基于融合结构的在线广告点击率预测模型

刘梦娟 曾贵川 岳 威 刘 瑶 秦志光

(电子科技大学信息与软件工程学院 成都 610054)

摘 要 点击率预测作为推荐系统和在线广告的关键环节,在学术界和工业界均受到了极大的关注.论文首先对几种典型的点击率预测模型进行研究,然后探索了基于融合结构的深度学习方法,并在此基础上提出一种基于融合结构的点击率预测模型,该模型能够灵活融合不同结构的深度神经网络来分别学习原始高维稀疏特征的高阶表示,从而使点击率预测模型能够利用更丰富的高阶特征信息.论文利用真实数据集来评价模型的预测性能,实验结果显示,基于融合结构的深度学习预测模型,能够比传统的点击率预测模型以及最新的基于深度学习的预测模型获得更好的性能.

关键词 点击率预测;逻辑回归;因子分解机;深度神经网络;融合结构

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2019.01570

A Hybrid Network Based CTR Prediction Model for Online Advertising

LIU Meng-Juan ZENG Gui-Chuan YUE Wei LIU Yao QIN Zhi-Guang

(*Department of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu* 610054)

Abstract As the key component of recommender system and online advertising, the click-through rate (CTR) prediction has received great attention in both the academia and the industry. The most common approaches to CTR prediction are regarding it as a regression prediction task in machine learning. At beginning, simple models like logistic regression (LR) and factorization machine (FM) are used to do predictions, however the prediction performances are not so good because only low-order feature interactions are explored. Therefore, models with stronger ability of feature representation learning are developed, for example, a Factorization-machine supported Neural Network (FNN) and a Product-based Neural Network (PNN), which are promising to exploit deep neural networks to learn sophisticated and selective feature interactions. The major downside of FNN and PNN is that they focus more on high-order feature interactions while capture little low-order interactions. In order to make full use of low- and high-order feature interactions, some hybrid architectures are proposed, containing both a shallow component and a deep component. In this paper, we firstly study several typical CTR prediction models, especially the deep learning models based on hybrid architectures, to describe the development process of CTR prediction; and then, inspired by existing works, a new click-through rate prediction model based on a hybrid network is proposed—DPSN (Deep & Product supported Stacking Network). The new model can integrate different deep neural networks (DNNs) to learn the high-order representation of original high-dimensional sparse features respectively, which enables the prediction model to take advantage of more abundant information of high-order feature interactions. In addition, we also design a

收稿日期:2018-04-29;在线出版日期:2019-01-25.本课题得到国家自然科学基金(61202445,61502087)、中央高校基本业务费项目(ZYGX2016J096)资助.**刘梦娟**,博士,副教授,主要研究方向为数据挖掘、广告计算、机器学习.E-mail:mjliu@uestc.edu.cn.**曾贵川**,硕士研究生,主要研究方向为广告计算、机器学习.**岳 威**,硕士研究生,主要研究方向为广告计算、机器学习.**刘 瑶**,博士,副教授,主要研究方向为数据挖掘、机器学习.**秦志光**,博士,教授,博士生导师,主要研究领域为数据挖掘、网络安全.

new embedding layer for DPSN, where nodes come from not only the embedding vector but also the weight of each feature, which are both pre-trained by FM model. To our best knowledge, this is the first attempt to improve the prediction performance by adding few weight nodes in the embedding layer. Furthermore, a simplified analysis of the parameter complexity is given; meanwhile, the convergence of the DPSN model is analyzed and proved. We evaluate the prediction performance of the proposed model based on two real-world data sets, iPinYou and Criteo, by using LogLoss and AUC metrics. In the first and second experiments, we verify the convergence of the DPSN model and illustrate the performance improvements of FNN and PNN by adding the weight nodes of each feature in the embedding layer. In the third experiment, we analyze the influences of different model parameters on the prediction performance of DPSN, including the number of hidden layers, hidden layer nodes, activation function, and embedded vector dimension. The fourth experiment is used to evaluate the effects of the new embedding layer on the prediction performance of the DPSN model. The fifth and sixth experiments respectively compare the influences of different architectures and negative sampling ratios on DPSN prediction performance. The last experiment is to compare the performance of the DPSN model with other typical CTR prediction models, and the experimental results demonstrate that the new model has better performance than major stat-of-the-art models on LogLoss metrics and AUC metrics.

Keywords　click-through rate; logistic regression; factorization machine; deep neural network; hybrid network

1　引　言

随着互联网的广泛普及以及大数据技术的快速发展，广告商利用互联网平台进行广告精准营销成为可能. 与传统广告相比，在线广告在覆盖范围、灵活性、针对性、成本和效果评估等方面拥有得天独厚的优势，而且已经发展成为具有数十亿美元的产业[1]. 在线广告的主要目标之一是在给定预算的情况下，最大化广告商的收益，例如最大化广告的点击次数或者转换次数[2]. 因此，在线广告的一个重要环节是对将广告投放到一个曝光机会(ad impression)的用户点击概率进行预测，应尽可能将广告投放到预测点击率高的曝光机会，这就是点击率(Click-Through Rate，CTR)预测问题.

CTR 预测是一个典型的回归问题. 目前工业界应用最广泛的预测方法是利用逻辑回归(Logistic Regression，LR)来学习 CTR 预测模型[2-4]. LR 的优点是简单、非常容易实现大规模实时并行处理，但是线性模型的学习能力有限，不能捕获高阶特征携带的信息(非线性信息)[5]，从而限制了 LR 的预测性能. 为此文献[6]提出可以利用 Poly2 模型[7]来进行 CTR 预测，该模型不仅考虑了一阶特征携带的信息，而且考虑了二阶特征组合携带的信息，但是由于 CTR 预测模型的输入特征通常是经过独热(one-hot)编码[8]后的高维稀疏二值化特征向量，将特征进行两两组合的计算复杂度会变得非常大，导致学习效率大幅降低. 因此在工业界，更多采用的是特征工程来完成手动的特征组合工作，以捕获特征间的高阶信息.

近几年，非线性模型在 CTR 预测中逐渐获得关注. 例如因子分解机模型(Factorization Machine，FM)[9-10]通过将高维稀疏特征映射到低维稠密向量中，并通过向量内积的方式来学习特征两两之间的隐含信息，从而大幅减少了特征两两组合导致的计算复杂度；FM 的缺陷在于每个特征都只学习一个隐含向量，在与其它特征进行组合时，同一个特征产生的影响力是相同的，而事实上当与不同特征域的特征组合时，隐含向量可能表现出不同的分布. 为此，文献[6]进一步提出了特征域相关的因子分解机模型 FFM(Field-aware Factorization Machines)，其基本思想是将特征分割为若干域，每个特征将针对不同特征域学习不同的隐含向量，利用 FFM 方案，作者分别在 Criteo 和 Avazu 举办的全球 CTR 预测大赛中获得了冠军[6]. Poly2、FM、FFM 都是在 LR 基础上增加对二阶特征组合的权重自动学习的模

型.除此之外,Facebook 的研究人员提出了另一种筛选特征和特征组合的方式,称为 GBDT+LR 方案[11],该方案利用 GBDT(Gradient Boost Decision Tree)来帮助筛选有区分度的特征和组合特征,作为 LR 模型的输入,从而增强 LR 的非线性学习能力.

深度学习在计算机视觉[12]、语音识别[13]、自然语言处理[14]等领域取得巨大成功,其在探索特征间高阶隐含信息的能力也被应用到了 CTR 预测中.文献[5]提出了 FNN(Factorization Machine supported Neural Network)模型,该模型利用一个带嵌入层(Embedding layer)的深度神经网络(Deep Neural Network,DNN)来完成点击率预测,其特点是通过 FM 模型预先训练得到每个特征域的稠密隐含向量,将隐含向量作为 DNN 的输入进行训练.文献[15]仍然使用 DNN 来预测点击率,不同之处在于 DNN 的结构中引入了一个 Product 层,DNN 的输入单元不仅包括每个特征域的隐含向量,还包括任意两个特征域向量的积运算,这种方案称为 PNN(Product-based Neural Network).

FNN 和 PNN 充分利用了 DNN 对特征高阶隐含信息的表示能力,但忽略了一阶特征携带的信息,而实验证明一阶特征对于 CTR 预测也是非常重要的,为此 Google 的研究人员在文献[16]中提出一种深度学习融合结构 Wide & Deep,该结构将线性模型和深度学习模型进行巧妙地融合,不仅考虑了低阶特征携带的信息,也考虑了高阶特征之间的交互信息,因此能够获得超过 FNN 和 PNN 的预测性能.文献[17]在 Wide & Deep 的基础上,将线性模型(Wide)替换为 FM 模型,从而提出 DeepFM.

本文借鉴 Wide & Deep 的思路,设计了一个基于融合结构的深度神经网络来建立 CTR 预测模型,称为 DPSN(Deep & Product supported Stacking Network),如图 1 所示.该结构由 Deep Network、Product Network 和 Stacking Network 三部分组成,其中 Deep Network 和 Product Network 分别用于学习特征间的高阶表示,Deep Network 是一个简单的 DNN,Product Network 则借鉴了 PNN 在 DNN 的输入位置增加一个 Product 层;最终通过 Stacking Network 将前两个部分的参数进行联合训练,以有效捕获不同特征表示之间的关系,从而得到最终的 CTR 预测值.与已有的深度学习方案类似,DPSN 中也引入了一个嵌入层,用于将原始高维稀疏的二值化特征映射为固定维度的低维稠密向量,作为 Deep Network 和 Product Network 的共同输入.为了验证 DPSN 的预测性能,本文利用 iPinYou 和 Criteo 的公开数据集完成了大量实验,

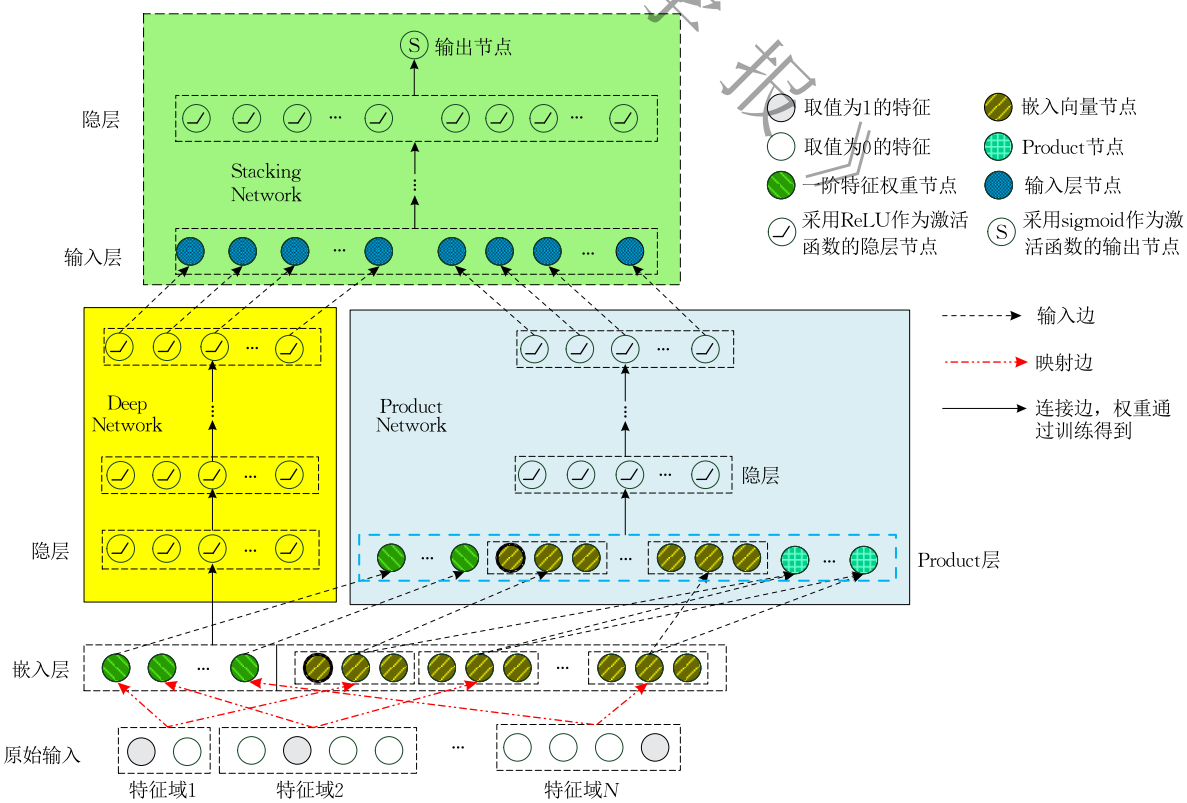


图 1 DPSN 采用的融合结构

结果显示在 LogLoss 和 AUC 指标上，DPSN 能够比传统的点击率预测模型以及已有的基于深度学习的预测模型获得更好的性能.

2　相关工作

相比于传统的浅层学习，DNN 在特征学习方面表现出巨大的潜力[5]，因此，近两年将 DNN 用于 CTR 预测已经成为一种研究趋势. 本节将对几种典型的基于 DNN 的 CTR 预测模型进行介绍和对比.

在此之前，首先对 CTR 预测模型原始特征的预处理方法以及在 DNN 方案中常用的映射方法进行简单介绍[5,18]. 在 CTR 预测中，使用的特征主要是分类特征(categorical features)，例如用户的性别(Gender)、所在的城市(City)等，分类特征不能直接用于预测计算，因此通常使用独热(one-hot)编码对分类特征进行预处理，例如 Gender 特征有两种可能取值(Female/Male)，因此 Gender 特征可编码为 2 比特，[0,1]表示 Female，[1,0]表示 Male；City 特征有三种可能取值(Beijing/Shanghai/Chengdu)，因此 City 特征可编码为 3 比特，分别对应[0,0,1]，[0,1,0]，[1,0,0]. 例如，一个位于北京的男性用户，其编码后的原始特征向量为

$$\underbrace{[1,\ 0]}_{\text{Gender=Male}}\ \underbrace{[0,\ 0,\ 1]}_{\text{City=Beijing}}.$$

在 CTR 预测中，通常将独热编码后的每个比特称为一个特征，例如[1,0]中第一个比特表示男性特征，第二个比特表示女性特征，样本中出现的特征取值为 1，其余取值为 0. 因此对于一个样本，编码后的特征向量是一个超高维度的稀疏向量，如果将该特征向量直接输入到 DNN 中会使得需要学习的参数非常多，产生巨大的计算开销. 因此在基于 DNN 的 CTR 预测模型中，通常会在输入层和第一个隐藏层之间增加一个嵌入层，用于降低 DNN 的输入单元数. 这里引入了特征域(field)和嵌入向量(embedding vector)的概念，首先将编码后的特征按照其物理属性划分为若干特征域，例如$[b^1,b^2]$表示 Gender 特征域，$[b^1,b^2,b^3]$表示 City 特征域，在每个样本中，每个特征域中只有一个特征的值为 1，其余为 0. 假设特征域 $\boldsymbol{c}$ 在整个样本集中有 M 种取值可能，则独热编码后的表示如式(1)，$\boldsymbol{c}$ 是一个由二值化元素组成的向量，每个元素 $b^i\in\{0,1\}$.

$$\boldsymbol{c}=(b^1,b^2,\cdots,b^M),\ \sum_{i=1}^{M}b^i=1 \tag{1}$$

如前所述，为了减少 DNN 中的输入单元数，在基于 DNN 的 CTR 预测模型中，需要将独热编码后的每个特征映射为一个固定维度的嵌入向量，再将所有嵌入向量拼接起来作为输入. 假设独热编码后的特征数为 n，每个嵌入向量的维度为 D，特征的嵌入向量 $\boldsymbol{v}$ 可以写为一个矩阵，如式(2)所示.

$$\boldsymbol{v}=\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\\ \vdots\\ \boldsymbol{v}_n\end{bmatrix}=\begin{bmatrix}v_1^1 & v_1^2 & \cdots & v_1^D\\ v_2^1 & v_2^2 & \cdots & v_2^D\\ \vdots & \vdots & \ddots & \vdots\\ v_n^1 & v_n^2 & \cdots & v_n^D\end{bmatrix} \tag{2}$$

对于一个样本，由于属于相同特征域的特征只有 1 个有取值，因此 DNN 模型的输入单元数为 $N\times D$，这里 N 表示特征向量中特征域的个数. 图 2 展示了将特征映射为嵌入向量作为输入的一个示例. 假设独热编码后的样本特征为[1,0,0,0,1]，其中前两个比特为 Gender 特征域，后三个比特为 City 特征域，嵌入向量的维度为 2，可将男性特征和 Beijing 特征映射为两个嵌入向量，假设为[0.2,0.8]和[0.6,0.4]，最后将两个嵌入向量拼接起来作为输入，因此映射后的嵌入单元的数量为 4. 需要说明的是不同的特征对应的嵌入向量是不同的，例如 Male 和 Female 分别对应的是不同的嵌入向量. 此外，如果样本中包含有数值特征，在 CTR 预测中通常将数值特征利用分箱技术转化为分类特征，再按照分类特征的预处理方法来编码.

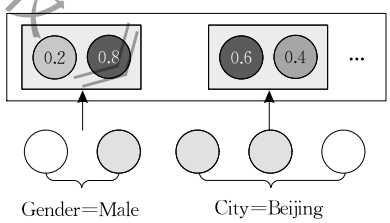
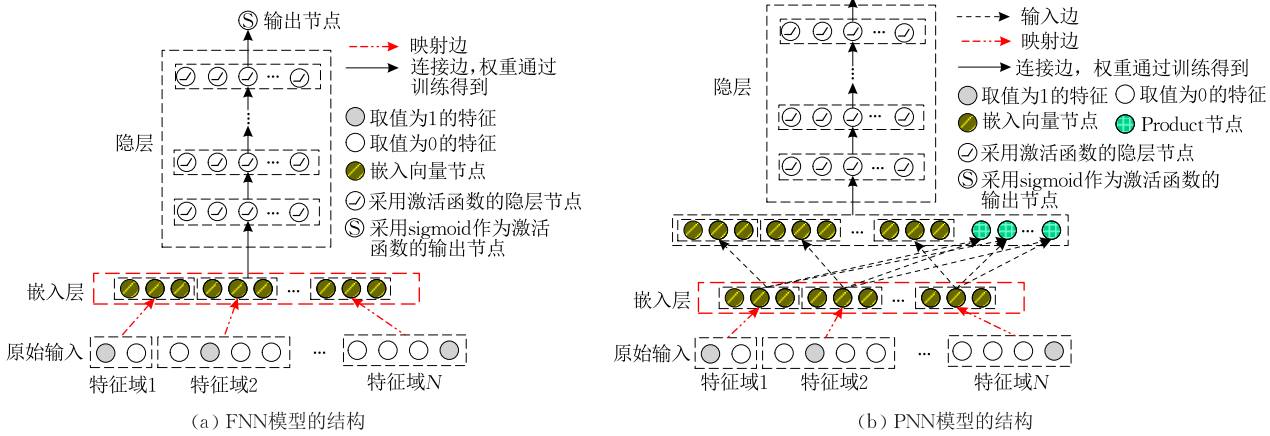


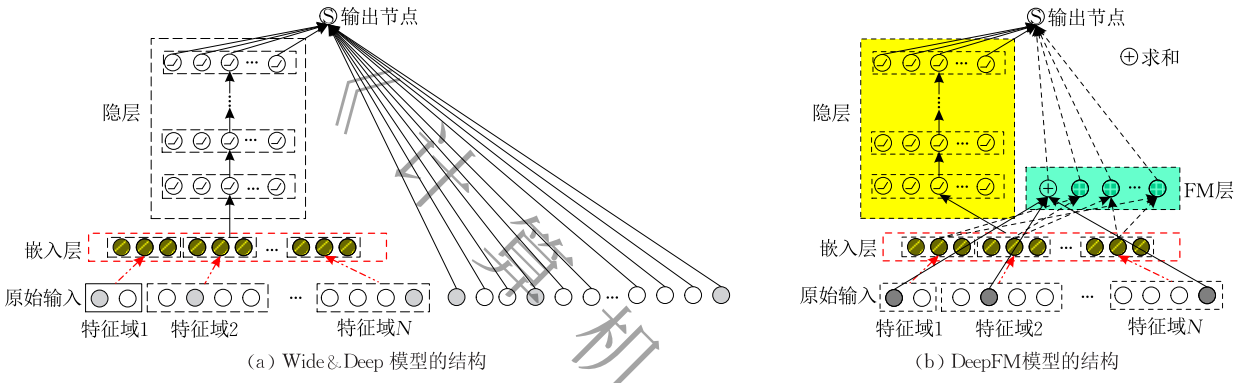
图 2　嵌入向量的示例

本节将详细介绍几种比较典型的基于 DNN 的 CTR 预测模型，各种模型的结构如图 3 和图 4 所示. 图 3(a)是 FNN 模型[5]，它是一个采用因子分解机模型 FM 预先初始化嵌入向量的前馈神经网络，其特点是嵌入的特征域向量是预先训练的，因此可以大幅降低 DNN 参数训练的计算复杂度. 图 3(b)是 PNN 模型[15]，不同于 FNN，它在 Embedding layer 和第一个隐藏层之间增加了一个 Product layer，以捕捉高阶特征之间的相互作用. 根据 product 操作的不同类型，有三种变化：IPNN、OPNN、PNN*，其中 IPNN 表示任意两个特征域的嵌入向量做内积，



(a) FNN模型的结构 (b) PNN模型的结构

图 3 FNN 模型和 PNN 模型的结构



(a) Wide&Deep 模型的结构 (b) DeepFM模型的结构

图 4 两种典型的融合结构

OPNN 表示任意两个特征域的嵌入向量做外积，PNN* 表示将内积和外积的输出结果拼接起来. 不同于 FNN，PNN 在嵌入层的输入中不仅考虑了一阶特征的嵌入向量，还考虑了任意两个特征嵌入向量之间的组合操作.

这里 FNN 模型和 PNN 模型忽略了一阶特征的相互作用. 为此 Google 提出一种能够同时考虑低阶和高阶特征相互作用的融合结构 Wide&Deep[16]，如图 4(a)所示，该结构将线性模型和 DNN 结合起来联合训练，相比于单独的线性模型和深度学习模型，融合结构的预测性能有一定提升，但是其 Wide 部分仍然依赖于特征工程. 文献[17]提出的 DeepFM 模型重新设计了融合结构，如图 4(b)所示，该结构将 FM 模型和 DNN 结合起来联合训练，优点是不需要特征工程支持，同时也可以学习低阶和高阶特征的相互作用.

表 1 展示了几种典型基于 DNN 的 CTR 预测模型与 DPSN 的特点对比. 分析发现几种典型方案均选择了前馈神经网络来学习高阶特征的表示，在网络结构上均在第一个隐藏层(Hidden layer)之前加入了嵌入层，其中 FNN、PNN 以及 Wide&Deep 和 DeepFM 的 Deep 部分的输入只依赖于嵌入向量；而在低阶部分，Wide&Deep 输入的是原始特征和交叉积的变换特征，DeepFM 输入的是原始特征及嵌入向量的内积. 基于上述分析，本文提出的 DPSN 模型采用了与 Wide&Deep 和 DeepFM 类似的融合结构，其特点是利用两个不同的前馈神经网络来分别学习特征之间的高阶表示，通过 Stacking Network 将两个前馈网络输出的特征域高阶表示拼接起来作为一个新的 DNN 输入进行 CTR 预测. 此外，DPSN 不同于其它深度学习模型之处在于嵌入层的设计，除了包括每个特征的嵌入向量，还包括特征的一阶权重，作为对嵌入向量表示的补充.

表 1 基于 DNN 的 CTR 预测模型的特点

	FNN	PNN	DPSN	Wide&Deep	DeepFM
特征工程	×	×	×	√	×
低阶特征	×	×	×	√	√
高阶特征	√	√	√	√	√
嵌入层	√	√	√	√	√
网络结构	DNN	DNN	3个DNN	LR+DNN	FM+DNN
嵌入向量训练	预训练嵌入向量			嵌入向量作为整体训练	

3　基于融合结构的 CTR 预测模型

本节将详细介绍论文提出的 DPSN 模型，如图 1 所示. 首先介绍模型设计的动机；然后按照由底向上的顺序逐一介绍模型的组成，包括：(1) 嵌入层，着重介绍从原始输入特征到嵌入层一阶特征权重单元和嵌入向量单元的映射方法；(2) Deep Network，前馈神经网络，用于学习输入仅为一阶特征相关信息的高阶特征表达；(3) Product Network，前馈神经网络，用于学习输入增加了二阶交叉特征的高阶特征表达；(4) Stacking Network，前馈神经网络，用于拼接前两个 DNN 的输出，并进一步挖掘不同的高阶特征表达之间的交叉信息；在此基础上，介绍模型参数的学习算法，并对模型的参数复杂度给出了简化的理论分析；最后分析和证明了 DPSN 模型的收敛性.

3.1　设计动机

本文在设计 DPSN 模型时，主要受到两个方面的启发. 一是 FNN 和 PNN 采用 FM 模型进行预训练，从而得到每个特征的嵌入向量，这里认为嵌入向量携带了对应特征的重要信息，因此可以在嵌入层采用嵌入向量来替代原始特征. 而 FM 模型中不仅可以学到特征的嵌入向量，还可以学到一阶特征在预测 CTR 时对应的权重，如 3.2 节的式(3)所示，这里一阶特征的权重也携带了反映该特征的重要信息，而在目前已有的基于 DNN 的预测模型中，都只考虑了每个特征对应的嵌入向量，而忽略了一阶特征的权重. 因此，本文尝试将一阶特征的权重也作为信息单元，引入到嵌入层，并在后续实验中验证该设计的有效性，在嵌入层中加入一阶特征权重单元的 FNN 和 PNN 方案都显著优于只考虑嵌入向量单元的方案.

第二个启发是 Wide & Deep 和 DeepFM 的融合结构，这两个方案采用的都是“深度模型＋浅层模型”的结构，将样本原始特征分别输入到两个不同的模型学习不同的特征表达，然后拼接起来联合学习，完成最终的 CTR 预测，其基本思路是在最终预测时比单一深度学习模型增加更多的有效信息. 受这一思想启发，本文设计出一种“深度模型＋深度模型”的结构，将经过预处理后的嵌入层单元分别输入到两个不同结构的 DNN 中，一个 DNN 只考虑一阶特征的信息作为输入进行高阶交叉特征的探索，另一个 DNN 同时考虑一阶和二阶组合特征的信息作为输入，进行高阶交叉特征的探索，从而得到更多不同的高阶特征的组合模式，提供给最终的 CTR 预测. 在后续实验中，论文也通过单一结构和融合结构的对比实验，验证了该设计的有效性. 下面将按照由底向上的顺序介绍本文提出的 DPSN 模型包含的各组成部分.

3.2　嵌入层

如前所述，CTR 预测中使用的原始特征通常是超高维度的稀疏向量，如果将其直接输入到 DNN 中，将会产生非常巨大的计算开销，因此将每个原始特征映射为固定维度的嵌入向量将会大幅降低输入单元的个数. 在 DPSN 中设计了一个不同于 FNN 和 PNN 的嵌入层，其中的单元除了每个特征对应的嵌入向量还包括一阶特征的权重，如图 5 所示.

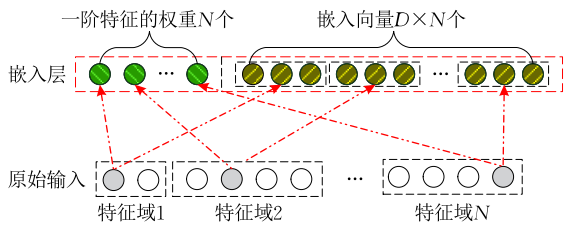


图 5　嵌入层的结构

在 CTR 预测中，可利用的特征主要来自于三个方面：广告特征(如广告创意、广告内容类别等)、上下文特征(如网页内容主题、广告位的位置及尺寸等)、用户特征(如用户类别标签、地理位置、人口统计特征等)[9]. 假设独热编码后的原始特征向量中有 N 个特征域，第 i 个特征域中有 M 个特征，每个特征对应的嵌入向量的维度为 D，则对应的嵌入层中有 N 个嵌入向量，每个嵌入向量中有 D 个单元节点，由于同一个样本中每个特征域只可能出现一个特征，因此嵌入层中的一阶特征的权重单元数为 N，嵌入层中总的单元数为 $(D+1)\times N$. 类似于 FM 模型，在 DPSN 的嵌入层每个特征都有自己的一阶权重，因此特征域 i 的 M 个特征的一阶权重可用向量 $\boldsymbol{we}_i=[we_1^i, we_2^i, \cdots, we_M^i]$ 表示.

每个特征的嵌入向量和一阶权重可以作为模型参数进行端到端的统一训练(例如 Wide & Deep 和 DeepFM)，也可以采用其它 CTR 预测模型进行预先训练. 在本文提出的 DPSN 中采用 FM 模型[10]在训练集上进行预训练，从而得到每个特征的嵌入向量和一阶权重，公式如式(3)所示，这里 $f(\boldsymbol{we}, \boldsymbol{v}, \boldsymbol{x}_i)$ 表示基于样本 i 的点击率预测值，$\boldsymbol{x}_i$ 表示样本 i 特征向量，$\boldsymbol{x}_i=(x_i^1, x_i^2, \cdots x_i^n)$，$n$ 表示独热编码后的特征数，x_i^k 和 x_i^l 分别表示样本 i 的第 k 个特征的值和第 l

个特征的值，$\boldsymbol{v}_k$ 和 $\boldsymbol{v}_l$ 表示第 k 个特征和第 l 个特征的隐含向量. 用训练好的 FM 模型中每个特征的一阶权重初始化嵌入层的一阶权重节点的值，用 FM 模型中每个特征的隐含向量初始化嵌入向量节点的值. 仍然沿用图 2 的例子，假设训练好的 FM 模型中，男性特征的一阶权重为 0.4，北京特征的一阶权重为 0.2，则嵌入层的映射结果如图 6 所示，包括 6 个单元节点.

$$f(\boldsymbol{we},\boldsymbol{v},\boldsymbol{x}_i)=\boldsymbol{we}^{\mathrm{T}}\boldsymbol{x}_i+\sum_{k=1}^{n}\sum_{l=k+1}^{n}(\langle\boldsymbol{v}_k\cdot\boldsymbol{v}_l\rangle\cdot x_i^k\cdot x_i^l) \tag{3}$$

其中，$\langle\boldsymbol{v}_k\cdot\boldsymbol{v}_l\rangle=\sum_{d=1}^{D}v_k^D v_l^D$.

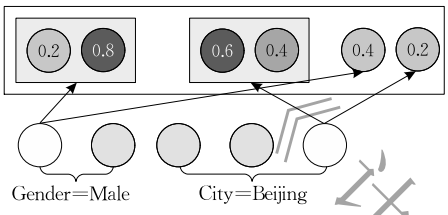


图 6 嵌入层映射的例子

3.3 Deep Network

Deep Network 是一个包含多个隐层的前馈神经网络，用于学习输入仅为一阶特征相关的高阶特征之间的交互信息，如图 1 所示. 在 Deep Network 中，嵌入层的每个节点与第 1 个隐层的每个节点全连接，第 1 个隐层中每个节点的输出值采用式(4)计算，其中 $\boldsymbol{h}_1\in\mathbb{R}^{n_1}$ 是第 1 个隐层节点的输出向量，n_1 是第 1 个隐层的节点数，$\boldsymbol{W}_0$ 表示嵌入层节点到第 1 个隐层节点的连接权重，$\boldsymbol{W}_0\in\mathbb{R}^{n_1\times n_0}$，$n_0$ 是嵌入层的节点数，$\boldsymbol{x}_0\in\mathbb{R}^{n_0}$ 是嵌入层的输出向量，$\boldsymbol{b}_0$ 表示第 1 个隐层的偏置向量，$\boldsymbol{b}_0\in\mathbb{R}^{n_1}$，隐层节点的激活函数 $f(\cdot)$ 采用 ReLU[20].

$$\boldsymbol{h}_1=f(\boldsymbol{W}_0\boldsymbol{x}_0+\boldsymbol{b}_0) \tag{4}$$

Deep Network 是一个前馈深度神经网络，每个隐层的节点数和隐层的层数可调整，隐层之间每个节点均采用全连接，第 $l+1$ 个隐层节点的输出值计算如式(5)所示，$\boldsymbol{W}_l$ 表示第 l 个隐层节点到第 $l+1$ 个隐层节点的连接权重，$\boldsymbol{W}_l\in\mathbb{R}^{n_{l+1}\times n_l}$，$n_l$ 和 n_{l+1} 分别是第 l 个隐层和第 $l+1$ 个隐层的节点数，$\boldsymbol{h}_l\in\mathbb{R}^{n_l}$ 是第 l 个隐层节点的输出值，$\boldsymbol{b}_l$ 表示第 $l+1$ 个隐层的偏置向量，$\boldsymbol{b}_l\in\mathbb{R}^{n_{l+1}}$，隐层中所有节点的激活函数 $f(\cdot)$ 都采用 ReLU；最后 1 个隐层节点的输出值将直接作为输入传输到 Stacking Network 输入层的部分节点；

$$\boldsymbol{h}_{l+1}=f(\boldsymbol{W}_l\boldsymbol{h}_l+\boldsymbol{b}_l) \tag{5}$$

需要说明的是，由于在 DNN 隐层节点的计算中是按“节点”进行交叉组合，因此输入如果只考虑特征的嵌入向量，将会使交叉组合不是按照每个“特征”来完成，各嵌入向量的每个单元节点将作为一个独立的输入值，只有一阶特征对应的权重才反映的是一个完整特征信息，示意如图 7 所示. 因此，直观地认为一阶特征的权重值在隐层节点的交叉组合中将起着更为有意义的作用.

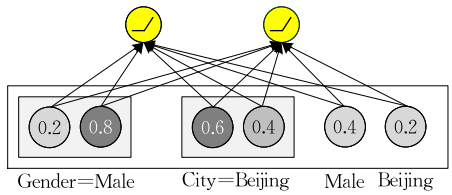


图 7 嵌入层到隐层节点的连接示意

3.4 Product Network

Product Network 也是一个包含多个隐层的前馈神经网络，不同于 Deep Network，它在嵌入层和第 1 个隐层之间增加一个 Product 层，如图 1 所示，该层不仅包含嵌入层的单元节点，还包含任意两个特征的嵌入向量进行两两内积的单元节点，计算如式(6)所示，这里 $p_{i,k}$ 表示嵌入向量 $\boldsymbol{v}_i$ 和 $\boldsymbol{v}_k$ 的内积，因此在 Product 层有 $N\times(N-1)/2$ 个 Product 节点，这里 N 表示嵌入层嵌入向量的个数；Product 层的节点与第 1 个隐层的节点全连接，隐层中每个节点的激活函数 $f(\cdot)$ 都采用 ReLU，因此每个节点输出值的计算方法与 Deep Network 相同；在 Product Network 中最后 1 个隐层节点的输出值将直接作为输入传输到 Stacking Network 输入层的部分节点.

$$p_{i,k}=\langle\boldsymbol{v}_i\cdot\boldsymbol{v}_k\rangle=\begin{bmatrix}v_i^1 & v_i^2 & \cdots & v_i^D\end{bmatrix}\begin{bmatrix}v_k^1\\ v_k^2\\ \vdots\\ v_k^D\end{bmatrix}=\sum_{t=1}^{D}v_i^t v_k^t \tag{6}$$

由于 DNN 中，是按照“节点”进行加权求和来完成特征交叉，没有考虑“叉积”这样的组合方式，因此在 Product Network 中输入信息不仅包含了一阶特征相关的信息，还包含了二阶特征的“叉积”信息，从而使输入 DNN 的信息更丰富，可以得到更多的不同于 Deep Network 的高阶特征组合模式.

3.5 Stacking Network

Stacking Network 主要用于将 Deep Network 和 Product Network 输出的高阶特征表示拼接起来作为一个新的 DNN 的输入，使得整个模型能够联

合起来进行参数训练. 为了进一步挖掘不同高阶特征之间的交叉信息，DPSN 中将拼接后的特征向量输入到一个新的 DNN 中，如图 1 所示. 这里隐层的数量是可调整的，当隐层数为 0 时，Stacking Network 就简化为拼接各高阶特征，然后进行预测输出的功能. 实际上，后续实验结果说明，增加少量隐层确实可以在一定程度上提升预测性能.

假设 Stacking Network 的输入层有 (n_D+n_P) 个节点，这里 n_D 表示 Deep Network 的最后 1 个隐层的节点数，n_P 表示 Product Network 的最后 1 个隐层的节点数，输入层的节点与第 1 个隐层的节点全连接，隐层之间的节点都采用全连接，隐层中每个节点的激活函数 $f(\cdot)$ 都采用 ReLU，因此隐层中每个节点的输出值都采用式(5)计算，最后输出节点用于计算预测点击率，输出节点的激活函数采用 Sigmoid 函数[21]，预测点击率 p 的计算公式如式(7)所示，这里 $\boldsymbol{W}_L^S$ 表示最后 1 个隐层到输出节点的权重向量，$\boldsymbol{W}_L^S\in\mathbb{R}^{n_L}$，$\boldsymbol{h}_L^S$ 表示最后 1 个隐层的输出向量，$\boldsymbol{h}_L^S\in\mathbb{R}^{n_L}$，$b_L^S$ 表示输出节点的偏置.

$$p=\mathrm{Sigmoid}(\boldsymbol{W}_L^S\boldsymbol{h}_L^S+b_L^S) \tag{7}$$

3.6　模型训练

为了对融合结构中的权重和偏置参数进行学习，本论文使用对数损失函数作为目标函数来优化模型参数，如式(8)所示，

$$L(\boldsymbol{\theta})=-\frac{1}{N}\sum_{i=1}^{N}(y_i\log p(\boldsymbol{x}_i,\boldsymbol{\theta})+(1-y_i)\log(1-p(\boldsymbol{x}_i,\boldsymbol{\theta})))+\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 \tag{8}$$

这里 $L(\boldsymbol{\theta})$ 是对数损失函数，$\boldsymbol{\theta}$ 表示融合结构的参数，$p(\boldsymbol{x}_i,\boldsymbol{\theta})$ 表示根据样本 i 的特征向量 $\boldsymbol{x}_i$ 基于融合结构当前参数 $\boldsymbol{\theta}$ 计算得到的预测点击率，y_i 表示样本 i 中关于点击行为的真实标记，有点击行为为 1，无点击行为为 0，N 表示训练数据集中的样本数，$\lambda\|\boldsymbol{w}\|_2^2/2$ 表示 L2 正则化项[22]，用于防止过拟合，λ 是正则化参数，由手动设置，$\boldsymbol{w}$ 是融合结构中所有节点之间边的权重向量；参数学习的目标是求解使对数损失函数最小的融合结构参数. 论文使用随机梯度下降(Stochastic Gradient Descent，SGD)算法[23-24]来求解式(8)中的融合结构参数 $\boldsymbol{\theta}$，包括节点之间边的权重和节点的偏置向量.

这里对梯度下降求解的基本过程及涉及的参数进行简要介绍. SGD 是利用了 Taylor 展开式的一阶近似推导出来的，将损失函数 $L(\boldsymbol{\theta})$ 在初值的邻域内展开，将导数 $\partial L/\partial\boldsymbol{\theta}$ 记为关于 $\boldsymbol{\theta}$ 的梯度向量 $g(\boldsymbol{\theta})$，得到如下表达式：

$$\begin{aligned}&L(\boldsymbol{\theta})\approx L(\boldsymbol{\theta}_0)+g(\boldsymbol{\theta}_0)^{\mathrm{T}}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)\\&\text{s.t. } L(\boldsymbol{\theta})<L(\boldsymbol{\theta}_0)\\&\Rightarrow g(\boldsymbol{\theta}_0)^{\mathrm{T}}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)<0\end{aligned} \tag{9}$$

为了尽快降低损失函数的值，梯度向量 $g(\boldsymbol{\theta}_0)$ 和参数差值向量 $(\boldsymbol{\theta}-\boldsymbol{\theta}_0)$ 的夹角需要达到 180°，即沿负梯度方向下降. 由此可以得到参数更新规则：

$$\boldsymbol{\theta}-\boldsymbol{\theta}_0=-\eta g(\boldsymbol{\theta}_0)\Rightarrow\boldsymbol{\theta}=\boldsymbol{\theta}_0-\eta g(\boldsymbol{\theta}_0) \tag{10}$$

其中，η 是一个比较小的正实数，也称为学习率或步长. 根据式(9)、式(10)得到的参数 $\boldsymbol{\theta}$ 能使得损失函数的值减小，因而更接近最优解. 将式(10)的结果记为 $\boldsymbol{\theta}_1$，按照相同的方式迭代地更新参数，参数最终会达到损失函数的极小值点，因此参数更新规则如式(11)所示，$t=0,1,2,\cdots$ 代表迭代轮次.

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\Delta\boldsymbol{\theta}_t=\boldsymbol{\theta}_t-\eta g(\boldsymbol{\theta}_t) \tag{11}$$

本节的最后对融合结构的参数复杂度进行一个简化的理论分析. 假设 N 为特征域个数，D 为嵌入向量的维度，所有 DNN 网络都有相同的隐层数 L，且每个隐层的节点数相同，记为 m，则：

(1) Deep Network 的参数个数为

$$(N\times D+N)\times m+m+(m^2+m)\times(L-1).$$

(2) Product Network 的参数个数为

$$\left(N\times D+N+\frac{N\times(N-1)}{2}\right)\times m+m+(m^2+m)\times(L-1).$$

(3) Stacking Network 的参数个数为

$$2\times m^2+2\times m+(m^2+m)\times(L-1)+1.$$

(4) 融合结构的参数复杂度为

$$(3L-1)m^2+(3L+1)m+(2D+1.5+0.5N)Nm+1.$$

因此，在 DPSN 模型中，对模型参数影响最大的是每个隐层的神经元节点数，按照 m 的 2 次方倍增长. 此外，每个 DNN 的隐层数也会影响模型参数，而嵌入向量维度和特征域数目通常只会影响从嵌入层到第一个隐层的模型参数. 后续实验将会逐一对各个超参数对模型预测性能及参数复杂度的影响进行分析.

3.7　模型的收敛性分析

如 3.6 节所述，本文使用对数损失函数来最小化模型误差，因此模型的目标函数如式(8)所示：

$$L(\boldsymbol{\theta})=-\frac{1}{N}\sum_{i=1}^{N}(y_i\log p(\boldsymbol{x}_i,\boldsymbol{\theta})+(1-y_i)\log(1-p(\boldsymbol{x}_i,\boldsymbol{\theta})))+\frac{\lambda}{2}\|w\|_2^2.$$

目前已有大量的研究[25-26]证明，只要目标函数是凸函数，在数据线性不可分或有正则项时，使用梯度下降法求解模型参数一定能够收敛，即在梯度下降法

中每一轮训练都是沿着梯度下降的方向更新参数，逐步逼近目标函数的极小值.因此，只需证明式(8)是一个凸函数，即可证明模型收敛.在证明之前首先介绍两个定理.

定理 1. 凸函数的复合还是凸函数.

定理 2. 凸函数的正线性组合还是凸函数.

证明.

首先对于式(8)中的 L2 正则项

$$\|\boldsymbol{w}\|_2^2=\sum_{i=1}^{n}\omega_i^2,$$

显然，该项是处处可微分的凸函数，其中 $\boldsymbol{w}$ 是模型的权重向量.因此只需证明式(8)中的对数损失函数是凸函数，记为

$$\boldsymbol{J}(\boldsymbol{\theta})=-\frac{1}{N}\sum_{i=1}^{N}(y_i\log p(\boldsymbol{x}_i,\boldsymbol{\theta})+(1-y_i)\log(1-p(\boldsymbol{x}_i,\boldsymbol{\theta}))) \tag{12}$$

由于 DPSN 模型使用 Sigmoid 函数作为最终输出单元的激活函数，因此式(12)中样本 i 的输出预测值为

$$p(\boldsymbol{x}_i,\boldsymbol{\theta})=\frac{1}{1+e^{-v_i}} \tag{13}$$

其中 v_i 是样本 i 最终输出节点的值.根据定理 1 与定理 2，证明 $\boldsymbol{J}(\boldsymbol{\theta})$ 是凸函数，只需证明式(14)是凸函数，

$$F(v)=-\log\left(\frac{1}{1+e^{-v}}\right)=\log(1+e^{-v}) \tag{14}$$

对式(14)求二阶导数，得：

$$F'(v)=\frac{-e^{-v}}{1+e^{-v}},\quad F''(v)=\frac{e^{v}}{(1+e^{v})^2}\geqslant 0.$$

由于二阶导数非负，$F(v)$ 是关于 v 的凸函数，从而证明式(12)为凸函数.因此得证式(8)为凸函数.

4 实验及性能评价

为了对提出的 DPSN 模型的预测性能进行评价，本文基于 iPinYou[①] 和 Criteo[②] 两个公开数据集完成了大量的实验.本节首先对数据集及评价指标进行介绍，然后描述了 8 种作为对比的典型 CTR 预测模型，最后通过 7 组实验展示 DPSN 模型的预测性能，并对实验结果展开讨论.

4.1 iPinYou 和 Criteo 数据集及评价指标

本文实验采用的 iPinYou 数据集是 iPinYou 公司在 2013 年发布的一个真实广告投放的数据集，包括曝光机会、竞价、点击、转化四类日志，其中曝光机会和点击日志可用于点击率预测.具体来说，在本文实验中，每个样本对应了一次广告曝光，特征信息包括用户的相关信息(例如用户类别标签、使用的浏览器、IP 地址、所在区域、城市等)、广告位的相关信息(例如广告位的宽度、高度、可见性、所在网站的域名以及 URL 等)，投放的广告 ID，以及最终的点击情况(用户点击为 1，无点击为 0).考虑到 CTR 预测模型是针对每个广告商的，因此本论文采用了其中 Advertiser ID 分别为 1458、3386、3358、3427 的四个广告商的投放和点击日志分别建立了四个数据集.所有实验均采用前 7 天的样本作为训练集，采用后 3 天的样本作为测试集，表 2 展示了四个数据集中样本的统计情况.

表 2 iPinYou 四个数据集的统计情况

		样本数	点击数	实际点击率(10^{-3})	特征域数	特征数	嵌入层单元数
Advertiser ID=1458	训练集	3083056	2454	0.79596	16	560802	176
	测试集	614638	515	0.83789	16	560802	176
Advertiser ID=3386	训练集	2847802	2076	0.72898	16	556884	176
	测试集	545421	445	0.81588	16	556884	176
Advertiser ID=3358	训练集	1742104	1358	0.77952	16	491700	176
	测试集	300928	260	0.86399	16	491700	176
Advertiser ID=3427	训练集	2593765	1926	0.74255	16	551158	176
	测试集	536795	366	0.68182	16	551158	176

从表 2 观察发现真实互联网上广告投放的点击率是非常低的，即数据集中正负样本的比例严重不平衡，会使得模型对正样本的学习不充分，从而降低 CTR 预测模型的精度[27].此外，可以发现独热编码后原始特征的数目是非常巨大的，达到 50 万量级，将其直接输入 DNN 必然导致巨大的计算开销，经过本文提出的嵌入向量映射后，使输入 DNN 的单元数目大幅下降为 176，这里假设每个嵌入向量的维度为 $D=10$.

Criteo 数据集是 Criteo 公司在 2014 年 Kaggle 平台上发起的展示广告点击率预测大赛的数据集.

① iPinYou：http://data.computational-advertising.org/

② Criteo http://labs.criteo.com/2014/02/download-kaggle-display-advertising-challenge-dataset/

该数据集包含 45 840 616 条真实展示广告的特征值与点击反馈，来自于 Criteo 公司连续 7 天的交易流量. 为了减小数据规模及保持正负样本的比例平衡，该数据集已经对样本进行了负采样，使得点击率提升至约 25.6%. 数据集包含 13 个数值特征以及 26 个分类特征，并且为了保护隐私，所有分类特征的值均被 Hash 到 32 位的字符串上，因此这些特征现实含义是无法知晓的，同时有些特征存在缺失值. 在本文的实验中，首先对缺失值进行了填补，然后按照 5∶1 的比例将样本按顺序划分为训练集和测试集. 此外，为了降低实验的时间开销与空间开销，在不显著降低模型性能的前提下，本文弃用了 5 个特征取值过多的分类特征，即只使用了 13 个数值特征和 21 个分类特征，并且将数值特征转化为分类特征进行预处理. 表 3 展示了本文实验采用的 Criteo 数据集的样本统计情况. 目前，iPinYou 和 Criteo 数据集已逐步成为学术界衡量 CTR 预测模型性能的基准数据集[5,15,17].

表 3　Criteo 数据集的统计情况

	样本数	点击数	实际点击率	特征域数	特征数	嵌入层单元数
训练集	38 202 927	9 789 350	0.25 624	34	569 354	374
测试集	7 637 689	1 956 088	0.25 611	34	569 354	374

实验使用 AUC[28] 和 LogLoss[29] 作为主要的评价指标，其中 AUC 是 ROC 曲线下的面积，AUC 值越大，说明 CTR 预测模型的性能越好；LogLoss 是交叉熵损失，LogLoss 越小，说明预测模型的性能越好.

4.2　对比模型

为了进行对比，本文选择了 7 种具有代表性的 CTR 预测模型，包括 LR[2]、FM[10]、GBDT＋LR[11] 以及最新的 FNN[5]、PNN[15]、Wide & Deep[16]、DeepFM[17] 等深度学习模型. 所有对比模型都是基于 TensorFlow① 实现的，对于包含深度神经网络的方案，为了简化，设置每个隐层节点的激活函数为 ReLU，输出节点的激活函数为 Sigmoid，最优化参数求解均采用随机梯度下降法. 此外，论文还对比了一种用投票机制做模型融合的方案，即分别用 FNN 和 PNN 学习模型得到 CTR 预测值，再将两个 CTR 预测值按各自 0.5 的权重求和作为融合模型的 CTR 预测值，这种融合方案记为 FNN＋PNN.

4.3　性能评价及分析

本节将通过 7 组实验对 DPSN 模型的性能进行评价，并对实验结果展开讨论. 其中第 1 组实验用于验证 DPSN 模型的收敛性；第 2 组实验用于说明增加一阶特征权重信息对 FNN 和 PNN 模型预测性能的提升；第 3 组实验是分析不同的模型超参对 DPSN 模型预测性能的影响，包括：隐层数、隐层节点数、激活函数、嵌入向量维度；第 4 组实验用于评价本文提出的新的嵌入层结构对 DPSN 模型预测性能的影响；第 5 组实验是对比单一结构和融合结构对模型预测性能的影响；第 6 组实验用于分析负采样对预测性能的影响；第 7 组实验是将 DPSN 模型与其它典型 CTR 预测模型的性能进行对比. 论文在 GitHub 上分享了本文的所有实验代码和实验结果②.

4.3.1　DPSN 模型的收敛性验证

本节将通过实验验证 DPSN 融合模型的收敛性. 图 8 展示了 DPSN 融合结构在 iPinYou 的 1458、3386 训练集上 LogLoss 随着训练轮次逐渐降低直至收敛的过程，学习率均设为 0.008. 观察发现，1458 在 232 轮次收敛，3386 在 175 轮次收敛，3386 的收敛速度快于 1458. 由此可证明 DPSN 融合结构由于采用了交叉熵损失作为损失函数，因此能够保证模型的收敛性. 为了获得更快的收敛速度，在后续实验中将学习率设置为 0.1.

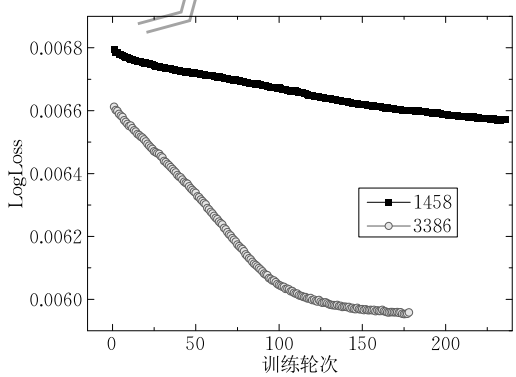


图 8　DPSN 在 iPinYou 1458 和 3386 上的训练过程

4.3.2　一阶特征权重节点对预测性能的影响

DPSN 的一个重要改进是将 FM 预训练的一阶特征的权重作为特征的信息加入到嵌入层，实验 2

① TensorFlow. https://www.tensorflow.org/
② 本文实验代码和实验结果. https://github.com/mjliu-advertising

对比分析了原始的 FNN 和 PNN 模型在加入一阶特征权重后预测性能的变化. 实验基于 iPinYou 的四个数据集完成,结果如表 4 所示. 观察发现,无论是 FNN 还是 PNN,加入一阶特征权重节点到嵌入层后,预测性能均是提升的,例如 1458 数据集,FNN 在 AUC 指标上提升了 0.369%,LogLoss 指标上提升了 0.320%,PNN 在 AUC 指标上提升了 0.526%,LogLoss 指标上提升了 0.122%.

表 4 一阶特征权重对 FNN 和 PNN 模型性能的影响

	原 FNN		新 FNN		原 PNN		新 PNN	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
1458	0.7061	6.561	0.7088	6.540	0.7062	6.547	0.7099	6.539
3386	0.7986	5.920	0.8003	5.902	0.8034	5.916	0.8045	5.909
3358	0.8103	6.067	0.8116	6.056	0.8091	6.058	0.8120	6.050
3427	0.7587	5.262	0.7591	5.257	0.7588	5.261	0.7608	5.263

分析提升的原因,主要是在深度神经网络中,隐层节点对于特征的组合是按输入的每个“节点”进行的,如果只考虑将嵌入向量作为输入,在进行组合时将会把一个嵌入向量的每个单元节点作为一个独立的输入,而不是按照“特征”为单位进行组合,而预训练的一阶特征的权重节点,包含了“特征”作为独立输入的重要信息. 因此在融合模型中,将预训练的一阶特征的权重作为节点加入到嵌入层中是一个有效的设计,进一步验证参见实验 4.

4.3.3 不同超参数对 DPSN 模型的性能影响

(1) 隐层数对预测性能的影响

实验 3-1 用于分析不同隐层数对 DPSN 模型预测性能的影响. 由于 DPSN 包括 Deep Network、Product Network、Stacking Network 三个前馈神经网络,因此需要设置三个神经网络中的隐层数目,在本实验中采用了一个简化的参数设置,首先假设三个神经网络中的隐层结构相同,即相同的层数和每层单元数. 为此,在实验 3-1 中,设置每个 DNN 中每个隐层的单元数分别为 100 和 500,隐层数从 1 层逐层增加到 5 层,实验结果如表 5 所示. 观察发现,并非隐层数越多,预测性能越好,无论是 AUC 指标还是 LogLoss 指标,当隐层数增加到 5 时,1458 和 3386 数据集的预测性能都不太理想,这是因为隐层层数越多,模型越复杂,出现过拟合[30]的可能性越大. 具体地,对于 1458 数据集,从 AUC 指标来说,预测性能最好的结构是每个 DNN 有 3 个隐层,每个隐层的节点数为 500,对于 3386 数据集,预测性能最好的是每个 DNN 有 2 个隐层,每个隐层的节点数为 500. 从 LogLoss 指标来看,这两个结构的隐层设置也能取得较好的效果.

表 5 实验 3-1 不同隐层数对性能的影响

评价指标	1458		3386	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
{100}	0.7093	6.5452	0.8029	5.9066
{100-100}	0.7057	6.5517	0.8030	5.9210
{100-100-100}	0.7106	6.5366	0.8009	5.9129
{100-100-100-100}	0.7107	6.5386	0.8007	5.9116
{100-100-100-100-100}	0.7069	6.5440	0.7981	5.9139
{500}	0.7119	6.5377	0.8016	5.8977
{500-500}	0.7118	6.5364	**0.8046**	**5.9021**
{500-500-500}	**0.7126**	**6.5361**	0.7993	5.9084
{500-500-500-500}	0.7103	6.5475	0.8002	5.9091
{500-500-500-500-500}	0.7086	6.5467	0.7889	5.9448

表 4 展示了不同隐层数和每个隐层不同神经元数对需要学习的模型参数的影响. 可以看到影响最大的确实是每个隐层的神经元数目,当每层 500 个节点时,即或每个 DNN 只有 1 个隐层,需要学习的模型参数也达到了 70 万量级,模型参数随着隐层数增加而增加,当增加到 5 层时,每个隐层 500 个节点的模型参数达到了 370 万量级. 本实验中选择的最优隐层数设置,也牺牲了一定的参数复杂度为代价,不过随着硬件性能的提升,这个代价是可以接受的.

表 6 不同隐层数下 DPSN 需要学习的模型参数

隐层层数	每层 100 个节点	每层 500 个节点
1	67601	738001
2	97901	1489501
3	128201	2241001
4	158501	2992501
5	188801	3744001

(2) Stacking Network 的隐层数对性能的影响

实验 3-2 是在实验 3-1 的基础上,进一步探索 Stacking Network 隐层数对模型预测性能的影响,用于验证 Stacking Network 中引入 DNN 的影响. 在本实验中,将 Deep Network 和 Product Network 的隐层数设为一致,改变 Stacking Network 中隐层数从 0 层逐层增加到 5 层,其中对于 1458 数据集,设置 Deep Network 和 Product Network 的隐层数为 3 层,对于 3386 数据集,设置隐层数为 2 层,所有隐层的节点数均为 500,实验结果如表 7 和图 9 所示. 观察发现,直接拼接 Deep 和 Product 网络的输

出，不进行进一步高阶特征交叉信息挖掘得到的预测性能(Stacking Network 的隐层数为 0)略微低于在 Stacking Network 中引入隐层进行高阶特征交叉信息挖掘得到的预测性能，因此在 Stacking Network 中引入 DNN 进行高阶特征交叉信息的挖掘对于提升模型的预测性能有一定帮助. 此外，不同的 Stacking Network 隐层数目对于模型的预测性能影响较小，其中对于 AUC 指标，1458 数据集在 3 个隐层的结构下，能够取得最好的预测性能，比 0 个隐层的最差性能提升了 0.3115%，3386 数据集在 2 个隐层的结构下，能够取得最优的预测性能，比 0 个隐层的最差性能提升了 0.6925%；对于 LogLoss 指标，1458 和 3386 数据集在 3 个和 2 个隐层的结构下也能取得较好的性能提升.

表 7　改变 Stacking Network 隐层数对性能的影响

Stacking Network 隐层数	1458		3386	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
0(每层 500 个节点)	**0.7104**	**6.5509**	0.7992	5.9162
1(每层 500 个节点)	0.7123	6.5427	0.8026	5.9092
2(每层 500 个节点)	0.7123	6.5365	**0.8046**	**5.9021**
3(每层 500 个节点)	**0.7126**	**6.5361**	**0.8023**	**5.9078**
4(每层 500 个节点)	0.7123	6.5389	0.8007	5.9136
5(每层 500 个节点)	0.7116	6.5354	0.7995	5.9089

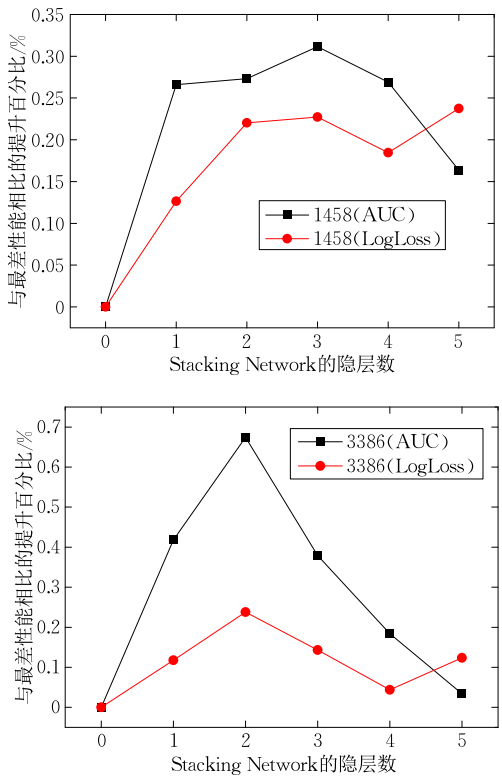


图 9　不同的 Stacking Network 隐层数对预测性能的提升

基于上述结果，将后续实验 DPSN 模型的隐层数进行如下设置：对于 1458 数据集，三个 DNN 的隐层数均为 3 层；对于 3386 数据集，三个 DNN 的隐层数为 2 层. 这是一个选择了性能最优而牺牲了模型参数复杂度的折中设置.

(3)每层神经元数目对预测性能的影响

实验 3-3 用于分析不同隐层的神经单元数目对模型预测性能的影响. 考虑每个隐层设置不同的神经元数，将使得实验非常复杂，为此，简化为所有隐层的神经元数相同，依次改变每个隐层的神经元数分别为{100,200,300,400,500,600,700}，实验结果如表 8 和图 10 所示. 观察发现，每个隐层的神经元数目增加并不能一直提升预测性能，但是会大幅增加学习参数的复杂度，如实验 3-1 表 6 最后分析所示，模型需要学习的参数按每层神经元数目的 2 次方的关系增长. 观察发现，当每层节点数从 100 增加到 500 时，AUC 指标逐渐增加，每层为 500 个节点时 AUC 指标达到最优，再依次增加每层节点数，

表 8　改变隐层神经元数目对预测性能的影响

隐层神经元数目	1458		3386	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
每层 100 个节点	0.7074	6.5444	0.8000	5.9081
每层 200 个节点	0.7080	6.5433	0.8031	5.8973
每层 300 个节点	0.7114	6.5367	0.8033	5.8960
每层 400 个节点	0.7121	6.5373	0.8033	5.9090
每层 500 个节点	**0.7126**	**6.5361**	**0.8046**	**5.9021**
每层 600 个节点	0.7082	6.5400	0.7971	5.9173
每层 700 个节点	0.7060	6.5540	0.7940	5.9277

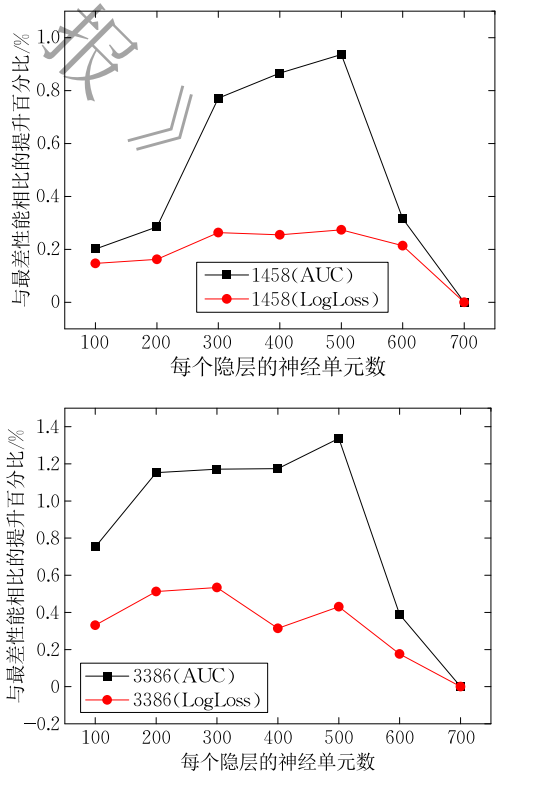


图 10　不同的隐层神经元数目对预测性能的提升

AUC 指标反而下降. 分析原因如下，随着每个隐层的节点数增加，模型可以学到更多有效的高阶特征组合，因此预测性能提升，然而当 500 个节点时，模型的学习能力已经接近极限，再增加隐层节点，这些新增节点学习到的高阶特征组合可能无效，甚至引入更多噪声，从而使模型最终的预测性能下降. 因此实验时选择适中的神经元数目即可，在本文的后续实验中仍然设置每个隐层的神经元数目为 500.

(4) 不同激活函数对预测性能的影响

实验 3-4 用于分析不同隐层神经元的激活函数对模型预测性能的影响. 根据文献[17]的分析，ReLU 和 Tanh 函数比 Sigmoid 函数更适合于深度模型，因此本文将比较 ReLU 和 Tanh 函数对模型预测性能的影响. 为了简化，融合结构中除最后的 CTR 预测输出单元之外的所有神经元的激活函数都设置为相同的激活函数，这里 CTR 预测输出单元仍然使用的是 Sigmoid 函数，实验结果如表 9 所示. 结果显示在 AUC 和 LogLoss 指标上，ReLU 函数比 Tanh 函数的预测性能略好.

表 9 不同激活函数对预测性能的影响

激活函数	1458		3386	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
ReLU	**0.7126**	**6.5361**	**0.8046**	**5.9021**
Tanh	0.7116	6.5425	0.8005	5.9065

(5) 嵌入向量维度对预测性能的影响

如前分析，嵌入向量的维度 D 与模型的学习复杂度直接相关，决定了每个 DNN 的输入单元数，实验 3-5 的目的是分析嵌入向量维度对模型预测性能的影响，希望使用小的维度，获得较好的预测性能，为此将嵌入向量维度分别设置为{5,10,20}，实验结果如表 10 所示. 结果显示嵌入向量维度从 5 到 20，预测性能没有明显提升，且会导致模型需要学习的参数数量有小幅增加，例如 1458 数据集中需要学习的参数个数从 2912501 增加到 2992501，再增加到 3152501. 分析参数量小幅增加的原因是，嵌入向量的维度 D 只对嵌入层到第一个隐层的权重参数有影响.

表 10 嵌入向量维度对预测性能的影响

嵌入向量维度	1458		3386	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
维度 D=5	0.7017	6.5723	0.8015	5.9372
维度 D=10	**0.7126**	**6.5361**	**0.8046**	**5.9021**
维度 D=20	0.7069	6.5450	0.8002	5.9246

4.3.4 一阶特征权重单元对预测性能的影响

在 DPSN 中，一个重要的改进是在嵌入层增加了预训练的一阶特征权重单元. 实验 4 用于对比在嵌入层增加一阶特征权重单元对 DPSN 模型预测性能的影响，实验结果如表 11 所示. 结果显示增加一阶特征权重单元后的预测性能比不加的性能在 AUC 指标上分别提升了 1.3614%(1458)和 0.7147%(3386)，在 LogLoss 指标上提升了 1.5950%(1458)和提升了 0.7308%(3386). 该结果说明在预测模型的输入中增加一阶特征的权重确实能有效提升模型的预测性能.

表 11 增加一阶特征权重单元对预测性能的影响

一阶特征权重单元	1458		3386	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
考虑	**0.7126**	**6.5361**	**0.8046**	**5.9021**
不考虑	0.7030	6.5831	0.7920	5.9456

4.3.5 模型结构对性能的影响

在 DPSN 中，采用了 2 个 DNN 模型来分别学习高阶特征的不同表示，并使用一个 Stacking Network 来将 2 个 DNN 输出的高阶表示进行拼接，从而得到一个整体的预测模型. 为了与仅使用单个 DNN 学习高阶特征的模型对比，实验 5 考察了分别只采用 Deep Network 和 Product Network 来学习高阶特征的 CTR 预测模型的性能，实验结果如表 12 和图 11 所示. 结果显示，预测性能最差的是仅采用 Deep Network 进行 CTR 预测的模型，这是因为它只考虑了一阶特征相关的信息进行高阶特征组合；仅基于 Product Network 进行 CTR 预测的模型性能优于 Deep Network，因为 Product Network 的模型不仅将一阶特征相关的信息作为输入，而且将二

表 12 不同模型结构对预测性能的影响

模型结构	1458		3386	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
Deep Network	0.7088	6.5405	0.7989	5.9163
Product Network	0.7099	6.5392	0.8045	5.9086
DPSN	0.7126	6.5361	0.8046	5.9021

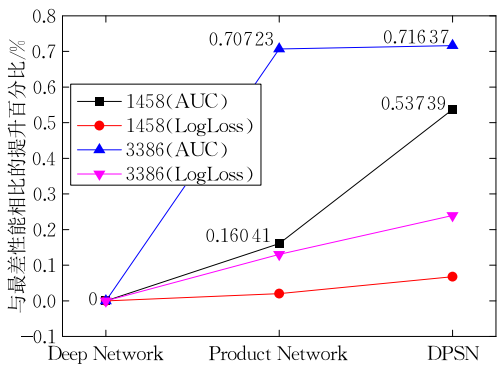


图 11 模型结构对预测性能的提升

阶特征(“叉积”)相关的信息也作为输入，因此能够获得比 Deep Network 更多的信息，由于两个模型的隐层节点数都是相同的，因此可以认为经过 Product Network 挖掘的高阶特征组合模式对于 CTR 预测更为有效.

本文提出的考虑两个模型融合的结构 DPSN 的性能优于单一结构，在 AUC 指标上，比 Deep Network 模型的性能提升了 0.5374%(1458)和 0.7164%(3386)，在 LogLoss 指标上分别提升了 0.0677%(1458)和 0.2397%(3386). 这是因为 DPSN 同时考虑了两个 DNN 模型挖掘得到的高阶特征组合，且进一步挖掘了这两个模型输出的高阶特征组合之间的交叉信息，从而能够在最终预测时采用更为有效的高阶特征组合进行预测. 综上分析，可以说明本论文提出的 DPSN 融合结构确实能够提升 CTR 预测性能.

4.3.6　负采样比率对模型预测性能的影响

如 4.1 节所述 iPinYou 数据集有较大的正负样本不平衡问题，解决这一问题的简单方法是进行负采样. 实验 6 用于验证不同的负采样比率对模型预测性能的影响. 图 12 分别列出了按照正负样本比率{1∶50,1∶100,1∶200,1∶500,1∶1000}对 1458 数据集进行负采样后的预测性能. 观察发现，当进行适当的负采样后(例如 1∶1000)，DPSN 模型的 AUC 指标确实有一定的提升，但是随着负采样比率提高到 1∶50，DPSN 模型的 LogLoss 指标的值将会大幅增加，从原始的 0.006 增加到 0.0144. 这是由于 1458 中的正样本的数量非常少(2454)，负采样后将使得训练集的总样本数减少为 125154，样本数太少会导致模型学习欠拟合，从而导致预测误差增加. 因此，对于不同的数据集，应该根据数据集的实际情况，合理设计负采样的比例. 另一方面，随机负采样会使训练集中样本的特征分布与测试集的样本特征分布出现差异，从而使得有些预测模型欠拟合，特别是对于 iPinYou 这样样本数比较少的数据集. 因此，在本文其它节给出的实验结果均是没有进行负采样的结果.

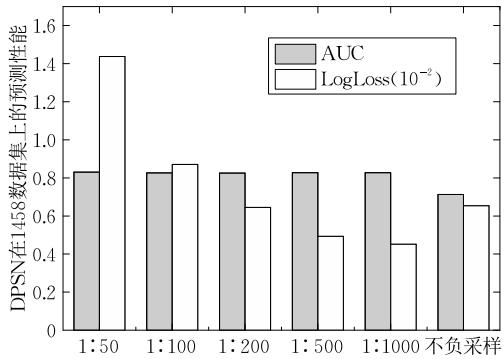


图 12　不同负采样比率下 DPSN 模型的性能

4.3.7　与典型 CTR 预测模型的性能对比

实验 7 用于对比 DPSN 与几种代表性的 CTR 预测模型的性能. 在需要进行特征到嵌入向量映射的模型(FM、FNN、IPNN、Wide & Deep、DeepFM、DPSN)中，设置嵌入向量的固定维度为 $D=10$；在包含 DNN 的模型中，隐层节点的激活函数采用 ReLU，输出节点的激活函数采用 Sigmoid. 此外，如 4.2 节所述，还增加了一种基于投票机制融合方案 FNN+PNN. 实验分别基于 iPinYou 和 Criteo 数据集完成，实验结果如表 13 和表 14 所示.

表 13　典型 CTR 预测模型在 iPinYou 上的性能对比

模型	1458		3386		3358		3427	
	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})	AUC	LogLoss(10^{-3})
LR	0.7017	6.5576	0.7902	5.9493	0.8102	6.0836	0.7515	5.2970
FM	0.7038	6.5651	0.7922	5.9606	0.8091	6.0682	0.7555	5.2875
GBDT+LR	0.6914	6.5853	0.7749	5.9782	0.8030	6.1168	0.7203	5.3394
FNN	0.7062	6.5613	0.7986	5.9136	0.8005	6.1267	0.7587	5.2624
PNN	0.7062	6.5466	0.8034	5.9159	0.8091	6.0577	0.7588	5.2614
Wide & Deep	0.7020	6.5569	0.7958	5.9148	0.8089	6.0627	0.7504	5.2864
DeepFM	0.7006	6.5615	0.7914	5.9480	0.8074	6.0757	0.7492	5.3078
FNN+PNN	0.7080	6.5388	0.8022	5.9074	0.8091	6.0766	0.7587	5.2545
DPSN	0.7126	6.5361	0.8046	5.9021	0.8100	6.0574	0.7593	5.2616

表 14　典型 CTR 预测模型在 Criteo 上的性能对比

	AUC	提升/%	LogLoss(10^{-3})	提升/%
LR	0.7656	0	4.787	0
FM	0.7871	2.808	4.633	3.217
FNN	0.8004	4.545	4.506	5.870
PNN	0.8005	4.559	4.504	5.912
Wide & Deep	0.8009	4.611	4.499	6.016
DeepFM	0.8002	4.519	4.503	5.933
FNN+PNN	0.8006	4.572	4.499	6.016
DPSN	0.8012	4.650	4.494	6.121

图 13 首先比较了三种浅层结构的 CTR 预测模型在 iPinYou 数据集上的性能. 可以发现 GBDT+LR 的预测性能是最差的，该模型采用 GBDT 来进行自动的有效特征组合筛选，然后将筛选出的组合特征向量拼接后输入到 LR 中，因此模型的预测性能依赖于 GBDT 对组合特征的筛选能力，在本实验中由于内存限制的原因，没有更细粒度的对 GBDT 进

行调优(例如决策子树的数量),因此性能在 iPinYou 四个数据集上都略低于 LR. FM 在 AUC 指标上比 LR 分别提升了 0.299%(1458)、0.262%(3386)、0.542%(3427),这是因为 LR 只考虑了一阶特征的信息作为输入,而 FM 不仅考虑了一阶特征的权重信息,而且考虑二阶特征组合的权重信息,因此能获得较好的结果.

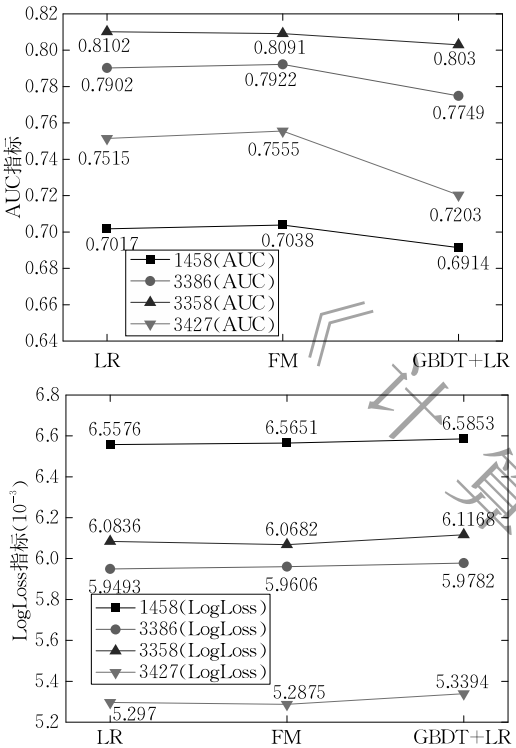


图 13 三种浅层结构的预测模型的性能对比

图 14 展示了 iPinYou 数据集上 5 种基于深度学习的预测模型与 DPSN 模型的性能对比. 首先可以观察到,单一结构的深度学习模型 FNN 和 PNN 的性能都优于浅层结构模型的预测性能,例如在 1458 和 3386 上,FNN 的 AUC 比 FM 提升了 0.341% 和 0.810%,PNN 的 AUC 比 FM 提升了 0.341% 和 1.412%. 这说明深度学习模型确实在特征的高阶组合学习方面具有显著的优势. 其次,PNN 的性能略优于 FNN,这是因为 PNN 的输入信息不仅考虑了一阶特征的相关信息,也考虑了二阶特征组合的权重信息,因此能获得比 FNN 更优的性能. 遗憾的是,在 iPinYou 数据集上,基于联合学习的 Wide & Deep 和 DeepFM 的预测性能都略低于单一结构的 FNN 和 PNN,分析原因可能是 Wide & Deep 中增加了手动特征工程部分,而本文实验中由于无法获知手动特征组合的信息,因此直接将线性部分的组合特征省略掉;DeepFM 则有可能是优化算法的选择与原论文不同,导致预测性能略低于单一结构. 另一方面,直接采用两种训练好的模型通过投票的方式进行融合的 FNN+PNN 获得了非常好的预测性能,例如对于 1458 数据集在 AUC 上比 FNN 提升了 0.269%,在 LogLoss 指标上提升了 0.139%. 最后,本文提出的 DPSN 模型在 iPinYou 的四个数据集上获得了最优的性能提升,说明了一阶特征权重信息和融合结构的设计确实能够挖掘出更多对 CTR 预测有效的高阶特征组合.

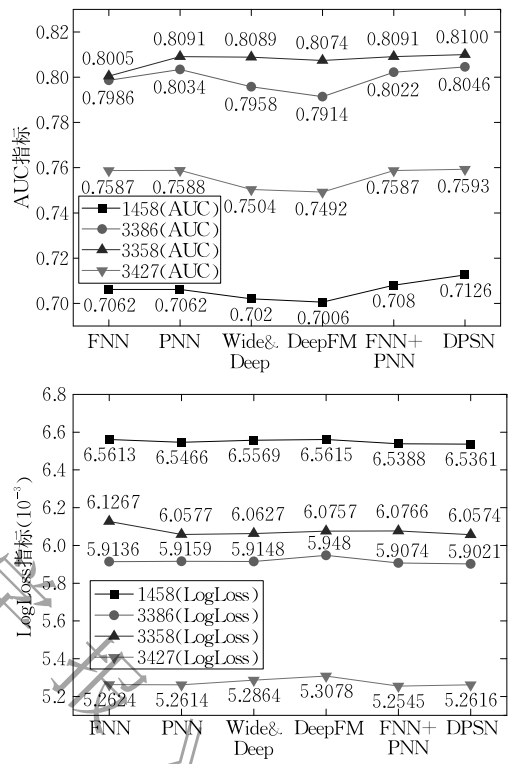


图 14 基于深度学习的预测模型的性能对比

表 15 展示了几种典型 CTR 预测模型在 iPinYou 1458 和 3386 数据集上需要学习的模型参数的量. 可以发现,由于基于独热编码后的原始特征向量,线性 LR 的模型参数与特征数一致;FM 模型虽然仍然基于原始特征,参数比 LR 模型有所增加,但是由于引入向量内积作为特征两两组合的权重,因此有效避免了对二阶特征组合的权重参数的学习;FNN 和 PNN 都是深度神经网络,因此模型参数只依赖于特征域数目、嵌入向量维度、模型结构,这里对于 1458 和 3386 数据集,这些基本参数都相同,因此需要学习的参数量也是相同的,而 PNN 由于引入了一个 Product 层,因此需要学习的参数量比 FNN 有所增加;DeepFM 是融合结构,需要学习两个模型的参数,且每个嵌入向量的值也是作为模型参数学习的,因此需要学习的参数量较大;DPSN 学习的模型

参数量略低于 DeepFM，但是它的嵌入向量和一阶特征权重需要基于 FM 预训练，如果将预训练参数也作为模型参数，则 DPSN 需要学习的参数量是最大的.

表 15　不同预测模型的模型参数复杂度

	LR	FM	FNN	PNN	DeepFM	DPSN
1458	560803	**6168823**	582001	642001	6750824	**2992501**
3386	556884	**6125725**	582001	642001	6707726	**1489501**

为了进一步验证 DPSN 模型的性能，实验 7 在一个更大的数据集 Criteo 上进行了典型模型的性能对比. 表 14 中显示的结果基本与 iPinYou 上的实验结果一致. 首先，在浅层结构的模型中，只考虑一阶特征的 LR 的性能略低于同时考虑一阶和二阶特征的 FM；其次，基于深度学习的预测模型的性能略优于浅层结构模型的性能；然后，Wide & Deep 和基于投票的融合结构的深度学习模型的性能又进一步优于基于单一结构的深度学习模型；最后，本文提出的 DPSN 模型确实能够获得目前最优的性能. 实验结果的对比如图 15 所示.

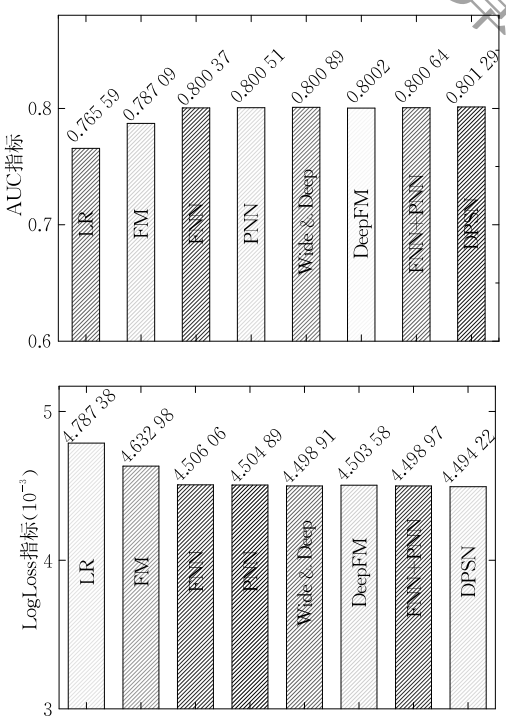


图 15　代表模型在 Criteo 数据集上的性能对比

综上所述，本文在第 4 节通过实验的方式对本文提出的 DPSN 模型的性能进行了全面评价. 实验 1 在 iPinYou 1458 和 3386 数据集上验证了 DPSN 模型的收敛性；实验 2 通过在基本的 FNN 和 PNN 模型中增加一阶特征权重信息作为输入，验证了在嵌入层增加预训练的一阶特征权重节点的有效性；实验 3 通过调整 DPSN 模型的超参数对 DPSN 的预测性能进行了全面分析；实验 4 验证了在嵌入层加入一阶特征权重信息对 DPSN 模型预测性能的提升；实验 5 通过单一 Deep Network 结构和 Product Network 结构，与 DPSN 融合结构的性能对比，验证了 DPSN 融合结构确实能将两个不同 DNN 学习到的高阶特征组合进行融合，从而得到更为有效的高阶特征组合模式；实验 6 通过不同比率的负采样实验，分析了负采样对 DPSN 模型性能的影响；实验 7 通过在两个数据集上将目前已有的典型 CTR 预测模型与本文提出的 DPSN 模型的预测性能进行了对比，从实验的角度验证了 DPSN 模型的优越性.

5　结束语

将深度学习模型应用于 CTR 预测已经逐渐成为一种新的研究趋势. 本论文在深入研究已有的深度学习模型的基础上，提出一种新的融合结构，使得可以分别利用两个不同的深度神经网络学习样本特征的高阶表示，从而捕捉到更多可用信息，提升 CTR 预测的准确率. 论文的主要贡献包括：(1) 设计了一个新嵌入层结构，不仅包括特征的嵌入向量节点，还包括一阶特征的权重单元，实验 4 证明了只需增加很少的参数学习复杂度，即可获得预测性能的提升；(2) 设计了一个新的融合结构，可以巧妙地融合不同的模型进行特征表示的学习，该结构不仅可以融合两个深度模型，也可以融合深度模型和浅层模型，或者两个浅层模型；(3) 通过真实广告投放的数据集对所提出的融合模型进行了性能验证，大量实验表明本文提出的 DPSN 模型确实能够有效提升预测性能，且详细讨论了不同模型参数对预测性能的影响.

尽管 CTR 预测模型的研究已经引起了无论是工业界还是学术界大量的关注，遗憾的是，目前 CTR 预测模型的性能在实际广告投放系统中的表现仍然不太理想. 分析原因包括实际数据与样本数据分布不一致，正样本太少，新广告投放冷启动等问题. 针对正样本太少的问题，目前热门的 GAN[31] 模型可以用来辅助生成正样本；针对新广告投放的冷启动问题，一种可行的解决方案是使用迁移学习，在不同商品的数据集上进行知识的迁移，从而提高预测准确率. 深度学习模型也比较适合应用迁移学习，可以减少模型训练的时间与开销，文献[32]中已经就 DNN 的可迁移性给出了很好的探索和证明. 此

外，强化学习在广告投放中的尝试也取得了很好的效果[33-34]；强化学习具有一些良好的性质，它可以使用无标签数据，直接读取实际活动中的用户行为，这种更加高效；而且它可以轻易实现在线学习，不用离线训练再发布预测，提高了效率；此外它可以利用更多数据特征，取得更好的广告效果. 强化学习已经在阿里和百度的广告商品系统中得到了实际运用，并取得了不错的效果①，因此借鉴强化学习的方法用于 CTR 预测也将是可行的尝试.

参 考 文 献

[1] Chapalle O. Offline evaluation of response prediction in online advertising auctions//Proceedings of the International Conference of World Wide Web. Florence, Italy, 2015: 18-22

[2] Chapalle O, Manavoglu E, Rosales R. Simple and scalable response prediction for display advertising. Journal of ACM Transactions on Intelligent Systems and Technology, 2015, 5(4): 61

[3] Richardson M, Dominowska E, Ragno R. Predicting clicks: Estimating the click-through rate for new ads//Proceedings of the International Conference of World Wide Web. Banff Alberta, Canada, 2007: 8-12

[4] McMahan H B, Holt G, Sculley. D, et al. Ad click prediction: A view from the trenches//Proceedings of the ACM Special Interest Group on Knowledge Discovery in Data. Chicago Illinois, USA, 2013: 11-14

[5] Zhang Weinan, Du Tianming, Wang Jun. Deep learning over multi-field categorical data: A case study on user response prediction//Proceedings of the European Conference on Information Retrieval. Padua, Italy, 2016: 45-57

[6] Juan Yuchin, Zhuang Yong, Chin Wei-Sheng, Lin Chih-Jen. Field-aware factorization machines for CTR prediction//Proceedings of the Association for Computing Machinery Conference on Recommender. Boston, USA, 2016: 43-50

[7] Chang Yin-Wen, Hsieh Cho-Jui, Chang Kai-Wei, et al. Training and testing low-degree polynomial data mappings via linear SVM. Journal of Machine Learning Research, 2010, 11(1): 1471-1490

[8] Beck J E, Woolf B P. High-level student modeling with machine learning//Proceedings of the International Conference on Intelligent Tutoring Systems. Berlin, Germany, 2000: 584-593

[9] Oentaryo R J, Lim Ee-Peng, Low Jia-Wei, et al. Predicting response in mobile advertising with hierarchical importance-aware factorization machine//Proceedings of the ACM International Conference on Web Search and Data Mining. New York, USA, 2014: 123-132

[10] Rendle S. Factorization machines with libFM. ACM Transactions on Intelligent Systems and Technology, 2012, 3(3): 1-22

[11] He Xinran, Pan Junfeng, Jin Ou, et al. Practical lessons from predicting clicks on ads at Facebook//Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York, USA, 2014: 1-9

[12] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks//Proceedings of the International Conference on Neural Information Processing Systems. Lake Tahoe, USA, 2012: 1097-1105

[13] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, Canada, 2013: 6645-6649

[14] Shen Yelong, He Xiaodong, Gao Jianfeng, et al. A latent semantic model with convolutional-pooling structure for information retrieval//Proceedings of the ACM International Conference on Conference on Information and Knowledge Management. Shanghai, China, 2014: 101-110

[15] Qu Yanru, Cai Han, Ren Kan, et al. Product-based neural networks for user response prediction//Proceedings of the IEEE International Conference on Data Mining. Barcelona, Spain, 2016: 1-6

[16] Cheng H-T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems//Proceedings of the Workshop on Deep Learning for Recommender Systems. Boston, USA, 2016: 1-4

[17] Guo Huifeng, Tang Ruiming, Ye Yunming, et al. DeepFM: A factorization-machine based neural network for ctr prediction//Proceedings of the International Joint Conference on Artificial Intelligence. Melbourne, Australia, 2017: 1-7

[18] Lee K-C, Orten B B, Dasdan A, Li Wentong. Estimating conversion rate in display advertising from past performance data//Proceedings of the ACM Special Interest Group on Knowledge Discovery in Data. Beijing, China, 2012: 768-776

[19] Liao Hairen, Peng Lingxiao, Liu Zhengchuan, Shen Xuehua. iPinYou global RTB bidding algorithm competition dataset//Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York, USA, 2014: 1-6

[20] Nair V, Hinton G E. Rectified linear units improve restricted Boltzmann machines//Proceedings of the International Conference on Machine Learning. Haifa, Israel, 2010: 807-814

[21] Marreiros A C, Daunizeau J, Kiebel S J, Friston K J. Population dynamics: Variance and the Sigmoid activation function. NeuroImage, 2008, 42(1): 147-157

[22] Xu Z, Chang X, Xu F, Zhang H. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. IEEE Transactions on Neural Networks & Learning Systems, 2012, 23(7): 1013-1027

[23] Ketkar N. Deep Learning with Python. Berkeley, CA, USA: Apress, 2017

[24] Bottou L. Neural Networks: Tricks of the Trade. Berlin, Germany: Springer, 2012

① 《不一样的技术创新》. https://yq.aliyun.com/articles/68617

[25] Boyd S, Vandenberghe L. Convex Optimization. Cambridge, UK: Cambridge University Press, 2004

[26] Bottou L, Curtis F E, Nocedal J. Optimization methods for large-scale machine learning. Society for Industrial and Applied Mathematics, 2018, 60(2): 223-311

[27] Japkowica N, Stephen S. The class imbalance problem: A systematic study. Intelligent Data Analysis, 2002, 6(5): 429-449

[28] Graepel T, Candela J Q, Borchert T, Herbrich R. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's bing search engine//Proceedings of the International Conference on Machine Learning. Haifa, Israel, 2010: 13-20

[29] Vovk V. The Fundamental Nature of the Log Loss Function. Bern, Switzerland: Springer, 2015, 9300: 307-318

[30] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014, 15(1): 1929-1958

[31] Goodfellow I J, Abadie J P, Mirza M, et al. Generative adversarial nets//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2014: 2672-2680

[32] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2014: 3320-3328

[33] Krakovsky M. Reinforcement renaissance. Communications of the ACM, 2016, 59(8): 12-14

[34] Volodymyr M, Koray K, David S, et al. Playing Atari with deep reinforcement learning//Proceedings of the Neural Information Processing Systems Conference. Lake Tahoe, USA, 2013: 1-9

LIU Meng-Juan, Ph.D., associate professor. Her research interests include data mining, computational advertising, and machine learning.

ZENG Gui-Chuan, M.S. candidate. His research interests include computational advertising and machine learning.

YUE Wei, M.S. candidate. His research interests include computational advertising and machine learning.

LIU Yao, Ph.D., associate professor. Her research interests include social network, machine learning and data mining.

QIN Zhi-Guang, Ph.D., professor. His research interests include data mining and network security.

Background

In recent years, online advertising has developed into a multi-billion dollar industry. As one of the most exciting advances in online advertising, advertising targeted delivery has received increasing attention, since it improves the efficiency and transparency in the online advertising ecosystem. The problem studied in this paper is to use the machine learning models, especially the deep learning models to predict the click-through rates (CTRs), which is a typical regression problem. Due to the huge commercial value of CTR prediction in online advertising, there are extensive researches on CTR prediction this field. The most widely used model is to establish the CTR estimator based on logistic regressions (LR); in addition, factorization machine (FM) and field-aware factorization machine (FFM) have also achieved good results in CTR prediction, because they can explore the sophisticated feature interactions by mapping them into a low dimensional space. Unfortunately, the performance of CTR prediction is not ideal at present, which is because user's click behavior is a very complicated psychological process. There're many factors affecting the click behavior, and the number of positive samples that are really clicked in the dataset is extremely small.

In this paper, we firstly study several typical CTR prediction models, especially the deep learning models based on fusion architecture; and then a new click-through rate prediction model based on a hybrid network is proposed. The new model can integrate flexibly different deep neural networks (DNNs) to learn the high-order representation of original high-dimensional sparse features respectively, which enables the prediction model to take advantage of more abundant information of high-order feature interactions. We evaluate the performance of the proposed model based on a real-world dataset, and the experimental results demonstrate that the new model has better performance than major state-of-the-art models. This work is supported by the National Natural Science Foundation of China (Nos. 61202445, 61502087), and the Fundamental Research Funds for the Central Universities (No. ZYGX2016J096). The authors of this paper have published several related papers on the conferences recommended by CCF, and applied for a number of related patents.