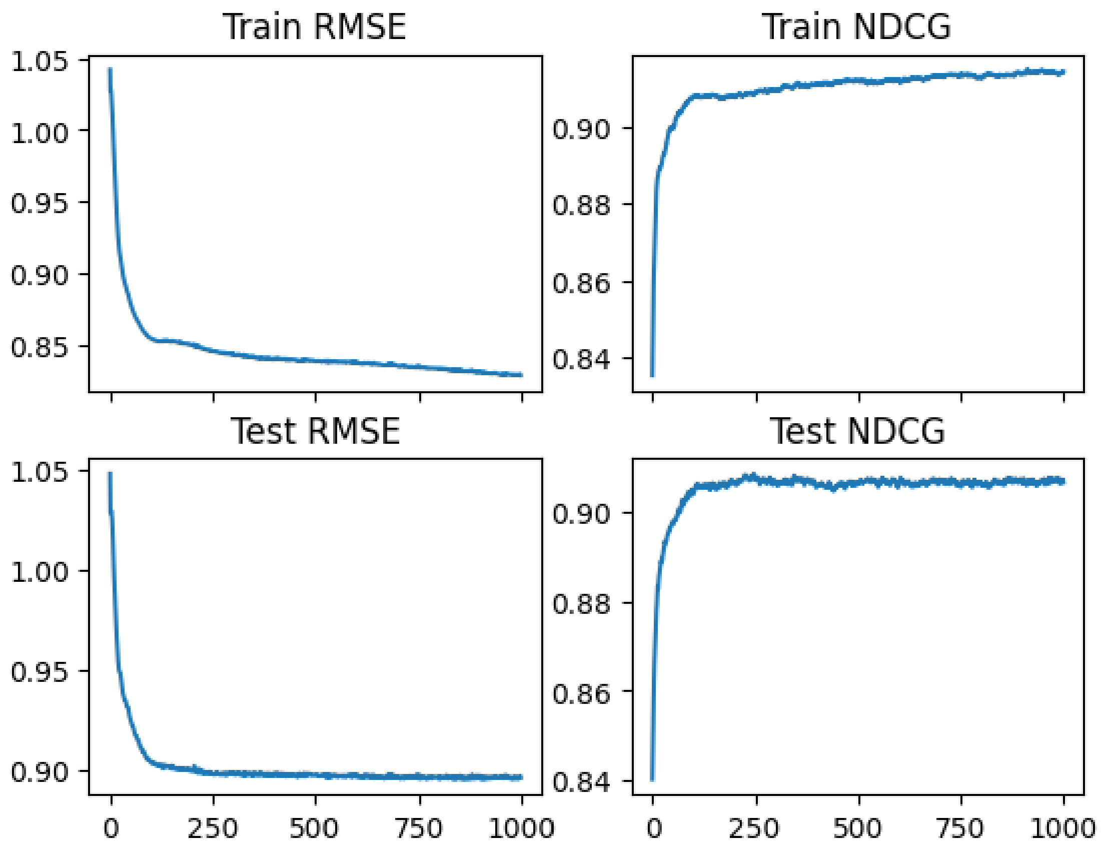
1) The code was line-to-line adapted for working with PyTorch and hence the structure of the solution is not “good”.
2) Some imported modules are not used at all.
3) Lack of the comments for some important parts of the solution.
4) An implementation of some code parts was questionable or not effective.

Results:

1) All found issues were attempted to be fixed and the result can be found in this GitHub repo.
2) Both of the notebooks from cloned repo were reformatted and run. The expected results were received: approximately 0.89 for RMSE.

3) Both notebooks were run with different sets of hyperparameters(types of optimizers(Adagrad, Adadelta, RMSProp), number of hidden layers(5, 7), size of an AutoEncoder embedding(7, 11)), but in all of the cases either the convergence of the model during a fine-tuning phase was unstable or an overall result (RMSE + NDCG) was worse than the “original” one.
4) Only one of the reformatted notebooks was considered as a solution(with the lowest RMSE obtained).
5) Some required parts of the solution(such as visuals or evaluation script) were added.

Fine-tuning process graphs:



X-axis - n_epochs, Y-axis - RMSE/NDCG value

P.S. In addition to GHRS and GLocal-K, another one solution was considered: DeepFM(3rd code block) from Deep-CTR Torch GitHub package, but in order to not install additional packages it was decided to just take into account that such a solution exists.