Jared’s contributions to the DEEP LEARNING final project thus far:

**DATA CLEANING**

What We’ve Done to This Point:

* We have data for 21 trading days (all of March 2018) for 10 stocks, stored in 210 individual text files.
* The 10 tickers for which we have data are:
  + AAOI (Applied Optoelectronics Inc.)
  + BABY (Natus Medical Inc)
  + CA (CA Technologies – has since been acquired by another company, but was trading during March 2018)
  + DAIO (Data I/O Corporation)
  + EA (Electronic Arts Inc)
  + FANG (Diamondback Energy Inc)
  + GABC (German American Bancorp Inc)
  + HA (Hawaiian Holding Inc)
  + IAC (InterActiveCorp)
  + JACK (Jack in the Box Inc)
* The data is from Nasdaq’s ITCH feed, and has far more than just executed trade messages: it also contains all bids, asks, cancels, and more. Therefore, a fair amount of data cleaning was necessary to pare down these files to only executed trade messages. Jared wrote a Python script that does this for all 210 text files, turning each of them into a CSV file. Additionally, data for each ticker was aggregated across trading days, giving 10 larger datasets (one for each ticker, containing all the trades for that ticker across the 21 days).
* In the end, we have just over 616,000 observations across all 210 of the individual datasets. When aggregating within tickers and across trading days there is significant variability in the volume of trades for the different stocks. Near the bottom are the DAIO and GABC stocks, each stock having roughly 3,000 trades across the 21 trading days. On the other hand, EA had over 180,000 trades over the same 21 days.
* Also, Jared consulted with Dr.’s Moon and Brough to reason through how we should construct our training and test sets. At this point, our initial approach will be to start on much smaller datasets (maybe just the data for a single stock on a trading day), and scale up. Upon scaling up we might consider a few things: training on a collection of tickers’ data and predicting onto the collection of the remaining tickers’ data (i.e. aggregate all of the data for the first 7 tickers and predict onto all of the data for the last 3 tickers). We might also train within a trading day across all 10 tickers and then predict onto a different trading day. These approaches will be informed by experimentation, as well as by a SME (Dr. Todd Griffith in the Finance dept, see below). If there are important intra-day dynamics or across-days dynamics this may change whether we’re aggregating across days or within days.

Work Still to Do:

* Add the “trading day” as a feature for the aggregated data sets. Otherwise, the timestamps will appear as though all trades are coming from the same day, when in reality they’re coming over the course of an entire month. There will potentially be valuable information coming from the day the trades occurred. Another thing we might do: the timestamps are given at the time of trade execution, counting the number of nanoseconds since midnight of that trading day. If we didn’t want to add ‘trading day’ as a feature, we could just add 8.46e+13 (the number of nanoseconds in a day) when aggregating across trading days in order to preserve the chronology of the data. However, I think the option of adding ‘trading day’ is easier and more reasonable for both modeling purposes and practical purposes (writing the code, being able to easily see when a trade occurred).
* Also, from Kegan’s reading it sounds like there is very non-trivial work that needs to be done for pre-processing data such that it can be fed through an LSTM for training and testing.

**PROBLEM CONSULTING**

What We’ve Done to This Point:

* Emailed Dr. Griffith to get his thoughts/advice on things specific to market microstructure dynamics. He has responded, but not yet replied for a specific meeting time. Jared will be meeting with him to ask questions relating to how we can translate qualitative aspects of microstructure data (ITCH data) into quantitative modeling decisions. Particularly, is there some canonical notion as to a time period or number of trades preceding a “trade of interest” that gives us a sufficient amount of information to predict that trade of interest? This will play into how long our input sequence (number of preceding trades) will be when training the RNNs.
* Talked with Dr. Moon regarding how we ought to split training data, as well as how we might want to train the non-neural-network models in a “real-world” manner. Particularly, in an applied setting we wouldn’t be able to predict a trade at current time T with trades at time T+1, T+2,…, but we can do this in the analysis of past data. If the objective is purely academic and our goal is to only predict trades accurately with no priority given to the “real-time” application this changes the dynamic and framing of the problem significantly. In particular, we’d want to train our other binary classifiers (k-nn, forests, GBM) in a way that is analogous to how we’d train the RNN. In the unrestricted setting this would be straightforward, but it would be much hairier in the restricted (‘real-time’) setting. In that setting, we’d have to find a way to restrict the training data for each trade we’re trying to classify such that there are no future observations upon which we’re training. We could also look into already-existing application of these other classifiers that take time dependency into account in their respective algorithms. However, some cursory searching shows that these methods aren’t very well-developed, so we may have some trouble generating analogous models for good comparison in the ‘real-time’ setting.

Work Still to Do:

* Ask Dr. Brough and/or Dr. Griffith what the ultimate goal is: building a model designed to do ‘real-time’ prediction in an applied setting, or building a model designed to classify past trades where training data aren’t restricted based on time?
* Ask Dr. Brough and/or Dr. Griffith if there’s a relevant time scale in which we’d consider the response for the next trade as being well-described by the information in the previous T trades. This will help us to determine if our input sequences will be a fixed number of immediately preceding trades and their responses, or whether the input sequences will be the trades that took place in some fixed time period immediately preceding the trade of interest.