Gene Horecka
Professor ChengXiang Zhai
CS 410 Text Information Systems
7 November 2021

Technology Review - Google's Multitask Ranking System

For this technology review, I am focusing on getting a better understanding of Google's multitask neural network ranking system, and how YouTube uses this technology to recommend videos to users at an industrial scale. I would also like to uncover the various challenges that the researchers at Youtube faced when building out their recommendation system.

My understanding of why YouTube decided to use multitask neural network architecture for their ranking system is mainly because of two overarching problems. First, there seem to be conflicting objectives that need to be taken into consideration when optimizing their recommendation system. Namely, a user may want to be recommended videos that other users have rated highly and have shared with their peers, in addition to the videos that the user has personally watched. Second, the data used for training the model contains some implicit bias. For example, a user might have clicked and viewed a video solely because it was ranked highly, and not because it was a video that the user generally likes the most. Thus, there needs to be special attention given to not creating a biased feedback loop that produces non-optimal recommendations that the user may not like.

Their approach to tackling these challenges is innovative and clever. I will try to describe my understanding of their approach at a high level. First, the recommendation system generates potential video candidates that can be shown to the user by looking at the video that the user is currently watching, along with their watch history, profile details, and other relative user information. The system then uses multiple candidate generation algorithms that each capture one feature of similarity between the query video and the candidate video. For example, one algorithm retrieves candidate videos by matching the topics of the query video, while another algorithm retrieves candidates based on how often a video has been watched together with the query video. Other algorithms take into account user history, context, and other aspects, and include methods such as collaborative filtering and co-occurrence graphs. After this step, there are a few hundred candidate videos that are ready to be ranked.

With the aforementioned problem statement and objectives in mind, the researchers have developed a neural network architecture to implement the ranking step. The model begins by first providing multiple input features and embeddings to a shared hidden layer. This is done so because feeding the input features directly into the

next layer, called the Mixture-of-Experts layer (discussed in the next paragraph), will drastically increase the cost of training.[1]

Next, the model utilizes the Mixture-of-Experts layer, or the MMoE layer, along with a Gating Network to solve the problem of the multiple objective functions. MMoE is basically a combination of Multi-Layer Perceptrons followed by ReLU activations, where each of the experts in the MMoE layer tries to learn a different feature of the input. The MMoE layer's output is then fed into the Gating Network. The outputs from the shared hidden layer and the Gating Networks are then fed into various objective functions to capture information such as user engagement and user satisfaction. Each of these objective functions is represented by a sigmoid activation function. While training, each of these objective functions looks at each of the experts via gates, and chooses one or more input features (aka experts) that are relevant to decide that objective function. An objective function could choose to share or not share experts with other objectives. This handles the problem of multiple conflicting objective functions. Now, I will shift the focus to demonstrate the part of the network that handles bias.

Explicit feedback data from the user about whether they liked the video recommendation or not is the most effective information that could be used to train a recommendation system. Since such types of feedback data are difficult to collect or are simply unavailable, implicit feedback is used for training. In this case, implicit feedback means that if a user clicks on a video recommendation, that implies the user likes the recommendation. However, this may not always be the case. Using such feedback data for training might not be the best because there can be bias in this data. Thus, extra care must be taken to remove these biases while training the model. Hence, a shallow tower is incorporated into the model architecture. This shallow tower is trained using features that contribute to bias, such as the position of the recommendation, and attempts to predict whether there is a bias component involved in the current sample. In addition, the selection bias output is provided as input to the engagement objective functions to train the network to remove these biases. Therefore, for the same sample, the model prediction is resolved into two components: namely a user-utility component from the main tower, and a bias component from the shallow tower.

Finally, I will touch briefly on the performance of the overall model architecture. The performance was tested by four and eight experts in the MMoE layer, which obtains the engagement and satisfaction matrices independently. Based on Table 1 represented in the paper, it can be observed that this model is able to perform better on both of the aforementioned metrics. Additionally, it can be seen that the introduction of a separate shallow tower to model bias serves to improve the engagement metrics, as shown in Table 2 in the paper.

---

[1] Reference: Recommending What Video to Watch Next: A Multitask Ranking System
https://daiwk.github.io/assets/youtube-multitask.pdf