



Article ID 1007-1202(2024)03-0198-11 DOI <https://doi.org/10.1051/wujns/2024293198>

Cite this article: ZHOU Liliang, YUAN Shili, FENG Zijian, *et al.* A Lambda Layer-Based Convolutional Sequence Embedding Model for Click-Through Rate Prediction[J]. *Wuhan Univ J of Nat Sci*, 2024, 29(3): 198-208.

A Lambda Layer-Based Convolutional Sequence Embedding Model for Click-Through Rate Prediction

□ ZHOU Liliang¹, YUAN Shili², FENG Zijian², DAI Guilan³, ZHOU Guofu^{4†}

1. Tenth Research Institute, China Electronics Technology Group Corporation, Chengdu 610000, Sichuan, China;
2. Department of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China;
3. Research Institution of Information Technology, Tsinghua University, Beijing 100084, China;
4. School of Computer Science, Wuhan University, Wuhan 430072, Hubei, China

© Wuhan University 2024

Abstract: In the era of intelligent economy, the click-through rate (CTR) prediction system can evaluate massive service information based on user historical information, and screen out the products that are most likely to be favored by users, thus realizing customized push of information and achieve the ultimate goal of improving economic benefits. Sequence modeling is one of the main research directions of CTR prediction models based on deep learning. The user's general interest hidden in the entire click history and the short-term interest hidden in the recent click behaviors have different influences on the CTR prediction results, which are highly important. In terms of capturing the user's general interest, existing models paid more attention to the relationships between item embedding vectors (point-level), while ignoring the relationships between elements in item embedding vectors (union-level). The Lambda layer-based Convolutional Sequence Embedding (LCSE) model proposed in this paper uses the Lambda layer to capture features from click history through weight distribution, and uses horizontal and vertical filters on this basis to learn the user's general preferences from union-level and point-level. In addition, we also incorporate the user's short-term preferences captured by the embedding-based convolutional model to further improve the prediction results. The AUC (Area Under Curve) values of the LCSE model on the datasets Electronic, Movie & TV and MovieLens are 0.870 7, 0.903 6 and 0.946 7, improving 0.45%, 0.36% and 0.07% over the Caser model, proving the effectiveness of our proposed model.

Key words: click-through rate prediction; deep learning, attention mechanism; convolutional neural network

CLC number: TP183

0 Introduction

With the advent of the era of smart economy, while the network services are developing in quantity, service providers pay more and more attention to the user experience,

and take user satisfaction as the guide to improve the fineness of services^[1]. Among them, the development of e-commerce is also devoted to customizing user experience. Many e-commerce platforms like Amazon and Google Store have introduced a click-through rate

Received date: 2023-12-29

Foundation item: Supported by the National Natural Science Foundation of China (62272214)

Biography: ZHOU Liliang, male, Senior engineer, research direction: avionics information systems and sensor management. E-mail: zhoul@163.com

† Corresponding author. E-mail: gfzhou@whu.edu.cn

(CTR) prediction system to improve economic benefits. The CTR prediction system can evaluate massive service information based on the user's historical information, screen out the products most likely favored by users, and push them to customers on the e-commerce platform. Existing research shows that users know little about other products that meet their personal preferences, and the search process takes a lot of time and effort^[2]. The application of the CTR prediction system can filter and effectively retain information, enable users to efficiently access relevant data, increase the length of time users stay on the platform, and thus achieve the ultimate goal of improving economic benefits^[3,4].

The main method of CTR prediction is to find suitable features from the user dimension, service dimension or time dimension based on user profile, user history and other information to model, and then to predict the probability of users clicking on the service. The click behavior refers to the process of accessing or evaluating products according to the user's interest in the network service platform. In the field of computational advertising, the revenue of an advertising platform relies on the product of the cost per click^[5] and the click-through rate (CTR)^[6], so the accuracy of CTR prediction will significantly affect the revenue of online advertising platforms^[7]. In the field of recommender systems, CTR prediction often affects the performance of recommendations. CTR is one of the main evaluation indicators of recommender systems^[8], and the CTR prediction model is frequently used in the ranking stage to recommend the ones that users are more inclined to click on. Therefore, improving the accuracy of CTR prediction has become the key to the research, which plays an essential role in promoting the development of e-commerce in the smart economy.

In actual e-commerce scenarios, users' short-term preferences reflect recent needs, and providing services similar to recent browsing records can indeed efficiently satisfy customers. However, user's demand for similar products is often limited. For example, after browsing a large number of smartphone entries, a customer chooses one to buy, but has no demand for the second. Another example is that after watching a historical documentary on a video website, customers want to change their taste and choose a science fiction film. This shows that it is not enough to only focus on the capture of short-term preferences, which limits recommended services to the same field. In order to improve the user's overall experi-

ence level, it is necessary to provide novel product information to attract the user's interest at all times, which requires capturing long-term preferences.

Sequence modeling is one of the main research directions of the CTR prediction model based on deep learning. It is a common method to explore the influence of user preferences hidden in the user's historical click behavior. The short-term and general preferences expressed from the user's recent behavior and entire historical behavior respectively will have different degrees of influence on the user's subsequent behavior. Ideally, the system should combine both when making predictions.

A variety of sequence models based on CTR have been proposed. Markov chain^[9,10] is a method to predict the user's next click behavior based on recent click behavior, but this method ignores the influence of the user's general preference on the user's click behavior. Wang *et al.*^[11] used embedding representation and fusion to express user general preferences and short-term preferences, but the information captured by embedding only is often limited. Tang *et al.*^[12] used a vertical filter in the convolutional neural network to fuse the user's recently clicked item information (point-level, which means user information) to predict the user's next click behavior. In addition, the Caser model^[12] proposed by them also fuses item information from the embedding element's perspective (union-level, which means service information) through a horizontal filter. The above models focus more on user's short-term preferences, and the research on users' general preferences is somewhat insufficient. The capture of user's general preferences only uses the method of user information embedding or item preference fusion, but in fact, it can be combined with the attention mechanism^[13] to be further optimized.

Aiming at the problem of combining user's short-term preferences and general preferences, based on the Caser model, this paper improves the capture of user's general preferences. We add a convolutional sequence embedding module based on the Lambda layer, combine the information of the embedding layer to capture the user's general preferences, and finally propose a Lambda layer-based Convolutional Sequence Embedding (LCSE) CTR prediction model. Different from other CTR prediction models (such as the DSIN model^[14]) that combine convolutional neural networks and attention mechanisms, the LCSE model has the following advantages: (1) In terms of user preference capture, the LCSE model combines the union-level information and point-

level information when using convolutional neural networks; (2) In terms of attention mechanism, the LCSE model uses a Lambda layer with a time complexity of $O(n)$ to replace the common scaling dot-product attention mechanism, which not only improves the prediction results of the model, but also improves the training efficiency. Experimental results on public datasets also demonstrate the effectiveness of our LCSE model.

1 Related Work

1.1 CTR Prediction Models Based on Feature Interaction Learning

The current research focuses on the CTR prediction model based on deep learning. The main research directions include feature interaction learning and sequence modeling.

The Wide & Deep model^[15] is a method proposed by Google in 2016 to improve the overall performance of the CTR prediction model. It combines the linear features of the linear model and the high-order features of the Deep model. The model brought an increase in the app download rate to the Google Play Store, proving its effectiveness. The Deep & Cross model^[16] uses the Cross network to replace the linear model in the Wide & Deep model, and the Cross network can extract high-order cross features with better interpretability, thus improving the expressiveness of the model to achieve better performance on model metrics like logloss. The DeepFM model proposed by Guo *et al.*^[17] uses a factorization machine to replace the logistic regression (LR) model of the Wide part. Compared with the Wide & Deep model, DeepFM has a stronger learning ability for sparse data. It can not only consider the interaction of high-order and low-order features, but also does not require additional manual feature engineering. The model has also achieved good results in the Huawei AppGallery. Based on the DeepFM model and the Deep & Cross model, the xDeepFM model^[18] proposed by the Social Computing Group of Microsoft Research Asia makes the feature interactions occur at the vector level. It has better memory and generalization ability, which can automatically learn high-level feature interactions explicitly and implicitly at the same time. It also has a certain level of improvement in the accuracy of prediction results.

Although these methods can explore the relationships between feature interactions, they ignore the preference information hidden in the user's click sequence,

which also affects the user's click behavior.

1.2 CTR Prediction Model Based on Sequence Modeling

The preferences displayed at different times in the user's historical click sequence will influence the user's subsequent behavior differently. Compared with the general preferences of users in the entire click history, the short-term preferences of users in recent click behaviors will have a greater impact on subsequent behaviors, but the importance of general preferences cannot be ignored as well.

By combining the matrix factorization machine and Markov chain, the Factorizing Personalized Markov Chains (FPMC) model^[19] uses the Markov transition matrix to predict the user's next click behavior according to the user's click behavior at the previous moment, and has also achieved good results in CTR prediction. The Hierarchical Representation model (HRM)^[19], which achieves better results on real-world datasets than the FPMC model, uses a hierarchical structure. It first fuses the recent service information consumed by the user. Then, it is combined with user representation information to obtain the final fusion representation and get the prediction results accordingly. The Caser model stands out by applying two innovative methods within a convolutional neural network. It uses a vertical filter to merge point-level user click data and a horizontal filter for integrating service information through embeddings, aiming to accurately predict user click behavior. The above models focus more on the study of user's short-term preferences. The extraction of user's general preferences only uses user information embedding or service preference fusion. Although the models achieve good prediction accuracy, the overall performance can be further improved by optimizing the capture of general preferences. Recently, many studies based on optimizing the capture of general preferences have introduced attention mechanisms to enhance the capture of general preference information.

The Deep Interest Network (DIN) model^[20] introduces the attention mechanism to obtain the user's general preference by calculating the relationships between the target service and the user's click history, which affects the user's next click behavior. The Short-Term Attention Memory Priority Model (STAMP)^[21] also introduces an attention mechanism to capture general preferences, and proposes a novel short-term attention model. It calculates interest weights according to the context,

and uses user's interests at different times to synthesize the attention vectors. These models only consider the relationships between services when using the attention mechanism, and better results can be achieved if information from the embedding layer is integrated, like the Caser model. Moreover, the traditional self-attention mechanism has high memory requirements, which makes it challenging to play a good role in the face of long sequence models^[22]. Therefore, we introduce the Lambda layer in the LCSE model to model the internal relationships of context elements with a lower memory cost. On the other hand, these two models are insufficient in capturing short-term preference information. The DIN model does not pay much attention to the short-term preferences, while the STAMP model only emphasizes the importance of the latest click. If the last click was a mistake, the prediction results may be affected adversely. There is still room for optimization in the capture of short-term preferences.

2 The Lambda Layer-Based Convolutional Sequence Embedding Model

The LCSE model proposed in this paper handles user short-term preferences and general preferences separately. We use convolutional neural networks to capture user short-term preferences from union-level and point-level, use Lambda layer-based convolutional neural network to fuse the user's general preferences, and finally obtain the prediction result according to the captured long-term and short-term preference information. The structure is shown in Fig. 1. Therefore, the core module of the proposed LCSE mainly includes the following four parts: (1) Embedding service features through the embedding layer; (2) Capturing short-term interest features of users through convolutional neural network from union-level and point-level; (3) Capturing user's long-term interest characteristics through convolutional neural network based on Lambda layer from union-level and point-level; (4) Obtaining the final CTR prediction results through the fully connected layer. Next, we will discuss each part of the model in detail.

2.1 The Embedding Layer

The LCSE model predicts the probability that the user will click on the target service i_t next time mainly by the user's click sequence $I = \{i_{t-n}, i_{t-n+1}, i_{t-1}\}$ (in

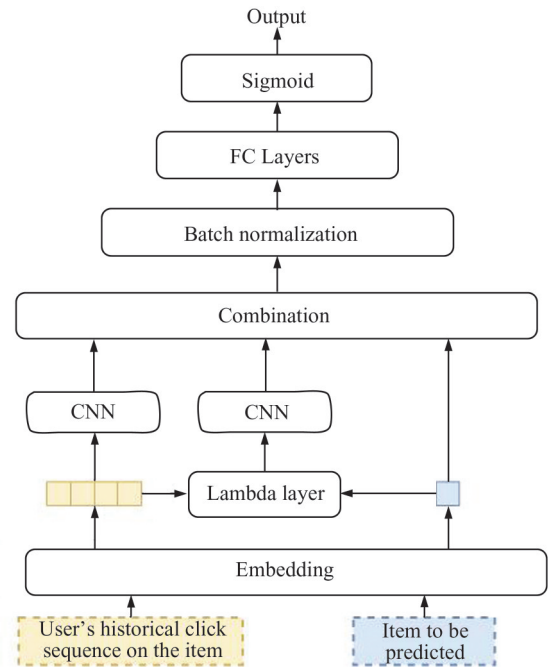


Fig. 1 Lambda layer-based Convolutional Sequence Embedding model's structure

which n represents the length of user's historical click sequence), so the user's click sequence $I = \{i_{t-n}, i_{t-n+1}, i_{t-1}\}$ and the target service i_t are imported into the model, in which the target service i_t contains k features.

The main function of the embedding layer is to encode the representation of attributes using vectors. Compared to one-hot encoding commonly used, vectors have lower dimensions after embedding and can get updated when training the neural network. We can assume that, through embedding, each service feature has a d dimension, and k features of the service i are connected to form the embedding vector e .

2.2 The Lambda Layer

The Lambda network^[22] is a linear attention mechanism proposed by Bello. Compared with the self-attention mechanism proposed by Bahdanau *et al.*^[23], it has lower time and space complexity. Compared with the recent linear attention mechanisms proposed by Katharopoulos *et al.*^[24] and Choromanski *et al.*^[25], it can model inner relationships between elements within data. Lambda has achieved good performance in areas such as image classification, proving its efficient handling of massive data. In an e-commerce scenario with the same huge amount of data, in the face of a large number of historical click sequences for each user, the Lambda layer can also give full play to its advantages in CTR predic-

tion. In the LCSE model, the Lambda layer is placed above the embedding layer, and better results are obtained by fusing the embedding layer information. Here we make a little modification to the Lambda layer proposed by Bello, and the modified content is mainly reflected in the use of location information.

Similar to other attention mechanisms, the Lambda layer's query matrix \mathbf{Q} , keyspace matrix \mathbf{K} and valuespace matrix can be obtained by linear mapping from the input X and the context C , respectively (equation (1)). Here we use the user's click sequence after embedding both as the input X and the context C , and add batch normalization operations after linear mapping to improve the training effect of the model.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{soft max}(\mathbf{Q}\mathbf{K}^T)\mathbf{V}$$

$$(\mathbf{Q} \in \mathbb{R}^{n \times d_q}, \mathbf{K} \in \mathbb{R}^{n \times d_k}, \mathbf{V} \in \mathbb{R}^{n \times d_v}) \quad (1)$$

$$\mu_m^c = \bar{\mathbf{K}}_m \mathbf{V}_m^T \quad (2)$$

$$\mu_m^p = \mathbf{E}_m \mathbf{V}_m^T \quad (3)$$

$$\bar{\mathbf{K}} = \text{soft max}(\mathbf{K}) \quad (4)$$

$$E_{i,m} = \begin{cases} \sin\left(\frac{i}{1000^{\frac{i}{n}}}\right) \times \mathbf{K}(i\%2 == 0) \\ \cos\left(\frac{i}{1000^{\frac{i}{n}}}\right) \times \mathbf{K}(i\%2 == 1) \end{cases} \quad (5)$$

The subsequent operations differ from other attention mechanisms in that the contribution of each context element m consists of two parts: content-based contribution μ_m^c (equation (2)) and location-based contribution μ_m^p (equation (3)), in which the content information is obtained by standardization of \mathbf{K} using the function soft max (equation (4)), and location information is obtained by encoding using the sine and cosine functions (equation (5)). The Lambda function is the sum of the above two parts (equation (6)).

$$\lambda = \lambda^c + \lambda^p = \sum_m \mu_m^c + \sum_m \mu_m^p \quad (6)$$

Finally, we apply Lambda to the query matrix \mathbf{Q} to get the output \mathbf{O} of the Lambda layer (equation (7)). The time complexity of the entire Lambda layer is $O(n)$.

$$\mathbf{O} = \lambda \mathbf{Q} = (\lambda^c + \lambda^p) \mathbf{Q} \quad (7)$$

$$\begin{cases} c_i^k = \phi(E_{i+i-h-1} \odot F_h^k) \\ c^k = [c_1^k, c_2^k, \dots, c_{L-h+1}^k] \end{cases} \quad (8)$$

$$f_H = \{\max(c^1), \max(c^2), \dots, \max(c^n)\} \quad (9)$$

In the process of implementing the Lambda layer, we also change it to the form of a multi-head attention

mechanism to enhance the learning ability of the model by splitting and merging. The multi-head attention mechanism learns features from H subspaces separately, and then splices the learning results of the H subspaces together to obtain better learning results, yet the number of the H subspaces is not the higher the better.

2.3 The Neural Convolutional Network Layer

The structure shown in Fig. 2 includes two convolutional neural network (CNN) layers^[26], which are good at processing sequential information. The data sequence can directly enter the CNN layer after being processed by the embedding layer. The Lambda layer can further process the data sequence before entering the CNN layer. The former CNN layer is used to capture short-term user preferences and the latter is used to capture long-term user preferences. Both of the layers capture the union-level and point-level information of the input sequence for feature extraction.

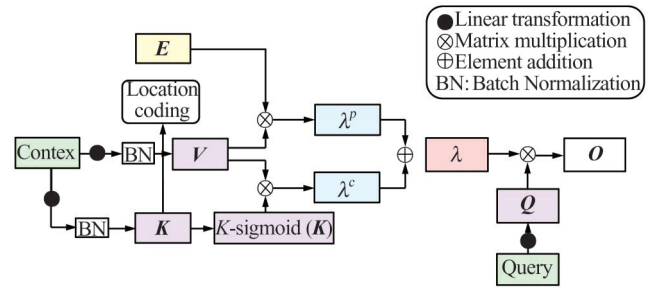


Fig. 2 Lambda Layer's structure

Figure 3 shows the structure of the convolution layer. The user click sequence, whether it is processed by the embedding layer or by the attention mechanism, is always a $\mathbb{R}^{L \times d}$ (where L is the length of the user's historical click sequence, and d is the dimension of user service after embedding) matrix \mathbf{E} . Here, the matrix \mathbf{E} can be convolved in a similar way to image processing. Dif-

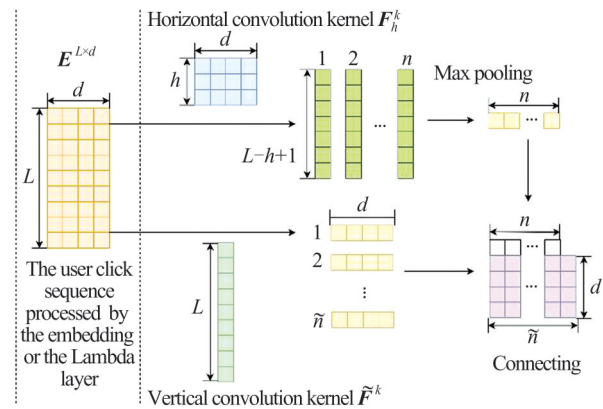


Fig. 3 Convolutional Layer's structure

ferent sequence information can be extracted by horizontal filter and vertical filter.

The horizontal filter can capture union-level information. It is mainly used to aggregate the joint features of multiple services from the embedding dimension, so that the historical information left by the user recently on the e-commerce platform can jointly decide the next prediction result. $F_h^k \in \mathbb{R}^{L \times d}$ is one of the n horizontal filters and it is a $h \times d$ matrix ($1 \leq k \leq n$), in which d is the dimension of user service after embedding. The convolution value c^k obtained after the k -th horizontal filter F_h^k is slid over the matrix E for i ($1 \leq i \leq L-h+1$) times is shown in equation (8), where \odot represents inner product operation and $\phi(\cdot)$ is the activation function. After the convolution operation, the most important feature f_H (equation (9)) extracted by the horizontal filter is obtained by the max pooling operation.

The vertical filter can capture point-level information mainly by weighted sum. It splits the user's multiple click history into multiple single events, which affect the prediction results separately. Suppose there are \tilde{n} vertical filters, each $\tilde{F}^k \in \mathbb{R}^{L \times 1}$ ($1 \leq k \leq \tilde{n}$) and L is the length of the user's historical click sequence. The convolution value \tilde{c}^k obtained by sliding the vertical filter \tilde{F}^k on E for d times is shown in equation (10). l represents the row corresponding to the matrix. Finally, $\tilde{c}^1, \tilde{c}^2, \dots, \tilde{c}^{\tilde{n}}$ are combined and output as a feature vector f_V of size $\tilde{n} \times d$, as shown in equation (11).

$$\tilde{c}^k = \sum_{l=1}^L \tilde{F}_l^k \cdot E_l \quad (10)$$

$$f_V = [\tilde{c}^1, \tilde{c}^2, \dots, \tilde{c}^{\tilde{n}}] \quad (11)$$

In the next step, we splice the feature f_H extracted by the horizontal filter and the feature f_V extracted by the vertical filter to get the output of the convolutional neural network layer.

2.4 The Fully Connected Network Layer

The input of the fully connected layer is the vector $v = [s, c, e_i]$ obtained by jointing the output vector s of the convolutional neural network that captures the user's short-term interest features, the output vector c of the attention-based convolutional neural network that captures the user's long-term interest features, and the feature vector e_i of the target service. The vector v needs to be processed by the Batch Normalization (BN) layer before being imported into the fully connected layer. After that it can be sent into the fully connected neural network. BN aims to alleviate the problem of gradient dis-

appearance during training by adjusting the data distribution, and to accelerate the speed of network convergence.

We use PReLU as the activation function (equation (12)) in the fully connected layer, and use Sigmoid as the activation function (equation (13)) in the output layer after the fully connected layer to get the predicted probability \hat{y} of a user clicking on the target service.

$$\text{PReLU} = \begin{cases} x & (x > 0) \\ ax & (x \leq 0) \end{cases} \quad (12)$$

$$\hat{y} = \text{sigmoid}(v) = \frac{1}{1 + e^{-v}} \quad (13)$$

Finally, we use the cross entropy loss function to calculate the loss of the model to complete the learning of model parameters, as shown in equation (14), where y represents whether the user clicks on the target service and Y represents the sample set.

$$\text{loss} = -\frac{1}{|Y|} \sum_{y \in Y} [y \log \hat{y} + (1-y) \log (1-\hat{y})] \quad (14)$$

3 Experiment and Analysis

3.1 Datasets

The public datasets are used in the experiments of this paper. In order to better simulate the actual e-commerce scenario, we selected two datasets from the Amazon dataset^[27] and one from the IMDB dataset^[28].

1) The Amazon dataset mainly records the service and user's comment information on the Amazon e-commerce website. It contains product information such as books and electronics. In this experiment, we use Movie & TV and Electronic these two categories as our datasets. Meanwhile, we regard the user's comment on the product as a click behavior, and randomly add items that have not been commented as negative samples of the user's click event. The ratio of positive and negative samples is 1 : 1.

2) The movie rating dataset mainly records the movie information in the IMDB and the user's movie rating information. In the experiments, we choose MovieLens-1M as the experimental dataset. Similarly, we take the user's rating of one movie as a click behavior, and randomly add movies the current user has not rated as the negative samples of the user's click event. The ratio of positive and negative samples is 1 : 1 as well.

Among many network services, movie and electronic product preferences are especially determined by personal taste, and user interest characteristics are more

significant in these scenarios, which is convenient for the model to learn users' diverse personality preferences^[28]. Therefore, this paper conducts experiments in the application scenarios corresponding to these three datasets, in order to achieve the purpose of giving full play to the role of the model and testing the accuracy. The basic information of the three datasets is shown in Table 1.

Table 1 Summary of the three datasets

Dataset	Number of users	Number of services	Number of service categories
Movie & TV	123 960	50 052	29
Electronic	192 403	23 001	801
MovieLens	610	972	19

3.2 Evaluation Criteria

Here, we use Area Under the ROC Curve (AUC) and logloss as our evaluation indicators. AUC reflects the probability that the model judges that the positive sample value is higher than the negative sample, when a positive and negative sample are randomly selected. Models with an AUC value closer to 1.0 perform better in prediction. Logloss can calculate how close the model's predictions are to the target results. The closer the logloss is to 0, the more accurate the model's predictions are.

3.3 Comparative Experiment

This experiment mainly compares the LCSE model with the following models:

1) Attentional Factorization Machine (AFM) model^[29]: The AFM model learns the importance of second-order combined features by introducing an attention mechanism.

2) Wide & Deep Learning (WDL) model^[15]: The WDL model captures features through a machine learning-based Wide model and a deep learning-based Deep model, and then performs joint training to obtain prediction results.

3) Deep & Cross Network (DCN) model^[16]: The DCN model mainly combines feature crossover through the Cross network with the results of a fully connected feed-forward neural network to obtain the final prediction probability.

4) Automatic Feature Interaction Learning via Self-Attentive Neural Networks (AutoInt) model^[30]: The AutoInt model learns the high-order feature interactions through multi-head self-attentive neural networks.

5) eXtreme Deep Factorization Machine (xDeepFM) model^[18]: The xDeepFM model improves the memory and generalization ability of the prediction model by compressing the interaction network, which makes the feature interactions occur at the vector level.

6) Deep Interest Network (DIN) model^[20]: The DIN model uses the attention mechanism to discover the user's interest preferences in the click sequence to predict the click rate.

7) Convolutional Sequence Embedding Recommendation (Caser) model^[12]: The Caser model extracts users' short-term preferences through convolutional neural networks for CTR prediction.

The first five models among the comparison models are all based on feature interaction learning. In comparison, the last two comparison models and the LCSE model proposed in this paper are based on sequence modeling. By introducing a variety of feature learning models and comparing them with sequence modeling, we can judge the importance of feature learning and sequence modeling, so as to confirm the superiority of the LCSE model based on sequence modeling more comprehensively. On this basis, the LCSE is compared with the DIN model, which reflects the impact of learning short-term user preferences on the model performance. Comparing LCSE with Caser, we examine the effect of general preferences on model performance as well. Compared with the two, LCSE adds the extraction of union-level information to the convolution layer, optimizes the attention mechanism, and adopts an improved Lambda layer. The validity of the overall design of the LCSE model will be tested in the comparative experiment section below, and the effects of each component of the LCSE model will be verified in the ablation experiment section.

In terms of the parameter setting, AFM, WDL, DCN, AutoInt, xDeepFM, DIN, Caser and LCSE use the same number of layers and dimensions of fully connected layers, of which the number of layers is 3, the dimensions are 256, 128, and 64, respectively. In terms of the optimizer, each model uses the Adam optimizer, where the learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

The results of the experiments on three public datasets are shown in Table 2. The performance of the three sequence modeling models is better than that of the feature learning models, indicating that sequence modeling performs better in these experimental scenarios. Regarding the evaluation index AUC, LCSE has higher values than other comparison models, which shows the superi-

ority of our model’s overall design.

Table 2 Comparison results of the proposed LCSE with other models

Model	Movie & TV		Electronic		MovieLens	
	AUC	logloss	AUC	logloss	AUC	logloss
AFM	0.835 7	0.552 9	0.796 2	0.626 1	0.800 1	0.499 1
WDL	0.833 6	0.542 7	0.803 8	0.616 9	0.832 4	0.467 2
DCN	0.828 7	0.539 7	0.775 2	0.598 8	0.881 3	0.451 3
xDeepFM	0.828 9	0.543 4	0.775 9	0.592 7	0.876 6	0.453 2
AutoInt	0.827 5	0.540 9	0.771 5	0.595 3	0.881 6	0.455 0
DIN	0.901 2	0.518 3	0.866 1	0.555 9	0.927 6	0.409 4
Caser	0.900 3	0.510 4	0.866 8	0.561 8	0.946 0	0.350 2
LCSE	0.903 6	0.505 4	0.870 7	0.545 5	0.946 7	0.341 0

3.4 Ablation Study

Ablation experiments evaluate the effectiveness of the model by modifying or removing parts of the model’s structure. The ablation experiment designed in this paper mainly includes three parts: (1) Unlike other convolutional neural networks based on attention mechanism, the LCSE model captures features from union-level and point-level when using convolutional neural networks, so it needs to demonstrate the effectiveness of the structure of horizontal convolution kernels; (2) the LCSE model uses convolutional neural networks to capture users’ short-term preferences and a Lambda layer-based convolutional neural network to capture users’ general preferences, thus it is necessary to prove the influence of the two kinds of preference information on the prediction results respectively; (3) The LCSE model uses the Lambda layer instead of the commonly used scaling dot-product attention mechanism, so the superiority of the

Lambda layer needs to be proved.

In order to demonstrate the effectiveness of the union-level information captured by the horizontal convolution kernel, in this experiment, the LCSE model with two horizontal convolution kernels is compared with the LCSE model with two convolution kernels removed (Model A) and the LCSE model with only the horizontal convolution kernel above the Lambda layer removed (Model B). The experimental data is based on model A. From Table 3, we can see that from model A to model B, the capture of union-level information in the user click sequence through the embedding layer is added. From model B to model LCSE, the capture of union-level information in data passing through the Lambda layer is added. With the gradual addition of two horizontal convolution kernels, both the AUC value and the cross-entropy are significantly improved, proving the effectiveness of the two kernels. The experimental results also confirm that the introduction of horizontal convolution kernels to capture union-level information has a positive effect on the overall performance of the model.

In order to prove the impact of short-term preferences and general preferences on CTR prediction results and the superiority of the Lambda layer, we designed the following three comparative models. Figure 4 shows the Caser model, Lambda+CNN model, ACSE model and LCSE model.

1) Caser model (Fig. 4(a)): a CTR prediction model that only focuses on capturing the short-term interests of users through convolutional neural networks. This model is the benchmark model for comparison in this ablation experiment.

2) Lambda+CNN model: a model that captures user interest only through the convolutional neural network on the improved Lambda layer (Fig. 4(b)), focusing on capturing user general preferences.

Table 3 Ablation experiment results about horizontal convolution kernel

Evaluation index	Model	Movie & TV		Electronic		MovieLens	
		Result	Relative increase/%	Result	Relative increase/%	Result	Relative increase/%
AUC	Model A	0.883 7	—	0.856 6	—	0.924 6	—
	Model B	0.901 9	2.05	0.869 9	1.55	0.946 5	2.17
	LCSE	0.903 6	2.25	0.870 7	1.65	0.946 7	2.39
logloss	Model A	0.534 5	—	0.569 1	—	0.415 8	—
	Model B	0.509 6	4.66	0.554 8	2.51	0.348 7	16.13
	LCSE	0.505 4	5.44	0.545 5	4.14	0.341 0	17.99

3) Attention-based Convolutional Sequence Embedding (ACSE) model: Unlike the LCSE model, the ACSE model uses the more common scaling dot product attention mechanism instead of the Lambda layer (Fig. 4(c)),

with both short-term and long-term preferences captured.

4) Lambda layer-based Convolutional Sequence Embedding (LCSE) model (Fig. 4(d)): our proposed model.

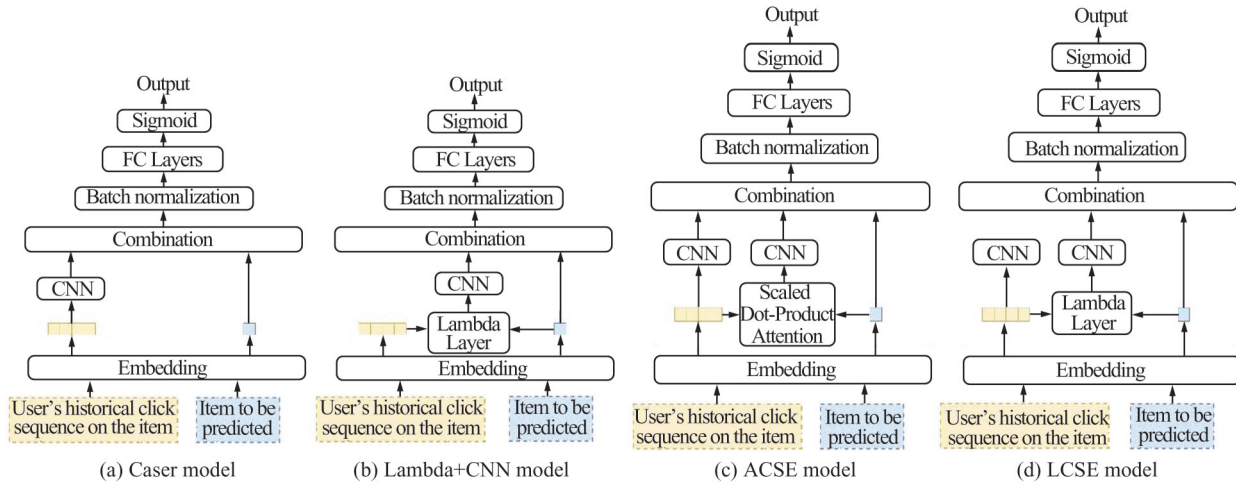


Fig. 4 Contrastive models for ablation experiments

The results of the contrastive experiments are shown in Table 4, and we can draw the following conclusions:

1) The LCSE model performs better than the Caser model and the Lambda+CNN model on the three datasets. The ACSE model is also a model that combines short-term and general preferences on the basis of the Caser model, and it also outperforms the Caser model, confirming the effectiveness of combining short-term and general preferences.

2) Compared with the ACSE model, the LCSE

model has better performance on the selected three datasets, which proves the superiority of the Lambda layer in the performance of the prediction results.

In addition, we also compared the training time of LCSE and ACSE by setting different embedding layer dimensions and maximum user history sequence lengths in the following configurations: Windows 10; Tensorflow2.1.0; AMD 3600 CPU; GTX 1080Ti GPU. The results in Fig. 5 show that LCSE's training time is shorter, which proves that the LCSE model also has higher training efficiency.

Table 4 Contrastive experiment results

Evaluation index	Model	Movie & TV		Electronic		MovieLens	
		Result	Relative increase/%	Result	Relative increase/%	Result	Relative increase/%
AUC	Caser	0.900 3	—	0.866 8	—	0.946 0	—
	Lambda+CNN	0.900 0	-0.03	0.867 8	0.12	0.944 8	-0.13
	ACSE	0.902 8	0.28	0.868 7	0.22	0.946 2	0.02
	LCSE	0.903 6	0.36	0.870 7	0.45	0.946 7	0.07
logloss	Caser	0.510 4	—	0.561 8	—	0.350 2	—
	Lambda+CNN	0.507 8	0.51	0.555 5	1.12	0.353 6	-0.97
	ACSE	0.506 3	0.80	0.554 8	1.25	0.348 1	0.60
	LCSE	0.505 4	1.04	0.545 5	2.90	0.341 0	2.63

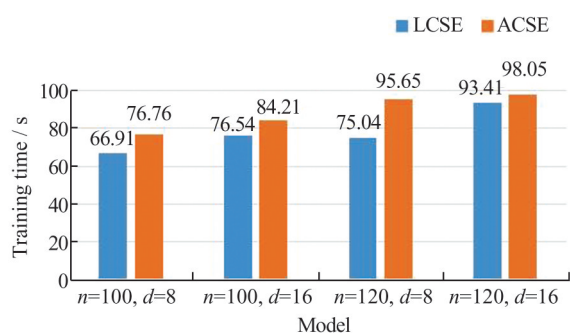


Fig. 5 Comparison of ACSE and LCSE on training time

4 Conclusion

With the continuous development of the Internet, predicting the user's next behavior through click rates and implementing precise delivery of products or advertisements are the embodiments of economic intelligence. In the fields of computational advertising and recommendation systems, e-commerce platforms hope to master users' short-term and long-term preferences, so as to push messages through customized business strategies.

Aiming to address the problem of ignoring the capture of general preferences, this paper proposes a CTR prediction model based on Lambda layer convolutional sequence embedding. The LCSE model uses a convolutional neural network and a Lambda layer-based convolutional neural network to capture the user's short-term and general preferences respectively, and finally predicts the user's next click behavior by combining these two types of preferences. The Lambda layer in the LCSE model is a type of attention mechanism. Compared with the common attention mechanism, the Lambda layer has the advantage of lower time complexity and better performance in predicting results.

Further optimization can be carried out in future work to capture both short-term and general preferences. For short-term preferences, there is still room for development in the identification of false clicks and outlier removal. For long-term preferences, we can continue to combine the related research on the linear attention mechanism, and further optimize on the basis of the Lambda layer to improve the overall performance of the model from the perspective of computational efficiency and prediction accuracy. Not limited to general preferences and short-term preferences, more detailed learning of user interest drift over time is also a future research direction, which can promote the customized development of network services and better demonstrate big

data intelligence.

Acknowledgements:

The authors are grateful to the College of Computer Science, Nanjing University of Posts and Telecommunications for kindly providing the required computational resources.

References

- [1] Quijano-Sánchez L, Cantador I, Cortés-Cediel M E, *et al.* Recommender systems for smart cities[J]. *Information Systems*, 2020, **92**: 101545.
- [2] Scholz M, Dorner V, Schryen G, *et al.* A configuration-based recommender system for supporting e-commerce decisions [J]. *European Journal of Operational Research*, 2017, **259** (1): 205-215.
- [3] Shoja B M, Tabrizi N. Customer reviews analysis with deep neural networks for e-commerce recommender systems[J]. *IEEE Access*, 2019, **7**: 119121-119130.
- [4] Zhou M Z, Ding Z Y, Tang J L, *et al.* Micro behaviors: A new perspective in e-commerce recommender systems[C]// *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. New York: ACM, 2018: 727-735.
- [5] Najafi-Asadolahi S, Fridgeirsdottir K. Cost-per-click pricing for display advertising[J]. *Manufacturing & Service Operations Management*, 2014, **16**(4): 482-497.
- [6] Fain D C, Pedersen J O. Sponsored search: A brief history [J]. *Bulletin of the American Society for Information Science and Technology*, 2006, **32**(2): 12.
- [7] He X, Pan W, Cheng H. Research on advertising click-through rate prediction model based on ensemble learning [C]// *Recent Advances in Data Science: Third International Conference on Data Science, Medicine, and Bioinformatics, IDMB 2019*. Singapore: Springer-Verlag, 2020: 82-93.
- [8] Chen M, Liu P. Performance evaluation of recommender systems[J]. *International Journal of Performability Engineering*, 2017, **13**(8): 1246.
- [9] Cheng C, Yang H Q, Lyu M R, *et al.* Where you like to go next: Successive point-of-interest recommendation[C]// *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. New York: ACM, 2013: 2605-2611.
- [10] He R N, McAuley J. Fusing similarity models with Markov chains for sparse sequential recommendation[C]// *2016 IEEE 16th International Conference on Data Mining (ICDM)*.

- New York: IEEE, 2016: 191-200.
- [11] Wang P F, Guo J F, Lan Y Y, *et al.* Learning hierarchical representation model for NextBasket recommendation[C]//*Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 2015: 403-412.
- [12] Tang J X, Wang K. Personalized top-N sequential recommendation via convolutional sequence embedding[C]//*Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. New York: ACM, 2018: 565-573.
- [13] Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need[C]//*Proceedings of the 31st International Conference on Neural Information Processing Systems*. New York: ACM, 2017: 6000-6010.
- [14] Feng Y F, Lv F Y, Shen W C, *et al.* Deep session interest network for click-through rate prediction[EB/OL]. [2023-06-21]. <http://arxiv.org/abs/1905.06482>.
- [15] Cheng H T, Koc L, Harmsen J, *et al.* Wide & deep learning for recommender systems[C]//*Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. New York: ACM, 2016: 7-10.
- [16] Wang R X, Fu B, Fu G, *et al.* Deep & cross network for ad click predictions[C]//*Proceedings of the AdKDD and TargetAd*. Halifax: NS, 2017: 1-7.
- [17] Guo H F, Tang R M, Ye Y M, *et al.* DeepFM: A factorization-machine based neural network for CTR prediction[EB/OL]. [2023-06-21]. <http://arxiv.org/abs/1703.04247>.
- [18] Lian J X, Zhou X H, Zhang F Z, *et al.* xDeepFM: Combining explicit and implicit feature interactions for recommender systems[C]//*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York: ACM, 2018: 1754-1763.
- [19] Yu F, Liu Q, Wu S, *et al.* A dynamic recurrent model for next basket recommendation[C]//*Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 2016: 729-732.
- [20] Zhou G R, Zhu X Q, Song C R, *et al.* Deep interest network for click-through rate prediction[C]//*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York: ACM, 2018: 1059-1068.
- [21] Liu Q, Zeng Y F, Mokhosi R, *et al.* STAMP: Short-term attention/memory priority model for session-based recommendation[C]//*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York: ACM, 2018: 1831-1839.
- [22] Bello I. Lambdanetworks: Modeling long-range interactions without attention[EB/OL]. [2023-06-21]. <http://arxiv.org/abs/2102.08602>.
- [23] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB/OL]. [2023-06-21]. <http://arxiv.org/abs/1409.0473>.
- [24] Katharopoulos A, Vyas A, Pappas N, *et al.* Transformers are RNNs: Fast autoregressive transformers with linear attention [C]//*Proceedings of the 37th International Conference on Machine Learning*. New York: ACM, 2020: 5156-5165.
- [25] Choromanski K, Likhoshesterov V, Dohan D, *et al.* Rethinking attention with performers[EB/OL]. [2023-06-21]. <http://arxiv.org/abs/2009.14794>.
- [26] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. *Advances in Neural Information Processing Systems*, 2012, **25**(2): 84-90.
- [27] Haque T U, Saber N N, Shah F M. Sentiment analysis on large scale Amazon product reviews[C]//*2018 IEEE International Conference on Innovative Research and Development (ICIRD)*. New York: IEEE, 2018: 1-6.
- [28] Harper F M, Konstan J A. The movielens datasets: History and context[J]. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2015, **5**(4): 1-19.
- [29] Xiao J, Ye H, He X N, *et al.* Attentional factorization machines: Learning the weight of feature interactions via attention networks[EB/OL]. [2023-06-21]. <http://arxiv.org/abs/1708.04617>.
- [30] Song W P, Shi C C, Xiao Z P, *et al.* AutoInt: Automatic feature interaction learning via self-attentive neural networks [C]//*Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. New York: ACM, 2019: 1161-1170.

□