Deep Interest Evolution Network for Click-Through Rate Prediction

点击率预测的深度兴趣演化网络

Guorui Zhou,[1]* Na Mou,[1]† Ying Fan,[1] Qi Pi[1]
Weijie Bian,[1] Chang Zhou,[1] Xiaoqiang Zhu[1] and Kun Gai[1]
[1]Alibaba Inc, Beijing, China
{guorui.xgr, mouna.mn, fanying.fy, piqi.pq, weijie.bwj, ericzhou.zc, xiaoqiang.zxq, jingshi.gk}@alibaba-inc.com

Abstract

Click-through rate (CTR) prediction, whose goal is to estimate the probability of the user clicks, has become one of the core tasks in advertising systems. For CTR prediction model, it is necessary to capture the latent user interest behind the user behavior data. Besides, considering the changing of the external environment and the internal cognition, user interest evolves over time dynamically. There are several CTR prediction methods for interest modeling, while most of them regard the representation of behavior as the interest directly, and lack specially modeling for latent interest behind the concrete behavior. Moreover, few work consider the changing trend of interest. In this paper, we propose a novel model, named Deep Interest Evolution Network (DIEN), for CTR prediction. Specifically, we design interest extractor layer to capture temporal interests from history behavior sequence. At this layer, we introduce an auxiliary loss to supervise interest extracting at each step. As user interests are diverse, especially in the e-commerce system, we propose interest evolving layer to capture interest evolving process that is relative to the target item. At interest evolving layer, attention mechanism is embedded into the sequential structure novelly, and the effects of relative interests are strengthened during interest evolution. In the experiments on both public and industrial datasets, DIEN significantly outperforms the state-of-the-art solutions. Notably, DIEN has been deployed in the display advertisement system of Taobao, and obtained 20.7% improvement on CTR.

兴趣建模的点击率预测方法有几种，但大多将行为直接表示为兴趣，缺乏对具体行为背后的潜在兴趣进行专门的建模

辅助

Introduction

Cost per click (CPC) billing is one of the commonest billing forms in advertising system, where advertisers are charged for each click on their advertisement. In CPC advertising system, the performance of click-through rate (CTR) prediction not only influences the final revenue of whole system, but also impacts user experience and satisfaction. Modeling CTR prediction has drawn more and more attention from the communities of academia and industry.

In most non-searching e-commerce scenes, users do not express their current intention actively. Designing models to capture user's interests as well as their dynamics is the key to advance the performance of CTR prediction. Recently, many CTR models transform from traditional methodologies (Friedman 2001; Rendle 2010) to deep CTR models (Guo et al. 2017; Qu et al. 2016; Lian et al. 2018). Most deep CTR models focus on capturing interaction between features from different fields and pay less attention to user interest representation. Deep Interest Network (DIN) (Zhou et al. 2018c) emphasizes that user interests are diverse, it uses attention based model to capture relative interests to target item, and obtains adaptive interest representation for CTR prediction. However, most interest models including DIN regard the behavior as the interest directly. As we know, latent interest is hard to be fully reflected by explicit behavior. Previous methods neglect to dig the true user interest behind behavior. Moreover, user interest keeps evolving, capturing the dynamic of interest is important for interest representation. 用户兴趣不断变化，捕捉兴趣的动态对兴趣表示很重要

包括DIN在内的大多数模型，都是将行为直接视为兴趣

Based on all these observations, we propose Deep Interest Evolution Network (DIEN) to improve the performance of CTR prediction. There are two key modules in DIEN, one is for extracting latent temporal interests from explicit user behaviors, and the other one is for modeling interest evolving process. Proper interest representation is the footstone of interest evolving model. At interest extractor layer, DIEN chooses GRU (Chung et al. 2014) to model the dependency between behaviors. Following the principle that interest leads to the consecutive behavior directly, we propose auxiliary loss which uses the next behavior to supervise the learning of current hidden state. We call these hidden states with extra supervision as interest states. These extra supervision information helps to capture more semantic meaning for interest representation and push hidden states of GRU to represent interests effectively. Moreover, user interests are diverse, which leads to interest drifting phenomenon: user's intentions can be very different in adjacent visitings, and one behavior of a user may depend on the behavior that takes long time ago. Each interest has its own evolution track. Meanwhile, the click actions of one user on different target items are effected by different parts of interests. At interest evolving layer, we model the interest evolving trajectory that is relative to target item. Based on the interest sequence obtained from interest extractor layer, we design GRU with attentional update gate (AUGRU). Using interest state and target item to compute relevance, AUGRU strengthens rela-

一个是从用户的显性行为中提取潜在的时间兴趣

另一种是对兴趣演化过程进行建模

兴趣演化的前提是合理的表示出兴趣来

这些额外的监督信息有助于捕获更多的兴趣表征语义，推动GRU的隐藏状态有效地表征兴趣

第二步

*Corresponding author is Guorui Zhou.

†This author is the one who did the really hard work for Online Testing. The source code is available at https://github.com/mouna99/dien.

tive interests' influence on interest evolution, while weakens irrelative interests' effect that results from interest drifting. With the introduction of attentional mechanism into update gate, AUGRU can lead to the specific interest evolving processes for different target items. The main contributions of DIEN are as following:

- We focus on interest evolving phenomenon in e-commerce system, and propose a new structure of network to model interest evolving process. The model for interest evolution leads to more expressive interest representation and more precise CTR prediction.
- Different from taking behaviors as interests directly, we specially design interest extractor layer. Pointing at the problem that hidden state of GRU is less targeted for interest representation, we propose one auxiliary loss. Auxiliary loss uses consecutive behavior to supervise the learning of hidden state at each step. which makes hidden state expressive enough to represent latent interest.
- We design interest evolving layer novelly, where GPU with attentional update gate (AUGRU) strengthens the effect from relevant interests to target item and overcomes the inference from interest drifting. Interest evolving layer models interest evolving process that is related to target item effectively.

In the experiments on both public and industrial datasets, DIEN significantly outperforms the state-of-the-art solutions. It is notable that DIEN has been deployed in commercial display advertisement system and obtains significant improvement under various metrics.

Related Work

By virtue of the strong ability of deep learning on feature representation and combination, recent CTR models transform from traditional linear or nonlinear models (Friedman 2001; Rendle 2010) to deep models. Most deep models follow the structure of Embedding and Multi-ayer Perceptron (MLP) (Zhou et al. 2018c). Based on this basic paradigm, more and more models pay attention to the interaction between features: Both Wide & Deep (Cheng et al. 2016) and deep FM (Guo et al. 2017) combine low-order and high-order features to improve the power of expression; PNN (Qu et al. 2016) proposes a product layer to capture interactive patterns between interfield categories. In these models, user's history behaviors are transformed into low-dimension vector after the embedding and pooling operation, which can not reflect the interest behind data clearly. DIN (Zhou et al. 2018c) introduces the mechanism of attention to activate the historical behaviors w.r.t. given target item locally, and captures the diversity characteristic of user interests successfully. However, DIN is weak in capturing the dependencies between sequential behaviors.

In many application domains, user-item interactions can be recorded over time. A number of recent studies shows that this information can be used to build richer individual user models and discover additional behavioral patterns. In recommendation system, TDSSM (Song, Elkahky, and He 2016) jointly optimizes long-term and short-term user interests to improve the recommendation quality; DREAM (Yu et al. 2016) uses the structure of recurrent neural network (RNN) to investigate the dynamic representation of each user and the global sequential behaviors of item-purchase history. He and McAuley (2016) build visually-aware recommender system which surfaces products that more closely match users' and communities' evolving interests. Zhang et al. (2014) measures users' similarities based on user's interest sequence, and improves the performance of collaborative filtering recommendation. Parsana et al. (2018) improves native ads CTR prediction by using large scale event embedding and attentional output of recurrent networks. ATRank (Zhou et al. 2018a) uses attention-based sequential framework to model heterogeneous behaviors. Compared to sequence-independent approaches, these methods can significantly improve the click prediction accuracy.

However, these traditional RNN based models have some problems. On the one hand, most of them regard hidden states of sequential structure as latent interests directly, while these hidden states lack special supervision for interest representation. On the other hand, most existing RNN based models deal with all dependencies between adjacent behaviors successively and equally. As we know, not all user's behaviors are strictly dependent on each adjacent behavior. Each user has diverse interests, and each interest has its own evolving track. For any target item, these models can only obtain one fixed interest evolving track, so these models can be disturbed by interest drifting.

In order to push hidden states of sequential structure to represent latent interests effectively, extra supervision for hidden states should be introduceced. DARNN (Ren et al. 2018) uses click-level sequential prediction, which models the click action at each time when each ad is shown to the user. Besides click action, ranking information can be further introduced. In recommendation system, ranking loss has been widely used for ranking task (Rendle et al. 2009; Hidasi and Karatzoglou 2017). Similar to these ranking losses, we propose an auxiliary loss for interest learning. At each step, the auxiliary loss uses consecutive clicked item against non-clicked item to supervise the learning of interest representation.

For capturing interest evolving process that is related to target item, we need more flexible sequential learning structure. In the area of question answering (QA), DMN+ (Xiong, Merity, and Socher 2016) uses attention based GRU (AGRU) to push the attention mechanism to be sensitive to both the position and ordering of the inputs facts. In AGRU, the vector of the update gate is replaced by the scalar of attention score simply. This replacement neglects the difference between all dimensions of update gates, which contains rich information transmitted from previous sequence. Inspired from the novel sequential structure used in QA, we propose GRU with attentional gate (AUGRU) to activate relative interests during interest evolving. Different from AGRU, attention score in AUGRU acts on the information computed from update gate. The combination of update gate and attention score pushes the process of evolving more specifically and sensitively.

Deep Interest Evolution Network

In this section, we introduce Deep Interest Evolution Network (DIEN) in detail. First, we review the basic Deep CTR model, named BaseModel. Then we show the overall structure of DIEN. Next, we introduce the techniques that are used for capturing interests and modeling interest evolution process.

Review of BaseModel

The BaseModel is introduced from the aspects of feature representation, model structure and loss function, respectively.

Feature Representation In our online display system, we use four categories of feature: *User Profile*, *User Behavior*, *Ad* and *Context*. It is notable that the ad is also item. For generation, we call the ad as the target item in this paper. Each category of feature has several fields, *User Profile*'s fields are *gender*, *age* and so on; The fields of *User Behavior*'s are the list of user *visited goods id*; *Ad*'s fields are *ad_id*, *shop_id* and so on; *Context*'s fields are *devise type id*, *time* and so on. Feature in each field can be encoded into one-hot vector, e.g., the female feature in the category of *User Profile* are encoded as $[0, 1]$. The concat of different fields' one-hot vectors from category *User Profile*, *User Behavior*, *Ad* and *Context* form $\mathbf{x}_p, \mathbf{x}_a, \mathbf{x}_c, \mathbf{x}_b$, respectively. In sequential CTR model, it's remarkable that each field contains a list of behaviors, and each behavior corresponds to a one-hot vector, which can be represented by]

$$\mathbf{x}_b = [\mathbf{b}_1; \mathbf{b}_2; \cdots; \mathbf{b}_T] \in \mathbb{R}^{K \times T}, \mathbf{b}_t \in \{0, 1\}^K, \quad (1)$$

where $\mathbf{b}_t$ is encoded as one-hot vector and represents t-th behavior, T is the number of user's history behaviors, K is the total number of goods that user can click.

The Structure of BaseModel Most deep CTR models are built on the basic structure of embedding & MLR. The basic structure is composed of several parts:

- **Embedding** Embedding is the common operation that transforms the large scale sparse feature into low-dimensional dense feature. In the embedding layer, each field of feature is corresponding to one embedding matrix, e.g., the embedding matrix of visited goods can be represented by $\mathrm{E}_{goods} = [\mathbf{m}_1; \mathbf{m}_2; \cdots; \mathbf{m}_K] \in \mathbb{R}^{n_E \times K}$, where $\mathbf{m}_j \in \mathbb{R}^{n_E}$ represents a embedding vector with dimension n_E. Especially, for behavior feature $\mathbf{b}_t$, if $\mathbf{b}_t[j_t] = 1$, then its corresponding embedding vector is $\mathbf{m}_{j_t}$, and the ordered embedding vector list of behaviors for one user can be represented by $\mathbf{e}_b = [\mathbf{m}_{j_1}; \mathbf{m}_{j_2}; \cdots, \mathbf{m}_{j_T}]$. Similarly, $\mathbf{e}_a$ represents the concatenated embedding vectors of fields in the category of advertisement.
- **Multilayer Perceptron** (MLP) First, the embedding vectors from one category are fed into pooling operation. Then all these pooling vectors from different categories are concatenated. At last, the concatenated vector is fed into the following MLP for final prediction.

Loss Function The widely used loss function in deep CTR models is negative log-likelihood function, which uses the label of target item to supervise overall prediction:

$$L_{target} = -\frac{1}{N} \sum_{(\mathbf{x},y) \in \mathcal{D}}^{N} (y \log p(\mathbf{x}) + (1-y) \log(1-p(\mathbf{x}))), \quad (2)$$

where $\mathbf{x} = [\mathbf{x}_p, \mathbf{x}_a, \mathbf{x}_c, \mathbf{x}_b] \in \mathcal{D}$, $\mathcal{D}$ is the training set of size N. $y \in \{0, 1\}$ represents whether the user clicks target item. $p(\mathbf{x})$ is the output of network, which is the predicted probability that the user clicks target item.

Deep Interest Evolution Network

Different from sponsored search, in many e-commerce platforms like online display advertisement, users do not show their intention clearly, so capturing user interest and their dynamics is important for CTR prediction. DIEN devotes to capture user interest and models interest evolving process. As shown in Fig. 1, DIEN is composed by several parts: First, all categories of features are transformed by embedding layer. Next, DIEN takes two steps to capture interest evolving: interest extractor layer extracts interest sequence based on behavior sequence; interest evolving layer models interest evolving process that is relative to target item. Then final interest's representation and embedding vectors of ad, user profile, context are concatenated. The concatenated vector is fed into MLP for final prediction. In the remaining of this section, we will introduce two core modules of DIEN in detail.

Interest Extractor Layer In e-commerce system, user behavior is the carrier of latent interest, and interest will change after user takes one behavior. At the interest extractor layer, we extract a series of interest states from sequential user behaviors.

The user behaviors in e-commerce system are rich, where the length of history behavior sequence is long even in a short period of time, like two weeks. For the balance between efficiency and performance, we take GRU to model the dependency between behaviors. where the input of GRU is ordered behaviors by their occur time. GRU overcomes the vanishing gradients problem of RNN and is faster than LSTM (Hochreiter and Schmidhuber 1997), which is suitable for e-commerce system. The formulations of GRU are listed as follows:

$$\mathbf{u}_t = \sigma(W^u \mathbf{i}_t + U^u \mathbf{h}_{t-1} + \mathbf{b}^u), \quad (3)$$

$$\mathbf{r}_t = \sigma(W^r \mathbf{i}_t + U^r \mathbf{h}_{t-1} + \mathbf{b}^r), \quad (4)$$

$$\tilde{\mathbf{h}}_t = \tanh(W^h \mathbf{i}_t + \mathbf{r}_t \circ U^h \mathbf{h}_{t-1} + \mathbf{b}^h), \quad (5)$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{u}_t) \circ \mathbf{h}_{t-1} + \mathbf{u}_t \circ \tilde{\mathbf{h}}_t, \quad (6)$$

where σ is the sigmoid activation function, $\circ$ is element-wise product, $W^u, W^r, W^h \in \mathbb{R}^{n_H \times n_I}$, $U^z, U^r, U^h \in n_H \times n_H$, n_H is the hidden size, and n_I is the input size. $\mathbf{i}_t$ is the input of GRU, $\mathbf{i}_t = \mathbf{e}_b[t]$ represents the t-th behavior that the user taken, $\mathbf{h}_t$ is the t-th hidden states.

However, the hidden state $\mathbf{h}_t$ which only captures the dependency between behaviors can not represent interest effectively. As the click behavior of target item is triggered by final interest, the label used in L_{target} only contains the ground truth that supervises final interest's prediction, while history state $\mathbf{h}_t$ $(t < T)$ can't obtain proper supervision. As we all know, interest state at each step leads to consecutive behavior directly. So we propose auxiliary loss, which uses

这是本篇论文的核心创新点

这个地方在表达这么个意思，item算是一个field，每个item都会表示成onehot的形式，并且会对应一个embedding向量，这个对应是基于位置上的对应关系。即OneHot为1的那个数，就是embedding向量的位置。 用户行为列表会有很多个item，而每个item会对应一个category，这样的话用户行为里面的每个item的embedding拼接起来，就是eb, 而对于category也是同样的道理。

这里的ea是某个广告对应的类别embedding，注意是某个

目标商品的点击行为是由最后的兴趣触发，而label只能监督最后的兴趣，对隐藏层的状态不能有效的监督

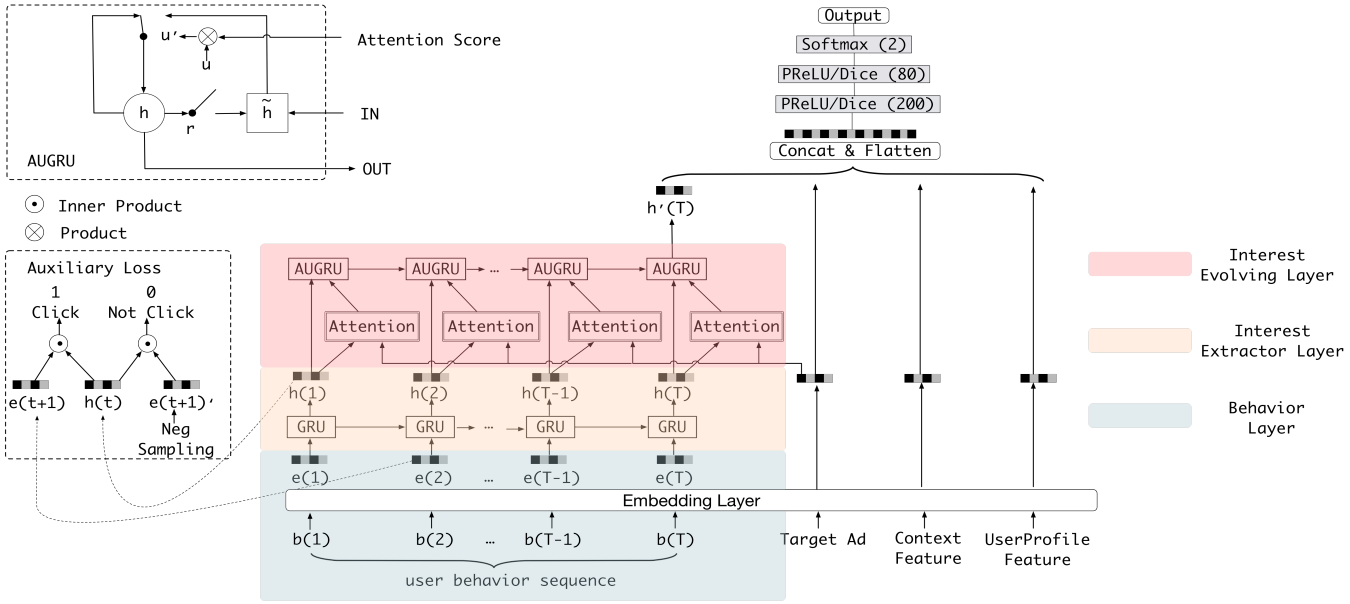


Figure 1: The structure of DIEN. At the behavior layer, behaviors are sorted by time, the embedding layer transforms the one-hot representation $\mathbf{b}[t]$ to embedding vector $\mathbf{e}[t]$. After the behavior layer, interest extractor layer extracts each interest state $\mathbf{h}[t]$ with the help of auxiliary loss. At interest evolving layer, AUGRU models the interest evolving process that is relative to target item. The final interest state $\mathbf{h}'[T]$ and embedding vectors of remaining feature are concatenated, and fed into MLR for final CTR prediction.

behavior $\mathbf{b}_{t+1}$ to supervise the learning of interest state $\mathbf{h}_t$. Besides using the real next behavior as positive instance, we also use negative instance that samples from item set except the clicked item.

There are N pairs of behavior embedding sequence: $\{\mathbf{e}_b^i, \hat{\mathbf{e}}_b^i\} \in \mathcal{D}_{\mathcal{B}}, i \in 1, 2, \cdots, N$, where $\mathbf{e}_b^i \in \mathbb{R}^{T \times n_E}$ represents the clicked behavior sequence, and $\hat{\mathbf{e}}_b^i \in \mathbb{R}^{T \times n_E}$ represent the negative sample sequence. T is the number of history behaviors, n_E is the dimension of embedding, $\mathbf{e}_b^i[t] \in \mathcal{G}$ represents the t-th item's embedding vector that user i click, $\mathcal{G}$ is the whole item set. $\hat{\mathbf{e}}_b^i[t] \in \mathcal{G} - \mathbf{e}_b^i[t]$ represents the embedding of item that sample from the item set except the item clicked by user i at t-th step. Auxiliary loss can be formulated as:

这个东西和交叉熵类似

$$L_{aux} = -\frac{1}{N}(\sum_{i=1}^{N}\sum_{t}\log\sigma(\mathbf{h}_t, \mathbf{e}_b^i[t+1]) + \log(1-\sigma(\mathbf{h}_t, \hat{\mathbf{e}}_b^i[t+1]))), \tag{7}$$

where $\sigma(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{1+\exp(-[\mathbf{x}_1, \mathbf{x}_2])}$ is sigmoid activation function, $\mathbf{h}_t$ represents the t-th hidden state of GRU. The global loss we use in our CTR model is:

$$L = L_{target} + \alpha * L_{aux}, \tag{8}$$

where α is the hyper-parameter which balances the interest representation and CTR prediction.

With the help of auxiliary loss, each hidden state $\mathbf{h}_t$ is expressive enough to represent interest state after user takes behavior $\mathbf{i}_t$, and the concat of all T interest points $[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T]$ compose the interest sequence that interest evolving layer can model interest evolving on.

Overall, the introduction of auxiliary loss has several advantages: from the aspect of interest learning, the introduction of auxiliary loss helps each hidden state of GRU represent interest expressively. As for the optimization of GRU, auxiliary loss reduces the difficulty of back propagation when GRU models long history behavior sequence. Last but not the least, auxiliary loss gives more semantic information for the learning of embedding layer, which leads to a better embedding matrix.

每个隐藏层都能表示出兴趣来

降低反向传播的难度

为嵌入矩阵提供了更多语义信息，方便这个东西学习

Interest Evolving Layer As the joint influence from external environment and internal cognition, different kinds of user interests are evolving over time. Using the interest on clothes as an example, with the changing of population trend and user taste, user's preference for clothes evolves. The evolving process of the user interest on clothes will directly decides CTR prediction for candidate clothes. The advantages of modeling the evolving process is as follows:

- Interest evolving module could supply the representation of final interest with more relative history information;
- It is better to predict the CTR of target item by following the interest evolution trend.

Notably, interest shows two characteristics during evolving:

- As the diversity of interests, interest can drift. The effect of interest drifting on behaviors is that user may interest in kinds of books during a period of time, and need clothes in another time.
- Though interests may affect each other, each interest has its own evolving process, e.g. the evolving process of books and clothes is almost individually. We only concerns the evolving process that is relative to target item.

兴趣漂移的例子

In the first stage, with the help of auxiliary loss, we has ob-

这个改进给我的感觉，就是在每一步会输出一个值求一个损失，类似于每一步都输出的RNN，只不过和那个不太一样的是，每一步都做一个二分类任务了
而之前不这么弄， 就相当于只有最后一个时间步输出的RNN， 这在这种行为序列模拟的场景下不太适合，中间的隐藏状态没法进行监督
而当每一个时间步都搞一个输出计算损失的话，就相当于中间的隐藏状态有监督了。 而这个原理我感觉和NLP里面的机器翻译的那种非常类似。

通过分析兴趣的演化特征，结合注意机制的局部激活能力和GRU的顺序学习能力，构建兴趣的演化模型。

tained expressive representation of interest sequence. By analyzing the characteristics of interest evolving, we combine the local activation ability of attention mechanism and sequential learning ability from GRU to model interest evolving. The local activation during each step of GRU can intensify relative interest's effect, and weaken the disturbance from interest drifting, which is helpful for modeling interest evolving process that relative to target item.

两层GRU，这是第二层

Similar to the formulations shown in Eq. (3-6), we use $\mathbf{i}'_t$, $\mathbf{h}'_t$ to represent the input and hidden state of GRU used in interest evolving module, where the input of second GRU is the corresponding interest state at Interest Extractor Layer: $\mathbf{i}'_t = \mathbf{h}_t$. The last hidden state $\mathbf{h}'_T$ represents final interest state.

And the attention function we used in interest evolving module can be formulated as:

$$a_t = \frac{\exp(\mathbf{h}_t W \mathbf{e}_a)}{\sum_{j=1}^{T} \exp(\mathbf{h}_j W \mathbf{e}_a)}, \tag{9}$$

where $\mathbf{e}_a$ is the concat of embedding vectors from fields in category ad, $W \in \mathbb{R}^{n_H \times n_A}$, n_H is the dimension of hidden state and n_A is the dimension of advertisement's embedding vector. Attention score can reflect the relationship between advertisement $\mathbf{e}_a$ and input $\mathbf{h}_t$, and strong relativeness leads to a large attention score.

Next, we will introduce several approaches that combine the mechanism of attention and GRU to model the process of interest evolution.

这里的at是个标量，表示的是当前隐藏状态与目标target的相关性

- **GRU with attentional input (AIGRU)** In order to activate relative interests during interest evolution, we propose a naive method, named GRU with attentional input (AIGRU). AIGRU uses attention score to affect the input of interest evolving layer. As shown in Eq. (10):

$$\mathbf{i}'_t = \mathbf{h}_t * a_t \tag{10}$$

AIGRU中，相关兴趣较少的量表可以通过注意分数来降低。理想情况下，相关兴趣较少的输入值可以减少到零，这样我们就可以对相对于目标项目的利益演进趋势进行建模

Where $\mathbf{h}_t$ is the t-th hidden state of GRU at interest extractor layer, $\mathbf{i}'_t$ is the input of the second GRU which is for interest evolving, and $*$ means scalar-vector product. In AIGRU, the scale of less related interest can be reduced by the attention score. Ideally, the input value of less related interest can be reduced to zero, then we can model the interest evolving trend that is relative to target item. However, AIGRU works not very well. Because even zero input can also change the hidden state of GRU, so the less relative interests also affect the learning of interest evolving.

0值也会改变隐藏状态，看GRU的更新公式就能看出来，input是0值，ht依然可以学习

- **Attention based GRU(AGRU)** In the area of question answering (Xiong, Merity, and Socher 2016), attention based GRU (AGRU) is firstly proposed. After modifying the GRU architecture by embedding information from the attention mechanism, AGRU can extract key information in complex queries effectively. Inspired by the question answering system, we transfer the using of AGRU from extracting key information in query to capture relative interest during interest evolving novelly. In detail, AGRU uses the attention score to replace the update gate of GRU, and changes the hidden state directly. Formally:

$$\mathbf{h}'_t = (1 - a_t) * \mathbf{h}'_{t-1} + a_t * \tilde{\mathbf{h}}'_t, \tag{11}$$

where $\mathbf{h}'_t$, $\mathbf{h}'_{t-1}$ and $\tilde{\mathbf{h}}'_t$ are the hidden state of AGRU.

Table 1: The statistics of datasets

Dataset	User	Goods	Categories	Samples
Books	603,668	367,982	1,600	603,668
Electronics	192,403	63,001	801	192,403
Industrial dataset	0.8 billion	0.82 billion	18,006	7.0 billion

In the scene of interest evolving, AGRU makes use of the attention score to control the update of hidden state directly. AGRU weakens the effect from less related interest during interest evolving. The embedding of attention into GRU improves the influence of attention mechanism, and helps AGRU overcome the defects of AIGRU.

- **GRU with attentional update gate (AUGRU)** Although AGRU can use attention score to control the update of hidden state directly, it use s a scalar (the attention score a_t) to replace a vector (the update gate u_t), which ignores the difference of importance among different dimensions. We propose the GRU with attentional update gate (AUGRU) to combine attention mechanism and GRU seamlessly:

$$\tilde{\mathbf{u}}'_t = a_t * \mathbf{u}'_t, \tag{12}$$

$$\mathbf{h}'_t = (1 - \tilde{\mathbf{u}}'_t) \circ \mathbf{h}'_{t-1} + \tilde{\mathbf{u}}'_t \circ \tilde{\mathbf{h}}'_t, \tag{13}$$

where $\mathbf{u}'_t$ is the original update gate of AUGRU, $\tilde{\mathbf{u}}'_t$ is the attentional update gate we design for AUGRU, $\mathbf{h}'_t, \mathbf{h}'_{t-1}$, and $\tilde{\mathbf{h}}'_t$ are the hidden states of AUGRU.

In AUGRU, we kept original dimensional information of update gate, which decides the importance of each dimension. Based on the differentiated information, we use attention score a_t to scale all dimensions of update gate, which results that less related interest make less effects on the hidden state. AUGRU avoids the disturbance from interest drifting more effectively, and pushes the relative interest to evolve smoothly.

Experiments

In this section, we compare DIEN with the state of the art on both public and industrial datasets. Besides, we design experiments to verify the effect of auxiliary loss and AUGRU, respectively. For observing the process of interest evolving, we show the visualization result of interest hidden states. At last, we share the results and techniques that we used for online serving. 网上服务的结果和技术是需要学习的

Datasets

In the section of experiment, we use both public and industrial datasets to verify the effect of DIEN. The statistics of all datasets are shown in Table 1.

public Dataset Amazon Dataset (McAuley et al. 2015) is composed of product reviews and metadata from Amazon. We use two subsets of Amazon dataset: Books and Electronics, to verify the effect of DIEN. In these datasets, we regard reviews as behaviors, and sort the reviews from one user by time. Assuming there are T behaviors of user u, our purpose is to use the $T-1$ behaviors to predict whether user u will write reviews that shown in T-th review.

Industrial Dataset Industrial dataset is constructed by impression and click logs from one online display advertising system. For training set, we take the ads that clicked at last

Table 2: Results (AUC) on public datasets

Model	Electronics (AUC)	Books (AUC)
BaseModel (Zhou et al. 2018c)	0.8538	0.8846
Wide&Deep (Cheng et al. 2016)	0.8546	0.8656
PNN (Qu et al. 2016)	0.8582	0.8935
DIN (Zhou et al. 2018c)	0.8608	0.8974
Two layer GRU Attention	0.8640	0.8993
DIEN	**0.9030**	**0.9281**

49 days as the target item. Each target item and its corresponding behaviors construct one instance. Using one target item a as example, we set the time that a is clicked as the last day, the behaviors that this user takes in previous 14 days as history behaviors. Similarly, the target item in test set is choose from the following 1 day, and the behaviors are built as same as training data.

Compared Methods

We compare DIEN with some mainstream CTR prediction methods:

- ***BaseModel*** BaseModel takes the same setting of embedding and MLR with DIEN, and uses sum pooling operation to integrate behavior embeddings.
- ***Wide&Deep*** (Cheng et al. 2016) Wide & Deep consists of two parts: its deep model is the same as Base Model, and its wide model is a linear model which uses manually designed cross-product feature for better interactions.
- ***PNN*** (Qu et al. 2016) Based on the BaseModel, PNN uses a product layer to capture interactive patterns between interfield categories.
- ***DIN*** (Zhou et al. 2018c) DIN uses the mechanism of attention to activate related user behaviors and obtains an adaptive representation vector for user interests which varies over different ads.
- ***Two layer GRU Attention*** Similar to (Parsana et al. 2018), we use two layer GRU to model sequential behaviors, and takes an attention layer to active relative behaviors.

Results on Public Datasets

Overall, the structure of DIEN consists GRU, AUGRU and auxiliary loss and other normal components, which is shown in Fig. 1. Each experiment is repeated 5 times, and we show the mean AUC in Table 2.

From Table 2, we can find *Wide & Deep* depends on the quality of manually designed features heavily and performs not well, while automative interaction between features (*PNN*) can improve the performance of *BaseModel*. At the same time, the models aiming to capture interests can improve AUC obviously: *DIN* actives the interests that relative to ad, *Two layer GRU attention* further activates relevant interests in interest sequence, all these explorations obtain positive feedback. DIEN not only captures sequential interests more effectively, but also models the interest evolving process that is relative to target item. The modeling for interest evolving helps DIEN obtain better interest representation, and capture dynamic of interests precisely, which improves the performance largely.

Results on Industrial Dataset

We further conduct experiments on the dataset of real display advertisement. The number of instances in industrial dataset is thousand times of public dataset, and the behaviors are richer than public dataset. As shown in Table 3, *Wide & Deep* and *PNN* obtain better performance than *BaseModel*. Different from only one category of goods in Amazon dataset, the dataset of online advertisement contains all kinds of goods at the same time. Based on this characteristic, attention-based methods improve the performance largely, like *DIN*. DIEN captures the interest evolving process that is relative to target item, and obtains best performance.

Table 3: Results (AUC) on industrial dataset

Model	AUC
BaseModel (Zhou et al. 2018c)	0.6350
Wide&Deep (Cheng et al. 2016)	0.6362
PNN (Qu et al. 2016)	0.6353
DIN (Zhou et al. 2018c)	0.6428
Two layer GRU Attention	0.6457
BaseModel + GRU + AUGRU	0.6493
DIEN	**0.6541**

Application Study

In this section, we will show the effect of AUGRU and auxiliary loss, respectively.

Table 4: Effect of AUGRU and auxiliary loss

Model	Electronics (AUC	Books (AUC)
BaseModel	0.8538	0.8846
Two layer GRU attention	0.8640	0.8993
BaseModel + GRU + AIGRU	0.8663	0.8994
BaseModel + GRU + AGRU	0.8666	0.8993
BaseModel + GRU + AUGRU	0.8675	0.9021
DIEN	**0.9030**	**0.9281**

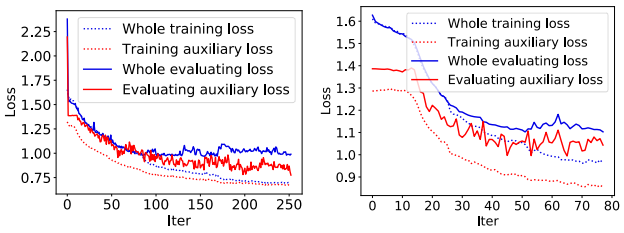
Effect of GRU with attentional update gate (AUGRU) Table 4 shows the results of different methods for interest evolving. Compared to *BaseMode*, *Two layer GRU Attention* obtains improvement, while the lack for modeling evolution limits its ability. *AIGRU* takes the basic idea to model evolving process, though it has advances, the splitting of attention and evolving lost information during interest evolving. *AGRU* further tries to fuse attention and evolution, as we proposed previously, its attention in GRU can't make fully use the resource of update gate. Compared to *AIGRU* and *AGRU*, it's not difficult to find that *AUGRU* obtains obvious improvements, which reflects it fuses the attention mechanism and sequential learning ideally, and captures the evolving process of relative interests effectively.

Attention和进化的分离版本

初步融合，但是没有考虑所有维度

Effect of auxiliary loss Based on the model that obtained with AUGRU, we further explore the effect of auxiliary loss. In the public datasets, the negative instance used in the auxiliary loss is randomly sampled from item set except the item shown in corresponding review. As for industrial dataset, we take the ad that been shown to user while not been clicked as negative instance.

As shown in Fig. 2, the loss of both whole loss L and auxiliary loss L_aux keep similar descend trend, which means



(a) Books (b) Electronics

Figure 2: Learning curves for public datasets, where α is set as 1. Because of the negative sampling used in auxiliary loss, the scale of it is large.

both global loss for CTR prediction and auxiliary loss for interest representation make effect.

In Table 4, we find the auxiliary loss can bring great improvements for both public datasets, The great performance of auxiliary loss in public datasets reflects the importance of supervision information for the learning of sequential interests. What's more, the supervision for each step of GRU also helps the model obtain more expressive embedding representation. For the result of online dataset shown in Table 3, model with auxiliary loss improves performance further. However, we can see that the improvement is not as obvious as that in public dataset. The difference derives from several aspects. First, for industrial dataset, it has a large number of instances to learn the embedding layer, which makes it earn less from auxiliary loss. Second, different from all items from one category in amazon dataset, the behaviors in industrial dataset are clicked goods from all scenes and all catagories in our platform. Our goal is to predict CTR for ad in one scene. The supervision information from auxiliary loss may be heterogeneous from the target item, so the effect of auxiliary loss for the industrial dataset may be less for public datasets, while the effect of AUGRU is magnified.

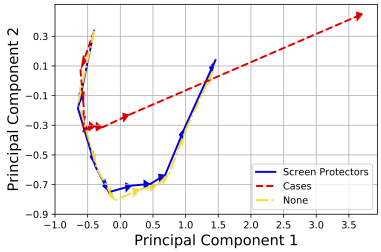
Visualization of Interest Evolution

The dynamic of hidden states in AUGRU can reflect the evolving process of interest. In this section, we visualize these hidden states to explore the effect of different target item for interest evolution. The selective history behaviors are from category *Computer Speakers, Headphones, Vehicle GPS, SD & SDHC Cards, Micro SD Cards, External Hard Drives, Headphones, Cases*, successively. The hidden states in AUGRU are projected into a two dimension space by principal component analysis (PCA) (Wold, Esbensen, and Geladi 1987). The projected hidden states are linked in order. The moving routes of hidden states activated by different target item are shown in Fig. 3(a). The yellow curve which is with *None* target represents the attention score used in eq. (13) are equal, that is the evolution of interest are not effected by target item. The blue curve shows the hidden states are activated by one goods from category *Screen Protectors*, which is less related to all history behaviors, so the blue curve shows similar route to yellow curve. The red curve shows the hidden states are activated by one goods from category *Cases*, the target item is strong related to the last behavior, which moves a long step shown in Fig. 3(a).

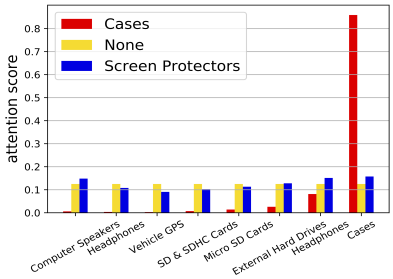
Table 5: Results from Online A/B testing

Model	CTR Gain	PPC Gain	eCPM Gain
BaseModel	0%	0%	0%
DIN (Zhou et al. 2018c)	+ 8.9%	- 2.0%	+ 6.7%
DIEN	+ 20.7%	- 3.0%	+ 17.1%

Correspondingly, the last behavior obtains a large attention score showed in Fig. 3(b).



(a) Visualization of hidden states in AUGRU



(b) Attention score of different history behaviors

Figure 3: Visualization of interest evolution, (a) The hidden states of AUGRU reduced by PCA into two dimensions. Different curves shows the same history behaviors are activated by different target items.*None* means interest evolving is not effected by target item. (b) Faced with different target item, attention scores of all history behaviors are shown.

Online Serving & A/B testing

From 2018-06-07 to 2018-07-12, online A/B testing was conducted in the display advertising system of Taobao. As shown in Table 5, compared to the BaseModel, DIEN has improved CTR by 20.7% and effective cost per mille (eCPM) by 17.1%. Besides, DIEN has decayed pay per click (PPC) by 3.0%. Now DIEN has been deployed online and serves the main traffic, which contributes a significant business revenue growth. It is worth noticing that online serving of DIEN is a great challenge for commercial system. Online system holds really high traffic in our display advertising system, which serves more than 1 million users per second at traffic peak. In order to keep low latency and high throughput, we deploy several important techniques to improve serving performance: i) *element parallel GRU & kernel fusion* (Wang, Lin, and Yi 2010): we fuse as many independent kernels as possible. Besides, each element of the hidden state of GRU can be calculated in parallel. ii) *Batching*: adjacent requests from different users are

merged into one batch to take advantage of GPU. iii) *Model compressing with Rocket Launching* (Zhou et al. 2018b): we use the method proposed in (Zhou et al. 2018b) to train a light network, which has smaller size but performs close to the deeper and more complex one. For instance, the dimension of GRU hidden state can be compressed from 108 to 32 with the help of Rocket Launching. After taking advantage of these techniques, latency of DIEN serving can be reduced from 38.2 ms to 6.6 ms and the QPS (Query Per Second) capacity of each worker can be improved to 360.

Conclusion

In this paper, we propose a new structure of deep network, namely Deep Interest Evolution Network (DIEN), to model interest evolving process. DIEN improves the performance of CTR prediction largely in online advertising system. Specifically, we design interest extractor layer to capture interest sequence particularly, which uses auxiliary loss to provide the interest state with more supervision. Then we propose interest evolving layer, where DIEN uses GRU with attentional update gate (AUGRU) to model the interest evolving process that is relative to target item. With the help of AUGRU, DIEN can overcome the disturbance from interest drifting. Modeling for interest evolution helps us capture interest effectively, which further improves the performance of CTR prediction. In future, we will try to construct a more personalized interest model for CTR prediction.

References

[Cheng et al. 2016] Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 7–10. ACM.

[Chung et al. 2014] Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[Friedman 2001] Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* 1189–1232.

[Guo et al. 2017] Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2782–2788.

[He and McAuley 2016] He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th international conference on world wide web*, 507–517.

[Hidasi and Karatzoglou 2017] Hidasi, B., and Karatzoglou, A. 2017. Recurrent neural networks with top-k gains for session-based recommendations. *arXiv preprint arXiv:1706.03847*.

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

[Lian et al. 2018] Lian, J.; Zhou, X.; Zhang, F.; Chen, Z.; Xie, X.; and Sun, G. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[McAuley et al. 2015] McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 43–52. ACM.

[Parsana et al. 2018] Parsana, M.; Poola, K.; Wang, Y.; and Wang, Z. 2018. Improving native ads ctr prediction by large scale event embedding and recurrent networks. *arXiv preprint arXiv:1804.09133*.

[Qu et al. 2016] Qu, Y.; Cai, H.; Ren, K.; Zhang, W.; Yu, Y.; Wen, Y.; and Wang, J. 2016. Product-based neural networks for user response prediction. In *Proceedings of the16th International Conference on Data Mining*, 1149–1154. IEEE.

[Ren et al. 2018] Ren, K.; Fang, Y.; Zhang, W.; Liu, S.; Li, J.; Zhang, Y.; Yu, Y.; and Wang, J. 2018. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. *arXiv preprint arXiv:1808.03737*.

[Rendle et al. 2009] Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, 452–461. AUAI Press.

[Rendle 2010] Rendle, S. 2010. Factorization machines. In *Proceedings of the 10th International Conference on Data Mining*, 995–1000. IEEE.

[Song, Elkahky, and He 2016] Song, Y.; Elkahky, A. M.; and He, X. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 909–912. ACM.

[Wang, Lin, and Yi 2010] Wang, G.; Lin, Y.; and Yi, W. 2010. Kernel fusion: An effective method for better power efficiency on multithreaded gpu. In *Proceedings of the 2010 IEEE/ACM Int'L Conference on Green Computing and Communications & Int'L Conference on Cyber, Physical and Social Computing*, 344–350.

[Wold, Esbensen, and Geladi 1987] Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2(1-3):37–52.

[Xiong, Merity, and Socher 2016] Xiong, C.; Merity, S.; and Socher, R. 2016. Dynamic memory networks for visual and textual question answering. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2397–2406.

[Yu et al. 2016] Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 729–732. ACM.

[Zhang et al. 2014] Zhang, Y.; Dai, H.; Xu, C.; Feng, J.; Wang, T.; Bian, J.; Wang, B.; and Liu, T.-Y. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 1369–1375.

[Zhou et al. 2018a] Zhou, C.; Bai, J.; Song, J.; Liu, X.; Zhao, Z.; Chen, X.; and Gao, J. 2018a. Atrank: An attention-based user behavior modeling framework for recommendation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[Zhou et al. 2018b] Zhou, G.; Fan, Y.; Cui, R.; Bian, W.; Zhu, X.; and Gai, K. 2018b. Rocket launching: A universal and efficient framework for training well-performing light net. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[Zhou et al. 2018c] Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018c. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1059–1068. ACM.