DIN原理与实践

# 背景

2018年之前，一些基于深度学习的模型被应用在点击率预估任务当中。这些模型采用了类似的embedding & MLP的形式，被称为CTR预估的Base Model（Embedding + MLP）。通常是把大规模高维稀疏输入特征首先被映射到低维Embedding向量中，然后转换为固定长度向量【采用sum pooling或avg pooling】，最后将这些定长的向量连接在一起【concatenate】输入到多层感知机（MLP）以学习特征之间的非线性关系。与逻辑回归模型对比，确实减少了很多的特征工程和增强了模型的能力。常见的模型有Wide&Deep、DCN、PNN、DeepFM等。这种方法相对传统机器学习有较好的效果提升，但是在其还是存在一些缺点：

* 用户的兴趣通常是多种多样的，而 Embedding & MLP 方法中有限的向量维度会成为用户多样化兴趣的瓶颈，如果扩大向量维度会极大地增加学习参数和计算负荷，并增加过拟合风险；
* 不需要将用户的所有兴趣都压缩到同一个向量中。比如说：用户购买了泳镜并不是因为上周购买了鞋子，而是因为之前购买了泳衣；

Embedding+MLP将用户特征转化为了一个定长的向量，这个定长的向量成为了瓶颈，会制约模型捕捉多变的用户兴趣。例如，在电子商务中，用户浏览电商网站时可以同时对多个不同的物品产生兴趣。在CTR预估问题中，就是从用户的浏览历史中去捕获用户的兴趣。而该方法不管候选广告是什么，都是将多个特征向量【Embedding】压缩到一个固定长度的表示向量来学习特定用户所有的兴趣表示，这限制了模型的能力，很难从历史行为中提取用户变化的兴趣。解决这个问题最简单的方法就是扩展向量的维度，但这样会增加学习的参数和在有限的数据中有过拟合的风险。

低维的定长向量难以表示大量的不同的的用户兴趣，但是如果用定长向量（扩展向量的维度）表示所有的用户兴趣，那么将这个向量的维度会变得特别大，这会导致要学习的参数特别多，加大过拟合风险并且会带来储存方面的困难。另一方面在预测一个候选集时，对于一个用户，并不需要将他所有的兴趣【用户的历史购买记录】放在同一个向量之中。因为只有用户部分的兴趣会影响当前行为（对候选广告点击或不点击）。例如，一位女游泳运动员会点击推荐的护目镜，这主要是由于购买了泳衣而不是上周购物清单中的鞋子。DIN模型更关注和候选集相关的用户历史行为。

受到上述启发，作者提出了Deep Interest Network模型，它通过考虑给定的候选广告和用户的历史行为的相关性，来计算用户兴趣的表示向量。具体来说就是通过引入局部激活单元，通过软搜索历史行为的相关部分来关注相关的用户兴趣，并采用加权和来获得有关候选广告的用户兴趣的表示。与候选广告相关性较高的行为会获得较高的激活权重，并支配着用户兴趣。该表示向量在不同广告上有所不同，大大提高了模型的表达能力。

训练具有大规模稀疏特征的工业深度网络是巨大的挑战。例如，基于SGD的优化方法仅更新出现在每个小批量中的稀疏特征【非零】的那些参数，因此需要加入正则化来降低过拟合的风险。但是，加上传统的l2正则化后，计算量过大，这需要为每个小型批处理在整个参数上计算L2范数。因此本文提出了一种工业技术：mini-batch aware regularization，仅出现在每个微型批处理中的非零特征参数才参与L2-范数的计算，从而使计算可接受。

另外还提出data adaptive activation function，通过输入的分布自适应调整修正点来推广常用的PReLU。

概括上述创新点，文章的贡献为：

* 指出使用固定向量来表示用户不同的兴趣的限制性和通过引入局部激活单元建立了一个新的模型DIN。
* 提出两个训练工业神经网络的技术：小批量感知正则化器（a mini-batch aware regularizer），它可以节省具有大量参数的深度网络上正则化的大量计算，并且有助于避免过度拟合；数据自适应激活函数（a data adaptive activation function），它通过考虑输入的分布来概括PReLU，并显示出良好的性能。
* 在对公共和Alibaba数据集进行了广泛的实验。结果证实了提出的DIN的有效性。

《Deep Interest Network for Click-Through Rate Prediction》

针对这些问题，DIN 模型通过考虑给定候选广告的历史行为的相关性，自适应地计算用户兴趣的表示向量。通过引入局部激活单元，DIN 模型通过软搜索历史行为的相关部分来关注相关的用户兴趣，并采用加权总和池化来获取有关候选广告的用户兴趣的表示形式。与候选广告相关性更高的行为会获得更高的激活权重，并且支配着用户兴趣。这样用户的兴趣表示向量就会随着广告的不同而变化，从而提高了模型在有限尺寸下的表达能力，并使得模型能够更好地捕获用户的不同兴趣。

训练具有大规模稀疏特征网络具有非常大的挑战，例如：基于 SGD 的优化方法可以采用 Mini-Batch 来更新参数，但加上 L2 正则化后其计算量会非常大，因为每个 Mini-Batch 都需要计算所有参数的 L2 范式；作者提出了一个新颖的 Mini-Batch 感知正则化方法，可以只计算非零特征参数的 L2 范式；此外，作者还考虑数据输入分布，从而设计邻一个能够自适应数据的激活函数。

# 简介

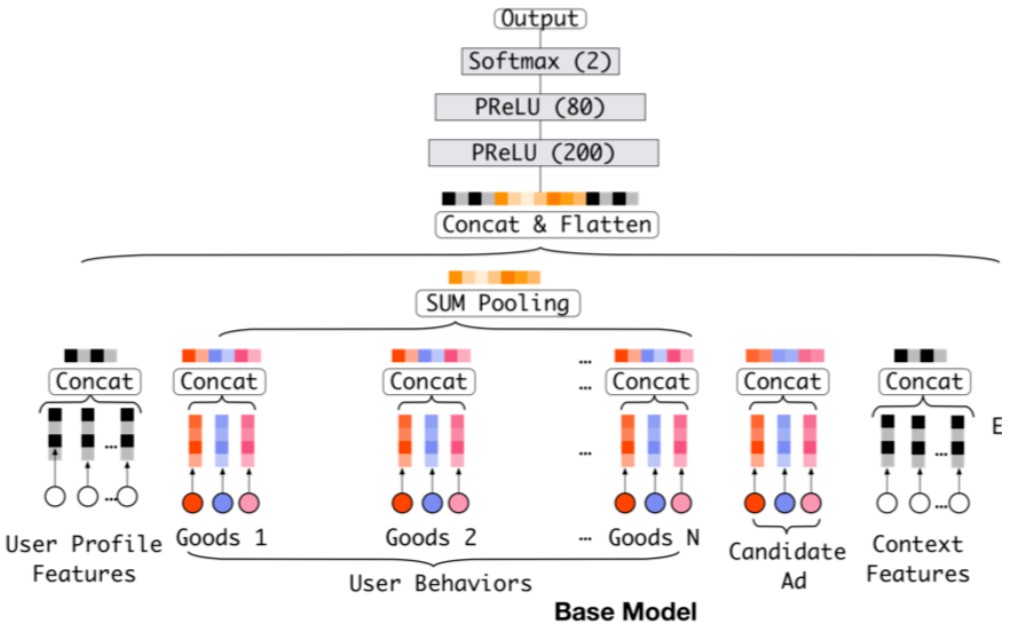
阿里2018在KDD上发表的论文《Deep Interest Network for Click-Through Rate Prediction》。文章的 核心就是使用一个局部激活单元（类似Attention机制）来提高与候选广告相关的历史信息的权重。

DIN模型的特点是引入了attention机制，充分挖掘用户历史行为序列中的信息。

除此之外，本文还提出了novel mini-batch aware regularization和新的激活函数PReLU ，新的正则化方法能够减小计算量，PReLU 考虑了输入数据的分布情况。

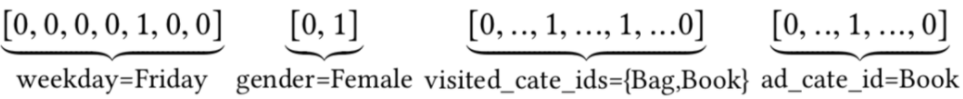
# Base Model

先看看Base Model，其网络结构主要由 Embedding 和 MLP 构成，如下图所示：

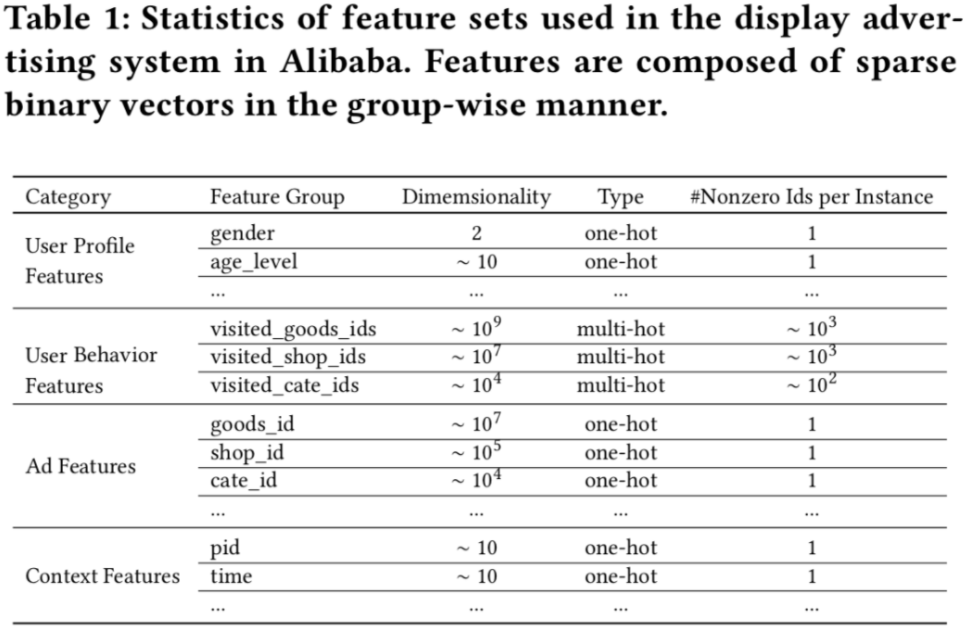


## 特征表示（Feature Representation）

工业点击率预测任务中的数据大多采用多组类别的形式，如下图所示，这通常要通过转化为高阶稀疏二元特征【one-hot或者multi-hot】。



文中用到的特征分为4部分：用户画像特征、用户行为特征、广告特征、上下文特征。所有的特征没有进行交叉，我们将用DNN捕捉交互特征。



## Embedding layer

输入是高维稀疏二元向量，Embedding层将其转化为低维密集表示。

对于第i个特征组t\_i，用表示第i个embedding字典，表示第i个特征组的维度。是D维度的embedding向量。Embedding操作后伴随着一个表的查找机制：

* 如果t\_i是一个one-hot向量，则t\_i的embedding表示是一个单一的embedding向量
* 如果t\_i是一个multi-hot向量，那t\_i的embedding表示是一个embedding向量列表

## Pooling layer和Concat layer

不同的用户有不同数量的历史行为，这就导致用户历史行为特征的数量是可变的，而全连接层只能接受定长的输入。比较常见的做法是采用池化层将变长的输入转化为固定长度。不同的用户有不同数量的历史行为，即multi-hot行为特征的向量会导致所产生的embedding向量列表的长度不同，而全连接需要固定长度的输入。比较常见的做法是采用池化（Pooling）层将变长的输入（embedding向量列表）转化为固定长度：



两个最常用的池化层是求和池化（sum pooling，各个对应元素进行累加）和平均池化（average pooling，各个对应元素求平均）。

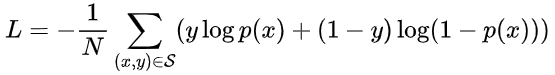
嵌入层和池化层共同将原始稀疏特征转化为固定长度的表示向量。然后将所有向量连接在一起（concatenate），以获得实例的总体表示向量。

## MLP

给出连接后的稠密表示向量，利用全连接层自动学习特征的组合。设计MLP的结构，以更好地提取信息。

## LOSS

负对数似然函数（the negative log-likelihood function）：



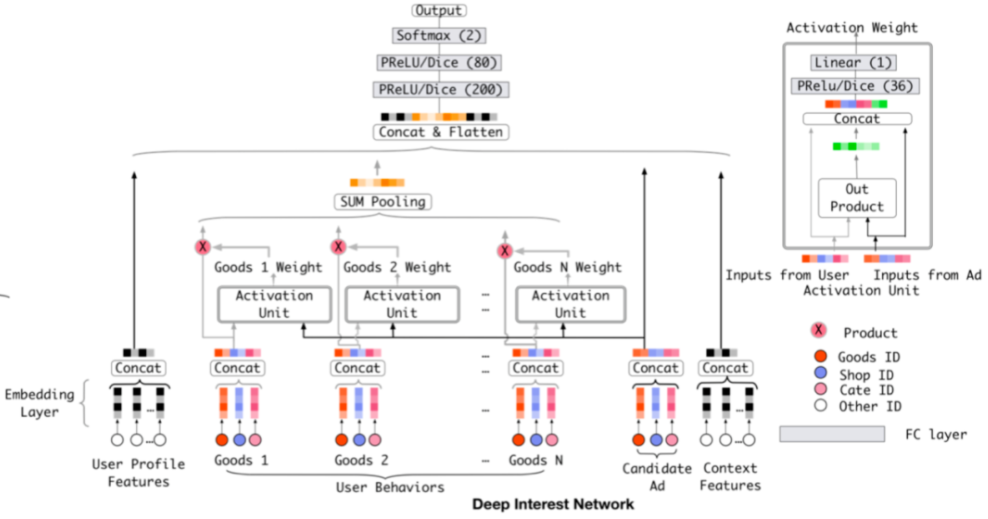
其中S代表训练的集合，p(x)为模型的输出，即点击候选广告x的概率。

一言以蔽之，在Base Model中，User Behaviors代表的就是用户的行为序列，传统的做法是先将稀疏的行为序列转换成embedding，然后再经过一层sum/average pooling后得到一个embedding，这个embedding就可以表征用户的多个兴趣了。

Base Model 有一个非常大的缺点——用户兴趣多样化表达受限。Base Model 直接通过池化所有的 Embedding 向量来获的一个定长的表示向量。由于表示向量长度固定，所以其表达也会受到一定的限制，假设最多可以表示 k 个独立的兴趣爱好，如果用户兴趣广泛不止 k 个则表达受限，极大的限制了用户兴趣多样化的表达，而如果扩大向量维度则会带来巨大的计算量和过拟合的风险；

# Deep Interest Network

假如我们想要知道用户是否对一件羽绒服感兴趣，是不是应该把注意力放在这个用户是否购买过类似的衣服上面？假如我们想要知道用户是否对iphone感兴趣，是不是应该把更多注意力放在用户是否买过类似数码产品身上？基于这种思想，paper作者借鉴attention的思想，提出了DIN网络，如下图所示：



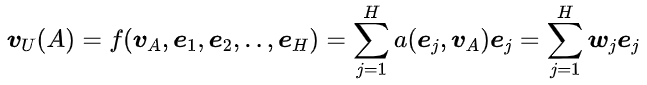
可以看到DIN和传统的模型区别在于引入了一个Activation Unit，这个Activation Unit输入是用户的历史行为的item和候选Ad item，输出是一个weight。这个weight代表要把多大的注意力放在这个item上面。

利用得到的weight乘上对应的物品embedding，然后做weight sum pooling，就可以针对不同的候选item，同一个用户生成不一样的兴趣embedding了。

Base Model是获得一个固定长度的用户的表示向量，但不管候选广告是什么，此表示向量对于给定用户均保持不变。这样，维度受限的用户表示向量将成为表达用户多样化兴趣的瓶颈。

与展示广告相关的行为极大地影响了点击操作。DIN通过给定一个候选广告，然后去注意与该广告相关的局部兴趣的表示来模拟此过程。DIN不会通过使用同一向量来表达所有用户的不同兴趣，而是通过考虑历史行为的相关性来自适应地计算用户兴趣的表示向量（对于给定的广告）。该表示向量随不同广告而变化。

相比于Base Model，DIN引入了一种新颖设计的局部激活单元，其他部分和base model完全一样。具体而言，将激活单元应用于用户历史行为特征中，将其作为池化层的权重来计算用户表示向量，以在给定候选广告的情况下自适应地计算用户表示，其公式如下：



其中{e1, e2, ..., eH }是用户U长度为H的embedding向量列表，是广告的embedding向量。a(·)是一个前向传播网络，输出为激活权重。

局部激活单元与attention机制类似。不同的是，的约束被放宽了，这是为了保证用户兴趣的强度。相比传统的attention机制，在a(·) 输出后的用于归一化的softmax函数也舍弃了。例如，如果一个用户的历史行为包含90％的衣服和10％的电子产品。给定T恤和电话的两个候选广告，T恤会激活大多数属于衣服的历史行为，并且可能比电话获得更大的（更高的兴趣强度）价值。传统的注意力方法是通过对的输出进行归一化而失去的数值规模的决心。

作者尝试了LSTM对用户历史行为数据进行序列化建模。但是效果并不好。因为用户历史行为的序列可能同时包含很多同时存在的兴趣。这些兴趣的快速跳跃和突然结束导致用户行为的序列数据很嘈杂。但这是一个研究的方向。【2019年的DIEN】

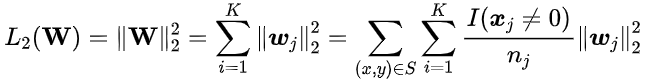
# Training Technique

在阿里巴巴的广告系统中，商品和用户数量达到了数亿。实际上，训练具有大规模稀疏输入特征的工业深度网络是巨大的挑战。

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

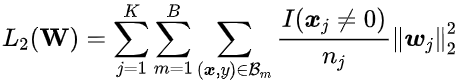
在工业界中深度学习网络中很经常发生过拟合现象，模型的过拟合一般来说需要正则化来进行抑制，但对于工业数据集来说，直接应用传统的正则化方法是不实际的，在训练网络上有稀疏输入和上百万的参数。同样的，对于阿里这样的存在上亿参数需要计算的深度网络，如果使用L1，L2的正则是不科学的。以L2正则化为例。在基于SGD的优化方法的情况下，仅需要更新每个微型批处理中出现的非零稀疏特征的参数，而无需进行正则化。但是，当添加L2正则化时，需要为每个小批量计算整个参数的L2-范数，这将导致计算量极大，直接采用L2正则化将会严重拖慢训练的速度，并且参数扩展到数亿个是不可接受的。

为了解决这个问题，考虑到L2正则的方法对网络参数进行调整时，虽然需要依赖整个模型参数，然而正则起作用的只有一些非零的参数。因此，作者提出了一种有效的小批量处理感知型正则化器（mini-batch aware regularization），它仅针对每个微型批处理（mini-batch）中出现的非0的稀疏特征的参数计算L2-范数，从而使计算成为可能。其公式化如下：



其中定义了整个embedding字典的参数，D是embedding的维度，K是特征空间的维度。是第j个特征对应的embedding向量，表示x含有第j个特征id，代表所有样本中特征id j出现的数量。

在mini batch中上面的公式可以被转化为如下公式：

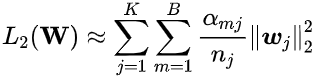


其中定义了mini-batches的数量，表示第m个mini-batch。可以看出两点：1）正则约束对出现的特征频次有关，出现频次越多正则化化约束越强，反正相同。2）该方法通过自适应的学习方式减少了部分的计算开销。

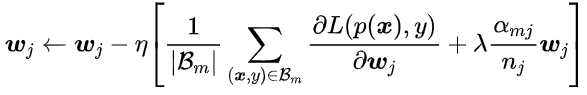
令amj表示第j个特征id是否出现在Bm中，其公式如下：



a mj表示mini bantch中是否有最少一个样本含有特征j，将a mj代入上面公式，使其可以进一步近似如下：

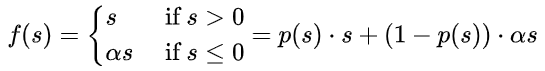


特征j的embedding权重的梯度为(第m个bantch 中embedding的权重w的更新方式应该为：)：



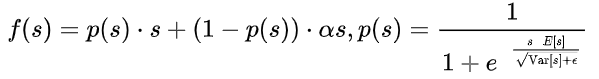
## Data Adaptive Activation Function

PReLU是ReLU激活函数的改良版。relu的公式为：y = max(0, wx+b)由于当输入小于0时，relu的取值为0，这可能导致网络停止更新。PRelu对公式进行了修改，使得输入小于0时输出不等于0。PRelu的公式可以形式化如下：



其中，p(s)是一个指示函数，当s>0的时候等于1，s<=0的时候等于0。

研究表明，PReLU能提高准确率但是也稍微增加了过拟合的风险。无论是ReLU还是PReLU突变点（硬修正点（a hard rectified point））都在0，论文作者认为，当每层的输入遵循不同分布时，这可能不适合，不应该都选择0点为突变点（硬修正点），而是应该依赖于数据的。于是提出了一种data dependent的方法：Dice激活函数。形式如下：



其中E(s)和Var(s)为mini bantch输入值的均值和方差，是一个很小的常量，论文中设置为1e-8。

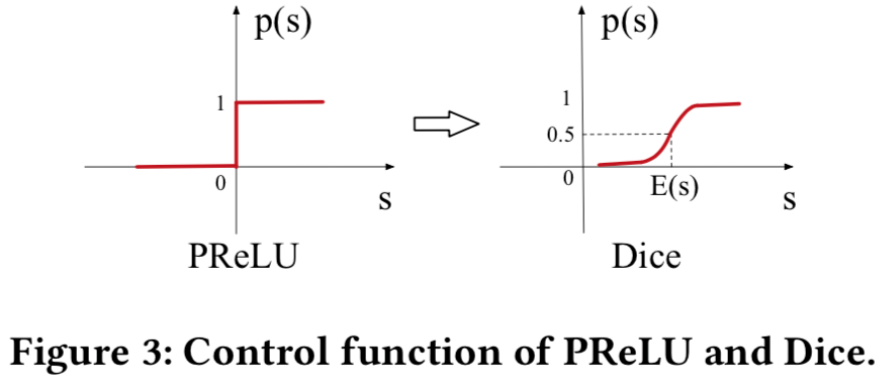
从上面的公式中可以看出，p的计算分为两步：

（1）首先，对x进行均值归一化处理，这使得整流点是在数据的均值处，实现了data dependent的想法；

（2）其次，经过一个sigmoid函数的计算，得到了一个0到1的概率值。巧合的是最近google提出的Swish函数形式为x\*sigmoid(x)，在多个实验上证明了比ReLU函数x\*Max(x,0)表现更优。

另外，期望和方差使用每次训练的mini batch data直接计算，并类似于Momentum使用了指数加权平均。alpha是一个超参数，推荐值为0.99。

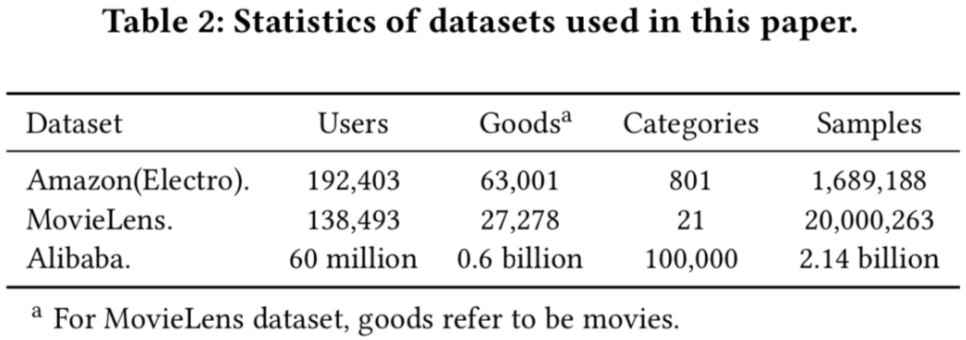
PReLU和Dice的p(s)函数图像如下图所示



Dice的关键思想是根据输入数据的分布来自适应地调整修正点，其值设置为输入的平均值。当则退化为PRelu。

# 实验

## 数据集



Amazon Dataset：包含来自Amazon的产品评论和元数据。选取电商类子集包含192403用户，63001物品，801个种类，和1689188个样本。特征包括：goods\_id、cate\_id、goods\_id\_list、cate\_id\_list。

MovieLens Dataset：选用20M，包含138493用户，27278电影，21种类和20000263样本数。为了适应CTR预估任务，将其转化为2元分类数据---评分4～5为正样本，其余为负样本。特征为：movie\_id、movie\_cate\_id、user rated movie\_id\_list, movie\_cate\_id\_list。

Alibaba Dataset：从阿里巴巴的在线展示广告系统收集了流量日志，其中两个星期的样本用于训练，第二天的样本用于测试。训练和测试集的规模分别约为20亿和1.4亿。embedding维度为12对于所有的16个组来说。

## Baseline

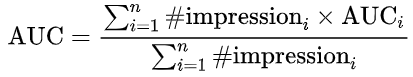
LR：在深度学习网络之前应用非常广泛；

BaseModel；

* Wide&Deep：wide：手工设计低阶特征的交叉，deep：自动提取高阶非线性特征
* PNN：embedding层之后引入乘积层捕获高阶特征交互。
* DeepFM：将Wide&Deep模型中的wide部分改为FM。

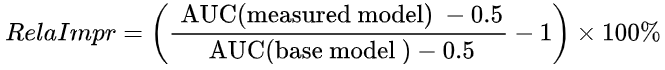
## Metrics

AUC：在原有基础上引入了用户加权AUC的变化形式，它通过对用户AUC进行平均来衡量用户内部订单的优劣，并且显示出与展示广告系统中的在线效果更为相关。计算公式如下：

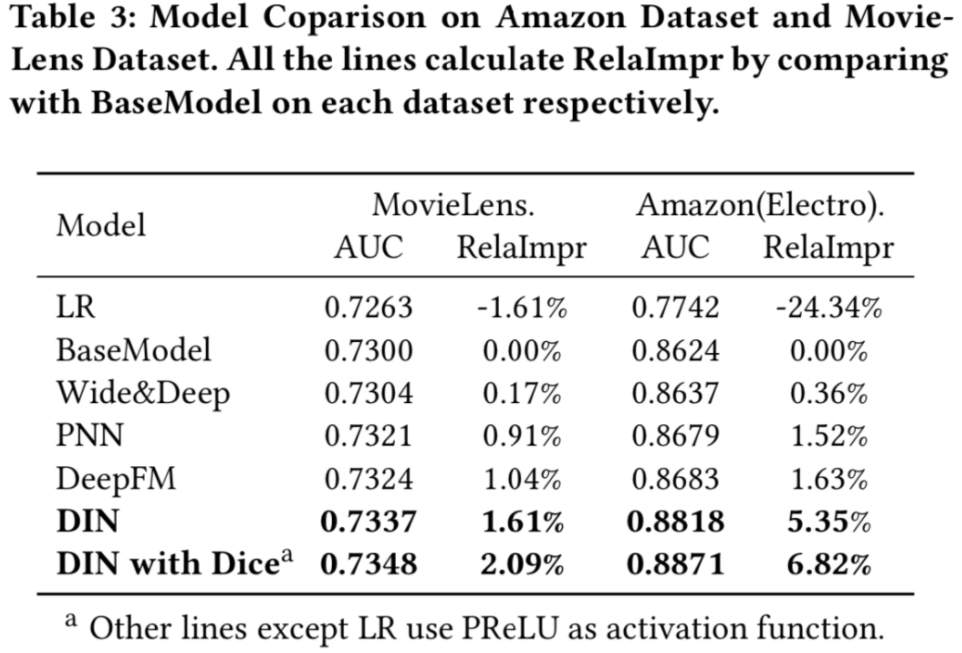


其中表示用户的数量。

RelaImpr：衡量模型的相对改进。对于随机猜测者，AUC的值为0.5。因此，RelaImpr的定义如下：



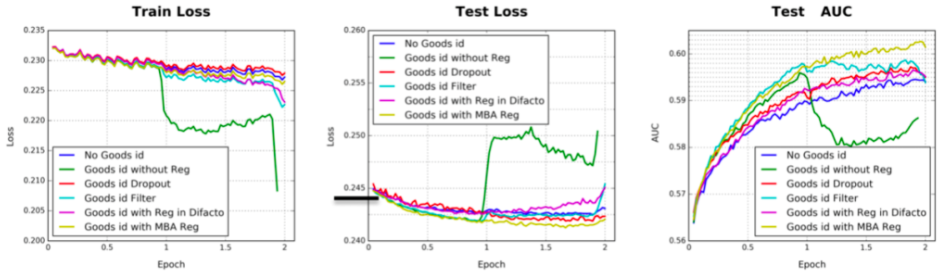
## Result



1、深度学习网络打败了LR，证明了深度学习提取高阶特征的能力。

2、PNN和DeepFM效果比Wide＆Deep更好。DIN在所有竞争对手中表现最好。特别是在具有丰富用户行为的Amazon Dataset上，DIN表现突出。这归功于DIN中局部激活单元结构的设计。DIN通过软搜索与候选广告相关的部分用户行为来关注局部相关的用户兴趣。通过这种机制，DIN获得了用户兴趣的自适应变化表示，与其他深度网络相比，极大地提高了模型的表达能力。此外，带Dice的DIN带来了对DIN的进一步改进，从而验证了所提出的Dice的有效性。

## 正则化参数

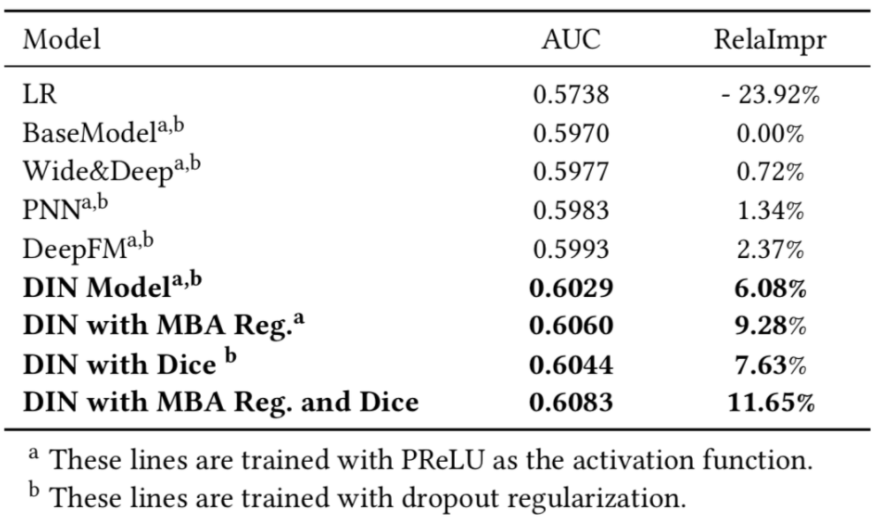


由于Amazon数据集和Movielens数据集的功能维度都不高（约10万），因此所有深度模型（包括我们提出的DIN）都不会遇到严重的过拟合问题。但是，当涉及包含较高维度稀疏特征的在线广告系统中的Alibaba数据集时，过度拟合将是一个很大的挑战。例如，当训练具有细粒度特征的深层模型（例如，表1中尺寸为6亿个goods\_ids的特征）时，不加正则化会在第一个epochs之后会发生严重的过度拟合，这会导致模型性能迅速下降。因此，检验几种常用正则化的性能：

* Dropout：在每一个样本中随机丢弃50%的特征id；
* Filter：按样本中的出现频率过滤访问的goods\_id，仅保留最频繁的那些。剩下的前2000万个goods\_id；
* Regularization in DiFacto：与频繁特征相关的参数不太会被过度正则化；
* MBA：Mini-Batch Aware regularization method；

Dropout可快速防止过拟合，但会降低收敛速度。DiFacto中的正则化会以较高的频率对goods\_id设置更大的惩罚，其效果要比Filter差。MBA效果最好。

## Result of Alibaba Dataset



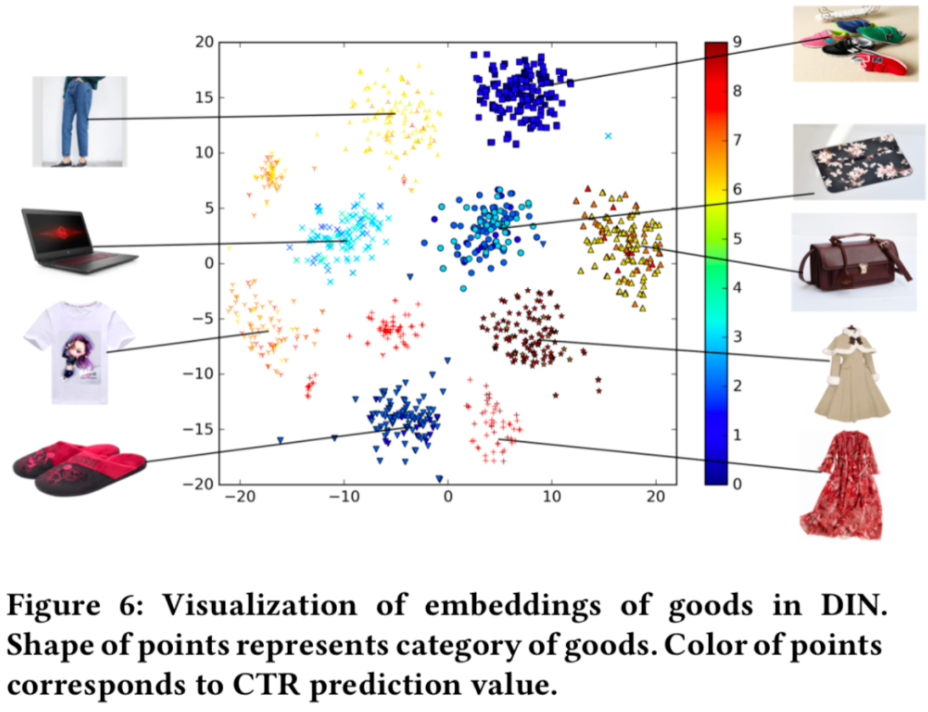
## online A/B testing

2017年5月至2017年6月在阿里巴巴的展示广告系统中进行了在线A/B测试。在将近一个月的测试中，与在线服务模型的最新版本BaseModel相比，接受了Regularizer和激活函数的DIN贡献了高达10.0％的点击率和3.8％的RPM（每千收入）促销。这是一项重大改进，证明了提出的方法的有效性。

原文：“值得一提的是，每天都有成千上万的用户访问我们的系统，对工业深层网络进行在线服务并非易事。更糟糕的是，在流量高峰时，我们的系统每秒为超过100万用户提供服务。需要以高吞吐量和低延迟进行实时CTR预测。例如，在我们的真实系统中，我们需要在不到10毫秒的时间内为每个访问者预测数百个广告。”

## DIN的可视化

文中以一个年轻妈妈为例，挑选了9个种类，每个种类100个商品作为ad的候选集。下图显示了带有t-SNE的商品Embedding向量的可视化图。



可以看到相同的种类的物品几乎都被聚到了同一个类中，说明DIN的局部激活单元的效果还不错。

# 总结

1、本文设计了DIN的新CTR预估模型来通过Local activation unit来获取针对不同广告而变化的用户兴趣的自适应表示向量。

2、还引入了两种新颖的技术（Mini-batch Aware Regularization、Data Adaptive Activation Function）来帮助培训工业深度网络并进一步提高DIN的性能。它们可以轻松地推广到其他行业深度学习任务。

1）用户有多个兴趣爱好，访问了多个good\_id，shop\_id。为了降低纬度并使得商品店铺间的算术运算有意义，我们先对其进行Embedding嵌入。那么我们如何对用户多种多样的兴趣建模那？使用Pooling对Embedding Vector求和或者求平均。同时这也解决了不同用户输入长度不同的问题，得到了一个固定长度的向量。这个向量就是用户表示，是用户兴趣的代表。

2）但是，直接求sum或average损失了很多信息。所以稍加改进，针对不同的behavior id赋予不同的权重，这个权重是由当前behavior id和候选广告共同决定的。这就是Attention机制，实现了Local Activation。

3）DIN使用activation unit来捕获local activation的特征，使用weighted sum pooling来捕获diversity结构。

4）在模型学习优化上，DIN提出了Dice激活函数、自适应正则，显著的提升了模型性能与收敛速度。

代码Github：https://github.com/BlackSpaceGZY/Recommended-System

数据集、代码详细介绍：https://zhuanlan.zhihu.com/p/144153291

参考：

<https://mp.weixin.qq.com/s/uIs_FpeowSEpP5fkVDq1Nw>

<https://mp.weixin.qq.com/s/dRPCcMybmkmVMxYtC4jLdA>

<https://mp.weixin.qq.com/s/qntA2ie6MWJcqoKv_l7QUg>

<https://mp.weixin.qq.com/s/GlcLCxnPjk9Dcx7lr5pVqQ>