TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 16/18 pp186–195
DOI: 10.26599/TST.2020.9010053
Volume 27, Number 1, February 2022

CAN: Effective Cross Features by Global Attention Mechanism and Neural Network for Ad Click Prediction

Wenjie Cai, Yufeng Wang*, Jianhua Ma, and Qun Jin

Abstract: Online advertising click-through rate (CTR) prediction is aimed at predicting the probability of a user clicking an ad, and it has undergone considerable development in recent years. One of the hot topics in this area is the construction of feature interactions to facilitate accurate prediction. Factorization machine provides second-order feature interactions by linearly multiplying hidden feature factors. However, real-world data present a complex and nonlinear structure. Hence, second-order feature interactions are unable to represent cross information adequately. This drawback has been addressed using deep neural networks (DNNs), which enable high-order nonlinear feature interactions. However, DNN-based feature interactions cannot easily optimize deep structures because of the absence of cross information in the original features. In this study, we propose an effective CTR prediction algorithm called CAN, which explicitly exploits the benefits of attention mechanisms and DNN models. The attention mechanism is used to provide rich and expressive low-order feature interactions and facilitate the optimization of DNN-based predictors that implicitly incorporate high-order nonlinear feature interactions. The experiments using two real datasets demonstrate that our proposed CAN model performs better than other cross feature- and DNN-based predictors.

Key words: click-through rate prediction; global attention mechanism; feature interaction; neural network

1 Introduction

In the online advertising industry, advertisers pay publishers/advertisement platforms to display their ads on the publishers' websites. One common mode of payment is based on cost per click in which each click brings direct benefits to advertisers[1]. Hence, the performance of the click-through rate (CTR) prediction scheme significantly affects the final revenue of platforms and has already become a hot topic in the area of recommendation systems.

The real-world datasets used in online advertising have distinct characteristics. Different from numerical variables that are naturally found in images and audios, the raw features of web-scale recommender systems are mostly categorical, thus leading to a large and sparse feature space that presents a challenge in feature exploration[2, 3]. For instance, the characteristics of a clothing advertisement may have the following categorical features: color {red, yellow, ...}, style {retro, trend, ...}, brand {Nike, Adidas, ...}, etc. Ads are generally composed of hundreds of features, and each feature contains hundreds of categories. One of the effective directions for processing highly sparse and categorical data is to construct a feature interaction, which is also called a cross feature. For instance, publishing food ads at lunch time is reasonable. In this case, the second-order cross feature ⟨advertisement category = Food, timestamp = Lunch Time⟩ is highly

• Wenjie Cai and Yufeng Wang are with Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, 210003, China. E-mail: 1018010235@njupt.edu.cn; wfwang@njupt.edu.cn.
• Jianhua Ma is with Digital Media Department in the Faculty of Computer and Information Sciences, Hosei University, Tokyo 163-8001, Japan. E-mail: jianhua@hosei.ac.jp.
• Qun Jin is with Networked Information Systems Laboratory, Department of Human Informatics and Cognitive Sciences, Waseda University, Tokyo 163-8001, Japan. E-mail: jin@waseda.jp.
∗ To whom correspondence should be addressed.
Manuscript received: 2020-07-20; revised: 2020-09-21; accepted: 2020-10-09

informative for prediction. This second-order feature interaction formed by the type of ad and timestamp exerts an important influence on the accuracy of ad click prediction. In another example, a third-order feature interaction applicable to male college students who are fond of science fiction movies may comprise {gender, age, movie genre, etc.}. The appropriate modeling of feature interactions has been shown to greatly improve prediction effects[4–7]. However, effectively and automatically modeling meaningful feature interactions on the basis of the experience of human experts is difficult, because web-scale advertising is typically discrete and categorical[8, 9]. Human experts might only design easily understandable but superficial feature interactions, such as those described in the case of the clothing advertisement. In addition, real advertising data contain numerous different features, and relying on manual selection will likely incur huge costs[10].

Instead of augmenting feature vectors manually, feature interactions may be learned automatically through machine learning (ML) models[11]. The most popular ML algorithm is the factorization machine (FM)[12, 13], which constructs the hidden factor vector of each feature and obtains the cross feature on the basis of the inner product of the hidden factor vectors. However, the FM algorithm can only construct second-order linear feature interactions. These feature interactions alone are not expressive enough for real-world data, which present a complex and nonlinear structure[14].

Recently, the application of deep neural networks (DNNs) has achieved great progress in natural language processing[15] and recommender systems. DNNs have already been applied to CTR prediction tasks[16, 17] as DNNs could automatically and implicitly learn expressive feature representations and capture high-order and nonlinear cross features[18]. For example, deep crossing[4] was used to concatenate feature vectors, and a residual network structure was designed to mine the relationship between features. However, simply concatenating feature vectors as inputs to neural networks makes neural networks difficult to optimize because of problems such as vanishing/exploding gradients, overfitting, and degradation[19]. As verified in Ref. [14], neural models, including deep crossing and wide deep[18], are extremely prone to overfitting after certain epoch training when original data are used as the input to the neural layer. Original feature data (referred to as a single raw feature) only carry relevant information and thus lack corresponding feature cross information.

A natural idea is to add a low-order feature interaction layer before the neural network layer so as to facilitate the learning process of the succeeding "deep" layers. Inspired by the above consideration, this study proposes a CTR prediction algorithm called CAN, which exploits the attention mechanism to obtain second-order feature interactions and combines DNNs to perform high-order nonlinear feature interactions. The attention mechanism has demonstrated effectiveness in a variety of tasks, such as question answering, text summarization, recommendations, etc. The global attention mechanism[20] is a type of attention mechanism that plays an important role in neural machine translation (NMT) by eliminating the constraint of embedding context information only into a fixed-length vector. Considering that the global attention mechanism can efficiently calculate attention weights for all hidden factors, we apply it to the acquisition of weights between all pairs of features. The approach can provide more abundant interaction information than the process of simply concatenating original feature vectors. The results of the interaction layer are then entered into the DNN layer for high-order nonlinear feature interaction. On the basis of the output of the DNN layer, the probability of a user clicking on advertisements is ultimately predicted.

The main contributions of this work are as follows:

• Prior to DNN-based implicit feature interaction, we explicitly exploit the second-order cross features on the basis of the global attention mechanism. This approach not only improves the prediction accuracy but also facilitates neural network optimization.

• We conduct extensive experiments on two real-world atasets. The results demonstrate that the proposed CAN signifantly outperforms several state-of-the-art cross feature prediction schemes based on DNNs.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 presents the proposed approach called CAN for learning feature interactions. Section 4 details the experimental results and analysis. Section 5 provides the conclusion and future work.

2 Related Work

An important direction of current CTR estimation is to build an efficient feature interaction model.

Traditional ML, such as logistic regression, can be considered as a first-order feature interaction. FM[5] embeds each feature into a low dimension latent vector

and makes recommendations via the product of two latent vectors[9]. The field-aware FM[6] algorithm divides features with similar attributes into the same fields; the second-order feature interaction is then obtained by FM in each field. However, FM-based algorithms have limited ability to express low-order interactions, i.e., their expression is not enough to represent feature interactions.

As a powerful approach to learning feature representation, DNNs have the potential to learn sophisticated feature interactions[21]. For example, in Ref. [5], deep crossing was used to design a stacking layer, which simply contacts all feature vectors together as the input to a special residual network structure. Deep crossing can automatically learn cross features and reduce the cost of feature interactions that are manually extracted by human users. However, original values lack the cross information of features; thus, the direct input of original values to the DNN layer results in optimization problems, such as vanishing/exploding gradients, overfitting, and degradation[14, 19].

Low-order feature interactions have been modeled on the basis of the original feature data before the DNN layer so as to enrich the inputs of the DNN layer with cross information and thereby alleviate the optimization problem of DNNs. For instance, the product-based neural network (PNN)[22] proposes two methods for performing second-order feature interaction based on inner products (called IPNN) and outer products (called OPNN). Neural factorization machine (NFM) offers a new second-order interaction layer called bi-interaction. Similar to the FM algorithm, bi-interaction multiplies and adds all features in pairs to obtain a combined vector with a fixed length, which is then used as the input to the neural network layer. These models combine second-order feature interactions with fully connected neural networks and achieve relatively good prediction results. However, all these algorithms simply multiply the original features. This type of value multiplication is not enough to characterize cross features. In real-world applications, different feature interactions usually have different levels of predictive power. Interactions with few useful features should be assigned low weights as their contribution to prediction is minimal. Therefore, simple value multiplication cannot distinguish the importance of feature interactions and may thus weaken prediction performance.

In this study, we propose an effective CTR prediction algorithm that is based on the global attention mechanism and DNN. The attention mechanism is widely used to distinguish the importance of feature interactions, and it has recently been applied to CTR prediction tasks. For instance, attentional factorization machine (AFM)[23] introduces the attention mechanism to the classic FM algorithm by adding attention weight information before the traditional feature pair. However, AFM does not explore the impact of higher-order feature interactions on prediction. Meanwhile, deep interest network (DIN)[11] obtains the attention weight value of the viewed product and the product to be recommended in the special product collection. However, DIN is designed for specific datasets, and the calculation of attention values through outer products and fully connected networks results in high complexity.

Different from existing algorithms, the global attention mechanism proposed herein has made outstanding achievements in NTM in terms of the construction of second-order cross features. Reference [20] proposed two attention mechanisms for NTM, namely, local attention mechanism and global attention mechanism. These mechanisms differ in terms of their focus on words. In NTM, the global attention mechanism calculates the attention weight between all word vectors in the entire sentence, whereas the local attention mechanism only calculates the weight relationship between the words around the target word vector by setting a moving window. Given the complexity and large volume of word vectors involved document translation, computational overhead tends to increase, thereby calling for more local attention mechanisms. In the field of recommendation systems, the number of features is much smaller than the text vocabulary in NMT. Hence, the complexity of global attention construction need not be considered, and the weight factors of all features can be easily calculated. These advantages result in rich cross feature information. Therefore, we refer to the global attention mechanism as the low-order feature interaction scheme in our work.

In brief, the proposed CAN scheme effectively exploits the influence of low- and high-order feature interactions on the prediction effect.

3 Cross Feature-Based CTR Prediction Model: CAN

3.1 Framework of proposed CAN model

The purpose of a CTR task is to predict the probability of a user clicking a recommended ad. Figure 1 presents the framework of the proposed CAN system. Each input data comprises categorical features and numerical features,

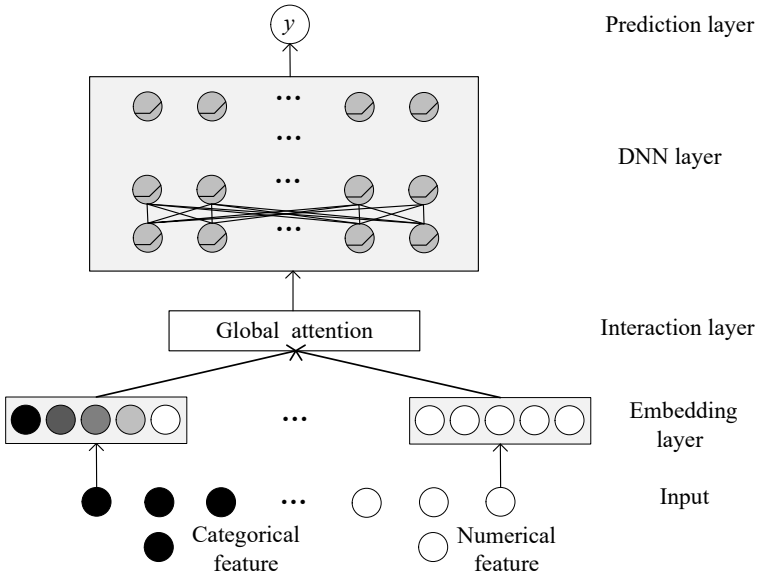

Fig. 1 Framework of cross feature-based CTR prediction: CAN.

and one hot encoding turns the categorical features into a series of binary inputs. The embedding layer is then used to project all numerical and categorical features into fixed-length vectors, which represent the value of each feature. Next, the output of the embedding layer is fed into the interaction layer. The interaction layer restructures the feature vectors on the basis of the attention mechanism. The DNN layer further models the higher-order nonlinear feature interactions. The final result is then predicted through the prediction layer. For a clear discussion, we present in Table 1 the main notations used in this study and their meanings.

3.2 Embedding layer

The embedding layer is illustrated in Eq. (1). The embedding layer is a fully connected layer that projects each categorical feature and numerical feature to a dense vector.

$$e_i = W_{embed,i} x_i \tag{1}$$

where e_i is the embedding vector and x_i is the input of the i-th feature. $W_{embed} \in \mathbb{R}^{d_e \times d_i}$ is the corresponding embedding matrix; and d_e, d_i are the embedding size and size of x_i, respectively.

Table 1 Main notations in this study and their meanings.

Notation	Description
e_i	Embedding vector of i-th field of feature
W	Parameter matrix for obtaining attention weights
φ_m, k	Degree of association between feature m and feature k
α_m, k	Attention weight between feature m and feature k
$\widetilde{e_m}$	Feature vector m after reconstruction by attention layer
y	True label of user clicking on the advertisement

Figure 2 illustrates the embedding process of two different types of features. $x_i (2 \leqslant i \leqslant m)$ is the original categorical feature represented by a set of high-dimensional binary numbers. We convert each feature into a fixed-length vector e_i by multiplying the value with the embedding matrix, which is W_{embed}, in Eq. (1). x_1 is a scalar value representing the original numerical feature. We adopt the same strategy to convert this feature into a fixed-length vector. After the processing of the embedding layer, the two types of original feature data are converted into low-dimensional fixed-length vectors. The output of the embedding layer is $e = [e_1, e_2, ..., e_m]$, where e_i is the embedding of the i-th feature.

3.3 Interaction layer

3.3.1 Feature interaction based on global attention mechanism

Once the numerical and categorical features are embedded in the same low-dimensional space, we proceed to the modeling of second-order cross features based on the global attention mechanism.

Figure 3 illustrates the application of the global attention mechanism to the construction of the second-

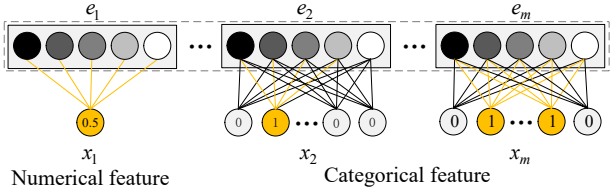

Fig. 2 Structure of embedding layer.

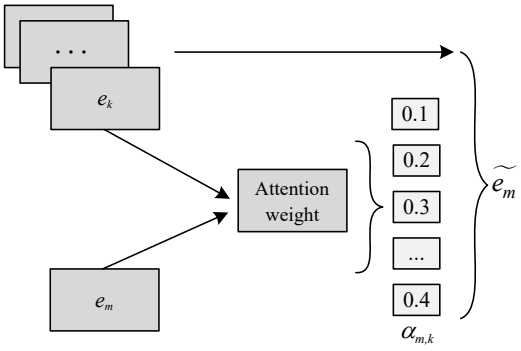

Fig. 3 Global attention mechanism used in CAN.

order feature interaction. Taking the m-th feature as an example, we can obtain the attention weight $\alpha_{m,k}$, which represents the degree of connection between any pair of embedding vectors e_m and e_k. A new vector $\widetilde{e_m}$ is constructed by weighted summation to characterize the interaction information of feature m relative to the other features. The detailed processes are described in the next subsection.

3.3.2 Attention weight

Figure 4 takes feature m as an example and illustrates how the global attention mechanism computes the attention weight $\alpha_{m,k}$, with the other features k.

$$\begin{aligned} \varphi_{m,k} &= e_m^{\mathrm{T}} \times W \times e_k, \qquad k = 1, 2, \ldots, n; \\ \alpha_{m,k} &= \frac{\exp(\varphi_{m,k})}{\sum_{l=1}^{N} \exp(\varphi_{m,l})} \end{aligned} \tag{2}$$

The specific implementation process is shown in Eq. (2). $e_m \in \mathbb{R}^d$ is the embedding vector of feature m, and d is the embedding size. $W \in \mathbb{R}^{d \times d}$ is the corresponding parameter matrix, and $e_k \in \mathbb{R}^d$ is the embedding vector of feature k. The parameter matrix W is used to construct the connection coefficient $\varphi_{m,k}$, between two features. After a softmax layer, the connection coefficient is normalized to obtain the attention weight $\alpha_{m,k}$, the weight coefficient represents the degree of correlation between features. The greater the weight value (α_m) is, the greater the relationship between features $\langle m, k \rangle$. A new vector $\widetilde{e_m}$ is obtained by the process of weighted summation shown in Eq. (3). The original vector e_m and the $\widetilde{e_m}$ vector have the same dimensions, but the latter contains weight information with other features.

$$\widetilde{e_m} = \sum_{k=1}^{n} \alpha_{m,k} \times e_k \tag{3}$$

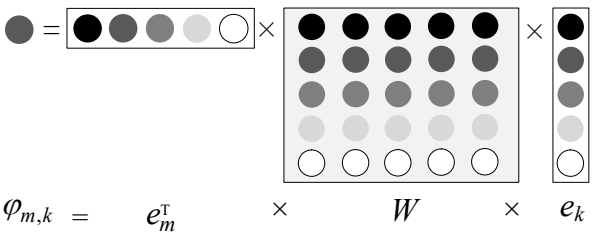

Fig. 4 Inferring cross feature attention weight.

3.3.3 Output of interaction layer

After attaining each new feature vector, we obtain the output of the interaction layer through Eq. (4).

$$z = \widetilde{e_1} \oplus \widetilde{e_2} \oplus \cdots \oplus \widetilde{e_n} \tag{4}$$

where $\oplus$ denotes the add-by-bit operation. After the addition, an output vector z with the same length as the embedding size is obtained. This output vector contains the interactive information between each feature.

3.4 DNN layer

As shown in Fig. 5, the deep network is a stack of fully connected layers, which are capable of learning high-order nonlinear interactions between features. Formally, fully connected layers are defined as follows:

$$z_{l+1} = f(W_l z_l + b_{l+1}) \tag{5}$$

3.5 Prediction layer

The output vector y is transformed into the final prediction score. The final score is defined in Eq. (6):

$$p = \mathrm{sigmod}(h^{\mathrm{T}}[y]) \tag{6}$$

where $y \in \mathbb{R}^m$ is the output of the DNN layer, vector $h \in \mathbb{R}^m$ denotes the neuron weights of the prediction layer, m is the same as the number of neurons in the last layer, and $\mathrm{sigmod}(x) = 1/(1 + \exp(-x))$.

3.6 Training

For binary classification, the commonly used loss function is the log loss along with a regularization term,

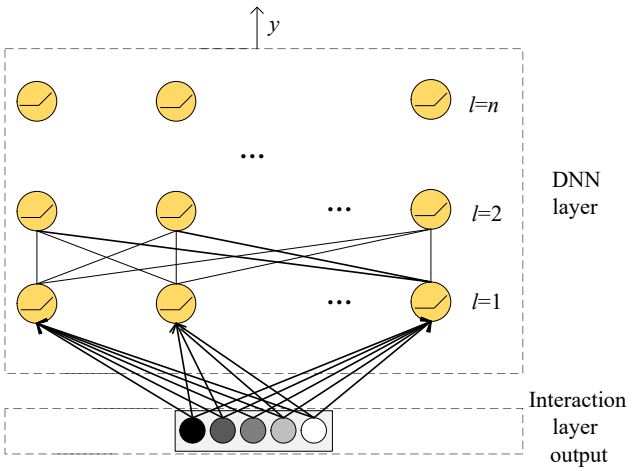

Fig. 5 Structure of DNN component in CAN.

as shown in Eq. (7).

$$\text{Logloss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i) + (1-y_i)\log(1-p_i) + \lambda \sum_{l} \|w_l\|^2 \tag{7}$$

where p_i denotes the probabilities computed from Eq. (6), y_i denotes the true labels, N is the total number of inputs, and λ is the L2 regularization parameter. The parameters to be learned in our model are $\{W_{embed,i}$ in the embedding layer, W in the interaction layer, W_l in the DNN layer, and b_l in the DNN layer$\}$. The parameters are updated by minimizing the total Logloss using Adam.

4 Performance Evaluation

In this section, we conduct extensive experiments to answer the following questions:

• (Q1) How does the proposed CAN perform in a real dataset relative to other schemes?

• (Q2) Does prediction performance improve with CAN relative to the method of directly applying a DNN layer without setting the attention layer?

• (Q3) What is the effect of embedding size on the CAN model?

4.1 Experiment setup

4.1.1 Dataset description

We evaluate our proposed models on the basis of the following datasets. The statistics of the datasets are summarized in Table 2.

Criteo: Criteo is a benchmark dataset for CTR prediction; it contains 45 million users' click records for displayed ads. It also contains 26 categorical feature fields and 13 numerical feature fields. Given the limited performance of the experimental machine, we sample 2 million sets of data as the training set.

MovieLens: MovieLens is a dataset of film reviews and contains the movie ratings of one million users. Each piece of data includes categorical characteristics, such as movie type, user age, and gender. The original user ratings range from 0 to 5. We treat the samples with ratings greater than 3 as positive samples and remove the neutral samples, i.e., the samples with ratings equal to 3.

Table 2 Statistics of experimental datasets.

Data	Sample	Field	Feature
Criteo	4.5×10^7	39	2.3×10^6
MovieLens	100 209	7	90 445

For both datasets, the labels in the test sets are not publicly available. Thus, we split the respective training data of the two sets for validation.

4.1.2 Baseline schemes

We compare CAN with the following four models: FM, NFM, DNN, and PNN.

• FM: FM is the most widely used model for modeling second-order feature interactions. It does not model nonlinear feature interactions via neural networks.

• NFM: NFM performs second-order feature interactions by applying the FM algorithm before the DNN layer.

• DNN: The embedding layer, output layer, and hyperparameter tuning process of the DNN are the same as those of CAN, PNN, and NFM. It is different from the CAN model because of its lack of interaction layer.

• PNN: The second-order feature interaction is constructed by the inner product or outer product algorithms before the DNN layer. In this experiment, we select the inner product method.

4.1.3 Evaluation metrics

We use two metrics for model evaluation: AUC (area under the ROC curve) and Logloss (cross entropy); both metrics are widely used to evaluate classification problems.

AUC: The AUC value is equivalent to the probability that a randomly chosen positive example ranks higher than a randomly chosen negative example[24]. A high AUC indicates good performance.

In practice, we can use the following formula to calculate the AUC value:

$$AUC = \frac{\sum_{i=1,j=1}^{i\leqslant M, j\leqslant N} \delta(r_i - r_j > 0)}{M \times N} + \frac{\sum_{i=1,j=1}^{i\leqslant M, j\leqslant N} 0.5 \times \delta(r_i - r_j = 0)}{M \times N} \tag{8}$$

where M represents the number of positive samples (items actually clicked by the user), N represents the number of negative samples (items actually not clicked by the user), r_i is the prediction score of the positive sample, r_j is the prediction score of the negative sample, and δ(condition) is the indication function. When the condition is true, δ(true) is 1; otherwise, δ(false) is 0.

Logloss: Logloss, as defined in Eq. (7), measures the distance between the predicted score and the true label for each instance. A low logloss indicates a good performance.

4.1.4 Implementation details

We utilize TensorFlow[25] to realize our scheme. The hyperparameters of each model are tuned by grid searching on the validation set. Finally, we adopt the embedding dimension (32), batch size (512), and L2 regularization with λ (0.0001) for all methods. Considering the huge size of data, we choose Adam as the optimization method; Adam is the most commonly used method for CTR estimation. We apply exponential decay in which the learning rate starts at 0.001 and the decay rate is set to 0.9. For all the DNN-based methods, we use four hidden layers with a size of (256, 256, 256, 256).

4.2 Quantitative results (answer to Q1)

Figures 6–9 show the performance of the models on the Criteo and MovieLens datasets, respectively, in terms of Logloss and AUC. The proposed CAN is marked in red in Figs. 6 and 7; the other models are marked in blue.

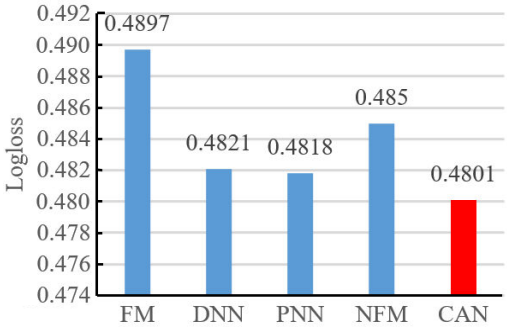

Fig. 6 Illustration of Logloss of various schemes using Criteo dataset.

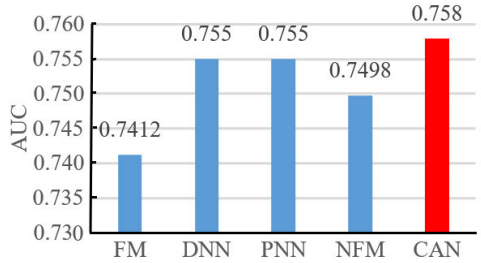

Fig. 7 Illustration of AUC of various schemes using Criteo dataset.

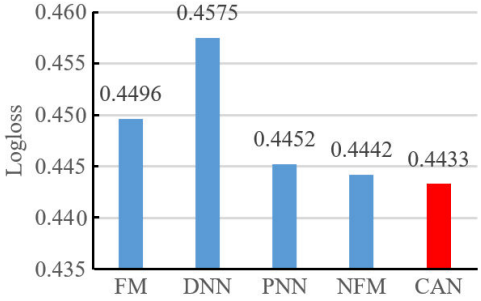

Fig. 8 Illustration of Logloss of various schemes using MovieLens dataset .

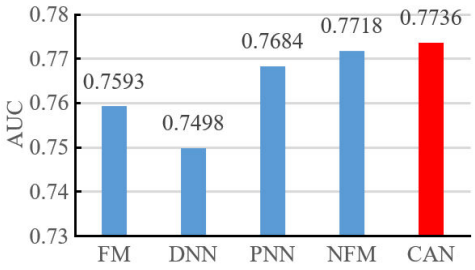

Fig. 9 Illustration of AUC of various schemes using MovieLens dataset.

For the Criteo dataset, the prediction results of the FM model are significantly worse than those of the other DNN-based models. The difference is explained as follows. For data with many types of features and for highly sparse data such as Criteo, the low-order model, i.e., FM, cannot adequately express the complex feature interactions. Interestingly, PNN and NFM are worse than DNN for a similar reason, that is, they only add low-order feature interactions before the neural network. CAN is better than DNN. This result implies that under extremely complex feature environments, the use of an attention mechanism to build a low-order feature interaction is more efficient than simple value multiplication, which is used in NFM and PNN.

For the MovieLens dataset, the prediction result of DNN is worse than that of the FM algorithm. This result indicates that for datasets with few features and a simple structure, such as MovieLens, low-order cross features can be sufficient to express the relationship between features. Relative to DNN, PNN, NFM, and CAN show a significantly improved prediction effect. This result indicates that the construction of low-order feature interactions obviously facilitates the optimization of neural networks and subsequently improves prediction performance. Similar to the result of the Criteo dataset, the attention mechanism in the MovieLens dataset is more efficient than PNN and NFM.

4.3 Influence of second-order feature interaction on prediction (answer to Q2)

In this study, we infer that a serious overfitting problem may exist when raw features are used directly as input to the neural network layer. In this part, our experiments on the MovieLens dataset explore whether constructing explicit second-order cross features through the attention mechanism can help alleviate the problem of overfitting. We choose the DNN and CAN algorithms

for comparison. To explore the problem of overfitting, we set the L2 regularization coefficient λ in the algorithm to 0 and set all the other parameter conditions to be the same.

We focus on the overfitting of the algorithms on the validation set as the epoch increases and not on the specific prediction effect of the algorithms. From Fig. 10, we can observe that with the continuous training of the model, the AUC value decreases rapidly and then tends to flatten after the DNN algorithm achieves the best prediction effect. Although the CAN algorithm still suffers from overfitting, its AUC value is lower than that of the DNN algorithm. This result shows that feeding the neural network with extensive feature cross information can indeed alleviate overfitting and other optimization problems to a certain extent.

4.4 Influence of embedding size (answer to Q3)

In this subsection, we investigate the performance of the models in terms of the output dimension of the embedding layer, i.e., the embedding size. As shown in Fig. 11, Logloss gradually decreases initially as the embedding size increases. When the embedding size increases to 32, Logloss reaches the minimum value. Similarly, as shown in Fig. 12, AUC gradually

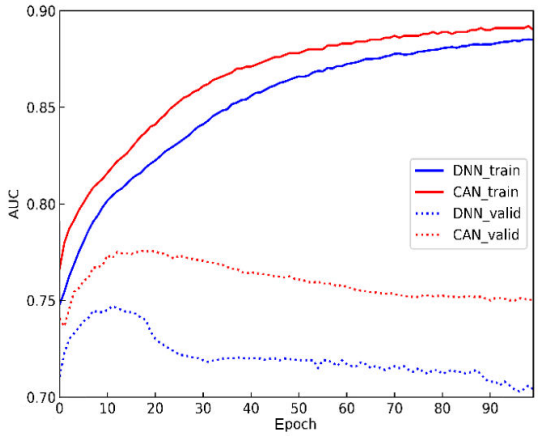

Fig. 10 Comparison of training and validation processes between DNN and CAN.

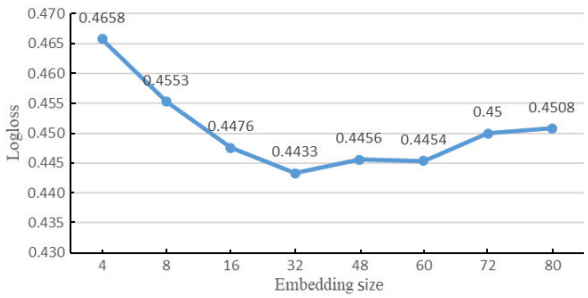

Fig. 11 Illustration of Logloss varying the embedding size.

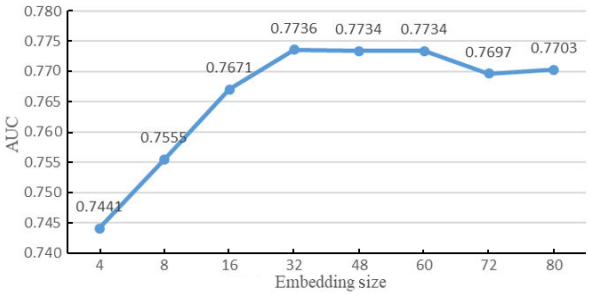

Fig. 12 Illustration of AUC varying the embedding size.

increases with the increase of embedding size. When the embedding size reaches 32, 48, and 60, AUC reaches the maximum value and remains almost unchanged. Regardless of the metric used (i.e., Logloss or AUC), prediction accuracy decreases when the embedding size is too large. An excessively small embedding size causes significant loss of original information, whereas an excessively large embedding size causes serious overfitting. According to the experimental results of Logloss and AUC, the embedding size of the CAN model is set to 32.

5 Conclusion

In this study, we propose a new feature interaction network on the basis of the concept of the global attention mechanism and DNN. The proposed scheme first designs a second-order feature interaction layer on the basis of the global attention mechanism. Then, high-order nonlinear feature interactions are mined through the neural network layer. The second-order cross feature layer can facilitate neural network layer training, and the neural network layer can address the insufficient ability of low-order feature interaction models to express cross features. Using CTR as the application scenario, we perform thorough experiments on two actual datasets. The results prove that the proposed CAN algorithm achieves better prediction accuracy than other DNN- and cross feature-based prediction models.

References

[1] H. F. Guo, R. M. Tang, Y. M. Ye, Z. G. Li, and X. Q. He, DeepFM: A factorization-machine based neural network for CTR prediction, present at 26th Int. Joint Conf. Artificial Intelligence, Melbourne, Australia, 2017.

[2] R. X. Wang, B. Fu, G. Fu, and M. L. Wang, Deep & cross network for ad click predictions, in *Proc. ADKDD'17*, Halifax, Canada, 2017.

[3] J. Zhang, Y. F. Wang, Z. Y. Yuan, and Q. Jin, Personalized real-time movie recommendation system: Practical prototype and evaluation, *Tsinghua Science and*

Technology, vol. 25, no. 2, pp. 180–191, 2020.

[4] Y. Shan, T. R. Hoens, J. Jiao, H. J. Wang, D. Yu, and J. C. Mao, Deep crossing: Web-scale modeling without manually crafted combinatorial features, in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 255–262.

[5] S. Rendle, Factorization machines, presented at 2010 IEEE Int. Conf. Data Mining,Sydney, Australia, 2010, pp. 995–1000.

[6] Y. C. Juan, Y. Zhuang, W. S. Chin, and C. J. Lin, Field-aware factorization machines for CTR prediction, in *Proc. 10th ACM Conf. Recommender Systems*, Boston, MA, USA, 2016, pp. 43–50.

[7] M. Blondel, A. Fujino, N. Ueda, and M. Ishihata, Higher-order factorization machines, in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 3351–3359.

[8] W. P. Song, C. C. Shi, Z. P. Xiao, Z. J. Duan, Y. W. Xu, M. Zhang, and J. Tang, AutoInt: Automatic feature interaction learning via self-attentive neural networks, in *Proc. 28th ACM Int. Conf. Information and Knowledge Management*, Beijing, China, 2019, pp. 1161–1170.

[9] J. X. Lian, X. H. Zhou, F. Z. Zhang, Z. X. Chen, X. Xie, and G. Z. Sun, xDeepFM: Combining explicit and implicit feature interactions for recommender systems, in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 1754–1763.

[10] X. R. He, J. F. Pan, O. Jin, T. B. Xu, B. Liu, T. Xu, Y. X. Shi, A. Atallah, R. Herbrich, S. Bowers, et al., Practical lessons from predicting clicks on ads at facebook, in *Proc. 8th Int. Workshop on Data Mining for Online Advertising*, New York, NY, USA, 2014, pp. 1–9.

[11] G. R. Zhou, X. Q. Zhu, C. R. Song, Y. Fan, H. Zhu, X. Ma, Y. H. Yan, J. Q. Jin, H. Li, and K. Gai, Deep interest network for click-through rate prediction, in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 1059–1068.

[12] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, Fast context-aware recommendations with factorization machines, in *Proc. 34th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Beijing, China, 2011, pp. 635–644.

[13] R. J. Oentaryo, E. P. Lim, J. W. Low, D. Lo, and M. Finegold, Predicting response in mobile advertising with hierarchical importance-aware factorization machine, in *Proc. 7th ACM Int. Conf. Web Search and Data Mining*, New York, NY, USA, 2014, pp. 123–132.

[14] X. N. He and T. S. Chua, Neural factorization machines for sparse predictive analytics, in *Proc. 40th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Shinjuku, Japan, 2017, pp. 355–364.

[15] P. S. Huang, X. D. He, J. F. Gao, L. Deng, A. Acero, and L. Heck, Learning deep structured semantic models for web search using clickthrough data, in *Proc. 22nd ACM Int. Conf. Information & Knowledge Management*, San Francisco, CA, USA, 2013, pp. 2333–2338.

[16] W. N. Zhang, T. M. Du, and J. Wang, Deep learning over multi-field categorical data, presented at European Conf. Information Retrieval, Cham, Germany, 2016, pp. 45–57.

[17] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al., Wide & deep learning for recommender systems, in *Proc. 1st Workshop on Deep Learning for Recommender Systems*, Boston, MA, USA, 2016, pp. 7–10.

[19] X. N. He, L. Z. Liao, H. W. Zhang, L. Q. Nie, X. Hu, and T. S. Chua, Neural collaborative filtering, in *Proc. 26th Int. Conf. World Wide Web*, Perth, Australia, 2017, pp. 173–182.

[20] M. T. Luong, H. Pham, and C. D. Manning, Effective approaches to attention-based neural machine translation, in *Proc. 2015 Conf. Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 1412–1421.

[21] H. Chen, C. T. Yin, R. M. Li, W. G. Rong, Z. Xiong, and B. David, Enhanced learning resource recommendation based on online learning style model, *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 348–356, 2019.

[22] Y. R. Qu, H. Cai, K. Ren, W. N. Zhang, Y. Yu, Y. Wen, and J. Wang, Product-based neural networks for user response prediction, presented at 2016 IEEE 16th Int. Conf. Data Mining (ICDM), Barcelona, Spain, 2016, pp. 1149–1154.

[23] J. Xiao, H. Ye, X. N. He, H. W. Zhang, F. Wu, and T. S. Chua, Attentional factorization machines: learning the weight of feature interactions via attention networks, in *Proc. 26th Int. Joint Conf. Artificial Intelligence*, Melbourne, Australia, 2017, pp. 3119–3125.

[24] T. Fawcett, An introduction to roc analysis, *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[25] M. Abadi, P. Barham, J. M. Chen, Z. F. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in *Proc. 12th USENIX Conf. Operating Systems Design and Implementation*, Savannah, GA, USA, 2016, pp. 265–283.

Yufeng Wang is currently a professor at the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, China. He is also the guest researcher with the Advanced Research Center for Human Sciences, Waseda University, Japan. His research interests focus on cyber-physical-social systems, crowdsourcing system, algorithmic mechanism design and data science, e-health and e-learning, etc.

Wenjie Cai is a master student in telecommunications and information engineering at Nanjing University of Posts and Telecommunications (NUPT). His main research interests include deep learning, and recommender systems.

Qun Jin is a full professor at the Networked Information Systems Laboratory, Department of Human Informatics and Cognitive Sciences, and Faculty of Human Sciences, Waseda University, Japan. He has been extensively engaged in research works in the fields of computer science, information systems, and social and human informatics. He seeks to exploit the rich interdependence between theory and practice in his work with interdisciplinary and integrated approaches. His recent research interests include human-centric ubiquitous computing, behavior and cognitive informatics, big data, data quality assurance and sustainable use, personal analytics and individual modeling, intelligence computing, blockchain, cyber security, cyber-enabled applications in healthcare, and computing for well-being. He is a senior member of ACM, IEEE, and Information Processing Society of Japan (IPSJ).

Jianhua Ma received the BS and MS degrees in communication systems from National University of Defense Technology (NUDT), China, in 1982 and 1985, respectively, and the PhD degree in information engineering from Xidian University, China, in 1990. He has joined Hosei University since 2000. He is currently a professor at Digital Media Department in the Faculty of Computer and Information Sciences, Hosei University, Japan. He is a member of IEEE and ACM. He has edited 10 books/proceedings, and published more than 150 academic papers in journals, books, and conference proceedings. His research interest is ubiquitous computing.