International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n2.08
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Exploration of the Impact of Different Deep Learning Network Architectures on Click-Through Rate Prediction Models Based on Gated Deep Cross Networks

Minyu Jian *

Department of Software Engineering, Jiangxi University of Science and Technology, Nanchang Jiangxi, China

*Corresponding Author: 2831722842@qq.com

ABSTRACT

This paper investigates advertisement click-through rate (CTR) prediction models based on Gated Deep Cross Networks and analyzes the impact of different deep learning network architectures on prediction performance. The paper provides an introduction to the background and related research of CTR prediction tasks, as well as a detailed discussion of the model structure and design. In the experimental section, experiments were conducted using publicly available datasets. Various network structure configurations were tested, including changes in activation functions and dropout rates. The results indicate that certain configurations in the experimental group exhibited superior performance compared to the control group, showing significant advantages. Overall, the paper demonstrates the strong performance of Gated Deep Cross Networks in CTR prediction and explores how different network structures affect model performance. Finally, the research work is summarized, and future research directions are proposed, providing valuable insights and references for CTR prediction tasks.

KEYWORDS

Advertisement Click-Through Rate Prediction; Gated Deep Cross Networks; Loss Function; Experimental Design

1. INTRODUCTION

With the advancement of internet advertising technology, click-through rate prediction plays a crucial role in personalized recommendations and ad placements. [1] As a typical machine learning problem, click-through rate prediction has driven innovation in model design and optimization methods. By exploring new neural network architectures and learning approaches, researchers continuously improve the prediction accuracy and generalization capability of the models. Click-through rate prediction models also have widespread applications in real life.

On major e-commerce platforms, video recommendation services, and music recommendation systems, these models analyze users' historical behavior and interest data to provide personalized content recommendations and enhance user experience. For example, Alibaba's innovative Deep Intent Awareness Network (DIAN) is a notable application in this field. [2] However, despite the success of existing models, challenges remain, such as further improving prediction accuracy and optimizing computational efficiency. This paper aims to enhance current models by specifically adjusting network structures, optimizing parameter settings, and validating the effectiveness of these improvements through experiments.

2. REVIEW OF CLICK-THROUGH RATE PREDICTION RESEARCH

Researchers have conducted numerous studies on click-through rate prediction algorithms. One such study [3] introduces an advertisement click-through rate prediction model based on convolutional neural networks, which integrates a mechanism for factor decomposition and a CNN structure. The model's foundation is a factorization embedding layer that maps high-dimensional sparse features (encoded using one-hot encoding) to latent vector spaces. This approach provides a direction for model construction, namely, to utilize convolutional and pooling layers following the factorization embedding layer for feature extraction and dimension reduction, and to integrate deep features from various layers through a fully connected layer. The model also optimizes performance by modifying activation functions and dropout layers. For instance, replacing the ReLU activation function with the Selu activation function effectively prevents overfitting.

In addition, a click-through rate prediction model based on deep neural networks [4] proposes the Long-Short Term Interest Network (LSTIN). Using AUC as the metric, this model demonstrates improved prediction performance compared to some foundational models in the CTR prediction domain. It enhances the model's perceptual capacity and provides advantages through sequence modeling. Furthermore, the model processes long-term and short-term interests differently, resulting in better performance.

Recently, researchers have proposed the Gated Deep Cross Network (GDCN), an emerging deep learning model that combines the advantages of deep neural networks and cross networks. By capturing high-order interactions between features, this model has made significant advancements [5]. This paper focuses on comparing different deep learning network structures to study their impact on click-through rate prediction performance, aiming to provide theoretical support and empirical validation for improving prediction accuracy and promotional effectiveness.

3. GDCN MODEL STRUCTURE

The Gated Deep Cross Network (GDCN) is composed of an embedding layer, a Gated Cross Network (GCN), and a Deep Neural Network (DNN). The embedding layer transforms high-dimensional sparse inputs into low-dimensional dense representations, converting sparse high-dimensional features into dense low-dimensional embedding matrices. A key distinction of GDCN is the introduction of an information gate that adaptively filters cross features in each layer, rather than uniformly aggregating all features. This allows GDCN to effectively utilize deeper high-order cross information without performance degradation, providing dynamic interpretability for each instance.

The Gated Cross Network (GCN) is the core structure of GDCN, designed to model feature interactions using information gates. In each gated cross-layer, there are two core components: feature crossing and information gates. The input to the (l + 1)-th order gated cross-layer comes from the output features of the previous l-th order layer. However, not all first-order and second-order feature combinations positively impact the model's predictions. As the number of feature crossings increases, the number of cross features grows exponentially, including many combinations that may be detrimental or even harmful to the model. The number of generated cross features grows exponentially with the depth of feature crossing, potentially introducing noise and reducing model performance, i.e., decreasing prediction accuracy or efficiency. This is why information gates are introduced in the model. These gating components adaptively learn and adjust the weights of each first-order and second-order feature crossing. As the number of cross-layers increases, each layer's information gate filters the cross features for the next layer, effectively controlling the information flow. The GDCN structure employs parallel and stacking structures, with DNN and GCN being parallel and stacked respectively. This paper evaluates the GDCN model by assessing both GDCN-p (parallel) and GDCN-s (stacked) configurations [5].

4. EXPERIMENTAL ANALYSIS

This study conducts experiments using GDCN on publicly available datasets, comparing the performance of the model before and after modifications. First, the section introduces the datasets used in the experiments, the model-related parameter settings, the improvements made, and the evaluation metrics for the experimental results.

4.1. Dataset

The experiments in this paper use the Frappe dataset [6] for comparative analysis. The Frappe dataset (see Table 1) contains app usage logs of users under different contexts (e.g., time, weather, city, etc.), with a total of 96,203 app records. In addition to userID and appID, each record includes 8 contextual features: weather, city, time, etc. After applying one-hot encoding, the features have a dimensionality of 5,382, and a label of 1 indicates that the user used the app.

Table 1. Frappe Dataset

Dataset	Instance	Feature	User	Item
Frappe	288,609	5,382	957	4,082

4.2. Parameter Setting

This study utilized the implementation publicly available in the GitHub repository of the authors Fangye Wang et al. [5] as the foundation for this research, enabling the training and evaluation of the model. The experimental environment is detailed in table 2.

Table 2. Experimental Environment

Experimental Environment	Detailed Information
Processor	AMD R5600H 6-core
RAM	16GB
GPU	NVIDIA RTX 3060
Deep Learning Framework	PyTorch 2.2.1
Programming Language	python 3.8.8

The experiments were conducted using PyTorch and based on the publicly available code. The following settings were used: the Adam optimizer was employed for all models with a default learning rate of 0.001. During training, the Reduce-LR-On-Plateau scheduler was utilized, which decreases the learning rate by a factor of 10 if the performance does not improve for 3 consecutive epochs. The batch size was set to 4096. The embedding dimension for all datasets was 16, and all activation functions were ReLU. For GCN, GDCN-S, and GDCN-P, if not specified otherwise, the default number of cross layers is 3.

4.3. Improvement Strategies

Introduce residual connections. When using GDCN-P for click-through rate prediction on the Frappe dataset, during forward propagation, the input data is first subjected to linear transformation and nonlinear activation through a multilayer perceptron, with a dropout parameter set to 0.5 to deactivate some neurons. To address issues such as vanishing and exploding gradients in deep neural networks [7], a newly introduced residual connection is then applied to the input data for linear transformation. The two sets of processed data are subsequently fused, combining the features learned by the multilayer perceptron with the supplementary information learned from the residuals. This enhances the model’s ability to learn complex features, reduces the risk of overfitting, and improves the model’s generalization capability.

Parameter Tuning for Each Layer. To further leverage the advantages of the GDCN model, enhance feature interactions, and capture higher-order interactions among features, we attempted to increase the number of GateCrossLayer layers and set the parameters of the MultiLayer Perceptron to (512, 512, 256). To prevent overfitting, early stopping was employed with a patience value of 3. For DCN-v2-p, the learning rate was adjusted from 0.001 to 0.002.

Modify Network Architecture. Additional MLP layers and two extra fully connected layers were incorporated into the DeepFM model [8]. These layers are concatenated after the outputs of the existing LR, FM, and MLP components. This modification helps capture more complex feature interactions and patterns. During forward propagation, the original LR, FM, and MLP components remain unchanged, but the new fully connected layers are added in the final prediction process.

4.4. Evaluation Metrics

In experiments with click-through rate prediction models, two common metrics are used for evaluation: AUC (Area Under the ROC Curve) and Logloss (Logarithmic Loss) [9].

4.5. Results and Analysis

In this section, five models were selected for comparative analysis. The experimental results are shown in Tables 3 and 4. Table 3 presents the model performance before modifying the network structure, while Table 4 shows the model performance after the modification. The comparison of the two metrics before and after the improvements is illustrated in Figures 1 and 2.

Table 3. Performance of the Model Before Improvements

Model Name	AUC	Logloss
DeepFM	0.977963	0.219289
GDCN-p	0.978069	0.331797
GDCN-s	0.978221	0.225335
DCN-v2-p	0.974238	0.220868
DCN-v2-s	0.977648	0.184890

Table 4. Performance of the Improved Model

Model Name	AUC	Logloss
DeepFM	0.978037	0.203236
GDCN-p	0.978260	0.322841
GDCN-s	0.979613	0.221773
DCN-v2-p	0.976018	0.217208
DCN-v2-s	0.978175	0.180090

From the experimental results in Tables 3 and 4, it can be observed that improving the network structure through experimentation leads to performance enhancements for the models on the Frappe dataset. The stacked structure model GDCN-s, after improvements, achieved a 0.14% increase in AUC and a 0.36% decrease in Logloss on the Frappe dataset. Similarly, the DCN-v2-s model, also with a stacked structure, saw a 0.05% increase in AUC and a 0.48% decrease in Logloss after improvements. For the DCN-v2-p model, changing the learning rate from 0.001 to 0.002 resulted in a 0.19% increase in AUC and a 0.37% decrease in Logloss. Overall, whether before or after the improvements, the GDCN model consistently outperforms other models. The AUC value predicted by GDCN-p is 0.02% higher than that of the DeepFM model, 0.22% higher than that of the DCN-v2-p model, and 0.01% higher than that of the DCN-v2-s model. The AUC value predicted by GDCN-s is 0.12% higher than that of the DeepFM model, 0.36% higher than that of the DCN-v2-p model, and

0.14% higher than that of the DCN-v2-s model. This indicates that the improvement methods are indeed effective and have significantly enhanced the predictive ability of the click-through rate prediction models.

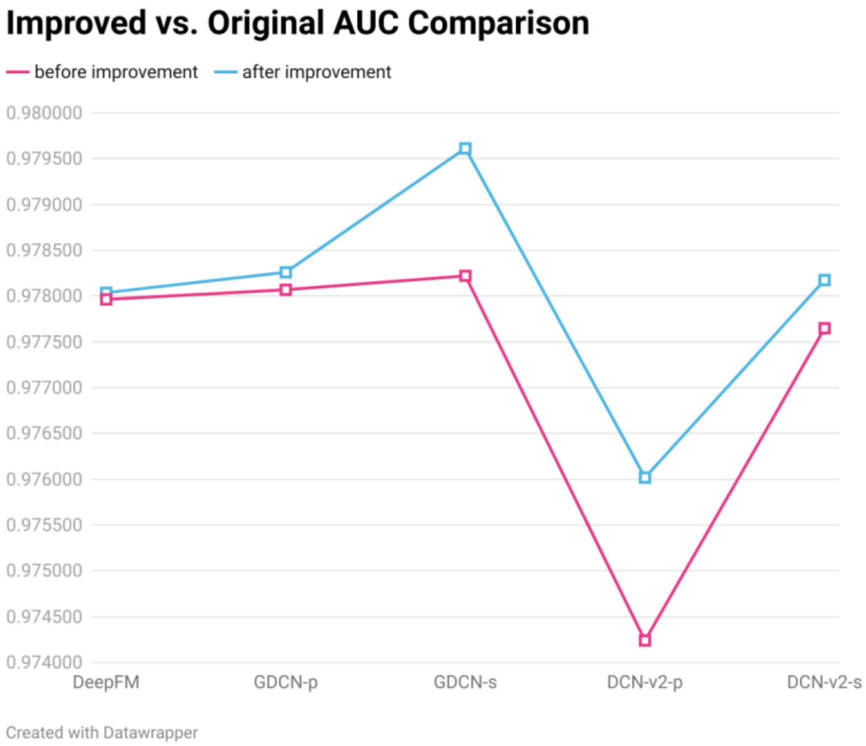

Figure 1. AUC Comparison Before and After Improvements

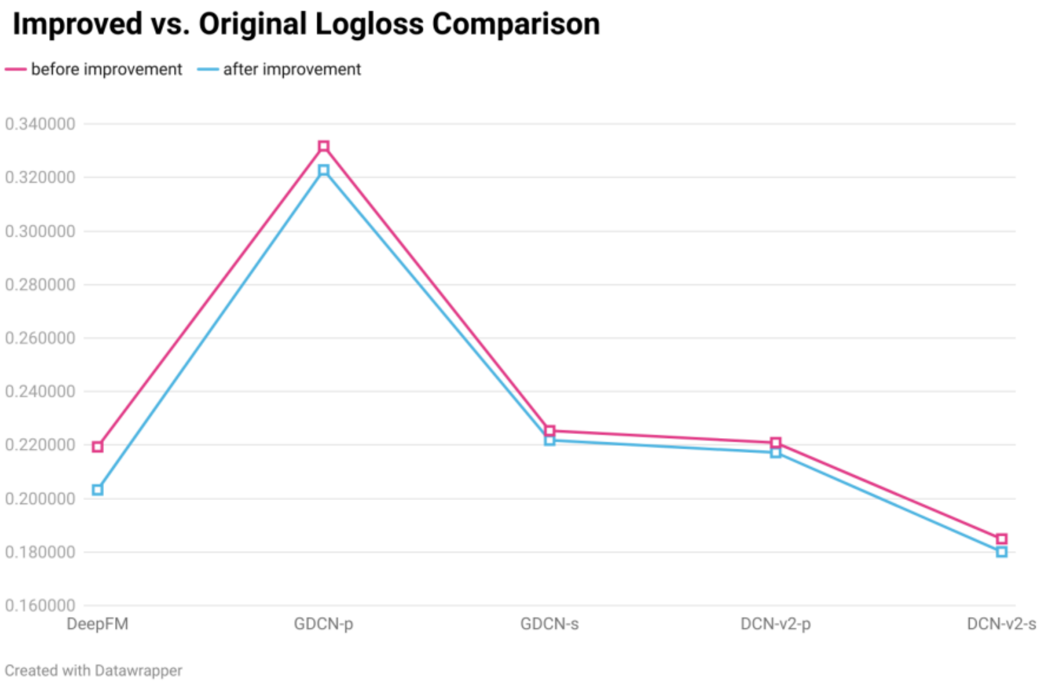

Figure 2. Logloss Comparison Before and After Improvements

5. CONCLUSION

This paper conducts a series of experiments on five models to explore their performance in click-through rate prediction. The improvement methods vary depending on the model. For instance, residual connections were introduced to the GDCN model, fully connected layers were added to the DeepFM model, and parameter fine-tuning such as setting the parameters for multilayer perceptrons and dropout rates was applied to optimize these five models to some extent.

REFERENCES

[1] Ying Xie, Dan Jiang, Xinmei Wang, et al. Robust transfer integrated locally kernel embedding for click-through rate prediction. Information Sciences. 2019; 491 (0):190-203. doi:10.1016/j.ins.2019.04.006.

[2] XIA Y, CAO Y, HU S, et al. Deep Intention-Aware Network for Click-Through Rate Prediction [M/OL]. arXiv, 2022 [2024-07-26]. http://arxiv.org/abs/2211.08650.

[3] Wan Man. Research on Advertisement Click-Through Rate Prediction Model Based on Convolutional Neural Networks [D]. Southwest University, 2018.

[4] Liu Hongli, Wu Sen, Wei Guiying, Li Xin, Gao Xiaonan. Click-Through Rate Prediction Model Based on Deep Neural Networks [J]. Journal of Engineering Science, 2022, Vol. 44 (11): 1917-1925.

[5] Wang Feng, Gu Hui, Li Dan, et al. Towards Deeper, Lighter, and Interpretable Cross Network for CTR Prediction [M/OL]. arXiv, 2023 [2024-04-21]. http://arxiv.org/abs/2311.04635.

[6] Linas Baltrunas, Karen Church, Alexandros Karatzoglou, and Nuria Oliver. 2015. Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. arXiv preprint arXiv:1505.03014 (2015).

[7] Li Yanzhi, Guo Limin, Zhang Weiguo, et al. High-Frequency and High-Order Spatial Interaction Recognition Algorithm for Autoclave Leakage Sound Research [J/OL]. Computer Measurement & Control, 1-10 [2024-07-26]. http://kns.cnki.net/kcms/detail/11.4762.TP.20240626.0932.002.html.

[8] Yang Lin. Research on Advertisement Click-Through Rate Prediction Based on Improved DeepFM Model [D]. Shandong University, 2022. DOI:10.27272/d.cnki.gshdu.2022.003856.

[9] Xia Yichun, Li Wanggen, Li Doudou, et al. Click-Through Rate Prediction Method Based on Improved Fi-GNN Model [J]. Computer Engineering and Design, 2024, 45 (06): 1720-1727. DOI:10.16208/j.issn1000-7024.2024.06.017.