Neuron-Connection Sharing for Multi-Task Learning in Video Conversion Rate Prediction

Abstract. As a fundamental task of industrial ranking systems, conversion rate (CVR) prediction is suffering from data sparsity problems. Most conventional CVR modeling leverages Click-through rate (CTR)&CVR multitask learning because CTR involves far more samples than CVR. However, typical coarse-grained layer-level sharing methods may introduce conflicts and lead to performance degradation, since not every neuron or neuron connection in one layer should be shared between CVR and CTR tasks. This is because users may have different fine-grained content feature preferences between deep consumption and click behaviors, represented by CVR and CTR, respectively. To address this sharing&conflict problem, we propose a neuron-connection level knowledge sharing. We start with an over-parameterized base network from which CVR and CTR extract their own subnetworks. The subnetworks are partially overlapped which corresponds to the sharing part, and the task-specific part can alleviate the conflict problem. The layer-level methods can be formulated as particular instances of this framework, which can automatically learn which neuron weights are shared or not without manual experience. As far as we know, this is the first time that a neuron-connection level sharing is proposed in CVR modeling. Both offline and online experiments on one of the biggest video recommendation platforms with nearly one billion users demonstrate the superiority of our work. The codes will be available on https://github.com/*/NeuronSharing4CVR.

Keywords: conversion rate · multi-task learning · connection sharing.

1 Introduction

Post-click conversion rate (CVR) prediction plays a key role across industrial ranking systems, such as recommender systems (RS) [5, 10, 8], online advertising [14], and search engines [11]. In video recommendation, an RS first calls up a large number of user-related videos, then ranks and presents them to users based on several metrics, such as click-through rate (CTR) or CVR. For net media contents, the CVR indicator is continuous, $label \in [0, 1]$, e.g., if someone watched 2 minutes of a video with a length of 5 minutes then $CVR = 0.4$. The number of CVR samples equals the number of content clicks, that is, the positive samples of the CTR task. The main challenge of CVR modeling is the data sparsity involving much fewer samples than CTR, which imposes a challenging model fitting process resulting in a poor generalization ability.

A feasible solution is to leverage the CTR task to share knowledge with the CVR task via multi-task learning (MTL) [5, 10, 13, 4, 3, 8]. ESMM series [5, 10]

utilize more CVR-related positive samples such as click (CTR), like, collection, and share the feature-embedding layer and output labels. MMOE series (MMOE, SNR, PLE [4, 13, 3, 8]) utilize a multi-gate mixture of expert subnetworks to provide knowledge sharing at the granularity of subnetworks. Each subnetwork will learn a different representation and then a task will use a weighted fusion of the representations to learn its final representation. As we have seen, existing literature provides sharing mechanisms based on subnetworks or neural layers, and we collectively refer to these models as layer-level CVR sharing methods.

Despite their success, the layer-sharing method often introduces representation conflict, especially when the tasks are loosely related [12, 8]. In other words, not every neuron connection in a layer should be shared across both CVR and CTR. For example, users may have random clicks or clicks for exaggerated titles or unreal cover images, and these behaviors will be learned by the CTR task through specific neuron-connection weights in the CVR&CTR multi-task model. However, this learned representation shouldn't be shared with CVR because the CVR score won't be high in this situation, and vice versa. In the real-world video platform, the CVR label is continuous and the CTR label is discrete, which leads to a more complex correlation between CVR and CTR.

To address the above **sharing&conflict** issue, we propose a novel neuron-connection sharing mechanism in multi-task CVR modeling, which can automatically learn which neuron connections are to be shared without manual experience. We start with an over-parameterized deep learning model[10], called the base network. CVR and CTR extract their subnetworks containing partial neuron connections from the base network. The two subnetworks are partially overlapped and trained in parallel. In particular, we automatically find the best subnetworks for each task in the base network by iterative magnitude pruning [2], i.e., we use a network pruning method to do network architecture search for CVR&CTR's suitable sharing structure. In this way, CVR and CTR can have their specific networks while also sharing certain node connections, in which the overlapping part and the specific part correspond to sharing representation and conflict reduction, respectively (Fig.2). Because the connection-sharing method provides a finer-grained sharing mechanism, the existing works can be formulated as particular instances of our approach, and we don't need to design sharing structures relying on manual experience.

Overall, Our main contributions can be summarized as: (1) Compared with existing layer-level sharing methods, the proposed approach dramatically expands the capacity to alleviate the sharing&conflict problem through fine-grained connection sharing in CVR multi-task modeling. (2) To the best of our knowledge, this is the first work applying the neuron-connection level sharing to solve the CVR multi-task learning. Besides it being a more generalized approach to layer sharing, it also provides an automatic design of sharing parts instead of manual designs. (3) We highlight the effectiveness of our approach against competitors and deploy our methods in an online video recommender system with nearly one billion users serving main traffic, confirming its value in real-world industrial applications.

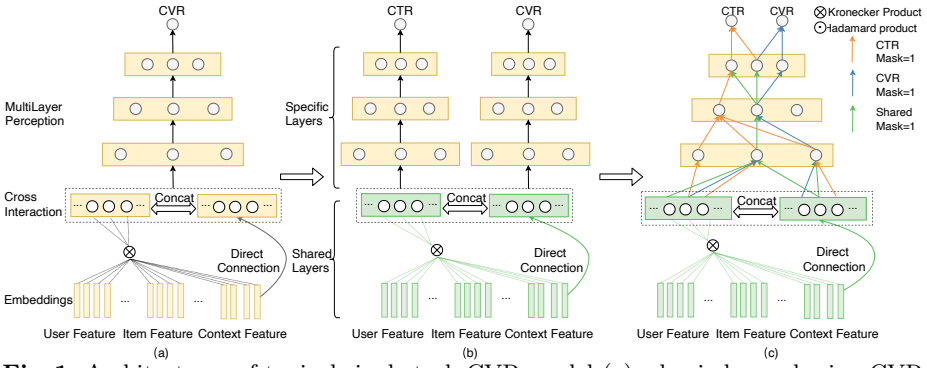

Fig. 1. Architectures of typical single-task CVR model (a), classic layer sharing CVR model (b), and our proposed neuron-connection sharing CVR model (c).

2 METHODOLOGY

In this section, we first present classic implementations of single-task CVR and layer-sharing CVR models. Then, we detail the neuron-connection sharing approach, including representation sharing, conflict reduction, and the effect comparison with existing works.

2.1 Single Task CVR & Layer Sharing CVR

Fig.1(a) presents a typical single task CVR model architecture [6]. The first layer is an embedding layer for input features. The second layer is a feature-cross layer that initially involves a Kronecker product of the embedding layer, and then concatenates the original embedding layer to form the final input features. The following comprises a sequence of fully connected (FC) layers, while the output is obtained via a sigmoid operation. For this paper, this architecture represents the single-task CVR baseline.

Fig.1(b) illustrates a classic implementation of layer-sharing CVR architecture based on the aforementioned single-task model. CVR and CTR share the embedding layer. Basic feature representations (user id, gender, item tag, etc.) learned from the CTR task can be utilized to compute the CVR score more accurately through feed-forward propagation within the network. The remaining layers are not shared because previous work verified that sharing certain layers can achieve better performance, but according to experiments, other layers cannot [5, 4]. As mentioned above, ESMM series and MMOE series algorithms are different implementations of the layer-level sharing method. For simplicity of verifying the sharing granularity's effect, we choose this most commonly used architecture in practice as the multi-task CVR baseline for this paper.

2.2 Neuron-Connection Sharing CVR

Unlike the layer-level sharing models that specify which layers to share, we design mask networks for CVR and CTR, respectively, allowing tasks to flexibly learn which neural connections should be activated either for both tasks or solely for a specific task. The proposed modeling is illustrated in Fig.2 and Algorithm 1.

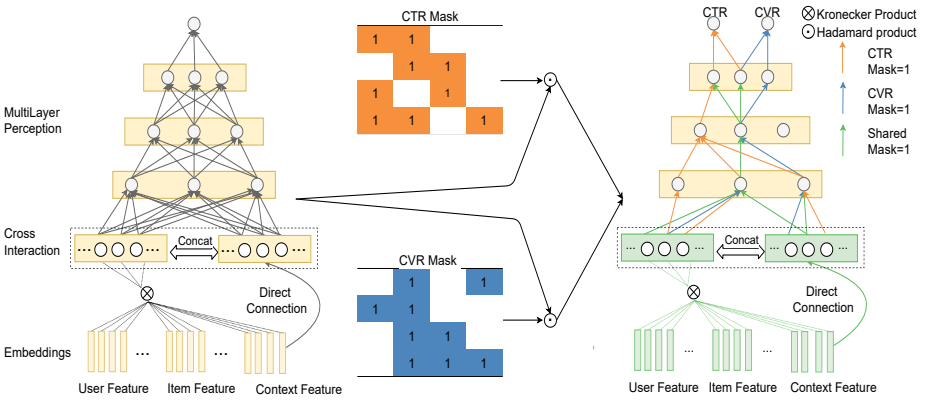

Fig. 2. Neuron-connection sharing CVR modeling. The final model (right) is generated through the Hadamard product between the base network (left) and the masks (middle). CTR mask=1 means used by CTR only, similar to CVR mask=1; shared mask=1 means shared by CVR and CTR. The bottom feature embeddings are fully shared.

Representation Sharing. In MTL, Closely related tasks tend to extract similar sub-nets so they can use similar parts of weights, while loosely related or unrelated tasks tend to extract sub-nets that are different in a wide range [1, 7]. Particularly, the inductive bias customized to the task is embedded into the subnet structure to some extent. In practice (Algorithm 1), we first use mixed samples of both CVR and CTR to train a big over-parameterized mixed network as the base shared network. Then, each task uses its own sample executing the "mask generation" to mask the neuron connections(weights) unimportant for the task itself to construct a mask. This process aims to identify the overlapping parts of these two masks, which represent the same inductive bias held by the two tasks, corresponding to knowledge that can be shared.

Conflict Reduction. After we get the fully shared base network $sNET$ and each task's own matrix $mask \in \{0, 1\}$ (Fig.2), the sub-network of the current task is obtained through $sNET \odot mask$, where $\odot$ denotes the Hadamard product. The exclusive part of a subnetwork represents each task's own specific inductive bias which can reduce the conflict while maximizing the represent sharing.

Effect Comparison. MMoE and ESMM series adopt manual experience or hyper-parameter search to decide the sharing layers, which is hard to guarantee the best benefits. Since the masks/subnetworks are learned automatically in the connection-sharing modeling, we avoid manually deciding which part of the network should be handled with representation sharing and which part with conflict-solving techniques as in previous works. In fact, layer-level sharing can be formulated into the framework of the connection-sharing model. If the values of some specific layers of both $masks[CTR]$ and $masks[CVR]$ are set as 1, the proposed method degenerates into layer-level sharing. In this way, connection-level sharing provides a more automatic and generalized sharing mechanism compared to existing works. The training details are illustrated in Algorithm 1.

Table 1. Statistics of the Experimental Dataset

Dataset	#User	#Video	#Impression	#Click	#Conversion
Tencent Video	10M	11M	297M	121M	49M

Algorithm 1 CVR&CTR neuron-connection sharing

Input: Shared network $sNET$; masks $masks[n\_tasks]$; pruning number and rate: $n\_pruning$ and q, batch number in mixed samples of CVR and CTR $n\_batches$.

▷ **Step 1: Generate CVR/CTR subnetworks**
Train $sNET$ with mixed CVR and CTR samples to get weights θ
for *tid* in [CVR,CTR] **do**
    Initialize $masks[tid][0] = 1^{|\theta|}$.
    for $i = 1$ to $n\_pruning$ **do**
        Train $sNET$ for an epoch of the *tid*'s samples after $sNET = sNET \odot masks[tid][i-1]$;
        Let x equal to the qth quantile of the absolute value of $sNET$'s weights, then update $masks[tid][i-1]$ by setting values less than x to 0 to obtain $mask[tid][i]$;
        Reset $sNET$'s weights to θ.
    end for
    Select best i of $masks[tid][i]$ based on validation set's performance as $masks[tid][best\_i]$.
end for
▷ **Step 2: Training the final sharing model**
Reinitialize $sNET$ parameters as θ.
for $batch = 1$ to $n\_batches$ **do**
    Obtain the *tid* of current sample batch,e.g., CTR;
    Update $sNET = sNET \odot masks[tid][best\_i]$;
    Train $sNET$ to calculate loss and do gradient updates of weights.
end for
$CVR\_model = sNET \odot masks[CVR][best\_i]$.
$CTR\_model = sNET \odot masks[CTR][best\_i]$.

3 Experiments

3.1 Experimental Setup

Datasets. In our trials, we employ traffic logs of 9 consecutive days in an industry video recommender system, for there are no public video datasets containing both clicks and conversions. Table 1 summarizes the dataset's statistics, with the first eight days used for training and the last day for testing.

Competitors. (1) **SingleTask** is the typical single-task model in Fig.1(a). (2) **LayerShare** is the classic layer-level sharing model in Fig.1(b). (3) **ConnectionShare** is our proposed method in Fig.1(c). (4) **NeuronShare** is a variant of **ConnectionShare**, which generates the sub-nets by removing specific neurons instead of connections. We use the same network hyper-parameters for all competitors except that the layer-level sharing model adds an extra tower.

Metric. Although our main target is CVR, we also evaluate CTR as an auxiliary observation because typical MTL solutions for CVR often degrade CTR's performance [8, 9]. **(1) offline metric**. Mean Squared Error (MSE) and Area under the ROC curve (AUC) are adopted for CVR and CTR, respectively. **(2) online metric**. We adopt the video view time. RS will recommend the highest k videos based on the ranking score, $rankscore = pCTR * pCVR * video\_length$, which combines three parts to simulate the final view time.

Table 2. Offline Comparison of Different Models

Model	MSE CVR	AUC CTR	MTL Gain CVR	MTL Gain CTR
1) SingleTask	0.13688	0.78572	-	-
2) LayerShare	0.13563	0.78808	+0.00125 (-0.91%)	+0.00236 (+0.30%)
3) ConnectionShare	**0.13226**	**0.78874**	**+0.00462 (-3.38%)**	**+0.00302 (+0.38%)**
4) NeuronShare	0.13531	0.78346	+0.00157 (-1.15%)	-0.00226 (-0.29%)

Table 3. Online A/B Test Comparison of Different Models

Rankscore		View		
pCTR Model	pCVR Model	Seconds	CVR	CTR
1) SingleTask	SingleTask	491.3	54.9%	24.7%
2) SingleTask	LayerShare	481.6	55.0%	**24.8%**
3) SingleTask	ConnectionShare	**498.6**	55.1%	24.7%
4) ConnectionShare	ConnectionShare	497.1	**55.3%**	**24.8%**

3.2 Experiment Results

The offline experiment results are illustrated in Table 2. For an industrial RS, a 1% relative reduction in CVR MSE is remarkable and can acquire significant online performance improvement. Specifically, (1) ConnectionShare CVR's MSE is reduced relatively by 3.38% and 2.49% compared to the single-task model and the layer-sharing model, respectively. This improvement validates the effectiveness of our approach for CVR modeling, which is achieved by expanding the capacity of knowledge transfer by fine-grained sharing. (2) ConnectionShare CTR achieves a slight gain, which is beyond expectation. This indicates that our approach can alleviate conflict by reducing knowledge negative transfer [8, 9]. (3) NeuronShare underperforms ConnectionShare, which further verifies the necessity of introducing finer-grained sharing.

The online A/B tests on the real-world video platform with nearly one billion users are conducted for 7 consecutive days, illustrated in Table 3. The first three experiments show that ConnectionShare CVR outperforms SingleTask and LayerShare models with 1.5% and 3.5% improvement in online view time respectively, which are remarkable in industrial video RS. Experiment 4) Combining ConnectionShare CVR with Connection CTR does not achieve further performance gains.

4 CONCLUSIONS

In this paper, we propose a neuron connection-level sharing scheme in CVR modeling. It automatically learns which connection weights need to be shared or not. Compared to the current literature on subnetwork/layer level sharing, we achieve a finer-grained sharing to transfer knowledge discreetly and meticulously and thus further extend the capacity to address sharing&conflict issues in MTL CVR modeling. This is the first time that sharing granularity at the neuron connection level is proposed in CVR modeling. Experiments on offline and online video recommender systems demonstrate the superiority of our method.

References

1. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=rJl-b3RcF7
2. Frankle, J., Dziugaite, G.K., Roy, D.M., Carbin, M.: Stabilizing the lottery ticket hypothesis. arXiv preprint arXiv:1903.01611 (2019)
3. Ma, J., Zhao, Z., Chen, J., Li, A., Hong, L., Chi, H.: Snr: Sub-network routing for flexible parameter sharing in multi-task learning. In: AAAI (2019)
4. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: 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. pp. 1930–1939 (2018)
5. Ma, X., Zhao, L., Huang, G., Wang, Z., Hu, Z., Zhu, X., Gai, K.: Entire space multi-task model: An effective approach for estimating post-click conversion rate. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1137–1140 (2018)
6. Naumov, M., Mudigere, D., Shi, H.J.M., Huang, J., Sundaraman, N., Park, J., Wang, X., Gupta, U., Wu, C.J., Azzolini, A.G., et al.: Deep learning recommendation model for personalization and recommendation systems. arXiv preprint arXiv:1906.00091 (2019)
7. Sun, T., Shao, Y., Li, X., Liu, P., Huang, X.: Learning sparse sharing architectures for multiple tasks. In: AAAI. pp. 1930–1939 (2020)
8. Tang, H., Liu, J., Zhao, M., Gong, X.: Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In: Fourteenth ACM Conference on Recommender Systems. pp. 269–278 (2020)
9. Torrey, L., Shavlik, J.: Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, pp. 242–264. IGI global (2010)
10. Wen, H., Zhang, J., Wang, Y., Lv, F., Bao, W., Lin, Q., Yang, K.: Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2377–2386 (2020)
11. Zhang, H., Wang, S., Zhang, K., Tang, Z., Jiang, Y., Xiao, Y., Yan, W., Yang, W.Y.: Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2407–2416 (2020)
12. Zhang, Y., Yang, Q.: An overview of multi-task learning. National Science Review **5**(1), 30–43 (2018)
13. Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., Kumthekar, A., Sathiamoorthy, M., Yi, X., Chi, E.: Recommending what video to watch next: a multitask ranking system. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 43–51 (2019)
14. Zhu, H., Jin, J., Tan, C., Pan, F., Zeng, Y., Li, H., Gai, K.: Optimized cost per click in taobao display advertising. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 2191–2200 (2017)