Early Detection of At-risk Students based on Knowledge Distillation RNN Models

Ryusuke Murata
Department of Advanced Information Technology
Kyushu University, Japan
murata@limu.ait.kyushu-u.ac.jp

Atsushi Shimada
Department of Advanced Information Technology
Kyushu University, Japan
atsushi@limu.ait.kyushu-u.ac.jp

Tsubasa Minematsu
Department of Advanced Information Technology
Kyushu University, Japan
minematsu@limu.ait.kyushu-u.ac.jp

ABSTRACT

Recurrent neural network (RNN) achieves state-of-the-art in several researches of the performance prediction. However, accuracy in early time steps is lower than that in late time steps, even though the early detection of at-risk students is important for timely interventions. To improve the accuracy in early time steps, we propose a knowledge distillation method for RNN. Our method distills the time-series information in the RNN model of late time steps into the RNN model of early time steps. This distillation makes the prediction of early time steps closer to that of late time steps. The experimental result showed that our method improved the detection rate of at-risk students compared with traditional RNNs, especially in early time steps.

Keywords

Student performance prediction, Early detection of at-risk students, Recurrent neural network, Knowledge distillation

1. INTRODUCTION

The detection of at-risk students is an essential task to ensure intervention as early as possible. At-risk students are those who may drop out of lecture courses and have low scores (e.g., grade point averages and quiz scores). When potential at-risk students are automatically detected in the early stage of courses, teachers can have sufficient time to encourage them to continue learning.

In recent years, prediction models based on recurrent neural networks (RNNs) have reached high performance [1, 3, 6, 7, 10, 11, 14, 19]. RNNs can handle time-series information such as weekly learning behavior and predict students' performance in each time step. Therefore, RNNs can detect at-risk students in each time step such as after each lecture. However, prediction accuracy in early time steps is lower than that in late time steps because it is difficult for RNNs to extract representative features from only the time-series information in early time steps.

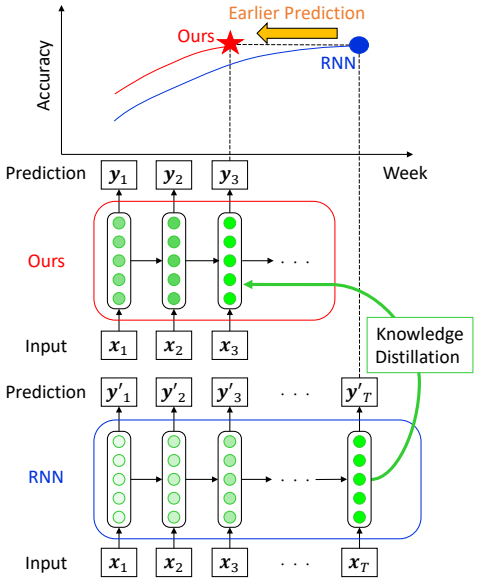


Figure 1: KD for the earlier detection of at-risk students.

To solve this problem, we propose a novel training strategy for improving the prediction in early time steps. Figure 1 shows an overview of our proposed method. Traditional RNNs can extract more representative features in later time steps, and prediction accuracy can also increase because RNNs can use longer time-series information. If RNNs can obtain more representative features from the inputs of earlier time steps, they can detect at-risk students earlier and maintain detection accuracy.

To transfer extracted features, we use *knowledge distillation* (KD) [4]. KD is a compression method for deep neural networks (DNNs), and many methods have been proposed in several fields such as visual recognition [2, 8] and natural language processing [5, 13, 15]. In KD, the model is compressed by training a small DNN model (student model) from a large DNN model (teacher model); that is, the knowledge in the teacher model is distilled to the student model. Further, KD does not require new annotations. In our method, KD is applied to transfer the representative features extracted from

longer time-series information. As shown in Figure 1, this distillation makes the prediction of early time steps closer to that of late time steps, allowing us to detect at-risk students earlier.

The contributions of this study are summarized as follows.

- We introduce KD to predict students' performance. To the best of our knowledge, this is the first study to apply KD to performance prediction.
- We propose the RNN-FitNets model to improve early performance prediction. This model performs as if the learning behaviors in all the time steps are inputted, even though the model only receives the learning behavior in early time steps.
- We evaluate the effectiveness of our model for detecting at-risk students based on the learning logs collected from a higher education course.

2. KNOWLEDGE DISTILLATION RNN MODEL

In this study, we propose RNN-FitNets, which is an integration of RNNs and FitNets [12]. RNN-FitNets distils the well extracted features in the later time step into the RNNs in the earlier time steps by using the architecture of FitNets. Therefore, RNN-FitNets can improve the prediction accuracy in the earlier time steps. For example, as shown in Figure 1, RNN-FitNets can extract representative features in time step 3, whereas traditional RNNs obtain the same feature in time step T.

Figure 2 shows the architecture. The teacher model is pre-trained using all the time steps $(1, 2, ..., T)$, and the student model is trained until time step t $(1 \leq t \leq T)$. During the pre-training of the teacher model and training of the student model, the same ground truth holds (e.g., the final grade is passed to all the time steps). The teacher and student models have the same structure; only the time steps differ between them. Therefore, unlike FitNets, no regressor that transforms the size of the hidden layer of the student model exists.

The student model is trained using two steps in each training epoch as with FitNets. First, it updates its parameter, except for the output layer. Given the t-th time step feature vector of the student model as $\boldsymbol{h}_t$ and T-th time step feature vector of the teacher model as $\boldsymbol{h}'_T$, the parameter is updated by minimizing the following hint loss function L_{HT}:

$$L_{HT} = \frac{1}{2}\|\boldsymbol{h}'_T - \boldsymbol{h}_t\|^2 \quad (1)$$

After updating the parameter, the entire student model, including the output layer, is updated by minimizing the distillation loss. Given the output of the student model as $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t$, T-th output of the teacher model as $\mathbf{y}'_T$, and ground truth as $\mathbf{y}_{\text{true}}$, the distillation loss L_{KD} is calculated as follows:

$$L_{KD} = \sum_{i=1}^{t}(\mathcal{H}(\mathbf{y}_{\text{true}}, \mathbf{y}_i)) + \lambda\mathcal{H}(\mathbf{y}'_T, \mathbf{y}_t). \quad (2)$$

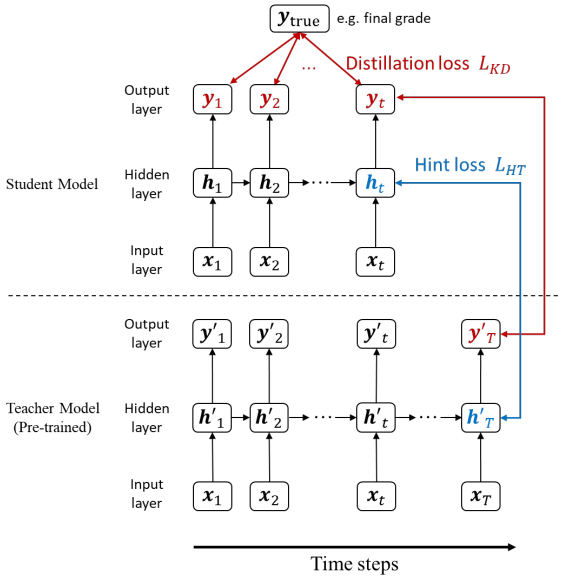


Figure 2: RNN-FitNets.

Table 1: Grade point average distribution.

GPA	A	B	C	D	F
Number of students	25	50	16	12	5

where $\mathcal{H}$ refers to the cross-entropy and λ is a hyperparameter that balances both cross-entropies.

3. EXPERIMENT

3.1 Dataset

We used the same dataset as [10]. The data were collected from the Information Science course at Kyushu University. This course started in April 2016 and 15 lectures were held weekly. Table 1 shows the grade point average of the 108 students that took this course. More than two-thirds of students received an "A" or "B." On this course, the teacher and students used a learning support system called M2B [9]. The M2B system consists of three subsystems: the learning management system, Moodle; the e-portfolio system, Mahara; and the e-book system, BookLooper. Moodle recorded students' attendance, submission of reports, and access to the course. Mahara recorded students' logbook in each lecture on the course. BookLooper recorded students' reading behavior such as turning pages, drawing highlights, and taking notes.

We also applied the feature engineering method used by [10]. The collected data were converted into active learner points, as shown in Table 2. As shown in the table, the learning behavior of each lecture was evaluated on a five-point scale (0–5). Attendance and report submission were evaluated based on whether the activities were on time, late, or not completed. The quiz was evaluated based on the ratio of collected answers. The other behaviors were evaluated by comparing the students in each lecture. Before inputting these features into the prediction model, the evaluated values were divided by 5 (i.e., they were normalized with in the range of 0 to 1).

Table 2: Criteria for active learner points.

Activities	5	4	3	2	1	0
Attendance	Attendance		Being late			Absence
Quiz	Above 80%	Above 60%	Above 40%	Above 20%	Above 10%	Otherwise
Report	Submission		Late			None
Course accesses	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
Word count in Mahara	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
Reading time in BookLooper	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
Highlights in BookLooper	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
Notes in BookLooper	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
Total Actions in BookLooper	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise

3.2 Evaluation Criteria

We applied 5-fold cross-validation to the 108 students in the dataset. The folds were made by preserving the percentage of samples for each student's grade of "A," "B," "C," "D," and "F." After the separation, we grouped grades "A" and "B" into the "no-risk" class and grades "C," "D," and "F" into the "at-risk" class because more than two-thirds of students received "A" or "B" (see Table 1). Therefore, we conducted a binary classification between "no risk" ("A" or "B") and "at-risk" ("C," "D," or "F"). For the evaluation, we calculated the recall, precision, and F-measure values for detecting at-risk students.

3.3 Comparison Models

To investigate the effectiveness of our model, we compared the evaluation values for predicting the final grades between the following three types of models:

- **RNN baseline model**

  Training the RNN-based prediction model using the learning behavior in all lecture weeks.

- **Week-by-week model**

  Training the RNN-based prediction model using the learning behavior in each lecture week. Therefore, there were 15 independent models (trained by only first-week behavior, trained until the second week of behavior, and so on).

- **RNN-FitNets**

  Training the student model from the RNN baseline model as the teacher model in each lecture week. As with the week-by-week model, there were 15 student models.

The three types of comparison models had the same architecture. We set the batch size to one. The length of time steps took an integer from 1 to 15 when the three types of comparison models predicted students at-risk. When the models were trained, the RNN baseline model used 15 time steps and the other models used the same time steps as the prediction. The input features of the model were the active learner points shown in Table 2; therefore, the number of features was nine. For the hidden layer, we used GRU with 32 units and the activation function was tanh. The output layer had two units and the activation function was softmax. We used RMSprop optimizer [16] for the hint loss and distillation loss. In both the optimizations, we set the learning rate to 0.001. In addition, we applied L2 regularization with a parameter of 0.004 for the optimization of the weights and biases in the hidden and output layers. λ in the distillation loss (Eq. (2)) was equal to the time step; for example, when RNN-FitNets was trained using the learning behavior until the second week, we set λ to 2. The aim was to make the second term in Eq. (2) the same scale as the first term (i.e., RNN-FitNets is equally affected by the teacher model and ground truth). All models were trained for 50 epochs.

3.4 Experimental Result

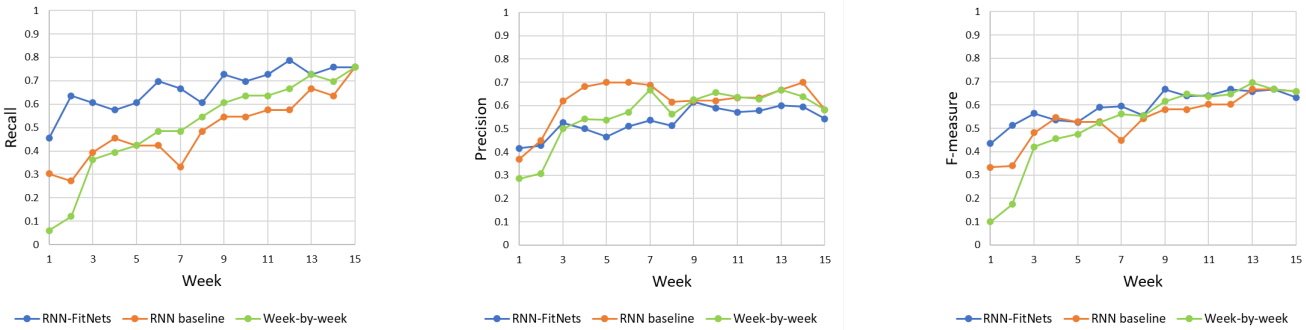
Figure 3 illustrates the evaluation of the three types of models. We summarize the results as follows:

- In most time steps, the recall values of the RNN-FitNets were higher than the values of the RNN baseline and week-by-week models. In other words, RNN-FitNets detected more at-risk students than other models.

- However, the precision values of the RNN-FitNets were lower than the value of the RNN baseline model, i.e., RNN-FitNets misdetected more no-risk students as at-risk.

- As shown by the F-measure values, the RNN-FitNets' values were higher than that of the RNN baseline and week-by-week models in most time steps. This difference was marked in early time steps. Therefore, the increase in the detection of at-risk students outweighed the increase in the misdetection, especially in early time steps.

- Comparing the evaluation values of the RNN baseline model with those of the week-by-week model, the former was superior in early time steps, although the values of the week-by-week model were close to or outperformed those of the RNN baseline model in late time steps.



(a) Recall (b) Precision (c) F-measure

Figure 3: Evaluation of three types of models.

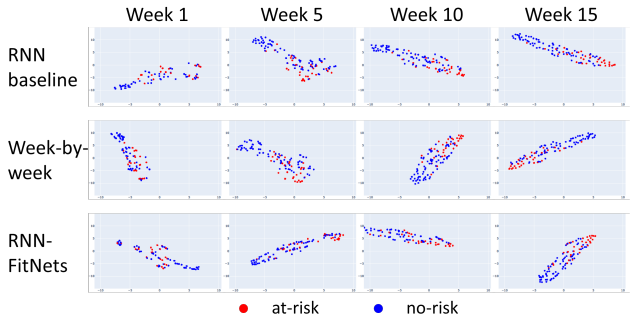


Figure 4: Visualization of the extracted feature vectors in the three types of models by t-SNE.

3.5 Discussion

The experimental result showed that the proposed models improved the detection rate of at-risk students, especially in early time steps. This improvement resulted from the distillation of time-series information. The evaluation values of the RNN baseline model were higher than those of the week-by-week model in early time steps. This result implies that the time-series information obtained by training in all the time steps is effective for early detection. In the RNN-FitNets, the time-series information was expressly passed through KD and that improved the model's performance.

To investigate whether the time-series information was distilled into the models in early time steps, we visualized the extracted feature of the three types of models. Figure 4 shows the visualization results of the feature vectors. Because the models have a 32-dimensional hidden state, we used t-SNE [17] and reduced the 32 dimensions to two dimensions for the visualization. Each point represents each feature vector for the students in the dataset. The red point is at-risk students and the blue point is no-risk students, as defined in Section 3.2. By observing the feature vectors of the RNN baseline and week-by-week models, the more time steps are used, the closer the red points are to each other and the more the shape of the mass of points becomes elongated. This means that the detection of at-risk students becomes easier in the feature vectors of late time steps. In the RNN-FitNets models, the tendency to gather red points and elongate appears in early time steps. This result shows that our KD method properly distills the time-series information extracted in the late time step.

4. CONCLUSION

In this study, we proposed RNN-FitNets, which extends FitNets, a KD method, for application to RNN architecture. RNN-FitNets transfers the time-series information extracted by the later time-step RNN into an earlier time-step RNN. Hence, the earlier time-step RNN learns the method of extracting the representative features in late time steps from short time-series data.

In the experiment, we applied RNN-FitNets to detect at-risk students in higher education. The results show that the proposed distillation model improves the detection rate of at-risk students from the base RNN models. The analysis of feature vectors indicated that our proposed model in earlier time steps extracted similar feature vectors to those of the base model in late time steps. This confirmed that our distillation strategy properly distilled the time-series information in later time steps into the model in earlier time steps.

In future work, we plan to investigate the availability of RNN-FitNets for other datasets. Moreover, we aim to formulate a new distillation method for time-series information for other models such as the Transformer model [18].

5. ACKNOWLEDGMENTS

This work was supported by JST AIP Grant Number JPMJCR19U1, and JSPS KAKENHI Grand Number JP18H04125, Japan

6. REFERENCES

[1] N. R. Aljohani, A. Fayoumi, and S.-U. Hassan. Predicting at-risk students using clickstream data in the virtual learning environment. *Sustainability*, 11(24):7238, 2019.

[2] W. Chen, C.-C. Chang, C.-Y. Lu, and C.-R. Lee. Knowledge distillation with feature maps for image classification. In *ACCV*, 2018.

[3] Y. He, R. Chen, X. Li, C. Hao, S. Liu, G. Zhang, and B. Jiang. Online at-risk student identification using rnn-gru joint neural networks. *Information*, 11(10):474, 2020.

[4] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[5] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[6] B.-H. Kim, E. Vizitei, and V. Ganapathi. Gritnet 2: Real-time student performance prediction with domain adaptation. *arXiv preprint arXiv:1809.06686*, pages 1–8, 2018.

[7] B.-H. Kim, E. Vizitei, and V. Ganapathi. Gritnet: Student performance prediction with deep learning. *arXiv preprint arXiv:1804.07405*, 2018.

[8] Q. Li, S. Jin, and J. Yan. Mimicking very efficient network for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7341–7349, 2017.

[9] H. Ogata, Y. Taniguchi, D. Suehiro, A. Shimada, M. Oi, F. Okubo, M. Yamada, and K. Kojima. M2b system: A digital learning platform for traditional classrooms in university. *Practitioner Track Proceedings*, pages 155–162, 2017.

[10] F. Okubo, A. Shimada, T. Yamashita, and H. Ogata. A neural network approach for students' performance prediction. In *LAK 2017 Conference Proceedings - 7th International Learning Analytics and Knowledge Conference*, ACM International Conference Proceeding Series, pages 598–599. Association for Computing Machinery, Mar. 2017. 7th International Conference on Learning Analytics and Knowledge, LAK 2017 ; Conference date: 13-03-2017 Through 17-03-2017.

[11] F. Okubo, T. Yamashita, A. Shimada, Y. Taniguchi, and K. Shin'ichi. On the prediction of students' quiz score by recurrent neural network. *CEUR Workshop Proceedings*, 2163, Jan. 2018. 2nd Multimodal Learning Analytics Across (Physical and Digital) Spaces, CrossMMLA 2018 ; Conference date: 06-03-2018.

[12] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[14] D. Sun, Y. Mao, J. Du, P. Xu, Q. Zheng, and H. Sun. Deep learning for dropout prediction in moocs. In *2019 Eighth International Conference on Educational Innovation through Technology (EITT)*, pages 87–90, 2019.

[15] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*, 2019.

[16] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[17] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[19] F. Xiong, K. Zou, Z. Liu, and H. Wang. Predicting learning status in moocs using lstm. In *Proceedings of the ACM Turing Celebration Conference-China*, pages 1–5, 2019.