Transformer-based Recommender System and Transfer Learning

Shuning Li, Ziyi Huang

ECE-GY 7123 Deep Learning, 2024 Spring
Final Project
Codebase: https://github.com/hzsnow/NYU-ECE-GY-7123-Deep-Learning-Final-Project/tree/main
sl10916@nyu.edu, zh2931@nyu.edu

Abstract

In this project, we first pre-trained a Stochastic Shared Embedding-based Personalized Transformer(SSE-PT) model, proposed by (Wu et al. 2020) on MovieLens 1M dataset. Then we performed a transfer learning on WiKi 1000 dataset to evaluate the pre-trained model.

Introduction

In recent years, deep learning has seen tremendous successes in recommender systems due to its ability to learn complex patterns and representations from large-scale data. Among all the models and architectures, we are particularly interested in Transformer, because since its introduction in (Vaswani et al. 2017), it has become the fundamental architecture for analyzing sequential data and is increasingly applied to recommender systems to improve their performance.

Traditionally, high-quality recommender systems often require models to be trained from scratch. Inspired by the success of pre-trained language and computer vision models, we would also like to explore transfer learning in recommender systems.

Literature Survey

There are abundant literatures in transformer-based recommender system, such as (Kang and McAuley 2018), (Sun et al. 2019), (Wu et al. 2020), (Chen et al. 2019), etc. They all use Transformers in the model design to capture the sequential dynamics in the user dataset.

Overview of the Project

In this final project, we first pre-trained a sequential recommender system using a transformer on the MovieLens 1M dataset. Specifically, the transformer is used for encoding the users' preferences represented in terms of a sequence of movies viewed before such that the transformer encoder generates a new representation of the item sequence. Then, we fine-tuned the pre-trained model for the transfer learning on the Wiki 1000 dataset.

Deviation from Proposal

In our original plan, the training dataset the transfer learning target dataset were MovieLens100K and LastFM-1k-user dataset. However, as the project went on, we realized that MovieLens100K's user and item dimaension space is too small (943 users and 1682 items), and LastFM-1K user has very long item sequences, which makes them not very suitable for the task.

Methodology

Baseline Model

The Self-Attentive Sequential Recommendation(SASRec) is a transformer-based recommendation model proposed by (Kang and McAuley 2018). It uses the self-attention mechanism to further exploits the sequential pattern from the users' interaction and makes it suitable for a sequence-to-sequence recommendation. Specifically, In the self-attention mechanism, each item's attention is weighted based on the whole user's interaction sequence via a sequence-to-sequence model, encouraging each past interaction to be related to the users' future preference.

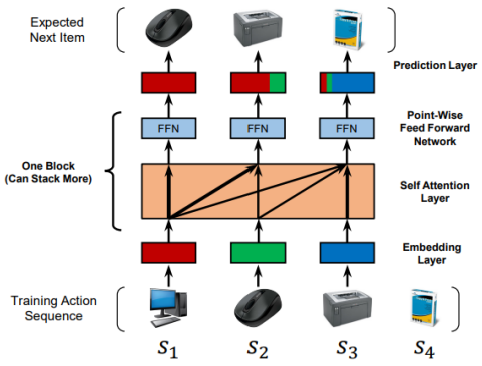

Figure 1: A simplified diagram showing the training process of SASRec created by (Kang and McAuley 2018).

To illustrate at a high level how the SASRec model processes a sequence of users' interactions with items to predict the next item the user is likely to interact with, let's look at the figure 1.

Traning Action Sequence Layer This input layer represents the sequence of items e.g., S1: Computer, S2: Mouse, etc..., that user interacted with.

Embedding Layer Each item in the sequence is then converted into an embedding vector to capture the item's feature.

Self-Attention Layer The self-attention layer computes the attention scores between all pairs of items in the sequence, which is used to determine the importance or relevance between each item in the sequence and the others.

Point-Wise Feed Forward Network The resulting attention score vectors are passed through the point-wise feed-forward network to apply some linear transformation and non-linear activation to generate higher-level representations.

Prediction Layer Lastly, the final output is used to predict the next item that the user is most likely to interact with.

Baseline Model Variant

The Stochastic Shared Embedding-based Personalized Transformer(SSE-PT) model is proposed by (Wu et al. 2020) to overcome the un-personalized characteristic of SASRec model and other original transformer models. Specifically, the SASRec and other original transformer models does not capture users' personalized user embeddings, and by integrating personalized embeddings into the transformer architecture, the model's author states that achieves a better performance.

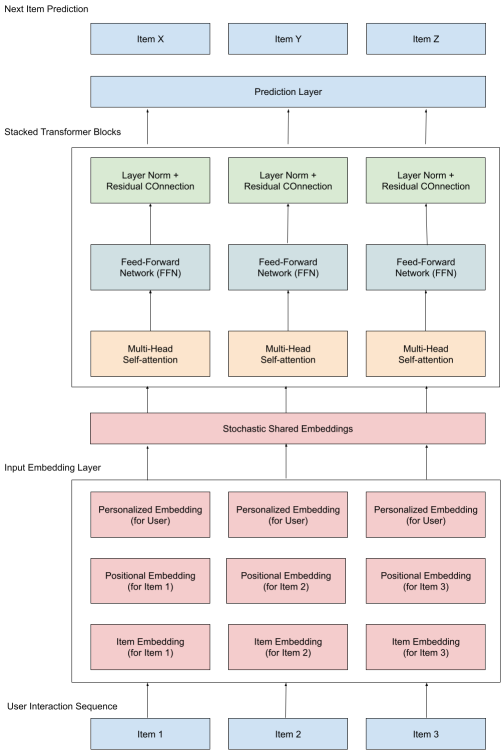

Figure 2: A simplified diagram showing the training process of SSEPT based on (Wu et al. 2020).

To illustrate at a high level how the SSEPT model processes a sequence of users' interactions with items to predict the next item the user is likely to interact with, let's look at the figure 2.

Input Layer The input layer represents the sequence of items user has interacted with.

Embedding Layer Each item in the sequence is converted into an embedding vector using item embedding, positional embedding, and personalized embedding. Notice that a personalized embedding is introduced to capture individual user preferences.

Stochastic Shared Embedding Stochastic Shared Embedding is used to improve the model generalization.

Transfomer Blocks The input embedding vectors are passed through multiple-stacked transformer blocks, consisting of multi-head self-attention, feed-forward network, and layer normalization and residual connections similar to the SASRec model.

Prediction Layer Finally, the model outputs the next item that the user is most likely to interact with.

Data Preprocessing

Dataset Overview

- Pre-train: MovieLens is a popular benchmark dataset for training recommender systems. We used the MovieLens 1M version, which is comprised of ratings, ranging from 1 to 5 stars, from 6040 users on 3416 movies. It has been cleaned up so that each user has rated at least 20 movies. Some simple demographic information such as age, gender, genres for the users and items are also available. The average item sequence length for each user is 163.50.
- Fine-tune: Wiki 1000 dataset is often used in recommendation systems containing user interactions with articles. This dataset has 8227 users and 1000 items, with an average sequence length for each user being 18.18.

Preprocessing MovieLens 1M dataset and Wiki 1000 dataset are downloaded from internet in their raw form. Thus, they are pre-processed into the form of[user_id, item_id], in which data of one user_id are grouped together and his/her item_id are sorted in the ascending order in timestamp.

Training Process

Loss Function and Optimizer Since the goal of SSEPT model is to predict the likelihood of user interactions with a specific item, which can be treated as a binary classification problem (i.e., "Will the user interact with this item next?"). Therefore, Binary Cross-Entropy (BCE) loss function is suitable because it is specifically designed for binary classification tasks. The BCE loss function measures the discrepancy between the predicted probabilities and the true binary labels, which allows the model to output the probabilities of positive interactions. As for the optimizer, we used Adam with the learning rate to 0.001 and use $\beta_1 = 0.9$ and $\beta_2 = 0.98$ as the exponential decay rates for the moment estimates.

Evaluation Metrics

- NDCG@10: Normalized discounted cumulative gain (NDCG) is a measure of ranking quality, which is often used to evaluate recommendation performance. To put it simple, it calculates a score given the ranking provided by our recommender system, and compares it to the best possible score from an ideal ranking system, so that it is possible to compare across queries or users.
- HIT@10: It measures whether or not at least one relevant item appears withing the top 10 items of your recommendation or search result.

Hyperparameters Below is a list of hyperparameters used based on (Wu et al. 2020)

- Maximum sequence length for each user: 200
- Number of transformer blocks: 2
- Number of embedding dimension for items: 50
- Number of embedding dimension for users: 50
- Number of units in the attention calculation: 100
- Number of attention heads: 1
- Dropout rate: 0.2
- L2 regularization coefficient: 0.0
- SSE threshold for user: 0.92
- SSE threshold for item: 0.9
- Number of negative examples per positive example: 100

Also, it is mentioned in (Wu et al. 2020) that a stack of 6 Transformer Blocks yields better ranking performance, so we have also tried that.

Transfer Learning Technique Even though the concept of transfer learning is simple: to freeze some layers and only train a part of the model on a different dataset, exactly how to implement it in the transformer-based recommender system was a question. With some trial and errors, we decide to freeze the encoder but change the embedding layers. However, the original implementation of embedding for both SASRec and SSE-PT are dataset-dependent, meaning different user/item size would yield different embedding layers, and made transfer learning infeasible. To overcome this, we made an alteration to the original model, so that its embedding layer size can be big enough for both the training data and the target data. In this case, we used 10000 for both user and item embedding matrix dimension.

Result

For the Pre-training on MovieLens 1M, we tried 3 different configurations: 2 transformer blocks in the encoder (original user/item embedding dimension setting), 6 transformer blocks in the encoder (original user/item embedding dimension setting)and 6 transformer blocks but with 10000 user/item embedding dimension.

Pre-train Result

Configuration 1: 2 transformer blocks + original user/item embedding dimension setting

- Epoch: 200
- Time: 718.154257 secs
- Validation: NDCG@10: 0.6060, HIT@10: 0.8366
- Testing: NDCG@10: 0.5761, HIT@10: 0.8124

Configuration 2: 6 transformer blocks + original user/item embedding dimension setting

- Epoch: 200
- Time: 718.154257 secs
- Validation: NDCG@10: 0.6123, HIT@10: 0.8391
- Testing: NDCG@10: 0.5850, HIT10: 0.8127

Configuration 3: 6 transformer blocks + 10000 user/item embedding dimension

- Epoch: 200
- Time: 749.454729 secs
- Validation: NDCG@10: 0.6083, HR@10: 0.8366
- Testing: NDCG@10: 0.5781, HR@10: 0.8101

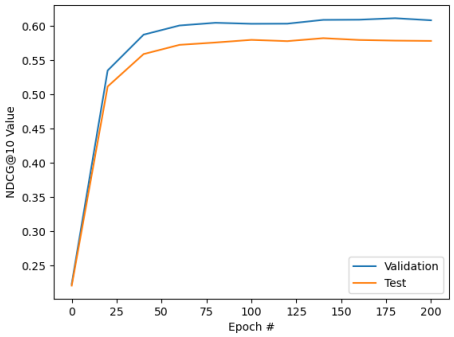

Figure 3: Pre-training Result NDGC@10 Plot

Transfer Learning Result

Then we used the same setup as Configuration 3 in Pre-training, froze its encoder and only trained its embedding layers and obtained the following result for Wiki1000 dataset. The values are substantially higher because the average sequence length are shorter for this dataset, however, we should notice that even at epoch 0 before a lot of fine-tuning, the NDGC@10 and HIT@10 is at a high level, meaning that transfer learning is a viable approach.

- Epoch: 200
- Time: 316.765916 secs
- Validation: NDGC@10: 0.8835, HIT@10:0.9169
- Testing: NDGC@10: 0.8689, HIT@10:0.9035

Conclusion

In this project, we explored effectiveness of the Stochastic Shared Embedding-based Personalized Transformer (SSE-PT) model for sequential recommendation tasks. Our experiments on the pre-trained model showed that increasing number of transformer blocks in the SSE-PT model

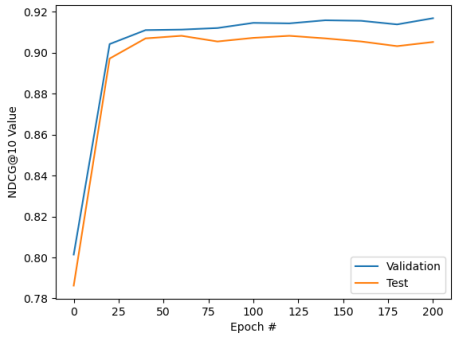

Figure 4: Transfer Learning Result NDGC@10 Plot

generally improved the ranking performance suggested by higher NDCG@10 and HR@10 scores. Our experiment on the transfer learning also showed a solid NDCG@10 and HR@10 scores, which implies the effectiveness of our pre-trained model. Therefore, we can conclude that SSE-PT model could capture

Future Work

Considering that the relative small dataset we used on transfer learning, we may use a larger dataset or a different dataset to further evaluate the performance of our pre-trained SSE-PT model.

References

Chen, Q.; Zhao, H.; Li, W.; Huang, P.; and Ou, W. 2019. Behavior sequence transformer for e-commerce recommendation in Alibaba. *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*.

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

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.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wu, L.; Li, S.; Hsieh, C.-J.; and Sharpnack, J. 2020. SSE-PT: Sequential Recommendation Via Personalized Transformer. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 328–337.