A Simple Neural Network For Projecting NBA Player Fantasy Success

Jake Daly, PID: A53312764

UCSD Machine Learning and Data Science

jmdaly@eng.ucsd.edu

ABSTRACT

In this experiment, we train a simple neural network to predict the fantasy success of NBA players for the current season using historical player stats since the 1997-98 NBA season. We will use an open source web-scraper to pull data (player stats) off of nba.com, then use select statistics as the features of our model which we will train over 17 seasons of data. The model, after learning the weights and biases of the selected features, will attempt to predict player fantasy success and make recommendations on which players should be drafted according to a standard set of league rules. Finally, we will compare these projections to those of claimed experts to see how well our model measures against the status quo recommendations.

1 INTRODUCTION

Fantasy sports have become a very popular pastime activity for sport enthusiasts, especially so in the professional football, basketball, and baseball fan communities. Fans can draft their favorite players onto their virtual teams, and compete against friends and others in customized fantasy sports leagues. As fun as these leagues can be, there are typically competitive incentives that come with winning matches. Of course, there is the personal glory that comes attached to the victory over a friend, but even more concretely than this, it is quite common for there to be financial incentives to win leagues. Just a $50 buy-in per person in a 10 person league yields a $500 jack-pot that players are competing over.

If a theme has emerged throughout the past decade of technological advancement, it's the renaissance of data. The National Basketball Association, as much as any organization over the last 10 years, has championed an era of data-backed growth and thinking. All technical aspects of the sport have been mapped to statistics and advanced metrics which can be easily accessed from massive databases and used to perform analytics.

The presence and accessibility of these mountains of data and increasing popularity of fantasy sports naturally invites the question: how can we employ this data to our advantage in fantasy basketball? In this paper we will attempt to address this question by projecting player success for the season by training a shallow neural network. This paper is organized as follows: Section 2 provides an overview of the data, and how it was obtained and stored. Section 3 provides an overview of what features were selected from the data and what decision were made to construct the model. Section 4 shares some results and conclusions about the analysis, and compares the model's recommendations to those made by ESPN's claimed fantasy basketball experts.

2 DATA

Python's rise to the top of the most popular programming languages has been fueled in part by the plethora of packages available to its users—even several different NBA statistics web scrapers are available. For our experiment, we have chosen *nba_api*, which is not only extensive and thorough in the type of statistics that can be pulled, but pulls them directly from nba.com (versus third party websites). The API itself is organized hierarchically in a way that reflects how the user sees the data when visiting the website, creating natural structure for anyone using the interface who is already familiar with the website.

The package accesses the website's endpoints via http requests. For security (for example, against DDOS attacks), the website uses data-rate limiting, a common security measure which effectively limits the amount of data that can flow off of the website. Because of this, we build infrastructure to handle request timeouts and other connection termination errors, while pulling these gigabytes of data off of the NBA's servers.

After being pulled off the web, the data is stored locally into "player-seasons". One player-season of data represents one player's statistics for one season in the league. For reasons that we will explain in the next section, these player-seasons have been organized by the number of seasons the player has been in the league. For example, "Year1" through "Year20" databases have been created. In the "Year1" database, we have all available game statistics for every player that has ever spent one season in the league (which is every player). The "Year20" database only contains those player-seasons in which it was the player's 20th season in the league. This distribution is visualized in **Figure 1**.

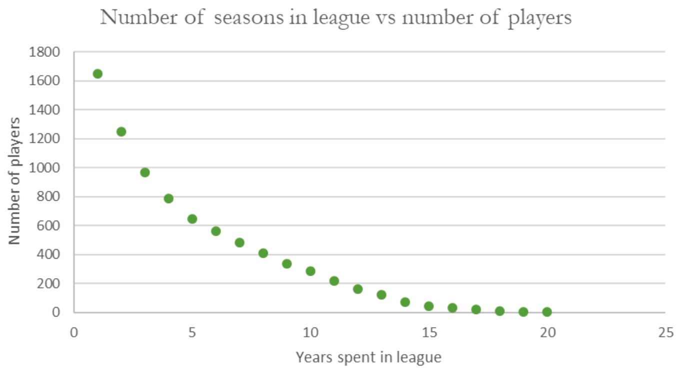

Figure 1: How many players (vertical axis) have played a given number of seasons (horizontal axis)

3 MODEL CONSTRUCTION

The following statistics were selected to be used as the features for the network:

- Basic:
  - Points, rebounds, assists, blocks, steals, minutes, games played, field goals made, field goals attempted, turnovers, free throws made, and free throws attempted
- Advanced:
  - Player impact estimate (PIE), usage percent, true shooting, awards, fantasy points from previous years

The basic stats are common stats that can be used to assess a player's fantasy value. All of these stats directly feed into how many fantasy points a player scores on a given night, which ultimately is what is needed to win games in a fantasy league. Which stats are worth how many points is a crucial detail of the model—it directly determines how valuable each player is. For example, if assists are worth 5 points each, but rebounds only one, our model should determine that guards (who tend to acquire more assists) are more valuable players to have then centers and forwards. For this model, we will use the default ESPN fantasy scoring points, which in general, awards +1 point to positive stats (eg. points, rebounds), and -1 negative stats (eg. fouls, or field goal attempts).

The advanced stats do not directly influence how many points a player scores in fantasy, but are useful in helping the model differentiate between better players and worse players in the league. The PIE stat is a player efficiency stat; usage percent attempts to determine how often a player is utilized; true shooting measures how efficiently players convert on field goals, weighted by how much the field goal is worth (2 or 3 points); awards are points given to a player determined by which awards they won the previous season (for example if they were awarded most valuable player, or first team all-NBA honors); and fantasy points from previous years is self-explanatory. While the basic stats give weight to the tangible features that give a player value, the advanced metrics attempt to identify the quality of a player through more complex advanced metrics.

After some trial and error, there are some additional conditions which we impose on the model to help improve its accuracy. First, we impose a playing time restriction on player-seasons. Without a playing time restriction, the data becomes biased towards players who did not have sufficient playing time to log meaningful statistics. We attempt to filter out this noise by limiting both the minutes per game stat, as well as the number of games played, to at least 5 minutes per game and 15 games played.

A second tactic we use to increase the efficacy of the model is that we create separate weights and biases based off of how many years the player has been in the league. This was done to provide the model with additional context for the player-seasons: for example, a first year ("rookie") player who scores 10 points per game should have a different trajectory than a 15th year player who averages the same amount. Creating different weights and biases allows the model to look at players based on how many years they have been in the league, which is important context for determining how successful they will be next season.

Lastly, our measure of these stats is for *season totals*, as opposed to *per game*. Under this scope, our model should recognize players who produce more over the course of an entire season. Players who are injured frequently or are "rested" by coaches for particular games are penalized in their projections. Ultimately, we would like our fantasy team to produce the most points over the course of the season, so season total stats are more reflective of this notion than per game ones.

To train the model, we start the script which begins with the "Year1" player-seasons, and performs the regression and gradient descent, finding the optimal weights and biases. **Figure 2** shows the gradient descent algorithm running on the "Year1" player-season data. Progressing through each year, it stores these optimized weights and coefficients in a data structure. "Year17" is the last year in which the number of players exceeds the number of features by a large enough margin to obtain useful coefficients. After obtaining each set of weights and biases, we step through the list of active players, determine how many years they have been in the league, and gather what their basic and advanced stats were for the previous year. We then access the corresponding weights and biases, and make the prediction for how the player will do for the following year.

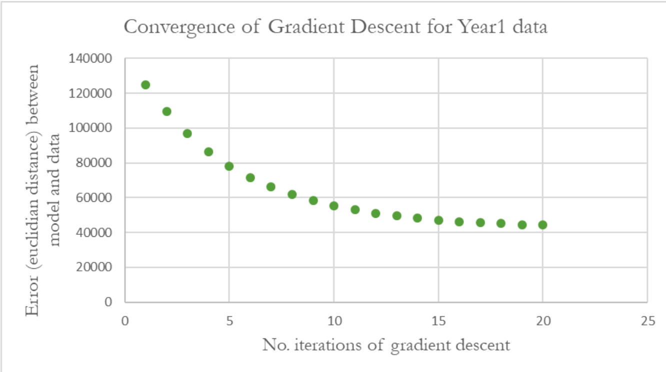

Figure 2: Convergence of the gradient descent algorithm, optimizing the weights and biases of the layer on successive iterations

4 RESULTS AND CONCLUSIONS

There are two types of output files that are of interest. In the first, *PlayerDataEveryYear.xlsx*, there is an ordered list of projections for every player, every season they've been in the league, organized as the various tabs in the excel notebook. These lists confirm that the model is working well: for "Year1" data, we should expect to see some of the league's most prominent rookie player-seasons since the 1997-98 season. At the top of the list for the "Year1" data is surely enough Blake Griffin's 2010-2011 rookie season, one of the most thunderous of all time. Also in this top ten, we find LeBron James, Karl-Anthony Towns, Kevin Durant, and others players that had break out rookie seasons. The top ten projections for every player that has played in a second season in the NBA and met the playing time requirements are given in **Table 1** as an example.

Name	Projection
Blake Griffin (2010-11)	1777.224038
Elton Brand (1999-00)	1704.267129
Pau Gasol (2001-02)	1639.539379
Karl-Anthony Towns (2015-16)	1563.697814
LeBron James (2003-04)	1549.399419
Carmelo Anthony (2003-04)	1513.789135
Ben Simmons (2017-18)	1451.553931
Kevin Durant (2007-08)	1446.947385
Dwight Howard (2004-05)	1446.865238
Lamar Odom (1999-00)	1426.428496

The second output is the list of players with projected fantasy points for a given season. Any season since 1997-98 may be selected, but here we will analyze the projections for the most recent, full NBA season (2018-19). This way, we can see how our model compares to the actual results and to the picks made by ESPN's panel of fantasy experts. This comparison for the top 50 players is showed in **Table 2** on the next page**.** Green squares indicate which model was more accurate in its prediction of fantasy rankings for the 2018-19 season.

Out of the 50 players shown, our model beats the ESPN expert ratings by the score of 32 to 18, indicating that the neural network was almost twice as successful in predicting NBA player fantasy success for the following year. In the top ten players, the experts and the model perform equally well, but as we move further down the list, the neural network becomes better at predicting and finding break out players that the ESPN experts neglect. As a result, we recommend using this model when assessing the value of players in an NBA fantasy basketball draft.

Name	Projected Points	Actual Points	Actual Rank	Model Rank	ESPN Rank
James Harden (2017-18)	2035.035945	2378	1	1	2
Anthony Davis (2017-18)	1932.206753	1773	15	2	1
Karl-Anthony Towns (2017-18)	1903.9749	2100	5	3	5
Giannis Antetokounmpo (2017-18)	1809.615963	2365	2	4	3
LeBron James (2017-18)	1742.058991	1601	21	5	6
Ben Simmons (2017-18)	1687.902339	1795	14	6	18
Russell Westbrook (2017-18)	1640.739581	1964	8	7	4
Nikola Jokic (2017-18)	1578.640358	2151	3	8	9
Andre Drummond (2017-18)	1551.397808	1950	9	9	22
Damian Lillard (2017-18)	1543.658241	1867	12	10	11
Kevin Durant (2017-18)	1485.300841	2041	6	11	7
DeMarcus Cousins (2017-18)	1467.235546	1439	81	12	43
Joel Embiid (2017-18)	1388.417031	1911	11	13	10
Clint Capela (2017-18)	1335.359677	1634	19	14	45
Victor Oladipo (2017-18)	1334.73931	1687	18	15	13
Bradley Beal (2017-18)	1332.607946	1769	17	16	27
Paul George (2017-18)	1315.955687	1921	10	17	15
Dwight Howard (2017-18)	1309.885483	1782	16	18	48
Stephen Curry (2017-18)	1302.018146	1641	18	19	8
DeMar DeRozan (2017-18)	1286.245312	1574	23	20	38
Julius Randle (2017-18)	1280.95331	1454	26	21	61
Chris Paul (2017-18)	1251.093404	1051	71	22	16
DeAndre Jordan (2017-18)	1242.295803	1415	31	23	50
Rudy Gobert (2017-18)	1219.944763	2018	7	24	21
Jimmy Butler (2017-18)	1216.60695	1240	53	25	23
Donovan Mitchell (2017-18)	1192.226887	1233	54	26	31
LaMarcus Aldridge (2017-18)	1172.747758	1826	13	27	34
Draymond Green (2017-18)	1151.971291	1423	84	28	25
Kemba Walker (2017-18)	1150.622247	1728	17	29	19
Jrue Holiday (2017-18)	1143.872889	1420	30	30	14
Blake Griffin (2017-18)	1138.565818	1601	22	31	33
Jayson Tatum (2017-18)	1136.592489	1144	62	32	51
Kyle Lowry (2017-18)	1113.734576	1580	91	33	28
Khris Middleton (2017-18)	1109.300597	1263	47	34	30
Devin Booker (2017-18)	1106.689754	1345	42	35	24
Jusuf Nurkic (2017-18)	1102.533011	1410	32	36	78
Kevin Love (2017-18)	1092.152229	1238	90	37	20
John Wall (2017-18)	1088.548106	831	89	38	17
John Collins (2017-18)	1088.263961	1249	49	39	97
CJ McCollum (2017-18)	1083.545164	1059	70	40	38
Marc Gasol (2017-18)	1073.401993	1324	39	41	37
Enes Kanter (2017-18)	1065.66516	1145	61	42	59
Hassan Whiteside (2017-18)	1055.626822	1263	46	43	73
Steven Adams (2017-18)	1054.385554	1401	34	44	63
Eric Bledsoe (2017-18)	1032.180113	1306	40	45	32
Kyrie Irving (2017-18)	1023.744605	1547	24	46	26
Lauri Markkanen (2017-18)	1020.309771	1102	88	47	83
Kyle Kuzma (2017-18)	1006.122701	1177	80	48	98
Tobias Harris (2017-18)	1002.003185	1602	20	49	42
Otto Porter Jr. (2017-18)	1000.101287	1322	92	50	41

Table 2: Neural network projections for player fantasy success versus ESPN's recommendations