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的输入特征映射到低维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最后通过一个 sigmoid 单元转化为点击概率，即 sparse features -> embedding vector -> MLPs -> sigmoid -> output。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。常见的模型有Wide&Deep、DCN、PNN、DeepFM等。

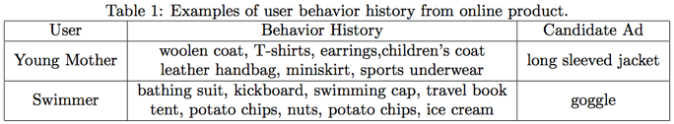
Base Model（Embedding + MLP）将用户特征转化为了一个定长的向量（采用sum pooling或avg pooling），这个定长的向量限制了模型的表达能力，会制约模型捕捉多变的用户兴趣。例如，在电子商务中，用户浏览电商网站时可以同时对多个不同的物品产生兴趣。在CTR预估问题中，就是从用户的浏览历史中去捕获用户的兴趣。而该方法不管候选广告是什么，都是将多个特征向量（Embedding）压缩（采用sum pooling或avg pooling）到一个固定长度的表示向量来学习特定用户所有的兴趣表示，这种方法的问题就是这些 embedding 向量的权重都是相同的，也就是认为不同的特征对 CTR 预估任务的重要性是相同的，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，但是如果用定长向量（扩展向量的维度，保证每个 embedding 的信息都被保留下来）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。

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：



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。

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，自适应地计算用户兴趣的表示向量。具体来说就是通过引入local-activation 机制（局部激活单元，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，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。一言以蔽之，DIN模型更关注和候选集相关的用户历史行为（对于电商来说，历史行为就是点击，添加购物车，下单，收藏等）。

# 简介

DIN模型是阿里2018年在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。引入了attention机制是DIN模型的一大亮点，充分挖掘用户历史行为序列中的信息。除此之外，论文还提出了mini-batch aware regularization和新的激活函数PReLU的改进版Dice，新的正则化方法能够减小计算量，Dice考虑了输入数据的分布情况，提高了训练模型的速度。

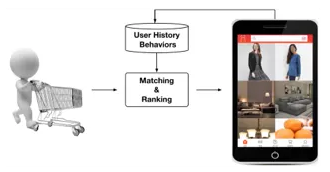
文章的贡献为：

* 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。

DIN模型充分利用/挖掘用户历史行为数据中的信息来提高CTR预估的性能。该算法的提出给CTR预估领域带来了新的研究思路。

# 模型设计

先来看一下阿里巴巴屏幕广告线上系统的整体框架：

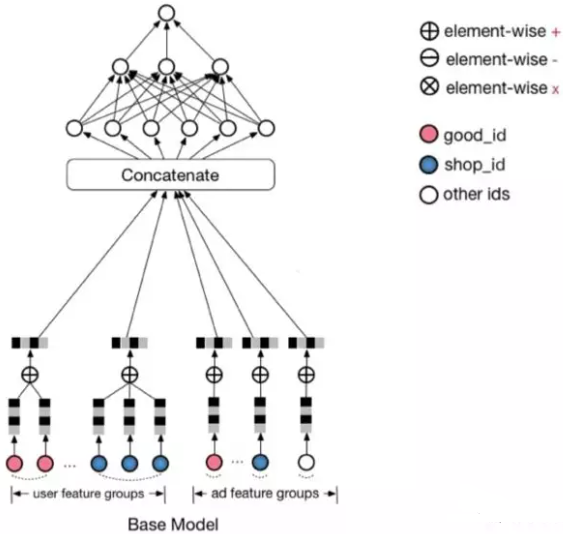


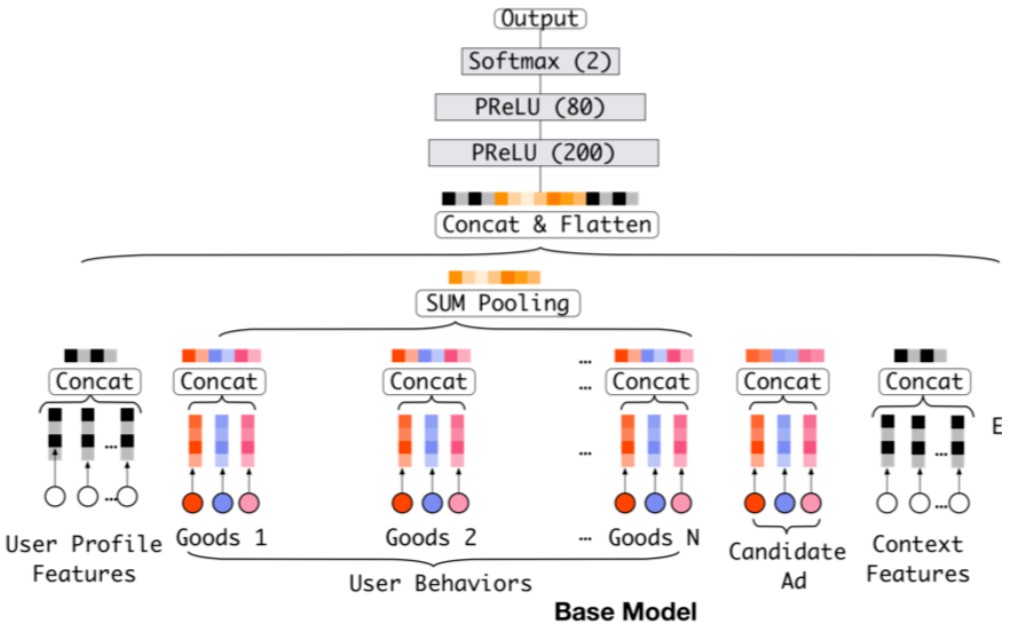
阿里巴巴屏幕广告线上系统的整个运行流程可以描述为：

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。

## Base Model

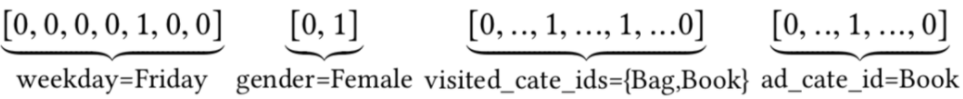
Base Model（基准模型）就是目前比较常见的多层神经网络（Embedding + MLP），即：先对特征进行embedding操作，得到一系列embedding向量之后，将不同group的特征concate起来之后得到一个固定长度的向量，然后将此向量喂给后续的全连接网络，最后输出 pCTR 值，具体网络结构如下：



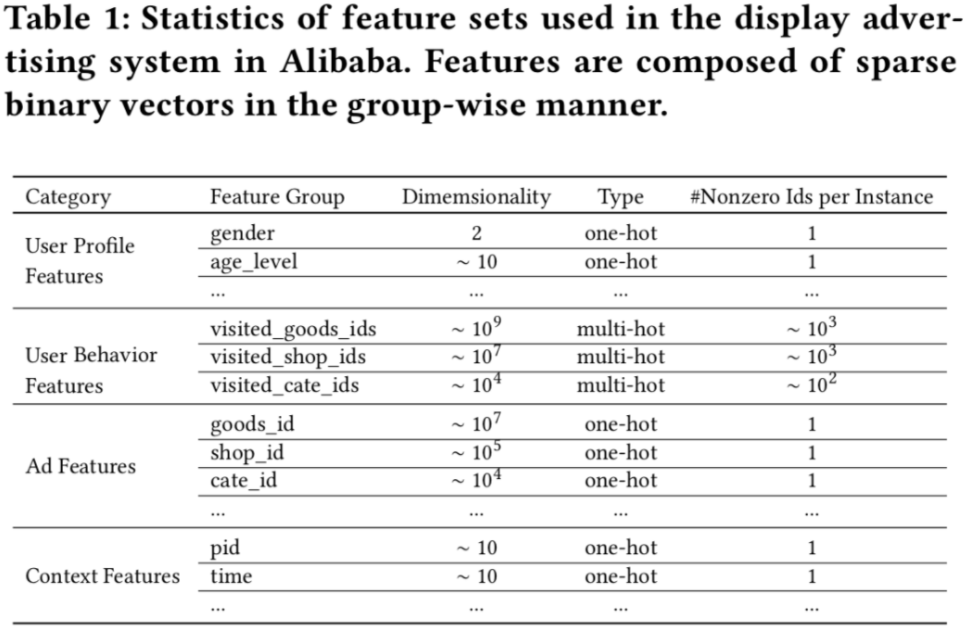


## 特征表示（Feature Representation）

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。



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，其中技术细节后面会展开介绍。

## 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）。

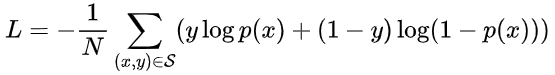
嵌入层和池化层共同将原始稀疏特征转化为固定长度的表示向量。然后将所有向量连接在一起（concatenate），以获得实例的总体表示向量。

## 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。采用的激活单元为 PReLU 的全连接网络；

## 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。

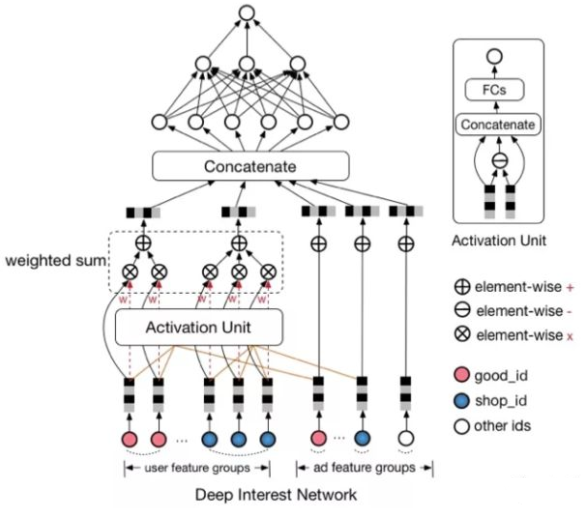
一言以蔽之，在Base Model中，User Behaviors代表的就是用户的行为序列，传统的做法是先将稀疏的行为序列转换成embedding，然后再经过一层sum/average pooling后得到一个embedding，这个embedding就可以表征用户的多个兴趣了。

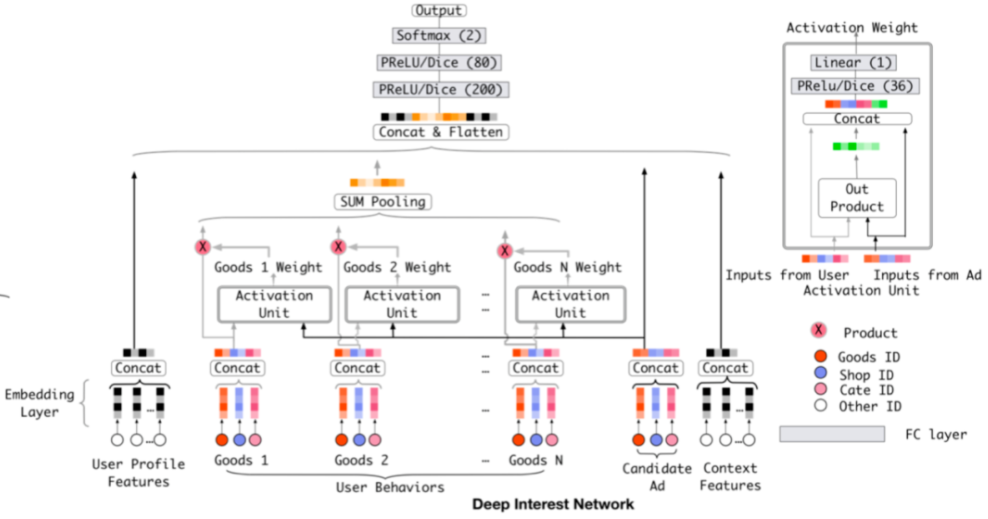
Base Model 有一个非常大的缺点——用户兴趣多样化表达受限。Base Model直接通过池化所有的 Embedding 向量来获得一个定长的表示向量。由于表示向量长度固定，所以其表达也会受到一定的限制，假设最多可以表示k个独立的兴趣爱好，如果用户兴趣广泛不止 k个则表达受限，极大的限制了用户兴趣多样化的表达，而如果扩大向量维度则会带来巨大的计算量和过拟合的风险。

# Deep Interest Network

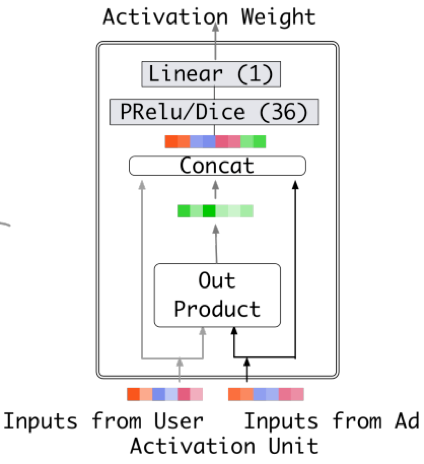
作者认为用户行为在电商应用场景中至关重要，所以在建模时更加关注用户行为。

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，模型结构如下图所示：





可以看到DIN和Base Model在模型结构层面上的区别在于如何聚合多个用户历史行为所对应的embedding向量上，Base Model直接对多个embedding向量进行等权的 sum-pooling【机灵一点的工程师最多加一个 time decay，让最近的行为产生的影响大一些，那就是在做 average pooling 的时候按时间调整一下权重】。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。因此，DIN引入了一个Activation Unit，这个Activation Unit输入是用户的历史行为item的 embedding 向量和候选广告item的 embedding 向量，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。激活单元所对应的网络结构如下图所示：



从上图中可以看到，激活单元的输入包括三个部分：

* 第一部分是原始的用户历史行为 embedding 向量。
* 第二部分是原始的广告 embedding 向量。
* 最后一部分是上述两个 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。主要通过显示知识帮助进行关联建模（简单的多层全连接无法学出内积，相当于做了一次特征工程）。

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，Vi表示behavior id i的嵌入向量，比如good\_id,shop\_id等。Vu是所有behavior ids的加权和，表示的是用户兴趣；Va是候选广告的嵌入向量；wi是候选广告影响着每个behavior id的权重，也就是Local Activation。wi通过Activation Unit计算得出，这一块用函数去拟合，表示为g(Vi,Va)。

|  |
| --- |
| 在阅读博客的时候发现也有如下这种表达方式，也记录下来。    其中是用户 U 所有的广告 A 的 Embedding 向量，{e1, e2, ..., eH }是用户U长度为H的历史行为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。注意力机制顾名思义，就是模型在预测的时候，对用户不同行为的注意力是不一样的，“相关”的行为历史看重一些，“不相关”的历史甚至可以忽略。与传统的attention机制不同的是，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。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。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iPhone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iPhone获得更大的（更高的兴趣强度）价值。之所以说很可能而不是肯定，是因为商品还有本身的性质，比如衣服可以换的很频繁，但iPhone不是，衣服的盈利远远不如iPhone的利润大等等。所以在计算attention的时候，还会有很多特征需要挖掘来计算得到可靠的权重。传统的注意力机制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，也可能的原因是兴趣本身就是共存的。但这是一个研究的方向。【2019年的DIEN】

|  |
| --- |
| Attention机制简单的理解就是，针对不同的广告，用户历史行为与该广告的权重是不同的。假设用户有ABC三个历史行为，对于广告D，那么ABC的权重可能是0.8、0.1、0.1；对于广告E，那么ABC的权重可能是0.3、0.6、0.1。这里的权重，就是Attention机制即上图中的Activation Unit所需要学习的。  为什么要引入Attention？举例：假设用户的兴趣的Embedding是Vu，候选广告的Embedding是Va，用户兴趣和候选的广告的相关性可以写作F(U,A) = Va \* Vu。如果没有Local activation机制，那么同一个用户对于不同的广告，Vu都是相同的。如果有两个广告A和B，用户兴趣和A，B的相似性都很高，那么在Va和Vb连线上的广告都会有很高的相似性。这样的限制使得模型非常难学习到有效的用户和广告的embedidng表示。 |

|  |
| --- |
| DIN的核心idea是，候选广告需要和用户行为中的每一个商品做权重的计算，然后用权重去做加权平均得到用户针对这个候选广告的兴趣向量。也就是说 DIN 是通过考虑用户的当前点击行为与历史行为的相关性来自适应地计算用户兴趣的表示向量，用户的历史行为的权重依赖于正在看的商品。  DIN使用activation unit来捕获local activation的特征，使用weighted sum pooling来捕获diversity结构。 |

# 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。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。

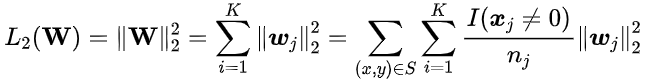
在工业界，特别是商品用户都过亿的情况下，训练有大量的稀疏输入特征的是一个巨大的挑战。接下来主要介绍两个重要的技术用来加速训练。

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

在深度学习网络中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，在阿里的特征构建中，直接使用了上亿维度的商品 id、用户 id、广告 id 等，这些大规模的稀疏特征和数亿个参数无法直接应用传统的 L2 正则化。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（Mini-Bacth）计算整个参数的L2-范数，这将导致计算量极大增加，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。

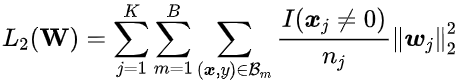
|  |
| --- |
| CTR 中输入稀疏而且维度高，通常的做法是加入 L1、L2、Dropout 等防止过拟合。但是论文中尝试后效果都不是很好。用户数据符合长尾定律long-tail law，也就是说很多的 feature id只出现了几次，而一小部分feature id出现很多次。这在训练过程中增加了很多噪声，并且加重了过拟合。  对于这个问题一个简单的处理办法就是：直接去掉出现次数比较少的 feature id。但是这样就人为的丢掉了一些信息，导致模型更加容易过拟合，同时阈值的设定作为一个新的超参数，也是需要大量的实验来选择的。因此，论文中提出了自适应正则的做法。 |

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。因为稀疏特征的众多，网络中大部分的参数都分布在embedding层，论文以embedding层为例来讲解了正则化的操作。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，表示样本x的第j个特征是否为0，代表所有样本中特征j出现的次数。

在mini batch中，上面的公式可以被转化为如下公式：

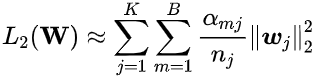


其中定义了mini-batch的数量，表示第m个mini-batch。可以看出两点：1）正则约束对出现的特征频次有关，出现频次越多正则化化约束越强，反正相同。2）该方法通过自适应的学习方式减少了部分的计算开销。

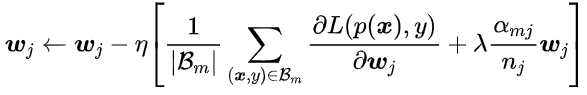
令amj表示第m批次（Bm）中含有特征j的样本至少出现过一次，其公式如下：



表示的是特征j在mini-batch Bm中至少出现过一次。将代入上面公式，使用最大值来代替求和，用速度换取精度，可以近似如下：



此时，我们便得到了一个 L2 正则化的近似计算解。特征j的embedding权重的梯度更新公式如下(第m个bantch 中embedding的权重w的更新方式应该为：)：



其中表示第j维特征的非零频次。

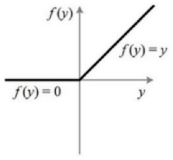
与传统的梯度更新方式的区别在于：

* 过滤频次为0的特征，仅计算出现在第 m 次 Mini-Batch 中的特征参数；
* 考虑特征出现的频率，频次越高，给与较小的正则化强度；频次越低，给予较大的正则化强度。

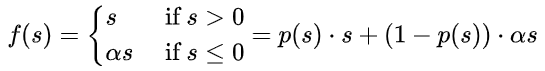
## Data Adaptive Activation Function

### 从ReLU到PReLU

relu的公式为：y = max(0, wx+b)。Relu 激活函数的图像如下：

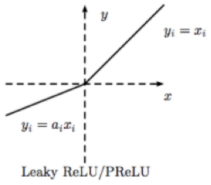


Relu 激活函数在值大于0时原样输出，小于0时输出为0。因此，当输入小于0时，网络会停止更新，这样导致了许多网络节点的更新缓慢。PRelu（也称Leaky Relu）对公式进行了修改，使得输入小于0时输出不等于0。PRelu的公式可以形式化如下：



其中，p(s)是一个指示函数（也称控制函数），当s>0的时候等于1，s<=0的时候等于0。是学习率。

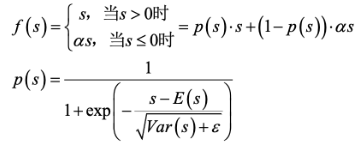
PRelu的函数图像如下：



PRelu激活函数即使输入小于0，网络的参数也可以更新，加快了收敛速度。

### 从PReLU到Dice

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。Dice 激活函数的全称是 Data Dependent Activation Function。其形式如下：



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。E(s)和Var(s)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：





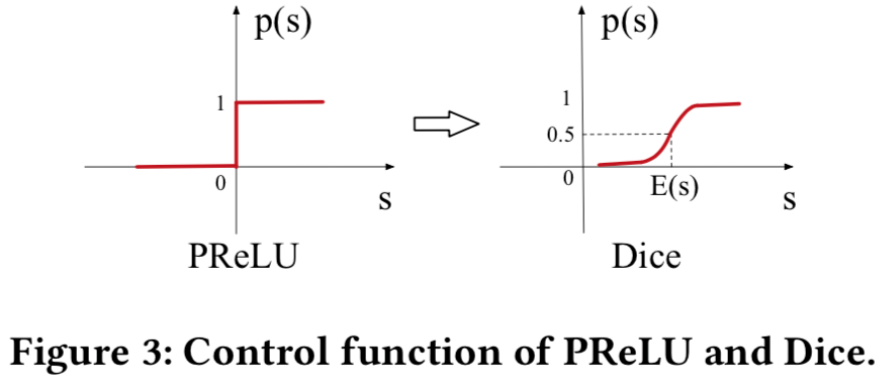
从上面的公式中可以看出，p的计算分为两步：

（1）首先，对x进行均值归一化处理，这使得修正点在数据的均值处，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的控制函数p(s)的函数图像如下图所示（其中PReLU的函数图像画错了，正确的函数图像请参考上面的PReLU图像）：

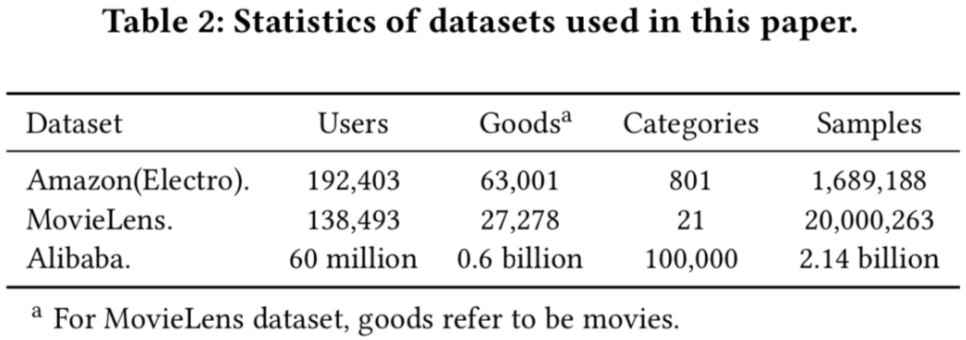


Dice激活函数，一方面平滑了激活函数在rectified point附近的曲线，另一方面激活函数会根据每层输入数据的分布来自适应调整rectified point的位置，从而对模型参数的更新和收敛起到一定的加速作用。

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。当均值为0方差为1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；

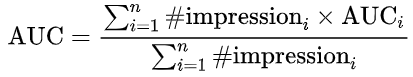
* Wide&Deep：wide：手工设计低阶特征的交叉，deep：自动提取高阶非线性特征
* PNN：embedding层之后引入乘积层捕获高阶特征交互。
* DeepFM：将Wide&Deep模型中的wide部分改为FM。

## Metrics

### GAUC

**用户级别的AUC：**AUC 表示正样本得分比负样本得分高的概率。在CTR实际应用场景中，CTR预测常被用于对每个用户候选广告的排序。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。实现了用户级别的 AUC 计算。

**基于用户加权的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n表示用户的数量，和分别表示第i个用户的曝光物品数量和AUC值。这样做的好处就是消除不同用户的偏差对模型的影响，更能反应出广告系统的在线性能。

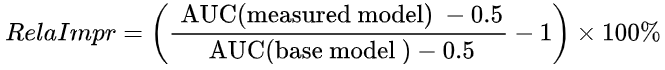
首先要肯定的是，AUC是要分用户看的，模型的预测结果只要能够保证对每个用户来说，用户想要的结果排在前面就好了。

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。

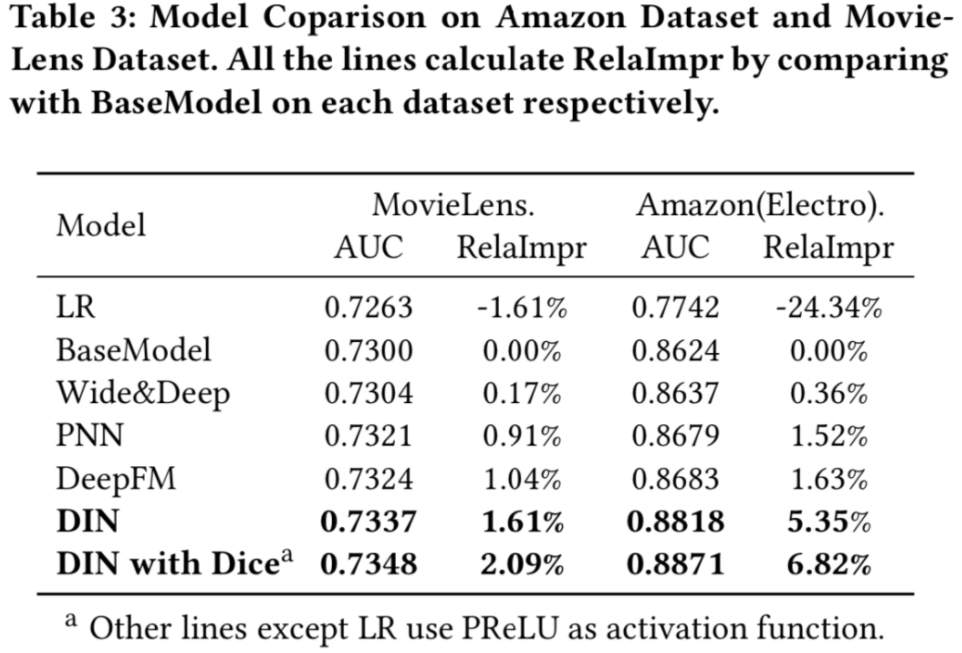
但是上文中所说的差异是在用户点击数即样本数相同的情况下说的。还有一种差异是用户的展示次数或者点击数，如果一个用户有1个正样本，10个负样本，另一个用户有5个正样本，50个负样本，这种差异同样需要消除。那么GAUC的计算，不仅将每个用户的AUC分开计算，同时根据用户的展示数或者点击数来对每个用户的AUC进行加权处理。进一步消除了用户偏差对模型的影响。实验证明，GAUC确实是一个更加合理的评价指标。

### RelaImpr

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：



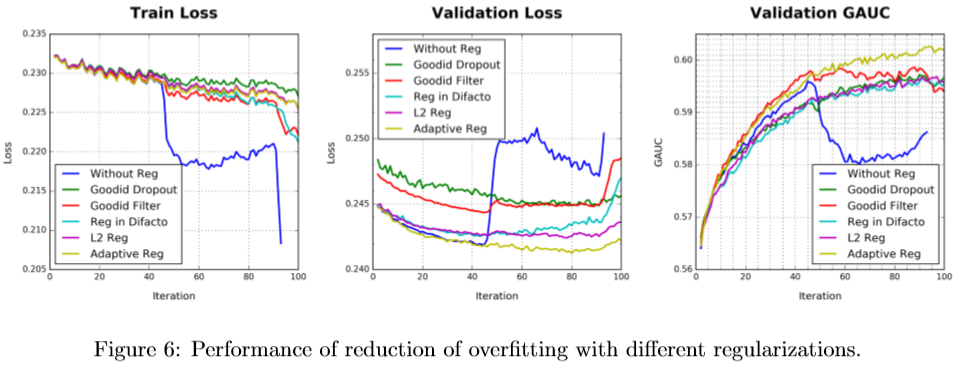
## Result



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

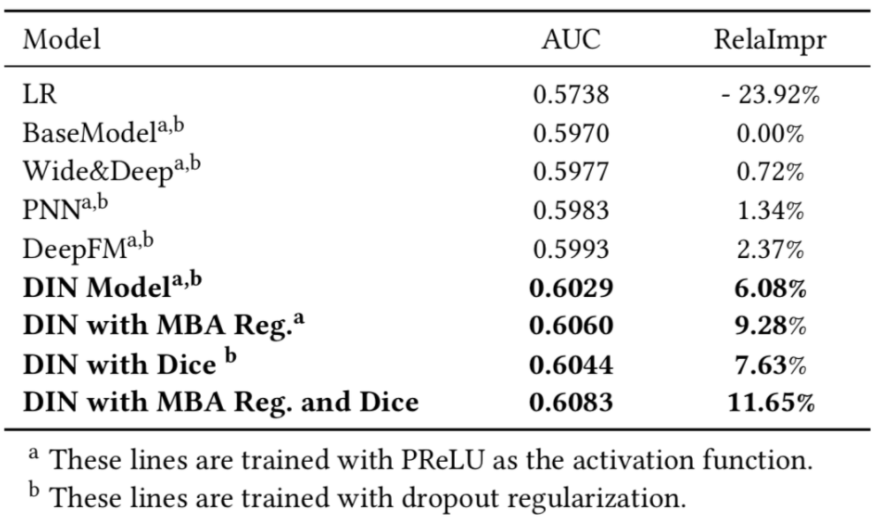


由于Amazon数据集和Movielens数据集的功能维度都不高（约10万），因此所有深度模型（包括我们提出的DIN）都不会遇到严重的过拟合问题。但是，当涉及包含较高维度稀疏特征的在线广告系统中的Alibaba数据集时，过度拟合将是一个很大的挑战。例如，当训练具有细粒度特征的深层模型（例如，表1中尺寸为6亿个goods\_ids的特征）时，不加正则化会在第一个epochs之后会发生严重的过度拟合，这会导致模型性能迅速下降。因此，检验几种常用正则化的性能：

* 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Base Model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（Revenue Per Mille，千次展示收入）。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。”

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。

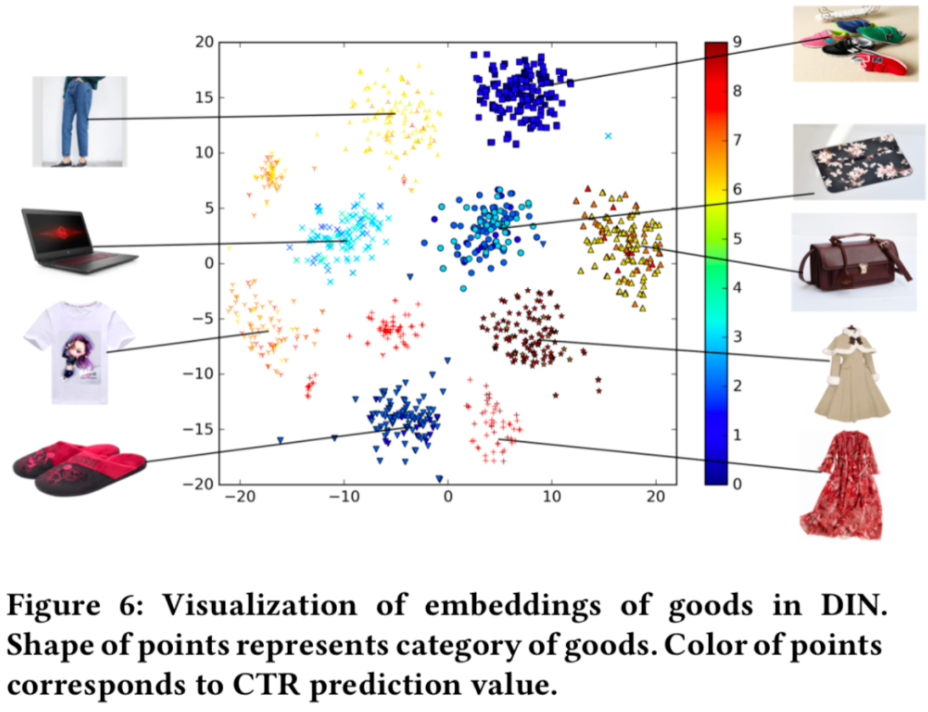
## Attention 结果展示

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。



## 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。

# 实践

参考：

https://github.com/jpegbert/code\_study/tree/master/DIN/demo1

# 总结

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。

Deep Interest Network（DIN）模型，将 NLP 中的注意力机制引入到 CTR 预估中，通过设计局部激活单元来自适应学习用户的兴趣；设计出小批量感知正则化和自适应激活函数Dice，以便与大规模稀疏数据的工业训练；引入基于用户加权的 GAUC 计算方式来代替传统的 AUC 计算。

这篇论文无疑是非常优秀的：

一是因为这篇论文的工程性很强。

二是因为这篇论文对用户行为的观察非常精准。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。

三是模型的微创新，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。

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>

<https://mp.weixin.qq.com/s/2YPX87ocw_6RyWjzxJwcUA>

<https://mp.weixin.qq.com/s/JWTDsMhHI50k6mVJwMr0nA>

<https://mp.weixin.qq.com/s/-EDxUwAKdXnmUKeLJTiAUw>

<https://mp.weixin.qq.com/s/kzhnhhl--nkZcgPTnfClyQ>

<https://mp.weixin.qq.com/s/L6tiUjdVB5PWH3nKBgPXkA>