ADL Final Report

許雅晴 r10922192, 薛博文 r11944005, 胡祖望 r10723059

Abstract

In this report, we will describe the methods and results of our participation in the 2022 ADL Final Hahow Kaggle competition. The goal of the competition was to predict courses that users would purchase in the future and to predict the categories or topics that users might be interested in. This is useful for online stores to provide better recommendations to customers and help them find products that they are interested in. Our team used the BERT model and DropoutNet model for our predictions in the competition, which had four tasks: ” Seen user topic prediction”, ” Unseen user topic prediction”, ” Seen user course prediction”, and ” Unseen user course prediction.”

We successfully completed all four tasks, achieving the following final Kaggle scores: ” Seen user topic prediction” with a score of 0.28983, ” Unseen user topic prediction” with a score of 0.32317, ” Seen user course prediction” with a score of 0.17420, and ” Unseen user course prediction” with a score of 0.13684. In the following report, we will provide more information on the BERT and DropoutNet models and how we used them in the competition.

I. Introduction

A. Motives

With the innovation of information technology and wide use of the internet, we enter an era of Information Overload. In this era, both consumers and producers have encountered great challenges: Consumers: How to find the information you are interested in from a large amount of information? Producers: How to make use of the information you have and use it to attract the attention of the users? In order to solve this problem, the recommendation system was born. With the rise of Deep learning and AI, recommendation system has been boosted by deep learning and has developed rapidly. From the traditional recommendation model (collaborative filtering, matrix decomposition) to the top of the wave of deep learning (DNN, Deep Crossing, DIN, DIEN, Deep FM, DropoutNet model), are now affecting our everyday lives. The recommendation system models the user’s interest by analyzing the user’s historical behavior, so as to actively recommend to the user information that can meet their interest and needs.

B. Objectives

With 2022 ADL Final Hahow Kaggle competition, our goal is to predict two things 1. courses that the users would buy in the future and 2. which topics they are interested in and each will be divided into “seen” and” unseen”. So, there are a total of four predictions we need to make, “Seen user topic prediction”, “Unseen user topic prediction”,” Seen user course prediction”, “Unseen user course prediction”.

C. Data description

The data files we have are in two main categories: user information, course information. User information contains,” user id”, “gender”, “occupation titles”, “interests” and “recreation names”. For course information we have all courses information such as “price” and “subgroup”, of which these two will be of importance in the future. And there’s also more specific details such as course chapters. Data will be split in to train, seen validation, unseen validation, seen test, unseen test. “Unseen” means that the users are new customers, of which will be a problem we will need to solve.

For example, “Unseen user topic prediction” means that we need to predict new users of what kind of topic they would purchase in the future.

D. Methods

We’ve tried many models, we’ve tried Alternating Least Square(ALS),simple NN model, DIEN, Deep FM ,Dropout Net and Bert model. For our final model of choice, we use Dropout net for course prediction problems and use Bert model for topic prediction problems.

II. Related work

A. DropoutNet Model

1) Cold-Start Problem: Recommendation system is designed to recommend what users will like, but users and products are always changing, it's difficult to make recommendations for users or items that have little or no previous interaction or information available. In our case "Unseen" suits this condition, that's why we decide to use Dropout net to solve Cold-Start Problem.

2) DropoutNet Model: For each user, user preference and user content are input into separate DNN, then they are concatenated together and passed to the fine-tuning network which outputs the latent representation. We do the same thing with items. Multiplying the two latent matrixes together, we will get our prediction result.

III. Approach

A. Course

a) Single-Layer Neural Network: As a starting point, we used a single-layer neural network as our baseline model for the competition. This simple model is easy to implement and can be a good choice for datasets that are relatively small and straightforward. However, it may not be sufficient for tasks that require more complex patterns or higher levels of abstraction.

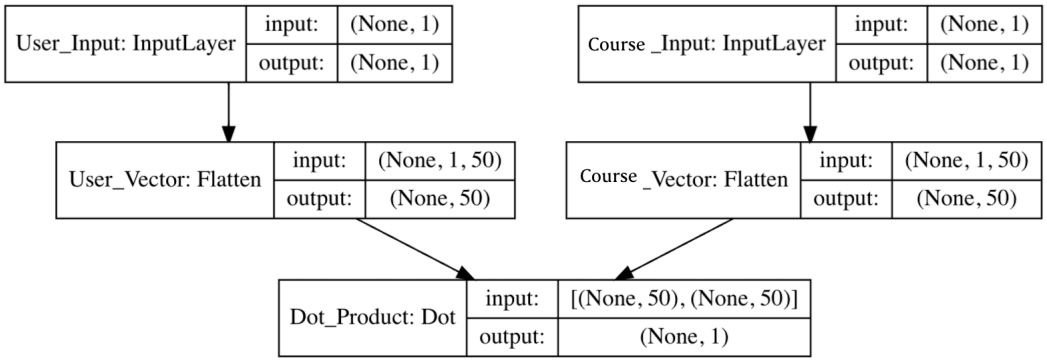

Fig. 1: In our model, we use the top 50 popular courses and user features such as interests and gender as input data for training.

We then compared the performance of the single-layer neural network with that of the Alternating Least Squares and DropoutNet models, which are more advanced techniques. This comparison allowed us to evaluate how well each model was able to capture patterns in the data and make accurate predictions.By comparing the performance of these models, we were able to understand the strengths and limitations of each model and determine which one was best suited for the task at hand. We will provide more details on our results and analysis in the following report.

b) Alternating Least Square: The Alternating Least Squares (ALS) algorithm is a useful technique for matrix factorization that can handle scalability and sparsity issues when working with large datasets. The goal of ALS is to find two matrices, U and P, such that their product is approximately equal to the original matrix of users and products.

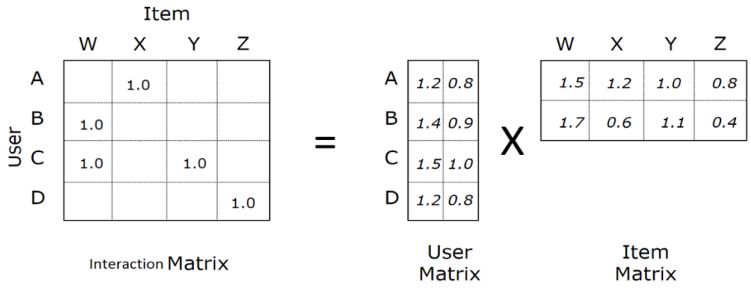

Fig. 2: Matrix Factorization of the Interaction Matrix

Once these matrices have been found, we can use them to predict how a user will feel about a particular product by multiplying the corresponding rows of U and P. This makes ALS a suitable choice for tasks like the one in this competition, where we are trying to predict user preferences for different products.

c) DropoutNet: While the models mentioned above can be effective at predicting user preferences, they do not address the ” cold-start” problem of recommending new courses to users or new users who have no previous interactions in the system. To address this issue, we used the DropoutNet model, which is designed to handle cases where there is a lack of data or information about a user.

DropoutNet is a neural network-based recommendation system that uses dropout, a regularization technique, to address the cold-start problem. It can make recommendations to new users or users with few interactions by considering their similarities with other users in the system. This helps to mitigate the sparsity issue that can occur when making recommendations to these types of users.

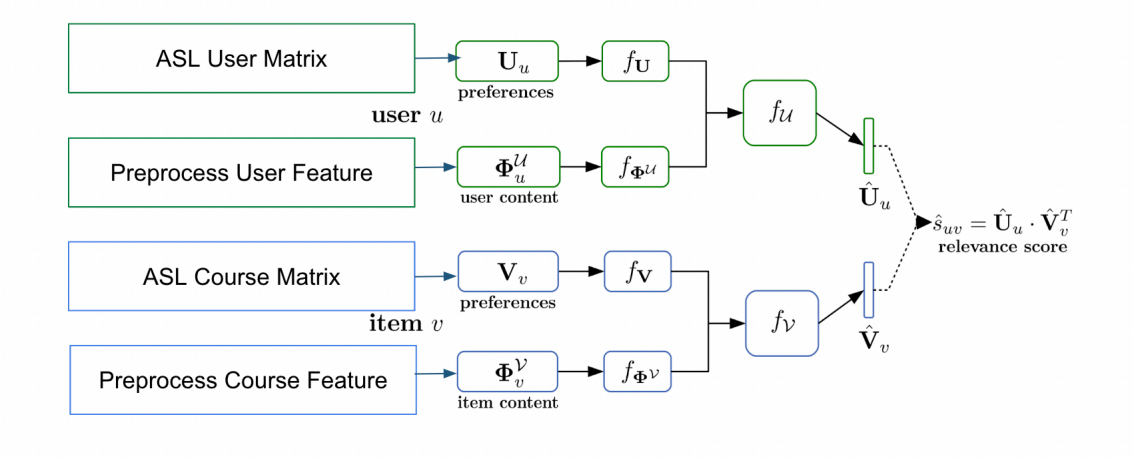

Fig. 3: The model architechure of DropoutNet.
It randomly dropout user preference or course preference information during training, which allows the model to make recommendations to new users or for new courses.

B. Subgroup

a) Approach Selection: This is a multi-label classification task. When we first found a way to do this task, we found there are some models which have to use the information for both user and subgroup sections for one label prediction within the multi-labels. However, we found that when using this way to train for our task, we will have too many parameters to train. Therefore, we want our model can almost use only users′ information but still do this recommendation task well. We found that when using to do the multi-label classification task, we can use all text information in users′ data to do multi-label classification without too many parameters that we have to train, and the performance of the model is still pretty well.

b) Data preprocess: We use Bert model to do this multi-label classification task. First, we deal with the text data to prepare our input, we concatenate all user information state by text together. The text data we used included “Interests” , “Occupation titles” and “Recreation names” , we concatenate all of these items and named them “Comment text” , these “Comment text” will be used as input for our model. Second, we count the volume of subgroup each user payment, recording these counting in 92 dimension vectors. After recording, we normalize the vectors by calculating their average. By doing data preprocess, we get input of our model and the label that we are going to use to calculate the loss and update our model.

c) Model description: Our model is constructed by connecting Bert model and one linear layer. There are three parts in the Bert model, embedding layer, encoder, and pooling layer, respectively. We first put “Comment text” into Bert tokenizer, then we put the tokenized vector into Bert’ s embedding layer, and we use the pooling layer to select the output we needed. After going through Bert model, we use one linear layer to transfer the output to 92 dimensions, which is the dimension we can use to compare with the ground truth label. While training the model, we use the label which had been preprocessed

to calculate the loss and update the parameters. When doing validation, we calculate the loss to view the state of our model, and we also take rank top-50 output to compare with the ground truth label by Map@k metric to know the ability of our model recommend the subgroup that the user willing to make payment to.

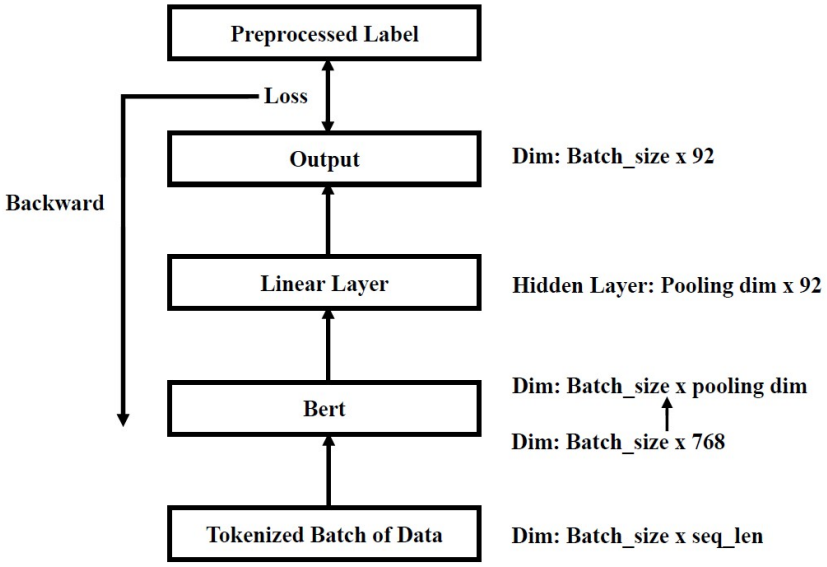

Fig. 4: The architechure of Bert Multi-Label Classification model.

d) Training Detail: Our model mainly has four parts, which are three parts of Bert model and one linear layer. The last two parts, which is the pooling layer and the linear layer are training from scratch, and the front two part are already pretrained. If we train all four parts together from the start, the parameters in the pretrained model in the front part will be polluted when we update post part of the model. Therefore, we freeze the front part of the model at the first epoch, after the classifier had been trained, we train the whole network together to make sure the update is effective.

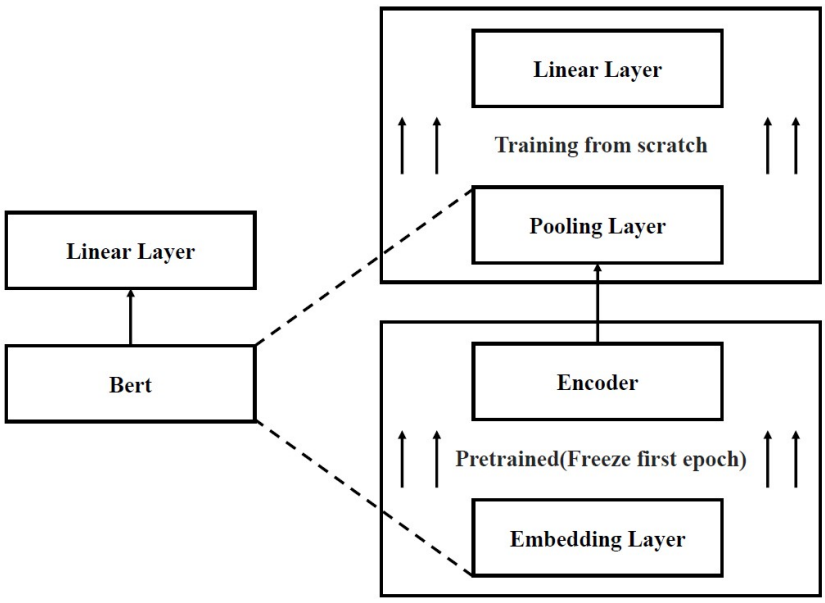

Fig. 5: Linear Layer and three parts of Bert model.
Because front parts of model were pretrained, we have to freeze it at first epoch.

IV. Experiments

A. Course

According to the results in TABLE I, the DropoutNet model performed exceptionally well on both the seen and unseen domains of the course challenge. Its ability to recommend new courses to users likely played a significant role in its success on the seen domain, while its ability to recommend courses to unseen users likely contributed to its strong performance on the unseen domain. Overall, DropoutNet demonstrated strong performance on this task.

Kaggle public scores		
Different Method	Seen Domain	Unseen Domain
Single-Layer Neural Network	0.059	0.074
Alternating Least Square	0.066	-
DropoutNet	**0.174**	**0.137**

TABLE I: Different methods and final results on the course challenge

More specifically, we experimented with different settings to improve the performance of the DropoutNet model, including adding additional features such as course published time, course subgroup or adjusting the model's hyperparameters, such as the learning rate or the number of hidden layers. The results of these experiments are presented in Table 2.

And we found that, by dividing the task into two parts - predicting free courses and predicting paid courses - we were able to gain a better understanding of the factors that influence users' purchasing decisions and tailor our recommendations accordingly. This allowed us to more accurately predict user behavior and improve the overall performance of the model.

In conclusion, our use of DropoutNet was effective in predicting courses and demonstrated the potential of this model for this task.

DroupoutNet Kaggle public scores	
Different Setting	Seen Domain
Basic	0.0843
Using addition features	0.0808
Dividing into two parts	**0.174**

TABLE II: Our experiments with the DropoutNet model on the seen domain

B. Subgroup

a) Pretrained model: We have tested a few kinds of pretrain-model, including "bert-base-chinese", "hfl/chinese-roberta-wwm-ext" and "hfl/chinese-roberta-wwm-ext-large". The training result shows that both "bert-base-chinese" and Roberta model can make a good prediction, "bert-base-chinese" have better performance on seen test data, and the Roberta model has better performance on validation data and unseen data. However, the performance of the large pretrain model is worse than the smaller pretrain model. When we observe the prediction of the model, we surprisingly found that almost all of the predictions just take the most popular subgroup as the recommendation, without considering the user's information.

Kaggle public scores		
Different pretrain model	Seen Domain	Unseen Domain
bert-base-chinese	0.2925	0.3099
hfl/chinese-roberta-wwm-ext	0.2878	0.2939
hfl/chinese-roberta-wwm-ext-large	0.2151	-

TABLE III: Our experiments with the Different pretrain model

b) Adding features: We had also considered adding other information as input for the model, there are two kinds of features we try to add, which is gender and total subgroup volume. First, we add gender into the model, the result shows that it may help a little bit but not induce significant improvement. We think it is because our Bert model is still big and our feature has only four dimensions, and the information is still not big enough to improve the model. Second, we try to concatenate the total subgroup volume to the model. However, the performance got worse after using the feature, and we found that the prediction also has the phenomenon that the model just takes the most popular subgroup as a recommendation. In conclusion, we think if we want to add other features to improve the performance

of the model, we should add more information but better not to add the information about most popular groups.

Kaggle public scores		
Different features using	Seen Domain	Unseen Domain
Basic	0.2925	0.3099
Using addition features(gender)	0.2878	0.2939
Using total subgroup volume	0.2118	-

TABLE IV: Our experiments with adding Different features

c) Counting subgroup: We did not count the volume of subgroups of each user payment in the beginning. At the time, our 92 dimension label only use zero and one to represent if the user bought the subgroup before. We found this label can not know the degrees of user' s favor inter different subgroups which user bought. Therefore, we count this information and use the new label. the score in the Table shows this information improves our model a lot. The model performs better not just in seen test data, but also in unseen test data.

Kaggle public scores		
Different features using	Seen Domain	Unseen Domain
Not counting subgroup	0.2551	0.2497
Counting subgroup	0.2925	0.3099

TABLE V: Different scores of counting the volume of users´ purchased record of subgroups or not

V. Discussion and Conclusion

We conducted some experiments and found that both course-related factors (such as release time, price, and subgroup) and user-related factors (such as gender, profession, interests, and recreation) are effective features. Based on these features, we can use DropoutNet and Multi-Label as our approaches to this problem.

To improve the recommendation system, a larger dataset is definitely needed. Besides, we also believe that having more information would allow us to utilize a wider range of methods to address the issue. During our experiments, we found that the current program is missing some potentially significant features, such as the level of difficulty of courses and user comments. In addition, there are factors related to the user that may be taken into account, such as the user' s age and the time when the courses were selected. Techniques like Transformers and RNN-based approaches like Seq2seq cannot be implemented under this level of supervision. We certainly hope these information could be utilized in the future to make the prediction become more accurate.

VI. Work Distribution

Name	Work
許雅晴	Coding & Report Writing
薛博文	Coding & Report Writing
陳韻帆	Coding & Report Writing
胡祖望	Coding & Report Writing

References

[1] Maksims Volkovs, Guangwei Yu, Tomi Poutanen. DropoutNet: Addressing Cold Start in Recommender Systems. , 2017.