ResFusion: A Residual Learning based Fusion Framework for CTR Prediction

Junmei Bao[1], Yangguang Ji[1], Yonghui Yang[1], Le Wu[1,2,*], and Ruiji Fu[2]
{hfut.baojunmei, jyguang1997, yyh.hfut, lewu.ustc}@gmail.com, rjfu@iflytek.com
[1] School of Computer Science and Information Engineering, Hefei University of Technology
[2] State Key Laboratory of Cognitive Intelligence iFLYTEK

Introduction

- **CTR prediction tasks** have been widely deployed in many online recommendation and advertising platforms.
- **Mainstream CTR models can be divided into two categories**: **the traditional machine learning models** (e.g., GBDT) that learn the linear feature combinations for prediction, and **deep learning based algorithms** (such as DeepFM) for modeling the compl-ex and sparse feature correlations.
- These single models suffer from **one-side problem in feature learning**. Currently, some researchers try to combine the above two types of models to enhance the prediction power of CTR models and have achieved great success.
- Most fusion models proposed based these two kinds of models **can't explicitly utilize the different prediction power** of these two kinds of models.
- We proposed a framework that fuses the two types of models based on ResNet.
  - We use ResNet to **effectively combine the two types of models** which is different from the existing fusion models.
  - Since our second model learns on the basis of the first model, it is also easier to train with **faster convergence**.

Overall architecture

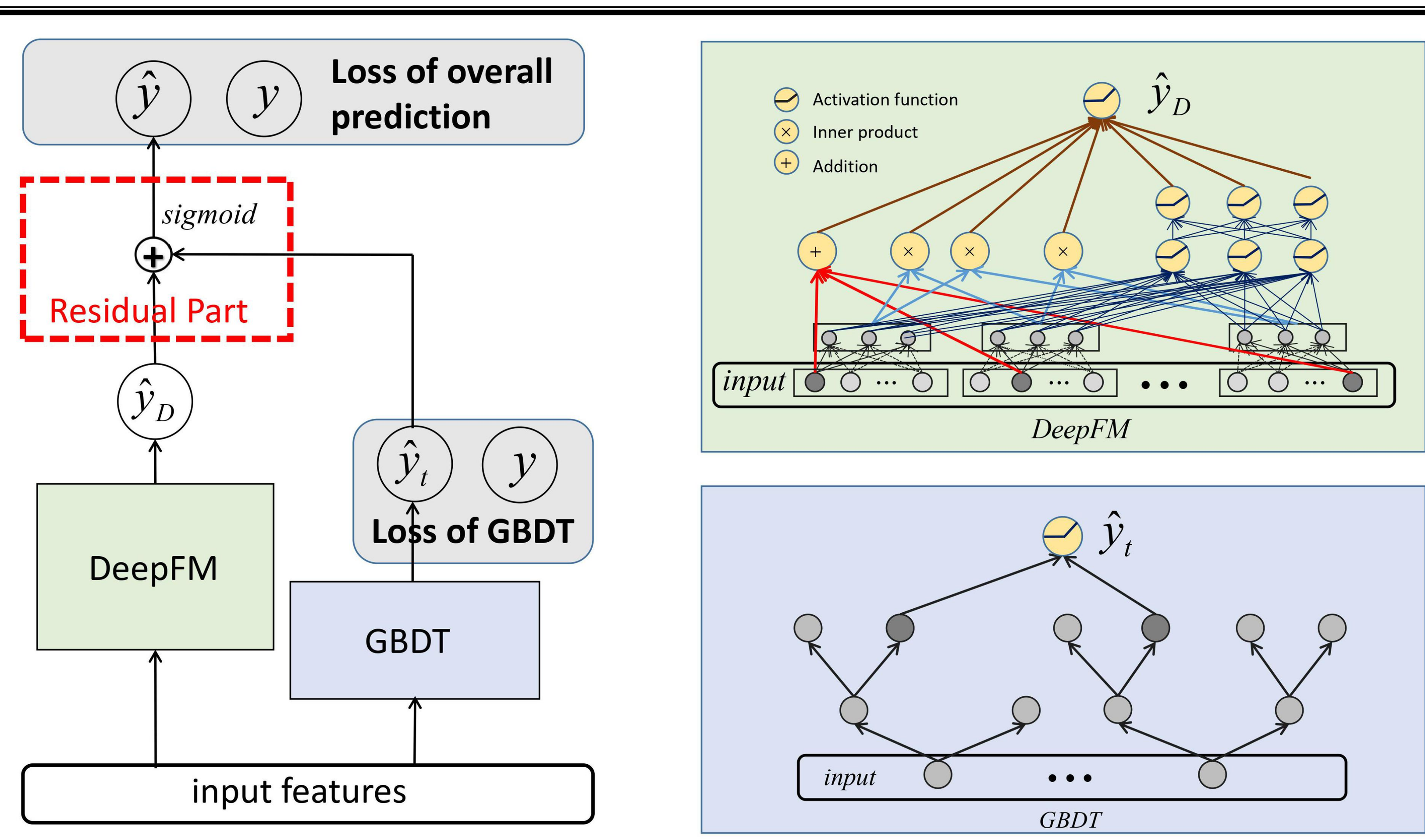


Fig.1. The overall architecture of our proposed framework

Model Training

The overall process of our framework can be divided into three steps:

- We **train a strong classifier called GBDT** and obtain the prediction score of GBDT $\hat{y}_t$;
- We **calculate the residual** between the true label y and the GBDT's output $\hat{y}_t$;
- We **train a DeepFM try to fit the residual values** and get the predicted value of the DeepFM $\hat{y}_D$;
- We **sum the output of two components**: $\hat{y}_t$, $\hat{y}_D$, and obtain the final prediction score $\hat{y} = \hat{y}_t + \hat{y}_D$.

Model Disscussions

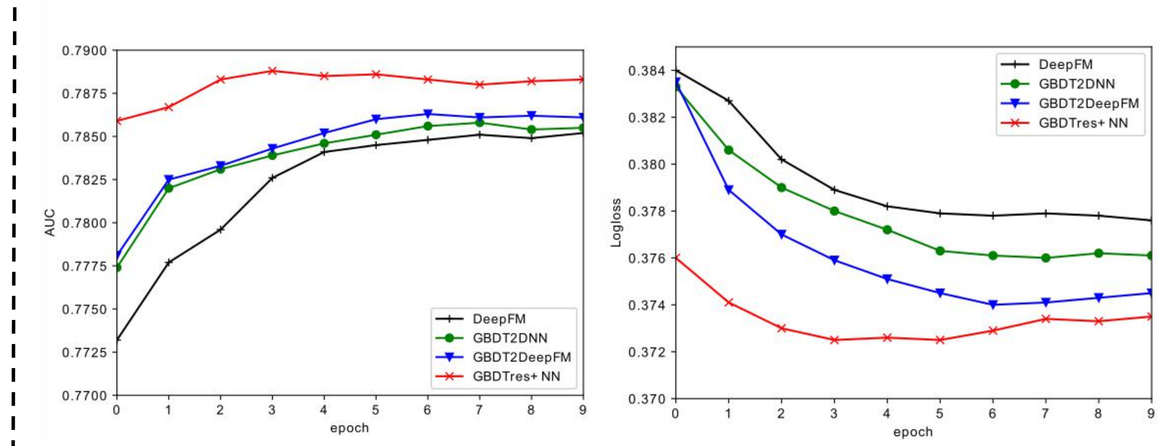
The **key point** of our framework is that **we use the residual values between the GBDT's output and the true label as the new label to train the DeepFM.**

- **Rapid Convergence**: In our framework, the second model is trained on the basis of the first model, so it needs to learn less content until reaching convergence with relatively faster speed.
- **Model Generalization**: Through the repeated joint learning of two completely different learning mechanisms, our framework can learn the hidden information more generally under the input data.
- **Model Flexibility**: Our framework is artfully sequentially links the two model as the fuse process during the model training process and can also be extended with feature fusion mothods.

Experiments

Table 1. The statistics of the three datasets

Dataset	Avazu	Cretio	Zhihu
Total instances	40M	45M	2M
Train	36M	40.5M	1.8M
Test	4M	4.5M	0.2M
Numerial features	0	13	131
Categorical features	23	26	18



AUC of different models   Logloss of different models

Fig. 2. The convergence speed comparison on various fusion models

Table 2. AUC and Logloss comparisons for different models

Models	*Avazu*		*Cretio*		*ZhiHu*	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
LR	0.5453	0.4554	0.5690	0.5650	0.6122	0.5613
FM	0.7759	0.3820	0.7674	0.5052	0.7319	0.4102
GBDT	0.7608	0.3895	0.8009	0.4495	0.8390	0.3706
DeepFM	0.7852	0.3779	0.7959	0.4569	0.7712	0.3787
GBDT+LR	0.7634	0.3877	0.8025	0.4423	0.8405	0.3700
GBDT2DNN	0.7858	0.3761	0.8031	0.4417	0.8409	0.3699
GBDT2DeepFM	0.7863	0.3741	0.8037	0.4412	0.8417	0.3702
$\overline{GBDT + DeepFM}$	0.7860	0.3767	0.8022	0.4367	0.8411	0.3707
NNres+GBDT	0.7872	0.3726	0.8030	0.4379	0.8420	0.3702
GBDTRes+NN	**0.7921**	**0.3720**	**0.8065**	**0.4348**	**0.8676**	**0.3679**

Table 3. AUC and Logloss comparisons with different number of iterations K.

Residual iteration	*Avazu*		*Cretio*		*ZhiHu*	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
K = 0	0.7608	0.3895	0.8011	0.4495	0.8390	0.3706
K = 1	0.7921	0.3720	0.8069	0.4350	0.8676	0.3679
K = 2	**0.7925**	**0.3717**	**0.8073**	**0.4341**	**0.8684**	**0.3679**
K = 3	0.7922	0.3720	0.8071	0.4351	0.8680	0.3680

Conclusion

- Our proposed framework alleviates the **challenge** that the **existing CTR models cannot fully learn from data with both sparse category and dense numerical features.**
- It gains performance improvement for **some advantages which we mentioned in the model discussions part**
- Extensive experimental results on three real-world datasets show the **effectiveness** of our proposed framework.

认知智能国家重点实验室
STATE KEY LABORATORY OF COGNITIVE INTELLIGENCE