LightGCN

OWiSR
Study replication
M. Drwal, M. Raczkowski
17.04.2025

Introduction

LightGCN - the paper

- LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation
- Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020 In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR '20), July 25–30, 2020, Virtual Event, China. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401063
- Source: https://arxiv.org/pdf/2002.02126

LightGCN - the concept

- loss:
- layers:
-

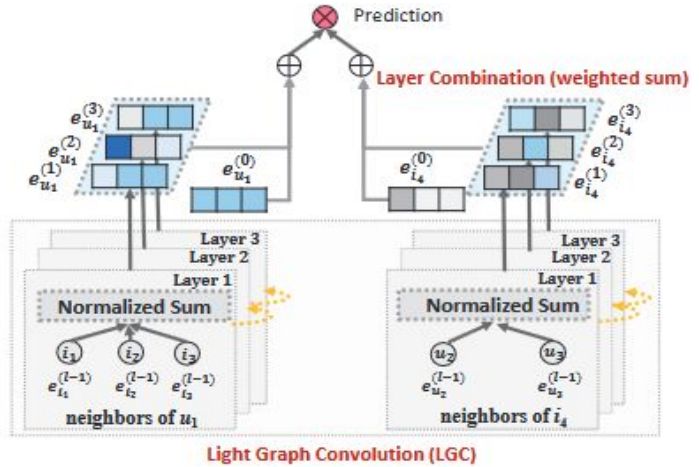


Figure 2: An illustration of LightGCN model architecture. In LGC, only the normalized sum of neighbor embeddings is performed towards next layer; other operations like self-connection, feature transformation, and nonlinear activation are all removed, which largely simplifies GCNs. In Layer Combination, we sum over the embeddings at each layer to obtain the final representations.

Replication

Replication

- code adjustments to enable running on Kaggle
- params: `--decay=1e-4 --lr=0.001 --layer=3 --seed=2020 --topks="[20]" --recdim=64`
- new dataset: finding data & data preparation
- extra metrics: F1, False Discovery Rate

Distinct Users: 29858

Distinct Items: 40981

Dataset (Gowalla)

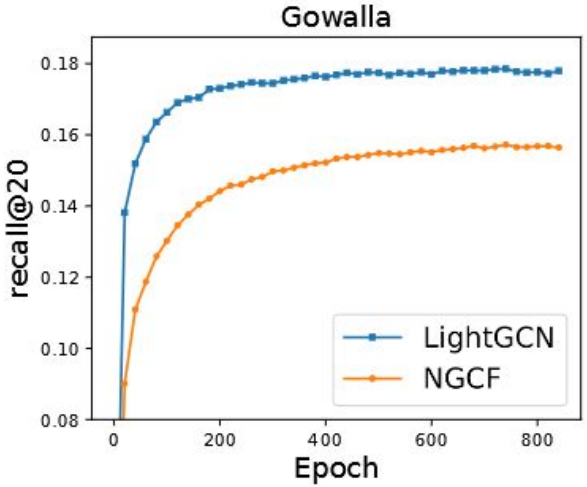
Train:

```
0 0 1 2 3 4 5 6 7 8 9 10 11
1 127 128 129 130 131 132 1
2 176 177 178 179 180 181 1
3 259 260 244 243 261 262 2
4 192 205 277 176 278 279 2
5 290 208 277 291 292 293 2
8 529 530 531 532 533 534 5
6 176 294 295 324 554 555 5
```

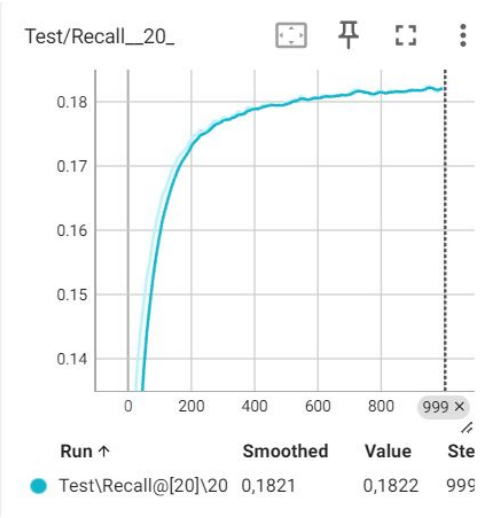
Test:

```
0 7580 3730 5983 5990 7608 1213
1 31613 34341 36414 36721 16213
2 1526 1513 1686 1500 35515 2675
3 32189 8437 5178 5179 6451 3280
4 5284 14418 26420 6988 28644
5 189 792 11355 2315 16498 186 2
6 6668 1776 529 6444 470 1096 17
```

Gowalla: recall@20



original results

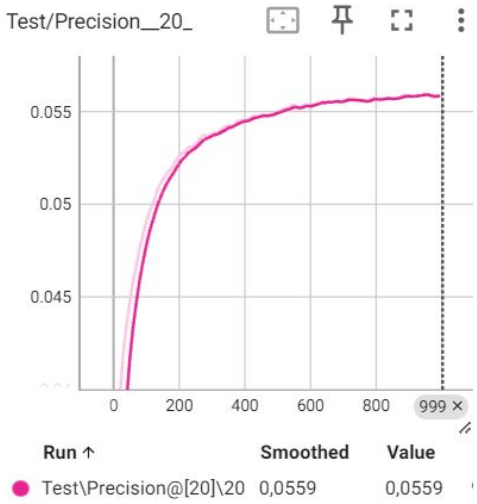


replication

0.1824 (original) vs. 0.18224158 (replicated)

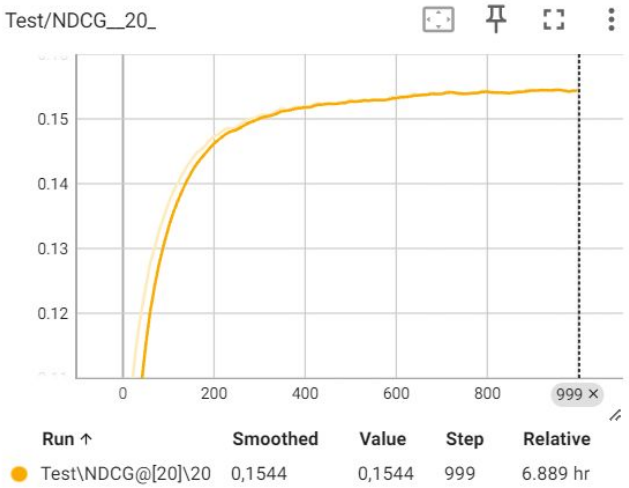
Gowalla: precision@20

0.05589
(original)
vs.
0.05589792
(replicated)



Gowalla: NDCG@20

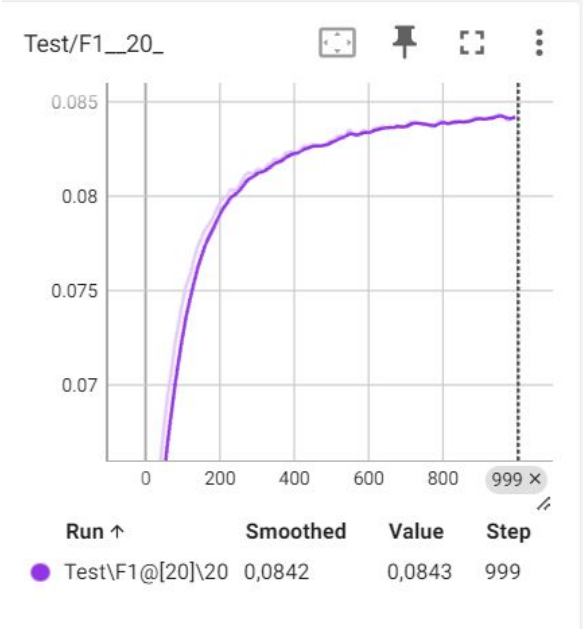
0.1547
(original)
vs.
0.15440831
(replicated)

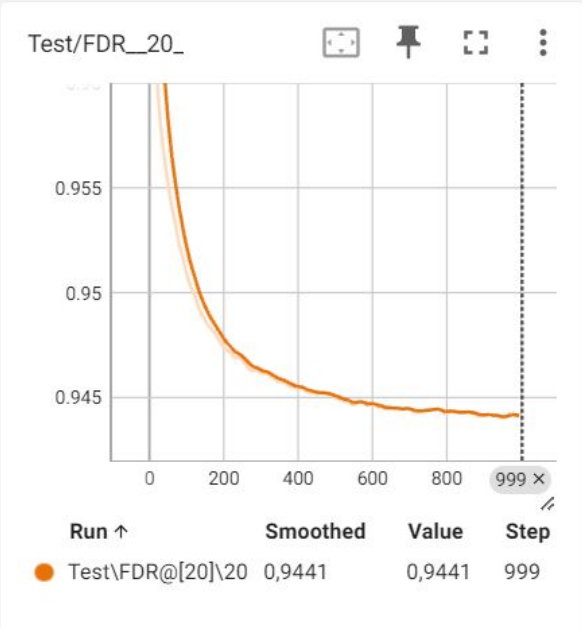


Gowalla: F1@20 and FDR@20

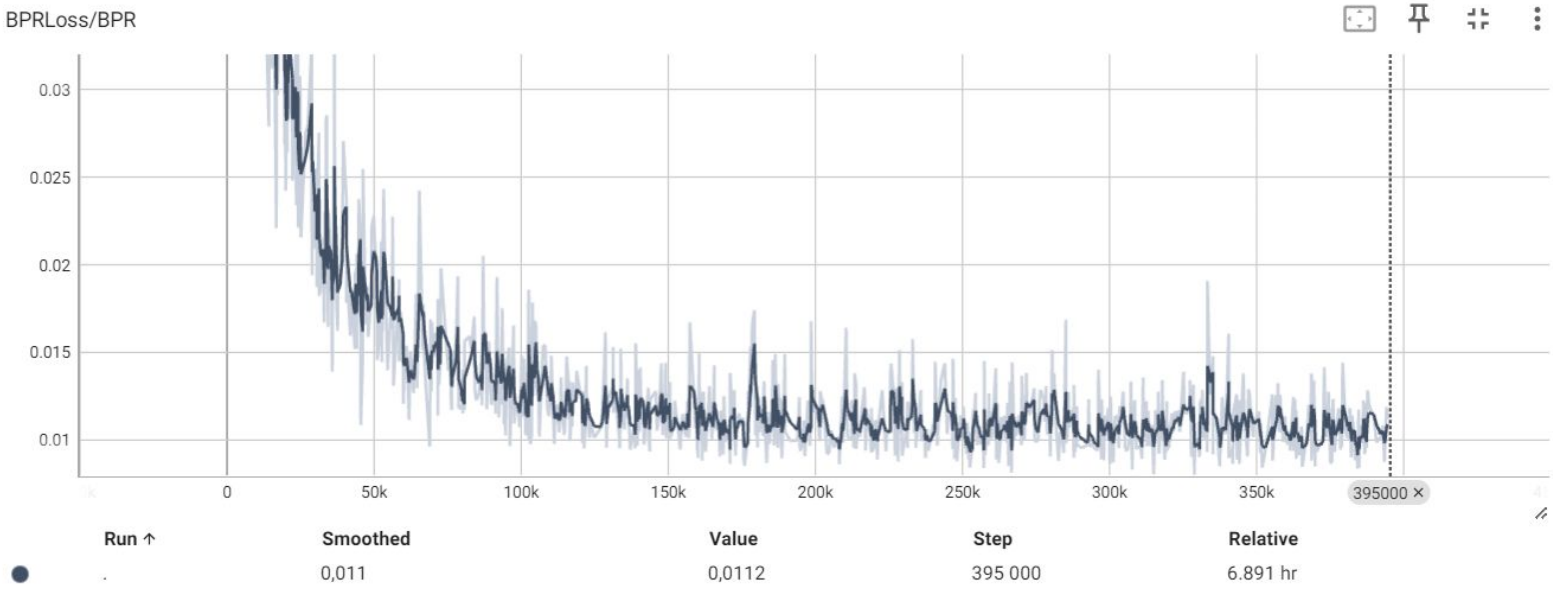
F1@20=0.08425803

FDR@20=0.94410208

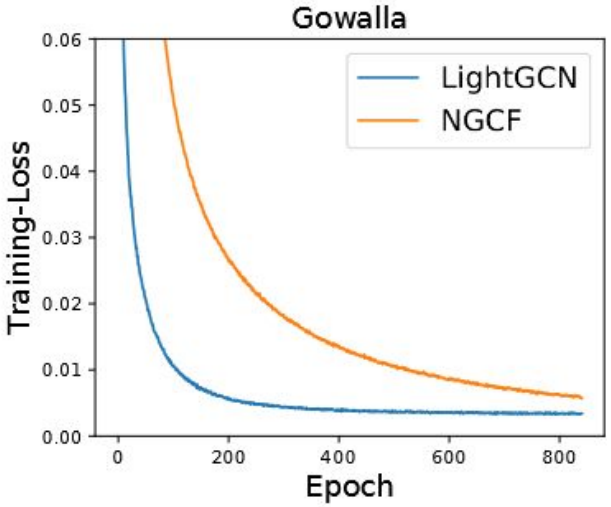


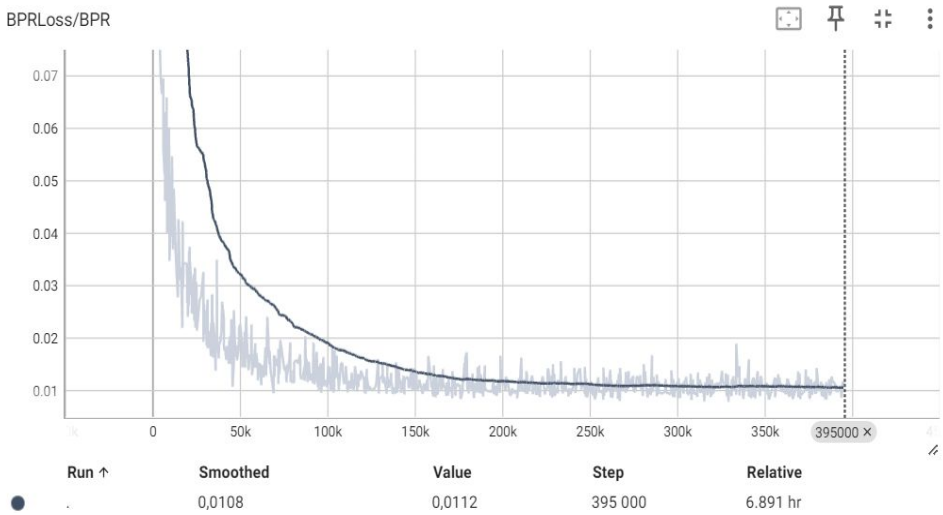


Gowalla: loss



Gowalla: loss





original results

replication

Replication summary

Metric	Original results	Replication results	(Replication-Original)/ Original [%]
precision	0.05589	0.05589792	0.01417
recall	0.1824	0.18224158	-0.08685
NDCG	0.1547	0.15440831	-0.18855
loss	albo nie	0.0112	

External dataset

Distinct Users: 14585
Distinct Items: 103342

Dataset (Amazon Electronics)

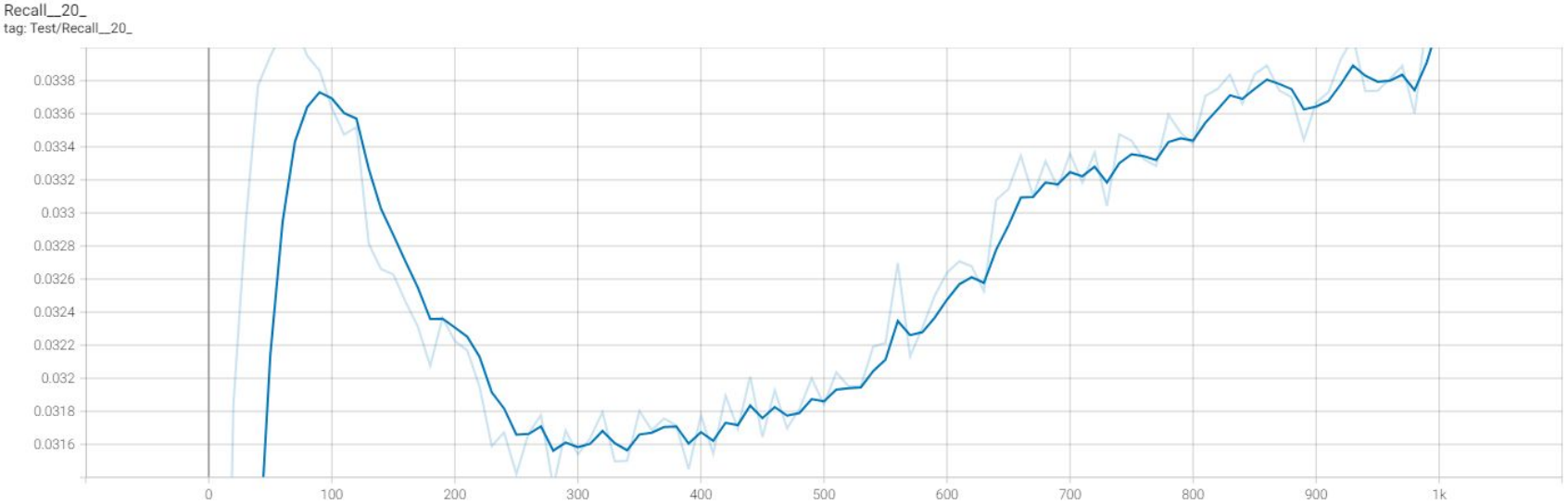
Train:

```
0 16 3936 22645 42669 67862 683
1 0 241 2424 6368 11526 12920 1
2 1 7365 23955 36363 37614 4234
3 2 772 1573 4118 5057 5072 516
4 3 2173 4131 23275 32313 45211
5 3 8406 8993 11141 11310 18723
6 3 5614 8737 23219 29011 33027
7 20390 24417 32313 48366 52223
```

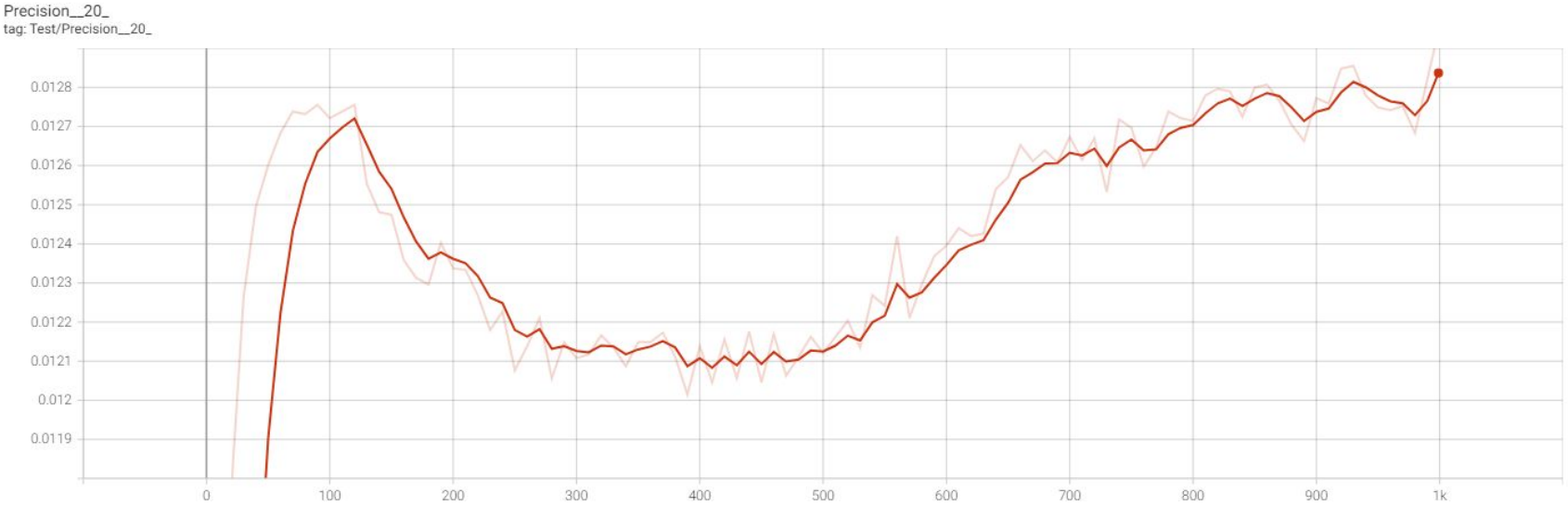
Test:

```
0 0 57131 76767 78450
1 40144 49395 60637 62738 6323
2 36351 44517 71668 76578
3 4120 5686 7459 8822 9532 956
4 2294 36188 49583 53872 70508
5 23432 26182 29786 36993 3855
6 9191 17899 49622 65020 82923
7 3 1639 35580 37108 48059 600
```

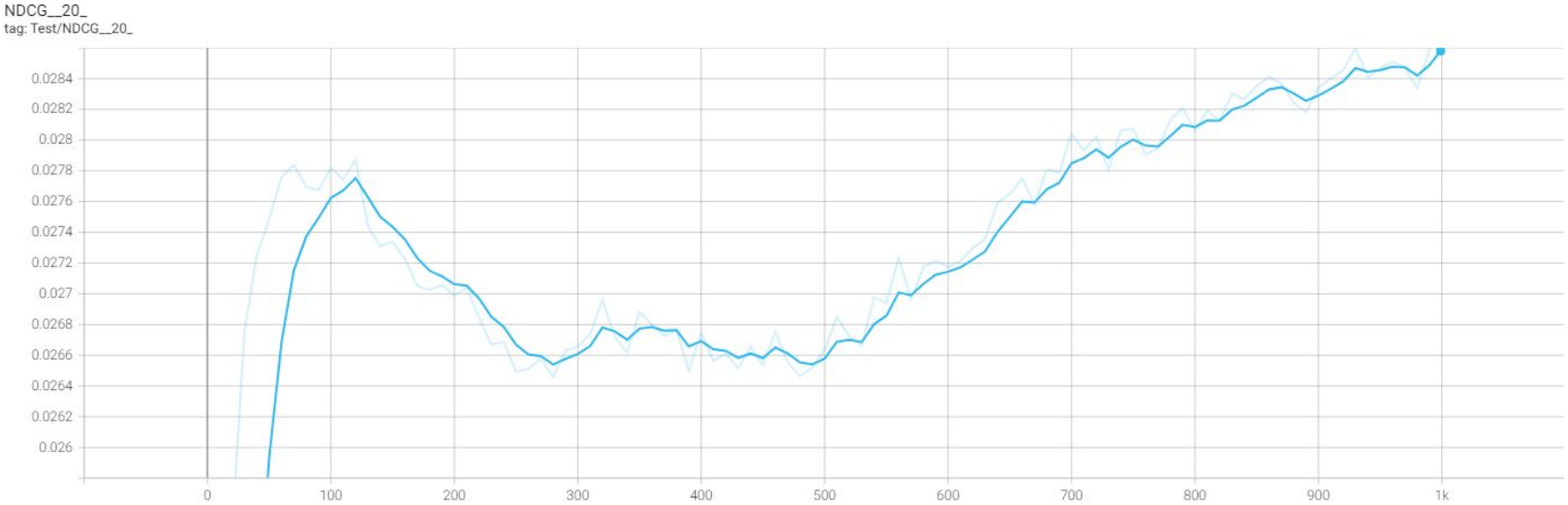
Amazon Electronics: recall@20



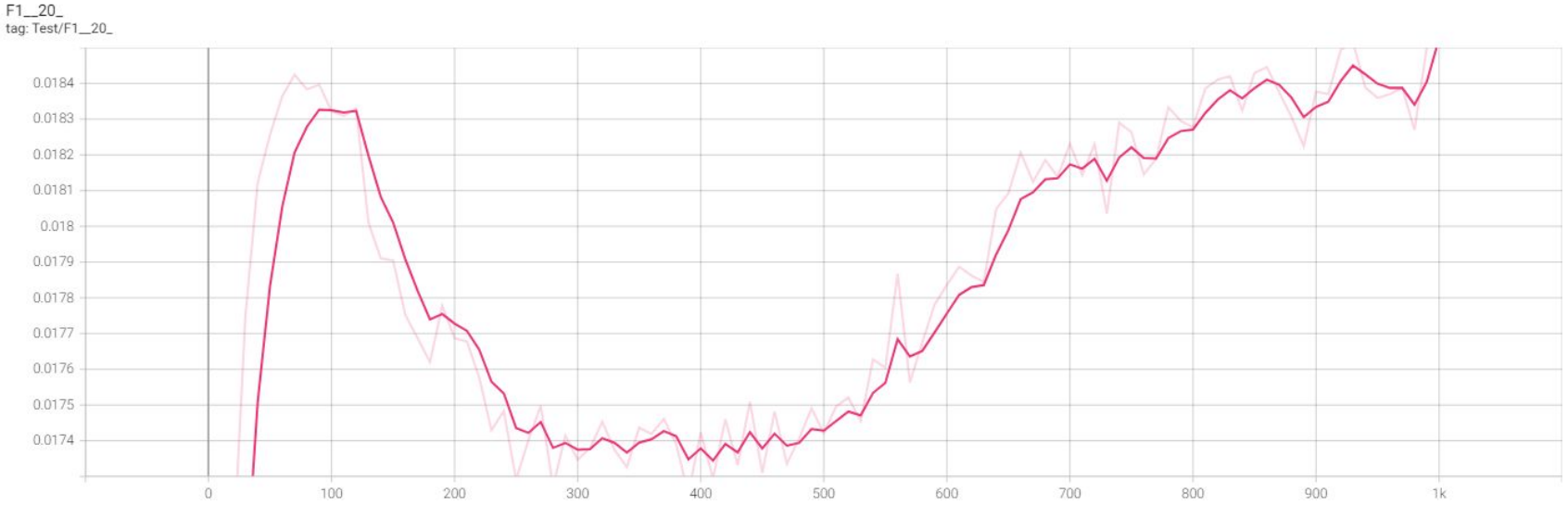
Amazon Electronics: precision@20



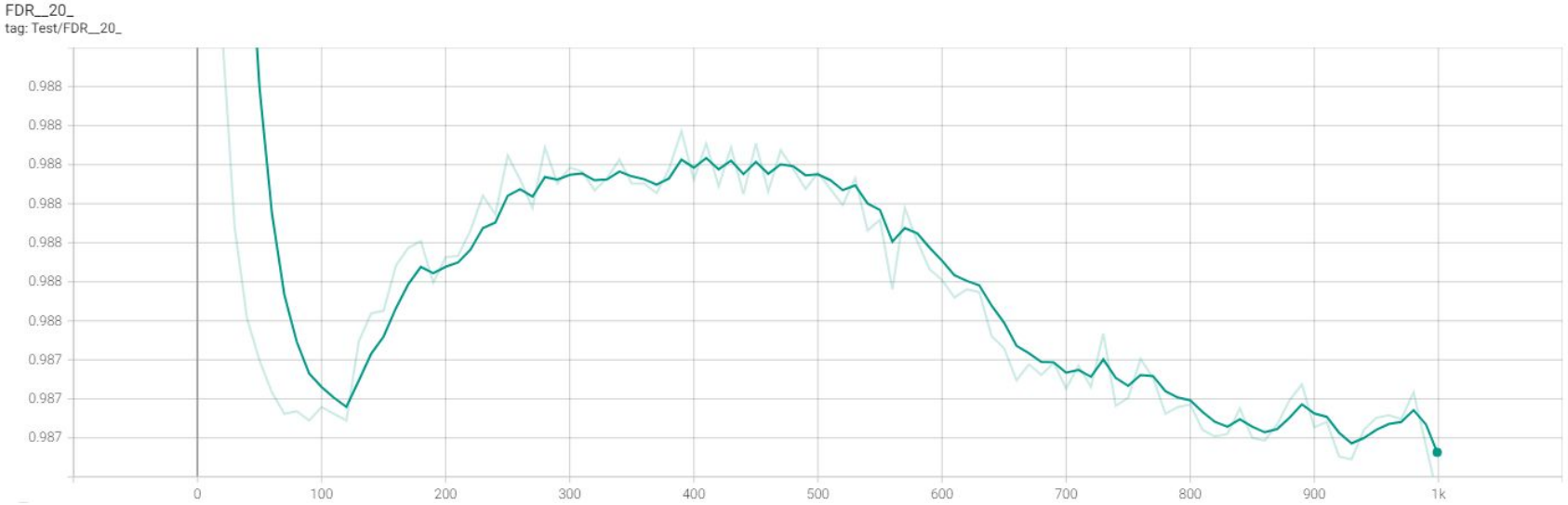
Amazon Electronics: NDCG@20



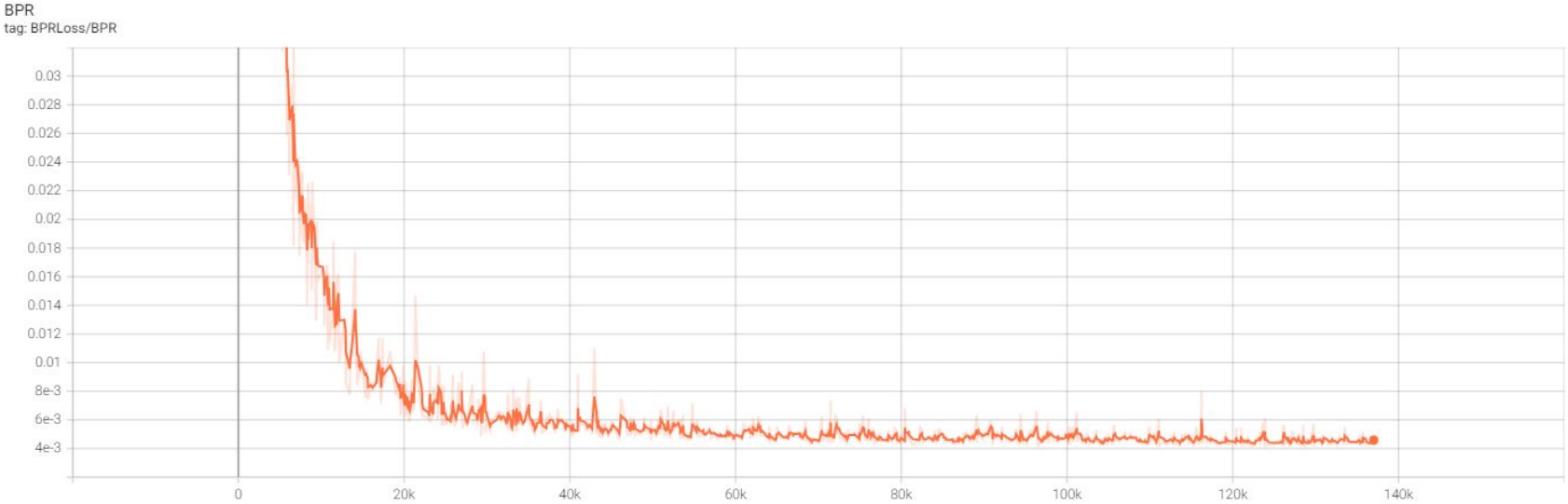
Amazon Electronics: F1@20



Amazon Electronics: FDR@20

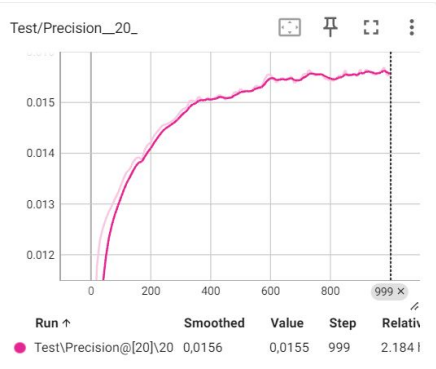


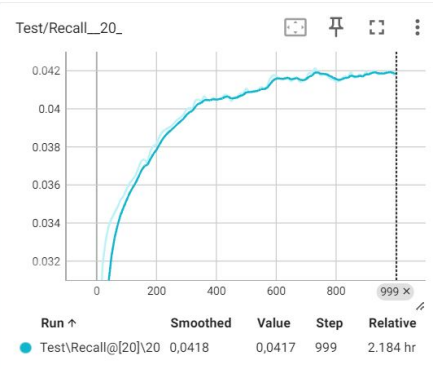
Amazon Electronics: loss



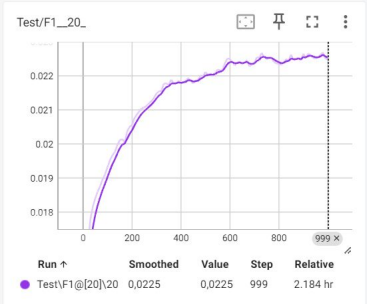
Fine-tuning (mini)

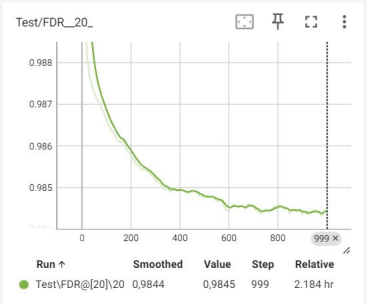
Amazon Electronics: greater regularisation

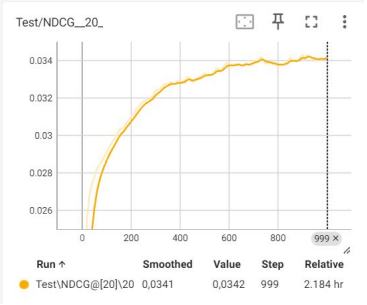




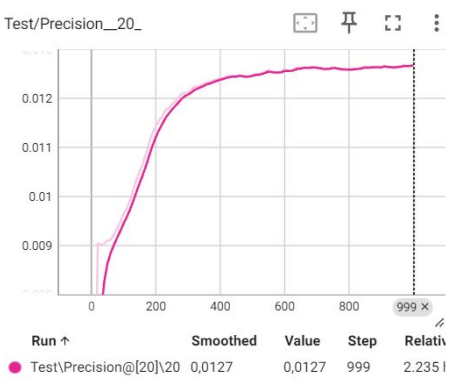
- `--decay=1e-3 --lr=0.001 --layer=3 --seed=2020 --topks="[20]" --recdim=64`
- `{'precision': ([0.01553544]), 'recall': ([0.04174712]), 'ndcg': ([0.03422559]), 'f1': ([0.0224886]), 'fdr': ([0.98446456])}`

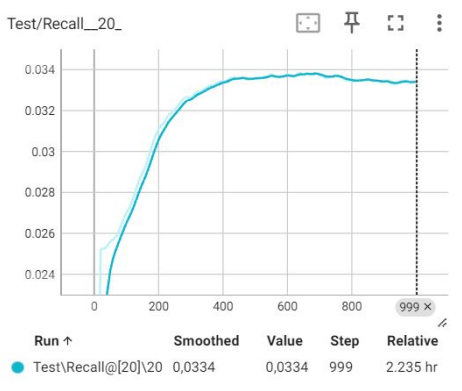




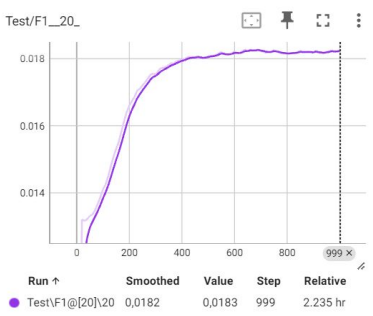


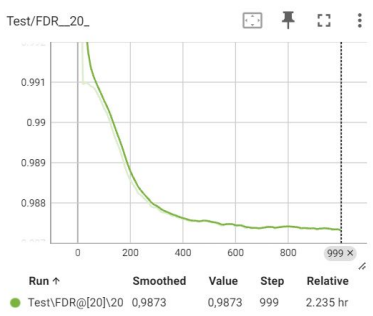
Amazon Electronics: smaller learning rate

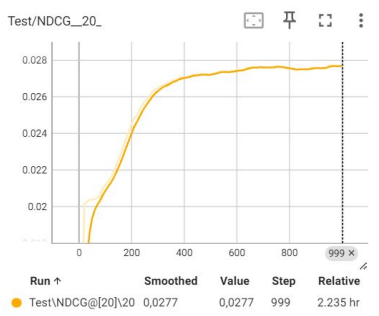




- `--decay=1e-4 --lr=0.0001 --layer=3 --seed=2020 --topks="[20]" --recdim=64`
- `{'precision': ([0.01269025]), 'recall': ([0.03343737]), 'ndcg': ([0.02770338]), 'f1': ([0.01825277]), 'fdr': ([0.98730975])}`

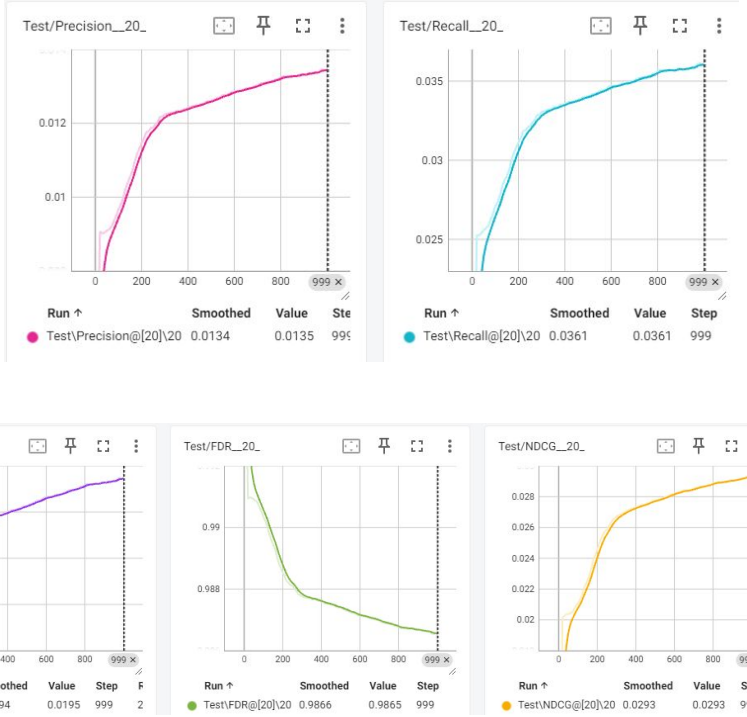






Amazon Electronics: greater regularisation and smaller learning rate

- `--decay=1e-3 --lr=0.0001 --layer=3 --seed=2020 --topks="[20]" --recdim=64`
- `{'precision': ([0.0135]), 'recall': ([0.0361]), 'ndcg': ([0.0293]), 'f1': ([0.0195]), 'fdr': ([0.9865])}`



Summary

Issues & conclusion

- Searching for paper which results would be reproducible in a home environment
- Code adjustments so that it could be run on remote machine
- GPU access limits
- Understanding logic behind given data
- Data preparation script
- Code adjustments so that a part of it would run on the local machine (
- Understanding the results

Sources

- https://arxiv.org/pdf/2002.02126
- https://www.kaggle.com/code/marekd6/lightgcn
- https://www.kaggle.com/datasets/marekd6/lightgcn-data-python-modules/data
- https://github.com/188686/LightGCN/tree/main
- https://www.kaggle.com/code/matixos13/notebook21fb19a7e8/notebook
- https://www.kaggle.com/datasets/matixos13/lightgcn-data-python-modules