Kasam Dhakal

CS410 – Text Information System

Date: 11/01/2022

Google's Multitask Ranking System

At RecSys '19, Google researchers published a new article on a novel neural network architecture that can enhance suggestions for YouTube users. Firstly, it starts by generating potential candidates that the user could see. In this step, the user's current video, user history, user information, and other things are examined. The YouTube system uses a variety of candidate-generating algorithms for the initial stage, with each method capturing a different feature of similarity between the query video and the candidate video. One system, for instance, produces candidates by matching the themes of the search video. Based on how frequently the candidate movie has been seen alongside the search video, another algorithm retrieves candidate videos. User history, context, and other factors are among those taken into consideration. A few hundred candidate videos are produced because of this process, which now need to be ranked. There are other approaches to examining similarity, such as collaborative filtering, co-occurrence graphs, etc. The general goal of YouTube's recommendation system is to consider the video that a user is presently watching and suggest the next video that the user might appreciate. The problem statement may be straightforward to read, however, there are two fundamental obstacles to its solution: Multiple Objective: While resolving the issue, several various and competing objective functions must be optimized. These goals are divided into two sections in the paper: a. Engagement Objectives: These can be assessed using the information on clicks, the user's level of involvement during the suggested video, etc. b. Satisfaction Objective: Data in likes, shares, comments, ratings, etc. can be used to quantify
them. Both goals include binary classification and regression exercises. 2. Removal of Implicit Bias: Implicit bias exists in the data used to train the model. This is since historically; users might have chosen to click on and view a video just because it was highly ranked rather than because

they liked it. The model will therefore offer biased, suboptimal recommendations that the user might not prefer if it is trained with such data. The model begins with a variety of input features and embeddings that are initially fed into a shared hidden layer, which is constructed by the researchers with various aims in mind. This is done because providing the input characteristics to the next layer directly raises the training cost dramatically.

Multi-Layer Perceptron activations are combined with ReLU activations to form MMoE. Each expert in the MMoE layer attempts to learn a distinct feature of the input. A sigmoid activation function is used to represent each objective function. During training, each objective function examines each expert and selects one or more of those who are pertinent to determining that objective function. It is possible for an objective function to decide whether to share experts with the other objective functions or not. MMoE is essentially a combination of Multi-Layer Perceptron’s followed by ReLU activations. Each MMoE layer expert tries to learn a different aspect of the input. A Gating Network receives the MMoE layer's output as input. These Gating Networks' output as well as that of the shared hidden layer are then input into the various goal functions, such as engagement and satisfaction.

Each objective function is represented by a sigmoid activation function During training, each of these objectives looks at each of the experts and chooses one or more of these experts that are relevant for deciding that objective function. An objective function might choose to share or not-share experts with the other objective functions. Therefore, this resolves the issue of several competing objective functions. Let's next examine the section of the network that deals with prejudice.

An explicit user response indicating whether they like or dislike the recommended video would make the best training data for a recommendation engine. Implicit feedback data is utilized for training because such data is not readily available or is expensive to obtain. It is assumed that a user loves a recommendation if they click on it, which is known as implicit feedback. This might not always be the case, though. Simply because a video is at the top of the list of recommendations, a user can choose to click on it. Due to the bias in the data, using it for training may not be optimum. Therefore, in training the model, this bias needs to be eliminated. To do this, a brief tower is added to the model building. The shallow tower is trained with biascontributing features, such as the position of the recommendation, and it tries to foretell whether bias is present in the current instance. To help the network learn to eliminate these biases, the selection bias output is also provided as an input to the engagement objectives. As a result, we factorize the model prediction for the same instance into two components: a bias component from the shallow tower and a user-utility component from the main tower.

The MMoE layer's 4 and 8 experts were used to test the model's performance. It separates the engagement and satisfaction matrices. It has been found that this model can perform more effectively on both metrics. The addition of a separate shallow tower to account for model bias has also been conclusively demonstrated to assist enhance the engagement metric. The Wide & Deep model architecture that Google researchers earlier presented is like this one. Wide and Deep Architecture is an idea that was designed to help machines think more like people. Humans memorize commonplace occurrences and then extrapolate those lessons to apply to unfamiliar situations. Like this, a wide linear model and deep neural network are jointly trained via a large and extensive architecture, which can help the computer think somewhat like humans. The deep portion in the scenario is indeed the MMoE portion, and the wide portion is the shallow tower.

Reference:

1. Recommending What Video to Watch Next: A Multitask Ranking System (daiwk.github.io)
2. A Multitask Ranking System: How YouTube recommends the Next Videos | by Suneet Bhatia | Medium