UE19CS345 – Network Analysis and Mining Course Project

Project Title

A Comparative Study of Recommendation Quality Using LightGCN

Team member 1 : PES1UG19CS058 Ananya Uppal
Team member 2 : PES1UG19CS254 Maitreyi P
Team member 3 : PES1UG19CS318 P Shreya
Team member 4 : PES1UG19CS542 Trisha Jain

Topic and its uniqueness

Define your problem statement crisply

- Assess the performance of a LightGCN recommender model by editing embedding weights. Perform a comparative study on the impact of different methods of aggregation weight calculation on the quality of recommendation.

Your Lab evaluations vis-a-vis what you are attempting to do

- Assignment 2 prompted us to use a baseline GCN model. In this project we've worked on a proposed improvement to the model that retains only a GCN's most important component - neighbourhood aggregation

Why you think this project is interesting or unique

- The model's deviation from state-of-the-art GCNs with its simplicity, along with the performance standard mentioned in the paper are what drew our interest. We believe it has a potential worth exploring.

Dataset

Dataset Used - **MovieLens** [https://movielens.org/]

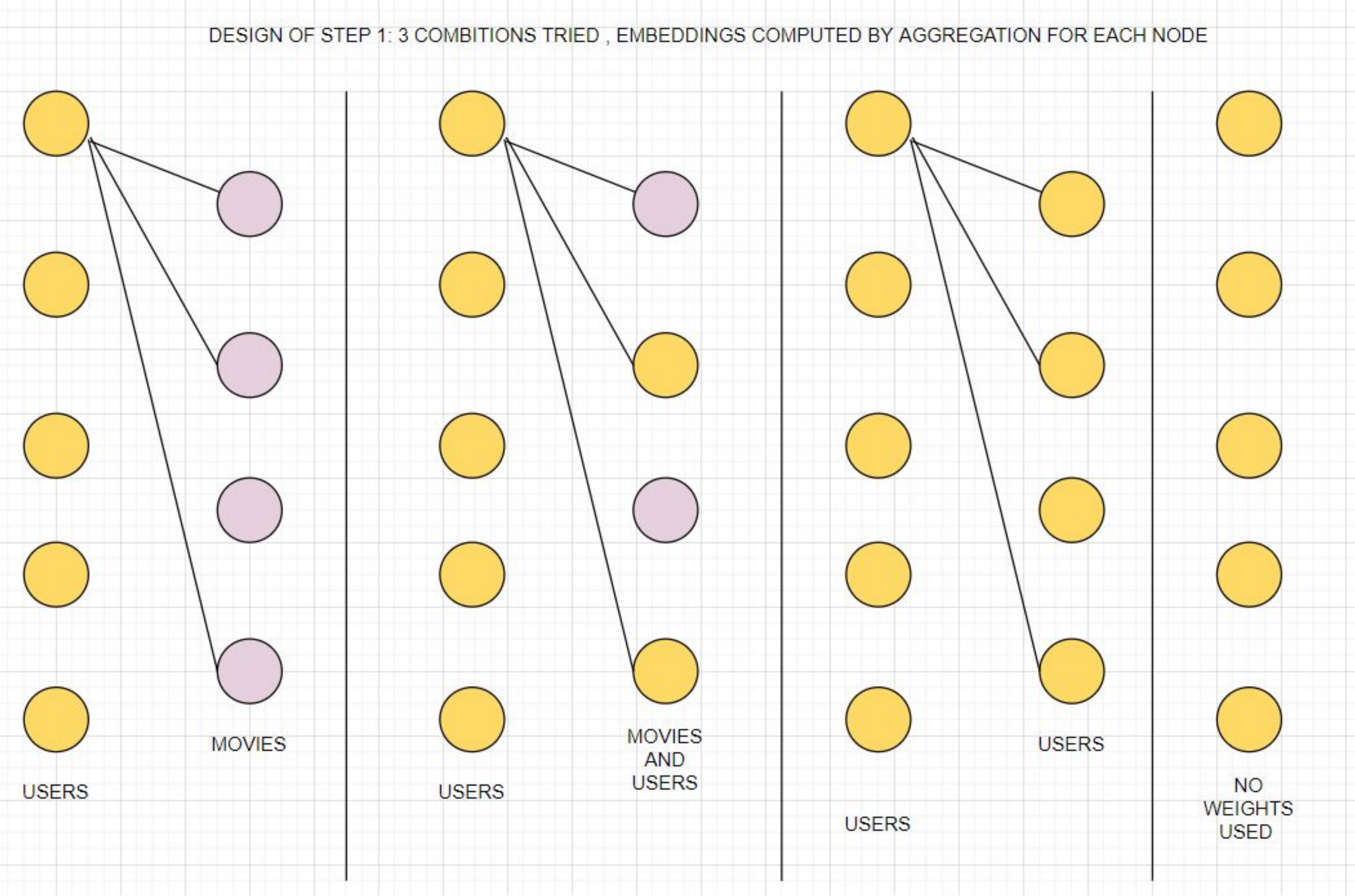
Why? MovieLens is baseline dataset used for the evaluation of recommender systems.

- Size : 1 Million
- Attributes :

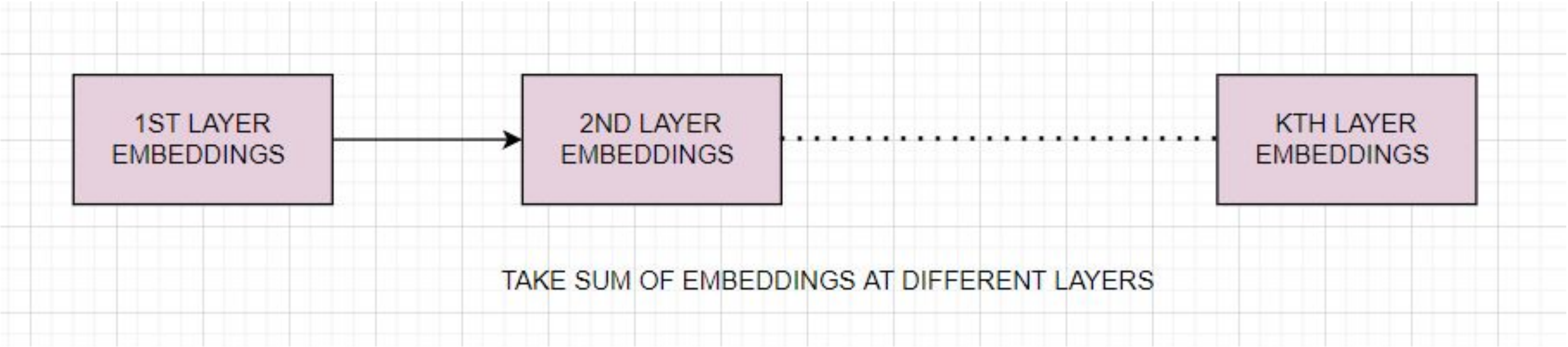
  For users - UserID, Age, Gender, Occupation, Zip-code of location

  For Items (Movies) - MovieID, Title, Genre

Overall design or approach in a diagram



Overall design or approach in a diagram



Evaluation Metrics

Metrics:

- Mean Average Precision@K
- Mean Average Recall@K

for the GCN recommender system at different values of embedding weights

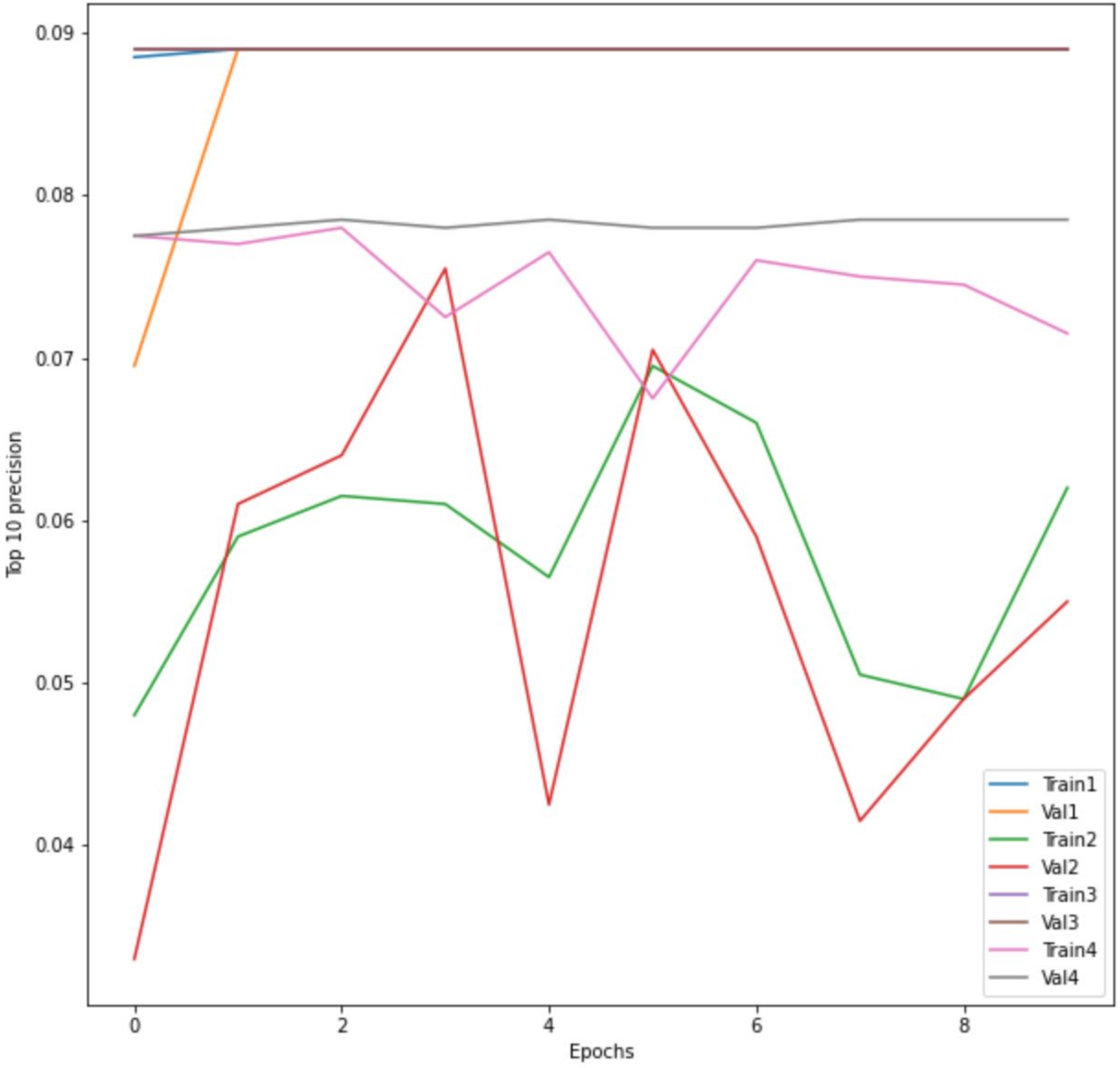
Why we have chosen these metrics:

MAP@K is a measure of how relevant the list of items are
MAR@K is a measure of how well the recommender is able to recall the positively rated items. These are the standard metrics of choice for evaluation of RS.

Techniques used for computing weights for aggregation

S.No.	Technique used	Representation in plots
1.	Using neighboring user counts	Train1, Val1
2.	Using neighboring item counts	Train2, Val2
3.	Using both neighboring item and neighboring user counts	Train3, Val3
4.	Aggregation without weights	Train4, Val4

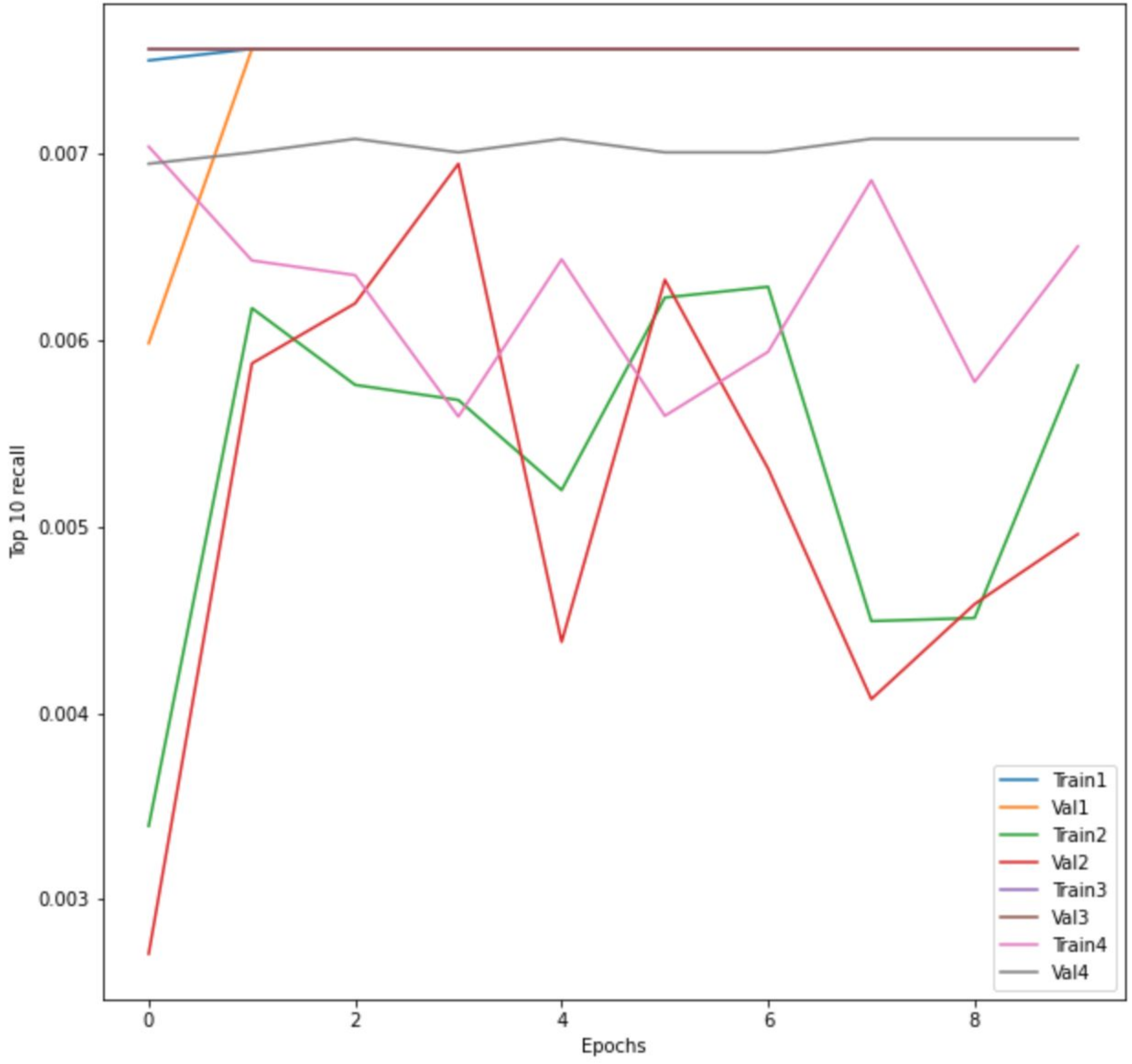
Visualisation of precision values:



Conclusions :

Precision and recall of the model can't be anticipated from using just the neighbouring item count values for aggregation

Visualisation of recall values:



Conclusions :

Precision and recall of the model can't be anticipated from using just the neighbouring item count values for aggregation

Final Result:

Best Outcome is produced by using the neighbouring user and item counts.

What are the remaining portions in this project ?

- In-depth analyses are towards the rationality of LightGCN from both technical and empirical perspectives.
- Optimisation of resource consumption and performance.
- Investigating different embedding aggregation methods to improve model performance.
- Incorporating regular collaborative filtering and comparing the quality of results.

Top few learning

Serial No	Top learning in this project
1	The two most common designs in GCNs, feature transformation and nonlinear activation, contribute little to the performance of collaborative filtering.
2	The most essential component in GCN is neighborhood aggregation for collaborative filtering.
3	Precision and recall of the model can’t be anticipated from using just the neighbouring item count values for aggregation.

Quantity and quality of work

No	Code functionality	% Complete	Runs without problem (Y/N)	If there are minor issues, indicate
1	Preparing the configuration dictionary and preparing the data	100%	Y	
2	Designing the GCN Layer using different techniques	100%	Y	
3	Building the models using different techniques	100%	Y	
4	Computing the ratings and embeddings	100%	Y	
5	Defining the evaluation metrics	100%	Y	
6	Creating Training, Validation and Testing samples	100%	Y	
7	Training the models	100%	Y	
8	Comparing the different models	100%	Y	
9	Prediction using the model-3	100%	Y	
10	Comparing LightGCN with baseline MF technique	100%	Y	

Top unresolved challenges

Serial No	Brief description of unresolved challenges	Type of challenge (scope/data/design/implementation / others)
1	Personalizing the layer combination weights so as to enable adaptive-order smoothing for different users (e.g., sparse users may require more signal from higher-order neighbours while active users require less)	Out of scope
2	Using a massive dataset comprised of millions of users and items	Lack of hardware resources (GPU)
3	Exploring LightGCN on graph based models like SOTA GCNs and factorization machines. They can exploit auxiliary information such as item knowledge graph, social network, and multimedia content, for recommendations. These models are also modeled by same neural operations as Neural Graph CF that may be unnecessary, thus having scope for improvement with LightGCN's simplification approach.	Out of scope + challenging implementation

Reference papers, if any

No	Paper Title	Authors
1	LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation	He, Xiangnan and Deng, Kuan and Wang, Xiang and Li, Yan and Zhang, Yongdong and Wang, Meng
2	Neural Graph Collaborative Filtering	Xiang Wang and Xiangnan He and Meng Wang and Fuli Feng and Tat-Seng Chua