深度兴趣网络总结（Deep Interest Network）

DIN 的主要贡献是利用/挖掘用户历史行为数据中的信息来提高 CTR 预估的性能。

1. 背景

深度学习在 CTR 预估领域具有广泛的应用，常见的算法比如 $Wide\&Deep$, $DeepFM$ 等。这些方法的一般思路是：通过 Embedding 层，将高维离散特征转换为固定长度的连续特征，然后通过多个全连接层，最后通过一个 $sigmoid$ 函数转化为 0-1 值，代表点击的概率，即 Sparse Features → Embedding Vector → MLP → Sigmoid → Output.

这种方法的优点在于：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。不过，研究者观察线上数据时发现，用户行为数据中有两个重要的特征：

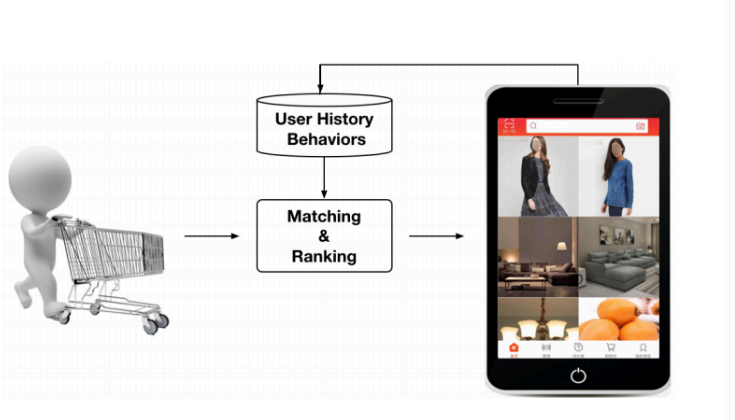
- diverse: 用户在浏览电商网站的过程中显示出的兴趣是十分多样性的；
- locally activated: 由于用户兴趣的多样性，只有部分历史数据会影响当前推荐的物品是否被点击，而不是所有的历史记录。

diverse 体现在年轻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，而爱好游泳的人同样兴趣广泛，历史记录设计浴装、旅游手册、踏水板、马铃薯、坚果等。locally activated 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、冰淇淋的关系就不大了，而跟他游泳相关的历史记录如游泳帽的关系就比较密切。

以上就是 DIN 提出的背景。

2. 模型设计

推荐系统的整体框架为：



主要包括两个阶段：

1. matching stage 针对该用户产生候选 ads，使用比如协同过滤的方法；

2. ranking stage 针对每个推荐 ad 预测 CTR，然后排序；

特征设计

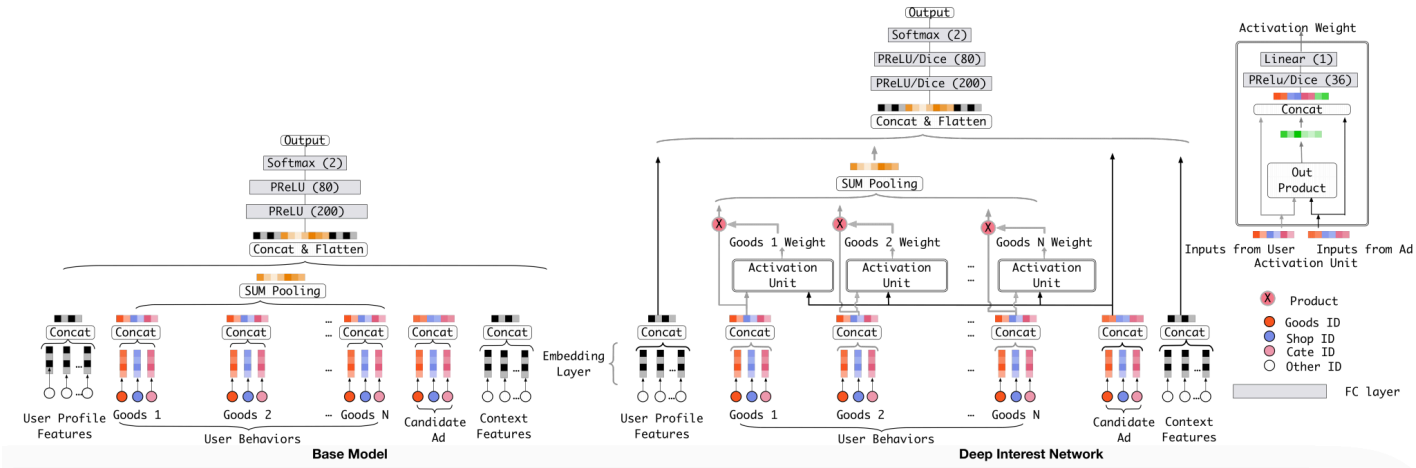
文章将涉及的特征分为四个部分：用户特征、用户行为特征、广告特征、上下文特征，具体如下：

Category	Feature Group	Dimemsionality	Type	#Nonzero Ids per Instance
User Profile Features	gender	2	one-hot	1
	age_level	~ 10	one-hot	1
	...	...	...	...
User Behavior Features	visited_goods_ids	~ 10^9	multi-hot	~ 10^3
	visited_shop_ids	~ 10^7	multi-hot	~ 10^3
	visited_cate_ids	~ 10^4	multi-hot	~ 10^2
Ad Features	goods_id	~ 10^7	one-hot	1
	shop_id	~ 10^5	one-hot	1
	cate_id	~ 10^4	one-hot	1
	...	...	...	...
Context Features	pid	~ 10	one-hot	1
	time	~ 10	one-hot	1
	...	...	...	...

其中，用户特征是 multi-hot 的，即多值离散特征。针对这种特征，由于每个用户涉及的非 0 值个数不一样，常见的做法就是将 id 转换称 embedding 之后，加一层 pooling，比如 average-polling, sum-pooling, max-polling。DIN 中使用的是 weighted-sum，其实就是加权的 sum-polling，权重经过一个 activation unit 计算得到。

Base Model (Embedding&MLP)

大部分深度学习模型都有一个类似的 Embedding&MLP 样式，可以称之为基准模型，如下图左所示。



Base Model 首先把 one-hot 或 multi-hot 特征转换为特定长度的 embedding，作为模型的输入，然后经过一个 DNN 部分，得到最终的预估值。特别地，针对 multi-hot 的特征，做了一次 element-wise + 的操作，即 sum-polling，这样，不管特征中有多少非 0 值，经过转换之后的长度都一样。

Base Model 的 loss 使用 negative log-likelihood 函数：

$$L = -\frac{1}{N}\sum_{(x,y)\in S}(y\log p(x) + (1-y)\log(1-p(x)))$$

其中，S 为大小为 N 的训练集，x 为输入数据，$y \in \{0,1\}$ 为 label，$p(x)$ 是经过 softmax 层的输出，表示 x 被点击的概率。

Deep Interest Network

Base Model 有一个很大的问题，它对用户的历史行为是同等对待的，没有做任何处理，这显然是不合理的。一个很显然的例子，离现在越近的行为，越能反映你当前的兴趣。因此，对用户历史行为基于 Attention 机制进行一个加权，这就是 DIN 模型的思路，如上图右所示。

对 Attention 机制简单的理解为，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，ABC 的权重可能是 [0.8,0.1,0.1]；对于广告 E，ABC 的权重可能是 [0.3,0.6,0.1]。这里的权重，就是 Attention 机制所需学习的。

加入 Activation unit 后，用户的兴趣计算为：

$$v_U(A) = f(v_A, e_1, e_2, \cdots, e_H) = \sum_{j=1}^{H} a(e_j, v_A) = \sum_{j=1}^{H} w_j e_j$$

其中，$\{e_1, e_2, \cdots, e_H\}$ 是用户 U 行为的 embedding 向量，v_A 是 广告 A 的 embedding 向量，

3. 模型训练

提出了两种训练技术：

Mini-batch Aware Regularization

过拟合是模型训练的一大挑战。文章提出，对于具有稀疏输入和上亿参数的网络来说，直接使用 ℓ_2 或 ℓ_1 正则是不现实的。因为，在无正则项的 SGD 优化方法中，每个 mini-batch 中只有非零稀疏特征需要被更新。而加入了 ℓ_2 正则项后，对每个 mini-batch，需要根据所有参数计算 L2-norm，这个计算量是无法接受的。

文章作者提出了 mini-batch aware regularizer，只需要对出现在 mini-batch 中的稀疏特征的参数做 L2-norm。设 $W \in \mathbb{R}^{D\times K}$ 表示 embedding dictionary 的参数，D 为 embedding vector 的维度，K 为特征空间的维度。Expand the ℓ_2 regularization on W over samples:

$$L_2(W) = \|W\|_2^2 = \sum_{j=1}^{K}\|w_j\|_2^2 = \sum_{(x,y)\in S}\sum_{j=1}^{K}\frac{I(x_j \neq 0)}{n_j}\|w_j\|_2^2$$

其中，$w_j \in \mathbb{R}^D$ 是第 j 个 embedding vector，$I(x_j \neq 0)$ 表示实例 x 是否有特征 id j，n_j 表示特征 id j 在所有 sample 中的出现次数。上式可以转化为：

$$L_2(W) = \sum_{j=1}^{K} \sum_{m=1}^{B} \sum_{(x,y)\in\mathcal{B}_m} \frac{I(x_j \neq 0)}{n_j} \|w_j\|_2^2$$

其中，$\mathcal{B}$ 表示 mini-batch 的个数，$\mathcal{B}_m$ 表示第 m 个 mini-batch，另 $\alpha_{mj} = \max_{(x,y)\in\mathcal{B}_m} I(x_j \neq 0)$，则上式可近似表示为：

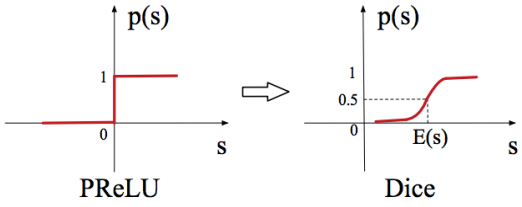
$$L_2(W) \approx \sum_{j=1}^{K} \sum_{m=1}^{B} \frac{\alpha_{mj}}{n_j} \|w_j\|_2^2$$

对于第 m 个 mini-batch，梯度为：

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

Data Adaptive Activation Function

首先，PReLU 是对 ReLU 激活函数的改进，ReLU 激活函数在值大于 0 时原样输出，小于 0 时输出为 0，这样会导致许多网络节点的更新缓慢。PReLU 的问题是，我们认为分割点都是 0，但实际上，分割点应该由数据决定，因此文章提出了 Dice 激活函数。



PReLU 激活函数为：

$$f(s) = \begin{cases} s & if\ s > 0 \\ \alpha s & if\ s \leq 0 \end{cases} = p(s) \cdot s + (1 - p(s)) \cdot \alpha s$$

自适应激活函数 Dice 为：

$$f(s) = p(s) \cdot s + (1 - p(s)) \cdot \alpha, \quad p(s) = \frac{1}{1 + e^{-\frac{s - E[s]}{\sqrt{Var[s] + \epsilon}}}}$$

其中，$E[s]$ 和 $Var[s]$ 分别是每个 mini-batch 输入的均值和方差，ϵ 为很小的常数，实验中值为 10^{-8}。Dice 可以看作是 PReLU 的一般形式，当 $E(s) = 0, Var[s] = 0$ 时，Dice 退化为 PReLU。