Click-Through Rate Prediction Algorithm Based on Modeling of Implicit High-Order Feature Importance

Qing Yang[1], Ning Li[1], Shiyan Hu[1], Heyong Li[1], Jingwei Zhang[2*]

[1] Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, China
[2] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, China
gtyqing@hotmail.com, 1479702018@qq.com, 541297760@qq.com, 3098478113@qq.com, gtzjw@hotmail.com

Abstract

Click-through rate (CTR) prediction plays a central role in online advertising and recommendation systems. In recent years, with the successful application of deep neural networks (DNNs) in many fields, researchers have integrated deep learning into CTR prediction algorithms to model implicit high-order features. However, most of these existing methods unify the weights of implicit higher-order features to predict user behaviors. The importance of such features of different dimensions for predicting user click behaviors are different. Base on this, we propose a prediction method that dynamically learns the importance of implicit high-order features. Specifically, we integrate the output features of deep and shallow components, and adaptively learn the weights of implicit high-order features from among all features through the designed attention network, which effectively capturing the deep interests of users. In addition, this framework has strong versatility and can be combined with shallow models such as Logistic Regression (LR) and Factorization Machines (FMs) to form different models and achieve optimal performance. The extended experiment is conducted on two large-scale datasets, AVAZU and SafeDrive, and the experimental results show that the performance of the proposed model is superior to that of existing baseline models.

Keywords: Recommendation algorithm, Click-through rate, Implicit high order feature, Attention network

1 Introduction

As the amount of interconnected data continues to grow, how to recommend valuable information to users has become a challenging problem. Among the many problems with recommendation systems, how to accurately predict the items clicked by users has become one of the core issues. This problem has attracts an increasing number of researchers in academia and industry. In recent years, many CTR prediction algorithms have been proposed, which are mainly divided into three aspects: CTR prediction models based on shallow algorithms, CTR prediction models based on deep networks, and CTR prediction models based on hybrid structures. The shallow algorithm focuses on modeling the features within the second order of users and items; the deep algorithm is a CTR prediction algorithm based on deep neural networks, which captures high-order features of users and items; the hybrid CTR recommendation algorithms combine shallow and deep models to jointly model low-order and high-order features of users and items.

The main task of the CTR prediction problem is to predict the probability of users clicking on an item, thereby generating a list of recommended items. Most of the current research work is to capture the interaction characteristics of users and items for modeling, and then carry out downstream tasks. Logistic regression (LR) is the most classic CTR prediction model. The model has a simple structure, is easy to implement in parallel, and has strong interpretability, but it cannot handle the nonlinear relationships between targets and features [1]. The CTR prediction task is a typical nonlinear data processing task where the data are sparse and contain multiple types of objects with high-dimensional complex relationships. To solve this problem, researchers have proposed some improved models to address high-order complex feature interactions; for example, a gradient boosting decision tree (GBDT) [2], Factorization machines (FMs) [3] and the Field-aware Factorization Machines (FFMs) [4]. As most of the features in a recommendation system are large-scale discretized feature, the representation abilities of these shallow models are still insufficient in the face of large-scale data.

With the rapid development of deep neural models, many deep models have been proposed to learn the interactions between low-order features and high-order features from the original features, examples include deep neural networks (DNNs), Deep&Cross network (DCN), Wide&Deep, DeepFM, etc. [5-8]. In CTR prediction, the DNN can automatically learn the interactions between high-order features and generate all interactions implicitly. The Deep&Cross network (DCN), which retains the advantages of the DNN model and also introduces a novel cross network that is more efficient in terms of learning certain bounded-degree feature interactions. Wide&Deep models first-order linear features and implicit high-order features simultaneously, and the model combines the “wide” and “deep” structural characteristics of DNNs. DeepFM improves the Wide&Deep model by replacing the “wide” component with an FM model so that the resulting model has the ability to model second-order interactive features. In recent years, the hybrid CTR recommendation algorithm based on the shallow model and the deep model is the mainstream CTR recommendation algorithm. Under the premise of the hybrid recommendation algorithm, various forms of feature interaction emerge in an

*Corresponding Author: Jingwei Zhang; E-mail: gtzjw@hotmail.com
DOI: 10.53106/160792642022092305016

endless stream. Interaction, multiple interactions of high-level features, interaction of first- and second-order features and high-level features, etc. Different interaction forms can achieve different feature fusion effects.

Table 1. Comparison of recommendation algorithms based on user CTR predict

	User and item feature mining	Feature interactions	Neural network structure optimization	Implicit higher-order feature weight optimization	Related references
Feature-based mining algorithms	✓	×	×	×	[12], [13], [14], [15]
Feature-based combination algorithms	✓	✓	×	×	[17], [18], [19], [20], [21], [22]
Structure optimization based on neural networks algorithms	✓	✓	✓	×	[23], [28], [31]
Our model	✓	✓	✓	✓	

With the widespread application of attention mechanism in intelligent algorithms, researchers have applied attention mechanism to CTR prediction algorithm to enhance the representation ability of feature information. This method can capture the potential interaction information between users and items to improved prediction performance [9-11]. However, existing methods often unify the weights of implicit high-order features to predict user behaviors, there are few ways to use attention mechanisms to solve problems regarding the imbalance among the importance of higher-order features.

In this paper, we design an attention network for a hybrid CTR prediction recommendation system to solve this problem of feature imbalance, thereby modeling the importance of high-order features. The attention network adopts the form of a combination of a linear function and an activation function, takes low-order eigenvectors and high-order implicit eigenvectors as inputs, and outputs the modeling weights of high-order implicit eigenvectors. Theoretically, useful and useless high-level implicit features can be better distinguished.

In summary, the main contributions of this paper are as follows:

- We propose a CTR prediction method based on implicit high-order feature importance modeling. This method uses an attention network to model the weight of each dimension of high-order features at the element level, which makes up for the deficiency of unified high-order features parameters into the model.
- We design an attention network to model the importance of high-order features, and flexibly integrate the attention network into different hybrid models to form two recommendation methods, which can be applied to different recommendation situations.
- We conduct extensive experiments on two real-world datasets, and the results show that the performance of our model is superior to that of several baseline models.

2 Related Work

At present, the research directions involving recommendation systems based on user CTR prediction are mainly divided into three kinds of algorithms: feature-based mining, feature-based combination and structure optimization based on neural networks. Among the studies based on feature mining [12-16], Huang et al. [12] introduces a new model of FO-FTRL-DCN, mainly based on the prestigious model of Deep&Cross Network, using FTRL to build a model to mine features. Liu et al. [13] proposed using convolutional neural networks to obtain image features of items and integrate them into neural networks to make recommendations. Zhu et al. [14] proposed a deep attention neural network for news recommendation. The developed model uses a convolutional neural network and an attention mechanism to gather the users' interest features and capture the users' click sequence features, and it then combines these features to perform recommendations. To solve the processing difficulty inherent in long user time series data, Qin et al. [15] proposed the use of a learnable method to search for the most relevant user behaviors from the entire user history. Then, these retrieved behaviors are input into a DNNs to make the final CTR prediction.

Although many studies have been sufficient for user and item feature mining, they have not fully considered the user or item feature combinations or combination optimization. Based on this, a large number of rese archers have proposed CTR prediction algorithms based on feature combinations [7, 17-22]. For example, Liu et al. [17] proposed a GCN-int model based on the interaction of Graph Convolutional Network to mine meaningful feature combinations. Bian et al. [18] propose a Co-Action Network (CAN) to approximate the explicit pairwise feature interactions without introducing too many additional parameters. To automatically identify the important interactive features in an FM model, Liu et al. [19] proposed the automatic feature interaction selection (AutoFIS) algorithm, which can automatically eliminate redundant feature interactions during the training process. Finally, useful feature interactions are screened through an attention gate. To model the meaningful combinations of high-order features, Song et al. [20] proposed the automatic feature interaction learning (AutoInt) model, which uses multi-head attention to model high-order feature interactions from different angles and automatically learns the high-order interactions among the input features. At present, the process of modeling feature interactions is based on their volumes and Hadamard products without considering the weights of feature interactions. Based on this, Huang et al. [21] proposed the feature-aware bilinear

network (FiBiNET), for modeling feature importance and feature interactions through a bilinear coupling function. The model learns the importance of embedded features through the SENET network and learns feature interactions through a bilinear coupling function. Considering that the features generated by DNNs are implicit high-order features and that the existing networks cannot determine the direct interactions between low-order and high-order features, Lian et al. [22] proposed a recommendation model that combines implicit and explicit feature interactions and proved the effectiveness of the method on large-scale datasets.

The above research has fully solved the possible problems related to feature interactions, but the structures of neural networks have not been fully optimized, leading to certain restrictions on the performance of the resulting models. Due to this shortcoming, on the basis of related research, many researchers have proposed various new network variants to address specific problems [23-31]. For example, to further model the interactions between and within feature domains, Chen et al. [23] proposed a two-way interaction mapping model called FLEN based on shared domains; FLEN can simultaneously capture the interaction information between and within feature domains. To model the importance of different second-order interaction features, Xiao et al. [28] proposed an FM model based on attention that fully explores useful feature interactions. The existing research mainly focuses on the representation of user characteristics, and rarely studies the correlations between users and items. Based on this, Lyu et al. [31] proposed calculating the correlation rankings of users and items by calculating the inner product of the corresponding features in the mapping space between these users and items. This approach has been proven to have excellent performance on industrial datasets.

Although the above research has solved the problem of CTR prediction from various angles, a large number of studies based on DNNs have not modeled the weights of implicit high-order features, which has limited the performance of the corresponding models to a certain extent. Aiming at the problem that the existing hybrid CTR prediction algorithms do not consider the high-order features with different weights at the element level, this paper designs a unique attention unit to model the high-order features output by the deep neural network, capture the weights of the implicit high-order features, and improve the performance of click-through rate prediction. Moreover, we integrate the designed attention network into two classical frameworks to form a new method that can be applied to different recommendation situations. Table 1 shows the comparison between our model and other recommendation algorithms based on user CTR prediction.

3 CTR Prediction Model Based on an Attention Network

To solve the above problems, we design a CTR prediction model based on an attention network, which can better learn the importance of implicit high-order features, thereby effectively capturing user interest. The overall structure of the model is shown in Figure 1. In this section, we introduce the architecture and implementation methods of our proposed model in detail.

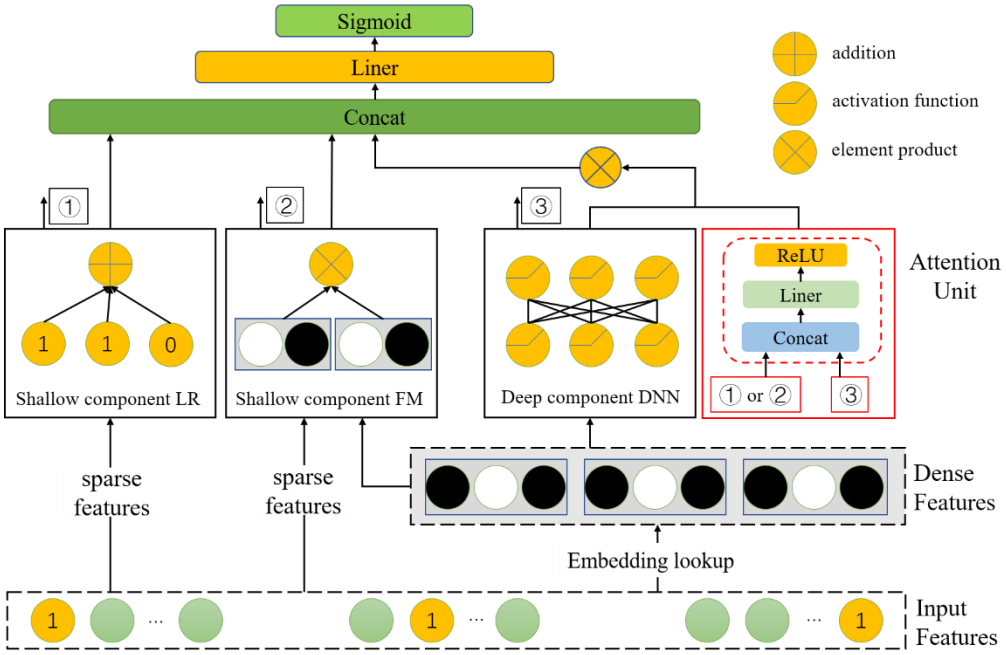

Figure 1. Structure diagram of the CTR prediction model based on the attention network

3.1 FM Unit

FM is a recommended algorithm model proposed by Rendle S [3] for learning feature interactions; it can model first-order linear features and second-order interactive features.

3.1.1 Embedding Layer

In the CTR prediction problem, the input features are converted into one-hot encoding, and the converted features have higher dimensions. To reduce the dimensionality of such

features, it is necessary to introduce embedding technology. The embedding layer is a fully connected layer, that can map an original feature to a dense vector. The attribute characteristics of users and items are represented as a sparse vector X, which is specifically represented as:

$$X = [x_1, x_2, ...x_m]. \quad \textbf{(1)}$$

where $x_i \in \mathbb{R}^{n_i \times 1}$, the subscript m represents the number of feature domains, x_i of domain i represents a feature vector that has been one-hot encoded, the feature embedding matrix v_i is used to convert the original encoding vector x_i into a low-dimensional dense vector, and the conversion process is expressed as follows:

$$e_i = v_i \bullet x_i. \quad \textbf{(2)}$$

where $v_i \in \mathbb{R}^{l \times n_i}$, $e_i \in \mathbb{R}^{l \times 1}$, l represents the dimensionality of the low-dimensional vector, and n_i represents the vector dimensionality of the domain i. e_i represents the dense feature vector of domain i after undergoing dimensionality reduction. The superposition of the low-dimensional vectors of all domains is expressed as:

$$E(x) = [e_1, e_2, ...e_m]. \quad \textbf{(3)}$$

After converting the embedding layer, the feature vector is mapped to the low-dimensional space $\mathbb{R}^{l \times m}$, and at the same time, the basic semantics of the original feature vector x_i of the domain are well preserved.

3.1.2 Feature Interaction Layer

Component 1 shown in Figure 1 represents a logistic regression (LR) component, and this part is represented as:

$$y_{LR}(x) = \sum_{i=1}^{m} w_i \bullet x_i. \quad \textbf{(4)}$$

where $w_i \in \mathbb{R}^m$ represents the weight of the first-order linear feature, and $x_i \in \mathbb{R}^{n_i \times 1}$ represents the original feature vector of domain i. The second part shown in Figure 1 is the FM component, which is composed of a first-order linear feature unit and a second-order inner product feature unit; this part is expressed as follows:

$$y_{FM}(x) = \sum_{i=1}^{m} w_i \bullet x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle e_i, e_j \rangle. \quad \textbf{(5)}$$

The first part of formula (5) is the LR component, and the second part $\langle e_i, e_j \rangle$ represents the inner product interactions between the features after the original feature x_i is transferred to the low-dimensional space. An FM can model first-order linear features and second-order interactive features in sparse and low-dimensional spaces, while FM components can only model features within the second order. This article refers to the FM and LR as shallow components. For high-order features, DNNs need to be used to model the features.

3.2 Deep Component

The deep component is essentially a forward feedback neural network, as shown in the third part of Figure 1, which is used to capture implicit high-order feature interactions, and the input data of the network are the mapping results of the embedding layer: $a^{(0)} = E(x)$. The process performed by the forward network is:

$$a^{(l)} = \sigma\left(w^{(l)} a^{(l-1)} + b^{(l)}\right), \quad \textbf{(6)}$$

$$y_{DNN}(x) = a^{(l)}. \quad \textbf{(7)}$$

In formula (6), l is the number of layers contained the neural network, σ is the activation function, $a^{(l)}$ is the output of layer l, $w^{(l)}$ is the weight parameter of layer l, and $b^{(l)}$ is the bias parameter of layer l. After completing the above process, the deep component outputs the implicit high-order component $y_{DNN}(x)$ (formula (7)).

3.3 Attention Network Components

After obtaining the output feature vectors of the shallow component and the deep component, they are combined in parallel to predict the CTR. In the shallow components, LR models first-order linear features, an FM models first- and second-order feature interactions, and the deep component is used to model implicit higher-order features. Existing methods can model first-order and second-order features (as well as their weights) and implicit high-order features. However, no method is currently available for modeling the weights of implicit high-order features, which may limit the performance of the model when capturing users' deep-level interests. Based on this, this paper designs the attention network shown in Figure 2 to model the importance of implicit higher-order features, and the shallow components can be LR or FM models. When the shallow component is an FM, this paper adopts the following method:

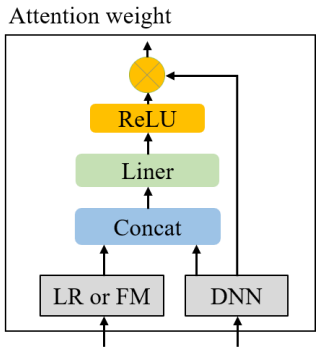

Figure 2. Attention network structure

$$\begin{cases} \lambda_1 = \sigma(w_1(y_{FM}(x) + y_{DNN}(x)) + b_1) \\ \lambda_{w1} = \dfrac{1}{1 + e^{-\lambda_1}} \end{cases}. \quad \textbf{(8)}$$

In formula (8), the input of the shallow component is the concatenation of the output feature of the FM and the implicit

high-order features output by the deep component. $w_i \in \mathbb{R}^{l\times 1}$ represents the parameter matrix of the linear layer of the attention network, b_1 represents the bias parameter of the layer, and σ is the activation function of the unit. $\lambda \in \mathbb{R}^{l\times 1}$ is the output vector of the attention network; $\lambda_{w1} \in \mathbb{R}^{l\times 1}$ is obtained after conducting normalization and denotes the weight coefficient of high-order features. When the shallow component is LR, similarly, the following methods are used to obtain the weight coefficients of higher-order features $\lambda_{w2} \in \mathbb{R}^{l\times 1}$:

$$\begin{cases} \lambda_2 = \sigma(w_2(y_{LR}(x) + y_{DNN}(x)) + b_2) \\ \lambda_{w2} = \dfrac{1}{1+e^{-\lambda_2}} \end{cases}. \quad (9)$$

By inputting shallow component features and high-order features to the designed attention unit, updating the attention parameters and other parameters of the model, the attention unit outputs the attention weight parameters of the high-order features. the model adaptively learns the importance coefficient λ_1 and λ_2 of implicit higher-order features from among all the features and obtained λ_{w1} and λ_{w2} after normalization. The learned attention weights vary greatly with the features of the other two shallow units; this strategy greatly improves the prediction ability of the model.

3.4 Output

The outputs of the attention-based deep and shallow components include low-order feature and implicit high-order feature interactions. For the final CTR prediction, we need only to concatenate these interactions and then apply a nonlinear activation function. The output mapping range of sigmoid function is (0,1), and the range is limited, which can make the data not easy to diverge in the process of transmission, optimization is stable, and is beneficial to classification. It can be used as the output layer, and the output represents the probability, as follows:

$$\hat{y} = sigmoid\left(y_s + \lambda_w \odot y_{DNN}\right). \quad (10)$$

where $\odot$ represents the dot multiplication operation between matrices. According to the different possible shallow components, two different models are formed. When using the combination of an FM and a DNN based on the attention network, y_s is $y_{FM}(x)$, λ_w is λ_{w1} , and the model is att-DeepFM. Similarly, when the shallow component is LR, y_s is $y_{LR}(x)$, λ_w is λ_{w2}, and the model is att-Wide&Deep.

3.5 Training

In CTR prediction models, the Logloss function is widely used, and its definition is as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i). \quad (11)$$

Formula (11) is the unified loss function of the two models in this paper, where y_i is the user's historical click behavior, $\hat{y}_i$ is the click rate predicted by the model, and N is the total number of training samples.

4 Experiments

In this section, this paper makes a large number of experimental comparisons between our model and the more advanced existing models. The experiment is conducted on two large-scale public datasets. First, the data and comparison models are introduced; then, the experimental process of this article is demonstrated; finally, the experimental results are provided, and conclusions are drawn.

4.1 Datasets and Experimental Settings

4.1.1 Datasets

We conduct experiments on two large-scale datasets. The SafeDrive dataset is used to predict the probabilities of claims made by car insurance policyholders. There are 57 fields in the dataset, and features are distributed in different fields according to their categories, including binary features, classification features, continuous features, discrete features and other categories. The AVAZU dataset consists of users' clicks on advertisements, which are sorted according to time series, and each piece of data contains 24 characteristic fields. Due to the large scale of the AVAZU dataset, this paper randomly selects the data at a rate of 20% five times, conducts five experiments under the same conditions, and averages the obtained experimental results. The data set introduction is shown in Table 2.

Table 2. Dataset Statistics

Dataset	Domain	Number of features	Occupied memory
AVAZU	24	1544250	5.87G
SafeDrive	57	595213	110M

4.1.2 Baseline

We compare our method with DNN, Wide&Deep, and DeepFM.

• DNN [5]: The DNN is a model proposed by Google; it consists of a forward feedback neural network, which can model the implicit high-order features of users or items.

• Wide&Deep [7]: This method is also a model proposed by Google that can model both first-order linear features and implicit high-order features simultaneously, where the "wide" part of the model is composed of logical regression models for learning low-order features, and the "deep" part is composed of a DNN for learning implicit high-order features. The model combines these two components to model features.

• DeepFM [8]: This method is a method proposed by Huawei Noah Lab. This method explores the limitations of the "wide" part of Wide&Deep, and models this part by using an FM so that the model has the ability to model second-order interaction features.

4.1.3 Evaluation Indicators

We adopt two evaluation metrics in our experiments: the area under the curve (AUC) and Logloss. The optimization goal in the model updating process is to reduce the value of Logloss; based on this, this paper uses Logloss as one of the evaluation indicators of the models.

In CTR prediction, predicting that users click on items means positive results, while predicting that users do not click on items means negative results. Then the CTR prediction will have four situations: the user clicks on the item and the prediction is positive (TP), the user clicks on the item but the prediction is negative (FN), the user does not click on the item but the prediction is positive (FP), the user does not click on the item and the prediction is negative (TN). In the ROC curve, the ratio of the actual clicks to the predicted clicks is the vertical axis, and the ratio of the actual unclicks to the predicted clicks is the horizontal axis, as follows:

$$TPR = \frac{TP}{TP+FN}, \tag{12}$$

$$FPR = \frac{FP}{TN+FP}. \tag{13}$$

Fill in the coordinate value into the coordinate to get the ROC curve, and the area under the ROC curve is the value of AUC. The ROC curve can be obtained by filling in the coordinate values, and the area under the ROC curve is the value of AUC. The AUC is a widely used metric for evaluating classification problems; the upper limit of the AUC is 1, and a higher AUC indicates better model performance.

4.1.4 Experimental Design and Implementation

In this paper, we use TensorFlow to implement all the models, and the computer specifications are Intel(R) Core (TM) i9-8950HK CPU @ 2.90GH. The dimensionality of the feature domain output of the embedding layer is set to 8. For shallow components, the experiment chooses to use LR or FM. When using LR, the model is att-Wide&Deep, and when using FM, the model is att-DeepFM. For the deep components, the number of network layers is set to two, and a rectified linear unit (ReLU) is used as the activation function. For the attention network component, ReLU or Softplus is utilized as the activation function. All models use Adam as the optimizer, the minimum batch size of input data is 256, the learning rate is set as 0.0004, and the number of iterations is set as 30. To prevent each model from overfitting, the dropout of the deep component is set to 0.4, and the regularization parameter is set to 0.02.

4.2 Analysis of the Experimental Results

4.2.1 Model Performance Comparison

In this paper, we embed the designed attention component into the hybrid algorithm based on FM and DNN to construct att-DeepFM; embed the attention component into the combined algorithm of LR and DNN to construct the att-Wide&Deep model. The method proposed in this paper and the baseline method are experimentally verified, and the experimental results are summarized in Table 3, which includes the experimental AUC and Logloss results obtained on the two large-scale public datasets (AVAZU and SafeDrive) by the method in this paper and the comparison methods. It can be concluded from Table 3 that att-DeepFM has the best performance when the experimental dataset is AVAZU, while the performance of att-Wide&Deep is not as good as that of att-DeepFM and DeepFM but is slightly better than that of Wide&Deep. When the dataset is SafeDrive, compared with other models, att-Wide&Deep with attention components have the best performance in terms of the AUC and Logloss. The experimental results in Table 3 verify the effectiveness of the attention network component in this paper.

Table 3. Experimental results obtained by the proposed model and the compared model on two large datasets

Model	AVAZU		SafeDrive	
	AUC	Logloss	AUC	Logloss
att-DeepFM	**0.7137**	**0.4037**	0.6250	0.1537
DeepFM	0.7136	0.4038	0.6275	0.1529
att-Wide&Deep	0.7011	0.4116	**0.6304**	**0.1525**
Wide&Deep	0.6998	0.4091	0.6303	0.1525
DNN	0.6948	0.4118	0.6256	0.1534

In order to analyze the performance of our model more intuitively, we gradually increase the number of iterations for comparative analysis. Figure 3 and Figure 4 show the comparison of the AUC results obtained in each round by att-DeepFM, DeepFM, att-Wide&Deep and Wide&Deep in 30 iterations. In Figure 3, it is seen that att-DeepFM performs better than DeepFM in every round. However, in Figure 4, Wide&Deep performs better in the first 10 rounds, and after the 10th round, att-Wide&Deep is ahead of Wide&Deep. From the comparison results of the above two experiments, it can be concluded that using an attention network to model the importance of implicit high-order features, the performance of our model is significantly improved compared with the baseline model.

Figure 5 and Figure 6 show comparisons between the Logloss values of the algorithm developed in this paper and the comparison models. As seen from Figure 5, the performance of the att-DeepFM algorithm in this paper is

superior to that of the DeepFM algorithm throughout the whole iterative process. However, it can be seen from Figure 6 that compared with Wide&Deep, att-Wide&Deep is not superior in terms of performance. This paper concludes that although att-Wide&Deep adds an attention network and models the importance of implicit higher-order features, the shallow components of att-Wide&Deep do not model second-order feature interactions, which may lead to the poor performance of the model. In terms of the AUC, the performance of att-Wide&Deep is still competitive.

Based on the above analysis, this paper draws a preliminary conclusion that the CTR prediction model based on an attention network can dynamically obtain the weights of implicit high-order features. The model can adaptively learn high weights for the implicit high-order features that are useful for predicting user behaviors, while the attention network components adaptively reduce the weights for the implicit high-order features that contribute little to the CTR prediction results. In this way, the model can fully integrate the implicit high-order features that are beneficial to CTR prediction, thus improving the prediction performance of the model.

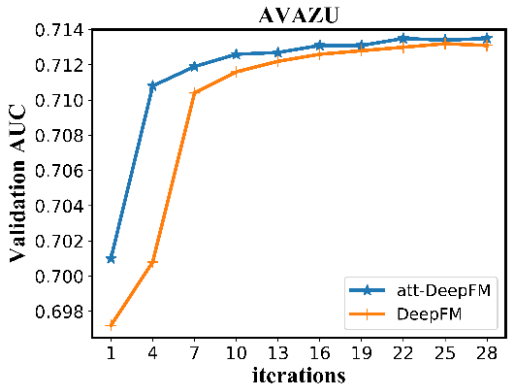

Figure 3. AUC performance of att-DeepFM and DeepFM

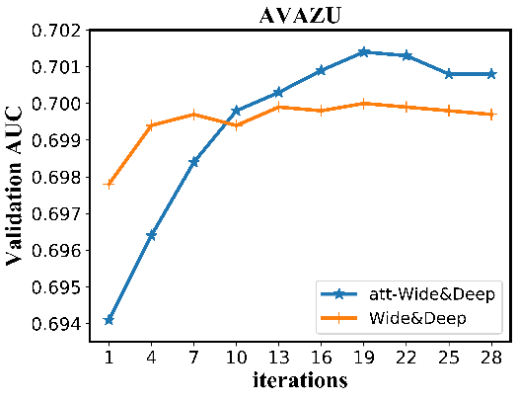

Figure 4. AUC performance of att-Wide&Deep and Wide&Deep

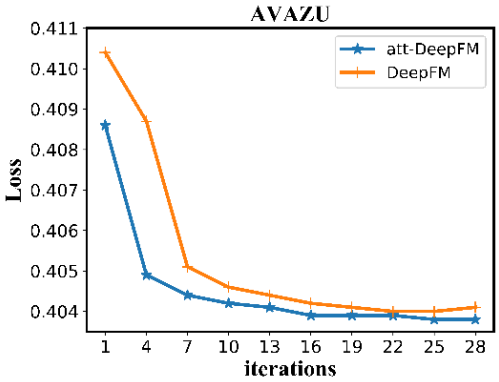

Figure 5. Logloss performance of att-DeepFM and DeepFM

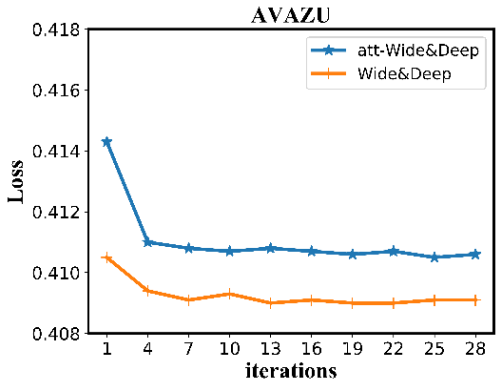

Figure 6. Logloss performance of att-Wide&Deep and Wide&Deep

4.2.2 Activation Function Performance Comparison

In order to observe the effect of the different activation functions in the attention network and deep component on the model performance. We keep the other parameters in the model unchanged. On the AVAZU dataset, we use the att-DeepFM method and select four activation functions for comparative analysis.

(1) Attention network activation function. The attention network is a combination of a linear layer and an activation function. After completing the concatenation and linear mapping of low-order features and implicit high-order features, the activation function is used to perform a nonlinear transformation on the linear output of the attention network. In the attention network component, the activation function used in this paper is ReLU. However, the experimental results of att-DeepFM in Figure 7 show that the overall performance is better when using the Softplus activation function.

(2) Deep component activation function. In deep component architectures, a common practice is to use nonlinear activation functions after linearly mapping the network layer [32]. Therefore, this experiment compares the effects of different activation functions for the deep component on the CTR prediction performance of the resulting models. The experimental results of att-DeepFM in Figure 8 show that this method achieves the best performance when using the ReLU activation function.

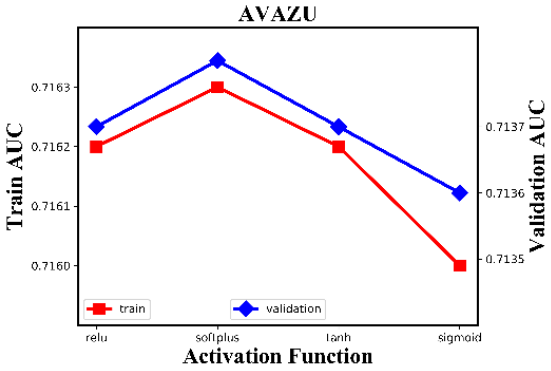

Figure 7. AUC corresponding to different activation functions for the attention unit of att-DeepFM

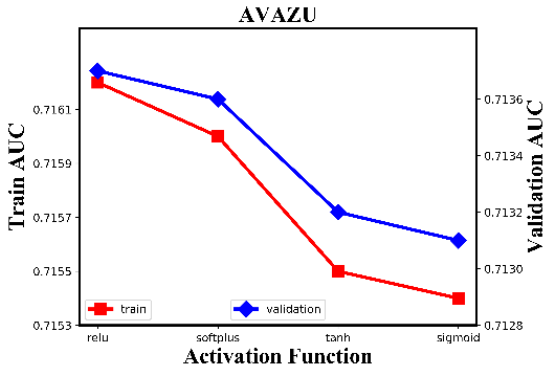

Figure 8. AUC corresponding to different activation functions for the deep component of att-DeepFM

4.2.3 Embedding Layer Size

We set the embedding layer size to 8, 16, and 32; evaluate the performance of the att-DeepFM model; and compare the running times and the total parameter changes yielded under different embedding layer sizes. During the experiment, we keep the other conditions of the model unchanged: the model activation function is kept unified, the optimizer remains unified, the number of iterations remains the same, etc.

The experimental results are shown in Figure 9. Within a limited range, as the embedding size doubles, the AUC of the model increases slowly. We also compare the running times of the model in all rounds and the numbers of model parameters. The results are shown in Table 4. As the embedding size increases, the number of model parameters shows a multiplicative trend. At the same time, the running time of the model slowly increases. Therefore, although the model performance increases to a certain degree, we still set the feature size of the embedding layer to 8.

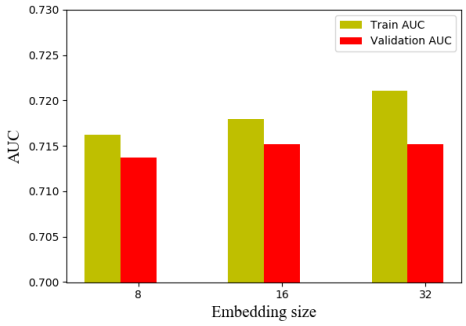

Figure 9. Effect of the embedding layer size on performance

Table 4. Comparison of the running times and numbers of parameters

Embedding size	8	16	32
Running time per round (seconds)	290	306	330
Total number of model parameters	19865	35633	67196

5 Conclusion

Motivated by the drawbacks exhibited by existing models, this paper designs an attention network to model implicit high-order features. This method can dynamically learn the weights of implicit high-order features, which can more effectively capture the interests of different users from the implicit high-order features. We also integrate an attention network into two different hybrid models to form two different recommendation methods: att-DeepFM and att-Wide&Deep, which are suitable for recommendation scenarios. We carry out experiments on two large-scale datasets, and the results show that the prediction performance of the method proposed in this paper is significantly improved. This proves the effectiveness of learning implicit high-order feature weights for CTR prediction.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61862013, 62167002, U1711263, U1811264), the Guangxi Natural Science Foundation of China (Grant No. 2018GXNSFAA281199, 2020GXNSFAA159117), Guangxi Key Laboratory of Trusted Software (No. KX202052), and Guangxi Key Laboratory of Automatic Detection Technology and Instruments (YQ21102).

References

[1] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. Hrafnkelsson, T. Boulos, J. Kubica, Ad click prediction: a view from the trenches, *2013 19th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Chicago, Illinois, USA, 2013, pp. 1222-1230.

[2] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, J. Q. Candela, Practical Lessons from Predicting Clicks on Ads at Facebook, *2014 8th International Workshop on Data Mining for Online Advertising*, New York, USA, 2014, pp. 1-9.

[3] S. Rendle, Factorization Machines, *2010 IEEE International Conference on Data Mining*, Sydney, NSW, Australia, 2010, pp. 995-1000.

[4] Y. Juan, Y. Zhuang, W. Chin, C. Lin, Field-aware Factorization Machines for CTR Prediction, *2016 10th ACM Conference on Recommender Systems*, Boston, MA, USA, 2016, pp. 43-50.

[5] P. Covington, J. Adams, E. Sargin, Deep Neural Networks for YouTube Recommendations, *2016 10th*

ACM Conference on Recommender Systems, Boston, MA, USA, 2016, pp. 191-198.

[6] R. Wang, B. Fu, G. Fu, M. Wang, Deep & Cross Network for Ad Click Predictions, *The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*ADKDD'17)*, Halifax, NS, Canada, 2017, pp. 1-7.

[7] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, Wide & Deep Learning for Recommender Systems, *2016 1st Workshop on Deep Learning for Recommender Systems*, Boston, MA, USA, 2016, pp. 7-10.

[8] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, DeepFM: A factorization-machine based neural network for CTR prediction, *2017 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 1725-1731.

[9] W. Zhang, L. Wang, Deep interaction network based CTR prediction model, *2020 13th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2020, pp. 286-289.

[10] J. Zhang, J. Cheng, Y. Zhou, Q. Yang, GRUIFI: A Group Recommendation Model Covering User Importance and Feature Interaction, *Journal of Internet Technology*, Vol. 22, No. 5, pp. 1143-1155, September, 2021.

[11] Q. Wang, F. Liu, P. Huang, S. Xing, X. Zhao, A Hierarchical Attention Model for CTR Prediction Based on User Interest, *IEEE Systems Journal*, Vol. 14, No. 3, pp. 4015-4024, September, 2020.

[12] G. Huang, Q. Chen, C. Deng, A New Click-Through Rates Prediction Model Based on Deep&Cross Network, *Algorithms*, Vol. 13, No. 12, Article No. 342, December, 2020.

[13] H. Liu, J. Lu, H. Yang, X. Zhao, S. Xu, H. Peng, Z. Zhang, W. Niu, X. Zhu, Y. Bao, W. Yan, Category-Specific CNN for Visual-aware CTR Prediction at JD.com, *2020 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event, California, USA, 2020, pp. 2686-2696.

[14] Q. Zhu, X. Zhou, Z. Song, J. Tan, L. Guo, DAN: Deep Attention Neural Network for News Recommendation, *2019 AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA, 2019, pp. 5973-5980.

[15] J. Qin, W. Zhang, X. Wu, J. Jin, Y. Fang, Y. Yu, User Behavior Retrieval for Click-Through Rate Prediction, *2020 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual Event, China, 2020, pp. 2347-2356.

[16] C. Xu, Q. Li, J. Ge, J. Gao, X. Yang, C. Pei, F. Sun, J. Wu, H. Sun, W. Ou, Privileged Features Distillation at Taobao Recommendations, *2020 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event, California, USA, 2020, pp. 2590-2598.

[17] Y. Liu, C. Li, H. Xiao, J. Cai, GCN-Int: A Click-Through Rate Prediction Model Based on Graph Convolutional Network Interaction, *IEEE Access*, Vol. 9, pp. 140022-140030, November, 2021.

[18] W. Bian, K. Wu, L. Ren, Q. Pi, Y. Zhang, C. Xiao, X. Sheng, Y. Zhu, Z. Chan, N. Mou, X. Luo, S. Xiang, G. Zhou, X. Zhu, H. Deng, CAN: Feature Co-Action Network for Click-Through Rate Prediction, *2022 15th ACM International Conference on Web Search and Data Mining (WSDM '22)*, Virtual Event, AZ, USA, 2022, pp. 57-65.

[19] B. Liu, C. Zhu, G. Li, W. Zhang, J. Lai, R. Tang, X. He, Z. Li, Y. Yu, AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction, *2020 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event, California, USA, 2020, pp. 2636-2645.

[20] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, J. Tang, AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks, *2019 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, 2019, pp. 1161-1170.

[21] T. Huang, Z. Zhang, J. Zhang, FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction, *2019 13th ACM Conference on Recommender Systems*, Copenhagen, Denmark, 2019, pp. 169-177.

[22] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, G. Sun, XDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems, *2018 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 1754-1763.

[23] W. Chen, L. Zhan, Y. Ci, M. Yang, C. Lin, D. Liu, FLEN: leveraging field for scalable CTR prediction, September, 2020, https://arxiv.org/abs/1911.04690.

[24] J. Pan, J. Xu, A. L. Ruiz, W. Zhao, S. Pan, Y. Sun, Q. Lu, Field-weighted Factorization Machines for Click-Through Rate Prediction in Display Advertising, *2018 World Wide Web Conference*, Lyon, France, 2018, pp. 1349-1357.

[25] X. Cheng, N. Li, G. Rysbayrva, Q. Yang, J. Zhang, Influence-Aware Successive Point-of-Interest Recommendation, *World Wide Web*, 2022, Available online: https://doi.org/10.1007/s11280-022-01055-w [Online; accessed on April 11, 2022]

[26] X. He, T. Chua, Neural Factorization Machines for Sparse Predictive Analytics, *2017 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Shinjuku, Tokyo, Japan, 2017, pp. 355-364.

[27] X. Lu, H. Zhang, A Content-Aware POI Recommendation Method in Location-Based Social Networks Based on Deep CNN and Multi-Objective Immune Optimization, *Journal of Internet Technology*, Vol. 21, No. 6, pp. 1761-1772, November, 2020.

[28] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, T. S. Chua, Attentional factorization machines: Learning the weight of feature interactions via attention networks, *2017 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 3119-3125.

[29] Q. Song, D. Cheng, H. Zhou, J. Yang, Y. Tian, X. Hu, Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction, *2020 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event, California, USA, 2020, pp. 945-955.

[30] R. Fei, Y. Zhu, Q. Yao, Q. Xu, B. Hu, A Deep Learning Method Based Self-Attention and Bi-directional LSTM in Emotion Classification, *Journal of Internet*

Technology, Vol. 21, No. 5, pp. 1447-1461, September, 2020.

[31] Z. Lyu, Y. Dong, C. Huo, W. Ren, Deep Match to Rank Model for Personalized Click-Through Rate Prediction, *2020 AAAI Conference on Artificial Intelligence*, New York, USA, 2020, pp. 156-163.

[32] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, K. Gai, Deep Interest Network for Click-Through Rate Prediction, *2018 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 1059-1068.

Biographies

Qing Yang is currently a Professor with the School of Electronics Engineering and Automation, Guilin University of Electronic Technology, China. Her research interests include intelligent information processing, massive data management and large-scale data processing optimization.

Ning Li received the bachelor's degree from Xinyang Normal University, China, in 2019. She is currently pursuing the master's degree with the Guilin University of Electronic Technology China. Her research interest includes intelligent information processing.

Shiyan Hu received the bachelor's degree from Henan University of Technology, China, in 2020. She is currently pursuing the master's degree with the Guilin University of Electronic Technology, China. Her research interest includes intelligent information processing.

Heyong Li received the bachelor's degree from Henan Polytechnic University, China, in 2018. He obtained a master's degree from Guilin University of Electronic Science and technology, China, in 2021. His research interest includes intelligent information processing.

Jingwei Zhang received the Ph.D. degree from East China Normal University, China, in 2012. He is currently a Professor with the School of Computer and Information Security, Guilin University of Electronic Technology, China. His main research interests include big data analysis and computing optimization, recommendation system.