GemNN: Gating-Enhanced Multi-Task Neural Networks with Feature Interaction Learning for CTR Prediction

Hongliang Fei[1], Jingyuan Zhang[1], Xingxuan Zhou[2], Junhao Zhao[2], Xinyang Qi[2], Ping Li[1]
1. Cognitive Computing Lab, Baidu Research
2. Baidu Search Ads (Phoenix Nest), Baidu Inc.
10900 NE 8th St, Bellevue, Washington 98004, USA
No.10 Xibeiwang East Road, Beijing 100193, China
{hongliangfei, zhangjingyuan03, zhouxingxuan, zhaojunhao, qixinyang, liping11}@baidu.com

ABSTRACT

Deep neural network (DNN) models have been widely used for click-through rate (CTR) prediction in online advertising. The training framework typically consists of embedding layers and multi-layer perceptions (MLP). At Baidu Search Ads (a.k.a. Phoenix Nest), the new generation of CTR training platform has become *PaddleBox*, a GPU-based parameter server system. In this paper, we present Baidu's recently updated CTR training framework, called Gating-enhanced Multi-task Neural Networks (GemNN). In particular, we develop a neural network based multi-task learning model to predict CTR in a coarse-to-fine manner, which gradually reduces ad candidates and allows parameter sharing from upstream tasks to downstream tasks to improve the training efficiency. Also, we introduce a gating mechanism between embedding layers and MLP to learn feature interactions and control the information flow fed to MLP layers. We have launched our solution in Baidu PaddleBox platform and observed considerable improvements in both offline and online evaluations. It is now part of the current production system.

CCS CONCEPTS

• **Information systems → Computational advertising**.

KEYWORDS

CTR prediction, Gating, Multi-task Learning, Feature Interaction

ACM Reference Format:
Hongliang Fei, Jingyuan Zhang, Xingxuan Zhou, Junhao Zhao, Xinyang Qi, and Ping Li. 2021. GemNN: Gating-enhanced Multi-task Neural Networks with Feature Interaction Learning for CTR Prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3404835.3463116

1 INTRODUCTION

The click-through rate (CTR) prediction is a critical task in commercial online advertising systems [7, 10, 14, 40, 44, 49]. In cost-per-click (CPC) advertising systems, the expected revenue is measured by cost per mille (CPM), which is the product of the bid price and CTR. Obviously, the accuracy of CTR prediction has a crucial impact on the revenue, and intensive efforts have been devoted to improving CTR models [2, 8, 9, 11, 12, 14, 17, 18, 25, 26, 29, 33, 34, 43, 46, 50, 51].

Industrial advertising systems usually select and rank ads from millions of candidates. It is a common practice (e.g., [10]) to leverage a multi-layer funnel-shaped structure with several stages to deliver ads: the candidate generation stage and the re-ranking stage. The candidate generation stage reduces the corpus size from millions to thousands or hundreds, and the re-ranking stage estimates CTR and CPMs of ad candidates and delivers top-ranked ads to users. During the re-ranking stage, deep neural network (DNN) based models have been widely used [3, 6, 22, 24, 26, 35, 36, 47, 49, 50, 52]. The solution framework usually follows a similar paradigm: embedding layers followed by an MLP. For those models, large-scale sparse input features are first converted into embedding vectors, then projected into fixed-length vectors, and concatenated to feed into several fully-connected (FC) layers to learn the nonlinear relations among features. It is worth noting that there are usually several sub-steps in the re-ranking sage in industrial advertising systems, in which there is one neural model for each sub-step to reduces ad candidates. For example, we can first rank ads based on user query and ad features, then incorporate ad material types and possible rank information to re-rank ads and finally deliver a few top ads to users.

There are well-known challenges in industrial settings. Firstly, users, ads, and ad material types are normally characterized by large-scale categorical features, which result in a tremendous amount of model parameters due to the sparse feature embeddings. Secondly, effective feature interactions are crucial to CTR models' success since they provide additional interaction information beyond individual features. Nevertheless, recent research [1, 41] revealed that vanilla DNNs cannot even efficiently approximately model 2nd or 3rd-order feature interactions. Therefore learning effective feature interactions is a critical issue for CTR models. Several papers have attempted to handle feature interactions [1, 3, 4, 15, 20, 28, 30, 32, 37, 39, 41, 42, 45], but very few works study it under computation and latency constraints in real industrial production settings.

There is a history of development in advertising technologies at Baidu Search Ads (a.k.a. "Phoenix Nest") [10, 44, 49]. As early as 2013, Baidu adopted MPI-based distributed deep learning platforms for CTR models. Recently, GPU-based ads systems [44, 49, 50] ("PaddleBox", www.paddlepaddle.org.cn), have replaced CPU-MPI platforms. Another major effort is the use of approximate near neighbor search and maximum inner product search [10, 38, 48, 53] to improve quality of recalls in the early stage of the training pipeline.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463116

Contributions. In this paper, we report Baidu’s latest framework for CTR prediction, called Gating-enhanced Multi-task Neural Networks (GemNN). Specifically, we develop a neural network based multi-task learning model for predicting CTR in a coarse-to-fine manner, allowing parameter sharing from upper-level tasks to lower-level tasks to improve training efficiency. Unlike ordinary multi-task neural network models that share intermediate layers among tasks, our method leverages commonalities between upstream and downstream tasks to share parameters and avoid duplicated computation. Besides, we introduce a gating mechanism between embedding layers and MLP to simultaneously model feature interactions and learn bit-wise level feature importance for the input to MLP. We have launched this solution in production and achieved considerable improvements in offline AUC and online metric cost per mille (CPM). This paper also reports extensive experiments to demonstrate our multi-task model’s utility and the choice of placing the gating layer.

Related Works. Our work is related to recent research on feature interaction learning [1, 3, 15, 16, 20, 21, 27, 28, 37, 41, 42]. Most notably, DCN [41] and DCN-M [42] learned effective explicit and implicit feature interactions at embedding layers and crossing layers. AutoInt [37] modeled feature interactions via self-attention. GateNet [20] utilized gating layers within each feature field and MLP to select salient latent information at the feature-level and higher hidden-layer level. Inspired by GateNet [20], in our framework, we also leverage a gating mechanism to control salient latent information flow. The difference between GateNet [20] and ours is that our gating layer is positioned between the embedding layer and MLP. Therefore our gating is applied to all features instead of each feature field individually. We will demonstrate the superiority of this configuration through offline and online evaluation. Compared with DCN-M [42] on gating perspective, our approach might be viewed as a simplified version of DCN-M without a mixture of experts. However, we explore how to effectively share parameters for the re-ranking task in a real production environment.

2 METHODOLOGY

In this section, we describe our multi-task model to learn feature interactions for CTR prediction. Given a user’s query and its relevant ad candidates from the retrieval stage, we aim to design a framework for delivering several highly ranked ads to users, including both ranks and their corresponding material types (mt) (e.g., size, position, representation, etc.). Although we have a much smaller ad candidate space after the retrieval stage, the combinatory space across ads, ranks, and material types is still huge.

2.1 Multi-task Neural Network based model

To develop a feasible solution under computation and latency constraints in real-production settings, we decompose the re-ranking procedure into three tasks in a coarse-to-fine manner. For each task, GemNN starts with an embedding layer, followed by one gating layer that models explicit feature interactions and selects salient feature information. Meanwhile, we allow parameter sharing from the coarse (upstream) task to fine (downstream) level tasks. Below we first introduce the three tasks, and then describe how we introduce the gating mechanism into them.

User-ad ranking (UAR). This is a coarse-level task, which takes user query and ad candidates from the retrieval stage to generate a shorter list (e.g. < 20). We cast it as a binary classification problem and use the cross-entropy loss that is commonly used for learning-to-rank systems, especially with a binary label (e.g., click or not). Since we only use rough features of ads, such as ids and bidding words, this task still serves as a retrieval model to reduce ad space.

Ad-mt matching (AMM). Given the top ad candidates from the query-ad ranking task, AMM selects material types for each ad candidate such that it will have a higher probability of being clicked. We adopt a two-tower styled DNN model [19] to match ads and all possible material types. The reason for using a two-tower model is that material type feature embeddings can be pre-computed and indexed, which will save a significant amount of time during ad serving. Similarly, we also cast this task as a binary classification problem. Clicked ads with a certain material type are positive samples, and non-clicked ads with material types are negatives.

User-ad-mt ranking (UAMR). Given a shorter list of ads from query-ad ranking and selected material types from ad-mt matching, UAMR leverages all available features to generate the final top ads with material type information based on projected CPMs. This task is the most fine-grained one, which estimates CTR and CPM based on all possible displayed ad queues. Similar to the previous two tasks, we also cast it as a binary classification problem.

From the above description, we find that the three tasks actually share several common features among users and ads. It is reasonable to enable parameter sharing from coarse tasks to finer tasks to avoid duplicated modeling on those features. Towards that end, we propose a parameter sharing mechanism as shown in Figure 1. We use the common features of users and ads to build UAR model, and then share the first layer of MLP from UAR and its predicted CTR (pCTR) value to the AMM task and QAMR task. Particularly, the shared parameters are concatenated with the embedding layer in AMM (left ad tower) and UAMR as warm-started features. During training, we jointly optimize the three tasks and update the shared MLP layer from UAR while freezing the shared pCTR feature. Such a design seamlessly connects the three tasks and allows parameter sharing from upstream tasks to downstream tasks.

2.2 Gate-enhanced Multi-task NN model

The gating mechanism has been widely adopted in many well-known deep models (e.g. LSTM [13], GRU [5], MMOE [31]). Gates normally output a scalar, which represents the importance of the whole vector embedding. In GateNet [20], they learn the bit-level salient information in the feature embedding so that they can enable gate output to contain fine-grained information about the feature embedding. GateNet [20] has demonstrated the benefit of bit-level v.s. vector-level weights.

We also use the gating mechanism to control salient information flow to downstream layers as shown in Figure 2. Different from GateNet [20], our gating layer is placed between the embedding layer and MLP (after batch normalization) for every DNN module in Figure 1. Therefore ours is applied on all concatenated feature bits (neuron units) instead of each feature field. Our design is a simplified version of DCN-M [42] without mixture of experts or weight matrix decomposition. Note that we do not share gating

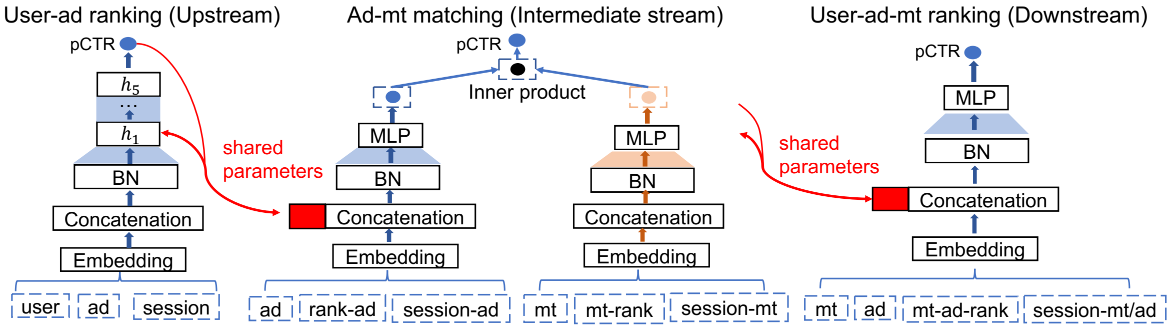

Figure 1: Our multi-task NN model with parameter sharing. Input features are either one-hot or multi-hot vectors in a multi-group categorical form. We share the first layer of MLP and the predicted CTR from the user-ad ranking task (built on common features of users and ads). During training, the shared pCTR feature is frozen. We have five layers in MLP for the user-ad ranking task and two layers for the other two tasks. All tasks are cast as binary classification problems. Double arrow curves indicate that gradients can be back-propagated to shared parameters h_1. BN: batch normalization.

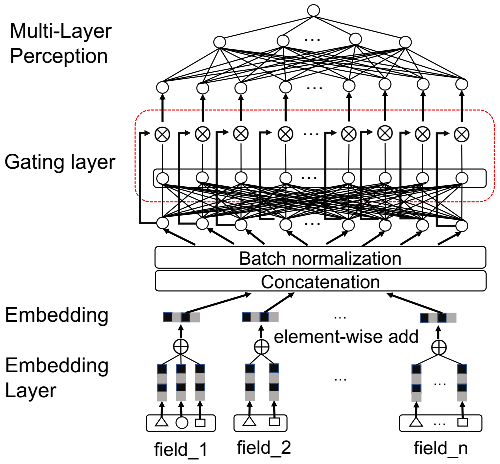

Figure 2: Gating-layer for DNN model. We use a general DNN architecture for demonstration. The gating layer is inserted between the normalization layer and MLP at every DNN module in Figure 1 to control the salient information flow. We perform sum-pooling for multi-hot features. ⊕ indicates element-wise add and ⊗ indicates element-wise product.

layers among tasks. Instead we let downstream tasks learn their gating weights for the shared parameters since we want to re-evaluate the importance for them.

Mathematically, let $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n]^T$ be the concatenated embeddings, where n is the number of feature slots and $\mathbf{e}_i \in \mathcal{R}^d$ is the embedding vector for field i. We calculate the gate value which represents the bit-level importance of concatenated embedding $\mathbf{E}$ as $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n]^T = act(\mathbf{W} * \mathbf{E} + \mathbf{b})$, where $act(.)$ is the activation function, $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n]^T$ is the bias term and $\mathbf{b}_i \in \mathcal{R}^d$. Both $\mathbf{W} \in \mathcal{R}^{nd \times nd}$ and $\mathbf{b}$ are trainable parameters. We use sigmoid activation function and set $d = 16$ in this paper.

With bit-level importance vector $\mathbf{G}$, we can compute the input to MLP as $\mathbf{Y} = [\mathbf{g}_1 \otimes \mathbf{e}_1, \mathbf{g}_2 \otimes \mathbf{e}_2, \cdots, \mathbf{g}_n \otimes \mathbf{e}_n]$, where $\otimes$ indicates element-wise product. Hence the input $\mathbf{Y}$ is filtered embeddings, which is controlled by bit-wise gate values. Since the gating layer is applied on all feature fields, we actually implicitly learn feature interactions and use them to decide gating values.

3 EXPERIMENT

We collect a period of user click history logs from Baidu search ads system for evaluation. The size of the training data is around 56 billion. There are 252, 65, and 102 feature slots for UAR, AMM, and UAMR, respectively. We omit the statistics of feature slots due to space constraints. We conduct both offline and online evaluations. The offline testing data size is about 500 million, and we compute the AUC of CTR prediction. The trained GemNN model is evaluated over the production environment of our search engine in an A/B testing manner. The relative improvement of online CPM is also reported as $(CPM_{new} - CPM_{old})/CPM_{old}$, where $CPM = bid \times CTR$.

3.1 Performance Evaluation

We compare GemNN with three baselines: GateNet [20], AutoInt [37] and DCN-M [42]. For all methods, the embedding dimension is 16 and the total number of features after embedding is around 100 billion. Adam optimizer [23] is applied with a mini-batch size of 2048. The learning rate, the number of hidden layers and the hidden dimension are set via a grid search for each sub-task. The number of hidden layers ranges from 3 to 6, with the hidden dimension from 16 to 1024. The learning rate ranges from 5.5e−6 to 8.5e−6. For GateNet, a bit-wise hidden gate is inserted into MLP layers. For AutoInt, the number of attention heads is 4, and the attention embedding size is 64. For DCN-M, we use the stacked structure with two cross layers. The number of experts is 3 and the rank of the weight matrix is 128.

Table 1 lists the best AUC performance of different models over different sub-tasks. The percentage value in "(·)" is the improvement over GateNet. Note that for a commercial search engine with massive user activities, an improvement of 0.1% in AUC is usually considered as significant for the CTR prediction and it will

Table 1: Performance of AUC for different models.

Method	UAR	AMM	UAMR
GateNet	0.8106	0.8211	0.8238
AutoInt	0.8198 (+0.92%)	0.8314 (+1.03%)	0.8313 (+0.75%)
DCN-M	0.8212 (+1.06%)	0.8343 (+1.32%)	0.8359 (+1.21%)
GemNN	**0.8221 (+1.15%)**	**0.8355 (+1.44%)**	**0.8373 (+1.35%)**

lead to a large increase in revenue. We can observe that the proposed GemNN achieves the best performance on the AUC offline evaluation. Compared with GateNet and DCN-M with the gating mechanism, GemNN has significant improvements on all the tasks, showing the effectiveness of the multi-task learning procedure. In addition, GemNN outperforms AutoInt. It implies that the multi-task learning procedure plus the gating mechanism can help capture feature interactions more effectively than self-attention. To summarize, with multi-task learning and the gating strategy, GemNN obtains the best results.

3.2 Analysis

In this section, we first conduct experiments to study the influence of the gating mechanism in GemNN under different settings. Then we do an ablation study to analyze the contribution of multi-task learning and gating in GemNN.

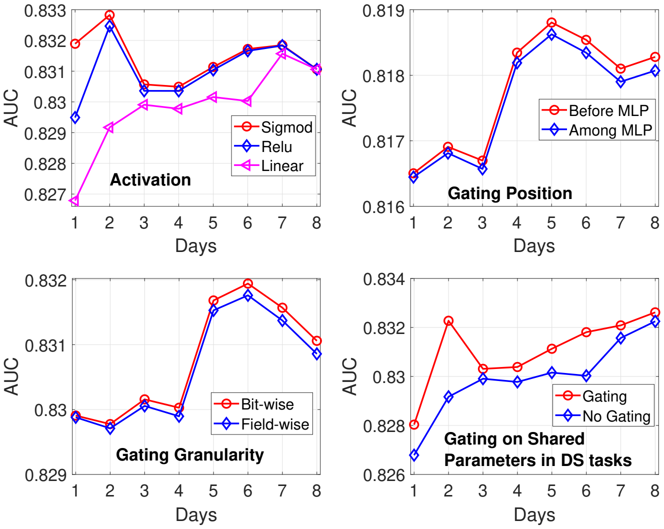

Figure 3: Performance of different strategies in the gating layer. "DS" in the lower-right subplot indicates downstream.

Activation Function Study. Different activation functions can be applied to the gating layer of GemNN. In the experiment, we test Linear, ReLu, and Sigmoid functions and show the AUC performance on UAMR in the upper left of Figure 3. We observe that ReLu outperforms the linear function and the best activation function on the gating layer is Sigmoid. Compared with Sigmoid, ReLu can achieve similar results. In practice, either ReLu and Sigmoid can be chosen as the activation function of the gating layer. In our experiment, we use Sigmoid.

Gating Layer Position Study. The gating layer can be placed either into MLP hidden layers, or between embedding layers and MLP. The former applies gating on the feature field individually, while the latter focuses on all the features to model high-order interactions among different fields. Here we study how the gating layer's different positions will influence the result of GemNN. The upper right of Figure 3 shows the AUC performance on UAR. We observe that the gating mechanism between embedding layers and MLP helps GemNN capture the implicit high-order feature interactions more effectively. Therefore, we put the gating layer between embedding layers and MLP.

Gating Granularity Study. In the gating layer, field-wise gating represents the feature-level importance of embeddings, while bit-wise gating learns the element-level importance. The field-wise representation focuses on coarse-grained information of feature embeddings, and the bit-wise gating contains the fine-grained importance of embeddings. In this section, we conduct experiments to explore the field-wise and bit-wise embedding gates. The lower left of Figure 3 shows the AUC result of GemNN on UAMR. We discover that bit-wise gating performs better than field-wise gating. In our experiment, we use bit-wise gating.

Gating Strategy for Shared Parameters. To improve training efficiency, GemNN allows parameter sharing from upstream tasks to downstream tasks. In this paper, we let downstream tasks learn their gating weights for the shared parameters. On the contrary, we can directly concatenate the shared parameters from UAR task with the output of the gating layer from lower-level tasks. In this section, we study how the gating mechanism on shared parameters will influence the result of GemNN. The lower right of Figure 3 shows the AUC performance on UAMR. We notice that learning task specific gating weights for shared parameters significantly outperforms its opposite. In the experiment, we re-compute the gating weights for shared parameters in the downstream tasks.

Ablation Study. To study how much contribution each component can make to the GemNN model, we conduct an ablation study in Table 2. The percentage value in "(·)" is the improvement over the baseline "GemNN w/o Gating or MTL" (multi-task learning). We observe that MTL helps improve the offline CTR prediction with an average of 0.07% improvement. After adding the gating mechanism, GemNN significantly improves the CTR prediction and achieves the best results. For the online CPM improvement metric, adding MTL to the baseline has 0.42% of improvement. After adding the gating mechanism, our full model GemNN obtains an improvement of 1.26% compared with the baseline.

Table 2: Ablation Study with "GemNN w/o Gating or MTL" as the baseline. "(·%)" is absolute improvement for offline test, while "·%" is relative improvement for online test.

Method	Offline Test			Online Test
	UAR	AMM	UAMR	CPM
GemNN w/o Gating or MTL	0.8204	0.8330	0.8346	0%
GemNN w/o Gating	0.8208 (+0.04%)	0.8339 (+0.09%)	0.8355 (+0.09%)	+0.42%
GemNN	0.8221 (+0.17%)	0.8355 (+0.25%)	0.8373 (+0.27%)	+1.26%

4 CONCLUSION

In this paper, we propose a multi-task model to decompose the CTR prediction problem into three tasks in a coarse-to-fine manner. Our model allows parameter sharing from upstream tasks to downstream tasks to avoid duplicated parameter learning. Meanwhile, the gating mechanism enables us to model feature interactions and control salient information flow from embedding layers to MLP. We have launched the method in Baidu search ads system and achieved considerable improvements in both offline and online evaluation.

REFERENCES

[1] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*. Marina Del Rey, CA, 46–54.

[2] Andrei Broder. 2002. A taxonomy of web search. *SIGIR Forum* 36, 2 (2002), 3–10.

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS@RecSys)*. Boston, MA, 7–10.

[4] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive Factorization Network: Learning Adaptive-Order Feature Interactions. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*. New York, NY, 3609–3616.

[5] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 1724–1734.

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. Boston, MA, 191–198.

[7] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review* 97, 1 (March 2007), 242–259.

[8] Bora Edizel, Amin Mantrach, and Xiao Bai. 2017. Deep Character-Level Click-Through Rate Prediction for Sponsored Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Shinjuku, Tokyo, 305–314.

[9] Daniel C. Fain and Jan O. Pedersen. 2006. Sponsored search: A brief history. *Bulletin of the American Society for Information Science and Technology* 32, 2 (2006), 12–13.

[10] Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. MOBIUS: Towards the Next Generation of Query-Ad Matching in Baidu's Sponsored Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. Anchorage, AK, 2509–2517.

[11] Hongliang Fei, Shulong Tan, Pengju Guo, Wenbo Zhang, Hongfang Zhang, and Ping Li. 2020. Sample Optimization For Display Advertising. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*. Virtual Event, Ireland, 2017–2020.

[12] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*. Macao, China, 2301–2307.

[13] Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* 12, 10 (2000), 2451–2471.

[14] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-Scale Bayesian Click-Through rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*. Haifa, Israel, 13–20.

[15] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*. Melbourne, Australia, 1725–1731.

[16] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He, and Zhenhua Dong. 2018. DeepFM: An End-to-End Wide & Deep Learning Framework for CTR Prediction. *CoRR* abs/1804.04950 (2018).

[17] Wei Guo, Ruiming Tang, Huifeng Guo, Jianhua Han, Wen Yang, and Yuzhou Zhang. 2019. Order-aware Embedding Neural Network for CTR Prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Paris, France, 1121–1124.

[18] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñonero Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD)*. New York City, NY, 5:1–5:9.

[19] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM)*. San Francisco, CA, 2333–2338.

[20] Tongwen Huang, Qingyun She, Zhiqiang Wang, and Junlin Zhang. 2020. GateNet: Gating-Enhanced Deep Network for Click-Through Rate Prediction. *arXiv preprint arXiv:2007.03519* (2020).

[21] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*. Copenhagen, Denmark, 169–177.

[22] Yuchin Juan, Damien Lefortier, and Olivier Chapelle. 2017. Field-aware Factorization Machines in a Real-world Online Advertising System. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW)*. Perth, Australia, 680–688.

[23] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, Yoshua Bengio and Yann LeCun (Eds.). San Diego, CA.

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[25] Feng Li, Zhenrui Chen, Pengjie Wang, Yi Ren, Di Zhang, and Xiaoyu Zhu. 2019. Graph Intention Network for Click-through Rate Prediction in Sponsored Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Paris, France, 961–964.

[26] Zeyu Li, Wei Cheng, Yang Chen, Haifeng Chen, and Wei Wang. 2020. Interpretable Click-Through Rate Prediction through Hierarchical Attention. In *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining (WSDM)*. Houston, TX, 313–321.

[27] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Fi-GNN: Modeling Feature Interactions via Graph Neural Networks for CTR Prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. Beijing, China, 539–548.

[28] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. London, UK, 1754–1763.

[29] Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. 2019. Feature Generation by Convolutional Neural Network for Click-Through Rate Prediction. In *Proceedings of the World Wide Web Conference (WWW)*. San Francisco, CA, 1119–1129.

[30] Bin Liu, Niannan Xue, Huifeng Guo, Ruiming Tang, Stefanos Zafeiriou, Xiuqiang He, and Zhenguo Li. 2020. AutoGroup: Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*. Virtual Event, China, 199–208.

[31] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. London, UK, 1930–1939.

[32] Junwei Pan, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu. 2018. Field-weighted Factorization Machines for Click-Through Rate Prediction in Display Advertising. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web (WWW)*. Lyon, France, 1349–1357.

[33] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on Long Sequential User Behavior Modeling for Click-Through Rate Prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. Anchorage, AK, 2671–2679.

[34] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating the Click-Through Rate for New Ads. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*. Banff, Canada, 521–530.

[35] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117.

[36] Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and J. C. Mao. 2016. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. San Francisco, CA, 255–262.

[37] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. Beijing, China, 1161–1170.

[38] Shulong Tan, Zhixin Zhou, Zhaozhuo Xu, and Ping Li. 2020. Fast Item Ranking under Neural Network based Measures. In *International Conference on Web Search and Data Mining (WSDM)*.

[39] Michael Tsang, Dehua Cheng, Hanpeng Liu, Xue Feng, Eric Zhou, and Yan Liu. 2020. Feature Interaction Interpretability: A Case for Explaining Ad-Recommendation Systems via Neural Interaction Detection. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia.

[40] Hal R. Varian. 2007. Position auctions. *International Journal of Industrial Organization* 25, 6 (2007), 1163 – 1178.

[41] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17*. Halifax, Canada, 12:1–12:7.

[42] Ruoxi Wang, Rakesh Shivanna, Derek Z Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H Chi. 2020. DCN-M: Improved Deep & Cross Network for Feature Cross Learning in Web-scale Learning to Rank Systems. *arXiv preprint arXiv:2008.13535* (2020).

[43] Shu Wu, Feng Yu, Xueli Yu, Qiang Liu, Liang Wang, Tieniu Tan, Jie Shao, and Fan Huang. 2020. TFNet: Multi-Semantic Feature Interaction for CTR Prediction. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*. Virtual Event, China, 1885–1888.

[44] Zhiqiang Xu, Dong Li, Weijie Zhao, Xing Shen, Tianbo Huang, Xiaoyun Li, and Ping Li. 2021. Agile and Accurate CTR Prediction Model Training for Massive-Scale Online Advertising Systems. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD)*. Virtual Event, Xi'an, Shaanxi, China.

[45] Niannan Xue, Bin Liu, Huifeng Guo, Ruiming Tang, Fengwei Zhou, Stefanos P Zafeiriou, Yuzhou Zhang, Jun Wang, and Zhenguo Li. 2020, early access. AutoHash: Learning Higher-order Feature Interactions for Deep CTR Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2020, early access).

[46] Tan Yu, Yi Yang, Yi Li, Xiaodong Chen, Mingming Sun, and Ping Li. 2020. Combo-Attention Network for Baidu Video Advertising. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. Virtual Event, CA, 2474–2482.

[47] Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei (Mark) Zhang. 2016. DeepIntent: Learning Attentions for Online Advertising with Recurrent Neural Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. San Francisco, CA, 1295–1304.

[48] Weijie Zhao, Shulong Tan, and Ping Li. 2020. SONG: Approximate Nearest Neighbor Search on GPU. In *Proceedings of the 35th IEEE International Conference on Data Engineering (ICDE)*. Dallas, TX, 1033–1044.

[49] Weijie Zhao, Deping Xie, Ronglai Jia, Yulei Qian, Ruiquan Ding, Mingming Sun, and Ping Li. 2020. Distributed Hierarchical GPU Parameter Server for Massive Scale Deep Learning Ads Systems. In *Proceedings of the 3rd Conference on Machine Learning and Systems (MLSys)*. Austin, TX.

[50] Weijie Zhao, Jingyuan Zhang, Deping Xie, Yulei Qian, Ronglai Jia, and Ping Li. 2019. AIBox: CTR Prediction Model Training on a Single Node. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. Beijing, China, 319–328.

[51] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*. Honolulu, HI, 5941–5948.

[52] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. London, UK, 1059–1068.

[53] Zhixin Zhou, Shulong Tan, Zhaozhuo Xu, and Ping Li. 2019. Möbius Transformation for Fast Inner Product Search on Graph. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 8216–8227.