Deep Knowledge Tracing with GRU and Learning State Enhancement

Xiaoyu Han[1], Shu Zhang[1,2(✉)], Juxiang Zhou[1,3], Zijie Li[1], and Jun Wang[1,3]

[1] Key Laboratory of Education Informatization for Nationalities, Ministry of Education, Yunnan Normal University, Kunming 650500, China
zhangshu@ynnu.edu.cn

[2] School of Information Science and Technology, Yunnan Normal University, Kunming 650500, China

[3] Yunnan Key Laboratory of Smart Education, Yunnan Normal University, Kunming 650500, China

Abstract. With the rapid development of artificial intelligence + education, knowledge tracing, as the core technology of adaptive education system, has gradually become a challenging research hotspot in the field of intelligent education. In recent years, the deep knowledge tracing model (DKT), which successfully applied neural network in knowledge tracing field for the first time, has made a great breakthrough in prediction accuracy, and has aroused the wave of knowledge tracing using neural network since then. In DKT, the recurrent neural network (RNN) stores the previous information of students in the hidden layer parameters. However, due to the continuous accumulation of hidden layer information, it is difficult to re-extract the important information at the earlier time, resulting in the deviation of prediction results. Meanwhile, the model does not consider the role of students' recent state. That often has a more important impact on students' current level of doing problems. Inspired by the above questions, we improved the DKT model and used the gate units of GRU model to determine the retention and forgetting of previous information, so as to solve the problem that the important information at the early time was difficult to use due to the continuous accumulation of hidden layer information. At the same time, the module of enhancing students' learning state is added in the model, and the recent learning information of students is effectively used to enhance students' recent learning state. The experimental results of the Assistment2009 and Assistment2017 public datasets show that the model proposed in this paper can effectively improve the accuracy of model prediction.

Keywords: Deep knowledge tracing · GRU model · Personalized learning

1 Introduction

In the teaching process, the individual needs of multiple students need to be met. Due to the limited attention and energy of teachers, students' learning state is constantly changing, so it is almost impossible for teachers to meet the personalized learning needs

Y. Xu et al. (Eds.): ML4CS 2022, LNCS 13657, pp. 677–686, 2023.
https://doi.org/10.1007/978-3-031-20102-8_53

of each student. Teachers generally judge students' mastery of knowledge points by their classroom performance and homework exercises. It is very complicated and time-consuming for teachers to mark students' homework exercises. If there are too many homework exercises to be marked, even if the teacher spends a lot of time, it is difficult to extract and summarize the knowledge points of each student's grasp of the situation. With the advent of the era of big data, the application of artificial intelligence in education is becoming more and more popular. The data generated by students in the learning process are stored in large quantities, and the ability of computers to process data is greatly strengthened. The development of educational data mining and educational data analysis has provided impetus for the development of learning forecasting. Knowledge tracing has become one of the important tools to meet students' individual needs.

Knowledge tracing refers to the computer modeling of relevant knowledge based on students' previous learning information, and the prediction of students' next answer performance based on students' previous problem-solving data [1]. To put it simply, the knowledge tracing task is to find a way to obtain the current knowledge state of students through the historical sequence data of students. Using the interactive information between students and questions, the purpose of predicting students' next answer performance is achieved.

Traditional knowledge tracing models include Bayesian knowledge tracing using Hidden Markov model [2] and PFA using Logistic regression model [3]. In 2015, neural network was successfully applied in the field of knowledge tracing for the first time, and it was named Deep Knowledge Tracing (DKT) [4]. DKT has made a great breakthrough in the prediction accuracy of knowledge tracing, which has aroused the wave of knowledge tracing by using neural network. For example, DKT+ model points out that there are two problems in DKT model. The model fails to reconstruct the observed input and the model fails to reconstruct the observed input, and add three regularization terms into the loss function to solve the above problems [5]. There are also models that use neural networks from different perspectives of students' learning as entry points, such as CKT model considering students' personalized differences [14]. AKT model using monotone attention mechanism to consider the connection between the current question and each question answered by learners in the past [7]. GKT model using graph neural network to model student proficiency [8] etc. All these methods contribute to the development of neural network in knowledge tracing.

DKT uses RNN [15] to learn the sequence of knowledge points with timing to predict students' future performance of knowledge points. In DKT, students' previous information is stored in the hidden layer parameters of RNN. However, with the continuous accumulation of hidden layer information, it is difficult to extract the important information at the earlier time, making it difficult to consider the important information at the earlier time in the current prediction, resulting in the deviation in prediction. The gate units of GRU model are used to determine the retention and forgetting of past information, in order to reduce the accumulation of unimportant information in hidden layer and solve the problem that RNN in DKT is difficult to predict with important information at earlier time. We found that students' recent learning state also has a certain impact on their current level of problem solving. If the student performs well in the previous problems, but performs poorly in the recent problems, it is highly likely that the student has

problems in his recent learning state. And it is likely to affect the current performance of students. We use recent learning information to enhance students' recent learning state. The experiment proves that the above methods can make the model achieve better prediction effect.

Our main contributions are summarized as follows:

(1) We use GRU model to solve the problem that RNN in DKT is difficult to predict with important information at earlier time.
(2) We through recent learning information to enhance students' recent learning state. The accuracy of model prediction is improved effectively.

2 Related Work

Knowledge tracing is one of the important practices of artificial intelligence in education. In recent years, while improving the knowledge tracing method proposed earlier, the DKT based on RNN is also proposed, which is the first successful practice of deep neural network in the field of knowledge tracing. DKT is to build a model based on RNN to predict students' future performance through previous learning information. RNN is very effective for sequential data. It can mine temporal and semantic information in data. The ability of RNN can be used to predict students' future performance from their previous learning information.

In 1982, John Hopfield proposed the embryonic single layer feedback neural network of RNN. Although RNN at this time has the ability to process time sequence information, the defects of gradient vanishing and gradient explosion of RNN make it difficult to achieve good effects in some long-dependent scenes. In 1997, Hochreiter S and Schmidhuber J proposed Long Short Term Memory (LSTM) [16]. LSTM uses three gate units, namely forget gate, update gate and output gate. LSTM solves the problem of RNN training effectively through its gate units. Since then, various variations of LSTM have appeared [9–11]. GRU model [12] was proposed in 2014 and is one of the most famous transformations of LSTM. GRU and LSTM solve the same problems and also use gate units. The GRU uses two gate units: reset gate and update gate. In DKT, students' previous information is stored in the hidden layer parameters of RNN, but with the continuous accumulation of the hidden layer information, it is difficult to extract the important information at earlier moments, making it difficult to consider the important information at earlier moments in the current prediction. In order to solve this problem, we want to use the gate units of LSTM and GRU to determine the retention and forgetting of previous information, so as to reduce the accumulation of unimportant information in the hidden layer, and solve the problem that it is difficult to use the important information at earlier time to predict. Through experiments, we find that LSTM and GRU can achieve similar effects, but GRU has a simpler structure and easier training than LSTM. Therefore, we finally use GRU model to solve the problem that RNN in DKT is difficult to make use of important information at earlier time to predict.

In the field of knowledge tracing, scholars have made a lot of attempts to consider students' current learning state. Such as LPKT [6] simulates the learning process of students. The model is divided into three parts: learning module, forgetting module

and prediction module. The model considers the current knowledge state of students through learning and forgetting. LFKT model [13] also considers students' learning and forgetting behaviors. LFKT model comprehensively considers four factors affecting knowledge forgetting, including knowledge repetition, knowledge learning interval, sequential learning interval and knowledge mastery degree. However, we found that students' recent performance also have a very important impact on their current learning state. For example, if a student gets three questions wrong in a row, there is a high probability that the student will also get the questions wrong at the current moment. Therefore, we added a module to enhance students' learning state in the model, using students' recent learning information to enhance students' recent learning state.

3 Model

3.1 Review of Deep Knowledge Tracing Model

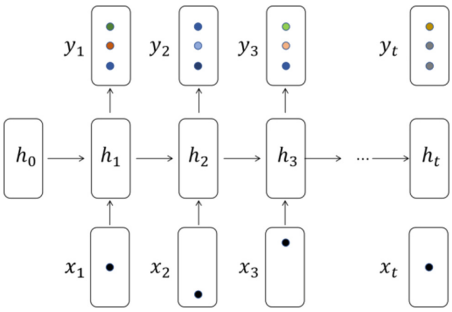

Fig. 1. DKT model schematic diagram.

Deep knowledge tracing is to predict students' future answer performance by using recurrent neural network (RNN) based on the relevant data of learners' knowledge point answers with time sequence and the relevant data of learners' correct or not answers to the knowledge point (as shown in Fig. 1). Where, x_t represents the data of students at time t, including information about knowledge points made by students at time t and information about correct or wrong answers. h_t represents the hidden layer state of recurrent neural network (RNN) at time t, and represents the comprehensive problem-solving information of students before time t. y_t represents the prediction of the students' performance in the next time. Because the model does not know which knowledge points the students will make next time by default, each prediction is the prediction of the correct probability of all the knowledge points the students will make next time.

The information transfer in the model can be simply described as follows: x_t will be put into the recurrent neural network to generate the original prediction data h_t (see Eq. 1). Then put the output of h_t through a fully connected layer into the activation function, control each element of the output between 0 and 1, and get the final prediction result (see Eq. 2, 3).

$$h_t = \tanh\left(W_x x_t + b_x + W_h h_{(t-1)} + b_h\right) \tag{1}$$

where, h_t is the hidden state at time t, x_t is the input at time t, and $h_{(t-1)}$ is the hidden state at time $t-1$.

$$I = h_t A^T + b \tag{2}$$

where, h_t and I are the input and output of the linear layer, A is the weight, b is the bias.

$$\text{Sigmoid}(I) = \sigma(I) = \frac{1}{1 + \exp(-I)} \tag{3}$$

3.2 DKT Model Improvement Framework

We cut and onehot encoding the data of students, so that each information of students contains information about the skills they have done and the correct or incorrect information of students' answers, and there is a time sequence between the data. Form a sequence of students doing the exercises $\{x_1, x_2, x_3, x_4, ..., x_T\}$. Input students' problem-solving sequence into our model accordingly, and get the prediction of the model for students' next problem-solving performance. Our model schematic diagram is as shown in Fig. 2.

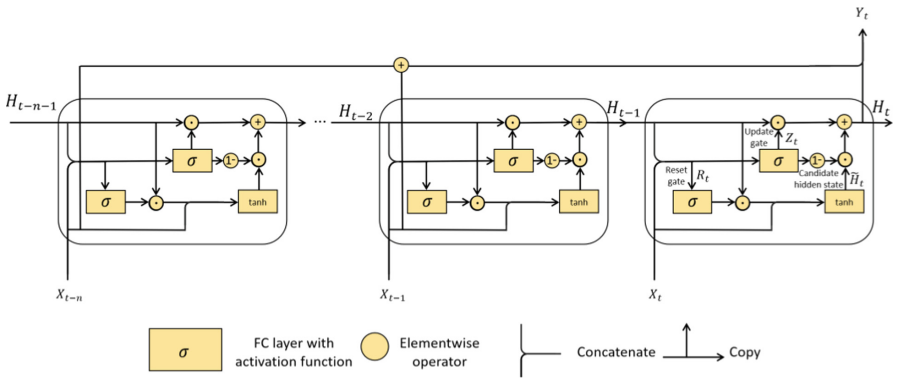

Fig. 2. Our model schematic diagram.

GRU Forms the Initial Forecast Data

The information of students' problem solving is input into our model as hidden layer information. Not all previous information is needed to predict current student performance. Therefore, we need reset gate and update gate, combined with the data of students doing questions at the current time X_t, to determine the hidden layer information needed to predict students' current answer performance. The information of the hidden layer is concated with the data of students doing the problem at the current moment. Through a fully connected layer, the vector with the same dimension as the hidden layer is obtained. Then, through the Sigmoid activation function, each element in the vector is controlled between 0–1. The reset gate and the update gate yield the same dimension as the hidden

dimension R_t and Z_t, respectively (see Eqs. 4, 5). The update gate is used to control the degree to which the information of the previous moment is brought into the current state. The larger the value of the update gate, the more information of the previous moment is brought into the current state.

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \tag{4}$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \tag{5}$$

The reset gate controls how much information from the previous state is written to the Candidate hidden state $\widetilde{H}_t$. The smaller the reset gate, the less information from the previous state is written. The reset gate yields a vector R_t with the same dimension as the hidden layer and each element being 0 to 1. The Candidate hidden state $\widetilde{H}_t$ is obtained by the initial processing of multiplying R_t by the elements of the hidden layer H_{t-1} (see Eq. 6). Because Candidate hidden state $\widetilde{H}_t$ is not directly output as the hidden state at time t, it is called Candidate hidden state. Update gate yields a vector Z_t with the same dimension as the hiddden layer and with each element value between 0 and 1. Multiply Z_t by the element of the last hidden state H_{t-1} to get the part of vector H_{t-1} that needs to remain in the current hidden state. Multiply $1 - Z_t$ by the candidate hidden state to get the part of the candidate hidden state that needs to be retained to the current hidden state. Add the two together to get the current hidden state, which is also the initial forecast data (see Eq. 7).

$$\widetilde{H}_t = \tan h\left(X_t W_{xh} + \left(R_t \odot H_{t-1}\right) W_{hh} + b_h\right) \tag{6}$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \widetilde{H}_t \tag{7}$$

Emphasis Students' Recent Learning State

$$Y_t = \sigma\left(\sum_{t-n}^{t-1} x_i W_{xy} + H_t W_{hy} + b_y\right) \tag{8}$$

where n is the number of recent problem-solving data of students that is needed to enhance their recent learning state, and H_t is the information of the hidden layer at time t.

If the student performs well in the previous problems, but performs poorly in the recent problems, it is highly likely that the student has problems in the recent learning state. And it is very likely to affect the current problem-solving performance. So if we use the recent learning information of students to enhance the students' recent learning state, the model will achieve better prediction results. The data of students' recent problem solving are spliced in the initial prediction result, and the final prediction of students' performance in the next problem solving is obtained through the full connection layer (see Eq. 8). The experiment proves that this method can make the model achieve better prediction effect.

4 Experiments

We use the Assistment2009 and Assistment2017 datasets to verify and illustrate the performance of the our model. This experiment is based on python3.8, PyTorch v1.9.1, cuda v11.1, and optimized using Adam optimizer. 70% of the data were taken as the training set and 30% as the verification set. The batch size was 64 and the number of training epochs was 70. The optimal value of 70 epochs was taken as the experimental result. The specific experimental results are shown in Table 1.

Table 1. The Assistment2009 and Assistment2017 datasets were used to test our model, and the AUC, ACC and RMSE changed with different models.

METRIC	ASSIST2009			ASSIST2017		
	AUC	ACC	RMSE	AUC	ACC	RMSE
DKT	0.8079	0.7648	0.4017	0.6917	0.6776	0.4547
OURS	0.8180	0.7693	0.3966	0.7084	0.6822	0.4520

According to the experimental data, our model performs well in the Assistment2009 and Assistment2017 datasets. In this task, it is feasible to use GRU model to solve the problem that RNN in DKT is difficult to predict with the important information at an earlier time, and to enhance students' recent learning state through students' recent learning information.

Experiments with Different Numbers of Questions

Table 2. When the size of hidden layer is 10 and the number of hidden layers is 1, the changes of AUC, ACC and RMSE of our model with the number of splicing questions.

METRIC	ASSIST2009			ASSIST2017		
	AUC	ACC	RMSE	AUC	ACC	RMSE
Q_{num}=0	0.8079	0.7648	0.4017	0.6885	0.6737	0.4571
Q_{num}=1	0.8115	0.7662	0.4006	0.6887	0.6738	0.4578
Q_{num}=2	0.8118	0.7665	0.4007	0.6882	0.6740	0.4583
Q_{num}=3	0.8098	0.7658	0.4016	0.6833	0.6724	0.4595
Q_{num}=4	0.8089	0.7631	0.4031	0.6867	0.6709	0.4591
Q_{num}=5	0.8056	0.7641	0.4025	0.6854	0.6702	0.4599

Where Q_{num} represents the number of student questions to be spliced.

As shown in Table 2, the prediction accuracy reached the highest when the original prediction results were combined with the recent two records of students. From the

previous learning records to enhance the students' recent learning state, improve the prediction accuracy, indicating that the students' recent learning state has an impact on the current students' performance.

Experiments of Different Size of Hidden Layer

Table 3. Join together the previous two problem records of students, and when the number of hidden layers is 1, the AUC, ACC and RMSE changes with the size of hidden layer of GRU.

METRIC	ASSIST2009			ASSIST2017		
	AUC	ACC	RMSE	AUC	ACC	RMSE
H = 10	0.8140	0.7686	0.3987	0.6974	0.6780	0.4557
H = 20	0.8167	0.7700	0.3974	0.7038	0.6799	0.4535
H = 30	0.8171	0.7694	0.3967	0.7068	0.6805	0.4526
H = 40	0.8180	0.7693	0.3966	0.7084	0.6822	0.4520
H = 50	0.8165	0.7663	0.3997	0.7091	0.6823	0.4516

Where H is the size of the hidden layer.

The information of students' previous problem solving is stored in the hidden layer of GRU model. The larger the size of hidden layer, the more information is stored. But the bigger the size of hidden layer is not the better. The size of hidden layer can not be too large, also can not be too small. As shown in Table 3, when the size of the hidden layer is 40, the best prediction result can be achieved in the ASSIST2009 dataset. However, in the ASSIST2017 dataset, when the size of the hidden layer is 50, the best prediction results can be achieved in this task. After comprehensive consideration, the hidden layer size of 40 is adopted in the following experiment.

Experiments of Different Number of Hidden Layers

As shown in Table 4, the effect decreases when the number of layers increases in the ASSIST2009 dataset. But in the ASSIST2017 dataset, the effect increases with the number of layers in this task.

5 Conclusion

In this paper, we put forward two problems of DKT model: the problem that RNN in DKT is difficult to predict with the important information at earlier time, and the model cannot enhance the students' recent learning state. Experiments show that it is feasible to use GRU model to solve the problem that RNN in DKT is difficult to predict with important information at an earlier time, and to enhance students' recent learning state through recent learning information in this task. In the future, we will conduct more

Table 4. When the initial prediction data is combined with two records of students' previous exercises, and the size of hidden layer of GRU is 40, the AUC, ACC and RMSE changes with the number of hidden layers.

METRIC	ASSIST2009			ASSIST2017		
	AUC	ACC	RMSE	AUC	ACC	RMSE
L = 1	0.8180	0.7693	0.3966	0.7084	0.6822	0.4520
L = 2	0.8168	0.7667	0.3993	0.7112	0.6830	0.4511
L = 3	0.8158	0.7648	0.4007	0.7117	0.6833	0.4509
L = 4	0.8155	0.7665	0.4095	0.7121	0.6832	0.4510

Where L represents the number of hidden layers.

experiments to verify the improvement effect of other methods on DKT model. For example, the idea of residual neural network can be used to solve the problem that RNN cannot perform long-term memory. At the same time, convolution neural network can be used to extract students' problem-solving patterns (for example, if students make mistakes in question A, question B is highly likely to make mistakes).

Acknowledgment. This work is supported by National Natural Science Foundation of China (Grant No. 62166050), Yunnan Fundamental Research Projects (Grant No. 202201AS070021), Scientific research foundation of Yunnan Provincial Department of Education (Grant No. 2022Y180) Yunnan Innovation Team of Education Informatization for Nationalities, and Kunming Key Laboratory of Education Informatization.

References

1. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Model. User Adapt. Interact. **4**(4), 253–278 (1994)
2. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized Bayesian knowledge tracing models. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 171–180. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_18
3. Pavlik Jr., P.I., Cen, H., Koedinger, K.R.: Performance factors analysis–a new alternative to knowledge tracing. Online Submission (2009)
4. Piech, C., Bassen, J., Huang, J., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems, 28 (2015)
5. Yeung, C.K., Yeung, D.Y.: Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: Proceedings of the Fifth Annual ACM Conference on Learning at Scale, pp. 1–10 (2018)
6. Shen, S., Liu, Q., Chen, E., et al.: Learning process-consistent knowledge tracing. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1452–1460 (2021)
7. Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2330–2339 (2020)

8. Nakagawa, H., Iwasawa, Y., Matsuo, Y.: Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 156–163 IEEE (2019)
9. Siami-Namini, S., Tavakoli, N., Namin, A.S.: The performance of LSTM and BiLSTM in forecasting time series. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 3285–3292. IEEE (2019)
10. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075 (2015)
11. Liang, X., Shen, X., Feng, J., Lin, L., Yan, S.: Semantic object parsing with graph LSTM. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) ECCV 2016. LNCS, vol. 9905, pp. 125–143. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_8
12. Dey, R., Salem, F.M.: Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1597–1600. IEEE (2017)
13. Li, X.G., Wei, S.Q., Zhang, X., Du, Y.F., Yu, G.: LFKT: deep knowledge tracing model with learning and forgetting behavior merging. Ruan Jian Xue Bao/J. Softw. **32**(3), 818–830 (2021). (in Chinese)
14. Shen, S., Liu, Q., Chen, E., et al.: Convolutional knowledge tracing: Modeling individualization in student learning process. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1857–1860 (2020)
15. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D **404**, 132306 (2020)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)