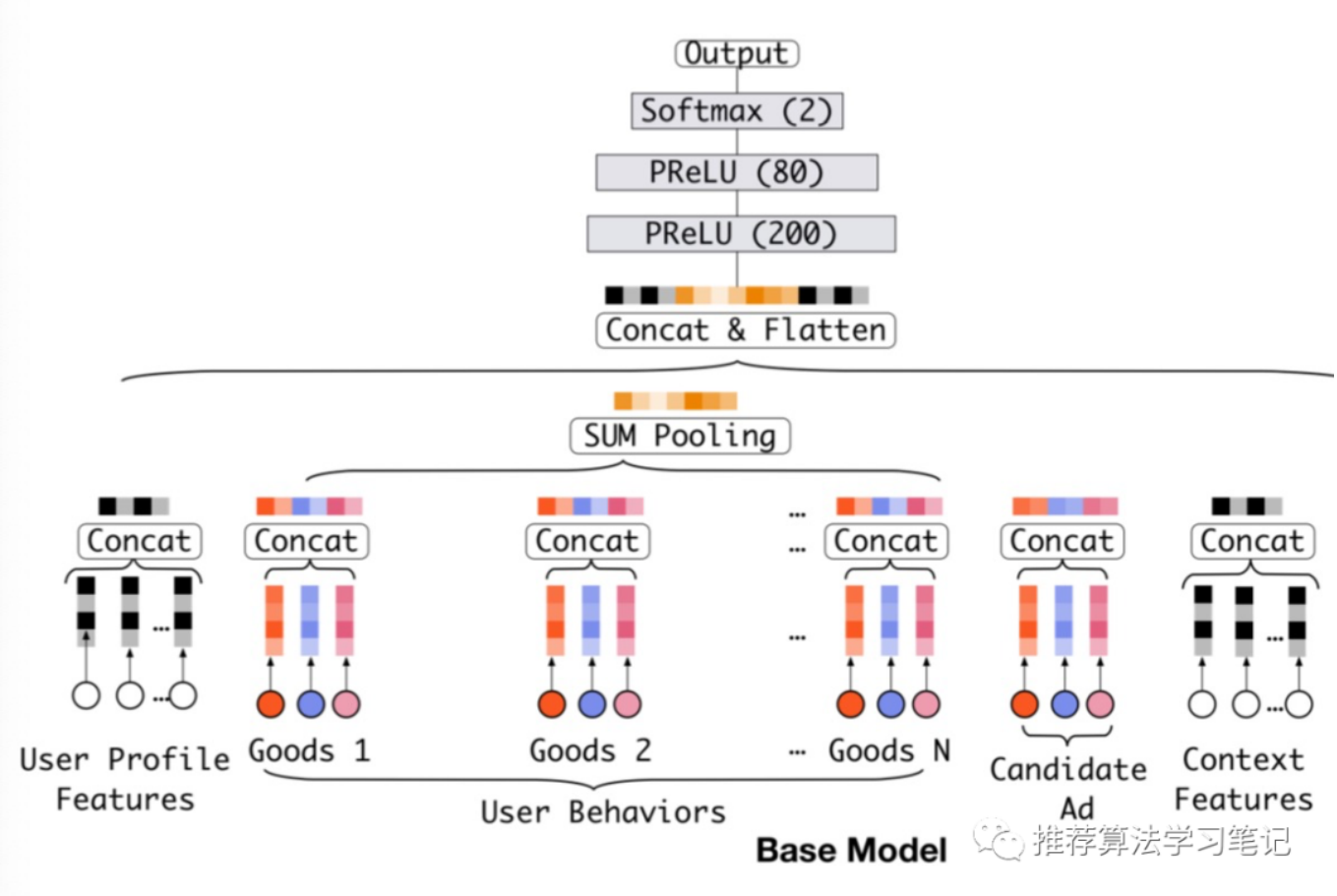
[深度模型] 推荐算法工程师必学模型：阿里深度兴趣网络DIN

原创 xxxhuang 推荐算法学习笔记 7月15日

一. 概述

本文主要介绍推荐算法工程师必学的经典模型DIN，paper名字为《Deep Interest Network for Click-Through Rate Prediction》

在逛淘宝的时候，我们的兴趣是多样非单一的。例如一个年轻的妈妈，她可能同时对衣服，护肤品，婴幼儿产品感兴趣。对于用户的多兴趣，一般是怎么建模的呢？如下图所示

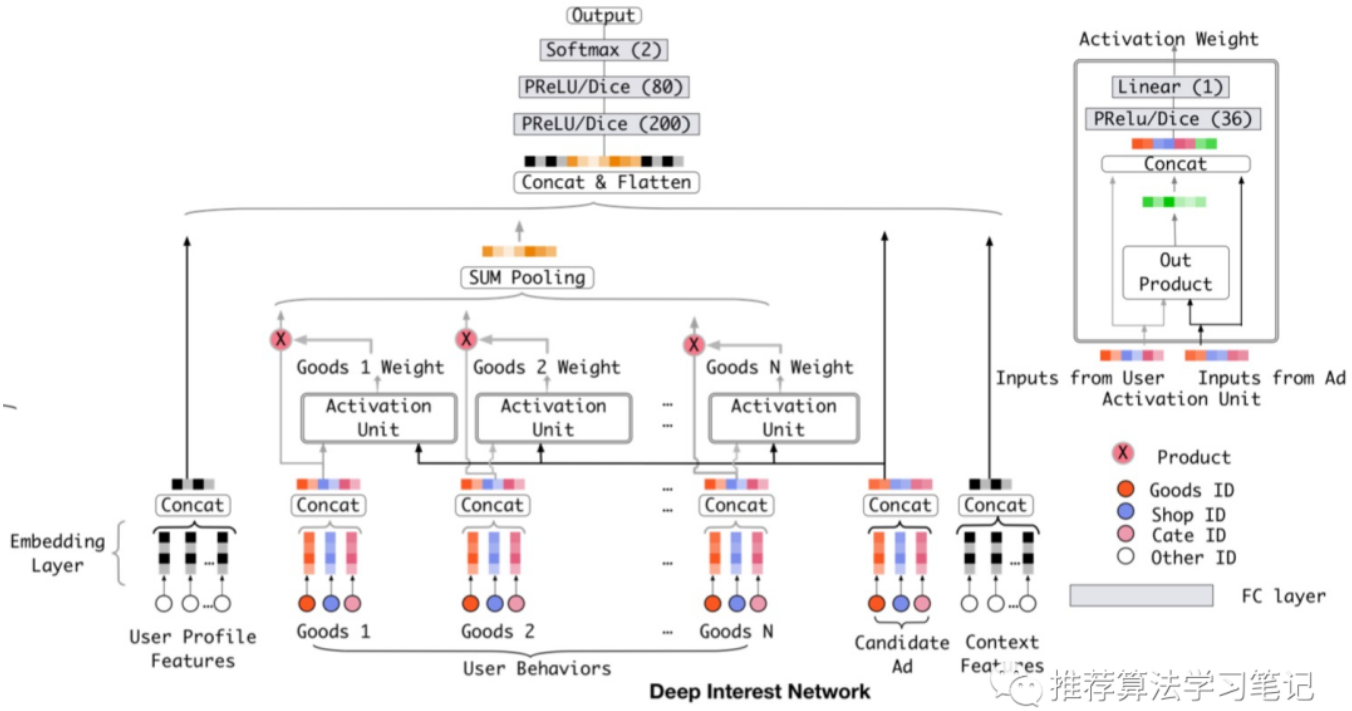


图中的User Behaviors代表的就是用户的行为序列，传统的做法是先将稀疏的行为序列转换成embedding，然后再经过一层sum/average pooling后得到一个embedding，这个embedding就可以表征用户的多个兴趣了。

二. 深度兴趣网络DIN

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和候选的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。

三. 训练的改进

（1）Mini-batch Aware Regularization

因为模型的embedding参数规模巨大，直接采用l2 regularization将会严重拖慢训练的速度。因此paper作者只正则化那些出现在mini batch中非0的稀疏特征对应的参数。

L2正则化的公式如下所示

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

其中K表示特征空间的维度，**w**j是第j个特征对应的embedding向量， I表示x是不是有第j个特征id

转换成mini-batch的方式如下所示

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

B表示mini batch的数量，**B**m表示第m个mini batch。令**a**mj表示第j个特征id是否出现在**B**m中，则公式可以近似表示为

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

因此最终的梯度计算可以表示为

$$w_j \leftarrow w_j - \eta \left[\frac{1}{|\mathcal{B}_m|} \sum_{(\boldsymbol{x},y)\in\mathcal{B}_m} \frac{\partial L(p(\boldsymbol{x}), y)}{\partial w_j} + \lambda \frac{\alpha_{mj}}{n_j} w_j \right], \quad (7)$$

（2）Data Adaptive Activation Function

PReLU是一个常用的激活函数，公式如下所示

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

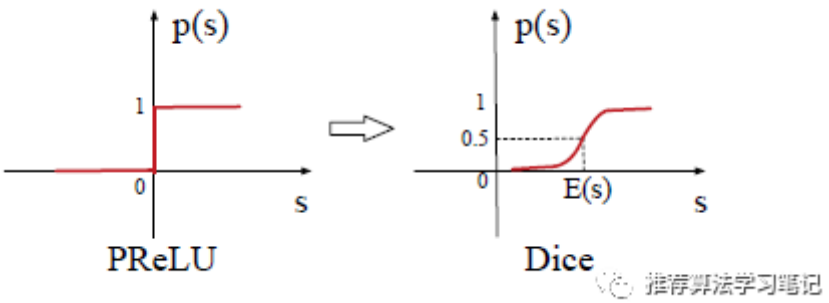
p(s)是一个指示函数，当s>0的时候等于1，s<=0的时候等于0

在PReLU的基础上paper作者提出了Dice激活函数，对模型指标的提升有一定的帮助，Dice公式如下所示

$$f(s) = p(s) \cdot s + (1 - p(s)) \cdot \alpha s, \quad p(s) = \frac{1}{1 + e^{-\frac{s - E[s]}{\sqrt{Var[s] + \epsilon}}}} \quad (9)$$

其中E[s]和Var[s]是mini batch的均值和方差

PReLU和Dice的p(s)函数图像如下图所示



四. 实验结果

Model	AUC	RelaImpr
LR	0.5738	- 23.92%
BaseModel[a,b]	0.5970	0.00%
Wide&Deep[a,b]	0.5977	0.72%
PNN[a,b]	0.5983	1.34%
DeepFM[a,b]	0.5993	2.37%
DIN Model[a,b]	**0.6029**	**6.08%**
DIN with MBA Reg.[a]	**0.6060**	**9.28%**
DIN with Dice [b]	**0.6044**	**7.63%**
DIN with MBA Reg. and Dice	**0.6083**	**11.65%**

[a] These lines are trained with PReLU as the activation function.
[b] These lines are trained with dropout regularization.

观察paper作者在阿里真实数据集上的表现，可以看到DIN在使用Mini-batch Aware Regularization和Dice的时候效果是最优的

五. 总结

以上便是DIN的全部内容，下一篇文章将讲述阿里另一个经典深度推荐模型DIEN，欢迎关注！