Deep Knowledge Tracing for Student Performance

Chris Wang, chrwang@stanford.edu

Sherry Ruan, Alex Kolchinski

Stanford University, CS 230: Deep Learning

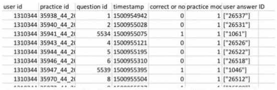
Introduction

- Being able to model student knowledge and ability based upon performance on assessments can help make predictions regarding student performance and develop curriculum that adapts to each individual student, creating interesting new innovations in the education space. Piech et al. introduces a deep learning knowledge (DKT) model that applies recurrent neural networks and long short-term memory.

- Our objective is to build upon this model applied to a different data set attempting to take into account additional question information. We apply our improved model to a new data set of student GMAT assessment data and analyze its performance. Performance will be assessed by how well our model can predict whether a student will answer a given question correctly or incorrectly.

Data Set/Features

- Our data set consisted of over 8 million data points, each of which included a GMAT question answered by a specific student and characteristics of that question, whether that question was answered correctly or incorrectly, and the timestamp at which the question was answered. The data set was provided to us from TAL education from their GMAT preparation software.



user id	practice id	question id	timestamp	correct or no	practice moc	user answer ID
1310344	35938_44_2(	1	1500954942	0	1	["26537"]
1310344	35940_44_2(	2	1500955028	0	1	["26531"]
1310344	35941_44_2(	5534	1500955075	1	1	["1061"]
1310344	35943_44_2(	4	1500955121	0	1	["26526"]
1310344	35944_44_2(	5	1500955195	0	1	["26522"]
1310344	35946_44_2(	6	1500955310	0	1	["26518"]
1310344	35947_44_2(	5539	1500955395	1	1	["1046"]
1310344	35970_44_2(	8	1500955504	0	1	["26512"]

Figure 1: Example of data points (these are all for one student). Further details about each question are also available including the topics covered by each question.

- The data set is then processed into a set of sequences, one for each student, consisting of all of the questions that student answered (along with an encoding of each question's topics), in order, as well as whether the question as answered correctly. This is the input to the DKT model. We had a few thousand students each answering up to a few thousand questions each.

Model and Techniques

- The model used was a DKT model using an RNN with LSTM. The processed question data was encoded into a question id (which may have been based on question characteristics). The model takes an input sequence of time series vectors and maps them to an output sequence of time series vectors through hidden states which store information from the preceding data. The latent units store values until cleared by a 'forget' gate in an LSTM.

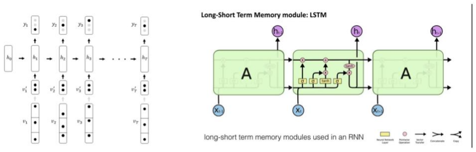
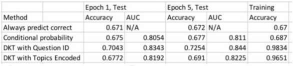


Figure 2: LSTM model at different time steps, left shows connection between variables in a simple RNN with input vectors at each time step encoded. The right figure shows the specific parameters at a single step of the LSTM. (second image source: colah.github.io/posts/2015-08-Understanding-LSTMs/)

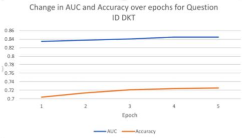
Experimental Results

- The model was implemented using Tensorflow and Python 3.6 on Jupyter Notebook and run on AWS's GPU.

- The model was run with a baseline conditional probability model, the DKT model with question ID only (all questions are assumed to be distinct, using no characteristic about the question), and the DKT model with question topics as well. When using the model with topics, since not all of the questions had topics, many of the questions were omitted. The test set had 25% of the 8 million data points (although they were truncated to prevent tensors from becoming too large. The remaining 75% were in the training set.



	Epoch 1, Test		Epoch 5, Test		Training
Method	Accuracy	AUC	Accuracy	AUC	Accuracy
Always predict correct	0.671	N/A	0.672	N/A	0.67
Conditional probability	0.675	0.8054	0.677	0.811	0.687
DKT with Question ID	0.7043	0.8343	0.7254	0.844	0.9834
DKT with Topics Encoded	0.6772	0.8192	0.691	0.8225	0.9651



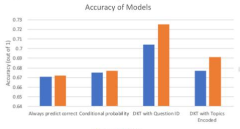


Figure 3: Top left table shows the accuracy and AUC of the model for various methods used after 1 and 5 epochs. Bottom left plots AUC and Accuracy change for most accurate model (on test set) over 5 epochs. The top graph compares the data in the table to its left for the test set.

Discussion

- A major challenge encountered throughout this project was the sparsity of questions. In other words, there were a large amount of data points, but a lot of them were on different questions. This made our tensors very sparse. A potential solution was using the topic data, since each question had multiple topics, many of which overlapped. However, this actually resulted in the opposite effect because it reduced the number of data points we had significantly.

- Overall, the results we were able to achieve had slightly higher than a baseline conditional probability model. This was not as high as anticipated, but due to the factors of data sparsity and missing topics, seemed to be as high as possible given our data set. There is certainly lots of inherent unpredictability in determining whether a student will answer a question correctly.

Future Work

- A more complete data set with data labels and a way to encode them into vectors (which would necessitate a new architecture for input into our model) may yield more accurate results. Natural language processing can be used for the questions and answers to better gauge question characteristics without needing labels.

- Beyond improvements to the model itself, the results from such a model can be applied to help create adaptive assessments for students.

References

- I would like to thank Sherry Ruan and Alex Kolchinski for being my mentors throughout the course of this project.

- C. Piech, et. al. *Deep Knowledge Tracing*, 2015 https://web.stanford.edu/~cpiech/bio/papers/deepKnowledgeTracing.pdf