Deep Session Interest Network for Click-Through Rate Prediction

Yufei Feng[1] , **Fuyu Lv**[1] , **Weichen Shen**[1] , **Menghan Wang**[1,2] , **Fei Sun**[1] , **Yu Zhu**[1] and **Keping Yang**[1]

[1]Alibaba Group, Hangzhou, China
[2]Zhejiang University, Hangzhou, China
fyf649435349@gmail.com, {fuyu.lfy, weichen.swc, xiangyu.wmh, ofey.sf, zy143829}@alibaba-inc.com, shaoyao@taobao.com

Abstract

Click-Through Rate (CTR) prediction plays an important role in many industrial applications, such as online advertising and recommender systems. How to capture users' dynamic and evolving interests from their behavior sequences remains a continuous research topic in the CTR prediction. However, most existing studies overlook the intrinsic structure of the sequences: the sequences are composed of sessions, where sessions are user behaviors separated by their occurring time. We observe that user behaviors are highly homogeneous in each session, and heterogeneous cross sessions. Based on this observation, we propose a novel CTR model named Deep Session Interest Network (DSIN) that leverages users' multiple historical sessions in their behavior sequences. We first use self-attention mechanism with bias encoding to extract users' interests in each session. Then we apply Bi-LSTM to model how users' interests evolve and interact among sessions. Finally, we employ the local activation unit to adaptively learn the influences of various session interests on the target item. Experiments are conducted on both advertising and production recommender datasets and DSIN outperforms other state-of-the-art models on both datasets.

1 Introduction

Recommender systems (RS) are becoming increasingly indispensable in assisting users to find their preferred items in web-scale applications such as Amazon and Taobao. Typically, an industrial recommender system consists of two stages: candidate generation and candidate ranking [Covington *et al.*, 2016]. The candidate generation stage adopts some naive but time-efficient recommendation algorithms (e.g. item-based collaborative filtering [Sarwar *et al.*, 2001]) to provide a relative small set of items from the huge whole item set for ranking. In the candidate ranking stage, complex but powerful models (e.g. neural network methods) are applied to rank the candidates so as to select the top-k items for recommendation. In this paper, we mainly focus on the candidate ranking stage and treat it as a Click-Through Rate (CTR) prediction task. It means we assume a relative small item set has been provided for ranking and we rank items according to their CTR score predictions.

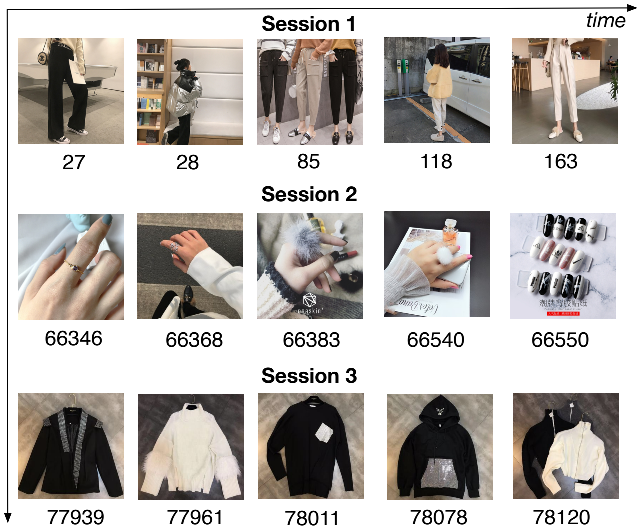

Figure 1: Here is one demo of sessions collected from a real-world industrial application. The number beneath the picture indicates the time gap, specified in seconds, between the time of clicking on the current item and that of clicking on the first item. Sessions are divided in the principle of whenever there exists a time gap of more than 30 minutes.

Some recent effective CTR models [Covington *et al.*, 2016; Zhou *et al.*, 2018c; Zhou *et al.*, 2018b; Zhou *et al.*, 2018a] show promising results by utilizing users' sequential behaviors, which reflect users' dynamic and evolving interests. However, these models overlook the intrinsic structure of the sequences: the sequences are composed of sessions. A session is a list of interactions (user behaviors) that occur within a given time frame. We observe that user behaviors are highly homogeneous in each session and heterogeneous cross sessions. As shown in figure 1, a user is sampled from a real-world industrial application and we split her behavior sequence into 3 sessions. Sessions are divided in the principle of whenever there exists a time gap of more than 30 minutes [Grbovic and Cheng, 2018]. The user mainly browses trousers in session 1, finger rings in session 2, and coats in session 3. The phenomenon illustrated in figure 1 is general. It reflects the fact that a user usually has a clear unique intention in one session while his/her interest can change sharply when he/she starts a new session.

Motivated by the above observations, we propose Deep Session Interest Network[1] (DSIN) to model users' sequential behaviors in the CTR prediction task by leveraging their multiple historical sessions. There are three key components in DSIN. First, we naturally divide users' sequential behaviors into sessions and then use self-attention network with bias encoding to model each session. Self-attention can capture the inner interaction/correlation of session behaviors and then extract users' interests of each session. These various session interests may be correlated with each other and even follow a sequential pattern [Quadrana *et al.*, 2017]. So in the second part, we apply bi-directional LSTM (Bi-LSTM) to capture the interaction and evolution of users' varying historical session interests. Because various session interests have different influences on the target item, finally we design the local activation unit to aggregate them w.r.t. the target item to form the final representation of the behavior sequence.

The main contributions of this paper are summarized as follows:

- We highlight that user behaviors are highly homogeneous in each session and heterogeneous cross sessions, and propose a new model named DSIN, which can effectively model the user's multiple sessions for the CTR prediction.
- We design a self-attention network with bias encoding to get accurate interest representation of each session. Then we employ Bi-LSTM to capture the sequential relationship among historical sessions. At last, we employ the local activation unit for aggregation considering the influences of different session interests on the target item.
- Two groups of comparative experiments are conducted on both advertising and production recommender datasets. The experiment results demonstrate the superiority of our proposed DSIN compared with other state-of-the-art models in the CTR prediction task.

The organization of the remaining parts of this paper is as follows. Section 2 introduces some related work. Section 3 gives the detailed description of our DSIN model. Section 4 presents our experiment results and analyses on both advertising and recommender datasets.

2 Related Work

In this section, we mainly introduce existing studies of the CTR prediction and session-based recommendation.

2.1 Click-Through Rate Prediction

Recent CTR models mainly pay attention to the interaction between features. Wide&Deep [Cheng *et al.*, 2016] combines the linear representation of features. DeepFM [Guo *et al.*, 2017] learns the second-order crossover of features and DCN [Wang *et al.*, 2017] applies a multi-layer residual structure to learn higher-order representation of features. AFM [Xiao *et al.*, 2017] argues that not all feature interactions are equally predictive and uses attention mechanism to automatically learn weights of cross-features. To sum up, the higher-order representation and interaction of features significantly improve the expressive ability of features and the generalization ability of models.

Users' sequential behaviors imply users' dynamic and evolving interests and have been widely proven effective in the CTR prediction task. YoutubeNet [Covington *et al.*, 2016] transforms embeddings of users' watching lists into a vector of fixed length by average pooling. Deep Interest Network (DIN) [Zhou *et al.*, 2018c] uses attention mechanism to learn the representation of users' historical behaviors w.r.t. the target item. ATRANK [Zhou *et al.*, 2018a] proposes an attention-based framework modeling the influence between users' heterogeneous behaviors. Deep Interest Evolution Network (DIEN) [Zhou *et al.*, 2018b] uses auxiliary loss to adjust the expression of current behavior to the next behavior and then models the specific interest evolving process for different target items with AUGRU. Modeling users' sequential behaviors enriches the representation of the user and improves the prediction accuracy significantly.

2.2 Session-based Recommendation

The concept of session is commonly mentioned in sequential recommendation but rare in the CTR prediction task. Session-based recommendation benefits from the dynamic evolving of users' interests in sessions. General Factorization Framework (GFF) [Hidasi and Tikk, 2016] uses sum pooling of items to represent a session. Each item has two kinds of representations, one represents itself and the other represents the context of the session. Recently, RNN-based approaches [Hidasi *et al.*, 2015; Hidasi *et al.*, 2016; Li *et al.*, 2018] are applied into session-based recommendations to capture the order relationship within a session. Based on that, [Li *et al.*, 2017] proposes a novel attentive neural networks framework (NARM) to model the user's sequential behavior and capture the user's main purpose in the current session. Hierarchical RNN [Quadrana *et al.*, 2017] is proposed to relays end evolves latent hidden states of the RNNs across users' historical sessions. Besides RNNs, [Liu *et al.*, 2018; Kang and McAuley, 2018] apply only self-attention based models to effectively capture long-term and short-term interests of a session. [Tang and Wang, 2018] uses convolutional neural network and [Chen *et al.*, 2018] adopts user memory network to enhances the expressiveness of the sequential model.

3 Deep Session Interest Network

In this section, we introduce the Deep Session Interest Network (DSIN) in detail. We first introduce the basic deep CTR model named BaseModel, then the technical designs of DSIN that model the extraction and interaction of users' session interests.

3.1 BaseModel

In this section, we mainly introduce feature representation, embedding, MLP and loss function in BaseModel.

[1] https://github.com/shenweichen/DSIN

Feature Representation

Informative features count a great deal in the CTR prediction task. Overall, we use three groups of features in BaseModel: *User Profile*, *Item Profile* and *User Behavior*. Each group consists of some sparse features: *User Profile* contains *gender*, *city*, etc.; *Item Profile* contains *seller id*, *brand id*, etc.; *User Behavior* contains the *item ids* of items that the user recently clicked on. Note that the side information of the item can be concatenated to represent itself.

Embedding

Embedding is a common technique which transforms large-scale sparse features into low-dimensional dense vectors. Mathematically, sparse features can be represented by $\mathbf{E} \in \mathbb{R}^{M \times d_{model}}$ respectively, where M is the size of sparse features and d_{model} is the embedding size. With embedding, *User Profile* can be represented by $\mathbf{X}^U \in \mathbb{R}^{N_u \times d_{model}}$ where N_u is the number of sparse features of *User Profile*. *Item Profile* can be represented by $\mathbf{X}^I \in \mathbb{R}^{N_i \times d_{model}}$ where N_i is the number of sparse features of *Item Profile*. *User Behavior* can be represented by $\mathbf{S} = [\mathbf{b}_1; ...; \mathbf{b}_i; ...; \mathbf{b}_N] \in \mathbb{R}^{N \times d_{model}}$ where N is the number of users' historical behaviors and $\mathbf{b}_i$ is the embedding of the i-th behavior.

Multiple Layer Perceptron (MLP)

First, embeddings of sparse features from *User Profile*, *Item Profile* and *User Behavior* are concatenated, flattened and then fed into MLP with the activation function such as RELU. The softmax function is used at last to predict the probability of the user clicking on the target item.

Loss Function

The negative log-likelihood function is widely used in CTR models, which is usually defined as:

$$L = -\frac{1}{N} \sum_{(x,y) \in \mathbb{D}} (y \log p(x) + (1-y) \log(1-p(x))) \quad (1)$$

where $\mathbb{D}$ is the training dataset, x is represented by $[\mathbf{X}^U, \mathbf{X}^I, \mathbf{S}]$ is the input of the network, $y \in \{0, 1\}$ represents whether the user clicked the item and $p(\cdot)$ is the final output of the network which represents the prediction probability that the user clicks the item.

3.2 Model Overview

In recommnder systems, users' behavior sequences consist of multiple historical sessions. Users show varying interests in different sessions. Also, users' session interests are sequentially related to each other. DSIN is proposed to extract users' session interest in each session and capture the sequential relationship of session interests.

As shown in figure 2, DSIN consists of two parts before MLP. One is the embedding vectors transformed from *User Profile* and *Item Profile*. The other models *User Behavior* and has four layers from the bottom up: (1) session division layer partitions users' behavior sequence into sessions; (2) session interest extractor layer extracts users' session interests; (3) session interest interacting layer captures the sequential relationship among session interests; (4) session interest activating layer applies the local activation unit to users' session interests w.r.t the target item. Finally outputs of session interest activating layer and embedding vectors of *User Profile* and *Item Profile* are fed into MLP for the final prediction. In the following sections we introduce these four layers in the latter part in detail.

Session Division Layer

To extract more precise users' session interests, we divide users' behavior sequences $\mathbf{S}$ into sessions $\mathbf{Q}$, where the k-th session $\mathbf{Q}_k = [\mathbf{b}_1; ...; \mathbf{b}_i; ...; \mathbf{b}_T] \in \mathbb{R}^{T \times d_{model}}$, T is the number of behaviors we keep in the session and $\mathbf{b}_i$ is users' i-th behavior in the session. The segmentation of users' sessions exists between adjacent behaviors whose time interval is more than 30 minutes followed by [Grbovic and Cheng, 2018].

Session Interest Extractor Layer

Behaviors in the same session are strongly related to each other. Besides, users' casual behaviors in the session deviate the session interest from its original expression. To capture the inner relationship between behaviors in the same session and decrease the effect of those unrelated behaviors, we employ multi-head self-attention [Vaswani *et al.*, 2017] mechanism in each session. We also make some improvements in the self-attention mechanism to achieve our goal better.

Bias Encoding. To make use of the order relations of the sequence, self-attention mechanism applies positional encoding to the input embeddings. Furthermore, the order relations of sessions and the bias existed in different representation subspaces also need to be captured. Thus, we propose bias encoding $\mathbf{BE} \in \mathbb{R}^{K \times T \times d_{model}}$ on the basis of positional encoding, where each element in $\mathbf{BE}$ is defined as follows:

$$\mathbf{BE}_{(k,t,c)} = \mathbf{w}_k^K + \mathbf{w}_t^T + \mathbf{w}_c^C \quad (2)$$

where $\mathbf{w}^K \in \mathbb{R}^K$ is the bias vector of the session, k is the index of sessions, $\mathbf{w}^T \in \mathbb{R}^T$ is the bias vector of the position in the session, t is the index of the behavior in sessions, $\mathbf{w}^C \in \mathbb{R}^{d_{model}}$ is the bias vector of the unit position in the behavior embedding and c is the index of the unit in the behavior embedding. After added with bias encoding, users' behavior sessions $\mathbf{Q}$ are updated as follows:

$$\mathbf{Q} = \mathbf{Q} + \mathbf{BE} \quad (3)$$

Multi-head Self-attention. In recommender systems, users' click behaviors are influenced by various factors (e.g. colors, styles and price) [Zhou *et al.*, 2018a]. Mulit-head self-attention can capture relationship in different representation subspaces. Mathematically, let $\mathbf{Q}_k = [\mathbf{Q}_{k1}; ...; \mathbf{Q}_{kh}; ...; \mathbf{Q}_{kH}]$ where $\mathbf{Q}_{kh} \in \mathbb{R}^{T \times d_h}$ is the h-th head of $\mathbf{Q}_k$, H is the number of heads and $d_h = \frac{1}{h} d_{model}$. The output of $\mathbf{head}_h$ is calculated as follows:

$$\begin{aligned} \mathbf{head}_h &= \text{Attention}(\mathbf{Q}_{kh}\mathbf{W}^Q, \mathbf{Q}_{kh}\mathbf{W}^K, \mathbf{Q}_{kh}\mathbf{W}^V) \\ &= \text{softmax}(\frac{\mathbf{Q}_{kh}\mathbf{W}^Q {\mathbf{W}^K}^T \mathbf{Q}_{kh}^T}{\sqrt{d_{model}}})\mathbf{Q}_{kh}\mathbf{W}^V \end{aligned} \quad (4)$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^Q$ are linear matrices. Then vectors of different heads are concatenated and then fed into a feed-forward network:

$$\mathbf{I}_k^Q = \text{FFN}(\text{Concat}(\mathbf{head}_1, ..., \mathbf{head}_H)\mathbf{W}^O) \quad (5)$$

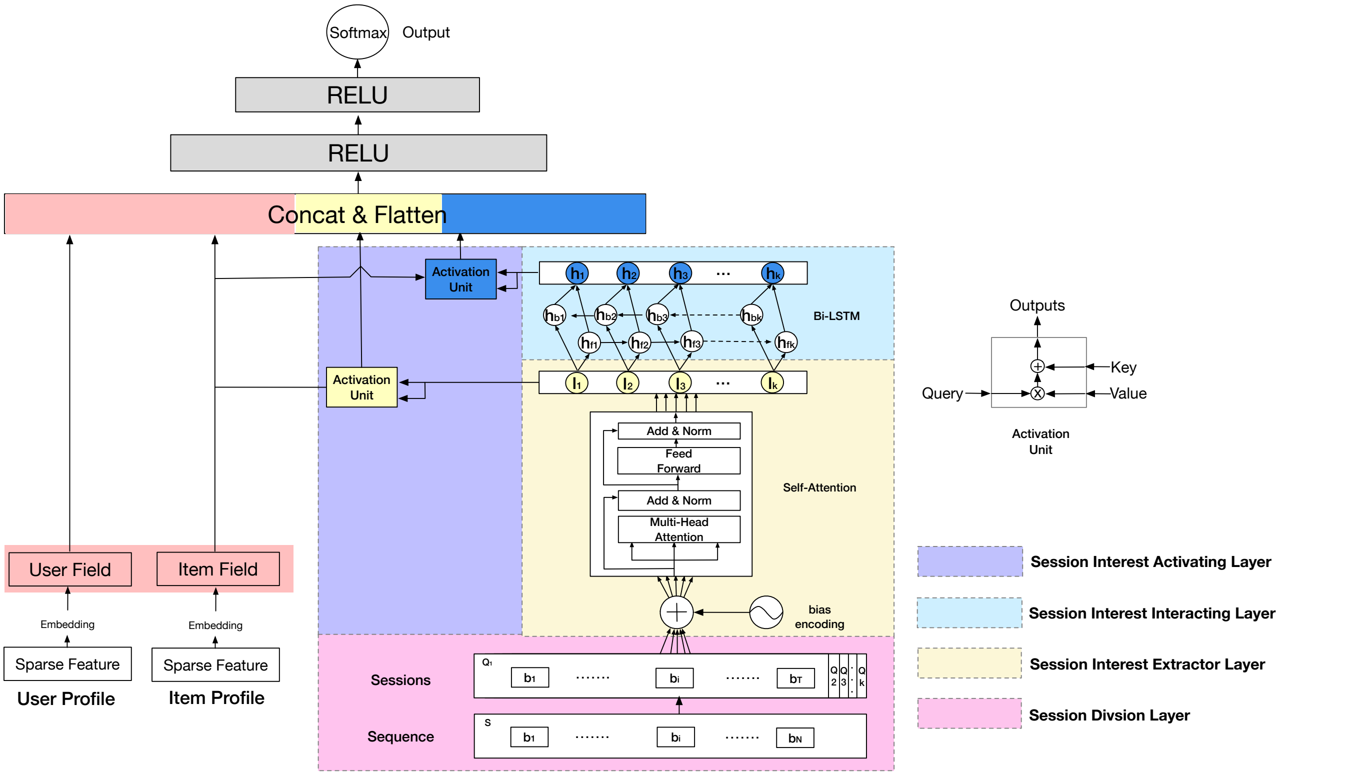

Figure 2: The overview of our proposed model DSIN. Overall, before MLP layers, DSIN has two main components. One is sparse features and the other processes the user behavior sequence. From the bottom up, the user behavior sequence **S** is first divided into sessions **Q**, which are then added with bias encoding and extracted into session interests **I** with self-attention. With Bi-LSTM, we mix session interests **I** with contextual information as hidden states **H**. Both Vectors of session interests **I** and hidden states **H** activated by the target item and embedding vectors of *User Profile* and *Item Profile* are concatenated, flattened and then fed into MLP layers for the final prediction.

where $\text{FFN}(\cdot)$ is the feed-forward network and $\mathbf{W}^O$ is the linear matrix. We also conduct residual connections and layer normalization successively. Users' k-th session interest $\mathbf{I}_k$ is calculated as follows:

$$\mathbf{I}_k = \text{Avg}(\mathbf{I}_k^Q) \tag{6}$$

where $\text{Avg}(\cdot)$ is the average pooling. Note that weights are shared in the self-attention mechanism of different sessions.

Session Interest Interacting Layer

Users' session interests hold sequential relations with contextual ones. Modeling the dynamic changes enriches the representation of the session interests. Bi-LSTM [Graves and Schmidhuber, 2005] is excellent at capturing sequential relations and naturally applied on modeling the interaction of session interests in DSIN. LSTM [Hochreiter and Schmidhuber, 1997] memory cell is implemented as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{I}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{I}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \\
\mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \tanh(\mathbf{W}_{xc}\mathbf{I}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{I}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \tanh(\mathbf{c}_t)
\end{aligned} \tag{7}$$

where $\sigma(\cdot)$ is the logistic function, and **i**, **f**, **o** and **c** are the input gate, forget gate, output gate and cell vectors which have the same size as $\mathbf{I}_t$. Shapes of weight matrices are indicated with the subscripts. Bi-direction means that there exist forward and backward RNNs, and the hidden states **H** are calculated as follows:

$$\mathbf{H}_t = \overrightarrow{\mathbf{h}_{ft}} \oplus \overleftarrow{\mathbf{h}_{bt}} \tag{8}$$

where $\overrightarrow{\mathbf{h}_{ft}}$ is the hidden state of the forward LSTM and $\overleftarrow{\mathbf{h}_{bt}}$ is the hidden state of the backward LSTM.

Session Interest Activating Layer

Users' session interests more related to the target item have greater impacts on whether the user will click the target item. The weights of users' session interests need to be reallocated w.r.t. the target item. Attention mechanism [Bahdanau *et al.*, 2014] conducts soft alignment between source and the target and has been proved effective as a weight allocation mechanism. The adaptive representation of session interests w.r.t. the target item is calculated as follows:

$$\begin{aligned}
a_k^I &= \frac{\exp(\mathbf{I}_k \mathbf{W}^I \mathbf{X}^I))}{\sum_k^K \exp(\mathbf{I}_k \mathbf{W}^I \mathbf{X}^I)} \\
\mathbf{U}^I &= \sum_k^K a_k^I \mathbf{I}_k
\end{aligned} \tag{9}$$

where $\mathbf{W}^I$ has the corresponding shape. Similarly, the adaptive representation of session interests mixed with contextual information w.r.t. the target item is calculated as follows:

$$a_k^H = \frac{\exp(\mathbf{H}_k \mathbf{W}^H \mathbf{X}^I))}{\sum_k^K \exp(\mathbf{H}_k \mathbf{W}^H \mathbf{X}^I)}$$
$$\mathbf{U}^H = \sum_k^K a_k^H \mathbf{H}_k \tag{10}$$

where $\mathbf{W}^H$ has the corresponding shape. Embedding vectors of $User\ Profile$ and $Item\ Profile$, $\mathbf{U}^I$ and $\mathbf{U}^H$ are concatenated, flattened and then fed into the MLP layer.

4 Experiments

In this section, we first introduce experiment datasets, competitors and evaluation metric. Then we compare our proposed DSIN with competitors and analyse the results. We further discuss the effectiveness of critical technical designs in DSIN empirically.

4.1 Datasets

Advertising Dataset

Advertising Dataset[2] is a public dataset released by Alimama, an online advertising platform in China. It contains 26 million records from ad display/click logs of 1 million users and 800 thousand ads in 8 days. Logs from 2017-05-06 to 2017-05-12 are for training and logs from 2017-05-13 are for testing. Users' recent 200 behaviors are also recorded in logs.

Recommender Dataset

To verify the effectiveness of DSIN in the real-world industrial applications, we conduct experiments on the recommender dataset of Alibaba. This dataset contains 6 billion display/click logs of 100 million users and 70 million items in 8 days. Logs from 2018-12-13 to 2018-12-19 are for training and logs from 2018-12-20 are for testing. Users' recent 200 behaviors are also recorded in logs.

4.2 Competitors

- YoutubeNet [Covington *et al.*, 2016] is a technically designed model which uses users' watching video sequence for video recommendation in Youtube. It treats users' historical behaviors equally and utilizes average pooling operation. We also experiment with YoutubeNet without $User\ Behavior$ to verify the effectiveness of historical behaviors.
- Wide&Deep [Cheng *et al.*, 2016] is a CTR model with both memorization and generalization. It contains two parts: wide model of memory and deep model of generalization.
- DIN [Zhou *et al.*, 2018c] fully exploits the relationship between users' historical behaviors and the target item. It uses attention mechanism to learn the representation of users' historical behaviors w.r.t. the target item.
- DIN-RNN has a similar structure as DIN, except that we use the hidden states of Bi-LSTM, which models users' historical behaviors and learns the contextual relationship.
- DIEN [Zhou *et al.*, 2018b] extracts latent temporal interests from user behaviors and models interests evolving process. Auxiliary loss makes hidden states more expressive to represent latent interests and AUGRU models the specific interest evolving processes for different target items.

4.3 Metrics

AUC (Area Under ROC Curve), which means the the probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example, reflects the ranking ability of the model. It is defined as follows:

$$\mathrm{AUC} = \frac{1}{m^+ m^-} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} (I(f(x^+) > f(x^-)))) \tag{11}$$

where D^+ is the collection of all positive examples, D^- is the collection of all negative examples, $f(\cdot)$ is the result of the model's prediction of the sample x and $I(\cdot)$ is the indicator function.

Model	Advertising	Recommender
YoutubeNet-NO-UB[a]	0.6239	0.6419
YoutubeNet	0.6313	0.6425
DIN-RNN	0.6319	0.6435
Wide&Deep	0.6326	0.6432
DIN	0.6330	0.6459
DIEN	0.6343	0.6473
DSIN-PE[b]	0.6357	0.6494
DSIN-BE-NO-SIIL[c]	0.6365	0.6499
DSIN-BE[d]	**0.6375**	**0.6515**

[a] YoutubeNet without *User Behavior*.
[b] DSIN with positional encoding.
[c] DSIN with bias encoding and without session interest interacting layer and the corresponding activation unit.
[d] DSIN with bias encoding.

Table 1: Results (AUC) on the advertising and recommender dataset

4.4 Results on the Datasets

Results on the advertising dataset and recommender dataset are shown in Table 1. YoutubeNet performs better than YoutubeNet-No-User-Behavior owing to *User Behavior*, while Wide&Deep gets the betters result due to combining the memorization of wide part. DIN improves AUC obviously by activating *User Behavior* w.r.t. the target item. Especially, the results of DIN-RNN in both datasets are worse than those of DIN due to the discontinuity of users' behavior sequences. DIEN obtains better results while auxiliary loss and specially designed AUGRU lead to deviating from the original expression of behaviors. DSIN gets the best results on both datasets. It extracts users' historical behaviors into session interests and models the dynamic evolving procedure of session interests, both of which enrich the representation of the user. The local activation unit helps obtain the adaptive representation of users' session interests w.r.t. the target item.

[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

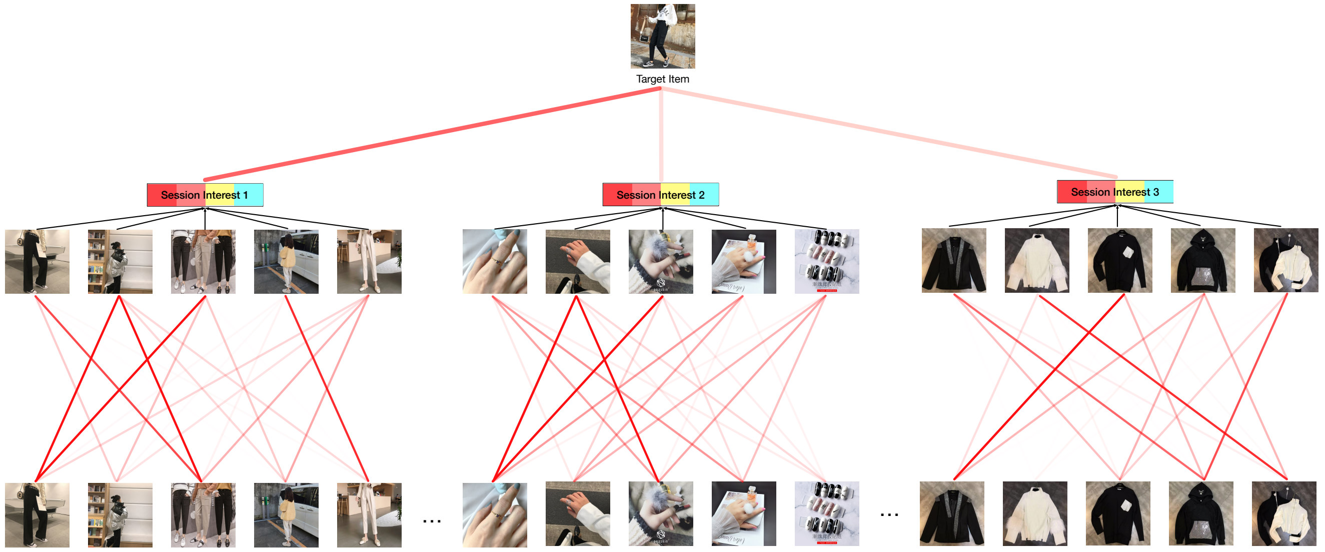

Figure 3: This figure visualizes the attention weights in the self-attention mechanism at the lower half and the activation unit work at the upper half in DSIN. Note that the attention weights in the self-attention mechanism are the sum of that in each head. Also, the darker the line color, the greater the weight.

4.5 Further Discussion

Effect of Multiple Sessions

As shown in Table 1, results show that DIN-RNN performs worse than DIN while DSIN-BE performs better than DSIN-BE-NO-SIIL. The difference between each pair is only the sequence modeling. [Zhou *et al.*, 2018c] explains that rapid jumping and sudden ending over behaviors causes the sequence data of user behaviors to seem to be noisy. It will lead to information loss in the process of information transmission in RNNs and further confuse the representation of users' behavior sequences. While in DSIN, we partition users' behavior sequences into multiple sessions for the following two reasons: (i) users' behaviors are generally homogeneous in each session; (ii) users' session interests follow a sequential pattern and are more suitable for sequence modeling. Both improve the performance of DSIN.

Effect of Session Interest Interacting Layer

As shown in Table 1, we conduct comparative experiments with DSIN-BE and DSIN-BE-NO-SIIL, where DSIN-BE performs better. With session interest interacting layer, users' session interests are mixed with contextual information and become more expressive, which improve the performance of DSIN.

Effect of Bias Encoding

As shown in Table 1, we conduct comparative experiments with DSIN-BE and DSIN-PE, where DSIN-BE performs better. Different from the two-dimensional positional encoding, the bias of users' sessions is also captured. Empirically, bias encoding successfully captures the order information of sessions and improves the performance of DSIN.

Visualization of Self-attention and the Activation Unit

As shown in figure 3, we visualize the attention weights in the the local activation unit and self-attention mechanism. To illustrate the effect of self-attention, we take the first session for example. The user mainly browses trouser-related items and occasionally coat-related items. We can observe that weights of trouser-related items are generally high. After self-attention, most representations of trouser-related behaviors are reserved and extracted into the user's interest in this session. Besides, the local activation unit works by making users' session interests related to the target item more prominent. In this case, the target item is a black trouser. The user's trouser-related session interest is assigned greater weight and has more influence on the final prediction. While the session 3 is coat-related, the user's color preference to black in it is also helpful to rank the trouser. And the session 2 is makeup-related and consequently contributes the least to the final prediction. So we conclude that each session is important for the final prediction score and use the attention mechanism to allocate different weights to different sessions.

5 Conclusion

In this paper, we provide a novel perspective on the CTR prediction task, where users' sequential behaviors consist of multiple historical sessions. User behaviors are highly in-session homogeneous and cross-session heterogeneous. Base on these observations, Deep Session Interest Network (DSIN) is proposed. We first use the self-attention mechanism with bias encoding to extract the user's interest of each session. Then we apply Bi-LSTM to capture the sequential relation of contextual session interests. We employ the local activation unit to aggregate the user's different session interest representations with regard to the target item at last. Experiment results demonstrate the effectiveness of DSIN both on advertising and recommender datasets. In the future, we will pay attention to utilizing knowledge graph as prior knowledge to explain users' historical behaviors for better CTR prediction.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Chen *et al.*, 2018] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 108–116. ACM, 2018.

[Cheng *et al.*, 2016] 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. ACM, 2016.

[Covington *et al.*, 2016] Covington, Paul, Adams, Jay, Sargin, and Emre. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[Graves and Schmidhuber, 2005] Alex Graves and Jrgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[Grbovic and Cheng, 2018] Mihajlo Grbovic and Haibin Cheng. Real-time personalization using embeddings for search ranking at airbnb. In *SIGKDD*, pages 311–320. ACM, 2018.

[Guo *et al.*, 2017] 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.

[Hidasi and Tikk, 2016] Balzs Hidasi and Domonkos Tikk. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery*, 30(2):342–371, 2016.

[Hidasi *et al.*, 2015] Balzs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[Hidasi *et al.*, 2016] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 241–248. ACM, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.

[Li *et al.*, 2017] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428. ACM, 2017.

[Li *et al.*, 2018] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *SIGKDD*, pages 1734–1743. ACM, 2018.

[Liu *et al.*, 2018] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. Stamp: short-term attention/memory priority model for session-based recommendation. In *SIGKDD*, pages 1831–1839. ACM, 2018.

[Quadrana *et al.*, 2017] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *the Eleventh ACM Conference*, 2017.

[Sarwar *et al.*, 2001] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295, 2001.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 565–573. ACM, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[Wang *et al.*, 2017] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, page 12. ACM, 2017.

[Xiao *et al.*, 2017] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*, 2017.

[Zhou *et al.*, 2018a] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. Atrank: An attention-based user behavior modeling framework for recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Zhou *et al.*, 2018b] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. *arXiv preprint arXiv:1809.03672*, 2018.

[Zhou *et al.*, 2018c] 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 *SIGKDD*, pages 1059–1068. ACM, 2018.