A Multitask Ranking System

Introduction: Let us discuss about recommendation of neural network architecture which can improve you tube user recommendation of the videos he/she cane view next. Let us start by understanding how recommendation system works. A typical recommendation system works in two stages:

First, **Candidate selection** it generates the possible candidates that can be shown to the user. This step looks at the current video being watched by the user along with user history, user details, user search criteria etc.

Here in this step Candidate generation algorithm look for

Query video and candidate video

Query Video: What user has searched for?

Candidate Video: How often the video has been watched with the query video

Other aspects include user history context etc.

These 2 parallel algorithms create hundreds of records which needs to be ranked.

Criteria to look for similarity are Co-Occurrence graphs, collaborative filtering etc.

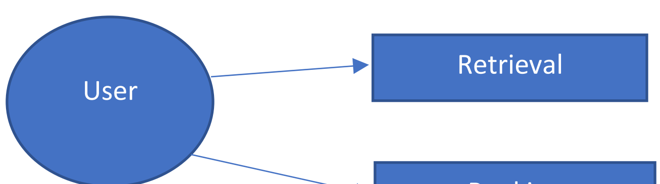
Second stage is to rank these selections.

This paper elaborates the ranking algorithm user to rank the candidates from the first stage.

Problem/Objective:

This is for YouTube (or any such content platform), the overall aim of a recommendation system is taking into account the video which a user is currently watching (along with user data) and recommend the next video that the user might watch and enjoy.

Two Stage System widely deployed in industries:



Recommender System

Challenges of building Multitask Ranking System

1. Different and conflicting objectives
2. Selection biases in the system

First Candidate selection, based on the user selection, user history, user details , user search etc.

1. Multiple Objective Functions: There are several different and conflicting objective functions that need to be optimized while solving the problem. The paper divides these objectives into two groups:

a. Engagement Objectives:

These can be measured by using data on clicks, the degree of engagement of the user while watching the recommended video, etc.

b. Satisfaction Objectives:

These can be measured by data in likes, shares, comments, rating, etc.

Both objectives contain binary classification tasks

(click or not, like or not, etc.) and regression tasks (time spent, rating given, etc.)

2. Removal of Implicit Bias:

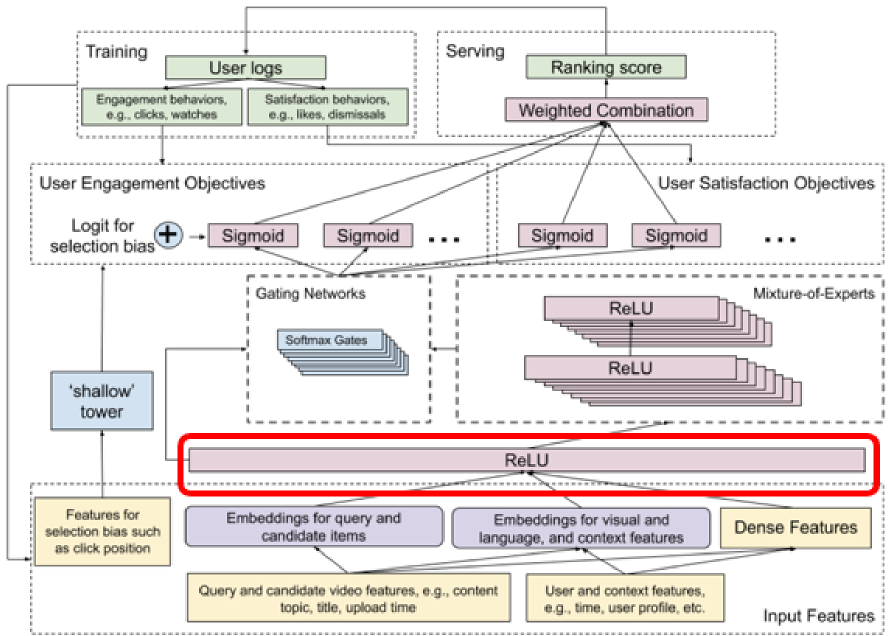
The data used for training the model contains some implicit bias. This is because a user

historically 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. So, if the model is trained using such data,

it will produce biased non-optimal recommendations which the user might not like.

With the above multiple objectives in mind, the researchers have designed the below network architecture:



The model starts with multiple input features and embeddings which are first fed into a shared hidden layer (highlighted above). This is done because supplying the input features directly to the next layer (called MMoE layer) directly significantly increases the cost of training.

Mixture-of-Experts and Network of Gates:

This paper aims to solve multiple objective function by employing a mixture of expert's layer along with a gating network. The output of the MMoE layer is fed into a Gating Network.

1. The output of these Gating Networks and that of the shared hidden layer are then fed into the various objective functions i.e. engagement as well as satisfaction.
2. During training, each of these objectives looks at each of the experts (through gates) and chooses one or more out of these experts (i.e. input features) that are relevant for deciding that objective function.

So, this takes care of the problem of the multiple conflicting objective functions. Next, let us to understand the part of the network that handles the bias.

Handling Bias

The ideal data to train a recommendation system would be an explicit feedback data from the user about whether they like the video recommendation or not. Since such kind of data is not available or is expensive to collect, implicit feedback data is used for training. Implicit feedback means that if a user clicks on the recommendation, it is believed that the user likes the recommendation. However, this may not be always true. A user might be clicking on a video just because it appears on top of the list of recommendations. Using such data for training might not be ideal as there is a bias in this data. Thus, this bias needs to be removed while training the model.

For this, a shallow tower is introduced into the model architecture. 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. The selection bias output is also fed an input to the engagement objectives to make the network learn to remove these biases. Thus, for the same instance, we factorize the model prediction into two components: a user-utility component from the main tower, and a bias component from the shallow tower.

Performance:

The model performance was tested with 4 as well as 8 experts in the MMoE layer and captures the engagement and satisfaction matrices separately. It is observed that this model can perform better on both the metrics.

Conclusion:

In this illustration, we started with the description of a few real-world challenges in designing and developing industrial recommendation systems, especially ranking systems. These challenges include the presence of multiple competing ranking objectives, as well as implicit selection biases in user feedback. To tackle these challenges, we proposed a large-scale multi-objective ranking system and applied it to the problem of recommending what video to watch next. To efficiently optimize multiple ranking objectives, we extended Multi-gate Mixture-of-Experts model architecture to utilize soft-parameter sharing. We proposed a lightweight and effective method to model and reduce the selection biases, especially position bias. Furthermore, via live experiments on one of the world's largest video sharing platforms, YouTube, we showed that our proposed techniques have led to substantial improvements on both engagement and satisfaction metrics.

References:

https://daiwk.github.io/assets/youtube-multitask.pdf

https://medium.com/@bhatia.suneet/a-multitask-ranking-system-how-youtube-recommends-the-next-videos-a23a63476073

https://www.youtube.com/watch?v=sr1KscMfBOY