INFO C260F
Homework 2
Getting started with prediction (part 1)
Vinitra Swamy, Madeline Wu, Wilton Wu
Team Name: “Four-layer Dip”

Link to github repository: https://github.com/vinitra/CognitiveTutorPrediction

In this assignment, we were given the goal of getting familiar with response data. Our task was to generate a model that would make predictions on a student’s N^{th} response, given his or her 1^{st} to $N\text{-}1^{th}$ responses. We were given data in two formats—a one row per response format, which included different features of the response and a one row per student format, which only included ordered responses for five questions.

For our approach, we wanted a challenge. We first tried to use the Bayesian Knowledge Tracing algorithm with the BKT package. Although it looked promising, we ran into a couple of installation issues, specifically with the Boost-Python library, so we eventually switched to Keras and tried to create a simple 4-layer Sequential Neural Network. After looking at the given data and the two data formats, we decided to simply use the second data set with the first five responses of each student to predict the sixth response. Given the data, we divided the students into training and testing sets—25 students for training and 5 for testing. Our input layer consisted of 25 nodes, one for each student, where the input vector is five-dimensional. Each input vector represents a student’s response to each of the five questions. Our output layer also consisted of 25 nodes, one for each student. Each output node predicts the response to the next question the student would be asked, which in this case is the 6^{th} question, therefore each output vector is a single value. If we were to extend this model to any N number of questions, only the dimensions of the input vector would need to be changed.

During our implementation and testing, it took us a while to get acclimated with Keras at first. A common error we ran into was our model returning a prediction of all 1’s (our output probabilities were all above 0.5). We spent time playing around with different number of layers, different activation functions, epochs, and batchsizes to try to overfit our data. In the end our overall accuracy, with 25 data points in our training set and 5 data points in our test set, was 100%. This may be because the last 5 student responses to the sixth question were all 1’s.