

# Deep Session Interest Network for Click- Through Rate Prediction

深度会话兴趣网络（DSIN）点击率预测

2019.05.16

分享人：宋彤彤

2019.06.28

# DSIN—点击率预测

- 1. 研究背景
- 2. DSIN 模型
- 3. 实验与结果
- 4. 未来研究方向

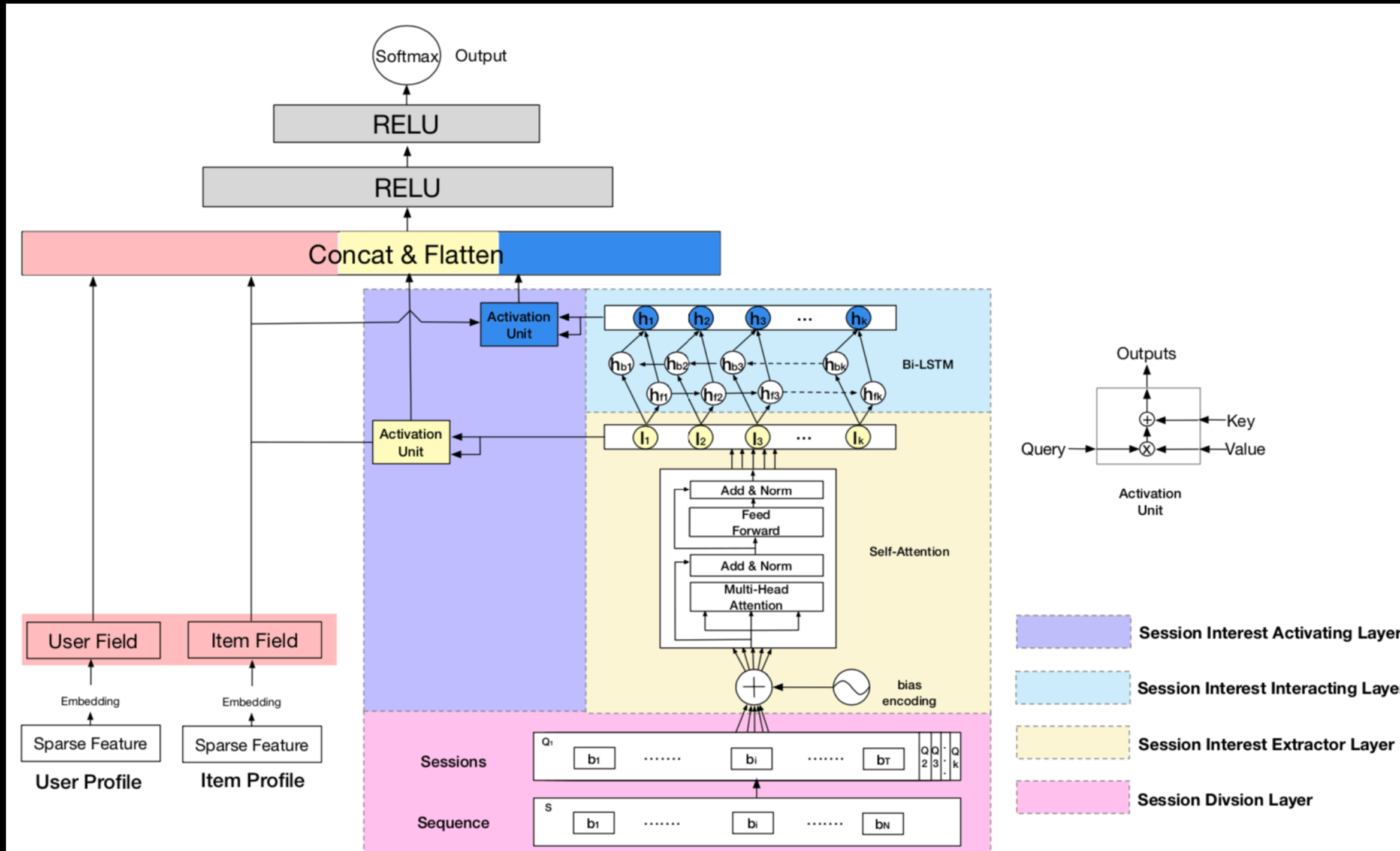
# 1. 研究背景

# 1. 研究背景

- DIN 和 DIEN 利用用户行为序列，挖掘用户动态演化兴趣。
- 但忽视用户行为序列是会话，一个session是在给定的时间范围内发生的交互的列表。
- 用户行为在会话内部高度同构，跨会话高度异构



## 2. DSIN 模型



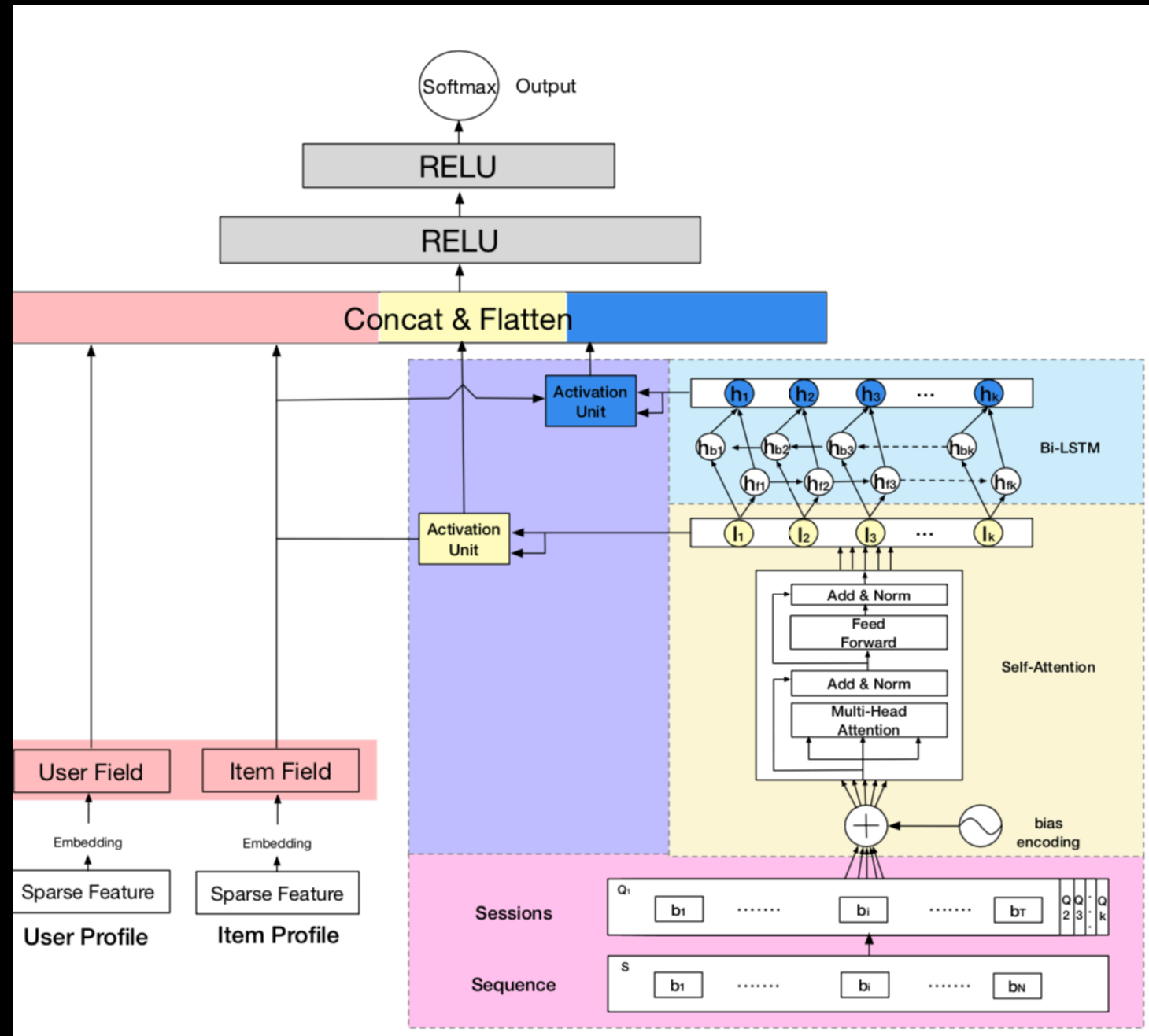
DSIN 整体模型

# 2.1 BaseModel - basic deep CTR model

- 全连接神经网络
- 输入特征：用户特征，待推荐物品特征，用户历史行为序列特征
- 特征通过 Embedding 层转换为对应的 embedding，拼接后输入到多层全连接（MLP）中，并使用log-loss 指导模型的训练。

# 2.2 DSIN

- 左：embedding 层，将用户特征和物品特征转换对应的向量表示
- 右：对用户行为序列进行处理
- 2.2.1 序列切分层
- 2.2.2 会话兴趣抽取层
- 2.2.3 会话间兴趣交互层
- 2.2.4 会话兴趣激活层





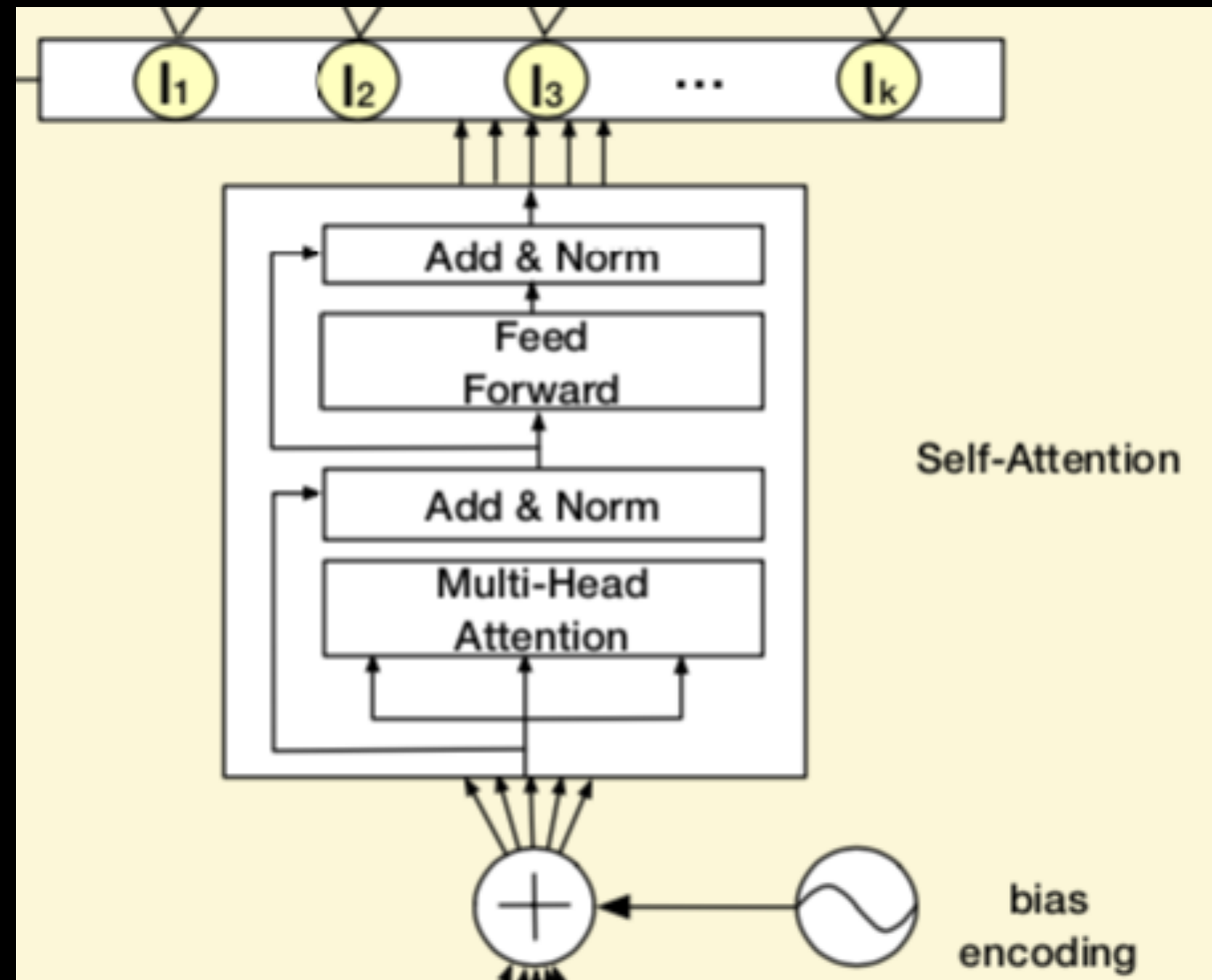
## 2.2.1 序列切分层



- 将用户的行为进行切分：
- 用户点击行为按时间排序，前后时间间隔大于30min 进行切分
- 切分后，用户的行为序列 S 转换成会话序列 Q 。  
第 k 个会话  $Q_k=[b_1;b_2;...;b_i;...;b_T]$ ，其中，T 是会话的长度， $b_i$  是会话中第 i 个行为——一个 d 维的 embedding 向量。

## 2.2.2 会话兴趣抽取层

- 这一步的目的就是去捕获 session list 中每个 session 内部的行为关系。将 session 作为输入，送入 transformer 结构。
- $I_k$  就代表第  $k$  个 session 对应的兴趣向量。



## 2.2.2 会话兴趣抽取层

session在送入transformer之前使用了一个操作**Bias Encoding**.

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

用了三个矩阵分别对session本身， session中的每个位置， 每个位置处id的embedding的每个维度都加上了一个偏置项。  
源码如下：

```
self.sess_bias_embedding = self.add_weight('sess_bias_embedding', shape=(self.sess_max_count, 1, 1),
                                           initializer=TruncatedNormal(
                                               mean=0.0, stddev=0.0001, seed=self.seed))
self.seq_bias_embedding = self.add_weight('seq_bias_embedding', shape=(1, seq_len_max, 1),
                                           initializer=TruncatedNormal(
                                               mean=0.0, stddev=0.0001, seed=self.seed))
self.item_bias_embedding = self.add_weight('item_bias_embedding', shape=(1, 1, embed_size),
                                           initializer=TruncatedNormal(
                                               mean=0.0, stddev=0.0001, seed=self.seed))
```

<https://deep-rec.blog.csdn.net>

## 2.2.2 会话兴趣抽取层

然后将偏置项加入到输入的session list中

$$Q = Q + BE$$

```
for i in range(self.ssess_max_count):
    transformer_out.append(
        inputs[i] + self.item_bias_embedding + self.seq_bias_embedding + self.ssess_bias_embedding[i])
return transformer_out
```

<https://deep-rec.blog.csdn.net>

这一步应该算是对原始transformer中position encoding的优化，利用偏置项来区分不同位置session，不同位置的item，以及不同位置的embedding值。

## 2.2.2 会话兴趣抽取层

接着把经过**Bias Encoding**处理的输入session list传入tranformer结构里：

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

<https://deep-rec.blog.csdn.net>

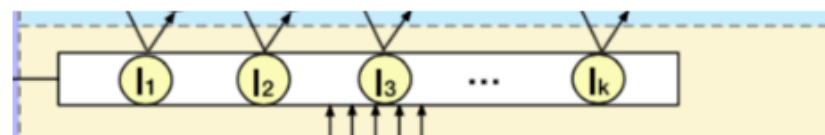
tranformer输出的结果被再次输入一个前向网络里面做了一层映射

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

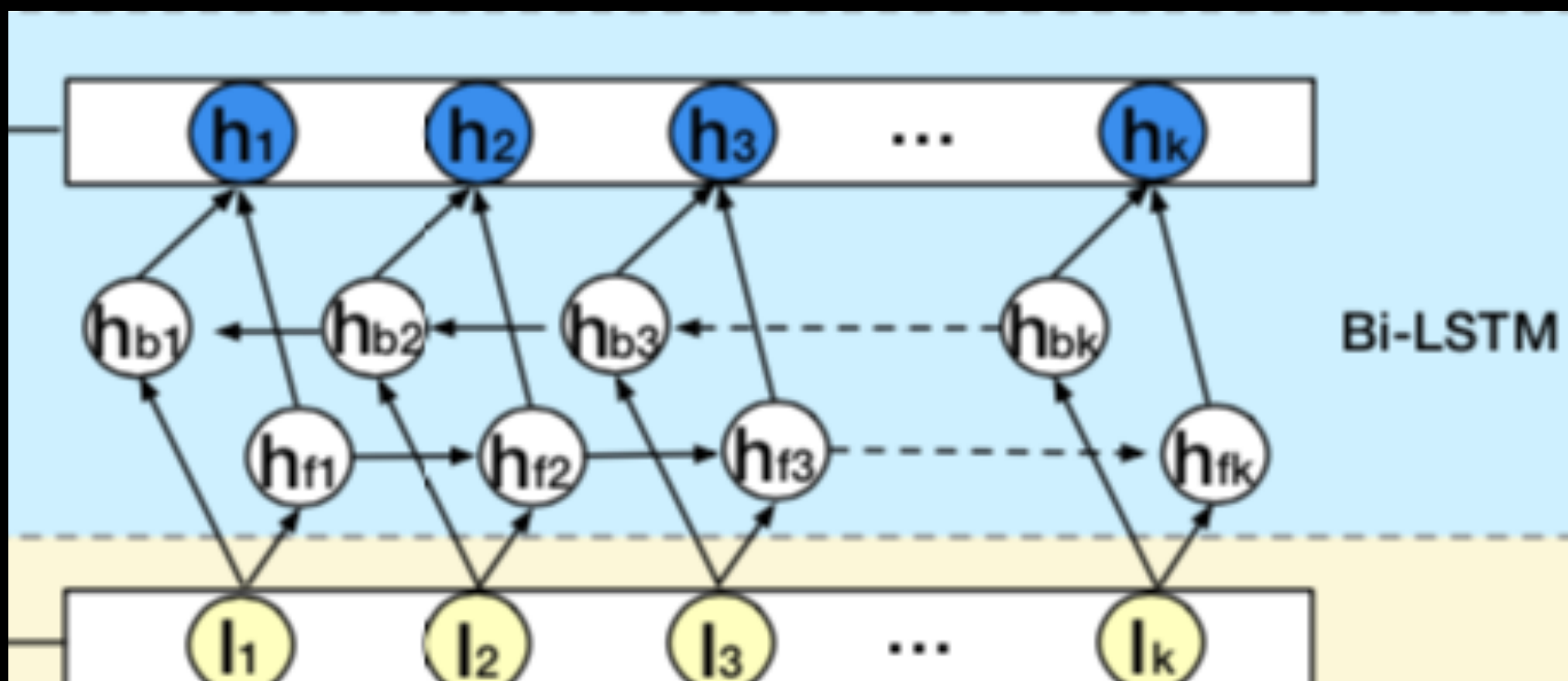
然后再用average pooling把每个session的维度进行压缩：

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

到这为止，或得到了每个session的一个内在表示，就是图里面的：



## 2.2.3 会话间兴趣交互层



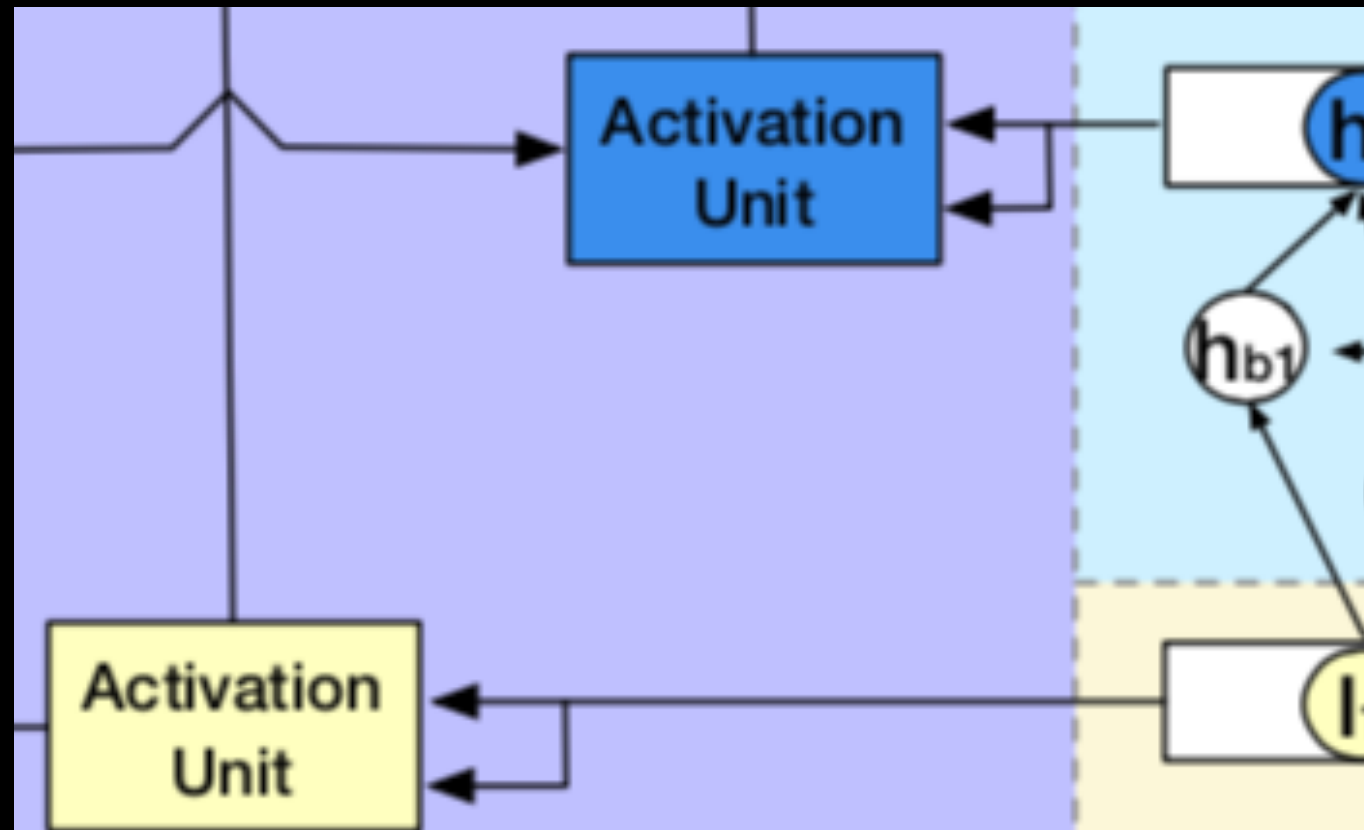
文章为了捕获不同session之间的顺序关系，使用了Bi-LSTM。Bi-LSTM是双向的，可以同时捕获上下文关系。

因此经过Bi-LSTM编码的输入，每个维度的输出向量其实都包含了输入数据同一位置的前后信息。这步获得的数据是图中的：



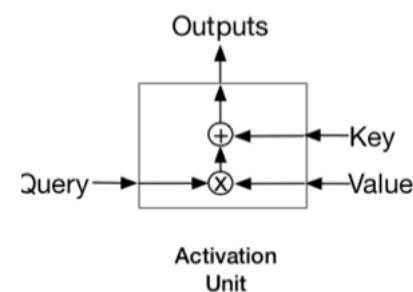
到这为止，模型已经同时捕获到了session内部和session之间的顺序关系。如果想简单一点，直接把这两者的输出结果和图中左侧的画像特征concat起来也可以。不过文章作者在concat前对两者的输出做了一层attention，用来判断session信息和目标item之间的相关性。

## 2.2.4 会话兴趣激活层



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

<https://deep-rec.blog.csdn.net>



<https://deep-rec.blog.csdn.net>

attention的query就是公式中的X，就是目标item的embedding。item的embedding是item画像特征所有embedding一起concat起来获得的。value和key就是前面获得两个输出I和H。

最后把以上这些向量都组合起来送入DNN中进行训练。

# 3. 实验与结果



Model	Advertising	Recommender
YoutubeNet-NO-UB <sup>a</sup>	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 <sup>b</sup>	0.6357	0.6494
DSIN-BE-NO-SIIL <sup>c</sup>	0.6365	0.6499
<b>DSIN-BE<sup>d</sup></b>	<b>0.6375</b>	<b>0.6515</b>

- 模型使用了两个数据集进行实验，分别是阿里妈妈的广告数据集和阿里巴巴的电商推荐数据集
- 对比模型如图，评价指标是AUC
- 对于DSIN，分三种：1、将Bias Encoding变为Transformer里的Positional Encoding；2、使用BE，但不添加会话兴趣提取层和相应的activation unit。3、前面提到的DSIN

Model	Advertising	Recommender
YoutubeNet-NO-UB <sup>a</sup>	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 <sup>b</sup>	0.6357	0.6494
DSIN-BE-NO-SIIL <sup>c</sup>	0.6365	0.6499
<b>DSIN-BE<sup>d</sup></b>	<b>0.6375</b>	<b>0.6515</b>

- Effect of Multiple Sessions 序列建模
- Effect of Session Interest Interacting Layer
- Effect of Bias Encoding
- Visualization of Self-attention and the Activation Unit

## 4. 未来研究方向

- 利用知识图作为先验知识来解释用户的历史行为以获得更好的CTR

谢谢