Recommending What video to Watch Next: A Multitask Ranking System

Andrew Ko

In this paper, I am going to review Google's multitask ranking system which aims to solve challenges such as: the goal of ranking or the optimization itself can be ambiguous or conflicting and there are often that a user watches videos that are not necessarily their favorite. The paper introduces an efficient multitask neural network architecture for the ranking system. To address the issue with selection bias from the biased training data, the paper suggests combining a shallow tower in figure 1 to the main model, which takes input from the main model and outputs a scalar value that corresponds to the weight of the input to adjust for that biased term. Overall, the ranking system works to address the following additional issues: Multimodal feature space and Scalability. [1] There are multiple features associated with videos which include but are not limited to the content, thumbnail, audio, title, description, and user demographics. This poses many challenges which require the ranking system to analyze the low-level content features to filter and recommend the right kind of videos. Scalability is also another important factor to evaluate the ranking system and is thus considered into the design of the system.

Each objective of the ranking system is to estimate one type of user behavior related to user utility.[1] In training the model, each objective are fed into the experts through gates correspondingly for deciding the objective function to allow the ranking to support multiple objectives. Consequently, the system adopts a Multi-gate Mixture-of-Expert model shown in figure 2. The so called MMoE layers are implemented to reduce the conflicts of multiple objectives that are in common in other general ranking systems with multiple objectives. The expert layers are essentially a combination of multi-layer perceptrons with ReLU activations of which the outputs are then fed into a Gating Network. Small number of experts are used for training efficiency. Unlike others, MMoE uses soft-parameters and learns by having each of these objects look at each of the experts to choose the ones that correspond to output relevant objective function. The objective function may share or not share the experts with its peers.

Other factor that the multitask ranking system aims to address is the biases that stem from implicit feedback. These feedbacks have been generally supported by many to support learning to rank different models as explicit feedbacks come at a high cost or are not available most of the time. However, there are many times where the reality is a little skewed from the feedbacks. To mitigate the effect of position bias i.e. when the users are more encouraged to watch the videos that are displayed near the top despite the difference in their actual user utility that speak otherwise, shallow tower is attached to the main model. The shallow tower is trained on the data that contributes to the position biases and then are added in the last level of the model, figure 2.[1] To allow the network to learn how to remove these types of biases, the output of the shallow tower is added to the User Engagement Objectives part of the model.

The paper then describes what experiments were conducted to evaluate the proposed ranking system to recommend what video to watch on YouTube for users. In summary, user implicit feedbacks were used to perform offline and live experiments. Along with how the users react to recommended videos through likes, watching or even do not display videos of certain types, the video platform provides a near-perfection environment to test the proposed ranking system. In the real world where data distribution and user activity vary drastically over time, the model is trained first with taking inputs in FIFO order where data of past days are consumed by the model first then the more recent data. The paper claims that they conducted two types of experiments which are conducted offline and live. For offline testing, AUC and squared error are tracked for classification tasks and regression tasks respectively. For conducting a live experiment, A/B testing was used. Hyperparameters were tuned with a combination of these testing results. The live experiment results on YouTube for MMoE is organized in figure 3. As we can see in figure 3, using the same model complexity by the same number of multiplications, results using MMoE show significantly better satisfaction metrics and also engagement metric. Paper explains that this meaningful improvement of MMoE over its counterpart can be explained with how the gating networks take input from the shared hidden layer. However, this method could potentially harm MMoE from modularizing input than compared to connecting MMoE layer directly from the input layer. Although having a one shared bottom hidden layer sound worse than having the input layer directly, the results of the live experiment speak otherwise. It shows that the gating networks can just as well effectively modularize the input information into the MMoE experts.

There are many other recommendations and ranking systems that are built on top of Neural Network Models. These models typically are suited for other traditional machine learning tasks such as CNN for computer vision and multi-headed attention for natural language processing. [1] However, often times, these architectures are not the best fit for recommending what video to watch next for following reasons. First, the ranking system takes input from multiple sources i.e., images, natural language and even just numbers, which wasn’t the popular case with traditional architecture. Also, many model architectures are not scalable and with multiple ranking objectives, it may hurt the accuracy of the system when they were designed to just capture a single type of information. Then there are other problems with the traditional neural network model architecture for ranking what videos to watch problem such as noise and how locally sparse the training data is and need of distributed training since the model takes input a large amount of training data.

In conclusion, ranking what videos to watch next is a complex problem that requires many aspects to be thoroughly considered. The presence of multiple ranking objectives to dealing with position biases in the implicit feedbacks, the ranking system implemented Multi-gate Mixture of Experts model architecture to utilize soft parameter sharing. [1] The result of experiment conducted on YouTube, the largest and most active video sharing platform in the world, show that there were significant improvements on engagement and satisfaction metrics. However, I believe there is not an end to solving the problem of recommending what videos to watch next. The authors could further research other or new types of neural networks model architecture for multiple ranking objectives for better performance. There is more recent architecture that provides better stability while not risking the output performance. There should also be better ways to address the known and unknown biases when recommending. Last but not least, the ranking system is already very complicated enough, to save the cost of the system, there should be more research on applying compression methods to save the cost and boost the performance of the video recommendation model on YouTube.

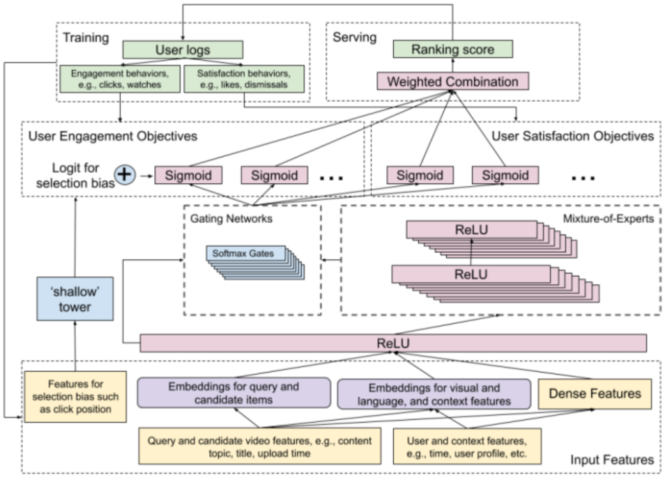
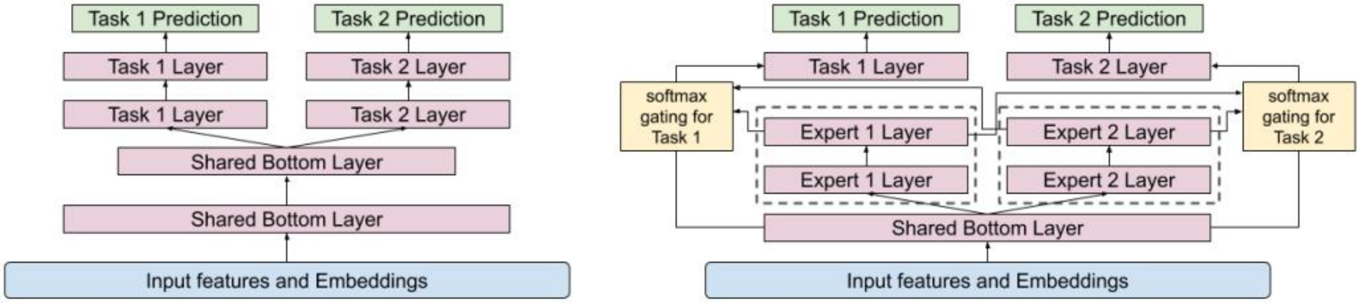

Training
User logs
Engagement behaviors, e.g., clicks, watches
Satisfaction behaviors, e.g., likes, dismissals
Serving
Ranking score
Weighted Combination
User Engagement Objectives
Logit for selection bias
Sigmoid
Sigmoid
...
User Satisfaction Objectives
Sigmoid
Sigmoid
...
Gating Networks
Softmax Gates
Mixture-of-Experts
ReLU
ReLU
'shallow' tower
ReLU
Features for selection bias such as click position
Embeddings for query and candidate items
Embeddings for visual and language, and context features
Dense Features
Query and candidate video features, e.g., content topic, title, upload time
User and context features, e.g., time, user profile, etc.
Input Features

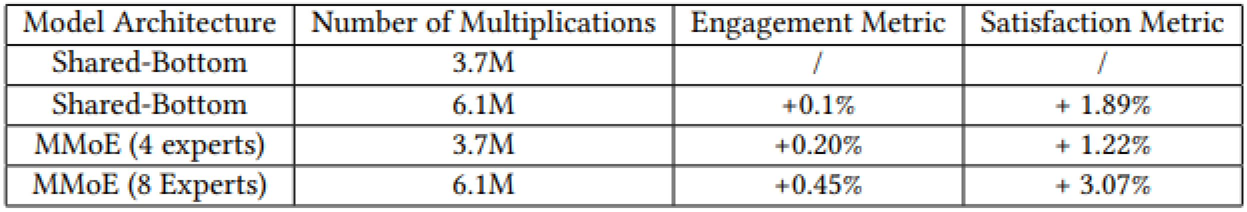
Figure 1. Model overview [1]



(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.

Figure 2. [1]



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%

Figure 3. [1]

References

[1] Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., Kumthekar, A., Sathiamoorthy, M., Yi, X., & Chi, E. (2019). Recommending what video to watch next. *Proceedings of the 13th ACM Conference on Recommender Systems*. https://doi.org/10.1145/3298689.3346997

[2] Bhatia, S. (2020, February 22). *A multitask ranking system: How YouTube recommends the next videos*. Medium. Retrieved November 3, 2022, from https://medium.com/@bhatia.suneet/a-multitask-ranking-system-how-youtube-recommends-the-next-videos-a23a63476073

[3] *Deep Neural Networks for YouTube recommendations*. (n.d.). Retrieved November 5, 2022, from https://static.googleusercontent.com/media/research.google.com/ko//pubs/archive/45530.pdf