

INFO C260F

Homework: RNN Modeling of behavior and performance

Vinitra Swamy, Madeline Wu, Wilton Wu

Link to github repository: <https://github.com/vinitra/RNNModelingForEdu>

All of the questions are answered in Markdown in the iPython notebook but detailed here for convenience.

1. Data Explorations for Skills

- a. The five most common skills were skill 87: Equation Solving Two or Fewer Steps, skill 30: Conversion of Fraction Decimals Percents, skill 71: Addition and Subtraction Fractions, skill 68: Addition and Subtraction Integers, and skill 33: Ordering Fractions.
- b. The five least common skills were skill 28: Reading a Ruler or Scale, skill 102: Recognize Quadratic Pattern, skill 98: Finding Slope from Ordered Pairs, skill 96: Finding Slope From Situation, and skill 99: Distributive Property.
- c. The most common skill was skill 87: Equation Solving Two or Fewer Steps with more than 5.77% of the responses associated with this skill.

2. Sequence Prediction Model: 70-30 Split

- a. The overall accuracy of skill prediction is 83.2%.
- b. We defined easiest to identify skills as the skills that had the highest proportion of accurate prediction. We decided not to use mere accurate prediction as a metric because it is unfair to skills that don't appear as often in the dataset. We defined hardest to identify skills analogously -- those that had the highest proportion in the cases of inaccurate prediction.
- c. Our easiest to identify skills are: 2, 69, 31, 3, and 47.
- d. Our hardest to identify skills are: 82, 79, 76, 89, 74.

3. Hyperparameter tuning for Sequence Prediction Model

- a. The overall accuracy of our skill prediction with changed hyperparameters is $100\% - 17.05\% = 82.95\%$. The hyperparameters we changed from our base model were increasing the batch size to 128 and increasing the epochs to 200.
- b. We tried many different permutations of different hyperparameters including adding more hidden layers, using different optimizers (RMSprop, SGD), increasing the batch size, and increasing the epoch size.
- c. Surprisingly, many of the permutations of different hyperparameters actually decreased the accuracy of our predications. We thought that adding more hidden layers would help increase our accuracy rates, but instead it took longer to train and never reached an accuracy of 65% on the validation data set.

4. DKT Model: 70-30 Split

- a. We got an accuracy of 93.0% and an AUC of 0.5989.

5. Hyperparameter tuning for DKT

- a. We tuned the hyperparameters by increasing epochs to 30 and batch size to 38. This resulted in an accuracy of 93.16% and an AUC of 0.6067.
- b. We increased the number of epochs because the longer we train the model, the better the model is able to understand the underlying patterns. We increased the batch size because in the input, the model is able to find relationships between different students more effectively.