Towards Understanding the Overfitting Phenomenon of Deep Click-Through Rate Prediction Models

Zhao-Yu Zhang*
zhangzhaoyu@lamda.nju.edu.cn
Nanjing University
Nanjing, China

Xiang-Rong Sheng*
Yujing Zhang*
xiangrong.sxr@alibaba-inc.com
jinghan.zyj@alibaba-inc.com
Alibaba Group
Beijing, China

Biye Jiang
Shuguang Han✉
biye.jby@alibaba-inc.com
shuguang.sh@alibaba-inc.com
Alibaba Group
Beijing, China

Hongbo Deng
dhb167148@alibaba-inc.com
Alibaba Group
Beijing, China

Bo Zheng
bozheng@alibaba-inc.com
Alibaba Group
Beijing, China

ABSTRACT

Deep learning techniques have been applied widely in industrial recommendation systems. However, far less attention has been paid to the overfitting problem of models in recommendation systems, which, on the contrary, is recognized as a critical issue for deep neural networks. In the context of Click-Through Rate (CTR) prediction, we observe an interesting one-epoch overfitting problem: the model performance exhibits a dramatic degradation at the beginning of the second epoch. Such a phenomenon has been witnessed widely in real-world applications of CTR models. Thereby, the best performance is usually achieved by training with only one epoch. To understand the underlying factors behind the one-epoch phenomenon, we conduct extensive experiments on the production data set collected from the display advertising system of Alibaba. The results show that the model structure, the optimization algorithm with a fast convergence rate, and the feature sparsity are closely related to the one-epoch phenomenon. We also provide a likely hypothesis for explaining such a phenomenon and conduct a set of proof-of-concept experiments. We hope this work can shed light on future research on training more epochs for better performance.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Deep Learning, Recommender Systems, Click-Through Rate Prediction, Overfitting, Sparse Feature

ACM Reference Format:
Zhao-Yu Zhang, Xiang-Rong Sheng, Yujing Zhang, Biye Jiang, Shuguang Han✉, Hongbo Deng, and Bo Zheng. 2022. Towards Understanding the Overfitting Phenomenon of Deep Click-Through Rate Prediction Models. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557479

1 INTRODUCTION

Deep learning techniques have driven advances in many application domains, ranging from computer vision [14, 23], natural language processing [9, 27] to recommender systems [6, 7]. Along with its industrial prevalence, some research studies have investigated the overfitting problem of Deep Neural Networks (DNN) [2, 24, 33]. These studies mainly experiment with image data sets, focusing on the connection between overfitting and model architectures [24, 29]. However, less attention has been paid to the overfitting phenomenon of deep neural models for recommender systems.

In this work, we study the overfitting problem of the deep click-through rate (CTR) prediction model [6, 7, 31, 32]. Despite the focus on CTR prediction, the analysis of this research can be easily generalized to other prediction tasks, such as deep conversion rate (CVR) prediction [5, 11, 20]. For industrial applications, the CTR prediction task is commonly formulated as a supervised learning problem: a CTR prediction model fits the historical user-item click interactions in the training stage and then is evaluated on new user behaviors for testing.

There are two main characteristics in CTR prediction, the **data** and the **model architecture**. Firstly, data in the recommender systems are high-dimensional and sparse. Deep models are trained on large-scale data sets with even billions of features but the vast majority of features have very low occurrences [16, 28, 30]. Secondly, considering the data characteristic of recommender systems, CTR prediction models generally follow an Embedding and MLP architecture [32], unlike the common deep architecture like CNNs in computer vision. Figure 1 illustrates the Embedding and MLP structure. The raw inputs, usually sparse category features represented by IDs, are first mapped into low-dimensional vectors by the embedding layer and then transformed into a fixed-length vector by pooling (e.g., mean pooling) for each feature field (e.g., sequence

[*] Zhao-Yu Zhang, Xiang-Rong Sheng, and Yujing Zhang contributed equally to this research. This work was done when Zhao-Yu Zhang was a research intern at Alibaba Group, and Shuguang Han is the corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557479

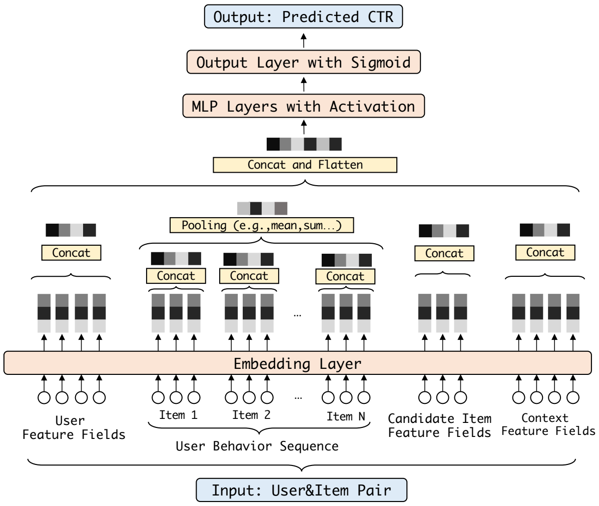


Figure 1: An illustration of Embedding and MLP structure for the deep CTR prediction model.

of history clicked item IDs), and finally concatenated together as input of the following MLP layers for final prediction.

By conducting extensive experiments on industrial recommender systems, we observe that the overfitting phenomenon of the deep CTR prediction model is peculiar. The model performance increases gradually within the first epoch while falls dramatically at the beginning of the second epoch (see Figure 2). This phenomenon will be referred to as the **one-epoch phenomenon** in the following. It differs from overfitting in tasks like computer vision, where the model is usually trained for hundreds of epochs, and the overfitting occurs gradually. The finding of the one-epoch phenomenon of CTR prediction models is also consistent with previous experiments in industrial applications [32] and academic research [34]. We believe a profound investigation of this topic will provide insights into understanding deep learning in recommender systems and drive forward the development of industrial CTR prediction models.

To this end, we conduct extensive experiments on public and production data sets to find the potential factors influencing the one-epoch phenomenon and offer insights into understanding it. The results show that the model structure, optimization algorithm with a fast convergence rate, and feature sparsity are closely related to the one-epoch phenomenon. We also discover that the one-epoch phenomenon exists widely in deep CTR prediction models. Moreover, we find training the models for more epochs does not improve over models trained in one epoch with appropriate hyper-parameters. Finally, we give a hypothesis to explain the one-epoch phenomenon and conduct proof-of-concept experiments, hoping to provide insights for the follow-up work.

The main contributions are summarized as follows:

- We conduct extensive experiments on industrial production data sets. The results show that the deep CTR prediction models exhibit the one-epoch phenomenon. Concretely, the models abruptly overfit the training data at the beginning of the second epoch, causing a severe drop in model performance.

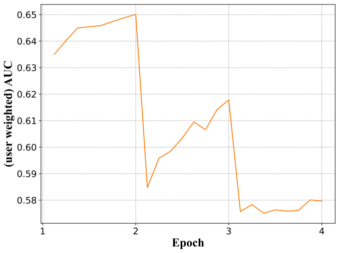


Figure 2: An illustration of testing user-weighted AUC for our industrial CTR prediction model when training for multiple epochs. The model performance falls rapidly at the beginning of the second epoch.

- We find that the model structure, optimization algorithm with a fast convergence rate, and feature sparsity are closely related to the one-epoch phenomenon. Although we can train the model for multiple epochs by restricting these factors, the best model performance is usually obtained by training only one epoch. The result may explain why most online industrial deep CTR prediction models only train the data once.
- We provide a hypothesis to explain the one-epoch phenomenon and design experiments for verification. Denote the representation after the embedding layer of sample $\boldsymbol{x}$ as $\text{EMB}(\boldsymbol{x})$. The key points are that the joint probability distribution $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}), y\right)$ is different between untrained and trained samples, and MLP layers quickly adapt to $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}), y\right)$ of trained samples at the second epoch, leading to the one-epoch phenomenon.

2 RELATED WORK

2.1 CTR Prediction

A fundamental goal in recommender systems is to recommend proper items to users, and an accurate CTR prediction model is critical for achieving this objective. In the deep learning era, many industrial CTR prediction models have made the transition from traditional shallow models [10, 18, 22] to deep models [6, 12, 21, 25, 31, 32]. As shown in Figure 1, a deep CTR prediction model generally follows an Embedding and MLP architecture. The embedding module first transforms each discrete ID from raw input into low dimensional vector. The embeddings of each feature field are then aggregated by various means (e.g., mean pooling) to obtain a fixed-length vector. The embedding vectors of different feature fields are concatenated as input into the MLP module for final prediction. Under this architecture, there are many studies on improving some of the components. For example, studies on user interest modeling [31, 32] focus on effective ways of aggregating user behaviors. Feature interaction [21] mainly focuses on the interaction of embedding vectors of different feature fields to generate high-order features and concatenate them into the input vectors of the MLP layers. The Embedding and MLP architecture has achieved state-of-the-art performance and has been deployed widely in industrial recommender systems.

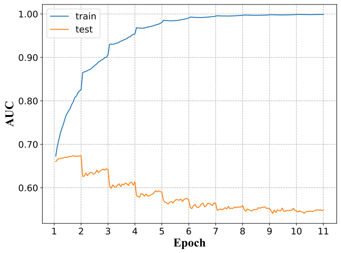


Figure 3: An illustration of the training and testing AUC for the production data set over different training epochs. Here, we can clearly observe the one-epoch phenomenon.

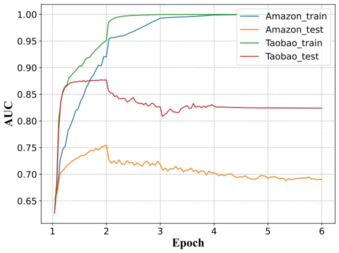


Figure 4: An illustration of the training and testing AUC for Amazon book and Taobao data sets over different training epochs. The one-epoch phenomenon occurs in both data sets.

2.2 Overfitting Phenomenon of DNN

Deep learning has achieved state-of-the-art performances in many application domains. Along with its empirical success, many researchers attempt to understand the overfitting phenomenon of DNN. Zhou [33] regards the DNN as the combination of the feature space transformation (FST) part and classifier construction (CC) part. Over-parameterization leads to overfitting in CC but not in FST. Salman and Liu [24] show that the overfitting of DNN is due to continuous gradient updating and scale sensitiveness of cross-entropy loss. In addition, there are some studies on the generalization ability of DNN, which is closely related to overfitting. Zhang et al. [29] observe that conventional generalization bounds are inadequate for over-parameterized DNN. Bartlett et al. [1] present a margin-based generalization bound for neural networks that scale with their margin-normalized spectral complexity. In conclusion, despite many exploratory studies, there is currently no widely-accepted explanation or widely-used theoretical tool, and understanding the overfitting of DNN remains an open problem.

Unlike models in other tasks, deep models in recommender systems usually follow the Embedding and MLP structure, facing high-dimensional sparse feature [16, 30]. The data involve billions of sparse features, and only part of the parameters are used in each forward pass. It brings new challenges to applying deep learning algorithms and analyzing their overfitting phenomenon. To the best of our knowledge, far less attention is paid to the overfitting phenomenon of deep models in recommender systems.

3 THE ONE-EPOCH PHENOMENON

To better illustrate the overfitting problem of deep CTR prediction models, we show the testing curve of the online CTR prediction model of Alibaba display advertising platform. The model utilizes hundreds of feature fields including raw IDs (e.g., ID or categorical attribute of an item), interaction features, and statistical features (e.g., historical stats of a user or item). As shown in Figure 2, the testing user-weighted AUC (Area Under the ROC Curve) drops dramatically at the beginning of the second epoch (and the beginning of other epochs as well). We name it the one-epoch phenomenon.

Since the industrial CTR prediction model is specifically tailored and optimized for our online applications, we further conduct a set of experiments with other commonly-adopted model architectures and a simplified production data set. We firstly examine the simplest DNN structure – a three-layer DNN with hidden units (200, 80, 1), and seven simple feature fields (as shown in Table 1) are adopted for experiments. Figure 3 plots the model performance for ten epochs. It is clear that the one-epoch phenomenon remains, even with the simplest DNN model structure and input features.

In addition to the production data set, we also examine the overfitting phenomenon on two widely-used public data sets (Amazon book[1] and Taobao[2]). Figure 4 offers an illustration of model performance over the number of training epochs, in which the one-epoch phenomenon still exists. This clearly demonstrates that the one-epoch phenomenon is a common problem, and it is not only restricted to one specific model structure or data set.

Similar observations are also made in previous studies [32, 34] on CTR prediction. Zhu et al. [34] discover that the best performance of many deep CTR prediction models on the Avazu data set is obtained by training only one epoch. Zhou et al. [32] observe that the CTR prediction model performance abruptly decreases at the beginning of the second epoch. Despite the pervasiveness of the one-epoch phenomenon for deep CTR prediction models, to the best of our knowledge, no previous studies have been devoted to understanding such a phenomenon.

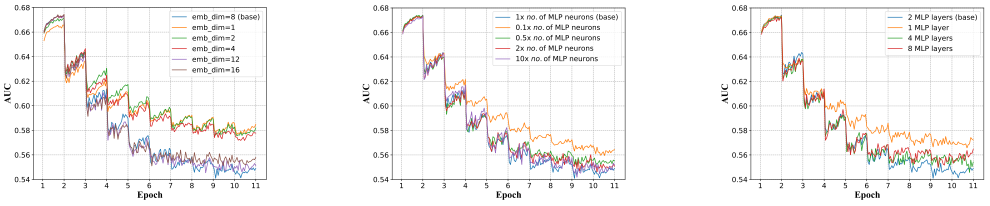
4 THE ANATOMY OF ONE-EPOCH PHENOMENON

We conduct extensive experiments on the production data set to understand the likely factors that cause the one-epoch phenomenon. The experiments are divided into two parts: model-related factors and feature-related factors. As for the experiments, all of them are conducted on our production data set. Unless otherwise specified, the hyper-parameters of the model are the default settings. Details about the experiment settings are provided in Section 6. And experimental results on public data sets are provided with the codes for reproducibility.

4.1 Model-Related Factors

In this section, we study the impact of various model-related factors, ranging from the model architecture, number of model parameters, batch size, activation function, to the choice of training optimizer and learning rate, on the one-epoch phenomenon. In addition to that, we also experiment with multiple commonly-adopted

[1] http://jmcauley.ucsd.edu/data/amazon/
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649



(a) Embedding dimension

(b) Number (“*no.*”) of MLP neurons.

(c) Number of MLP layers.

Figure 5: The amount of model parameters (including the embedding dimension, the number of hidden units, and the number of MLP layers) has no clear effect on the one-epoch phenomenon.

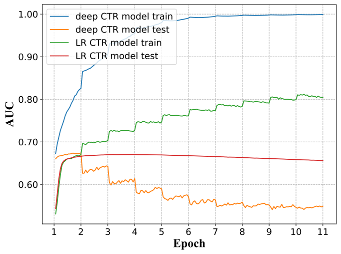
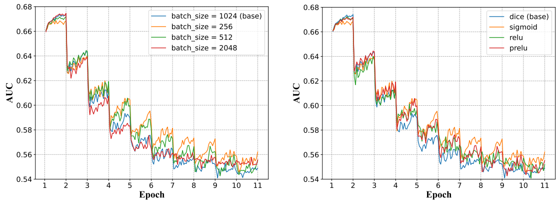


Figure 6: A comparison of the convergence curves between the DNN model and the logistic regression (LR) model, in which the one-epoch phenomenon is only observed for DNN.



(a) Batch size

(b) Activation function

Figure 7: A comparison of testing AUC for models with different batch sizes and activation functions. Neither shows a significant effect on the one-epoch phenomenon.

techniques such as weight decay [19] and dropout [26], aiming to alleviate the overfitting problem.

4.1.1 Model Structure. For CTR prediction, Figure 6 provides a comparison of the convergence curves between the DNN model and the LR model. In spite of slight overfitting, LR does not show the one-epoch phenomenon. It is worth noting that we have also experimented with a variety of parameter settings (e.g., learning rate, optimizer) for the LR model, and the model convergence curves present very similar patterns. On the contrary, the DNN-based models exhibit a clear one-epoch overfitting phenomenon. This illustrates that the one-epoch phenomenon is closely related to the adopted deep neural model architecture that consists of the utilization of embedding features and the corresponding MLP structure.

4.1.2 The Amount of Model Parameters. In addition to the default setups for embedding dimension (8 in the above experiments) and hidden units of the MLP layer, we conduct an extensive amount of experiments by varying model parameters. In this section, we analyze the effect of model parameters on the one-epoch phenomenon.

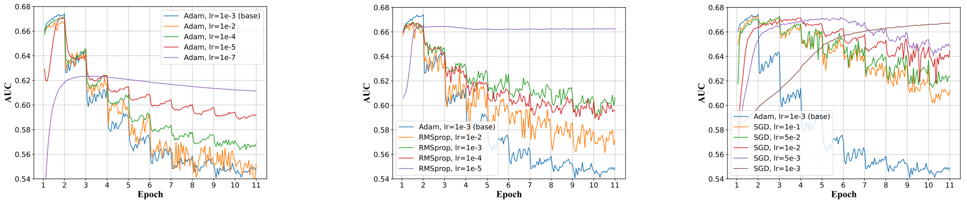
Figure 5(a) plots the model convergence curves over a variety of embedding dimension sizes, Figure 5(b) illustrates the effect of model convergence with different number of hidden units, and Figure 5(c) compares the model performance for different number of MLP layers. It is clear that with the adopted deep CTR prediction model, different setups for model parameters do not mitigate such a one-epoch phenomenon.

For a more special case where the embedding dimension size is set to 1, meaning that every feature is only represented by one scalar value, the total amount of parameters is roughly the same as the LR model. We observe that, even in this case, such a DNN-based model still suffers from the one-epoch phenomenon. Thereby, we believe that **the Embedding and MLP model structure rather than the amount of model parameters is a more related factor to the one-epoch phenomenon**.

4.1.3 Activation Function. We study the effect of different activation functions on the one-epoch phenomenon. In addition to the dice unit [32], we also employ sigmoid, relu [8] and prelu [13]. Figure 7(b) provides the model convergence curves for different activation functions, in which we find that the activation function almost has no influence on such a phenomenon.

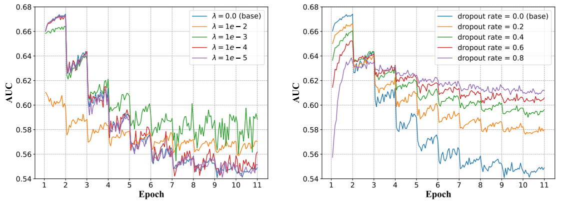
4.1.4 Batch Size. We also analyze the effect of different batch sizes in Figure 7(a). Same to the activation function, changing batch sizes does not help alleviate the one-epoch problem.

4.1.5 Optimization Algorithm. In this section, we focus on how optimization algorithms (including training optimizer and learning rate) affect the one-epoch phenomenon. In addition to the widely-adopted Adam optimizer [17], we further take into account RMSprop [15] and Stochastic Gradient Descent (SGD). Figure 8 presents the model convergence curves of each optimizer over different training epochs. Compared to SGD, Adam and RMSprop show faster convergence rates in most of the cases, while they are more prone to the one-epoch phenomenon. We further observe that the learning rate may also be connected with the one-epoch problem. With an extremely small learning rate, such a phenomenon is less obvious, but it is at the expense of model performance. In summary, **an**



(a) Adam (b) RMSprop (c) SGD

Figure 8: A comparison of testing AUC for models with different optimizers. "lr" is short for learning rate.



(a) Weight decay. λ is the weight decay factor defined in Equation 1

(b) Dropout

Figure 9: A comparison of testing AUC for models with different weight decay and dropout settings. They do not show any sign of mitigating the one-epoch phenomenon.

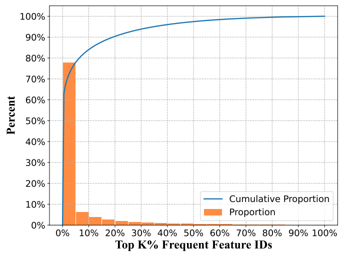


Figure 10: The cumulative proportion of IDs on the production data set. We sort all IDs in descending order by occurrence frequency.

optimization algorithm that facilitates a faster model convergence could be at the risk of the one-epoch problem.

4.1.6 Weight Decay and Dropout. Weight decay [19] is a widely-adopted technique to restrict model complexity and thereby alleviate overfitting. At each iteration of the model training process, the weights of a neural network $\boldsymbol{\theta}$ are commonly updated with the computed gradient $\nabla f(\boldsymbol{\theta})$, whereas the weight decay technique further shrinks $\boldsymbol{\theta}$ with the below formula:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \nabla f(\boldsymbol{\theta}_t) - \lambda \boldsymbol{\theta}_t, \tag{1}$$

where α stands for the learning rate and λ denotes the weight decay factor. Figure 9(a) plots the convergence curves over different choices of weight decay factors. We find that the weight decay algorithm neither improves the model performance nor helps alleviate the one-epoch problem.

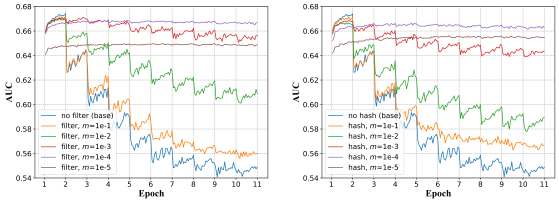
Dropout [26] is another common technique to relieve the overfitting problem. The key idea is to randomly drop units (along with their connections) from the neural network during training, which prevents the units from co-adapting too much. In our experiment, we adopt one dropout layer before each fully connected layer, so there are 3 dropout layers in total. Figure 9(b) presents the model performance after applying the dropout layers, with each curve indicating one dropout rate. Again, we observe that dropout does not help solve the one-epoch problem.

4.2 Feature-Related Factors

The feature set used in industrial CTR prediction models can be roughly categorized into four types: user feature fields (e.g., age and gender), user behavior sequences (e.g., a sequence of clicked items), candidate item feature fields (e.g., item ID and category ID), and contextual feature fields. Table 1 gives an example of the production data set. Note that all features are preprocessed into discrete features (continuous features are discretized by buckets) and each feature value is represented by an ID, which is a common practice in industrial scenarios.

Here, we illustrate the feature sparsity in the production data set. A sparser feature field corresponds to a larger number of unique IDs and a smaller average occurrence per ID. The production data set contains seven feature fields, and Table 1 gives the number of unique IDs and the mean occurrences of each ID for each feature. The fine-grained features (e.g., item ID and history item IDs) are much sparser than the others. Besides, IDs exhibit a long-tailed distribution. For the whole data set, we sort all feature IDs in descending order according to the occurrence frequency and plot the distribution in Figure 10. We observe that the bottom 50% frequent IDs only account for the 2.5% occurrences.

To show that the feature sparsity is related to the one-epoch phenomenon, we reduce the feature sparsity and illustrate the generalization performance. We use two techniques, **filter** and **hash**, to reduce the feature sparsity. Given a ratio m, the filter method only reserves the top m ratio frequent IDs and filters out the other IDs (replacing them with a default ID). As for the hash method, it maps each ID to a space whose size is the m ratio of the number of all IDs. Note that the filter and hash are applied to the whole data set rather than a specific feature field. After filter or hash, the feature sparsity is alleviated. Figure 11 gives the results, which show that as m gradually reduces, the one-epoch phenomenon is alleviated accordingly. However, at the same time, performance degradation is inevitable.



(a) Filter (b) Hash

Figure 11: Test AUCs of models trained on the production data set with filter and hash, respectively. m is the ratio of the number of IDs after and before compression. The smaller m, the lower the feature sparsity. The one-epoch phenomenon is related to large feature sparsity.

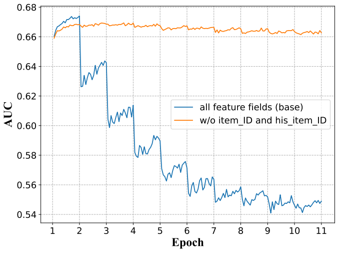


Figure 12: If sparse feature fields (item ID and history item IDs) are not used for training, the model will not experience the one-epoch phenomenon.

Another straightforward method to reduce sparsity is to exclude the fine-grained feature fields. In the production data set, we find that when the model does not use item ID and history item IDs, the one-epoch phenomenon does not occur. However, it leads to worse performance than a model that is trained with all features. The result is demonstrated in Figure 12.

From the experiment results, it can be concluded that **the feature sparsity is closely related to the one-epoch phenomenon**. Although the one-epoch phenomenon can be alleviated or even eliminated by reducing feature sparsity, it often leads to inferior model performances. The above experiments also explain why models trained with only coarse-grained features do not encounter the one-epoch phenomenon.

4.3 Summary and Discussion

Through extensive experiments, we find that the **model structure**, **optimization algorithm with a fast convergence rate** (e.g., Adam optimizer with a large learning rate), and **large feature sparsity** (e.g., using fine-grained features like item IDs) are closely related to the one-epoch phenomenon. We also verify that some factors have no obvious effect on the one-epoch phenomenon, including the number of model parameters, activation function, batch size, weight decay, and dropout.

It’s worth mentioning that although we can alleviate the one-epoch phenomenon by changing some factors, it also brings a more or less performance degradation. We find the best performance is achieved by training only one epoch. Most industrial recommender systems only train each sample once and our experiments may provide a reasonable explanation for this practice. Given the experiment result, a natural question is whether we can design a new training paradigm that surpasses the model trained with one epoch. As an exploratory, we have tried methods like fine-tuning part of the parameters and learning rate decay in the second epoch. However, these methods do not exhibit significant performance gain. We hope that these experiments can shed light on future research on training more than one epoch with better performance.

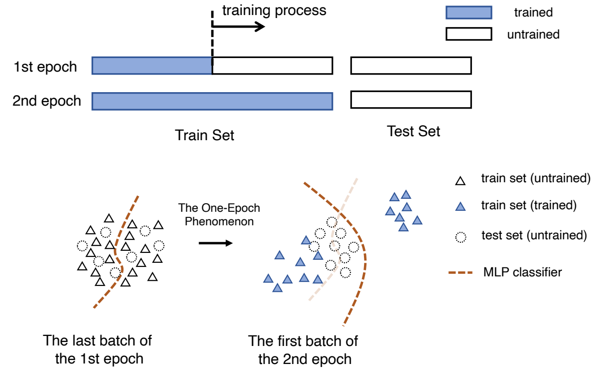


Figure 13: An illustration of the hypothesis. Each EMB(x) is represented by a triangle or a circle. At the beginning of the second epoch, MLP layers quickly fit $\mathcal{D}(\mathrm{EMB}(x_{\text{trained}}), y)$.

5 HYPOTHESIS

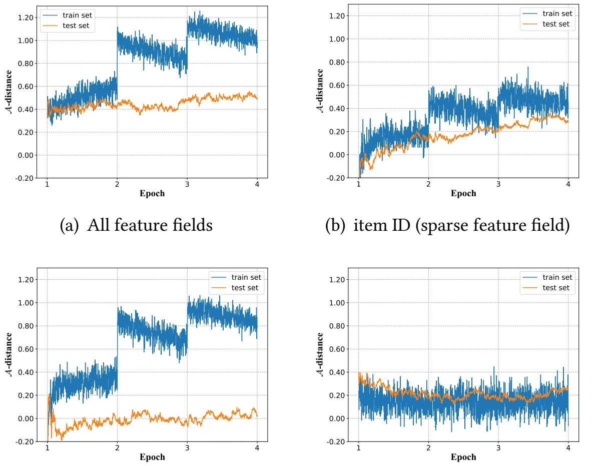
In this section, we give a hypothesis to explain the one-epoch phenomenon. Let EMB($\boldsymbol{x}$) denote the intermediate representation of a sample $\boldsymbol{x}$ after embedding layer. The MLP layers are trained on the joint probability distribution $\mathcal{D}$ (EMB($\boldsymbol{x}$), y). Denote a trained sample of the model as $\boldsymbol{x}_{\text{trained}}$ and an untrained sample as $\boldsymbol{x}_{\text{untrained}}$. For example, a sample of the train set is $\boldsymbol{x}_{\text{untrained}}$ in the first epoch and $\boldsymbol{x}_{\text{trained}}$ in the second epoch, and a sample of the test set is $\boldsymbol{x}_{\text{untrained}}$. We hypothesize that $\mathcal{D}$ (EMB($\boldsymbol{x}_{\text{untrained}}$), y) is significantly different from $\mathcal{D}$ (EMB($\boldsymbol{x}_{\text{trained}}$), y). At the beginning of the second epoch, MLP layers quickly adapt to the empirical distribution $\mathcal{D}$ (EMB($\boldsymbol{x}_{\text{trained}}$), y), and the overfitting occurs suddenly, causing the one-epoch phenomenon. Figure 13 gives an illustration of our hypothesis. In the following, we design a series of proof-of-concept experiments to verify this hypothesis.

5.1 Difference of the Joint Distribution

We show that $\mathcal{D}$(EMB($\boldsymbol{x}_{\text{untrained}}$), y) is significantly different from $\mathcal{D}$(EMB($\boldsymbol{x}_{\text{trained}}$), y) . However, it is difficult to straightforwardly calculate the change of $\mathcal{D}$ (EMB($\boldsymbol{x}$), y). To obtain the quantitative variation, we propose to utilize the separability characteristic of the embedding vectors. In the CTR prediction task, the separability characteristic describes the difficulty to distinguish unclicked and clicked samples. Specifically, we represent the separability of unclicked and clicked samples by $\mathcal{A}$-distance[3] between $\mathcal{D}$ (EMB($\boldsymbol{x}$), $y = 0$) and $\mathcal{D}$ (EMB($\boldsymbol{x}$), $y = 1$). To compute the $\mathcal{A}$-distance, we need to train a binary classifier h to distinguish

Table 1: For each feature field of the production data set, below are the number of unique IDs and average number of occurrences of each ID. It's obvious that item_ID and his_item_ID are the two most sparse feature fields.

type	user feature fields		user behavior sequences		candidate item feature fields		context feature field	
feature field	user_age	user_gender	his_item_ID	his_item_cate	item_ID	item_cate	scene_ID	**all**
unique IDs	10	3	21,925,711	17,727	880,613	8,223	19	**22,106,604**
mean occurrences	1,007,307	3,357,690	23	28,411	11	1,224	530,161	**50**



(a) All feature fields

(b) item ID (sparse feature field)

(c) History item IDs (sparse feature field)

(d) History item categories (non-sparse feature field)

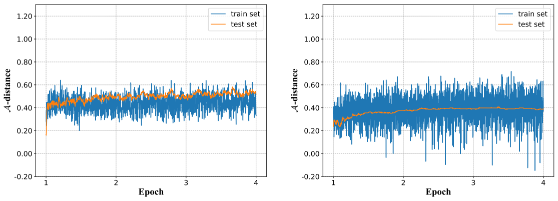
Figure 14: $\mathcal{A}$-distance between $\mathcal{D}(\mathbf{EMB}(x), y = 1)$ and $\mathcal{D}(\mathbf{EMB}(x), y = 0)$ in the train set and test set. We use all feature fields or a single feature field to calculate $\mathcal{A}$-distance.

which domain a sample comes from. Let $\text{err}(h)$ represent the loss of the classifier. Denote the $\mathcal{A}$-distance between $\mathcal{D}$ $(\text{EMB}(\boldsymbol{x}), y = 0)$ and $\mathcal{D}$ $(\text{EMB}(\boldsymbol{x}), y = 1)$ as $\mathcal{A}(\mathcal{D}(+, -))$, which is calculated as:

$$\begin{aligned}\mathcal{A}(\mathcal{D}(+,-)) &= \mathcal{A}\left(\mathcal{D}\left(\text{EMB}(\boldsymbol{x}), y=0\right), \mathcal{D}\left(\text{EMB}(\boldsymbol{x}), y=1\right)\right) \\ &= 2(1 - 2\,\text{err}(h)).\end{aligned} \tag{2}$$

Note that a larger $\mathcal{A}$-distance means a greater embedding distribution difference between the clicked and unclicked samples, so it is easier to be separated by the MLP classifier. We can easily measure the change of joint distribution $\mathcal{D}(\text{EMB}(\boldsymbol{x}), y)$ via the change of $\mathcal{A}(\mathcal{D}(+, -))$.

During the training process, we calculate $\mathcal{A}(\mathcal{D}(+, -))$ on train set and test set, respectively. And LR model is used as the binary classifier for $\mathcal{A}$-distance. The results are shown in Figure 14(a). For the train set, a sample is untrained ($\boldsymbol{x}_{\text{untrained}}$) in the first epoch, while it is trained ($\boldsymbol{x}_{\text{trained}}$) in the second epoch. It can be found that $\mathcal{A}(\mathcal{D}(+, -))$ suddenly increases at the beginning of the second epoch, which verifies that $\mathcal{D}$ $(\text{EMB}(\boldsymbol{x}_{\text{trained}}), y)$ is different from $\mathcal{D}$ $(\text{EMB}(\boldsymbol{x}_{\text{untrained}}), y)$ and $\mathcal{D}$ $(\text{EMB}(\boldsymbol{x}_{\text{trained}}), y)$ is much easier to fit. For the test set, all samples are untrained from beginning to the end and the $\mathcal{A}(\mathcal{D}(+, -))$ is stable, which shows that $\mathcal{D}(\text{EMB}(X_{\text{untrained}}), y)$ has no mutation during the training process.



(a) Filter $m = 1e\text{-}4$

(b) Learning rate $1e\text{-}7$

Figure 15: When the one-epoch phenomenon does not occur, $\mathcal{A}$-distance between $\mathcal{D}(\mathbf{EMB}(x), y = 1)$ and $\mathcal{D}(\mathbf{EMB}(x), y = 0)$ in the train set and test set will not suddenly increase.

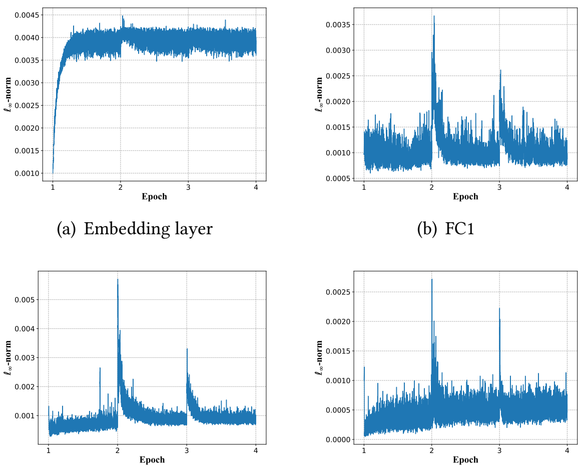
Further, we try to analyze the relationship between the variation of $\mathcal{D}(\text{EMB}(\boldsymbol{x}), y)$ and the feature sparsity. We follow the above experimental framework but use the embedding vector of each feature field instead of all feature fields to calculate $\mathcal{A}(\mathcal{D}(+, -))$, and other settings of the experiment remain unchanged. We find that the fine-grained feature fields (item ID and history item IDs) have a sudden increase in $\mathcal{A}(\mathcal{D}(+, -))$, while the other features fields (such as age, gender, and history item categories) are not. We show the results of item ID, history item IDs, and history item categories in Figures 14(b)-14(d). This experiment reveals that the difference between $\mathcal{D}(\text{EMB}(\boldsymbol{x}_{\text{trained}}, y)$ and $\mathcal{D}(\text{EMB}(\boldsymbol{x}_{\text{untrained}}, y)$ is mainly dominated by the sparse feature fields.

Finally, we also analyze the change of $\mathcal{D}(\text{EMB}(\boldsymbol{x}), y)$ when the one-epoch phenomenon does not happen. We use learning rate $1e\text{-}7$ and filter $m = 1e\text{-}4$, respectively. Note that the one-epoch phenomenon does not occur (as illustrated in Figure 8(a) and Figure 11(a)) in these settings. The result is shown in Figure 15. The curves do not have sudden change between epochs, which indicates that difference between $\mathcal{D}(\text{EMB}(\boldsymbol{x}_{\text{trained}}), y)$ and $\mathcal{D}(\text{EMB}(\boldsymbol{x}_{\text{untrained}}), y)$ could be the necessary condition of the one-epoch phenomenon.

5.2 Fast Adaptation to the Trained Samples

We show that the MLP layers quickly adapt to $\mathcal{D}(\text{EMB}(\boldsymbol{x}_{\text{trained}}), y)$ at the second epoch, in which parameters of the MLP layers change suddenly. Particularly, we monitor the parameter changes (i.e., the update values of parameters calculated by the optimizer) of each training step for embedding layer and MLP layers (including the output layer) during the training process. We adopt ℓ_∞-norm and the results are shown in Figure 16. We find that the parameter changes of the embedding layer are generally stable, while the variation of the MLP layers suddenly increases at the second epoch.

We conduct another experiment to verify the hypothesis that the rapid change of the MLP layers in the second epoch relates to the



(a) Embedding layer

(b) FC1

(c) FC2

(d) FC3 (output layer)

Figure 16: Parameter changes of the Embedding and MLP layers during training. The latter suddenly increases at the beginning of the second epoch, while the former does not. "FC" is short for "fully connected".

one-epoch phenomenon. In detail, we fine-tune part of the model parameters after the end of the first epoch, and freeze the others. The results of fine-tuning all parameters, embedding layers, and MLP layers are shown in Figure 17. We observe that only fine-tuning the MLP layers leads to the one phenomenon. Only fine-tuning the embedding layer, i.e., freezing the MLP layers after the first epoch, alleviates the one-epoch phenomenon. The result validates the hypothesis that fast adaptation to the trained samples of the MLP layers causes the one-epoch phenomenon.

5.3 Summary

According to our hypothesis and the verification experiments, $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}_{\text{trained}}), y\right)$ is different from $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}_{\text{untrained}}), y\right)$. The MLP layers quickly adapt to the empirical distribution of trained samples $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}_{\text{trained}}), y\right)$ at the beginning of the second epoch. Thus, it leads to the two characteristics of the one-epoch phenomenon: (a) it happens exactly at the beginning of the second epoch, and (b) the test performance exhibit a sharp decrease.

We provide validation experiments to show that the joint distribution $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}), y\right)$ suddenly changes at the beginning of the second epoch via the $\mathcal{A}$-distance metric. And we find the sudden change is mainly caused by fine-grained feature fields. Furthermore, we conduct experiments to show that the parameter update values of MLP layers suddenly increase at the beginning of the second epoch which strongly supports our hypothesis about the MLP layers' fast adaption to the trained samples. And we find that the one-epoch phenomenon can be alleviated via freezing the MLP layers at the second epoch.

From the experiment results, we can extend our hypothesis: the distribution difference not only exists between the trained and untrained samples but also exists between samples trained at different times. As a result, the test AUC drops rapidly at the beginning of each epoch (from the second epoch), which could be called the "**each-epoch phenomenon**". In fact, the experiments in this section (see Figures 14, 16, and 17) have verified this more generalized hypothesis. This research mainly focus on the one-epoch phenomenon, and investigate the difference between $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}_{\text{trained}}), y\right)$ and $\mathcal{D}\left(\text{EMB}(\boldsymbol{x}_{\text{untrained}}), y\right)$. We would also like to explore the each-epoch phenomenon in the future.

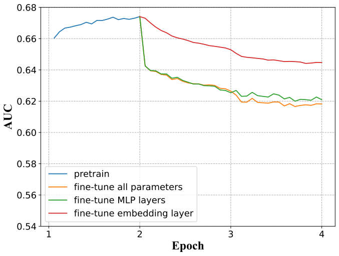


Figure 17: After one epoch of pretraining with default settings, fine-tune part of the model parameters and freeze the others. This figure shows the corresponding test AUCs.

6 EXPERIMENT DETAILS

This section describes the details of the data sets and settings. For reproduction, the codes and the results on public data sets are provided at https://github.com/Z-Y-Zhang/one_epoch_phenomenon.

6.1 Data Sets

We use one production data set collected from online recommender systems and two widely-used public data sets.

Production data set is collected from Alibaba display advertisement platform. We randomly select about 1% of 1-day samples for training and 0.1% of the next-day samples for testing. 7 feature fields are selected: user age, user gender, history (clicked) item IDs, history (clicked) item categories, item ID, item category, and scene ID. There are 10,073,072 samples and 22,106,604 different IDs in total. Table 1 gives the number of unique IDs and the mean occurrences of each ID for each feature field.

Amazon book data set contains book reviews and metadata from Amazon. Following previous work [4, 31], we regard reviews as positive samples and randomly select products not rated by a specific user as negative samples, so as to generate the CTR prediction data set. This data set contains 150,016 samples and 425,970 unique IDs in total. The feature fields contain user ID, history (clicked) item IDs, history (clicked) item categories, item ID, and item category. Table 2 gives the number of unique IDs and the mean occurrences of each ID for each feature field.

Taobao data set is a collection of user behaviors (including click, purchase, adding item to shopping cart, and item favoring) from Taobao's recommender system. Following [4], we use clicked behaviors for each user to generate a CTR data set. The feature fields contain user ID, history (clicked) item IDs, history (clicked) item categories, item ID, and item category. Because each user has only one sample in this data set, the user ID is useless for training and we exclude this feature field. This data set has 987,648 samples

Table 2: Number of unique IDs and average number of occurrences of each ID on Amazon book and Taobao data sets. User_ID, item_ID and his_item_ID are the three most sparse features.

feature field		user_ID	his_item_ID	his_item_cate	item_ID	item_cate	**all**
Amazon book	unique IDs	75,008	347,016	1,573	85,473	959	**425,970**
	mean occurrences	2.00	19.35	4,269.04	1.75	156.43	**32.76**
Taobao	unique IDs	\	4,039,879	9,411	725,540	7,849	**4,049,291**
	mean occurrences	\	22.24	9,548.70	1.36	125.8	**44.99**

and 4,049,291 unique IDs. The number of unique IDs and the mean occurrences of each ID for each feature field are in Table 2.

6.2 Settings

Our base model contains an embedding layer and 3 fully connected layers (200x80x1), that is, 2 MLP layers (200x80) with dice [32] activation and 1 output layer with sigmoid activation. The embedding dimension of each feature field is 8. The batch size is 1024 for the production data set, 128 for Amazon book data set, and 512 for Taobao data set. We use mean pooling to aggregate the user history behavior embedding sequence. The model is optimized by Adam [17] with learning rate $1e$-3 to minimize the binary cross-entropy loss. In each experiment, parameters different from the default settings have been described in the context of this paper.

7 CONCLUSION AND DISCUSSION

In this paper, we discover that the commonly-adopted deep CTR prediction models exhibit the one-epoch phenomenon: at the beginning of the second epoch, the model performance degrades dramatically, which is a clear sign of overfitting. Such a phenomenon has been witnessed widely in real-world CTR prediction models. Through extensive experiments, we observed that the model structure, optimization algorithm with a fast convergence rate, and feature sparsity are closely related to the one-epoch phenomenon. This explains why online industrial deep CTR prediction models only train the data once. To obtain a better understanding of the one-epoch phenomenon, we propose a likely hypothesis and further validate this premise with a set of experiments.

Although this research focuses on click-through rate prediction, the above analysis can be easily generalized to other prediction tasks like conversion rate (CVR) prediction. To this end, our findings in this research are of general interest to researchers and practitioners in recommendation systems. We hope that the investigation of the one-epoch phenomenon can shed light on future research on training more epochs for better performance.

ACKNOWLEDGEMENTS

This work is supported by Alibaba Group through Alibaba Innovation Research Program. We would like to thank Guorui Zhou, Xiaoqiang Zhu, Weijie Bian, Zhangming Chan and other colleagues for the helpful discussions.

REFERENCES

[1] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. 2017. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30.* 6240–6249.

[2] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* 116, 32 (2019), 15849–15854.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems* 19 (2007), 137.

[4] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, Xinchen Luo, Shiming Xiang, Guorui Zhou, Xiaoqiang Zhu, and Hongbo Deng. 2022. CAN: Feature Co-Action Network for Click-Through Rate Prediction. In *The 15th ACM International Conference on Web Search and Data Mining.* 57–65.

[5] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1097–1105.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.* ACM, 7–10.

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

[8] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.* IEEE, 8609–8613.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics.* Minneapolis, MN, USA, 4171–4186.

[10] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[11] Siyu Gu, Xiang-Rong Sheng, Ying Fan, Guorui Zhou, and Xiaoqiang Zhu. 2021. Real Negatives Matter: Continuous Training with Real Negatives for Delayed Feedback Modeling. In *Proceedings of The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 2890–2898.

[12] 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 26th International Joint Conference on Artificial Intelligence.* Melbourne, Australia., 2782–2788.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision.* 1026–1034.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition.* Las Vegas, NV, 770–778.

[15] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on* 14, 8 (2012), 2.

[16] Biye Jiang, Chao Deng, Huimin Yi, Zelin Hu, Guorui Zhou, Yang Zheng, Sui Huang, Xinyang Guo, Dongyue Wang, Yue Song, et al. 2019. XDL: An Industrial Deep Learning Framework for High-Dimensional Sparse Data. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data.* 1–9.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations.*

[18] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[19] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[20] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.* 1137–1140.

[21] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2019. Product-Based Neural Networks for User Response Prediction over Multi-Field Categorical Data. *ACM Transactions on Information Systems* 37, 1 (2019), 5:1–5:35.

[22] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 10th International Conference on Data Mining*. IEEE, 995–1000.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[24] Shaeke Salman and Xiuwen Liu. 2019. Overfitting mechanism and avoidance in deep neural networks. *arXiv preprint arXiv:1901.06566* (2019).

[25] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, and Xiaoqiang Zhu. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *Proceedings of The 30th ACM International Conference on Information and Knowledge Management.* 4104–4113.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*. 5998–6008.

[28] Minhui Xie, Kai Ren, Youyou Lu, Guangxu Yang, Qingxing Xu, Bihai Wu, Jiazhen Lin, Hongbo Ao, Wanhong Xu, and Jiwu Shu. 2020. Kraken: memory-efficient continual learning for large-scale real-time recommendations. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–17.

[29] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communication of the ACM* 64, 3 (2021), 107–115.

[30] Yuanxing Zhang, Langshi Chen, Siran Yang, Man Yuan, Huimin Yi, et al. 2022. PICASSO: Unleashing the Potential of GPU-centric Training for Wide-and-deep Recommender Systems. In *2022 IEEE 38th International Conference on Data Engineering (ICDE).* IEEE.

[31] 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 33rd AAAI Conference on Artificial Intelligence.* Honolulu, Hawaii, USA, 5941–5948.

[32] Guorui Zhou, Xiaoqiang Zhu, Chenru 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.* ACM, 1059–1068.

[33] Zhi-Hua Zhou. 2021. Why over-parameterization of deep neural networks does not overfit? *Science China Information Sciences* 64, 1 (2021).

[34] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open Benchmarking for Click-Through Rate Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 2759–2769.