A Deep Learning Approach to Player Forecast Systems

Elliott Chartock: elboy@stanford.edu

Motivation

Each March, Baseball Prospectus unveils their player projections and people take notice. Such forecast systems are pivotal for the success of the billion dollar fantasy sports industry. Additionally, there are practical applications to building highly accurate player prediction models. In the current Moneyball age of baseball, sabermetrics often guide management decisions to draft, trade, or drop players on their roster. Baseball is a field with exhaustive data records that remains heretofore absent of deep learning influence.

Problem Statement

- Build a player forecast system that predicts season performance statistics for professional baseball players.
- Train fully-connected and recurrent neural networks to predict future statistics and compare to non-deep learning baselines.
- Evaluate models with R^2, square of Pearson correlation.

Data

- All player seasons from 1871 - 2016
- Threshold of at least 100 at bats per season
- Omit missing data omission and minimum AB threshold: 102,816 -> 17,130 player seasons

Data split	Train	Dev	Test
Number of player seasons	13,704	1,713	1,713

ID	Year	G	AB	R	H	2B	3B	HR
ortizda01	2015	146	528	73	144	37	0	37

RBI	SB	CS	
108	0	1	. . .

Label	2016 HR	2016 SB
	38	2

Figure 1: David Ortiz 2015 statistics and 2016 home runs and stolen bases labels.

Last-k Baseline

To predict target home runs for player p in year t, we do:

$$p_t\,[HR] = \sum_{i=t-k}^{t-1} p_i\,[HR]$$

- Based off Marcel model, which yields decent player projection results
- Tested k = 1, ... , 5

Last-1 Fully Connected

To predict target home runs for player p, we input last year's statistics for player p as network input features.

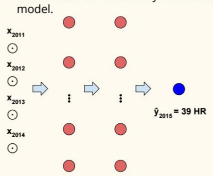
- 17 input features
- Prediction target is home runs

Last-k Fully Connected

Concatenate statistics for last k seasons.

- 17 * k input features
- Tested k = 5
- 2 layers, h_1 = 200, h_2 =100
- Batch Norm and ReLU applied at each layer

Figure 2: Below figure shows architecture for Last-k fully connected model.



Recurrent Neural Network

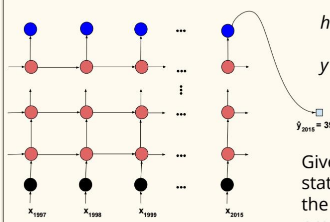


Figure 3: Above figure shows RNN architecture. Input is all previous season statistics and output is the label prediction. We ignore all outputs except for the last timestep.

$$h^l_t = \tanh(W^l_{hh} h^l_{t-1} + W^l_{xh} x_t)$$

$$y^l_t = W^l_{hy} h^l_t$$

Given a sequence of season statistics $x_1, x_2, \ldots, x_t$ the RNN with L layers predicts nexts season's label y^L_t

Results

	Home Runs			
Model	Last-k	Last-1 FCN	Last-5 FCN	RNN*
Train R^2	NA	0.538	0.591	-
Train Loss	NA	1.431	1.310	-
Test R^2	0.526	0.564	**0.606**	-
Test Loss	NA	1.317	1.206	-

* Results forthcoming

Analysis & Discussion

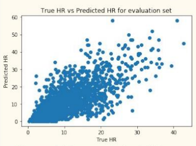


Figure 4: Best model exhibits clear positive correlation with strong outliers

Worst Predictions	True HR	Pred HR	Prev HR	Highest Predictions	True HR	Pred HR	Prev HR
Ryan Howard, 2006	58	23	22	Barry Bonds, 2004	45	43	45
Kevin Mitchell, 1989	47	16	19	Mark McGwire, 1997	58	41	52
Hank Aaron, 1957	44	16	26	Mark Reynolds, 2010	32	39	44
Tino Martinez, 1997	44	17	25	Alfonso Soriano, 2007	33	37	46
Aramis Ramirez, 2001	34	7	6	Alex Rodriguez, 2004	36	36	47

Last-1 fully connected is learning identity function of previous year's HR. Last-5 fully connected network is the best performing model, meaning that statistics of prior seasons help predict future player performance. This leads me to believe that there exist temporal patterns in a player's career that can be captured by an RNN.

Conclusion & Future Work

My models' superior performance over their non deep learning baseline show promise for the use of deep learning within sabermetrics. Further analysis is necessary to compare my models predictions with leading industry predictions, such as PECOTA. My next steps are to complete RNN predictions and then predict on more statistics.