1 Introduction

In the past decade, video platform has become more and more popular and useful in internet world. One of the most used video platforms is YouTube, which allows uses to store, upload, convert, and play back video content on the internet through large-scale, structured system. The most powerful tool of video platform is Recommendation System to recommend what video to watch next. Here we will review a novel model architecture of Google's multitasking recommendation system that improve recommendation quality on YouTube.

2 Basic of Recommendation System

Recommendation System is composed of two parts: candidate generation and ranking. For candidate generation part, the system provides user possible candidates of video based on user's information, history, and the video that is watched. For example, the system pre-selects 50000 candidates. The system then applies multiple candidate generation algorithms to down-size the candidates to 5000, then to 500 eventually. The principle of algorithm for selecting candidates is to capture the similarity of query video and candidate video. As mentioned in [2], one algorithm generates candidates by matching topics of query video. Another algorithm retrieves candidate videos based on how often the video has been watched together with the query video [2]. Other aspects considered include user history, context, etc [2].

For ranking part, machine learning algorithms using a learning-to-rank framework are widely adopted [2]. For example, [3] explored both point-wise and pair-wise learning to rank framework with linear models and tree-based methods [2]. [4] used a linear scoring function and a pair-wise ranking objective [2]. This article mainly focuses on the optimization of the ranking part, which will be summarized in next session.

3 The two main Challenges to solve for video recommendation system

3.1 Multi-objective Learning

It has been mentioned that there are often objective conflicts when recommending video to users. For example, we may want to recommend videos that users rate highly and share with their friends, in addition to watching [2]. There are two groups in the multi-objective functions, which are Engagement Objective and Satisfaction Objectives. Engagement Objective is to capture user behaviors such as watch and clicks. Satisfaction

Objectives is to capture likes, rating, comments, etc.

At the beginning of the model as shown in Figure 1, multiple input features and embedding are fed into the shared hidden layer. Then mixture of experts layer(MMoE) are applied with a Gating Networks. Each of the experts in the MMoE layer tries to learn a different feature of the input then the output is sent to Gating Newworks[1] (Figure 2). The output of Gating Newtworks is then used to fulfill Engagement Objective and Satisfaction Objectives

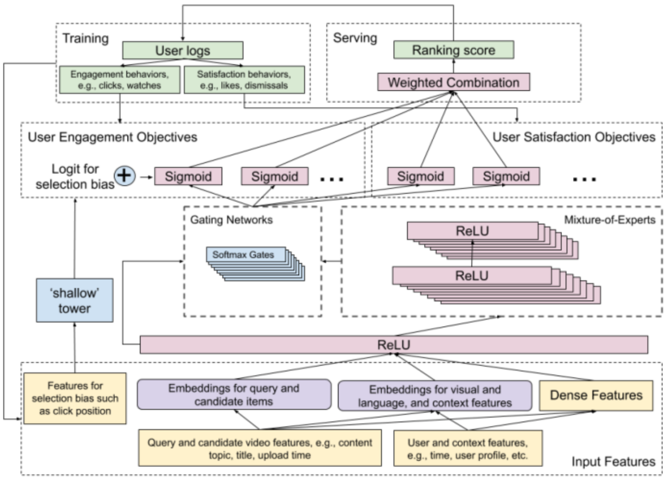

Figure 1: Model architecture of our proposed ranking system. It consumes user logs as training data, builds Multi-gate Mixture-of-Experts layers to predict two categories of user behaviors, i.e., engagement and satisfaction. It corrects ranking selection bias with a side-tower. On top, multiple predictions are combined into a final ranking score.

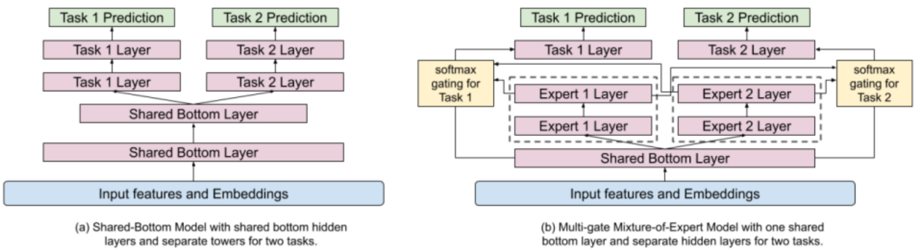

Figure 2: Replacing shared-bottom layers with MMoE.

3.2 Bias processing and removal

It has been mentioned that there is also implicit bias when recommending video to users. For example, a user might have clicked and watched a video simply because it was being ranked high, not because it was the one that the user liked the most [2]. As a result, models trained using data generated from the current system will be biased, causing a feedback loop effect [5].

To solve this problem, shallow tower is used to this model as shown in Figure 3. The shallow tower is trained using features that contribute to the bias like position of the recommendation and tries to predict whether there is a bias component involved in the current instance [1].

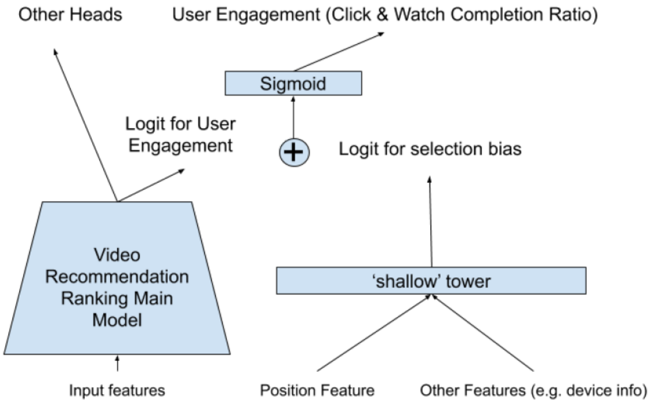

Figure 3: Adding a shallow side tower to learn selection bias (e.g., position bias).

4 Experiment Result

The model is tested by comparing the engagement Metric and Satisfaction Metric among Share-Bottom, MMoE (4 experts), MMoE(8 experts) model Architecture, as shown in Table 1. The result shows that MMoE can increase the performance significantly. On the other hand, Shallow tower method also improve the Engagement Metric significantly, as shown in Table 2.

Recommending What Video to Watch Next: A Multitask Ranking System

Model Architecture	Number of Multiplications	Engagement Metric	Satisfaction Metric
Shared-Bottom	3.7M	/	/
Shared-Bottom	6.1M	+0.1%	+ 1.89%
MMoE (4 experts)	3.7M	+0.20%	+ 1.22%
MMoE (8 Experts)	6.1M	+0.45%	+ 3.07%

Table 1: YouTube live experiment results for MMoE.

Method	Engagement Metric
Input Feature	-0.07%
Adversarial Loss	+0.01%
Shallow Tower	+0.24%

Table 2: YouTube live experiment results for modeling position bias.

5 Conclusion

A new model of recommendation system is developed in this article. The model is the optimization and extension of the Wide & Deep model by adding Multi-gate Mixture-of-Experts and Shallow Tower. The Multi-gate Mixture-of-Experts helps solve multi-objective problem, and the Shallow Tower helps solve implicit bias issue. The test result shows that this model does improve the recommendation quality of video.

[1] Suneet Bhatia (2022, February). A Multitask Ranking System: How YouTube recommends the Next Videos

[2] Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., ... Chi, E. (2019, September). Recommending what video to watch next: a multitask ranking system.

[3] David C Liu, Stephanie Rogers, Raymond Shiau, Dmitry Kislyuk, Kevin C Ma, Zhigang Zhong, Jenny Liu, and Yushi Jing. 2017. Related pins at pinterest: The evolution of a real-world recommender system. In Proceedings of the 26th International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee, 583–592.

[4] Antonino Freno. 2017. Practical Lessons from Developing a Large-Scale Recommender System at Zalando. In Proceedings of the Eleventh ACM Conference on Recommender Systems. ACM, 251–259.

[5] Ayan Sinha, David F Gleich, and Karthik Ramani. 2016. Deconvolving feedback loops in recommender systems. In Advances in Neural Information Processing Systems. 3243–3251