Deep Knowledge Tracing: A Review

Zainab Imam Attahiru
St Catharine's College
zia21@cam.ac.uk

Abstract

Intelligent tutoring research has seen the application of myriad techniques to develop adaptive learning platforms that motivate learners. Knowledge Tracing has been one of those techniques and recent developments have resulted in a rising interest in its application. This paper will focus on reviewing and analysing the first knowledge tracing model that applied deep learning techniques - Deep Knowledge Tracing. This direction is motivated by the author's interest in the field and the possibility of extrapolating more insights on knowledge tracing as a whole.

1 Introduction

The COVID-19 pandemic has initiated a mandatory increase in the use of digital platforms to deliver learning content across institutional and geographical boundaries. This has highlighted both the potential and drawbacks in the state of the current tools in circulation. A prominent shortcoming is the adequacy of these platforms as learning environments especially at the primary and secondary school levels. This is not a recent problem but has been one of the key challenges in developing intelligent tutoring systems. A common remedy has been to incorporate personalisation into these systems by adapting content presentations to the needs (or preferences) of the learner.

To efficiently infer these personal learning modalities, an understanding of the student's knowledge state is required. This leads us to one of the important ideas in building the foundations of intelligent tutoring systems – knowledge tracing. Knowledge Tracing attempts to model a student's knowledge state as he or she interacts with coursework. Several methodologies have been deployed to solve this problem, from probabilistic techniques ([15], [2]), logistic models ([3]) and deep learning architectures ([1], [6], [16]).

This paper examines the most popular application of deep learning techniques to knowledge tracing, Deep Knowledge Tracing. The goal is to determine which parameters and factors influence the accuracy of this model and its potential to be scaled to other learning scenarios such as distributed learning. The initial scope of this paper sought to demonstrate the efficiency of DKT using the Flower federated learning framework. However implementation bottlenecks have hindered the accomplishment of this task. In the current scope, the hope is to provide some insight into DKT through benchmarking analysis and derive some conclusions and commentary on future work.

2 Background

In this section, the background that laid the foundations for the current state of knowledge tracing will be explored together with current related work. These works are driven by rigorous research in cognitive science, education, neuroscience, and psychology.

2.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) is a cognitive model that captures the knowledge state of a student while they are learning in a skill. BKT represents the knowledge state of the student as latent

variables in the form of a Hidden Markov Model, and the observed variables as the sequence of problem responses. The change in knowledge state is determined by computing the probabilities of transitioning to from one state to another, guessing a response, and making a mistake.

There are some shortcomings in the standard BKT that have hindered its application to learning environments. First, it assumes no forgetting – once mastery is achieved, the learning curve stabilises. Research in education regarding spaced repetitions [13] have proven that this is not the case. Additionally, both latent and observed variables are represented in binary form, therefore ignoring complex representations of knowledge states and problems respectively.

However, there have been other variations that have been developed to resolve some of these problems, including the Dynamic BKT, which allows different skills to be represented in a single model [2]. Other work in this field include the development of a framework to develop BKT models [12].

2.2 Logistic Models

These models predict how well a student will perform by defining the relationship between the student and the knowledge concept as a logistic function. The earliest form, Learning Factors Analysis (LFA) captures the difficulty of learning a particular knowledge concept and the benefit of engaging in prior practice. LFAs are traditionally used in data mining scenarios. A extended version of the LFA, Performance Factors Analysis reconfigures the LFA to allow for adaptive selection of practice questions [3]. Models based on variations of logistic regression have also been employed in language learning [4].

2.3 Deep Knowledge Tracing

Introduced in 2015 [1], Deep Knowledge Tracing (DKT) is the first application of deep learning models to knowledge tracing. Using Recurrent Neural Networks (RNN), DKT maps the students' coursework interaction sequence to a prediction of how well they will perform in the next task in that sequence. The use of RNNs for this task is grounded in its ability to represent relevant encodings of past observations in its hidden states that could be useful in future predictions, thus allowing them to retain long range dependencies. These encodings are retained using gates that control the update mechanism of the network.

RNNs have been useful in other applications such as language translation, and music generation. Figure 1 shows a simple network definition of an RNN for student learning;

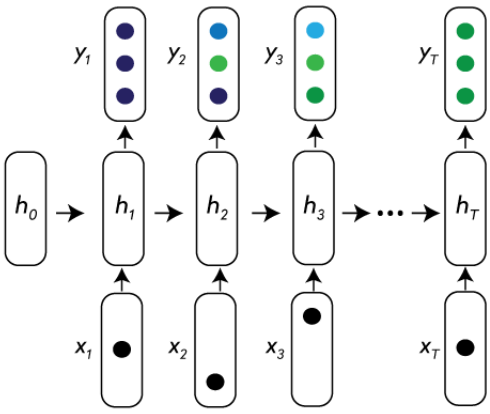

Figure 1: The student interaction represented by a vector x and the vector y represents the probability of performing well on the next task. Adapted from [1]

Most implementations of DKT use a special kind of RNNs called Long Short-Term Memory networks. LSTMs are more accurate than other RNNs as they deal with the vanishing gradients problem using an additional gate, the forget gate. The data in DKT are typically represented in as one-hot encodings of

the student interaction, denoted by $x_t = \{q_t, a_t\}$ where q_t is the question and a_t is the corresponding answer. The relationship between the input (x_t), output (y_t), and hidden states(h_t) of the LSTM are defined by the equations below;

$$h_t = tanh(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = \sigma(W_{yh}h_t + b_y)$$

Leveraging on the architecture they are built on, DKT have exhibited promising results in knowledge tracing. Their structure allows for more complex representations of student knowledge and has been applied to open-ended learning tasks such as programming [14]. Additionally, the need for expert annotations is eliminated, allowing for easier data pre-processing before training begins. However, there have been some criticisms regarding the interpretability of these models [5] due to the difficulty of figuring out the knowledge representation in the hidden states.

2.4 Other Deep Learning Approaches to KT

A variety of deep learning-based solutions to knowledge tracing have been developed to deal with the shortcoming in preceding models such as BKT and DKT. Models such as Dynamic Key Value Memory Network (DKVM) deal with issues regarding the interpreting the knowledge state. DVMN stores knowledge concepts in a memory module and then, applies an erase and add update mechanism [6]. However, it fails to handle long range dependencies.

Another example is the use of transformer models which have been proven to give insights into knowledge state transitioning is the self-attentive neural knowledge tracing (SAINT). SAINT uses an encoder and decoder structure, thus separating the encoding and decoding of questions and answers using self-attention layers [7]. The separation of the questions and answers was assumed to have a degrading effect on training [1], yet SAINT has shown superior performance on EdNet, one of the largest education-based datasets.

Aside from these, other models have focused on understanding the semantics of content representation using bidirectional LSTMS [8] and using graphical neural networks to represent the relationship between knowledge concepts [10].

3 Implementation

The implementation in this paper is used for the purpose of performing benchmarking analysis on the model components and variables that affect the prediction accuracy of DKT. Google Colab was used as the working environment and notes have been provided within the notebook to serve as brief explanations on DKT within the context of the codebase.

3.1 Objectives

The core aim of running this experiment is to gain some insight on the model variables and design factors that might affect the prediction accuracy of DKT models. For this aim, the analysis focuses on these three objectives

- Determine the influence of batch sizes and epoch count on model accuracy
- Observe the effects of different dataset sizes on training
- Determine the effect of the number of hidden layers on training

These objectives are motivated by my interest in the application of knowledge tracing to distributed learning environments.

3.2 Model and Dataset

The DKT model used for the benchmarking analysis in this paper uses a recurrent neural network with a tanh activation function and sigmoid function for the gated units. This conforms to the standard implementation of DKTs as seen in the literature cited in the preceding section as well as code samples that were explored during the research for this paper. The ASSISTments 2009-2010 dataset was used with a 70:30 split for the training and test set. The code repository can be found here here

4 Results

4.1 Mini-batching and Epochs

Model prediction appears to remain within the same range despite changes in epochs and batch sizes. The entire dataset was used for this analysis and no other hyper-parameters were modified. The goal was to determine if model accuracy is affected by batch sizes or the number of training epochs. The training speed is much slower for smaller batch sizes of course but the difference is almost negligible.

Table 1: Area Under the Curve (AUC) variations for training using different epochs and batch sizes

	32 batch size	64 batch size
1 epoch	0.841	0.839
5 epochs	0.838	0.842
10 epochs	0.854	0.843
15 epochs	0.845	0.843

It is important to note that this was carried out using the standard network hyper-parameters and no changes in size were applied to the database. As this analysis is constrained by the dataset size (interaction sequences of 10,217 students) compared to the largest education database, EdNet (millions of interactions), it cannot be conclusively confirmed that mini-batching and epochs don't affect model accuracy in DKT. However, for the sake of the goal of the analysis which is directed towards the application of the model in low computational power environments, the insight from this portion of the experiment suffices.

4.2 Learning Rate and Layers

Increasing and decreasing the number of the hidden layers in the RNN seemed to have no effect on training. This further expounds the initial conclusion by other researchers that the hidden layers in the DKT model provide negligible interpretability regarding the presentation and extrapolation of student knowledge. This could mean that less complex representations would work relatively well on moderate datasets - a conclusion that would be favourable for the deploying on edge devices.

4.3 Dataset Effect

Changing the size of the data showed the most pronounced effect. Training the data on 64 batches (of 64 batch sizes) rather than the original 160 batches led to a decrease in model accuracy from **0.845** to **0.642**. This poses a significant challenge for later applications that attempt to train DKT using less data.

5 Review and Future Work

The analysis conducted in this work is in no way groundbreaking but was rather intended to be an exploratory experiment into the workings of DKT. Further work in this area will be conducted regarding all components that were placed under analysis - batching, model design, and data.

On batching, efforts will be concentrated on deploying the model on Flower, a popular federated learning framework, to properly simulate a heterogenous learning environment. Current state-of-the-art models are trained using lots of data while assuming the homogeneity of the learners. Similar to how research in federated learning examines the possibility of heterogeneity in the dataset due to geographical and cultural backgrounds, learners come from equally varying cultures with differing societal approaches to learning. Knowledge tracing paradigms need to adopt techniques that take this into account.

Model design could also be improved through the use of quantization techniques to reduce computational complexity. Although model compression in recurrent neural networks have not been as notable as those undertaken in convolutional neural networks, there are still work that has been done regarding reducing the matrix multiplication at the hidden layers. Additionally, reviewing

and exploring other knowledge tracing models based on deep learning architectures and performing a rigorous comparative analysis based on the questions posited in the objectives sections of the implementation is required.

While the ASSISTments data has been the popular choice for deep learning analysis of KT models, its representation of problem sequences as binary interactions provides an insufficient representation of learning data in general. EdNET has a larger dataset and more complex interactions of students with learning. Other datasets worthy of reviewing include KT1 - KT5 that have more complex representations of exercises undertaken by students, thus providing a diverse and more challenging learning problem for knowledge tracing models.

6 Conclusion

Work undertaken in the field of knowledge tracing presents exciting prospects for the future of learning and adaptive learning systems. However, the review provided in this paper gives only a cursory overview and shallow analysis of Deep Knowledge Tracing. This is largely due to the implementation bottlenecks in deploying the initial scope of the project.

Nonetheless there are interesting prospects to consider for future research and analysis. Although those highlighted in this paper are strongly correlated with distributed learning frameworks and incorporating heterogeneity into current models, other tangents such as the incorporating of active learning to deal with data scarcity could also be explored. A recent paper on federated knowledge tracing goes in-depth and extensively discusses some of the points that were raised here. [9]

7 References

References

[1] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. & Sohl-Dickstein, J. Deep Knowledge Tracing. *Advances In Neural Information Processing Systems*. **28** (2015), https://proceedings.neurips.cc/paper/2015/file/bac9162b47c56fc8a4d2a519803d51b3-Paper.pdf

[2] Kaser, T., Klingler, S., Schwing, A. and Gross, M. Dynamic Bayesian Networks for Student Modeling. *IEEE Transactions On Learning Technologies*. **10** pp. 450-462 (2017)

[3] Pavlik Jr, P., Cen, H. and Koedinger, K. Performance Factors Analysis - A New Alternative to Knowledge Tracing. *Frontiers In Artificial Intelligence And Applications*. **200** pp. 531-538 (2009,1)

[4] Settles, B. and Meeder, B. A Trainable Spaced Repetition Model for Language Learning. *ACL*. (2016)

[5] Khajah, M., Lindsey, R. and Mozer, M. How Deep is Knowledge Tracing?. *ArXiv*. **abs/1604.02416** (2016)

[6] Zhang, J., Shi, X., King, I. and Yeung, D. Dynamic Key-Value Memory Networks for Knowledge Tracing. *Proceedings Of The 26th International Conference On World Wide Web*. pp. 765-774 (2017), https://doi.org/10.1145/3038912.3052580

[7] Choi, Y., Lee, Y., Cho, J., Baek, J., Kim, B., Cha, Y., Shin, D., Bae, C. and Heo, J. Towards an Appropriate Query, Key, and Value Computation for Knowledge Tracing. *Proceedings Of The Seventh ACM Conference On Learning @ Scale*. pp. 341-344 (2020), https://doi.org/10.1145/3386527.3405945

[8] Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y. and Hu, G. EKT: Exercise-Aware Knowledge Tracing for Student Performance Prediction. *IEEE Transactions On Knowledge And Data Engineering*. **33**, 100-115 (2021)

[9] Wu, J., Huang, Z., Liu, Q., Lian, D., Wang, H., Chen, E., Ma, H. and Wang, S. Federated Deep Knowledge Tracing. *Proceedings Of The 14th ACM International Conference On Web Search And Data Mining*. pp. 662-670 (2021), https://doi.org/10.1145/3437963.3441747

[10] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Yu, P. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions On Neural Networks And Learning Systems*. **32** pp. 4-24 (2019)

[11] Mandalapu, V., Gong, J. and Chen, L. Do we need to go Deep? Knowledge Tracing with Big Data. *ArXiv*. **abs/2101.08349** (2021)

[12] Badrinath, A., Wang, F. and Pardos, Z. pyBKT: An Accessible Python Library of Bayesian Knowledge Tracing Models. *ArXiv*. **abs/2105.00385** (2021)

[13] Tabibian, B., Upadhyay, U., De, A., Zarezade, A., Schölkopf, B. and Gomez-Rodriguez, M. Enhancing human learning via spaced repetition optimization. *Proceedings Of The National Academy Of Sciences*. **116**, 3988-3993 (2019), https://www.pnas.org/content/116/10/3988

[14] Wang, L., Sy, A., Liu, L. and Piech, C. Deep Knowledge Tracing On Programming Exercises. *Proceedings Of The Fourth (2017) ACM Conference On Learning @ Scale*. pp. 201-204 (2017), https://doi.org/10.1145/3051457.3053985

[15] Corbett, A. and Anderson, J. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling And User-Adapted Interaction*. **4**, 253-278 (1994), https://doi.org/10.1007/BF01099821

[16] Pandey, S. and Karypis, G. A Self-Attentive model for Knowledge Tracing. (2019)