BiRNN-DKT: Transfer Bi-directional LSTM RNN for Knowledge Tracing

Bin Xu, Sheng Yan(✉), and Dan Yang

Northeastern University, NO. 3-11, Wenhua Road, Heping District, Shenyang, China
xubin@mail.neu.edu.cn, yansheng1117@foxmail.com, yangdan@ise.neu.edu.cn

Abstract. In recent years, online education is transforming from mobile education to intelligent education, and the rapid development of machine learning breathe into intelligent education with powerful energy. Deep Knowledge Tracing (DKT) is a state of the art method for modeling students' abilities which is changing by time. It can accurately predict students' mastery of knowledge or skill as well as their future performance. In this paper, we study the structure of the DKT model and proposed a new deep knowledge tracing model based on Bidirectional Recurrent Neural Network (BiRNN-DKT). We have also optimized the incorporating of data preprocessing and external features to improve model performance. Experiments show that the model can not only predict students' performance by capturing their history performance, but also get more accurate learning status simulation by integrating past and future context sequence information into the model multiple knowledge concepts. Compared with the traditional model, the proposed BiRNN-DKT gets an improvement in predicting students' knowledge ability and performance, and has great advantages in measuring the level of knowledge acquired by students.

Keywords: Knowledge tracing · Deep learning · Intelligent education · Data processing · Bidirectional Recurrent Neural Network · Student modeling

1 Introduction

Adaptive Learning (AL) is a concept that is now more talked about in the field of education technology. Its core problem is to improve students' learning efficiency through personalized learning path planning. Knowledge tracing is a time-based modeling of students' knowledge so that we can accurately predict students' mastery of knowledge points and the next performance of students. Early knowledge tracking models rely on first-order Markov models, such as Bayesian Knowledge Tracking (BKT) [2]. Deep Learning (DL) is an emerging method of machine learning research. The development of deep learning also provides a new method

This work was supported by the National Natural Science Foundation of China under Grant no. U1811261, and the National Natural Science Foundation of China under Grant no.51607029.

W. Ni et al. (Eds.): WISA 2019, LNCS 11817, pp. 22–27, 2019.
https://doi.org/10.1007/978-3-030-30952-7_3

for the development of adaptive learning. Deep Knowledge Tracking (DKT) [5] is a model that uses deep learning algorithms to evaluate student abilities using Recurrent Neural Networks (RNN). Some studies have shown that the DKT model has made great progress compared to the traditional Bayesian knowledge tracking model and has been proven to be used for course optimization [5,9,10]. For the data that crosses over a long time interval, LSTM-RNN is more suitable than base RNN [8].

The purpose of our research is to improve the effectiveness of the DKT model under the same data set, and introduce a deep knowledge tracking model based on the Bidirectional Recurrent Neural Network (BiRNN) to model the learning sequence and evaluate the students' ability at various moments.

2 Related Work

In order to build an accurate student model, we need to understand the process of student learning. The knowledge tracking task can be summarized as: the observation sequence of a student's performance on a particular learning task $x_0, x_1, x_2, ..., x_t$, predicting their next performance x_{t+1}.

2.1 Recurrent Neural Networks

RNN are mainly used to process and predict sequence data. In a Fully Connected Neural Network or Convolutional Neural Networks, the network results are from the input layer to the hidden layer to the output layer. The layers are fully connected or partially Connected, but the nodes between each layer are unconnected [1]. The source of the Recurrent Neural Networks is to characterize the relationship between the current output of a sequence and the previous information. From the network results, the RNN will remember the previous information and use the previous information to influence the subsequent output. RNN suffer from the now famous problems of vanishing and exploding gradients, which are inherent to deep networks.

2.2 Bidirectional Recurrent Neural Networks

BiRNN connects two hidden layers of opposite directions to the same output. With this form of generative deep learning, the output layer can get information from past (backwards) and future (forward) states simultaneously [3]. Those two states' output are not connected to inputs of the opposite direction states. By using two time directions, input information from the past and future of the current time frame can be used unlike standard RNN which requires the delays for including future information.

2.3 Deep Knowledge Tracing

DKT is a Seq2Seq model, and the structure of the model is LSTM [5]. The knowledge tracking problem can be described as: Given a student's observation sequence $x_0, x_1, x_2, ...$ to predict the next performance x_{t+1}, usually $x_t = \{q_t, a_t\}$, where q_t represents the question component of the answer (eg corresponding knowledge point), a_t represents whether the corresponding answer is correct, usually $a_t = \{0, 1\}$. The DKT model is based on the LSTM model to model knowledge tracking tasks.

3 Methodology

In the time series, students do not have continuous consistency in the degree of mastery of knowledge points, and the fluctuations are large. It can be concluded that a student's mastery of knowledge in the sequence is not necessarily complete. Here, we will use the Bidirectional Recurrent Neural Network model to integrate past and future contextual sequence information for prediction in order to achieve a better simulation of student learning.

3.1 Model

The batch input $X_t \in R^{n*d}$ (the number of samples is n, the number of inputs is d) for a given time step t and the hidden layer activation function are $tanh$. In the architecture of the Bidirectional Recurrent Neural Network, it is assumed that the time-step forward hidden state is $H_t \in R^{n*h}$ (the number of positive hidden cells is h), and the reverse hidden state is $H'_t \in R^{n*h}$ (The number of reversed hidden cells is h). As shown in Fig. 1.

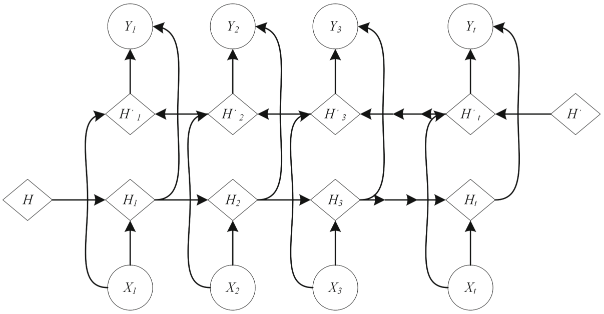

Fig. 1. Time expansion sequence of Bidirectional Recurrent Neural Network knowledge tracing model.

We can calculate the forward hidden state and the reverse hidden state separately :

$$H_t = tanh(X_t W_{xh}^{(f)} + H_t - W_{hh}^{(f)} + b_h^{(f)}) \quad (1)$$

$$H'_t = tanh(X_t W_{xh}^{(b)} + H'_t - W_{hh}^{(b)} + b_h^{(b)}) \quad (2)$$

The weight of the model is given by $W_{xh}^{(f)} \in R^{d*h}$, $W_{hh}^{(f)} \in R^{h*h}$, $W_{xh}^{(b)} \in R^{d*h}$, $W_{hh}^{(b)} \in R^{h*h}$. The model's Biases is $b_h^{(f)} \in R^{1*h}$, $b_h^{(b)} \in R^{1*h}$. Then we join the hidden states H_t and H_t' in both directions to get the hidden state H and input it to the output layer. The output layer calculates the output $O_t \in R^{n*q}$(the number of outputs is q):

$$Y_t = sigmoid(HW_{hq} + B_q) \tag{3}$$

The Weight $W_{hq} \in R^{2h*q}$ and the Biases $B_q \in R^{1*q}$ are the model parameters of the output layer. The number of hidden units in different directions can also be different.

The input data is x_t encoded by one-hot. If the model input data involves M knowledge components (such as knowledge points), each question has two results 0, 1 (corresponding to error and correct respectively), then the model input length is $2M$. For example, suppose the knowledge component of a topic is i. If the answer is correct, the $(i+1)^{th}$ bit of the corresponding input is 1 and the remaining positions are 0; if the answer is wrong, the i^{th} bit is 1 and the remaining positions are 0. The output of the model is y_t, and the length is M, which corresponds to the degree of mastery of each knowledge component (corresponding to the correctness of the questions corresponding to the knowledge component).

4 Experiment

4.1 Datasets

The problem with the Skill Builder problem set is based on specific skills, and the question can have multiple skill tags. Students must answer three questions correctly to complete the assignment. If the student uses a guide ("Prompt" or "Resolve this issue as a step"), the issue will be flagged as incorrect.

We observed that the proportion of a few abnormal student data in the raw data was too high, and there was about 20% duplicate data in the ASSISTments data set. And when the problem involves multiple knowledge points, the same problem exists in the case of reusing data. So we cleaned the data and got a new data as input to our model. The statistics of the datasets are shown in Table 1.

Table 1. The statistics of the datasets.

Dataset	Records	Students	Exercise tags
Before data processing	401757	4217	124
After data processing	328292	4217	124

In real life, students have a lot of behavioral characteristics in the process of answering questions. This information can be of great help in assessing student abilities and predicting the correctness of the next question. Therefore, we added overlap_time (time of student overlap time), hint_count (number of student attempts for this question), and first_action (type of first action: try or ask for a prompt) in the model.

4.2 Implementation of Model

We decided to build the model based on Python and using the Keras API. The focus of Keras' development is to support rapid experimentation [7]. Ability to turn your ideas into experimental results with minimal delay. We implemented the model using the loss function used in the original DKT algorithm. Our hidden layer contains 200 hidden nodes. We use the Adam algorithm to optimize the stochastic objective function to speed up our training process. We set the batch size of each epoch to 100. We set Dropout to 0.4 to avoid over-fitting of the model and use 20% of the data set as a test dataset to verify model performance [4].

4.3 Result

We trained the original DKT model and the knowledge tracking model using Bidirectional Recurrent Neural Network on the preprocessed and post-processed datasets. We used Area Under Curve (AUC) and accuracy as evaluation criteria to evaluate the model. The higher the AUC, the better the effect of the model fitting student ability. The higher the accuracy, the better the global accuracy of the model. We will organize the prediction results obtained from the experiment into Tables 2 and 3. By observing the experimental results, we can see that the model has a better effect after adding the Bidirectional Recurrent Neural Network, and it has a good performance in predicting the correctness of the next answer. And after the data cleaning and the addition of external features, the effect of the model has been significantly improved. Therefore, it can be concluded that the improved model can better simulate the student's learning state and better predict the students' understanding of the knowledge points.

Table 2. Model prediction results without introducing external features.

Datasets	DKT		BiRNN-DKT	
	AUC	ACC	AUC	ACC
Before data processing	0.81	0.78	0.82	0.84
After data processing	0.82	0.79	0.84	0.90

Table 3. Model prediction results with introducing external features.

Datasets	DKT		BiRNN-DKT	
	AUC	ACC	AUC	ACC
Before data processing	0.78	0.79	0.81	0.87
After data processing	0.80	0.81	0.85	0.93

5 Conclusion and Future Work

In this paper, we have improved the original DKT model, and the new model has added the Bidirectional Recurrent Neural Network to the original. The new model can be used to model past and future students' learning of knowledge points. It effectively reflects the level of knowledge of students after reviewing and previewing, and makes the model's prediction of student performance better. The experimental data shows that compared with the traditional model, the deep knowledge tracking model based on BiRNN has a good performance in predicting students' problems, and has certain advantages in measuring students' level. Moreover, after preprocessing the data and adding external features, the performance of the model is improved.

In the future, we will explore how to improve the model to solve the problem of long sequences. We will validate the validity of the model on more data, compare and discuss the impact of more external features on the model to improve model performance. At the same time, the problem is extended to more general exercises, and knowledge tracking will be conducted in a wider range of fields and more perspectives to help students more effectively.

References

1. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Comput. **1**(2), 270–280 (1989)
2. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User-Adap. Inter. **4**(4), 253–278 (1994)
3. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. IEEE Trans. Sig. Process. **45**(11), 2673–2681 (1997)
4. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
5. Piech, C., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems, pp. 505–513 (2015)
6. ASSISTments Data (2015). https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010. Accessed 07 Mar 2016
7. Ketkar, N.: Introduction to Keras. Deep Learning with Python, pp. 95–109. Apress, Berkeley (2017). https://doi.org/10.1007/978-1-4842-2766-4_7
8. Zheng, H., Shi, D.: Using a LSTM-RNN based deep learning framework for ICU mortality prediction. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 60–67. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_6
9. Wang, L., Sy, A., Liu, L., et al.: Deep knowledge tracing on programming exercises. In: Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale, pp. 201–204. ACM (2017)
10. Zhang, L., Xiong, X., Zhao, S., et al.: Incorporating rich features into deep knowledge tracing. In: Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale, pp. 169–172. ACM (2017)