Predicting Major League Baseball Pitch Type

with Deep Learning

Joel Klein   
*Luddy School of Informatics, Computing, and Engineering*  
*Indiana University - Bloomington*Salt Lake City, UT  
joe.klein@iu.edu

Jake Sauberman  
*Luddy School of Informatics, Computing, and Engineering*  
*Indiana University - Bloomington*Millburn, NJ  
jake.saub@iu.edu

Ben Perkins   
*Luddy School of Informatics, Computing, and Engineering*  
*Indiana University - Bloomington*Burlington, VT  
benperki@iu.edu

***Abstract* — In 2019 Major League Baseball completed its 150th season with about 68.5 million fans attending regular season games [1]. Although the game existed for almost a century and a half, only in the last seven years has video technology and cloud services enabled pitch by pitch tracking to further improve both in-game and out-of-game baseball analytics as well as the fan viewing experience. Today, tracking capabilities capture every pitch including pitch velocity, location, spin rate, batted ball launch angle and exit velocity, and the result of the pitch (i.e. strike, ball, out, single, home run). This paper aims to leverage this publicly available data and deep learning techniques to assess the feasibility of predicting the next pitch type a major league baseball pitcher will throw to a hitter in an at-bat. Our research shows that applying the most advanced algorithms predict the next pitch type with 47.6% accuracy.**

# Introduction

The quest to accurately predict which pitch type the pitcher will throw next has long been a focal point for teams in Major League Baseball. Teams prepare pages-long, detailed reports before each game on the opposing pitchers’ tendencies in an effort to crack the code and give their batters an edge. During an at-bat, batters must be prepared to hit different pitches coming in at different speeds, moving in different directions, and thrown in different directions. The overwhelming failure rate in batting is encapsulated by a 24.4% average hit rate in the 2021 season [2]. Any additional information that might tip a batter off to what is coming next is a major advantage to tip the odds better in their favor.

Major League Baseball rules prohibit the use of any real-time technology, eliminating the possibility of using such a model in-game for real-time decision making [3]. However, several potential applications would exist in the realm of gameplanning. Teams could deploy the model to search for game situations where an opposing team’s pitcher becomes highly predictable. They could help their own pitchers become less predictable.

Beyond organizational use, such a model could add to the entertainment of a fan’s experience watching a baseball game on television. Advanced statistics have gained widespread use in baseball television broadcasts, between announcers’ commentary and on-screen graphics that are displayed intermittently throughout the action [4]. In a key, high-leverage at-bat, a graphic in the corner of the screen displaying the pitcher’s pitch type probabilities for the next pitch could greatly add to the viewer’s experience and make them feel like they are taking part in the strategy of the game.

# Literature Overview

## Major League Baseball At-Bats

Baseball is a sport made up from a series of chained events. The pitcher throws to the batter, who reacts by deciding whether to swing. If he swings and puts it in play, the fielders and runners react accordingly. The commonality between each one of these events within a baseball game is that they are all started by the pitcher.

One of the choices a pitcher makes before every pitch is which type of pitch he wants to throw. Every pitcher has their own unique pitch arsenal, made up from different ways to grip the baseball that affect how the ball moves and how fast it flies through the air. These “pitch types”, which are classified by different labels, represent the target variable of this study. Pitchers make this choice in tandem with their catcher, who provides the pitcher with the sign indicating which pitch to throw. This decision is made based on the game state, the tendencies of the batter, and the strengths of the pitcher, among other factors.

## Major League Baseball Sign Stealing History

Pitch type prediction made headlines in the fall of 2019, when reports surfaced that during the 2017 season, the Houston Astros stole opposing catchers’ signs electronically and relayed the decoded signs to the batter. They captured the catcher’s signs through a real-time video feed via a camera in the center field bleachers, which was then relayed to a monitor behind the team’s dugout. A member of the team would then alert the batter by banging on a trash can in the dugout [5]. The elaborate scheme created a competitive advantage so significant, the Astros went on to win the World Series.

With a functional model to predict pitch types based on legally-available information, teams would be able to harness deep learning to create a similar, if not slightly diminished, competitive advantage.

# Data

## Statcast

The dataset used in this study comes from publicly-available data, courtesy of Major League Baseball. During each game, several state-of-the-art tracking cameras, known as Hawkeye, record every pitch from multiple angles. They track the movement of every player on the field, the flight of the pitch as it flies toward the batter, and the trajectory of batted balls [6]. Much of this data, including all information about the features of the pitch and the current game state, is made publicly available on a per-pitch level on MLB’s data website, Baseball Savant, dating back to the 2015 season [7].

## Preparation/Scope

Due to the time series nature of the data, in order to avoid data leakage issues throughout the model process, the data was split according to the season to which it belonged. Training data incorporated the 2017 and 2018 seasons (785,777 rows), validation data came from 2019 (344,941 rows), while the test set was made up of 2020 and 2021 data (267,635 rows). This methodology ensured that a prediction could not be based on data that occurred in the future.

Since pitcher and batter embeddings were a part of the model’s scope, only pitchers and batters who made up 90% of the at-bats in the 2017 and 2018 seasons were included in the final dataset. This included 441 unique batters and 512 unique pitchers. The “cold start problem” was handled for players who appeared for the first time after the 2018 season by excluding them from the dataset. Additionally, all at-bats that took place in extra-innings (innings beyond the normal nine-inning format due to a tie game after the bottom of the ninth inning) were excluded. The COVID-shortened 2020 season featured a new rule that placed a runner on second base at the beginning of extra innings that was not present in other seasons [8]. Removing extra-inning data kept all seasons consistent under the same ruleset.

In order to further reduce the scope and make the prediction process more manageable, the target categorical variable was reduced in dimensionality. Statcast data originally reported 16 different pitch types. After grouping similar pitch types based on velocity, horizontal movement, and vertical movement, six unique pitch classes emerged as the new target variable – fourseam, twoseam, cutter, slider, curveball, and changeup.

* 1. *Game State*

The final features of the dataset can be put into four categories, the first being the game state. These features describe the current situation of the game, and include the inning, ball-strike count, number of outs, runners on base, game score, pitch number of the game, and the handedness of the batter and pitcher. These are all features that the players on the field are aware of at all times during the game itself.

* 1. *Previous Pitch Results*

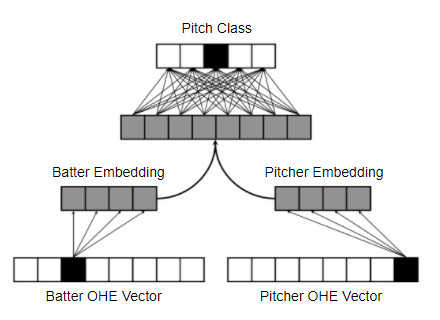
The next category of features include the results of the previous pitch. This is the event that is most fresh in the minds of the batter and the pitcher as they approach the next pitch. The hypothesis was that what happened on the previous pitch may affect the pitcher’s next pitch decision. These features included the characteristics of the previous pitch such as velocity and spin rate, its location, its outcome, and its pitch class.

* 1. *Pitcher/Batter Historical Statistics*

Thirdly, in order to provide additional information about the talent level and tendencies of the batter and pitcher, historical statistics were calculated. For the batter, this included their rolling 50-pitch weighted on-base average (wOBA) [9] and swinging strike percentage (“whiff rate”) for each pitch class. For the pitcher, these same features were calculated, in addition to rolling 50-pitch velocity and spin rate features and usage rates for each pitch class. These usage rates were calculated on a short term and long term basis, with short term representing a rolling 30-pitch average, and long term stretching back to the most recent 100 pitches for its rolling average. The usage rates also served as a constraint for the model’s predictions. Not every pitcher throws each of the six pitch classes; for example, less than 25% of pitchers in the dataset threw a cutter. Therefore, for those pitchers, any cutter prediction from the model would be wrong every time. With these usage features, the model is aware of which pitches are in each pitcher’s arsenal.

* 1. *Pitcher/Batter Embeddings*

Our research hypothesis prior to modeling is that the batter and pitcher matchup will heavily influence the pitch types selected by the pitcher. For certain power hitting right-handed batters like Miguel Sanó, a right-handed pitcher might be more prone to throw a slider. There are 443 unique batters and 521 unique pitchers in the 2017 and 2018 seasons in scope. Traditionally, categorical vectors are typically fed to machine learning algorithms as one-hot encoded (OHE) vectors which would lead to 964 sparse vector inputs to the model. [10] suggested an approach to represent batters and pitchers as n-dimensional embedding vectors where the embeddings are learned by a fully-connected network where the inputs to the network are the OHE vectors. Fig. 1, the inputs are then passed to embedding layers which are then concatenated and passed to another fully connected layer to predict the pitch class. This model generates 9-dimensional embeddings for the

Fig. 1. (batter|pitcher)2vec model architecture generating embeddings

pitcher and hitter respectively which are passed as inputs to the deep learning models. The embeddings were evaluated using 2D tSNE where similar pitchers such as Josh Hader and Taylor Rogers (hard throwing left-handed pitchers with a wipeout slider) were clustered in the same location. The batter and pitcher vectors alone predicted the pitch class with 42% accuracy (the majority class - four seam fastball - accounts for 36% of all pitches).

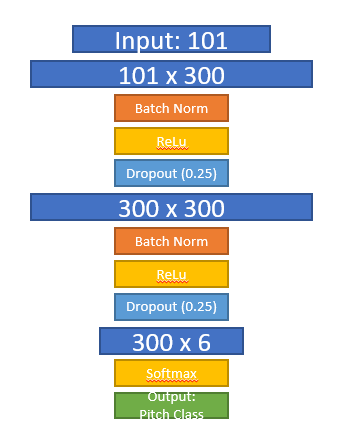
# Methodology

## Fully Connected Neural Networks

The first set of deep learning models attempted were a set of three fully connected networks. Fully connected networks are advantageous over traditional machine learning algorithms due to the large size of this data and the built-in feature extraction within the network. These networks were primarily trained to assess the lift in accuracy when adding feature sets as input such as the previous pitch outcome and batter/pitcher embeddings to the current game state features.

The first model fit contained only the current game state features such as the count, outs, runners on base as well as the historical pitcher/batter statistics like fourseam usage and fourseam whiff %. The second model added in the results of the previous pitch as input features such as previous pitch class and previous pitch outcome i.e. ball, strike, out, hit. The third model added in the batter and pitcher 9-dimensional embeddings as inputs.

Each model architecture was tuned with varying batch sizes (2,000 or 10,000), number of layers (3 to 5), hidden layer size (300 or 500), dropout (.2 to .4), and learning rate (.001 to .0001). All models were trained with the Adam optimizer and the categorical cross entropy loss function. Early stopping prevented the model from overfitting if there were 20 epochs with no additional improvement in validation accuracy. Softmax activation was applied to the final layer as this is a multiclass classification problem. Fig. 2, the best performing architecture leveraged the current game state, historical batter/pitcher stats, and previous pitch type. The batter/pitcher embeddings did not boost model performance.

Fig. 2. Fully Connected Network Architecture

## Recurrent Neural Network

The best performing fully connected included the previous pitch type and result. Pitches selected within an at-bat are often heavily influenced by preceding pitches within the at-bat beyond just the last pitch thrown. Pitches within an at-bat occur sequentially. Thus, a recurrent neural network, specifically an LSTM architecture is able to pass information from previous pitch sequences within the at-bat as inputs to inform the upcoming pitch prediction. LSTMs consist of a cell state which contains the information from previous time steps in the sequence, along with the input gate, output gate, and forget gate which regulates new data flowing in and out of the cell state. Discuss the improvements of why LSTM may be beneficial for this problem. Include model architecture, tuning procedure.

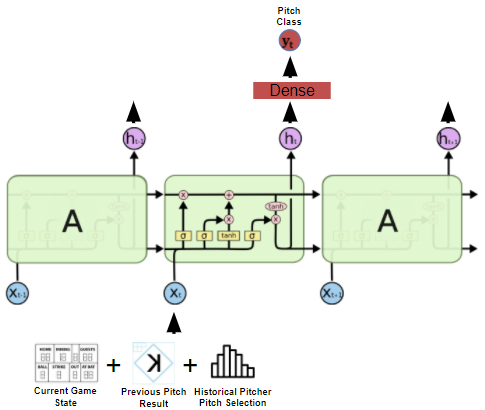
The LSTM model was tuned with varying batch sizes (2,000 or 5,000), number of layers (1 to 3), hidden layer size (300 or 500), dropout (.2 to .4), and learning rate (.001 to .0001). Early stopping and softmax activation was applied to the final layer similar to the fully connected network. Fig. 3, the best performing architecture leveraged the current game state, historical batter/pitcher stats, and previous pitch type. 

Fig. 3. LSTM Network Architecture

The final architecture used batch size 2,000, 2 hidden layers, hidden size 300, dropout rate .25, and learning rate .0001.

## Recurrent Neural Network w/ Attention Mechanism

A major limitation in the traditional RNN or LSTM architecture is that all the information from the previous time steps (*t-1, t-2, … t-k*) are encoded in one complex hidden state vector before making the prediction at time *t*. For long sequences, such as long at-bats lasting more than eight pitches, the information from the first few pitches needs to travel through several layers within the LSTM encoder. It is possible that meaningful information from those first few pitches are lost during this encoder phase. As suggested in [11], adding an attention mechanism allows all the hidden states leading up to the prediction at time t to be weighted into the final prediction. Fig 4, the weights of those hidden states are determined by training an additional fully connected layer to determine which information from the hidden states should be passed to the decoder. The decoder then takes as input the context vector and the previous state input.

The LSTM model with the attention mechanism was tuned with varying batch sizes (2,000 or 3,000), number of layers (1 to 2), hidden layer size (300 or 500), dropout (.2 to .4), and learning rate (.001 to .0001). Early stopping and Softmax activation was applied to the final layer similar to the fully connected and LSTM networks. The best performing architecture leveraged the current game state, historical batter/pitcher stats, and previous pitch type. The final architecture used batch size 2,000, 1 hidden layer, hidden size 300, encoder and decoder dropout rate .25, and learning rate .0001.

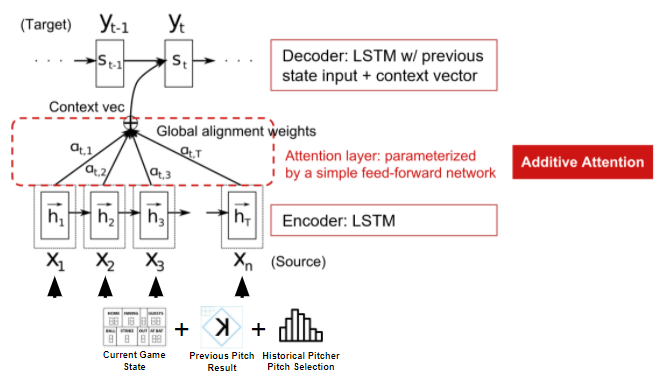


Fig. 4. LSTM w/ Attention Mechanism Network Architecture

# Results

Fig 5, at the major league level, the best performing model was the LSTM model which generated a validation accuracy of 48.43% (12 percentage points better than predicting the majority class). Most models converged in accuracy between 50 and 100 epochs across different learning rates. The historical pitcher and batter statistics were the most important features to the models attempted based on the list in accuracy.

1. Model Results on 2019 Season (Validation Set)

| ***Model*** | ***Features*** | ***Accuracy*** | ***Precision*** | ***Recall*** |
| --- | --- | --- | --- | --- |
| Fully Connected | Game state + statistics | 46.36% | 45.63% | 46.36% |
| Fully Connected | Game state + statistics + previous pitch info | 47.09% | 45.95% | 47.09% |
| Fully Connected | Game state + statistics + previous pitch info + embeddings | 46.80% | 46.19% | 46.80% |
| LSTM | Game state + statistics + previous pitch info | 48.43% | 47.02% | 48.43% |
| LSTM w/  attention mechanism | Game state + statistics + previous pitch info | 48.18% | 46.82% | 48.18% |

1. Weighted average precision and recall

The method to add in batter and pitcher embeddings did not add lift for the fully connected or the recurrent neural network models. Adding the attention mechanism to the LSTM network unexpectedly did not increase performance. This is likely due to 21 pitches being the longest sequence of pitches in any at-bat. The attention mechanism may be more effective if perhaps the entire game was passed in as a sequence rather than individual at-bats.

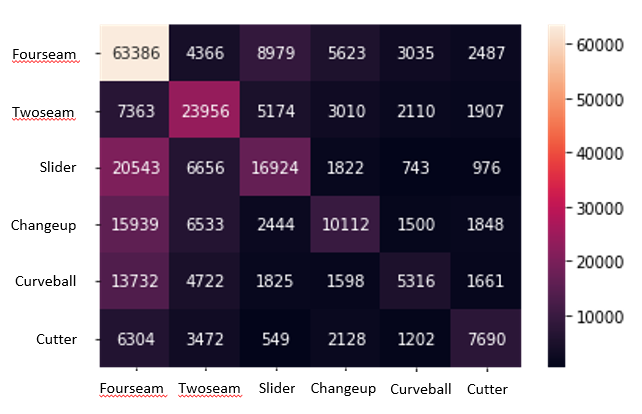


Fig. 5. Best model 2020-2021 season pitch prediction confusion matrix

The LSTM model was selected as the best model to predict the test data (2020 and 2021 seasons) based on the validation accuracy. The best performing model predicted with 47.60% accuracy on the hold out data, performing best at predicting fourseam fastballs (58.90% F1 Score).

# Conclusion

The model performance results confirm that predicting the next pitch a major league pitcher will throw with a high accuracy level is quite challenging when considering all pitchers. A more effective approach may be to limit the model to a specific pitcher and leverage only that individual’s history. Another approach is to feed in the entire game as a sequence to predict the next pitch type thrown. Additional features like changes in pitch velocity throughout the game may also improve the model performance for starting pitchers.

##### References

1. Associated Press, “MLB draws 45.3 million fans at stadiums this season,” *espn.com*. October 2021.
2. Fangraphs.com, “Leaderboards”, 2021.
3. M. Rivera. “Memo Details New MLB Rules to Discipline Electronic Sign-Stealing.” *ESPN*, ESPN Internet Ventures, July 2020.
4. Feldman, Jacob. “How MLB Broadcasters Approach Advanced Stats.” *Sports Illustrated*, Sports Illustrated, March 2019.
5. K. Rosenthal, E. Drellich. “The Astros Stole Signs Electronically in 2017 - Part of a Much Broader Issue for Major League Baseball.” *The Athletic*, The Athletic, June 2020.
6. B. Jedlovec. “Introducing Statcast 2020: Hawk-Eye and Google Cloud.” *Medium*, MLB Technology Blog, July 2020.
7. MLB.com, “Statcast”, 2021.
8. “Designated Runner: Glossary.” *MLB.com*.
9. P. Slowinski. “wOBA.” *Sabermetrics Library*.
10. M. Alcorn, “(batter|pitcher)2vec: Statistic-Free Talent Modeling With Neural Player Embeddings,” MIT Sloan Sports Analytics Conference. pp. 4-7, 2018.
11. D. Bahdanau, K. Cho, Y. Bengio, "Neural machine translation by jointly learning to align and translate“, arXiv preprint arXiv:1409.0473. 2014.
12. I. Sutskever, O. Vinyals, Q. Le, “Sequence to Sequence Learning with Neural Networks”, Advances in neural information processing systems. 2014.
13. J. Shaikh, “Essentials of Deep Learning – Sequence to Sequence modelling with Attention (using python)”, Analytics Vidhya. 2018.
14. A. Persson, “Pytorch Seq2Seq with Attention for Machine Translation”, Youtube. 2020.
15. A. Vaswani, N. Shazeer, N. Parmar, J Uszkoreit, L. Jones, A. N. Gomez, I. Polosukhin, “Attention is all you need”, Advances in neural information processing systems. 2017.