Attention 机制在推荐算法中的应用 | 深度兴趣网络(DIN)算法介绍及浅析（文末招聘！）

原创 张发波 金科优源汇 1周前

点击上方“蓝字”关注我们

作者 | 张发波
编辑 | 张婵

> 2020 年回看最近一两年的 CTR（Clicked Through Rate，点击率）预估算法论文就会发现，这两年新提出的一系列 CTR 预估算法都能看到 attention 的影子，足以见得 attention 逐渐成为 CTR 预估算法的一个标配。

> 其中比较有代表性的是由盖坤领导的阿里妈妈精准定向检索及基础算法团队于 2018 年提出的 DIN（Deep Interest Network，深度兴趣网络）算法（论文下载地址：https://arxiv.org/pdf/1706.06978.pdf。），该算法充分利用/挖掘用户历史行为数据中的信息来提高 CTR 预估的性能。DIN 算法的提出给 CTR 预估领域带来了新的研究思路，本文将对这一算法进行介绍，并做一些初步的解析，供大家参考。

1. 论文背景

在工业界 CTR 预估领域中，用户的历史行为特征（如最近一周的浏览商品、最近一周的点击商品等）是一类非常重要的特征，能够有效地刻画用户的兴趣和行为偏好。如何最大程度地利用丰富的用户历史行为数据进行精准的 CTR 预估一直以来都是推荐算法领域一个非常重要的研究方向。

纵观最近几年 CTR 预估算法的发展，不难发现对于用户历史行为的挖掘已形成一套相对固定的基本范式，即：通过 embedding 层，将高维离散特征转换为固定长度的连续特征，然后通过多个全联接层，最后通过一个 sigmoid 单元转化为点击概率，即 sparse features -> embedding vector -> MLPs -> sigmoid -> output。

比较典型的 CTR 预估算法，例如 Wide&Deep，DeepFM，xDeepFM 等，均借鉴了这一范式的核心思想。这一类方法的优点在于：通过神经网络可以拟合高阶的非线性关系，同时减少了人工特征的工作量。

面对众多的用户历史行为特征，如何处理这些特征的 embedding 向量呢？通常有两种做法：

- 一种是直接把这些向量 concate 起来，这样可以保证每个 embedding 的信息都被保留下来；

- 还有一种就是利用 pooling 将多个 embedding 向量进行压缩，这种方法无疑会造成一定程度的信息丢失，常用的 pooling 方法包括 sum-pooling 和 average-pooling，这两种方法还有一个问题就是这些 embedding 向量的权重都是相同的，也就是认为不同的用户行为特征对 CTR 预估任务的重要性是相同的，但是事实可能并非如此。

阿里巴巴的研究者们通过观察收集到的线上数据，发现了用户历史行为数据中有两个很重要的特性：

- Diversity：用户在浏览电商网站的过程中显示出的兴趣是十分多样性的。
- Local activation: 由于用户兴趣的多样性，只有部分历史数据会影响到当次推荐的物品是否被点击，而不是所有的历史记录。

这两种特性是密不可分的。举个简单的例子，观察下面的表格：

Table 1: Examples of user behavior history from online product.

User	Behavior History	Candidate Ad
Young Mother	woolen coat, T-shirts, earrings,children's coat leather handbag, miniskirt, sports underwear	long sleeved jacket
Swimmer	bathing suit, kickboard, swimming cap, travel book tent, potato chips, nuts, potato chips, ice cream	goggle

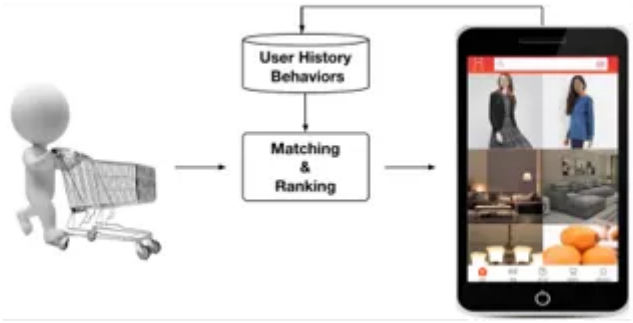
Diversity 体现在年轻的母亲的历史记录中体现的兴趣十分广泛，涵盖羊毛衫、手提袋、耳环、童装、运动装等等。而爱好游泳的人同样兴趣广泛，历史记录涉及浴装、旅游手册、踏水板、马铃薯、冰激凌、坚果等等。Local activation 体现在，当我们给爱好游泳的人推荐 goggle (护目镜) 时，跟他之前是否购买过薯片、书籍、冰激凌的关系就不大了，而跟他游泳相关的历史记录如游泳帽的关系就比较密切。

基于上述观察，盖昆团队在 DIN 中引入了 local-activation 机制（对应于 NLP 中的 attention 机制）来有侧重的利用不同的用户行为特征；同时为了解决模型训练中存在的一些问题，又提出了 mini-batch aware 正则和自适应激活函数来辅助模型进行训练；模型评估方面，盖昆团队创新地采用了 GAUC 这一指标，而不是常规的 AUC，来消除了用户本身的差异。

2. 模型设计

2.1 整体框架

我们先来看一下阿里巴巴屏幕广告线上系统的整体框架：



阿里巴巴屏幕广告线上系统的整个运行流程可以描述为：

1. 检查用户历史行为数据。
2. 使用 matching module 产生候选 ads。
3. 通过 ranking module 做 point-wise 的排序，即得到每个候选 ads 的点击概率，并根据概率排序得到推荐列表。
4. 记录下用户在当前展示广告下的反应(点击与否)，作为 label。

2.2 特征设计

论文将所涉及到的特征分为四个部分：用户特征、用户行为特征、广告特征、上下文特征，具体如下：

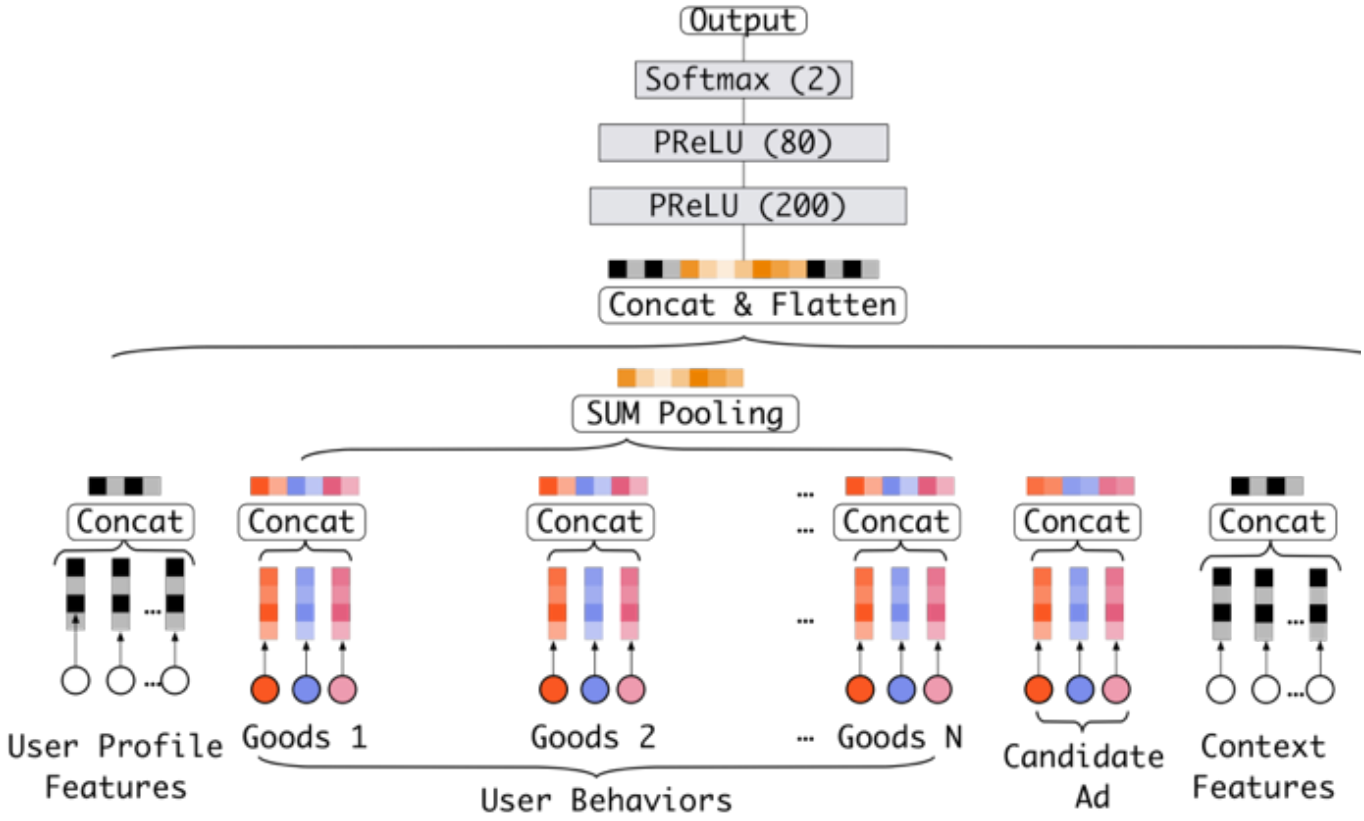
Table 2: Feature Representations and Statistics in our display advertising system.

Feature Category	Feature Name	Dimension	Type	#Nonzero Ids/Sample
User Profile Features	gender	2	one-hot	1
	age_level	~ 10	one-hot	1
	...	...	...	...
User Behavior Features	visited good_ids	$\sim 10^9$	multi-hot	$\sim 10^3$
	visited shop_ids	$\sim 10^7$	multi-hot	$\sim 10^3$
	visited cate_ids	$\sim 10^4$	multi-hot	$\sim 10^2$
	...	...	...	...
Ad Features	good_id	$\sim 10^7$	one-hot	1
	shop_id	$\sim 10^5$	one-hot	1
	cate_id	$\sim 10^4$	one-hot	1
	...	...	...	...
Scene Features	pid	~ 10	one-hot	1
	time	~ 10	one-hot	1
	...	...	...	...

其中，用户行为特征是 multi-hot 的，即多值离散特征。针对这种特征，由于每个涉及到的非 0 值个数是不一样的，常见的做法就是将 id 转换成 embedding 之后，加一层pooling层，比如average-pooling，sum-pooling，max-pooling。DIN中使用的是weighted-sum，其实就是加权的sum-pooling，权重经过一个activation unit 计算得到，其中技术细节我们后面还会展开介绍。

2.3 基准模型

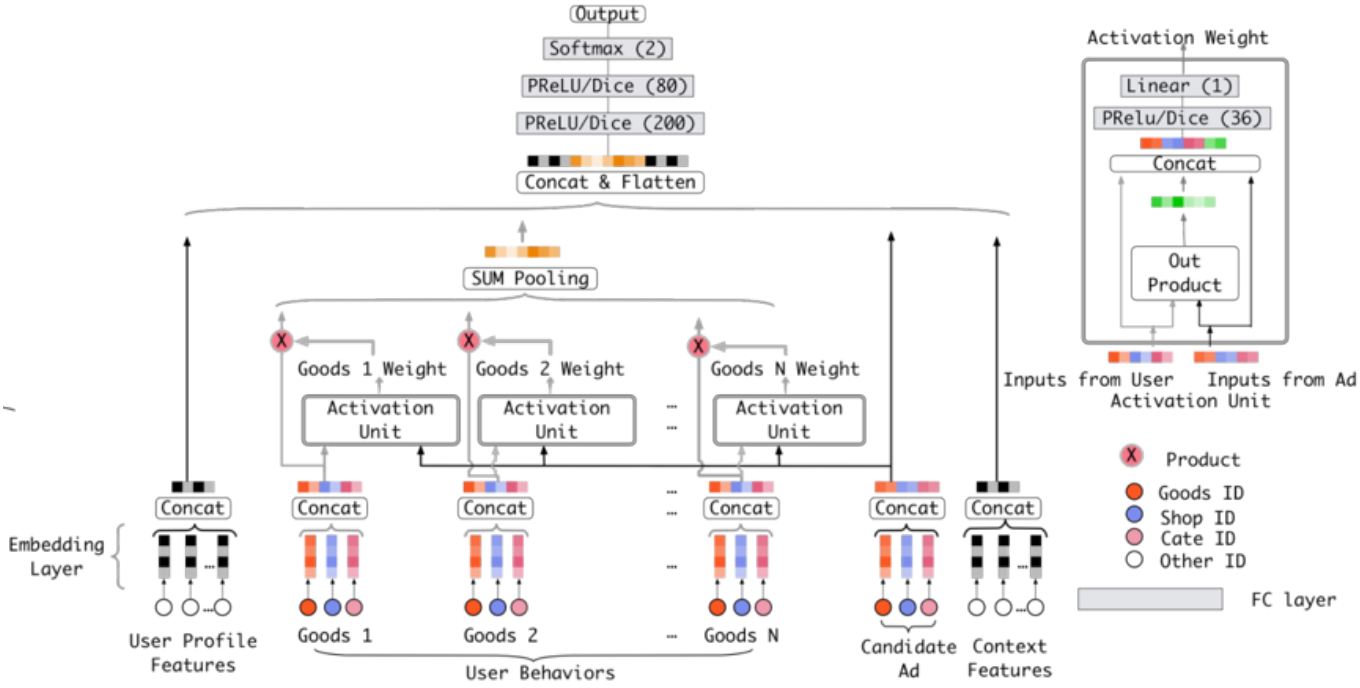
在介绍 DIN 之前，我们先来看一个基准模型。这个基准模型就是目前比较常见的多层神经网络，即：先对特征进行 embedding 操作，得到一系列 embedding 向量之后，将不同 group 的特征 concate 起来之后得到一个固定长度的向量，然后将此向量喂给后续的全连接网络，最后输出 pCTR 值，具体网络结构如下：



可以看到，针对 multi-hot 的特征，基准模型特意地做了一次 element-wise+ 的操作，这一操作其实就是 sum-pooling。这样一来，不管特征中有多少个非 0 值，经过转换之后的 embedding 特征向量长度都是一样的！

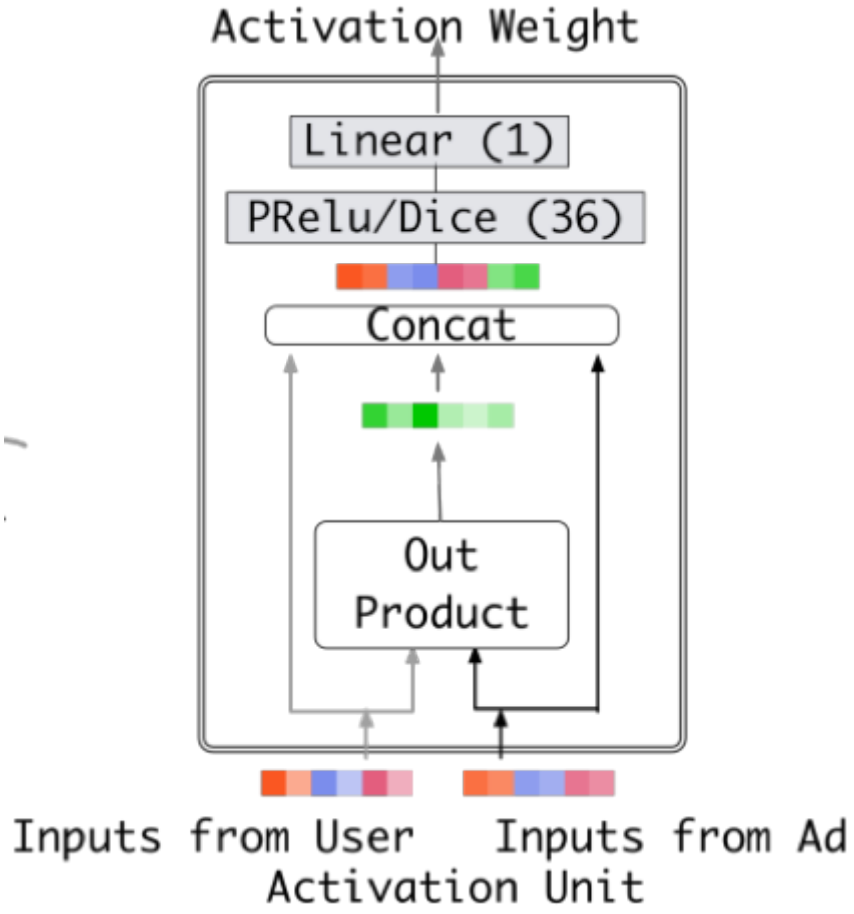
2.4 DIN模型

DIN 模型整体的网络结构如下图所示：



对比 DIN 模型和基准模型，我们可以发现在模型结构层面上两者的差别主要体现在如何聚合多个用户历史行为所对应的 embedding 向量上，基础模型直接对多个 embedding 向量进行等权的 sum-pooling，这种方法肯定会带来信息的丢失，而且相对重要的 embedding 向量也无法完全突出自己所包含的信息。所以，DIN 采取了一个比较直接的方式，就是 weighted-sum pooling，而 attention 的本质就是 weighted-sum pooling，即让模型更加关注有用的信息。对于 attention 机制不太熟悉的小伙伴，可以参考 Google论文《Attention is All You Need》（论文下载地址：https://arxiv.org/abs/1706.03762。）。

在论文当中，attention 权重的计算方式是利用用户各个历史行为的 embedding 向量和广告的 embedding 向量，通过一个激活单元（activation unit）来分别计算得到用户各个历史行为的 embedding 向量的权重。激活单元所对应的网络结构如下图所示：



激活单元的网络结构

可以看到，激活单元的输入包括三个部分：

- 第一个是原始的用户历史行为 embedding 向量。
- 第二个是原始的广告 embedding 向量。
- 最后一个是上述两个 embedding 向量经过外积运算后得到的向量。关于最后一个输入向量的作用，论文给出的解释是有利于模拟用户各个历史行为与广告之间的关系。

需要指出的是，DIN 模型所采用的 attention 机制并不是 NLP 任务中所惯常采用的 attention 机制，DIN 放宽了对于各个 attention 权重之和等于 1 的限制。关于这一点，论文中也做出了解释：的取值某种程度上可以当作被激活的用户兴趣强度的近似，传统的 attention 机制通过放缩操作施加归一化限制会造成这一部分信息的损失。

3. 其他改进

3.1 自适应正则化（Mini-batch Aware Regularization）

CTR 中输入稀疏而且维度高，通常的做法是加入 L1、L2、Dropout 等防止过拟合。但是论文中尝试后效果都不是很好。用户数据符合长尾定律 long-tail law，也就是说很多的 feature id 只出现了

几次，而一小部分 feature id 出现很多次。这在训练过程中增加了很多噪声，并且加重了过拟合。

对于这个问题一个简单的处理办法就是：直接去掉出现次数比较少的 feature id。但是这样就人为的丢掉了一些信息，导致模型更加容易过拟合，同时阈值的设定作为一个新的超参数，也是需要大量的实验来选择的。因此，论文中提出了自适应正则的做法，即：

1. 针对 feature id 出现的频率，来自适应的调整他们正则化的强度；
2. 对于出现频率高的，给与较小的正则化强度；
3. 对于出现频率低的，给予较大的正则化强度。

参数更新计算公式如下：

$$w_j \leftarrow w_j - \eta \left[\frac{1}{|B_m|} \sum_{(x,y)\in B_m} \frac{\partial L\left(p(x), y\right)}{\partial w_j} + \lambda \frac{\alpha_{mj}}{n_j} w_j \right]$$

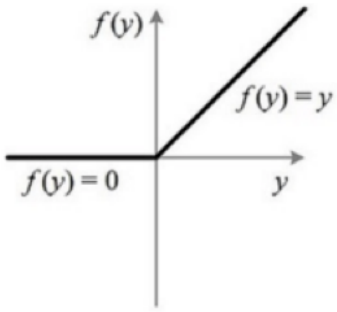
其中

$$\alpha_{mj} = \max_{(x,y)\in B_m} I\left(x_j \neq 0\right)$$

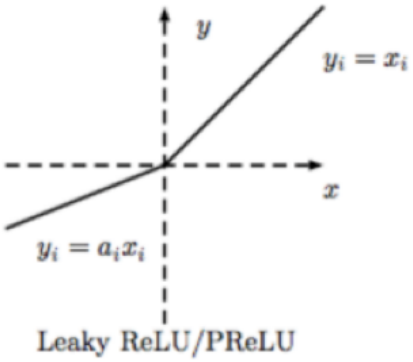
3.2 Dice激活函数

- **3.2.1 从 Relu 到 PRelu**

Relu 激活函数形式如下：



Relu 激活函数在值大于 0 时原样输出，小于 0 时输出为 0。这样的话导致了许多网络节点的更新缓慢。因此，后面又出现了 PRelu，也叫 Leaky Relu，形式如下：



这样，即使值小于 0，网络的参数也得以更新，加快了收敛速度。

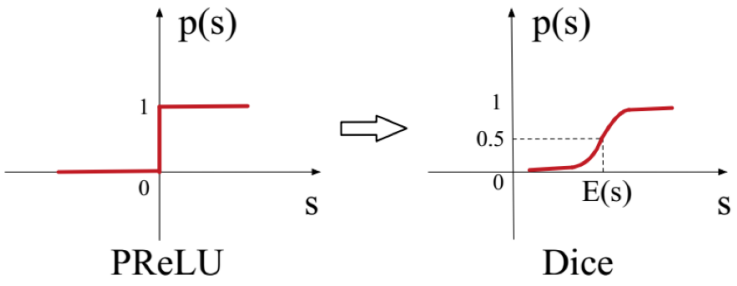
- **3.2.2 从 PReLU 到 Dice**

尽管对 Relu 进行了修正得到了 PRelu，但是仍然有一个问题：即我们认为分割点都是0。但实际上，分割点应该由数据决定。因此，论文中提出了 Dice 激活函数，Dice 激活函数的全称是 Data Dependent Activation Function，其形式如下：

$$f(s) = \begin{cases} s, & 当s > 0时 \\ \alpha s, & 当s \le 0时 \end{cases} = p(s) \cdot s + (1 - p(s)) \cdot \alpha s$$

$$p(s) = \frac{1}{1 + \exp\left(-\frac{s - E(s)}{\sqrt{Var(s) + \varepsilon}}\right)}$$

p 函数曲线如下所示：



论文认为采用 PRelu 作为激活函数时，它的 rectified point 固定为 0，这在每一层的输入分布发生变化时是不适用的。所以，论文针对该激活函数进行了改进，一方面平滑了激活函数在 rectified point 附近的曲线，另一方面激活函数会根据每层输入数据的分布来自适应调整 rectified point 的位置，从而对模型参数的更新和收敛起到一定的加速作用。

3.3 评估指标 GAUC

模型使用的评价指标是 GAUC，我们先来看一下 GAUC 的计算公式：

$$GAUC = \frac{\sum_{i=1}^{n} 对应客户i的曝光物品数量 \times 模型对于客户i的AUC}{\sum_{i=1}^{n} 对应客户i的曝光物品数量}$$

我们首先要肯定的是，AUC 是要分用户看的，我们的模型的预测结果，只要能够保证对每个用户来说，他想要的结果排在前面就好了。

假设有两个用户 A 和 B，每个用户都有 10 个商品，10 个商品中有 5 个是正样本，我们分别用 TA，TB，FA，FB 来表示两个用户的正样本和负样本。也就是说，20 个商品中有 10 个是正样本。假设模型预测的结果大小排序依次为 TA，FA，TB，FB。如果把两个用户的结果混起来看，AUC 并不是很高，因为有 5 个正样本排在了后面，但是分开看的话，每个用户的正样本都排在了负样本之前，AUC 应该是 1。显然，分开看更容易体现模型的效果，这样消除了用户本身的差异。

但是上文中所说的差异是在用户点击数即样本数相同的情况下说的。还有一种差异是用户的展示次数或者点击数，如果一个用户有 1 个正样本，10 个负样本，另一个用户有 5 个正样本，50 个负样本，这种差异同样需要消除。那么 GAUC 的计算，不仅将每个用户的 AUC 分开计算，同时根据用户的展示数或者点击数来对每个用户的 AUC 进行加权处理。进一步消除了用户偏差对模型的影响。实验证明，GAUC 确实是一个更加合理的评价指标。

4. 实验结果

4.1 Attention 结果展示

下图是对 Local Activation 效果的一个展示，可以看到，对于候选的广告是一件衣服的时候，用户历史行为中跟衣服相关的权重较高，而非衣服的部分，权重较低。

4.2 离线测试

为了评估 DIN 模型的效果，论文在公开数据集上进行了大量且细致的离线实验。下图是对使用不同正则项的结果展示，可以发现，在使用自适应正则的情况下，模型的验证集误差和验证集 GAUC 均是最好的。

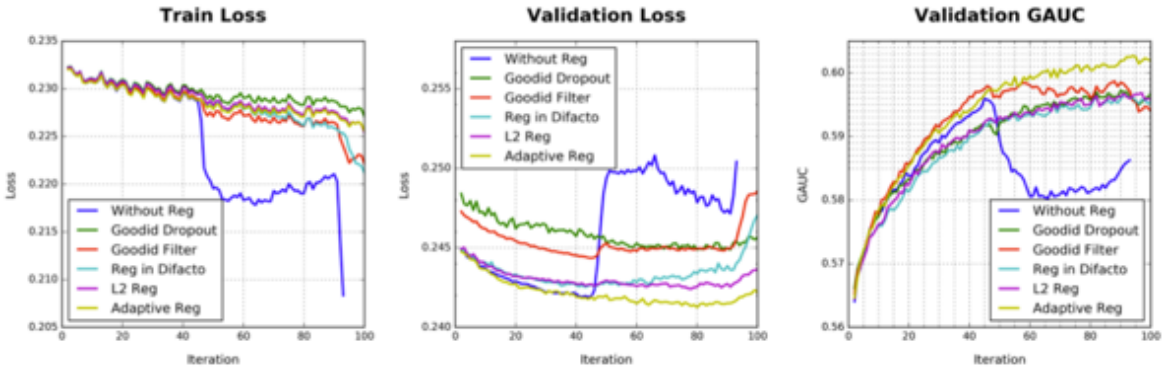


Figure 6: Performance of reduction of overfitting with different regularizations.

下图对比了基准模型和 DIN 模型的实验结果，可以看到，DIN 模型在加入 Dice 激活函数以及自适应正则之后，模型的效果有了一定的提升。

Table 3: Comparison of model performance.

	GAUC	GAUC gain on Base
Base Model	59.59%	0.0%
Base Model with Drop out	59.70%	0.11%
Base Model with adaptive_reg	60.31%	0.72%
DIN Model with adaptive_reg	60.60%	1.01%
DIN Model with adaptive_reg and Dice	60.83%	1.24%

Figure 7: Performance of DIN and Base Model.

4.3 在线测试

因为该模型最终是要部署到线上真实生产环境中的，论文中也给出了 DIN 模型的线上 AB 实验表现：相较于基础模型，DIN 模型表现十分突出，线上 CTR 增长 10%，RPM（Revenue Per Mille，千次展示收入）增长 3.8%。在 CTR 预估领域，百分之零点几的CTR增长都会带来巨大的

业务增长。所以，上述结果足以证明 DIN 模型的效果还是非常出色的，该模型也成为了阿里妈妈新一代的 CTR 预估模型。

5. 总结

最后，对这篇论文做一点总结。这篇论文无疑是非常优秀的，一篇优秀的论文多强调几遍也不为过。说这篇论文好，主要有三个原因：

- 第一个原因是因为这篇论文的工程性很强。工程性很强的论文首先是便于实现的，其次你可以从字里行间看到很多实践出真知的影子，比如 DIN 这篇论文中 GAUC 这样的 metric 的改进，以及 Dice 这样的激活函数的创新，都是对经典知识在实践中改进的例子。
- 第二个原因是因为这篇论文对用户行为的观察非常精准。有句话说做推荐其实就是“揣摩人心”，你把用户的行为和习惯揣摩好了，才能够以此出发，从技术上映射用户的习惯。DIN 这篇论文有效的利用了用户兴趣多样性以及当前候选商品仅与用户一部分兴趣有关这一特点，引入注意力机制，这是非常精准的动机。
- 第三个原因是模型的微创新，从低维到高维是创新，从离散到连续是创新，从单一到融合也是创新，这篇论文把 NLP 大行其道的注意力机制引入推荐领域，当然是典型并且有效的创新手段，也是所有算法工程师应该学习的地方。