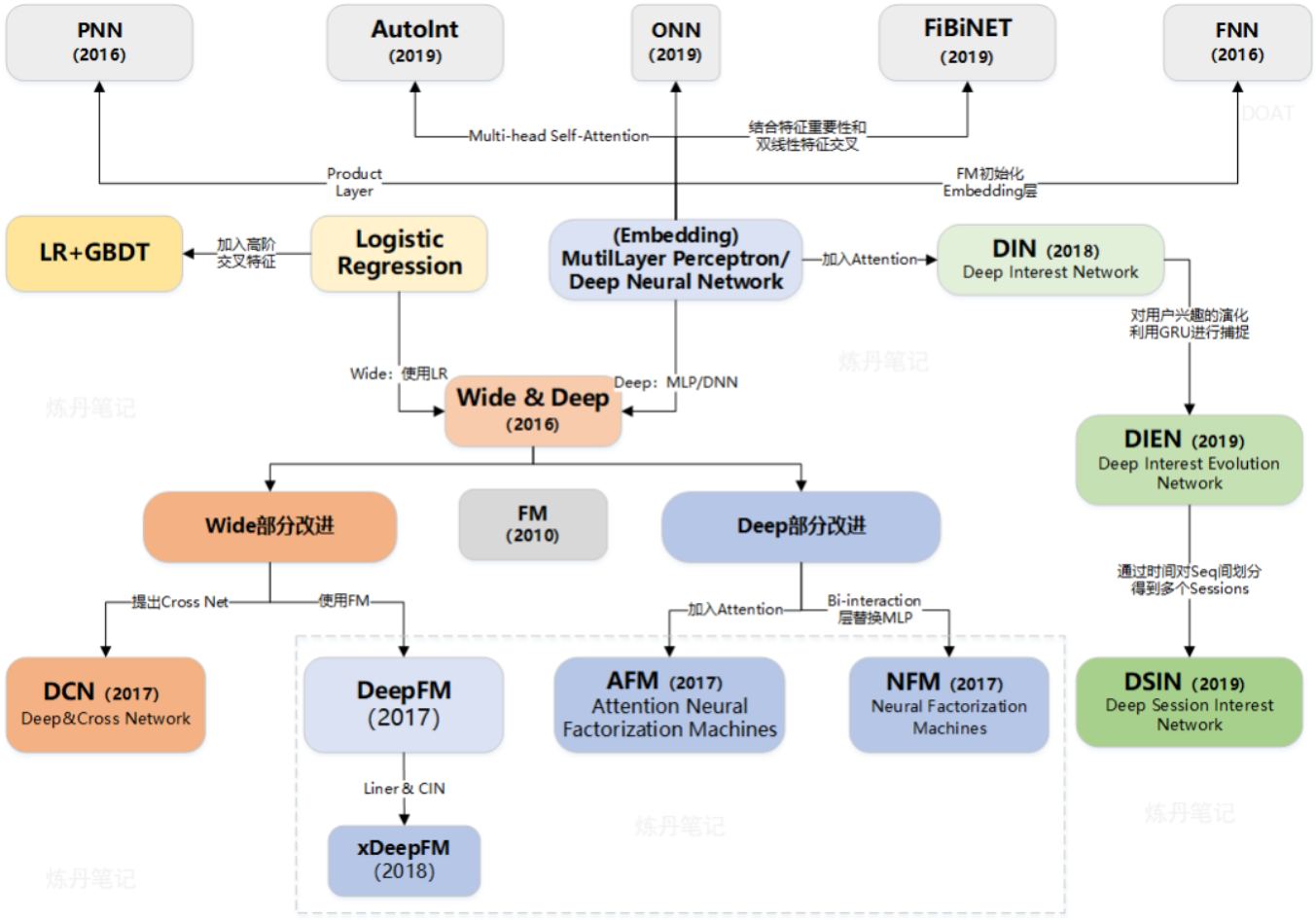
CTR预估系列炼丹入门手册

原创 九羽 炼丹笔记 1周前

收录于话题
#搜索推荐前沿算法

12个

CTR预估系列家谱



炼丹之前，先放一张CTR预估系列的家谱，让脉络更加清晰。

（一）FiBiNET：结合特征重要性和双线性特征交互进行CTR预估

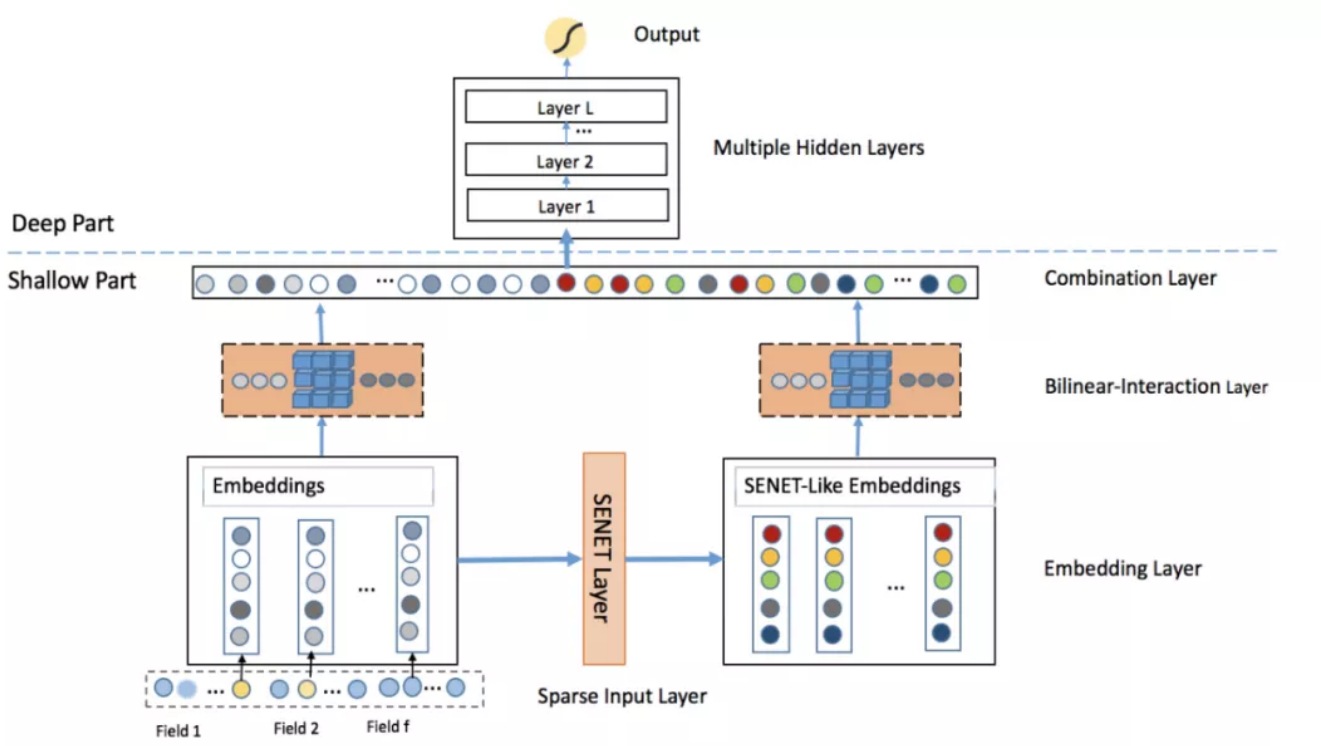


Figure 1: The architecture of our proposed FiBiNET

1.1 背景

本文发表在RecSys 2019，主要通过动态学习不同特征的特征重要性权重，解决CTR预估中对不同场景下不同特征的权重（ReWeight）重定义问题，同时，双线性的使用解决稀疏数据在特征交叉建模时的有效性问题。

1.2 创新

由模型结构图我们可以发现，本文核心结构主要有两个，Embedding Layer中的**SENET Layer** 和 **Bilinear-Interaction Layer** 。

（1）其中 SENET Layer又包含3个步骤，分别是

- 对每个Field用Max Pool或者Mean Pool 操作的Squeeze Step；
- 对每个Field用两层FC层计算特征重要性权重的Excitation；
- 对原始每个Field利用Excitation得到的特征重要性权重重新赋权的ReWeight。

（2）而 Bilinear-Interaction Layer 层提出一种结合Inner Product和Hadamard Product方式，并引入额外参数矩阵W，学习特征交叉。

主要通过3种方式得到交叉向量，分别是

1）Field-All Type：所有特征组交叉时共享额外参数矩阵W；

2）Field-Each Type：每个特征组Field-i 维护一个参数矩阵W-i；

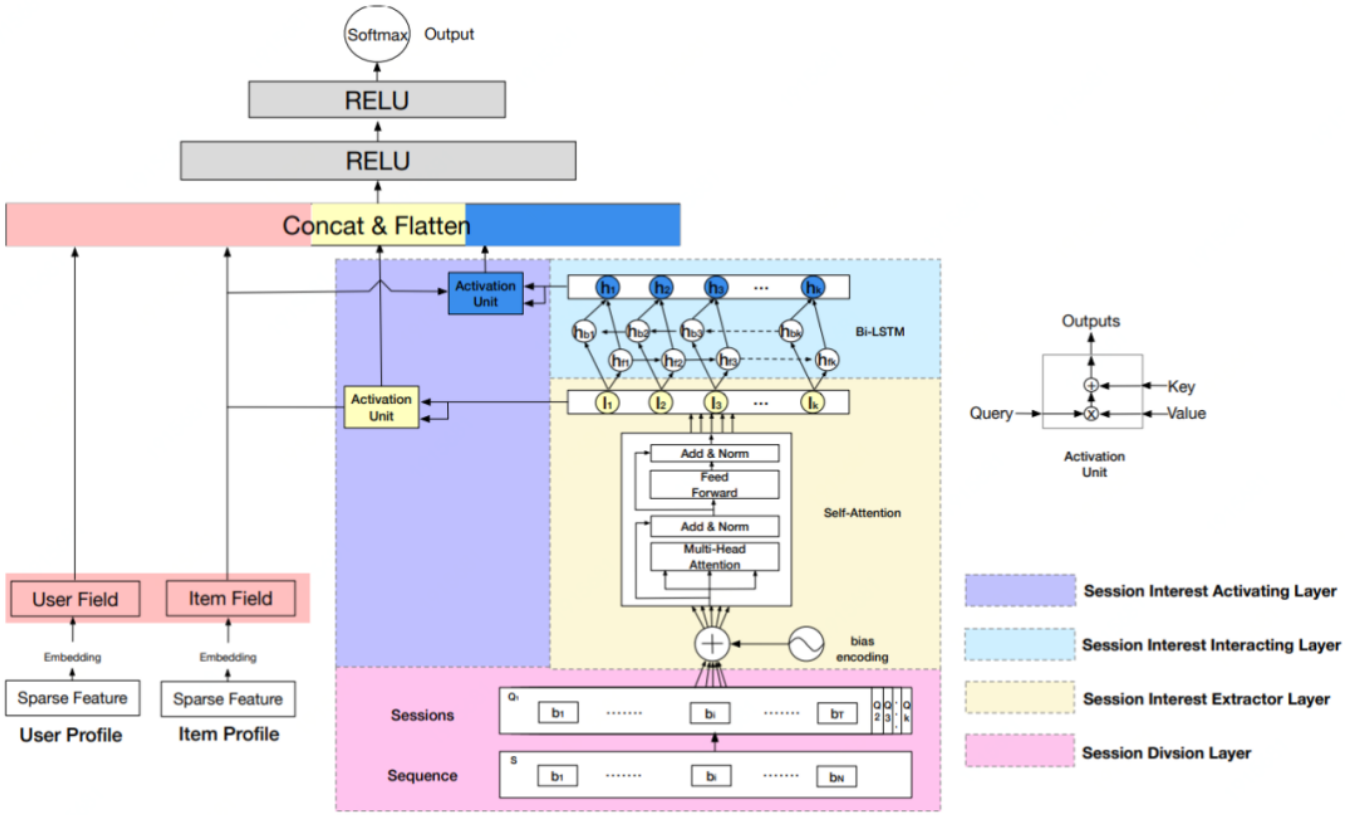
3）Filed-Interaction Type：每对交互特征P(i,j)都有一个参数矩阵W(i,j)。

最后，Bilinear-Interaction Layer 将原始的特征Embedding向量 和 SENET层输出的Embedding向量分别得到交叉向量p和q。

1.3 效果

	Criteo		Avazu	
Model	AUC	Logloss	AUC	Logloss
FNN	0.8057	0.4464	0.7802	0.3800
DeepFM	0.8085	0.4445	0.7786	0.3810
DCN	0.7978	0.4617	0.7681	0.3940
XDeepFM	0.8091	0.4461	0.7808	0.3818
DeepSE-FM-All	**0.8103**	**0.4423**	**0.7832**	**0.3786**

（二）DSIN：利用用户时序行为中兴趣变化进行CTR预估



2.0 前言

在读本文之前，我们需要先搞清楚两个概念，Sequence和Sessions。

基于用户行为Behavior Sequence进行兴趣特征挖掘的方式目前被用于绝大数的CTR任务中。Sequence和Sessions的相同点在于它们都是由Behaviors组成的，但不同的是**Sessions是根据一定的规则将用户的历史点击行为Behavior 进行划分得到的**，也就是说，通过用户的点击时间对Sequence进行划分后，可以得到多个Sessions。

2.1 背景

本文发表在IJCAI 2019，主要通过**将用户的历史点击行为划分为不同session**，然后利用Transformer对每个Session进行学习得到兴趣向量后，使用BiLSTM学习用户在多个Session之间

的兴趣变化，从而更好地完成CTR预估。

2.2 创新

由模型结构图我们可以发现，本文核心结构主要有4个。

1）**Session Division Layer** 完成对用户历史点击行为的划分，以30分钟为划分依据，得到多个Sessions；

2）**Session Interest Extractor Layer** 使用Bias Encoding 方式表征不同Session间的顺序，同时考虑到用户兴趣可能来自不同因素，利用 multi-head self-attention对每个session 建模得到兴趣向量表征；

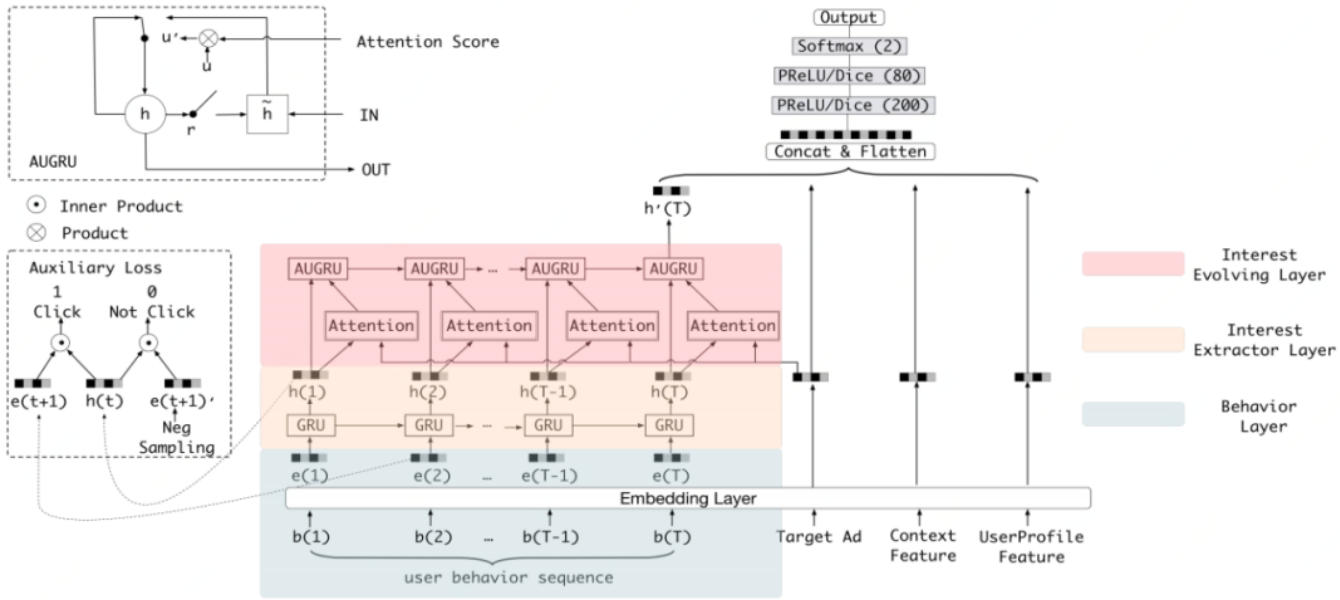
3）**Session Interest Interacting Layer** 在得到用户的兴趣向量表征之后，利用Bi-LSTM学习不同Session之间的由顺序带来的兴趣变化；

4）**Session Interest Activating Layer** 利用注意力机制学习不同Session和Item之间的相关性，混合上下文信息的兴趣信息之后，对距离Item较近的Session赋予更大的权重。

2.3 效果

Model	Advertising	Recommender
YoutubeNet-NO-UB[a]	0.6239	0.6419
YoutubeNet	0.6313	0.6425
DIN-RNN	0.6319	0.6435
Wide&Deep	0.6326	0.6432
DIN	0.6330	0.6459
DIEN	0.6343	0.6473
DSIN-PE[b]	0.6357	0.6494
DSIN-BE-NO-SIIL[c]	0.6365	0.6499
DSIN-BE[d]	**0.6375**	**0.6515**

（三）DIEN：深度兴趣进化网络：



3.0 前言

DIEN在笔者实际应用中，并未复现出原文的效果。这其中的原因可能很多，我想很多读过该论文的读者在进行复现，也可能遇到过类似的问题。但从学习的角度，不影响我们去对论文中一些创新点去进行学习和思考。

3.1 背景

在推荐场景下，用户的兴趣会随着时间（如季节等）和空间（如不同场景等）的变化而发生变化，只通过用户历史数据中的兴趣因素，而不关注兴趣的变化，使得现有的一些模型无法很好的在CTR预估任务中对用户兴趣的变化进行刻画捕捉。DIEN利用双层GRU对用户兴趣序列进行刻画捕捉。

3.2 创新

由模型结构图我们可以发现，本文核心结构主要有3个。

（1）**Behavior Layer** 将用户浏览过的商品按照浏览时间转换成对应的embedding。

（2）**Interest Extractor Layer** 兴趣抽取层利用GRU提取用户兴趣特征。具体地，作者加入了一个二分类模型来计算兴趣抽取的准确性，选取下一个行为作为正样本，也从未点击的样本中选取一个作为负样本，别与抽取出的兴趣表征结合输入到设计的辅助网络中，得到预测结果，并通过logloss计算辅助网络的损失。

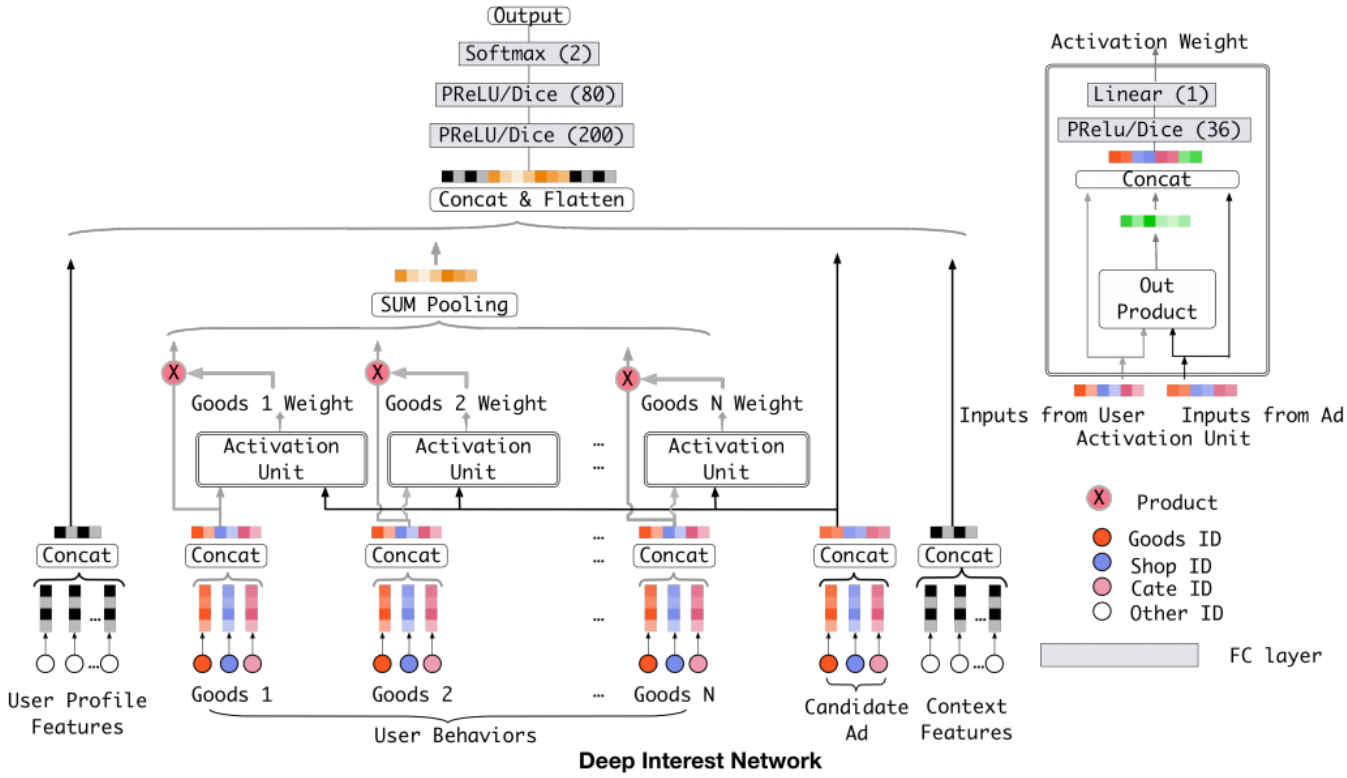
（3）**Interest Evolution Layer** 兴趣演化层，利用Attention（局部激活能力）配合GRU（序列学习能力）的形式，，从时序特征中构建与目标Item相关的兴趣演化特征。

3.3 效果

Table 4: Effect of AUGRU and auxiliary loss (AUC)

Model	Electronics (mean ± std)	Books (mean ± std)
BaseModel	0.7435 ± 0.00128	0.7686 ± 0.00253
Two layer GRU attention	0.7605 ± 0.00059	0.7890 ± 0.00268
BaseModel + GRU + AIGRU	0.7606 ± 0.00061	0.7892 ± 0.00222
BaseModel + GRU + AGRU	0.7628 ± 0.00015	0.7890 ± 0.00268
BaseModel + GRU + AUGRU	0.7640 ± 0.00073	0.7911 ± 0.00150
DIEN	**0.7792 ± 0.00243**	**0.8453 ± 0.00476**

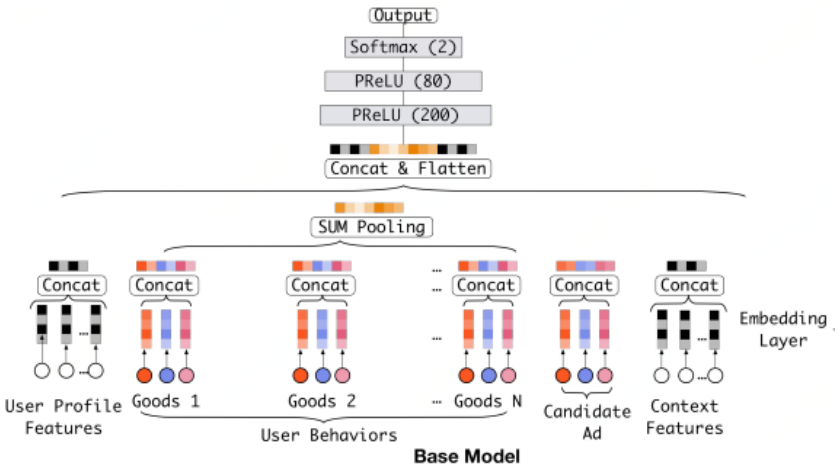
（四）DIN：深度兴趣网络



4.0 前言

DIN（Deep Interest Network）模型在介绍之前，需要对BaseModel（Embedding+MLP）具有一定了解才能更好的理解DIN模型的创新点和它能够在实际生产环境有效的原因。

4.1 背景



在BaseModel中，不同维度的Embedding在最后进行拼接，这样方法通过NN可以拟合高阶非线性关系，但用户的Embedding是不变的，这就很难获取用户多兴趣的场景。

4.2 创新

（1）**Attention机制使用。**其实这点挺有历史的巧合，因为DIN的提出并没有借鉴已有的Attention概念，它的的出发点是看到消费者往往不是在看同一类东西，模型上如何建模用户的多峰兴趣。因为直接显示构建多峰兴趣有些麻烦，所以作者转而用反向激活函数构建一个和预测Item有关的非参数化兴趣表达。巧合的是，当DIN团队构建好模型后发现这和NLP领域刚提出的Attention结构一样，所以论文里就必须写成Attention。作者认为问题的出发点和解题思路更重要，而不是简化的看这是不是Attention。

（2）**DICE激活函数的设计。**由于ReLU和PReLU激活函数都是在0处进行变化，但并非所有输入都会在0处变化，因此文章设计了Dice激活函数，根据每层的输入来自适应的调整激活点的位置。

（3）**GAUC的使用。**相比于常用的AUC，在CTR预估场景中，不同的用户之间存在着差异，这种差异可以理解为一个闲逛的购物者和一个要买小米手机的购物者间的差异。因为AUC表示的是正样本排在负样本前面的概率，所以不能很好地解决不同用户点击率分布的差异。文章提出GAUC作为线下评估指标，通过曝光点击进行加权平均，较少用户之间个性差异对模型造成的影响。

（4）**Adaptive正则化方法。**CTR预估场景下，构造的模型越复杂参数越多，越容易过拟合。实际场景中，存在着大量的长尾数据，这些数据的存在一方面在训练过程中增加了复杂度，另一方面在结果上产生了过拟合。直接去掉这些长尾数据是一种简单的处理方式，但也丢掉了很多信息。因此，DIN文章中给出了自适应正则化调整的方式，对高频减小正则，对低频增大正则。

4.3 效果

Model	MovieLens.		Amazon(Electro).	
	AUC	RelaImpr	AUC	RelaImpr
LR	0.7263	-1.61%	0.7742	-24.34%
BaseModel	0.7300	0.00%	0.8624	0.00%
Wide&Deep	0.7304	0.17%	0.8637	0.36%
PNN	0.7321	0.91%	0.8679	1.52%
DeepFM	0.7324	1.04%	0.8683	1.63%
DIN	**0.7337**	**1.61%**	**0.8818**	**5.35%**
DIN with Dice[a]	**0.7348**	**2.09%**	**0.8871**	**6.82%**

[a] Other lines except LR use PReLU as activation function.

（五）xDeepFM：线性、显式、隐式的组合拳

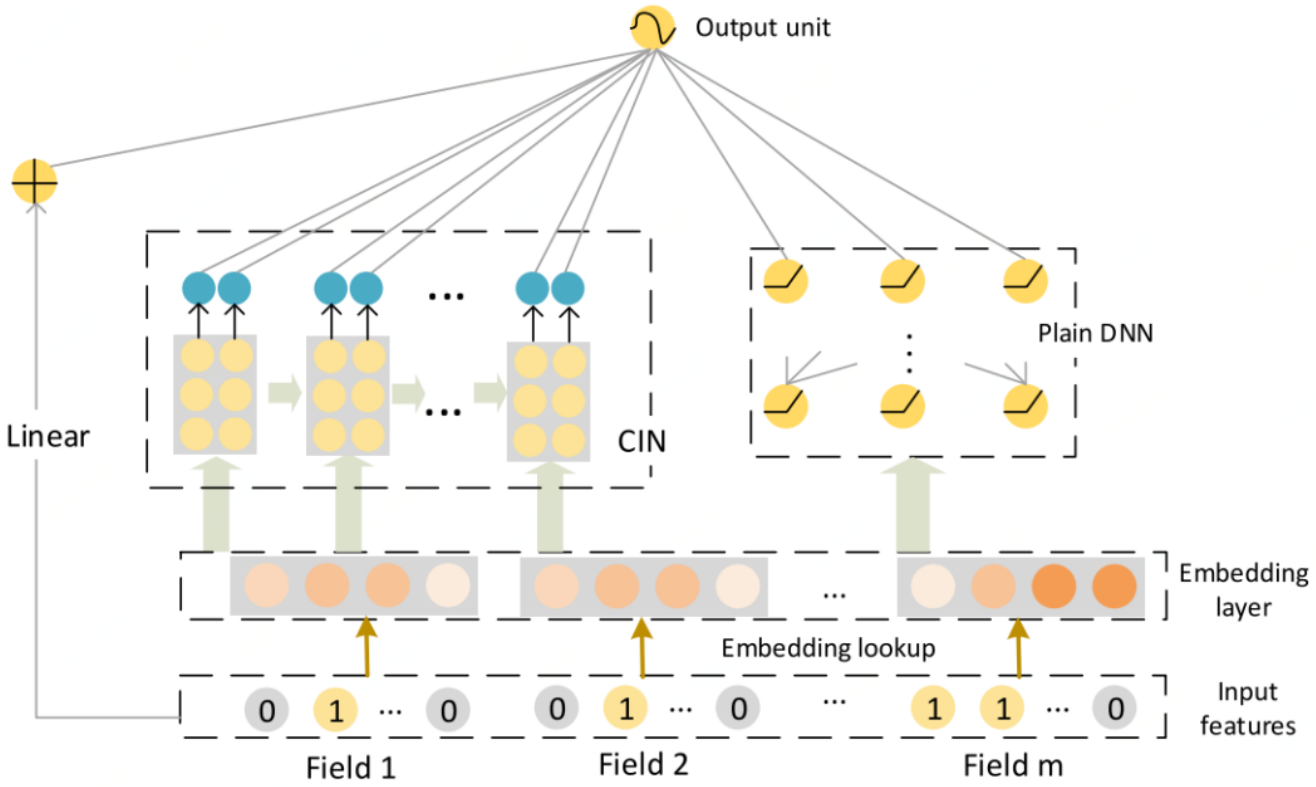


Figure 5: The architecture of xDeepFM.

5.1 背景

使用CIN交叉网络和DNN的双路结构，同时以显式和隐式的方式学习高阶特征。

5.2 创新

（1）**Linear**、CIN与DNN的结构化组合。Linear部分从输入数据中学习线性特征，交叉网络CIN部分由交互（interaction）和压缩（compression）两步通过VectorWise角度学习高阶交叉特征，同时共享Embedding输入的方式让模型更具有适应性；

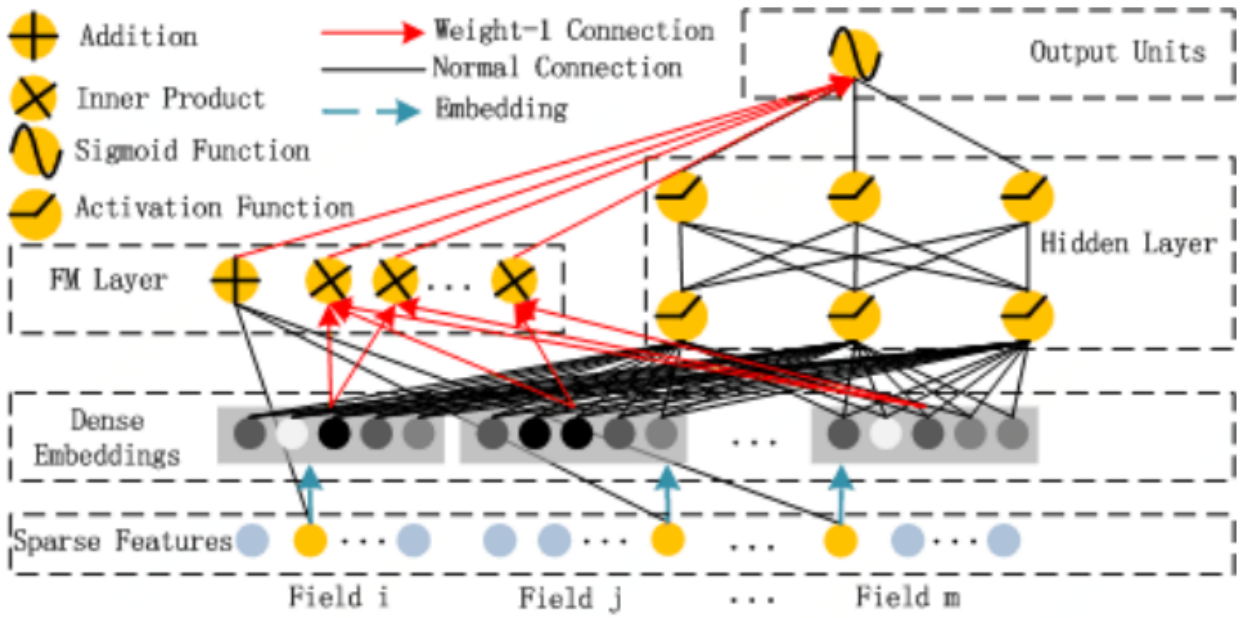
（2）**CIN**。交叉网络通过VectorWise学习高阶特征之间的交互，特征交互部分在Embedding的每个维度上进行外积计算。特征压缩部分借鉴了卷积和池化的方式。同时最后将多层CIN的输出进行拼接。

（3）**DNN**。通过BitWise隐式的学习高阶特征。

5.3 效果

	Criteo			Dianping			Bing News		
Model name	AUC	Logloss	Depth	AUC	Logloss	Depth	AUC	Logloss	Depth
LR	0.7577	0.4854	-,-	0.8018	0.3608	-,-	0.7988	0.2950	-,-
FM	0.7900	0.4592	-,-	0.8165	0.3558	-,-	0.8223	0.2779	-,-
DNN	0.7993	0.4491	-,2	0.8318	0.3382	-,3	0.8366	0.2730	-,2
DCN	0.8026	0.4467	2,2	0.8391	0.3379	4,3	0.8379	0.2677	2,2
Wide&Deep	0.8000	0.4490	-,3	0.8361	0.3364	-,2	0.8377	0.2668	-,2
PNN	0.8038	0.4927	-,2	0.8445	0.3424	-,3	0.8321	0.2775	-,3
DeepFM	0.8025	0.4468	-,2	0.8481	0.3333	-,2	0.8376	0.2671	-,3
xDeepFM	**0.8052**	**0.4418**	3,2	**0.8639**	**0.3156**	3,3	**0.8400**	**0.2649**	3,2

（六）DeepFM：将Wide&Deep模型的LR部门替换为FM



6.1 背景

解决在高纬度特征的情况下，人工特征工程的高成本和无法实现的问题。同时解决FM只能获取二阶特征，无法获取高阶特征的问题。解决传统DNN在隐式交叉方式在高稀疏特征无法很好获取表征的问题。

6.2 创新

（1）FM部分通过显式向量和点积的方式学习二阶交叉特征，配合DNN部分使模型对高阶特征组合能够更好的进行特征提取。

（2）Deep部分将原始稀疏特征表示映射为稠密表示，同时FM和DNN部分共享Embedding层特征表达。

6.3 效果

	Company*		Criteo	
	AUC	LogLoss	AUC	LogLoss
LR	0.8641	0.02648	0.7804	0.46782
FM	0.8679	0.02632	0.7894	0.46059
FNN	0.8684	0.02628	0.7959	0.46350
IPNN	0.8662	0.02639	0.7971	0.45347
OPNN	0.8657	0.02640	0.7981	0.45293
PNN*	0.8663	0.02638	0.7983	0.45330
LR & DNN	0.8671	0.02635	0.7858	0.46596
FM & DNN	0.8658	0.02639	0.7980	0.45343
DeepFM	**0.8715**	**0.02619**	**0.8016**	**0.44985**

（七）AFM：FM串行结构加入Attention机制：

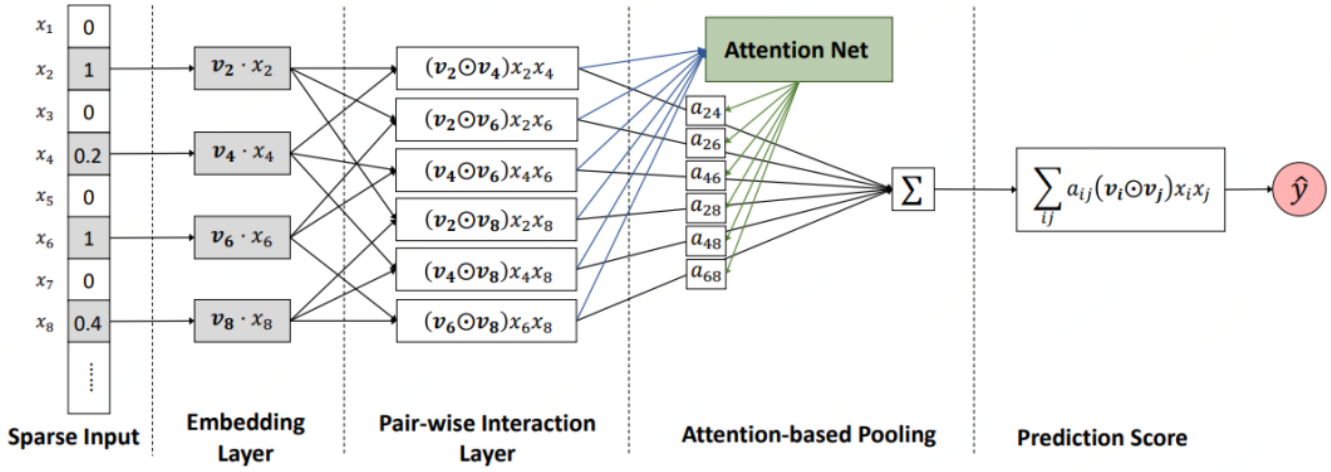


Figure 1: The neural network architecture of our proposed Attentional Factorization Machine model.

7.1 背景

串行结构，将FM的输出作为后续神经网络的输入，利用FM解决稀疏特征问题及浅层交互特征，利用深度网络解决深层交互特征的获取。

7.2 创新

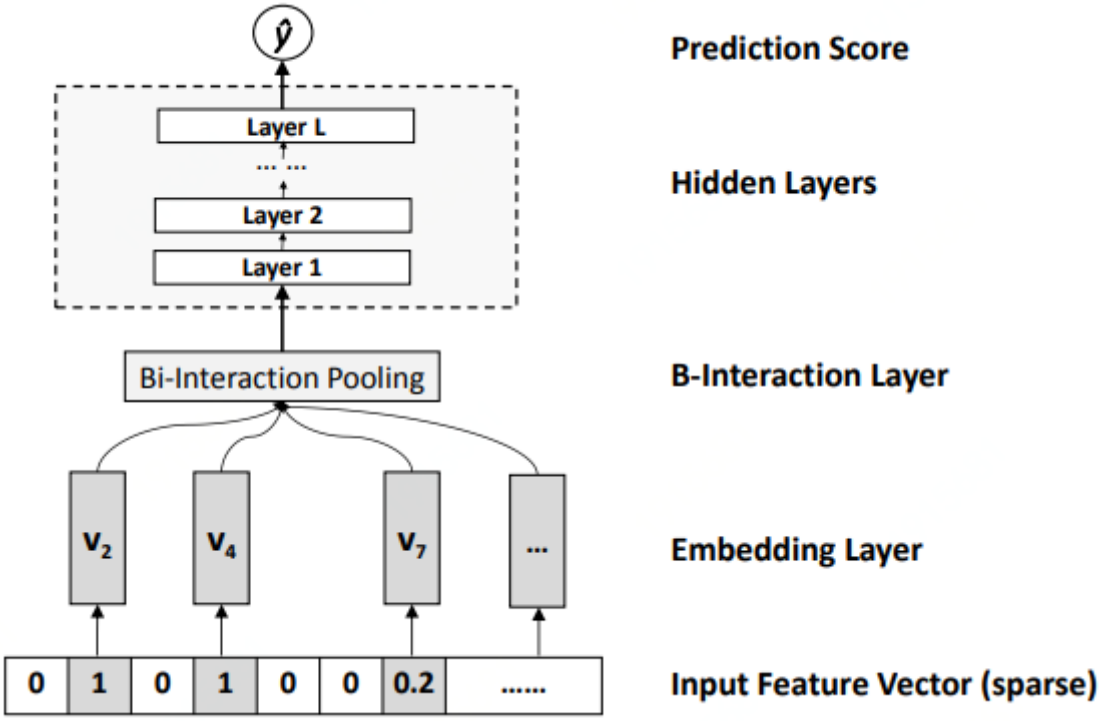
（1）和NFM一样，文章使用**Pair-wise Interaction**层，同时在Sum Pooling之后接了一个FC层。该方法从思路上具有一定的创新性。但实际中应用却存在一定的问题，较高的时间复杂度在实际复现中无法通过矩阵实现。

（2）**Attention-based Pooling Layer** 关注不同的交叉特征和目标之间的关系，通过加权的方式处理不同特征对结果的贡献度，同时利用MLP进一步处理训练数据中未出现样本的的评估问题，从而达到泛化模型的目的。

7.3 效果

	Frappe		MovieLens	
Method	**Param#**	**RMSE**	**Param#**	**RMSE**
LibFM	1.38M	0.3385	23.24M	0.4735
HOFM	2.76M	0.3331	46.40M	0.4636
Wide&Deep	4.66M	0.3246	24.69M	0.4512
DeepCross	8.93M	0.3548	25.42M	0.5130
AFM	**1.45M**	**0.3102**	**23.26M**	**0.4325**

（八）NFM：FM串行引入DNN并加入Bi-interaction Pooling操作



8.1 背景

在FM基础上，串行加入DNN模型，并使用Bi-interaction Pooling操作对二阶交叉特征进行处理，解决传统FM作为线性模型表达有限的问题和对高阶交叉特征学习不充分的问题。

8.2 创新

Bi-interaction Pooling操作作为NFM的核心，本质是Pooling操作。在实际场景中，我们发现直接对Embedding向量进行Concat拼接之后接MLP进行模型训练在获取二阶交叉特征时的效果是一般的。Bi-interaction Layer与FM二阶交叉相比，没有引入额外的参数，将Embedding向量进行两两交叉相乘后对所有对应元素求和的方式，以线性复杂度具有良好的实际应用价值，同时在曾经的**CTR竞赛中一度成为王者模型**。

8.3 效果

	Frappe				MovieLens			
	Factors=128		Factors=256		Factors=128		Factors=256	
Method	**Param#**	**RMSE**	**Param#**	**RMSE**	**Param#**	**RMSE**	**Param#**	**RMSE**
LibFM [28]	0.69M	0.3437	1.38M	0.3385	11.67M	0.4793	23.24M	0.4735
HOFM	1.38M	0.3405	2.76M	0.3331	23.24M	0.4752	46.40M	0.4636
Wide&Deep [9]	2.66M	0.3621	4.66M	0.3661	12.72M	0.5323	24.69M	0.5313
Wide&Deep (pre-train)	2.66M	0.3311	4.66M	0.3246	12.72M	0.4595	24.69M	0.4512
DeepCross [31]	4.47M	0.4025	8.93M	0.4071	12.71M	0.5885	25.42M	0.5907
DeepCross (pre-train)	4.47M	0.3388	8.93M	0.3548	12.71M	0.5084	25.42M	0.5130
NFM	**0.71M**	**0.3127****	**1.45M**	**0.3095****	**11.68M**	**0.4557***	**23.31M**	**0.4443***

（九）FNN：使用FM参数进行Embedding初始化

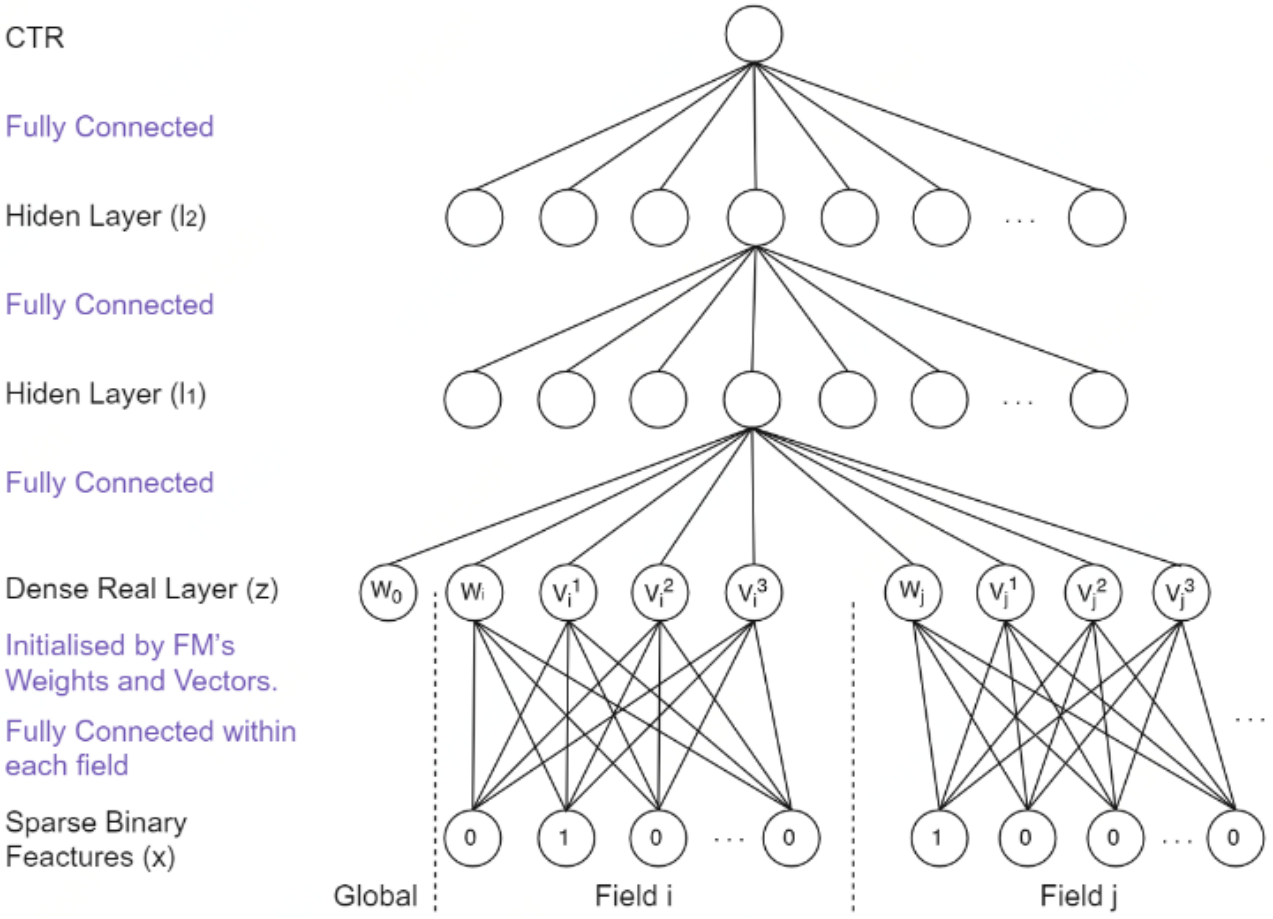


Fig. 1. A 4-layer FNN model structure.

9.1 背景

传统的FM受限于计算复杂度，对特征只能进行二阶交叉，对更高阶的特征交叉却无能为力，同时实践证明，高阶的特征交叉组合方式对CTR预估模型具有更好的效果，为了解决这个问题，引入DNN。其实FNN的思想比较简单，从模型结构图里我们就可以看出，主要是利用FM后直接接DNN网络，利用FM参数初始化Embedding层。

9.2 创新

二段式训练方式，使用FM层模型的参数初始化Embedding，替换随机初始化方式，然后接DNN进行高阶特征提取。但这种方式优点和缺点都比较明显，优点是使用预训练初始化的方式，降低了训练的不稳定性。缺点是二阶段训练方式在应用中便利性不够。

9.3 效果

Table 1. Overall CTR estimation AUC performance.

	LR	FM	FNN	SNN-DAE	SNN-RBM
1458	70.42 %	70.21 %	**70.52 %**	70.46 %	70.49 %
2259	69.66 %	69.73 %	**69.74 %**	68.08 %	68.34 %
2261	62.03 %	60.97 %	62.99 %	**63.72 %**	**63.72 %**
2997	60.77 %	60.87 %	61.41 %	**61.58 %**	61.45 %
3386	80.30 %	79.05 %	**80.56 %**	79.62 %	80.07 %
all	68.81 %	68.18 %	**70.70 %**	69.15 %	69.15 %

（十）DCN：替换Wide&Deep模型的Wide部分为Cross

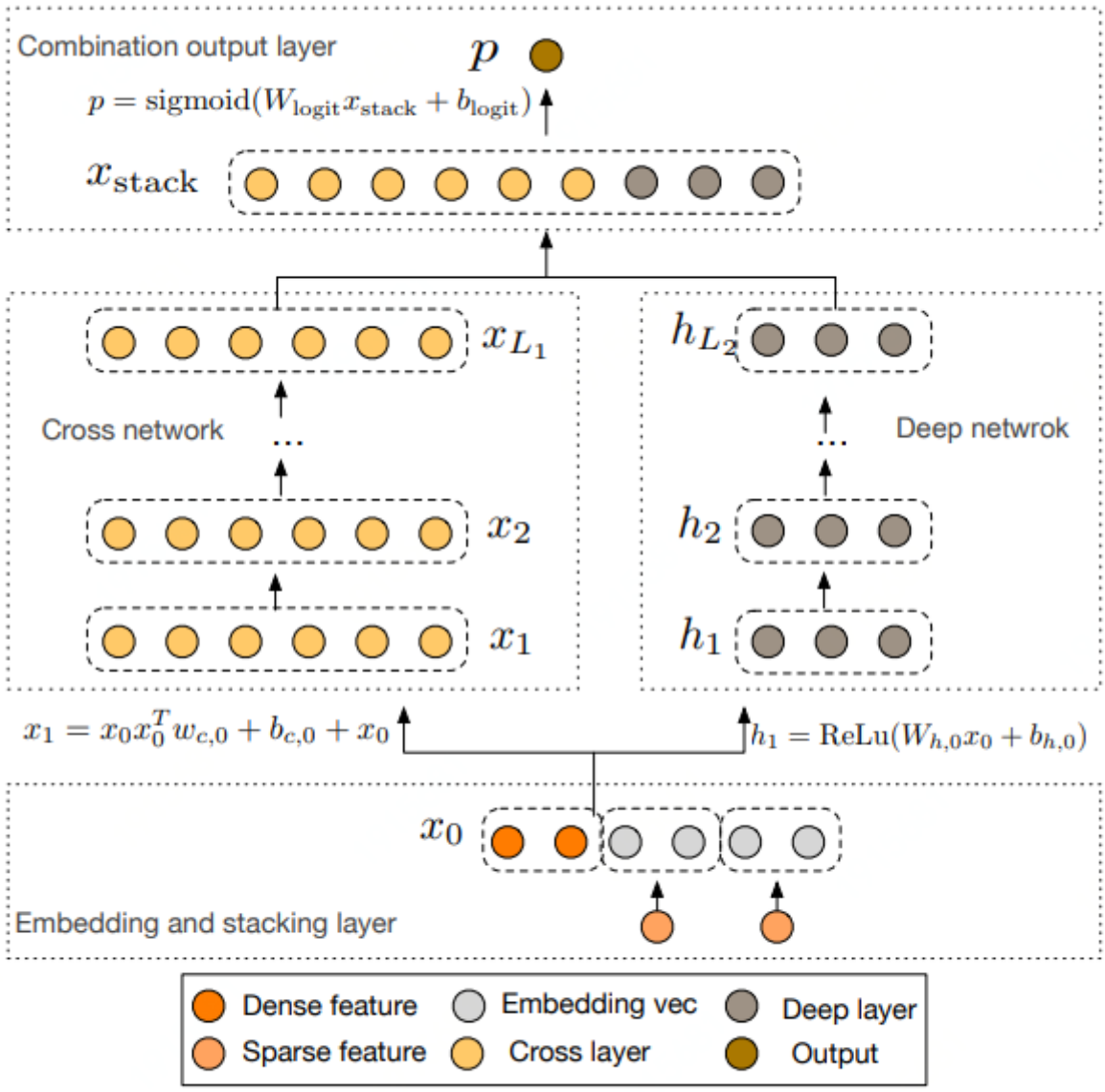


Figure 1: The Deep & Cross Network

10.1 背景

通过对比Wide&Deep和DCN网络的架构图，不难发现后者是对前者Wide部分进行了替换，提出CrossNetwork替换Wide用于自动化特征交叉，从而无需人工特征工程。

10.2 效果

模型主要由**Embedding and Stacking layer、Cross Network、Deep Network、Combination output layer** 组成。但相较于其他模型，他的创新点主要在Cross Network 部分，它的设计理念是通过参数共享的方式减少向量压缩变换时产生参数量过多的情况，从而减少模型的过拟合，增强模型的泛化能力。同时Cross Network的方式将模型复杂度降为层级线性增长，具有很好的应用价值。

10.3 效果

Model	DCN	DC	DNN	FM	LR
Logloss	**0.4419**	0.4425	0.4428	0.4464	0.4474

（十一）PNN：在Embedding MLP模式中设计加入Product Layer

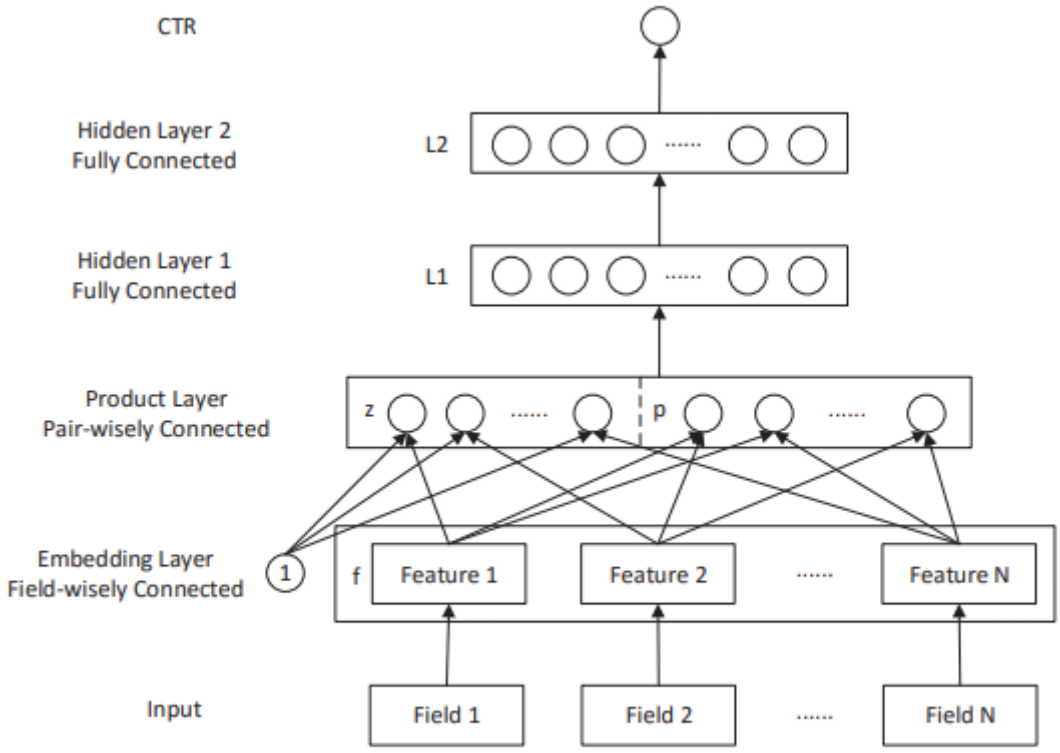


Fig. 1: Product-based Neural Network Architecture.

11.1 背景

传统Embedding+MLP的方式并不能很好对高阶交叉特征进行获取，同时FNN用FM初始化参数接DNN的方式也并不完美，针对这些缺点PNN进行了改进，并不只是关注高阶交叉特征，通过设计Product层对特征进行交叉组合，改变原有对特征进行ADD的操作方式。

11.2 创新

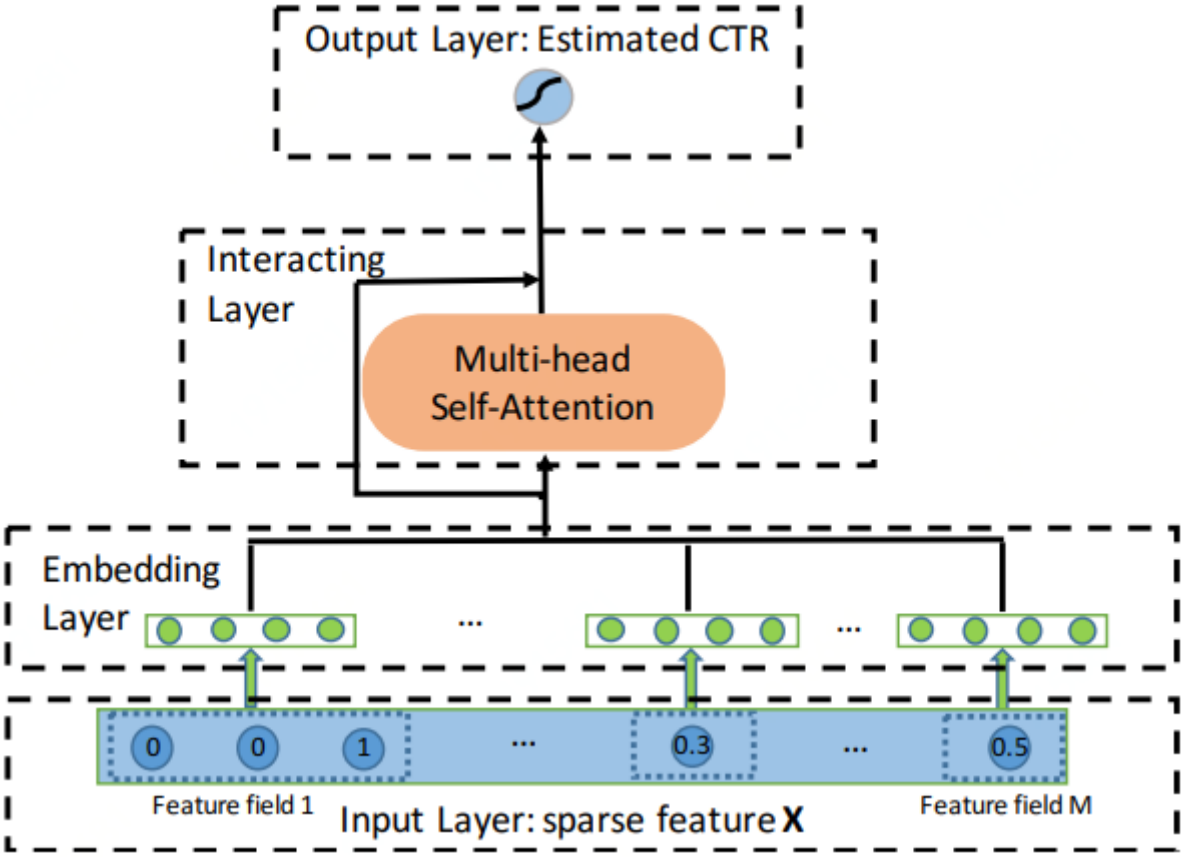
Product Layer的设计，是本文的核心结构。Product层利用内积（Inner PNN）和外积（Outer PNN）两种方式实现对特征的交叉组合。其中，IPNN本质上利用共享参数来减少参数量，采用一阶矩阵分解来近似矩阵结果的同时牺牲了一定的精度，保证计算开销控制在可接受的范围内。OPNN，从公式层面，该方法的时间空间复杂度比IPNN更高，作者使用了Sum Pooling的方式来降低复杂度，但同时也造成了精度的严重损失。

11.3 效果

TABLE I: Overall Performance on the Criteo Dataset.

Model	AUC	Log Loss	RMSE	RIG
LR	71.48%	0.1334	9.362e-4	6.680e-2
FM	72.20%	0.1324	9.284e-4	7.436e-2
FNN	75.66%	0.1283	9.030e-4	1.024e-1
CCPM	76.71%	0.1269	8.938e-4	1.124e-1
IPNN	**77.79%**	**0.1252**	**8.803e-4**	**1.243e-1**
OPNN	77.54%	0.1257	8.846e-4	1.211e-1
PNN*	77.00%	0.1270	8.988e-4	1.118e-1

（十二）AutoInt：利用多头注意力构造高阶特征

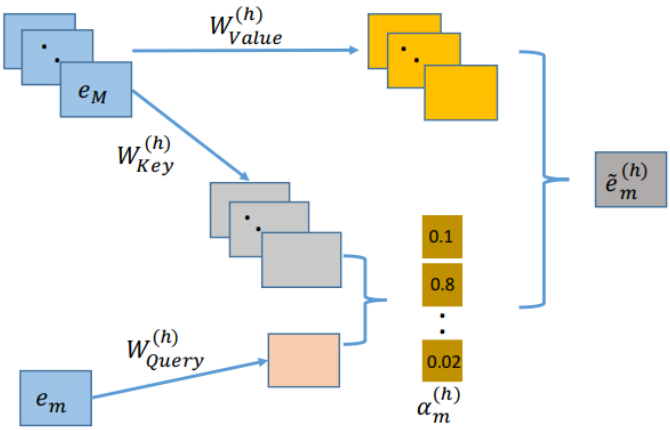


12.1 背景

CTR预估领域面临着诸多挑战，输入特征的稀疏高维问题，高阶特征的计算复杂度问题等。本文将数值特征和类别特征映射到同一个低维空间，利用带残差连接的多头注意力机制显式的进行交

叉特征获取，提出了一种能够自学习特征高阶交叉的方法。

12.2 创新



Interacting Layer 是本篇论文中最核心的创新点，借鉴了NLP问题中的Multi-head Self-Attention方法，利用Key-Value Attention，每个Attention Head对应三个转换矩阵，Query、Key、Value。本文利用内积的方式计算每个特征与其他特征的相似度，然后通过计算softmax归一化注意力分布后，加权得到新特征。以上步骤为一层Attention，作者简单的拼接多个Attention head的输出，引入标准的残差连接作为最终输出，构造更高阶的组合特征。

12.3 效果

Model	Criteo		Avazu		KDD12		MovieLens-1M	
	AUC	Logloss	AUC	Logloss	AUC	Logloss	AUC	Logloss
Wide&Deep (LR)	0.8026	0.4494	0.7749	0.3824	0.7549	0.1619	0.8300	0.3976
DeepFM (FM)	0.8066	0.4449	0.7751	0.3829	0.7867	0.1549	0.8437	0.3846
Deep&Cross (CN)	0.8067	0.4447	0.7731	0.3836	0.7869	0.1550	0.8446	0.3809
xDeepFM (CIN)	0.8070	0.4447	0.7768	0.3832	0.7820	0.1560	0.8467	0.3800
AutoInt+ (ours)	**0.8080**	**0.4437**	**0.7771**	**0.3811**	**0.7892**	**0.1544**	**0.8486**	**0.3757**

引用

1）FiBiNET：Combining Feature Importance and Bilinear feature Interaction for Click-Through Rate Prediction

2）Deep Session Interest Network for Click-Through Rate Prediction

3）AutoInt_Automatic Feature Interaction Learning via Self-Attentive Neural Networks

4）Deep Interest Evolution Network for Click-Through Rate Prediction

5）Deep Interest Network for Click-Through Rate Prediction

6）Product-based Neural Networks for User Response Prediction

7）Deep & Cross Network for Ad Click Predictions

8）Deep Learning over Multi-field Categorical Data_A Case Study on User Response Prediction

9）Attentional Factorization Machines - Learning the Weight of Feature Interactions via Attention Networks