Preliminary Evaluation of the Test-Time Training Layers in Recommendation System (Student Abstract)

Tianyu Zhan, Zheqi Lv, Shengyu Zhang*, Jiwei Li*

Zhejiang University
{yuzt,zheqilv,sy_zhang,jiwei_li}@zju.edu.cn

Abstract

This paper explores the application and effectiveness of Test-Time Training (TTT) layers in improving the performance of recommendation systems. We developed a model, TTT4Rec, utilizing TTT-Linear as the feature extraction layer. Our tests across multiple datasets indicate that TTT4Rec, as a base model, performs comparably or even surpasses other baseline models in similar environments.

Introduction

The rapid advancement of deep learning has significantly influenced the development of recommendation systems, leading to the emergence of several prominent sequential recommendation models, including DIN(Zhou et al. 2018), GRU4Rec(Hidasi et al. 2016), SASRec(Kang and McAuley 2018), Bert4Rec(Sun et al. 2019), ComiRec(Cen et al. 2020), ICLRec(Chen et al. 2022), and so on. With the advancement of technology and actual demand, recommendation technology has begun to develop toward personalization(Lv et al. 2023, 2024a,b; Sun et al. 2022), multimodality(Zhang et al. 2021, 2022, 2023a; Ji et al. 2023), privacy protection(Liao et al. 2023), cold-start(Liu et al. 2023; Chen, Wang, and Yin 2021), session-based(Su et al. 2023), disentangle(Chen, Xu, and Wang 2021), LLM-based(Zhang et al. 2023b; Bao et al. 2023), generative-based(Zhao et al. 2024; Lin et al. 2024), and so on.

Recently, a novel architecture known as Test-Time-Training (TTT) layers(Sun et al. 2024) has been proposed. This architecture employs self-supervised learning to update the weights of hidden states by performing gradient descent for each token. Two variants of TTT have been introduced: TTT-Linear and TTT-MLP, with TTT-Linear demonstrating superior performance and efficiency in short-context scenarios. Given the short-context nature and high-efficiency demands of recommendation systems, we conducted a preliminary exploration of TTT-Linear's performance within this context.

In this preliminary exploration, we developed a simple but effective sequential recommendation model, TTT4Rec, utilizing TTT-Linear as the feature extraction layer.

*Corresponding authors.

Preliminaries

TTT's central idea leverages self-supervised learning to compress historical context $x_1, \ldots, x_t$ into a hidden state s_t. Here, the context is treated as an unlabeled dataset and the state as a model. The hidden state s_t corresponds to W_t, the weights of a model f, which can be a linear model, a small neural network, or another form. The output is defined as:

$$z_t = f(x_t; W_t)$$

The output token represents the prediction on x_t made by f with the updated weights W_t. The update rule is a step of gradient descent on a self-supervised loss ℓ:

$$W_t = W_{t-1} - \eta \nabla \ell(W_{t-1}; x_t)$$

where η is the learning rate.

The algorithm maps an input sequence $x_1, \ldots, x_T$ to an output sequence $z_1, \ldots, z_T$ using the hidden state, update rule, and output rule described above. Even during testing, the new layer trains a unique sequence of weights $W_1, \ldots, W_T$ for each input sequence.

Model Architecture

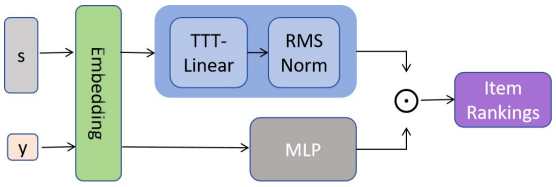

Figure 1: Model Architecture of TTT-Linear in Recommendation System

We define a data instance as $\mathcal{X} = \{u, v, s\}$ and the corresponding label as $\mathcal{Y} = \{y\}$, where u represents the user ID, v denotes the item ID, and s indicates the user's click sequence.Figure1 illustrates the architecture of TTT4Rec. .

Embedding Layers: Consider the user's click sequence $s = \{v_i\}_{i=1}^{N}$, with "0" serving as a padding token. For each item ID v_i, latent representations are obtained through an embedding layer. This layer processes v_i to produce its ID embedding $e_v \in \mathcal{R}^K$.

Feature Extraction Layer: The set of embeddings $E_s =$

$\{e_1, e_2, e_3, \ldots, e_N\}_{i=1}^{\mathcal{N}}$ is then processed by the TTT-Linear layer for feature extraction. The extracted features are subsequently normalized using RMSNorm, and the hidden state F_s corresponding to the last valid click is output.

Target Prediction: To conserve resources, the predicted target y is passed through the same embedding layer to retrieve the target item embedding e_y. This embedding is then fed into a two-layer MLP to extract the target features F_t. Finally, the dot product of F_t with the click sequence features F_s yields the predicted click probability for the target.

$$R = F_s \cdot F_t$$

Experiment Setup

Datasets: Our evaluation was carried out on three widely recognized datasets frequently used in recommendation research: Amazon Beauty, Amazon Electronics, and MovieLens-1M. For negative sampling, we adopted a 1:4 ratio during the training phase, where each positive instance is paired with four negative samples. In contrast, a 1:99 ratio was applied during the testing phase to closely approximate real-world recommendation scenarios, enhancing the robustness of our model's performance assessment.

Baselines: To evaluate TTT4Rec, we implemented several sequential recommendation models, including DIN(Zhou et al. 2018), GRU4Rec(Hidasi et al. 2016), SASRec(Kang and McAuley 2018), and ComiRec(Cen et al. 2020).

During the experiments, all models except ComiRec converged within the first 10 epochs. To fairly assess the performance, the experiments were conducted with a learning rate of 0.001 and 10 training epochs(50 epochs for ComiRec).

Experimental Results

Dataset	Model	Metrics			
		NDCG@5	NDCG@10	HR@5	HR@10
Beauty	DIN	0.2466	0.2684	0.3546	0.4223
	GRU4Rec	0.2390	0.2692	0.3645	0.4571
	SASRec	0.2512	0.2849	0.3854	0.4893
	ComiSA	**0.2594**	0.2783	0.3842	0.4813
	ComiDR	0.2582	**0.2925**	0.3833	**0.4902**
	TTT4Rec	0.2571	0.2890	**0.3866**	0.4848
Electronic	DIN	0.3286	0.3664	0.4434	0.5603
	GRU4Rec	0.3450	0.3844	0.4666	0.5882
	SASRec	0.3420	0.3821	0.4641	0.5881
	ComiSA	0.3065	0.3301	0.4221	0.5418
	ComiDR	0.3228	0.3623	0.4414	0.5637
	TTT4Rec	**0.3486**	**0.3892**	**0.4712**	**0.5968**
ML-1m	DIN	0.4703	0.5081	0.6318	0.7485
	GRU4Rec	0.5209	0.5566	0.6762	0.7863
	SASRec	0.5217	0.5574	0.6790	**0.7919**
	ComiSA	0.4040	0.4450	0.5417	0.6982
	ComiDR	0.4072	0.4549	0.5550	0.7022
	TTT4Rec	**0.5239**	**0.5577**	**0.6834**	0.7877

Table 1: Performance comparison of different methods. The best performance is highlighted in bold.

Overall Performance: Table 1 summarizes the evaluations of TTT4Rec and baseline models across the datasets. TTT4Rec matches or surpasses the baselines in NDCG and Hit metrics. While ComiRec performs better overall due to advanced techniques, its training efficiency is lower, requiring 50 epochs on the Beauty dataset to match the performance TTT4Rec achieves in a single epoch (Figure 2).

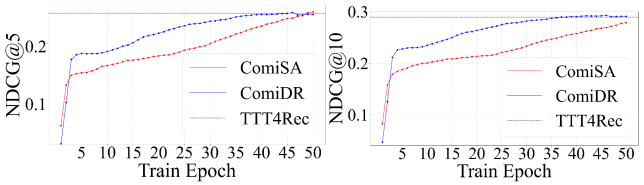

Figure 2: The NDCG@5 metrics for ComiSA and ComiDR evolve during training, with the dashed line showing TTT4Rec's performance at epoch 1 on the Beauty dataset.

Hyper-parameter Analysis: In our experiments, we identified two parameters that had a significant impact on performance, as shown in Figure 3:

1. **initializer_range:** This parameter sets the standard deviation for initializing TTT layer parameters. A value that's too small leads to minimal parameter variation, impairing functionality, while a value that's too large makes the parameter space too broad, complicating the search for optimal parameters.
2. **mini_batch_size:**This parameter represents the number of tokens processed simultaneously in the mini-batch TTT parallelization strategy. The results suggest that optimal performance is achieved with batch sizes of 1 or 10, indicating that both online and batch gradient descent (GD) are more effective for TTT-Linear in this context. Mini-batch GD updates the current token using W derived from several tokens apart, which may link unrelated tokens and adversely affect performance.

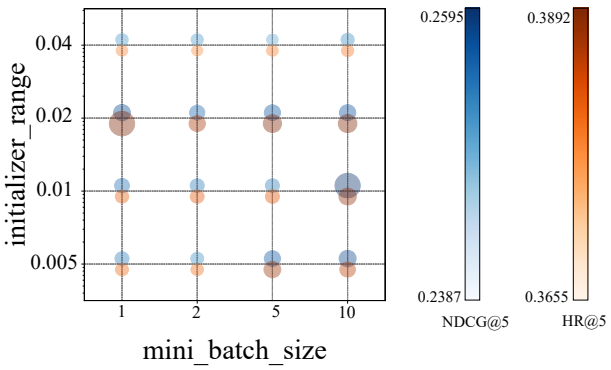

Figure 3: Hyperparameter Grid Search.(Beauty Dataset)

Conclusion and Future Work

As a base model utilizing TTT-Linear for feature extraction, TTT4Rec has shown performance that is on par with, or even exceeds, other base models within the same environment. Future research will aim to further refine the parameter W in TTT-Linear.

Acknowledgements

This work was in part supported by 2030 National Science and Technology Major Project (2022ZD0119100), National Natural Science Foundation of China (No. 62402429, 62441605), the Key Research and Development Program of Zhejiang Province (No. 2024C03270), ZJU Kunpeng&Ascend Center of Excellence.

References

Bao, K.; Zhang, J.; Zhang, Y.; Wang, W.; Feng, F.; and He, X. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 1007–1014.

Cen, Y.; Zhang, J.; Zou, X.; Zhou, C.; Yang, H.; and Tang, J. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2942–2951.

Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, 2172–2182.

Chen, Z.; Wang, D.; and Yin, S. 2021. Improving cold-start recommendation via multi-prior meta-learning. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, 249–256. Springer.

Chen, Z.; Xu, Z.; and Wang, D. 2021. Deep transfer tensor decomposition with orthogonal constraint for recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4010–4018.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations 2016*.

Ji, W.; Liu, X.; Zhang, A.; Wei, Y.; Ni, Y.; and Wang, X. 2023. Online distillation-enhanced multi-modal transformer for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, 955–965.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206. IEEE.

Liao, X.; Liu, W.; Zheng, X.; Yao, B.; and Chen, C. 2023. Ppgencdr: A stable and robust framework for privacy-preserving cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4453–4461.

Lin, X.; Yang, C.; Wang, W.; Li, Y.; Du, C.; Feng, F.; Ng, S.-K.; and Chua, T.-S. 2024. Efficient Inference for Large Language Model-based Generative Recommendation.

Liu, W.; Zheng, X.; Chen, C.; Su, J.; Liao, X.; Hu, M.; and Tan, Y. 2023. Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, 383–394.

Lv, Z.; He, S.; Zhan, T.; Zhang, S.; Zhang, W.; Chen, J.; Zhao, Z.; and Wu, F. 2024a. Semantic Codebook Learning for Dynamic Recommendation Models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9611–9620.

Lv, Z.; Zhang, W.; Chen, Z.; Zhang, S.; and Kuang, K. 2024b. Intelligent model update strategy for sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, 3117–3128.

Lv, Z.; Zhang, W.; Zhang, S.; Kuang, K.; Wang, F.; Wang, Y.; Chen, Z.; Shen, T.; Yang, H.; Ooi, B. C.; et al. 2023. Duet: A tuning-free device-cloud collaborative parameters generation framework for efficient device model generalization. In *Proceedings of the ACM Web Conference 2023*, 3077–3085.

Su, J.; Chen, C.; Liu, W.; Wu, F.; Zheng, X.; and Lyu, H. 2023. Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation. In *Proceedings of the ACM Web Conference 2023*, 165–176.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Sun, T.; Wang, C.; Song, X.; Feng, F.; and Nie, L. 2022. Response generation by jointly modeling personalized linguistic styles and emotions. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(2): 1–20.

Sun, Y.; Li, X.; Dalal, K.; Xu, J.; Vikram, A.; Zhang, G.; Dubois, Y.; Chen, X.; Wang, X.; Koyejo, S.; Hashimoto, T.; and Guestrin, C. 2024. Learning to (Learn at Test Time): RNNs with Expressive Hidden States. arXiv:2407.04620.

Zhang, J.; Liu, Q.; Wu, S.; and Wang, L. 2023a. Mining Stable Preferences: Adaptive Modality Decorrelation for Multimedia Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 443–452.

Zhang, J.; Zhu, Y.; Liu, Q.; Wu, S.; Wang, S.; and Wang, L. 2021. Mining latent structures for multimedia recommendation. In *Proceedings of the 29th ACM international conference on multimedia*, 3872–3880.

Zhang, J.; Zhu, Y.; Liu, Q.; Zhang, M.; Wu, S.; and Wang, L. 2022. Latent structure mining with contrastive modality fusion for multimedia recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 9154–9167.

Zhang, Y.; Feng, F.; Zhang, J.; Bao, K.; Wang, Q.; and He, X. 2023b. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488*.

Zhao, J.; Wenjie, W.; Xu, Y.; Sun, T.; Feng, F.; and Chua, T.-S. 2024. Denoising diffusion recommender model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1370–1379.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1059–1068.