Tech Review: YouTube’s Multitask Ranking System

Introduction

YouTube has as video recommendation system that create a list of videos to recommend to the viewer to watch after the current video. It consists of candidate generators (which generate several hundred candidates to be ranked), a ranking system, and post-ranking adjustments (deduplication, diversification, etc.). This tech review focuses on the ranking part of the system, and evaluates its design goals, its Multi-gate Mixture-of-Experts design, its experiment setup, its scalability, and its application in an industrial recommendation system. It reviews the positions of the creators of this system [1] and includes some observations and considerations based on my industry experience.

Body

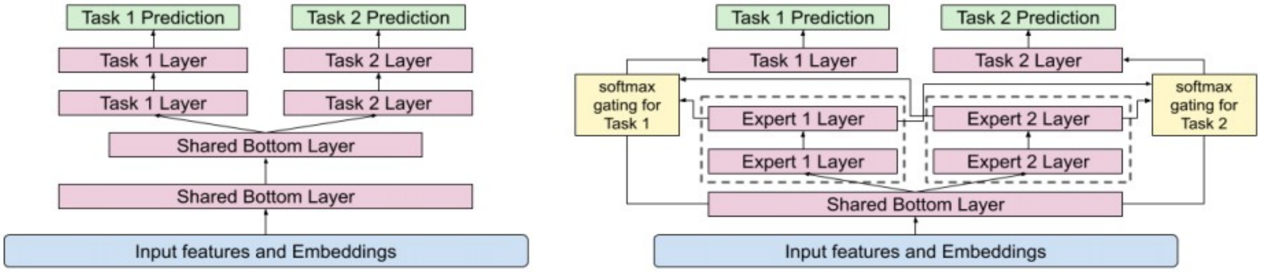
Design Goals

The ranking system of YouTube’s video recommender ranks several hundred candidates provided by the candidate generators based on multiple objectives, including engagement objectives (clicks, watches, etc.) and satisfaction objectives (ratings, etc.). The objectives are not highly correlated (click-baits tend to have high click-through rates and low ratings). The ranking system also aims to mitigate the selection bias based on the rank position, user device, etc. As an industrial system, the ranking system needs to be efficient, scalable, and configurable.

Design: Multi-gate Mixture-of-Experts (MMoE) and Shallow Tower

The ranking system learns from users’ implicit feedbacks (clicks and watches) and explicit feedbacks (ratings and likes). It performs point-wise evaluation as opposed to pair-wise evaluation for better scalability. The number of point-wise evaluations is linear to the number of candidates to be ranked, whereas the number of pair-wise evaluations would be a quadratic function. Note that based on my experience, even though pair-wise ranking produces better diversity (better exploration) than point-wise ranking, such diversity may also be achieved by adjustments (down ranking consecutive videos with high content similarity, for instance) after the initial point-wise ranking, without incurring quadratic cost.

The ranking system uses a Multi-gate Mixture-of-Experts (MMoE) architecture to handle independent or conflicting objectives. It has a shared bottom layer to process the user/content features and embeddings, and instead of the common practice of using the completely separate task layers for the objectives, it enhances the learning of multiple objectives (tasks) using soft parameter sharing among the expert layers.

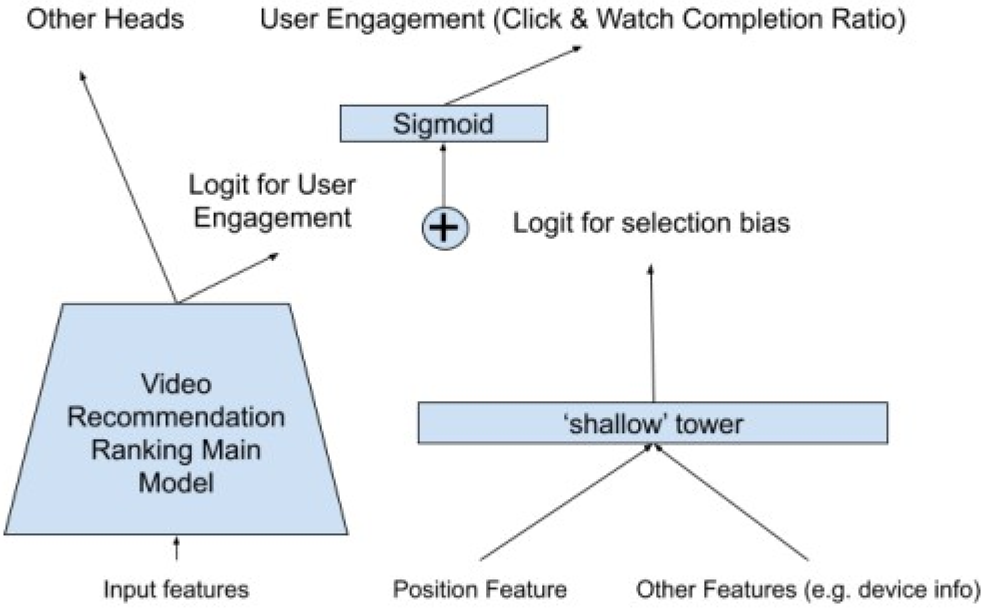


(a) Shared-Bottom Model with shared bottom hidden layers and separate towers for two tasks.

(b) Multi-gate Mixture-of-Expert Model with one shared bottom layer and separate hidden layers for two tasks.

Credit: Zhao, et al, Recommending What Video to Watch Next: A Multitask Ranking System

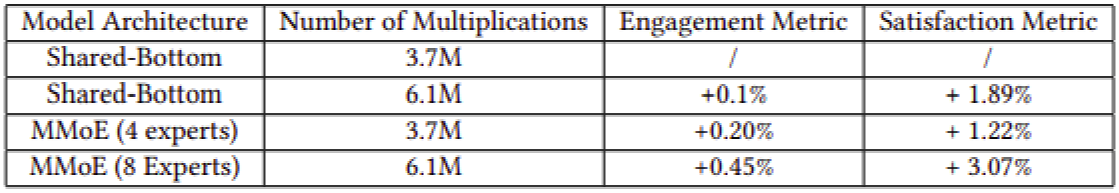
The ranking system mitigates the position bias (videos getting more clicks not because they are more relevant to the users but because they are ranked higher by the current ranking system) with shallow tower next to the main ranking model. Note that the shallow tower is in the same neural network as the main ranking model, so that it learns the position bias from the production system, as opposed to random experiments. The input to the shallow tower includes device and user session information because a certain position may be on the first page in a device with a large screen (hence smaller bias) but not on the first page in a device with a smaller screen (hence larger bias).


Other Heads
User Engagement (Click & Watch Completion Ratio)
Sigmoid
Logit for User Engagement
+
Logit for selection bias
Video Recommendation Ranking Main Model
'shallow' tower
Input features
Position Feature
Other Features (e.g. device info)

Credit: Zhao, et al, Recommending What Video to Watch Next: A Multitask Ranking System

Experiments

Zhao, et al, conducted experiments on YouTube’s recommendation system by training the proposed model using several days worth of user activity data and compared it with the baseline model in offline experiments and online A/B tests. For offline tasks, they monitored the AUC for classification tasks (e.g. click vs. no click) and the squared errors of the regression tasks (e.g. ratings and watching times). For the A/B tests, they compared the engagement and satisfaction metrics of the live users. The MMoE architecture outperformed the Shared-Bottom architecture, controlled for the size of the neural network.



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%

Credit: Zhao, et al, Recommending What Video to Watch Next: A Multitask Ranking System

In my opinion, the experiments are effective in measuring the short term utility (exploitation) by individual users. Further studies are needed to determine the long term utility (exploration) as well as the externality (network effect). For instance, are several days worth of experiment sufficient for evaluating the exploration? Do the experiments need to be sub-divided by user segments (new users, occasional users, power users, etc.)? How does the experiment affect the YouTube ecosystem, including the content producers, in additional to the viewers in the test group? Does the ranking system (indirectly) encourage the content producers to produce better contents? How does it affect the control group users that are in some way connected to the test group users (via collaborative filters, for instance)?

Real-world Implications

Flexibility

The multitask ranking system is highly flexible as it produces separate scores for various engagement-based and satisfaction-based tasks. Therefore the final scoring function can be easily adjusted for various user segments (new users vs. power users), markets (prioritizing growth vs. monetization), and market conditions (balancing short term and long term goals).

Scalability of the Production System

The point-based ranking system strikes a good balance between effectiveness and efficiency. ,It scales linearly with respect to the candidate pool. The MMoE model is more effective than the Shared-Bottom Model without sacrificing efficiency, while at the same time being more efficient than completely separating the tasks.

Scalability of the Development Work

YouTube-sized organizations inevitably have many ML engineers and ML teams working on different aspects of the ranking system. One might imagine some would focus on the engagement goals whereas others might focus on satisfaction goals. The MMoE model, by design, does not have clear separation of the expert layers as the Shared-Bottom Model has in its task layers. Therefore the development work on an expert layer, while primarily affects the corresponding task, will affect the other tasks (being worked on by other ML engineers) to a lesser extent. Therefore the MMoE model requires proper infrastructure for back-testing, experimentation, and monitoring, in order for the large ML team(s) to maintain high product quality and development velocity.

Explainability

As YouTube and other social media face increasing scrutiny, we need to be able to understand and justify ML models, as opposed to treating them as a black-box. [2] As we adopt the MMoE model, we would need to develop corresponding techniques to explain the softmax of the expert layers.

Conclusions

The multitask ranking system, as part of YouTube's video recommender, is an effective, scalable, and flexible system for identifying engaging and quality contents for individual viewers. It warrants further evaluation in its impacts on the larger video producer/consumer ecosystem, as well as investments in the testing/monitoring framework and explainability study in its underlying MMoE model.

References

[1] Recommending What Video to Watch Next: A Multitask Ranking System

[2] Machine Learning Explainability