MULI-MODAL FEATURE INTERACTIONS BASED FUSION MODEL FOR SHORT VIDEO UNDERSTANDING CHALLENGE

Yongquan Lu[1]*,*Liao Zhang*[2]*,*Xiulai Song*[1],*Haizhang Zou*[1]

[1] Guangxi Key Laboratory of Intelligent Processing of Computer Images and Graphic, Guilin University of Electronic Technology
[2] Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Technology, Xiamen University
aa147138@163.com, leochang@stu.xmu.edu.cn, julysxl@163.com, superzhazha@qq.com

ABSTRACT

The short video understanding challenge aims to better understand video content and recommend videos to users. The challenge provides multi-modal features include face features, video content features, title features and audio features which bring more complexity for recommender systems. Manually crafting these higher-order feature interactions by the engineer is time-consuming and can not generalize well. In this paper, we develop a model based on xDeepFM which incorporates these multi-modal features into a united framework and can be trained end-to-end. The proposed model can not only learn the high-order feature interactions explicitly and implicitly but also have good generalization ability. Experimental results on Byte-Recommend100 show that our model achieves competitive results.

Index Terms— Video content understanding, recommender systems, feature interactions, deep learning

1. INTRODUCTION

Combinatorial features play an important role in the recommendation system, which directly affect the accuracy of the recommendation system. Traditional recommendation systems construct the higher-order feature interactions by engineers, which have limited generalization ability and are time-consuming. Therefore, learning feature interactions automatically is necessary, but the existing related works [1, 2] can only learn implicit feature interactions, and feature interaction occurs at the element level rather than the vector level.

In recent years, deep learning achieved great success in speech recognition, computer vision and natural language processing. Many state-of-the-art works [3, 4, 5, 6, 7, 1, 8, 9, 2] also have done in recommendation systems based on deep learning. According to the way of feature construction, the deep learning based recommendation systems can be roughly divided into two categories: 1) Learning implicit features containing semantics from raw data automatically, such as extracting effective implicit features from text, image; 2) Learning feature interactions from multiple related features automatically. Feature interactions refer to learning the cross combination between two or more features.

In this paper, for achieving competitive results, we propose a deep fusion model which incorporates multi-modal features into a united framework and combines various models to learn high-order feature interactions automatically. Our model achieves 0.79363 AUC in the final private test set.

2. DEEP FUSION MODEL

In this section, we demonstrate our model in detail. Firstly, we show a modified xDeepFM [3] model which can accept multi-modal features as inputs. Figure 1 shows the architecture of the modified xDeepFM. Secondly, we specify how to integrate the all modified models to form a fusion model. Figure 2 shows the architecture of our fusion model.

2.1. Modified xDeepFM

Based on xDeepFM, we add some modules for extracting title, audio and video features. As Figure 1 shows, we first transform the sparse user interaction data into sparse features and transform the dense user interaction data into dense features. Secondly, we transform the title data into sequence features. Thirdly, we use an embedding layer to reduce the dimension of these features and obtain embedded feature vectors. Finally, these embedded feature vectors are fed into three modules: a linear layer, a Compressed Interaction Network (CIN) layer and a plain Deep Neural Network (DNN) layer. For video and audio data, we firstly use Principal Component Analysis (PCA) [10] to extract the principal components of the original features and reduce feature dimensions. Secondly, we use the whitening operation to reduce the correlation between features. Then, we use an embedding layer to

* Means same contribution

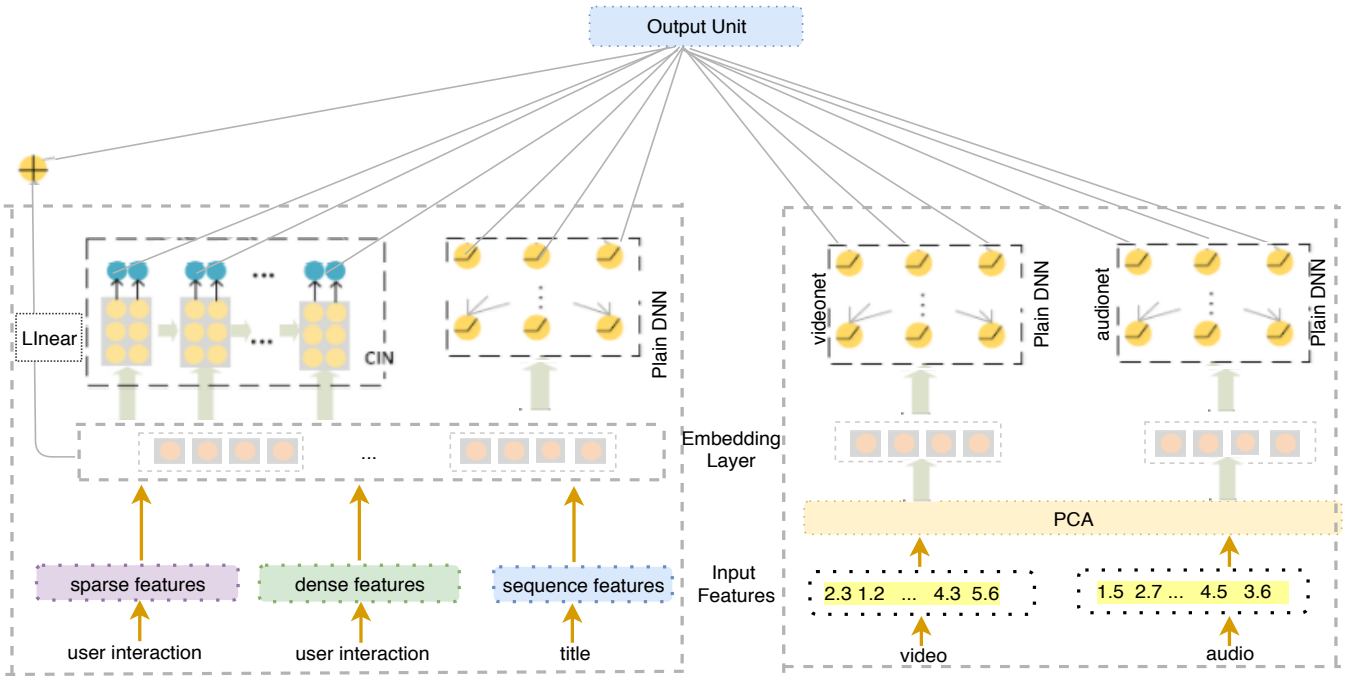

Fig. 1. The architecture of the modified xDeepFM

obtain embedded feature vectors for audio and video features. Finally, the embedded audio feature vectors and video feature vectors are fed into two different plain DNN layer for further feature extraction. Finally, all features are fused together and trained in a multi-task way.

2.2. Deep fusion model

In order to extract features from multiple dimensions and achieve better performance, we first modify the following models: DeepFM [5], AutoInt [6], PNN [7], NFM [8], NFFM [9], DCN [2] and FNN [1] as modifying the xDeepFM. And then, we fuse these modified models together to form a fusion model. We introduce some of these models briefly as follows: DeepFM consists of two parts: the neural network part and the factorized part, which are respectively responsible for the extraction of low-order features and high-order features. These two parts have the same input. AutoInt uses the multi-head self-attention mechanism for automatic cross-learning of features to improve the accuracy of prediction tasks. PNN proposes a product layer to capture interactive patterns between interfield categories and a fully connected layers to explore high-order feature interactions. NFM combines Factorization Machine (FM) with a neural network to improve the ability of capturing multi-order interactive information between features. DCN introduces a new crossover network which can learn the features interaction more effectively. We combine the above modified models to build our deep fusion model and train our model in a multi-task way and end-to-end.

3. FEATURE ENGINEERING

In this section, to further boost our model performance, we extract some useful features by our experiences and domain knowledge.

Portrait of users. The essence of building features is to portray the user's portrait by modeling the user and then use the data to express the user. Based on business considerations, we believe that the user interest is a relevant and important factor in whether the user browses and likes the video. Therefore, how to portray user interest is one of the key issues. We restore the scene when the user browses the video and imagine what the user might do with the favorite video. We assume that when users hit like for a video, they may have an interest in the author or pay attention to the channel. In addition, the user may be more interested in videos that are uploaded in the same city. So we construct the interaction features, named "user concerns" which contains following channels, following authors, following cities. As Table 1 shows, the user has distinct preferences for different authors. As can be seen in Table 2, the user has distinct preferences for different channels.

Time feature extraction. The dataset of this challenge contain relative time which can not be used to estimate the time of the uploaded video. But, after analysing the number of video item, we find that if the number of video item is relatively small during one period, so the current time maybe is night. We can extract this time information as a new feature. We can also obtain another time feature that the author is ac-

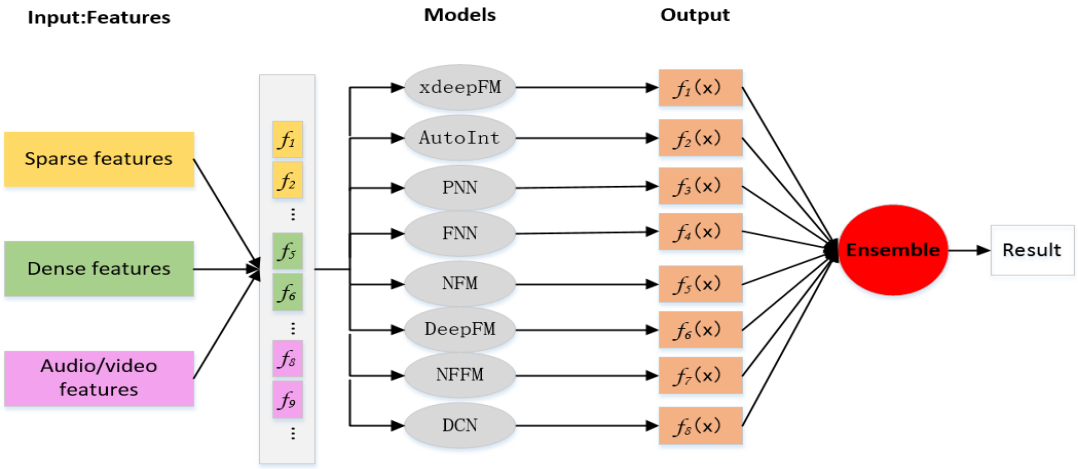

Fig. 2. The architecture of our fusion model.

Table 1. The influence of following author

uid	author_id	uid_author_id_item_nunique	finish_mean
10	111968	41	0.000
10	17444	40	0.275
53	2635	39	0.410
53	52568	31	0.000
72	5312	45	0.822
75	17767	150	0.000

Table 2. The influence of following channel

uid	channel	finish_mean	like_mean
86	0	0.526	0.005
86	1	0.037	0.000
125	0	0.460	0.001
125	1	0.017	0.000
243	0	0.410	0.003
243	3	0.138	0.000

tive in a certain period of time if the author uploaded many videos.

Facial feature extraction. If there are faces appearing in a video, this video is easier to be browsed and get likes. So we add face features which contain face numbers, face sex and face beauty level to our model.

Title feature extraction. For each video item, we add a sequence feature which contains title feature to our model.

Audio, video feature extraction. We do not feed the audio or video feature to our model directly, because these 128-dimensional data contains some noises which will lead to poor performance. Therefore, we firstly use PCA to extract the principal components and experimentally reduce the dimension to 64-dimension. In addition, we reduce the correlation between features and make all features have the same variance by the whiten operation. Then, we use two DNN layers with 256*256 dimensions to extract the features of the audios and videos respectively.

Table 3. The improvements achieved by adding the features

model	AUC	improvement
xDeepFM_baseline	0.7107	-
Add user portrait features	0.7146	0.0039
Add time features	0.7164	0.0018
Add face features	0.7186	0.0022
Add title features	0.7206	0.0020
Add audio and video features	0.7307	0.0101

Table 4. The results of each single model

model	Finish	Like
xDeepFM	0.7307	0.9243
AutoInt	0.7306	0.9222
PNN	0.7291	0.9237
FNN	0.7296	0.9204
NFM	0.7306	0.9261
DeepFM	0.7276	0.9223
NFFM	0.7289	0.9287
DCN	0.7298	0.9238

4. EXPERIMENTS

4.1. Datasets and evaluation criteria

This paper uses the dataset collected by ByteDance Company. This dataset, named Byte-Recommend100M, consists of tens of thousands of different users and 100 Millions of different videos. Multi-modal features in Byte-Recommend100M in-

clude face features, video content features, title features and audio features, which are all in form of embedding vector. AUC (area under ROC curve) is adopted as the challenge metric. The participants of the challenge should predict the click (finish+like) probability on each item of test dataset. The weight for 'finish' and 'like' towards final scores are: final score = 0.7 * finish + 0.3 * like.

4.2. Implementation details

In our scheme, for every single model, we set the embedding size to 8, the dimension of hidden layers to 256*256 and the dimension of hidden layers of the cross-compressed network to 256*256. The dimension of the multi-task network is 128, the weight decay rate is 0.00001 and the seed is set as 1024.

Based on the initial settings of our model, we conduct the experiments to verify the effectiveness of our constructed features. Table 3 shows the improvements of adding the different features. From Table 3, we find that the all features that we added to our model bring obvious improvements comparing with the baseline model (xDeepFM). The user portrait feature has improved our model by nearly 0.004, and it shows that the analysis of user interaction data is useful. The time feature and the facial feature improve the AUC of our model 0.001 and 0.002, respectively. The title feature improve our model by 0.002. We observe that adding audio and video features generates higher AUC improvement (+0.01) than adding other features. It demonstrates that the processing of audio and video features are effective.

We conduct the experiments to show that all modified models have comparable precision for predicting the test set. The results of every single model are shown in Table 4. From Table 4, we can find that all modified models shows similar precision. For fusing multiple models to achieve better performance, we construct our fusion model in a weighted manner, and the weights are determined by the corresponding performance. The fusion framework is shown in Figure 2. In the end, our fusion model obtains 0.79363 AUC in the final test set data.

5. CONCLUSION

By participating in short video understanding challenge, we propose a model that incorporates multi-modal features, including video features, text features, and audio features, as well as user interaction behavior data. With careful architecture design, accurate portrait of the users, good dimensionality reduction for audio and video features, a rich fusion for features and models and careful parameter tuning, our deep fusion model presents competitive performance on the final test set.

Due to the time limit, there are still some ideas in our plan that have not been implemented. We hope to implement and test our plan in the future. The ideas are as follows: (1) When processing video and audio features, we hope to use Gated Recurrent Unit (GRU) to extract more powerful features. (2) For the missed audio and video data item, we want to use clustering to find some similar items and fill the missed data with the average value of these similar items. (3) We plan to build user interest trends based on browsing history sequence and the attention mechanism for better results.

6. REFERENCES

[1] Weinan Zhang, Tianming Du, and Jun Wang, "Deep learning over multi-field categorical data," in *European conference on information retrieval*. Springer, 2016, pp. 45–57.

[2] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*. ACM, 2017, p. 12.

[3] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun, "xdeepfm: Combining explicit and implicit feature interactions for recommender systems," in *ACM SIGKDD*. ACM, 2018, pp. 1754–1763.

[4] Wu Gang, Viswanathan Swaminathan, Saayan Mitra, and Ratnesh Kumar, "Context-aware video recommendation based on session progress prediction," in *IEEE ICME*, 2017.

[5] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He, "Deepfm: A factorization-machine based neural network for ctr prediction," 2017.

[6] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang, "Autoint: Automatic feature interaction learning via self-attentive neural networks," 2018.

[7] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang, "Product-based neural networks for user response prediction," in *ICDM*. IEEE, 2016, pp. 1149–1154.

[8] Xiangnan He and Tat-Seng Chua, "Neural factorization machines for sparse predictive analytics," in *ACM SIGIR*. ACM, 2017, pp. 355–364.

[9] Li Zhang, Weichen Shen, Shijian Li, and Gang Pan, "Field-aware neural factorization machine for click-through rate prediction," *arXiv preprint arXiv:1902.09096*, 2019.

[10] Ian Jolliffe, *Principal Component Analysis*, pp. 1094–1096, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.