Recsys RL

Цель данной задачи была поэкспериментировать с моделями actor critic на рекомендательной системе. Данные были взяты с move lens 1m, где были проставлены рейтинги фильмам, сделанные юзерами. В этой задаче нужно было воспроизвести статью и попробовать провести еще эксперименты.

Как лучше смотреть файлы:

* Scripts/run\_model – файл запуска модели
* Infrastructure/rl\_trainer.py – файл для загрузки DRR модели, также пред загрузки данных и в дальнейшем обучении модели, когда обучает модель предзагружает параметры PMF для эмбедингов юзеров и айтемов, также создает энвайронмент для взаимодействия модели со средой
* Agents/drr\_agent.py – DRR модель, эта модель загружает в зависимости от заданных параметров разные актор и критики модели, также пред загружает разные шумы и имеет несколько различных тренировок для различных моделей, также загружает replay\_buffer и может сохранять модели
* Policies/MLP\_policy.py – файл где написаны все актор модели
* Values/MLP\_value.py – файл где написаны все критик модели
* Infrastructure/environment.py – файл где записаны энвайронмент для взаиможействия агента со средой имеет 2 основных функции, reset для того чтобы очистить память для нового юзера и step, получаешь вознограждение , юзера и память после взаимодействия со средой
* Infrastructure/eval.py содержит в себе класс EvalDataset и метод run\_evalution, с помощью класса EvalDataset получаем положительные и отрицательные айтамы, чтобы по нему итерироваться в дальнейшем, с помощью run\_evalution, проверяем на метриках precision@k и ndcg@k на сколько хорошо обученна наша модель
* Infrastructure/noise.py – записаны разные шумы для нашей модели
* Infrastructure/PMF.py – записан класс PMF для пред тюнинга эмбедингов юзеров и айтемов
* Infrastructure/pytorch\_util.py – написаны методы для приведения к numpy и создания нейронных сетей
* Infrastructure/replay\_buffer.py – написано 2 баффера для наших моделей
* Infrastructure/state.py – написаны различные стейты для преобразования наших айтемов и юзеров к каким-то определенным эмбедингам для дальнейшего взаимодействия с ними
* Infrastructure/utils.py – различные вспомогательные функции