DCNet: A Dual Branch Model of Convolutional Attention Over Attention and Enhanced-CNN for Click-Through Rate Prediction

Bin Yang
China Unicom Research Institute
Beijing, China
researcher_yang@outlook.com

Tianmu Sha
Beijing University of Posts and Telecommunications
Beijing, China
stianmu@bupt.edu.cn

Ziyang Wang
Peking University
Beijing, China
wzyyzw@stu.pku.edu.cn

Ying Xing*
Beijing University of Posts and Telecommunications
Beijing, China
xingying@bupt.edu.cn

ABSTRACT

Click-Through Rate (CTR) prediction is a key task in online advertising and recommender systems, and it is necessary to effectively model the feature interactions and utilize the features as much as possible, to obtain more accurate prediction results. However, current researches on CTR prediction have some major problems. First, they did not consider different levels of the attention map, which means applying attention mechanisms within and between embedding vectors comprehensively, and they can hardly have a fine-grained grasp of the importance of features as well. Furthermore, global feature interactions are always neglected, leading to some drawbacks in the practical application of CTR prediction. In this paper, we propose a new model named DCNet which is constructed by Dual branches of Convolution related Network for CTR prediction. The Convolutional Attention Over Attention (ConvAOA) branch is specially designed for adapting CTR prediction. It sequentially processes features by column-level and row-level convolutional attention mechanism, which can dynamically increase the attention to the importance of features and reserve meaningful information within and between different features. The Enhanced-CNN (EnCNN) branch generates local patterns (features that are neighboring) and aggregate features to generate new features, which can grasp global feature interactions. EnCNN also adopts the idea of residual networks to retain global information for further feature enhancement as well. These outputs of these two branches are used for further feature interaction and combination. Extensive experiments have been conducted on three real-world datasets, and the experimental results are highly significant in CTR prediction, which show that DCNet is an advanced model that outperforms current mainstream models in terms of AUC and log loss. And it also proves its effectiveness and its applicability in the field of CTR prediction. The resources of DCNet are available at: https://github.com/***.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

CTR Prediction, Attention Mechanism, Enhanced CNN, Deep Neural Network

ACM Reference Format:
Bin Yang, Tianmu Sha, Ziyang Wang, and Ying Xing. 2022. DCNet: A Dual Branch Model of Convolutional Attention Over Attention and Enhanced-CNN for Click-Through Rate Prediction. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Click-Through Rate (CTR) prediction is an important task for advertising and recommender systems [1–4], which estimates the probability that a user will click on a given item. There are massive clicks in the World Wide Web. Therefore, how to predict whether an advertisement is clicked or not through the given information, and establish a reasonable prediction model of click-through rate can help the platform in the World Wide Web to make advertising cooperation decisions. And the performance of CTR prediction models plays an essential role in the revenue of a commercial system [5].

In CTR prediction, if original high-dimensional sparse features are directly fed into the model, it is usually difficult for the model to learn the rules hidden behind the data. Since most features are related to a certain extent, reasonable feature combination can better capture the interactions between features. Therefore, it is necessary to consider how to effectively model the feature interactions and utilize the features as much as possible. Traditional models such as Logistic Regression (LR) [6], tree models [7], and Bayesian models [8] lack learning and representation of feature interactions. Factorization Machine (FM) [9] and its variants [10–12] were proposed to model the interactions of pairs of features as inner products of vectors. With the increasingly wide application of deep learning (DL), some DL models have been proposed for CTR prediction, such as Wide & Deep [13], DeepFM [1], and xDeepFM [14]. These models feed raw features into deep neural networks to learn feature interactions in an explicit or implicit way.

Convolutional neural network (CNN) has been the cornerstone to achieve breakthrough results in the field of computer vision [15–17]. And CNN has been applied in CTR prediction [18] in recent years. Because the design of shared weights and pooling mechanisms reduces the number of parameters for finding important local patterns, and can be used to identify sparse but important feature interactions. Unlike traditional feature embedding methods in CTR prediction which mostly use embedding vectors, CNN performs feature processing in the form of embedding matrix. In addition, the research based on CNN for CTR prediction has just started. Few studies can prove the effectiveness of CNN for CTR prediction. How to fully adapt CNN to CTR in order to give full play to its advantages is waiting to be explored.

There are some main concerns in recent researches of CTR prediction. Convolutional neural network (CNN) is rarely used in CTR prediction at present, and it is not well adapted to utilize the advantage of CNN. First, different levels of attention can be interpreted as attention mechanisms applying within and between embedding vectors. But recent studies did not consider different levels of the attention map comprehensively, which might make meaningful information lost [19, 20]. Different features bear different importance for the target task. Assuming the features to have equal weights is unable to effectively identify the most important features, and it also reduces the efficiency of learning. Additionally, Useful interactions are usually sparse compared to the combinatorial space of raw features, making it difficult to learn them in a low cost way [4]. Second, using CNN alone to extract and process features can only exploit local feature interactions, and will cause many useful global feature interactions to be lost [21]. Therefore, recent studies can hardly utilize the advantages of CNN into CTR prediction.

In this paper, we propose a new model called Dual branch Convolution related Network (DCNet) for the CTR prediction task. The core structure of the model has two main branches, namely, the Convolutional Attention Over Attention (ConvAOA) branch and the Enhanced-CNN (EnCNN) branch, for the purpose of solving the problems mentioned above.

The ConvAOA branch perfectly solves the problem of not considering different levels of attention map comprehensively and neglecting feature importance. It performs attention over attention, which consists of column-level attention and row-level attention to sequentially process features. Considering different levels of attention can grasp relationships within and between features to the maximum extent. The ConvAOA structure can dynamically learn feature importance and perform adaptive feature refinement.

The EnCNN branch is designed to overcome the weakness of classical CNN which losses global feature interactions. First, local features are extracted by classical CNN. Then global feature interactions are considered by aggregating local features. Moreover, through the residual network, we use the shortcut mechanism between different convolutional layers to learn global features. In the meanwhile, global information is retained during the CNN calculation.

After the features are processed by two branches, the output is sent to the interaction layer for further feature fusion. Finally, they are combined and fed into a multilayer perceptron (MLP) to obtain the prediction results. Experimental results on three real large-scale datasets show that DCNet clearly surpasses other state-of-the-art models, fully demonstrating the effectiveness of our model.

In summary, our main contributions are as follows.

- Given the success of the application of attention in computer vision, we firstly proposed ConvAOA in the field of CTR, to dynamically learn the weights of features using attention mechanism in both column-level (inter-field-level) and row-level (between-fields-level), which is an improvement especially suitable for CTR scenarios. And it can significantly promote the performance of CTR prediction. This method allows the model to pay more attention to the importance of features and interactions.
- We propose a new CNN model called EnCNN. The basic CNN structure learns useful local feature patterns. In the meanwhile, global feature interactions based on the local patterns are generated. In addition, a residual network is used to retain information during the CNN calculation, which can improve the accuracy. This structure is able to effectively improve performance of CTR prediction.
- Experimental results performed on three real large-scale datasets, outperform the results obtained using other state-of-the-art deep models, demonstrating the overall effectiveness of DCNet.

The rest of the paper is organized as follows. In Section 2, we review the related works associated with our proposed model. The specific details of DCNet are described in Section 3. In Section 4, we present the results of the experiments conducted on real-world datasets, as well as experimental related explorations and discussions. Section 5 summarizes and concludes this paper.

2 RELATED WORK

In this section, we will introduce several mainstream types of models or components applied in CTR prediction.

2.1 Classical Models in CTR Prediction

The early stage CTR prediction was solved mainly by Logistic Regression (LR) [6]. Gradient Boosting Decision Tree (GBDT) [7] is a commonly used nonlinear model. The Factorization Machine (FM) model [9] was proposed to solve the problem of feature combination in sparse data scenarios, and is widely used in advertising and recommender systems. Field-aware Factorization Machine (FFM) [10, 11] is an advanced version of FM. It further refines the hidden vector in FM by introducing the concept of field.

Deep learning has been widely applied in CTR prediction models [22–24]. Factorization Machine supported Neural Network (FNN) [25] pretrains the embedding of the original features. Product-based Neural Networks (PNN) [4, 26] can combine features between different feature domains. Wide & Deep [13], DeepFM [1], xDeepFM [14], AutoInt [27], IFM [28], DIFM [29], DCNV2 [30], are also some advanced deep learning models in CTR prediction.

2.2 Attention Mechanism in CTR Prediction

Attention mechanism is widely used in the area of computer vision. Squeeze-and-Excitation Network (SENet) [31] utilizes the attention mechanism to improve the representation of network. FiBiNET [15]

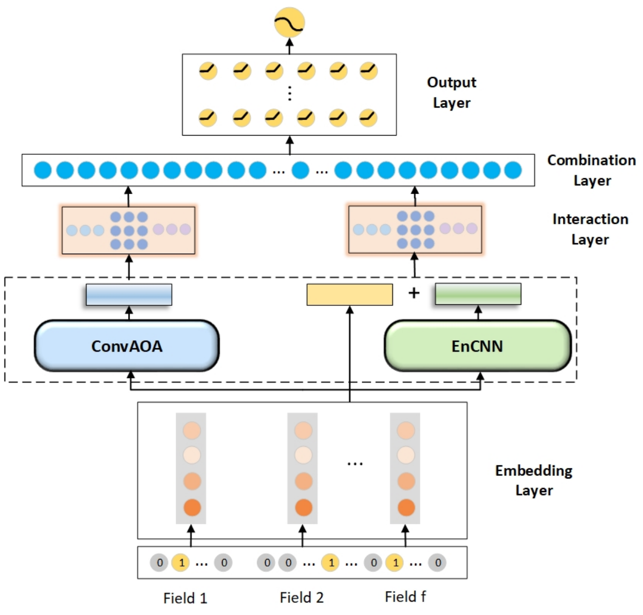

Figure 1: The framework of DCNet.

can learn the feature importance and fine-grained feature interactions dynamically, which utilized SENet to increase the weights of important features for a specific CTR prediction task, and decrease the weights of uninformative features through the SENet mechanism.

2.3 CNN in CTR Prediction

Convolutional neural network (CNN) has a wide range of applications in computer vision [17, 32, 33]. There are models [18, 21, 34] using CNN for CTR prediction in recent years. Convolutional Click Prediction Model (CCPM) [21] applies convolutional layers to explore the correlation of local features. But this model mainly learns the interactions between local features. A study [34] proposed that generating new features has an impact on the performance of the CNN-based model. However, the strengths of CNN are not fully exploited in CTR scenarios.

3 PROPOSED MODEL

We propose DCNet based on ConvAOA and EnCNN for feature interaction in CTR prediction. In this section, we describe the architecture of DCNet, as shown in Figure 1.

DCNet consists of the following parts: embedding layer, dual convolutional branch architecture of ConvAOA and EnCNN, interaction and combination layer, and output layer.

3.1 Embedding Layer

The Embedding layer is the first layer of the model and its main function is to convert a feature into a vector. In most CTR prediction tasks, the data are collected and represented in a multi-field form. When there are a huge number of features, the usual practice is to convert them to embeddings.

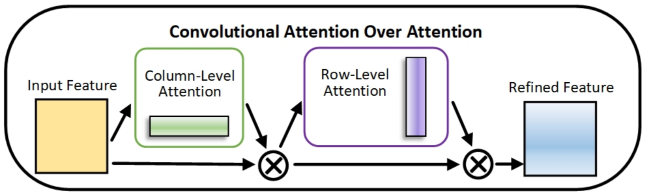

Figure 2: The architecture of ConvAOA. Including column-level at-tention and row-level attention.

The embedding layer is able to embed high-dimensional sparse vectors into low-dimensional dense real-valued vectors, where each field in an instance can be represented as a low-dimensional vector. Traditional feature embedding methods in CTR prediction usually use vectors as the form of embedding. However, embedding matrix is more suitable for convolution operation. Therefore, We can combine different vectors into a embedding matrix. We represent the output of the embedding layer as an embedding matrix $E = \left[e_1, e_2, \cdots, e_f\right]$, $E \in R^{f \times k}$, where f denotes the number of fields, and k denotes the embedding size.

3.2 Convolutional Attention Over Attention

Different features have different importance to the target task. We introduce ConvAOA to pay attention to different importance of features. This attention structure is specially designed for CTR prediction, which is an effective attention mechanism with CNN. The attention map can be derived sequentially along two independent levels, column-level and row-level attention, which deal with relationships within and between embedding vectors. And then the attention map is multiplied by the input feature embedding matrix for further feature refinement. The structure of ConvAOA is given in Figure 2.

Take the feature embedding matrix $E \in R^{f \times k}$ as input, ConvAOA derives the column-level attention map $M_c(E) \in R^{f \times 1}$ and the row-level attention map $M_r(E) \in R^{1 \times k}$ sequentially, as shown in Figure 2. In column-level attention, we focus on column-level itself, so the column dimension is pooled to 1. In row-level attention, We apply one-dimensional convolution on two-dimensional matrix for the following reasons. In the CTR model, each dimension of the embedding represents a distinct aspect of an item. Therefore, each column of the resulting matrix obtains distinct features from the activation matrix. The overall attention process can be summarized as:

$$E' = M_c(E) \otimes E \tag{1}$$

$$E'' = M_r\left(E'\right) \otimes E' \tag{2}$$

where E' denotes the intermediate output by column-level attention, E'' denotes the final output, and $\otimes$ denotes element-wise multiplication.

Column-Level Attention Part. We compute the column-level attention map by exploiting the inner-field-level relationship of the features. Figure 3 depicts the computation process of column-level attention. We first use the average-pooling and max-pooling operations to generate two different descriptors E^c_{avg} and E^c_{max}.

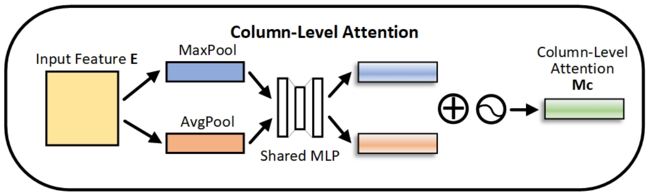

Figure 3: The structure of column-level attention part.

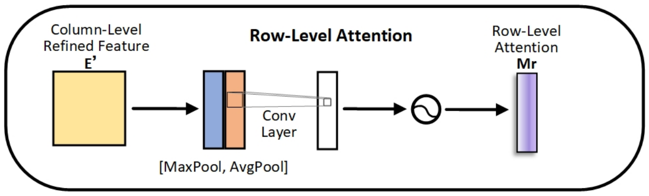

Figure 4: The structure of row-level attention part.

They are fed into a network which is composed of MLP to generate column-level attention map $M_c(E) \in R^{f \times 1}$, The weights of MLP are shared for both E^c_{avg} and E^c_{max}. The activation size of the hidden layer in MLP is set to $R^{f/r \times 1}$. where r is the reduction rate. The column-level attention is calculated as shown below:

$$\begin{aligned} M_c(E) &= \sigma(MLP(AvgPool(E)) + MLP(MaxPool(E))) \\ &= \sigma\left(W_1\left(W_0\left(E^c_{avg}\right)\right) + W_1\left(W_0\left(E^c_{max}\right)\right)\right) \end{aligned} \tag{3}$$

where σ denotes the sigmoid function, the weights of the MLP are $W_0 \in R^{f/r \times f}$, $W_1 \in R^{f \times f/r}$, and the ReLU activation function is after W_0.

Row-Level Attention Part. We use the column-level refined feature to generate a row-level attention map. Figure 4 depicts the computation process of row-level attention part. We first apply average-pooling and max-pooling along the column-level axis and concatenate them along the column-level axis to generate two maps, E^r_{avg} and E^r_{max}. Then we use a convolutional layer on the maps to generate a row-level attention map, $M_r \in R^{1 \times k}$, which encodes the places to be emphasized or suppressed. This process is computed as shown below:

$$\begin{aligned} M_r(E) &= \sigma(c([AvgPool(E); MaxPool(E)])) \\ &= \sigma\left(\left[E^r_{avg}; E^r_{max}\right]\right) \end{aligned} \tag{4}$$

where σ, c denotes the sigmoid and convolution function.

In short, the ConvAOA architecture adaptively refines the intermediate feature maps by means of two sequential submodules: the column-level and row-level modules, thus focusing on important features and suppressing unnecessary ones.

3.3 Enhanced-CNN By ResNet And Aggregation

EnCNN is designed to identify and retain useful feature interactions and then automatically generate new features based on CNN. This structure perfectly adapts CNN to CTR prediction and fully utilizes the advantages of CNN. Figure 5 depicts the architecture of Enhanced-CNN.

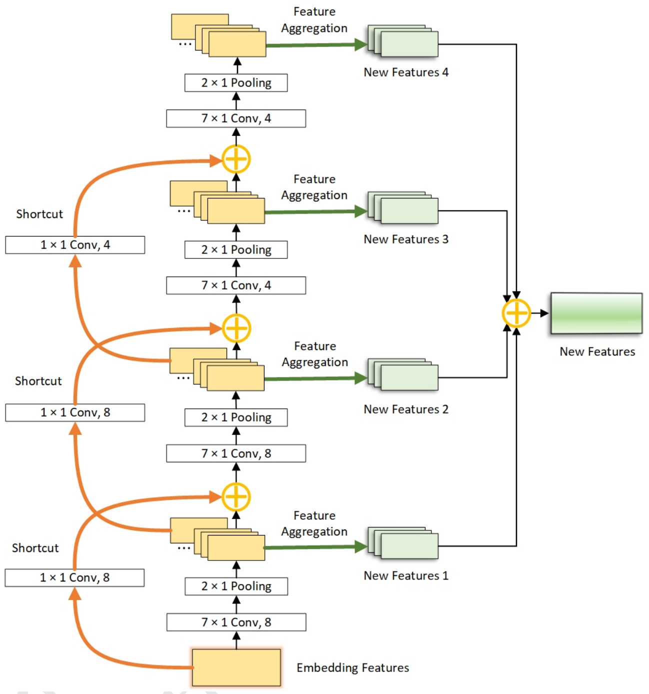

Figure 5: The architecture of Enhanced-CNN.

We used EnCNN to capture the global feature interactions and reserve information of features. EnCNN learns useful local feature patterns while feature aggregation generates global feature interactions based on the local patterns. In addition, we use residual learning here to further retain global information of features through shortcut connections, which can feed previous information to next convolutional block. Thus, important features can be generated and extracted efficiently.

a. Convolutional Layer. The embedding matrix $E \in R^{f \times k}$ is considered as an input, where f is the number of fields and k is the embedding size. The embedding matrix is reshaped into a matrix $E^1 \in R^{f \times k \times 1}$ which is then fed into the convolutional neural network. The convolutional layer convolves a weight matrix $W^1 \in R^{h^1 \times 1 \times 1 \times n^1_c}$ to capture the local feature interactions, where h^1 is the height of the first convolutional weight matrix and n^1_c is the number of feature maps in the first convolutional layer. Assuming that the output of the first convolutional layer is denoted as $C^1 \in R^{f \times k \times n^1_c}$, the convolutional layer is computed as follows:

$$C^1_{u,v,i} = tanh\left(\sum_{n=1}^{1}\sum_{j=1}^{h^1} E^1_{u+j-1,v,n} W^1_{j,1,1,i}\right) \tag{5}$$

where $C^1_{u,v,i}$ denotes the i -th feature map in the first convolutional layer, u, v denote the row and column indices of the i -th feature map,and $tanh$ is the activation function. The pooling layer uses max-pooling. We take h' to be the height of the pooling layer, and the width of the pooling layer is set to be 1. The output of the first pooling layer is $P^1 \in R^{f/h' \times k \times n^1_c}$, and the pooling layer is computed as follows:

$$P^1_{u,v,i} = max\left(C^1_{uh',v,i}, \cdots, C^1_{uh'+h'-1,v,i}\right) \tag{6}$$

b. Residual Network. For models using CNN, multi-layer networks are often built by stacking new layers upwards. Residual learning is better than direct learning of the original features. It allows the stacking layer to learn global new features based on previous features and thus have better performance. In order to retain global information of the original embedding matrix and learn new features during the deepening of CNN layers, we use the shortcut mechanism between different convolutional layers with the idea of residual learning. A new mapping is generated by a 1x1 convolution kernel to make the dimensions the same. Then the input feature map is added to the output feature map. The residual units and the learning features from shallow to deep can be represented as follows:

$$X^l = ReLU\left(P^{l-1} + F\left(P^{l-2}, W^{l-2}\right)\right) \tag{7}$$

where X^l denotes the input of the l -th residual block. F is the residual function. P^{l-1} and P^{l-2} denotes the output of the pooling function.

c. Feature Aggregation. The stage output P^1 after the pooling layer only extracts neighboring features, and no consideration is given to global feature interactions. Therefore, to obtain global interactions of non-neighboring features, a fully connected layer is used to aggregate local neighboring feature patterns and generate significant new features for feature enhancement. The weight matrix is denoted as $AW^1 \in A^{f/h'kn^1_c \times f/h'kn^1_r}$, and the bias is denoted as $AB^1 \in A^{f/h'kn^1_r}$, where n^1_r denotes the number of new feature maps in the first aggregation operation. The output of the first feature aggregation is described as follows:

$$A^1 = tanh\left(P^1 \cdot AW^1 + AB^1\right) \tag{8}$$

d. Concatenation. Assuming that there are t feature aggregation operations, new features generated in the i -th round are denoted as A^i. After combining the new features of each round, we can obtain the final new features A, which can be expressed by the following form:

$$A = \left(A^1, A^2, \cdots, A^t\right) \tag{9}$$

Then original features and new features are concatenated to obtain the new embedding matrix E'''.

$$E''' = \left(E^T, A^T\right) \tag{10}$$

Finally, E''' is fed into the interaction layer.

3.4 Interaction and Combination Layer

The interaction layer uses the bilinear-interaction method [15] to learn feature interactions. It outputs a ConvAOA interaction vector $a = E'' = [a_1, a_2, \cdots, a_n]$, and a EnCNN interaction vector $b = E''' = [b_1, b_2, \cdots, b_n]$, where $a_i, b_i \in R^k$ are vectors, n is the number of field interactions.

The combination layer concatenates two interaction vectors. It can be represented in the following form:

$$\begin{aligned} c &= [a_1, \cdots, a_n, b_1, \cdots, b_n] \\ &= [c_1, \cdots, c_{2n}] \end{aligned} \tag{11}$$

where $c_i \in R^k$.

Finally, we feed it into output layer for prediction.

3.5 Output Layer

The output layer consists of a multilayer perceptron (MLP). Let $o^0 = [c_1, \cdots, c_{2n}]$ denote the output of the combination layer. Then, o^0 is fed into the MLP as:

$$o^l = \sigma\left(DW^l o^{l-1} + b^l\right) \tag{12}$$

where o^l is the output of the l -th layer and σ is the activation function. DW^l and b^l represents the weight and bias. The final output in the last hidden layer can be expressed as follows:

$$\widehat{y} = sigmoid\left(DW^{m+1} o^m + b^{m+1}\right) \tag{13}$$

where $\widehat{y}$ is the predicted value of CTR and we use sigmoid here, m is the number of layers. We use cross entropy as the loss function:

$$loss = -y log \widehat{y} - (1-y)\, log\,(1-\widehat{y}) \tag{14}$$

where $y \in \{0,1\}$ is the label, and $\widehat{y} \in (0,1)$ is the predicted CTR value.

4 EXPERIMENTS

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

(RQ1) How does our model perform, compared to the state-of-the-art methods for CTR prediction?

(RQ2) How does ConvAOA structure in our model boost the performance of our model?

(RQ3) How does EnCNN structure in our model boost the performance of our model?

(RQ4) How do the parameter settings of networks influence the performance of our model?

We will answer these questions after presenting some fundamental experimental settings.

4.1 Experimental Setup

The experimental setup of this paper is shown as follows:

a. Datasets. We evaluate our model on the following three datasets, and we randomly divide the datasets into two parts, 80% for training and 20% for test. Our test set is larger and the results are more convincing with the 8:2 division method because larger test set is able to measure the generalization effect of the model better. Besides, this division method is more convincing.

Criteo: The Criteo dataset contains instances with 45 million advertising click records. There are 13 continuous feature fields and 26 anonymous classification fields in the Criteo dataset.

Avazu: The Avazu dataset contains advertising click logs with 40 million data instances. For each click record, there are 24 fields that represent elements of individual ad impressions.

Table 1: Some Typical Commands

Command	A Number	Comments
`\author`	100	Author
`\table`	300	For tables
`\table*`	400	For wider tables

MovieLens-1M: The MovieLens-1M dataset contains 1 million rating data. We treat samples with a rating less than 3 as negative samples, the others as positive samples.

b. Baselines. To verify the effectiveness of the joint ConvAOA and EnCNN architecture, we conducted comparison experiments with some advanced models, including Wide & Deep [13], DeepFM [1], xDeepFM [14], FiBiNET [15], AutoInt [27], IFM [28], DIFM [29], DCNV2 [30].

c. Evaluation Metrics. The evaluation metrics are Area Under ROC (AUC) and log loss.

AUC: AUC is a widely used metric for evaluating classification problems and is a good measure of CTR prediction.

Log loss: Log loss is a metric based on probability.

d. Parameters Settings. DCNet is implemented with Pytorch. The embedding size is set to 10. The depth of output layers is set to 3. All activation functions are RELU. The number of neurons per layer is 400, and the dropout rate is 0.2. For the EnCNN part, the number of layers in this structure is set to 4, and the number of convolutional kernels at four layers is [8, 8, 4, 4]. For the optimization method, we use Adagrad with the learning rate as 0.02, and the batch size is 4096. Each experiment was run three times and the results were averaged at last.

4.2 Performance Comparison (RQ1)

In this section, we compare the performance of different models on the Criteo, Avazu and MovieLens-1M datasets.

Table 1 summarizes the overall performance of all the compared models. On the three evaluated datasets, our model achieves the best performance. On the Criteo, Avazu and MovieLens-1M datasets, our model achieved AUC of 0.8029, 0.7727, and 0.8249, and log loss of 0.4482, 0.3836 and 0.3404, obtaining almost all the best results.

On three different datasets, our model outperforms FiBiNET by 0.60%, 0.69%, and 0.19% in AUC, 0.55%, 0.43% and 0.25% in log loss. This result shows that combining ConvAOA with EnCNN is an effective CTR prediction method for different datasets. On the one hand, the prediction using ConvAOA can dynamically adjust the importance of features. On the other hand, EnCNN can extract global feature interactions, which achieves further feature enhancement. As a result, our model achieves better performance.

In the field of CTR prediction, due to the large amount of advertising, the 0.1% increase of AUC is already a fairly significant improvement, which can bring considerable economic benefits. The improvement brought by our model in CTR prediction is the best under the 8:2 division ratio. Compared with the baseline models themselves, the number of parameters of our model is maintained in the same order of magnitude while the preformance is significantly improved. Therefore, our model is quite advanced and effective for CTR prediction.

4.3 Effectiveness of ConvAOA (RQ2)

We conducted a series of comparison experiments on ConvAOA. Each variant is created by removing or replacing some components. Table 2 shows the results of this part.

We represent original features as OF in the following part of this paper, and the experiments we designed for ConvAOA are shown as follows:

SENet (SE): Only using SENet for feature processing.

SENet+OF (SE+OF): Representing the addition of the original features to complement the embeddings of the SENet output, which is FiBiNET.

ConvAOA (CA): Only using ConvAOA for feature processing.

ConvAOA+OF (CA+OF): Representing the addition of original features and the embeddings output by ConvAOA.

We replicate the model of FiBiNET in which the SENet structure is used to process the features, and then the interaction vectors are combined with the vectors extracted from the original features. The results obtained while using only SENet without original feature vectors and while adding the original feature vectors to SENet are shown in the first two rows of Table 2. The results indicate that for FiBiNET, the performance is greatly improved by combining the original feature vectors.

Our model uses the more advanced ConvAOA structure to process the features dynamically. We can see that under the condition of using only ConvAOA without combining the original features, we can achieve AUC of 0.8006 and 0.7720 and log loss of 0.4504 and 0.3842 on the Criteo and Avazu datasets, which is fairly impressive.

It can also be seen that the original features play an important role in the performance of FiBiNET, and once the original features are removed, the performance of the model will have a significant decline. For instance, AUC declines 1.21% and 2.06% on Criteo and Avazu datasets using FiBiNET, which is a huge loss in CTR prediction. In contrast, interaction vectors without combining original features do not have a significant effect on ConvAOA. AUC only declines 0.01% and 0.05% on Criteo and Avazu datasets using ConvAOA, which is negligible. In addition, when original features are removed, the parameters of our model decrease significantly and the performance of our model remains basically unchanged. This shows that the ConvAOA component we proposed is more robust and achieves excellent and stable results without relying on the complement of original features. The redundancy and complexity of the model are reduced while achieving high performance.

4.4 Effectiveness of EnCNN (RQ3)

In this subsection, we will focus on the role of Enhanced-CNN component on the model performance through ablation experiments. We summarize the results of the experiments in Table 3 and have the following findings.

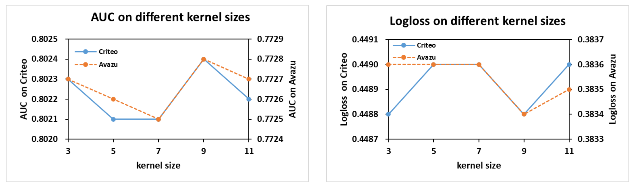

Figure 6: Parameters study of kernel size of row-level attention.

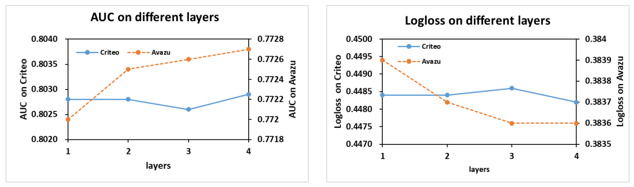

Figure 7: Parameters study of number of convolutional layers.

SENet (SE): Only using SENet for feature processing.

SENet+EnCNN (SE+EC): Using SENet and EnCNN.

SENet+EnCNN+OF (SE+EC+OF): Representing the addition of EnCNN on top of the FiBiNET model.

ConvAOA (CA): Only using ConvAOA for feature processing.

ConvAOA+EnCNN (CA+EC): EnCNN and ConvAOA.

ConvAOA+EnCNN+OF (CA+EC+OF): Our model.

In FiBiNET, the output from SENet is supplemented by the original embeddings. In this case, although the feature space can be enriched to some extent, the original features are not maximally utilized. Therefore, raw features can be further processed by CNN to enhance the feature representation.

Better results are obtained after adding the EnCNN component. Specifically, when we use SENet and EnCNN for feature processing on the Criteo dataset, the AUC is improved by 1.57% and the log loss is optimized by 1.41% compared with the method only using SENet embedding. This is an impressive improvement in CTR prediction, when the model without EnCNN has already achieved high performance. Our focus on local features and global features maximizes the benefits of CNN. The results above indicate the importance of the EnCNN.

4.5 Hyper-parameters Investigation (RQ4)

In this subsection, we will investigate the impact of hyper-parameters on the model by changing one hyper-parameter while other hyper-parameters are kept constant.

a. Kernel Size of Row-Level Attention in ConvAOA.

Different sizes of convolution kernels will generate different number of features in the calculation. In order to investigate the effect of kernel size on row-level attention, we set up a series of convolutional kernels of different sizes from 3 to 11 for the experiment, as shown in Figure 6.

Best experimental results are obtained in both datasets with the kernel size of 9. However, the improvement is not notable with calculation amount becoming extremely large. Therefore, setting kernel size to 3 is more cost-effective.

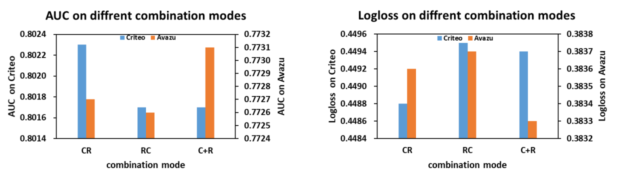

Figure 8: Impact of combination mode on ConvAOA.

b. Number of Convolutional Layers in EnCNN.

As shown in Figure 7, the experimental results on the Criteo and Avazu dataset show that the performance improves as the number of convolutional layers increases. Therefore, the EnCNN part can capture useful feature interactions better by choosing a reasonable number of layers.

5 DISCUSSION

Combination Mode in ConvAOA. ConvAOA structure consists of column-level attention part and row-level attention part, and both parts have their own functions. Different combination modes on two attention parts may have an impact on ConvAOA. We represent column-level attention as C and row-level attention as R. The combination modes are CR and RC for serial connection, and C+R for parallel connection. We explored the experimental effects under different combination modes, as shown in Figure 8.

The difference between these results is tiny. The serial mode CR gets the best overall performance among three modes. The parallel mode C+R only does better on Avazu dataset.

6 CONCLUSION

In this paper, we propose a new model named DCNet for feature interaction to accomplish the CTR prediction task. We innovatively adapted CNN to CTR prediction, giving full play to the advantages of convolutional to improve the performance of CTR prediction, which was rarely studied before. The model consists of two branches. In the ConvAOA branch, the attention map can be derived sequentially along two independent convolutional attention mechanism to deal with relationships within and between embedding vectors. this architecture can dynamically learn the importance of features and perform feature refinement. In the EnCNN branch, we take advantage of CNN to identify useful local patterns, and address the weaknesses of classical CNN by introducing a feature aggregation operation to generate global new features. In addition, we use the idea of residual network to retain global information. After experimental validation on different datasets, DCNet outperforms other state-of-the-art models in CTR prediction with remarkable performance, proving its effectiveness and huge application potential in advertising and recommendation.

REFERENCES

[1] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[2] Yanwu Yang and Panyu Zhai. Click-through rate prediction in online advertising: A literature review. *Information Processing & Management*, 59(2):102853, 2022.
[3] Congcong Liu, Yuejiang Li, Xiwei Zhao, Changping Peng, Zhangang Lin, and Jingping Shao. Concept drift adaptation for ctr prediction in online advertising systems. *arXiv preprint arXiv:2204.05101*, 2022.
[4] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems (TOIS)*, 37(1):1–35, 2018.
[5] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018.
[6] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230, 2013.
[7] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the eighth international workshop on data mining for online advertising*, pages 1–9, 2014.
[8] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Omnipress, 2010.
[9] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.
[10] Yuchin Juan, Damien Lefortier, and Olivier Chapelle. Field-aware factorization machines in a real-world online advertising system. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 680–688, 2017.
[11] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM conference on recommender systems*, pages 43–50, 2016.
[12] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–22, 2012.
[13] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.
[14] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1754–1763, 2018.
[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
[16] Xiangyang Xu, Mian Zhao, Peixin Shi, Ruiqi Ren, Xuhui He, Xiaojun Wei, and Hao Yang. Crack detection and comparison study based on faster r-cnn and mask r-cnn. *Sensors*, 22(3):1215, 2022.
[17] Praveen Kulkarni and TM Rajesh. A multi-model framework for grading of human emotion using cnn and computer vision. *International Journal of Computer Vision and Image Processing (IJCVIP)*, 12(1):1–21, 2022.
[18] Patrick PK Chan, Xian Hu, Lili Zhao, Daniel S Yeung, Dapeng Liu, and Lei Xiao. Convolutional neural networks based click-through rate prediction with multiple feature sequences. In *IJCAI*, pages 2007–2013, 2018.
[19] Hongchang Gao, Deguang Kong, Miao Lu, Xiao Bai, and Jian Yang. Attention convolutional neural network for advertiser-level click-through rate forecasting. In *Proceedings of the 2018 World Wide Web Conference*, pages 1855–1864, 2018.
[20] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 169–177, 2019.
[21] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. A convolutional click prediction model. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1743–1746, 2015.
[22] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. Deep learning for click-through rate estimation. *arXiv preprint arXiv:2104.10584*, 2021.
[23] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
[24] Aljo Jose and Sujala D Shetty. Distilledctr: Accurate and scalable ctr prediction model through model distillation. *Expert Systems with Applications*, 193:116474, 2022.
[25] Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data. In *European conference on information retrieval*, pages 45–57. Springer, 2016.
[26] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1149–1154. IEEE, 2016.
[27] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1161–1170, 2019.
[28] Yantao Yu, Zhen Wang, and Bo Yuan. An input-aware factorization machine for sparse prediction. In *IJCAI*, pages 1466–1472, 2019.
[29] Wantong Lu, Yantao Yu, Yongzhe Chang, Zhen Wang, Chenhui Li, and Bo Yuan. A dual input-aware factorization machine for ctr prediction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3139–3145, 2021.
[30] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*, pages 1785–1797, 2021.
[31] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
[33] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.
[34] Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. Feature generation by convolutional neural network for click-through rate prediction. In *The World Wide Web Conference*, pages 1119–1129, 2019.