Title: Final Project
Name: Michael Morrison
Format: pdf

Introduction

Baseball is one of the only team sports that relies so heavily on a single interaction between one player from each team, the pitcher and hitter at each at bat. In the world of the MLB, the pitcher reigns supreme, with even the most skilled batters only getting a hit in 1/3 of their plate appearances.

If the batter were to know what pitch they were going to see next, this would be a massive advantage for them. A real-life example of this occurring can be tracked to the 2017 Houston Astros playoff run. During their home games, the Astros would use drums to indicate to the batter at the plate to indicate what pitch was coming next. This is a practice known as stealing signs, as they would intercept the calls between the dugout and the pitcher. Doing this lead Astros players to have an incredibly high playoff batting average while at home, and this helped them eventually win the world series.

Although this is an example of foul play, the story shows knowing pitch type is incredibly advantages. For this project, I decided to create a Recurrent neural network that has the ability to predict pitch type based upon all of the pitcher's previous pitches.

Analysis

The data used for this project was in the form of two datasets from MLB pitch data from the years 2015-2018. The first dataset is "atbats.csv" and gives the data for a single at bat, including the outcome. This dataset includes the outcome of the at bat, as well as important data about the game at that current time. "pitches.csv" breaks down the unique at bats by pitch. This includes pitch metrics like speed and location. Each game, at bat, pitch, team, and player have a unique id that makes the timeline of each pitch trackable across the dataset. There is also an extra dataset "player_names.csv" that allows us to pull actual player names from the ids provided in the other datasets.

There is a mix of float, integer, and string datatypes for the values in all the datasets. Floats are used for pitch metrics, integers for inning number and pitch counts, and strings for names and outcomes.

The dataset has a massive size, with the "pitches.csv" being the largest and having close to 2 million values.

Original dataset: https://www.kaggle.com/datasets/pschale/mlb-pitch-data-20152018

To prepare the data for the model, I merged the "atbats.csv" to the "pitches.csv" by the at bat identification number. I then trimmed the data by removing several columns that were involved with pitch location and several other unneeded columns. This allowed me to get all the game information at every pitch. I then created two datasets by individual pitcher, one for known variety putcher Yu Darvish, and one for known heat thrower Max Scherzer.

The data then needed to be placed into sequences. A sequence of size 5 was chosen for this model because at bats usually average around 4-6 pitches. The sequential data was then split into testing and training for both the X (game information) and y (pitch type) with an 80/20 split.

Methods

The model that was chosen for this task was a LSTM Recurrent Neural Network. A LSTM model was chosen because of the nature of pitch choice in the game of baseball. The next pitch of an at bat is largely determined by the pitches thrown right before, not the pitches from much earlier in the game.

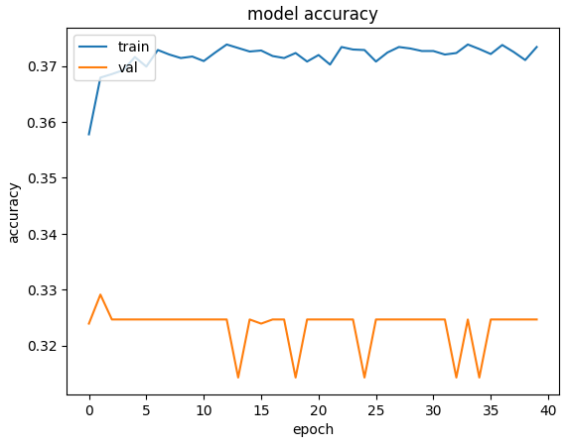
The model constructed with 4 hidden layers in total, two LSTM layers with a batch size of 100 and 2 dropout layers to ensure that the model does not rely on one pitch over and over. The final dense layer outputs a one hot encoded vector of probabilities relating to the pitch type. The model was then compiled using categorical cross entropy.

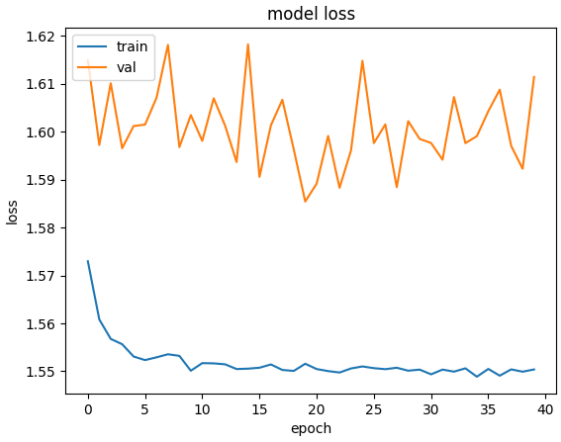
This same model was created for both the Darvish and Scherzer datasets and trained for 40 epochs each.

Results

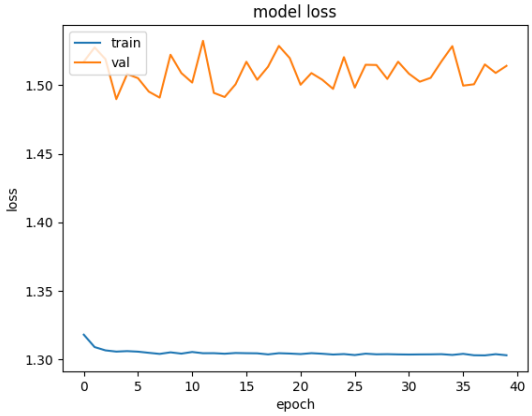
Although the models did not perform well accuracy wise, they did highlight some interesting details between the two pitchers.

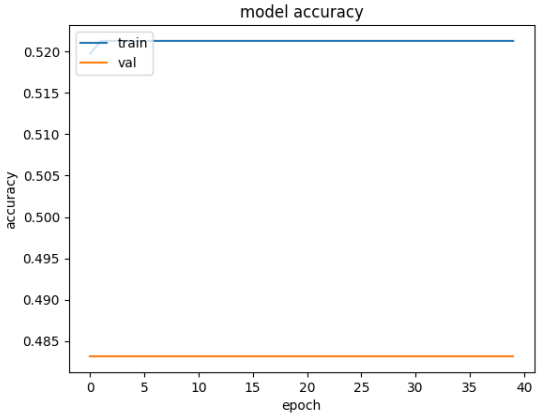
First, the Darvish model emphasized variety. Darvish's most popular pitch is used 30% of the time, and this caused the model to output some different values to try to adapt to this. Even with this however, Darvish keeps his reputation for being unpredictable and the model accuracy peaked at 38%.





Scherzer's model, although more accurate, did not have a large variety and relied heavily on predicting his fastball for almost every pitch. The total model accuracy came out to 52%.





This difference in the outputs form the models do highlight the differences in the playstyle of each player, but individual predictions are not very accurate. Because of this I would not rely on this model to determine the upcoming pitch, but instead it would be showing of overall pitcher tendencies.

Reflection

This project taught me that the initial model goal is not always going to be achievable as envisioned, but using models on the data can lead to other answers to previously unasked questions.

For similar projects in the future, I would try to differ the data used for the inputs and have a more complex model. I would also change the model output to focus on outputting a string of pitches instead of individual ones. Also, I think it would also be advantages to use a generative model instead of a recurrent neural network. Although the sequential data and relationships might be lost, there could be a strong hidden relationship between pitch type and game setting.