Google's multitask ranking system

UIUC | CS 410: Text Information Systems
Susovan Samanta (susovan2)

In this paper, I will give an overview of a Multitask Ranking system built by Google. The basic idea of this system is to suggest a new video to the user according to the current video the user is watching. In this review paper I will try to explain the approach and challenges of this system.

In general, a recommender system works in two stages,

1. Candidate Generation: In this stage the recommender system will use multiple candidate generation algorithms. Each of these algorithms will capture one aspect of similarity between query and candidate video. to generate the list of candidates that can be recommended to the user. For example, 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. Other aspects considered include user history, context etc. This step results in a few hundred candidate videos which now need to be ranked. There are a lot of other ways of looking at similarity which include co-occurrence graphs, collaborative filtering, etc.

2. Ranking Stage: In this stage, the recommender has a few hundred candidates retrieved from the candidate generation and applies a model to rank and sort the candidates.

We will concentrate on the second part of the recommender system, i.e., Ranking stage.

The main goal of a video recommendation system is to take and analyze the video which a user is currently and recommend the next video that the user might watch and enjoy.

But design a real world and large-scale recommender system is full of challenges:

- Conflict of objectives - Sometimes we need to decide on conflicting goals for the recommender system. For example, we may want to recommend a video which user shared with friends in addition watching. This is further divided into two categories.

    - Engagement objectives – This is the degree of engagement of the user with the vide, for example, user clicks.
    - Satisfaction objectives – For example, user like a video or leaving a recommendation on the video etc.

- Implicit Bias – For example, a user might have clicked a video just because it was ranked high, but the user might not like it.

To address above challenges, a multitask neural network architecture ranking system was proposed.

Google's multitask ranking system

UIUC | CS 410: Text Information Systems
Susovan Samanta (susovan2)

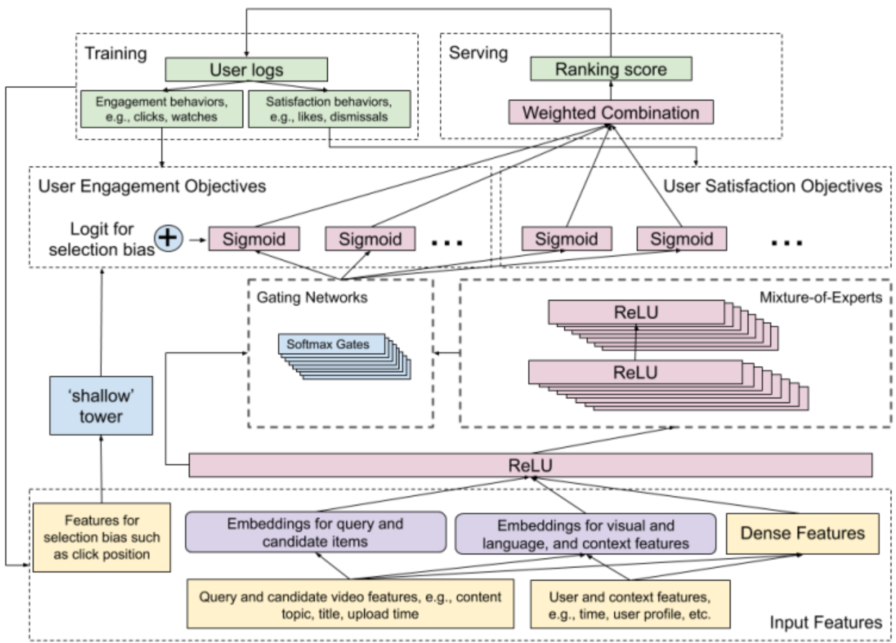

Figure 1: Model architecture of our proposed ranking system. It consumes user logs as training data, builds Multi-gate Mixture-of-Experts layers to predict two categories of user behaviors, i.e., engagement and satisfaction. It corrects ranking selection bias with a side-tower. On top, multiple predictions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

The first problem of conflicts objectives can be handled through the Multi gate Mixture of Experts (MMoE) model architecture. MMoE is a soft parameter sharing model which adapts the Mixture of Experts (MoE) structure. Main idea is to use the shared layers to avoid using one set of expert parameters for each objective. In this model it can use multiple sets of expert parameters controlled by a gating network. This is done because supplying the input features directly to the MMoE layer can significantly increases the cost of training.

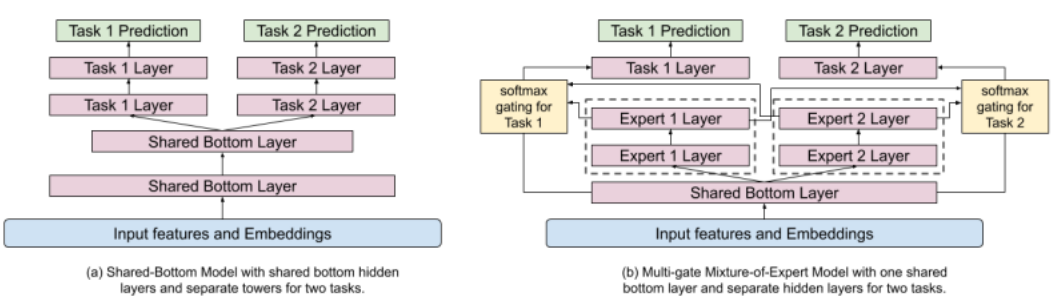

Figure 2: Replacing shared-bottom layers with MMoE.

Google's multitask ranking system

UIUC | CS 410: Text Information Systems
Susovan Samanta (susovan2)

Handling Implicit Bias:

To develop a large scale successful ranking system, we need lot of training data. One option is to use explicit feedback. But this will not be cost efficient and because of this reason implicit feedback is being used often in ranking system.

For the video recommender system, it's common that users will click the videos being displayed on the top of the list. It's not actually related the relevance and user preference.

The goal of this model is to eliminate such position base bias. The model prediction was divided into to components, a user-utility component from the main tower and a bias component from the shallow tower.

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.

This proposed model architecture is like the Wide & Deep model architecture proposed by Google researchers earlier. The concept of Wide and Deep Architecture was developed to make machines think more like humans. As explained in the Google AI Blog post, Humans memorize everyday events and then generalize those learning to apply to things we haven't seen before. Similarly, a Wide and Deep Architecture jointly trains a wide linear model (for memorization) alongside a deep neural network (for generalization) which can help the machine think somewhat like humans. In the above scenario, the deep part is the MMoE portion while the wide part is the shallow tower.