DeepShot: A Deep Learning Approach To Predicting Basketball Success

Vamsi Saladi Department of Computer Science
Stanford University University
Stanford, CA 94305
vamsi99@stanford.edu

Abstract

This project focuses on using neural networks for the purpose of predicting basketball player success. More specifically, given a player's early career statistics and physical attributes along with other statistical features, we hope to predict which of the five categories their future success falls under: released within 4 years of data point, remained in the league as role player, became a starter, became a starter and all star, and became a perennial all star. We approach this problem with 2 and 3 layer neural networks and attempt along with a baseline model of logistic regression. Using our models, we were able to achieve state of the art results of 70% accuracy. Link to the code can be found here: https://drive.google.com/open?id=13VZ6dWdM3KT1AQmy8pGuzvUJMBNvmKgS

1 Introduction

The NBA is now the second most watched sports league in the US, and the tenth most followed sport in the world. The growth in popularity of basketball domestically and internationally makes it an ideal place to apply new deep learning techniques, since there is remarkable financial and social benefits of being able better scout and rate basketball players. In fact, basketball player valuation is one of the most highly sought after skills in the world of scouting, since NBA players make the largest average annual salary amongst all sports leagues in the United States. Thus, understanding which players are worth the money and which players are not drastically impacts the organization's success and the economic success of the organization's location.

With this in mind, the following project attempts to help aid basketball scouts and evaluators. Specifically, given a player's current level of production and physical characteristics, we attempt to develop methods to allow us to predict the career success of said player, and thus determine if they are worth keeping on a team, and how important they are to the team's success.

2 Related Works

2.1 Ivankovic's Study

Applying neural networks to the sport of basketball is a recent but promising idea. Since many sports rely so strongly on statistics and numerical analysis, it makes it ideal to use deep learning methods to help aid basketball coaches and teams.

One of the earliest papers done on this topic was done by Ivankovic et. al. [4] in November of 2010 in a Hungarian Polytechnic Journal. Titled "Appliance of Neural Networks in Basketball Scouting", Ivankovic looked at basketball data from the First B basketball league in Serbia. Due to the early nature of the work, we can immediately see from drawbacks: the Serbian basketball league is by no

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

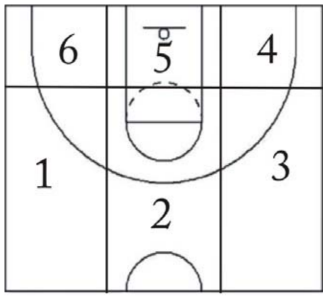

Figure 1: Regions of the basketball court as determined by Ivankovic

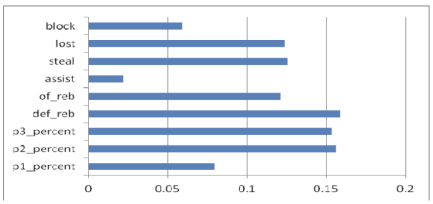

Figure 2: Factors that were important to success

means one of the more competitive or successful leagues in the world. Thus, it makes the data a little less reliable for our purposes, but that does not diminish from the work at all.

Importantly, Ivankovic took the approach of dividing the basketball court into six different parts, which are shown below and attempting to find the importance of obtaining points from each of those regions (pictured below) along with the importance of one of my several statistical categories on the outcome of a game (also pictured below).

The general idea behind the remainder of the study was that the statistical categories that contributed most to success and the regions of the court where these statistics were obtained from could be combined to determine which players were most valuable. Specifically, by analyzing large quantities of data, Ivankovic determined which factors correlated most with success in a game and where they occured on the court, and was able to link those to the statistics and player movement of a particular player, and determine whether that player was efficient and valuable to a team.

One aspect of this research that is incredibly useful is the player movement data. However, this kind of data is harder to come by in large quantities in the NBA since a lot of broadcasts and televised games are banned from being used for research for commercial reasons. However, this dimension to the analysis was impressive and useful. However, more pertinent to our study is the parameters that Ivankovic determined were most influential to success. In particular, Ivankovic determined that offensive and defensive rebounds, paint rebounds in general, and steals were incredibly indicative of a player's success.

As aspect of this revelation that is important to consider is that in 2010, basketball was dominated by larger players who were more inclined to score and closer to the basket than in modern day basketball. However, this limitation in the study does not take away from reaffirming our own thought that defensive abilities and rebounding skills are very valuable to a player's success, which is our biggest takeaway from this study.

2.2 Barron's Study in Soccer Success

Another important and useful paper was the study published in October of 2018 by Barron, Ball, Robbins, and Sunderland about using neural networks to predict player success in soccer. Now, although this is a different sport, it provides insight into the world of sports in general, and does by taking data from the highest level of the sport: the UEFA Champions league. Their aim was to objectively identify key performance indicators in professional soccer that influence outfield players' league status using an artificial neural network. Specifically, they wanted to focus on outfield players since the goalkeeper position is more of an outlier in the sport in terms of movement and action.

Barron et. al. proceeded as follows. First, they initially divided up their player data by looking at how many minutes each player played. Based on their minute contribution they made a distinction between substitutes and starting players. This is an incredibly important distinction to make, especially in the case of basketball. In basketball, it is entirely possible

for you to have an incredibly successful and illustrious career as a backup. This means that we must take into account minutes played per game as important binary indicator for any data before passing into the network we design. More specifically, it means that we should have an additional binary feature that indicates if a given data point is coming from a role player or a starter.

Additionally, another detail that was useful to take inspiration from was the idea that we could classify success in tiers. In this study, Barron takes success in three levels: went to lower league, remained at the same level of reasonable championship football, and rose to the elite level of Premier League or La Liga football. We take inspiration from this, and use the following levels of success: released within 4 years of data point, remained in the league as role player, became a starter, became a starter and all star, and became a perennial all star.

The study then used a 3-layer perceptron network with a learning rate of 0.1 and a momentum of 0.5 and a binary cross entropy loss and was trained for a maximum of 300 epochs. From this training, the network was able to predict on testing data with roughly a 63 percent accuracy rate. We use this as a comparable level of success that we wish to achieve in our model, if not do better.

3 Dataset and Features

We are using three distinct data frames to draw from for the training data. First, we have a dataset that has the list of all NBA players who played from the year 1950 onwards, and several characteristics of them. I combine this dataframe with another list of all NBA players that made the hall of fame to be my true results.

Next, we have a database that is a superset of the previous database, which has the information above along with the start and end years of their careers and the position in basketball that they played. This is the data frame that will serve as the training data in combination with your third dataset, which is a list where every row is individual season statistics for every NBA player since 1950. We have the points they scored per game, the rebounds they collected, etc. This third dataframe includes just about every measurable statistic available in sports analytics, including value over replacement player, player efficiency rating, etc. Thus, we can consolidate physical traits of the player with their season by season statistics to get our training data. After all of this, we have 3922 training examples.

	Player	height	weight	collage
0	Curly Armstrong	180.0	77.0	Indiana University
1	Cliff Barker	188.0	83.0	University of Kentucky
2	Leo Barnhorst	193.0	86.0	University of Notre Dame
3	Ed Bartels	196.0	88.0	North Carolina State University
4	Ralph Beard	178.0	79.0	University of Kentucky

Figure 3: The first dataframe header

	name	year_start	year_end	position	height	weight	birth_date	college
0	Alaa Abdelnaby	1991	1995	F-C	6-10	240.0	June 24, 1968	Duke University
2	Kareem Abdul-Jabbar	1970	1989	C	7-2	225.0	April 16, 1947	University of California, Los Angeles
3	Mahmoud Abdul-Rauf	1991	2001	G	6-1	162.0	March 9, 1969	Louisiana State University
4	Tariq Abdul-Wahad	1998	2003	F	6-6	223.0	November 3, 1974	San Jose State University
5	Shareef Abdur-Rahim	1997	2008	F	6-9	225.0	December 11, 1976	University of California

Figure 4: The third datagrame header

4 Methods

First, we address data preprocessing. We go through all of the data and take out any data before 1982, which was 3 years after the 3 point line was implemented. This was to ensure that the game could be applied to modern day basketball. Next, I had to make sure that for each of the columns in the seasons data, I had valid and usable data. First, I had to make sure that the player had data for some of the more eccentric metrics. Then, I had to go through and fill out some of the percentages which were just not given for some reason in the data file itself. For example, I had to go and fill in 3P%, 2P%, FT%, the true y-value of the player data, which is what I trained my logistic regression to predict.

Next, I organized the data with the fields that were deemed most important by research for the purpose of a simple baseline. The fields I choose were: Games Played, Points Scored, Free Throw Percentage, 3 point Percentage, 2 point Percentage , Effective Field Goal Percentage, Offensive Rebound Percentage, Steal Percentage, Turnover Percentage, Assist Percentage, Block Percentage

4.1 Baseline: Logistic Regression

The baseline model was trained to do one simple thing. Given a players' essential features gathered from his data, predict what level of success the player would achieve. More specifically, the network would predict which of the 5 categories of success the player will end up falling in: released within 4 years of data point, remained in the league as role player, became a starter, became a starter and all star, and became a perennial all star.

In my implementation of Logistic Regression, we were able to achieve an overall accuracy of 39.89%.

4.2 2-Layer Neural Network

Next, we designed a 2-layer neural network that had an input layer of size 11 (from the important features that we selected) and we tried different hidden layer sizes ranging from 16 to 22. The output layer was 5 units (since there are five categories of prediction) and we use a softmax function as activation function. For each 2-layer network, we also did hyperparameter tuning. We first did a grid search of learning rates between 10^{-8} and 0.1, and then finetuned once more in the chosen range.

As we will later see, the most successful 2-layer neural network was one that had a hidden size of 21 units, which achieved a 60.20% accuracy.

4.3 3-Layer Neural Network

Finally, we designed a 3-layer neural network that had an input layer of size 11 (from the important features that we selected) and we tried different hidden layer sizes ranging from 16 to 22. Both hidden layers used the same hidden size in this case. The output layer was 5 units (since there are five categories of prediction) and we use a softmax function as activation function again. For each 3-layer network, we also did hyperparameter tuning. We first did a grid search of learning rates between 10^{-8} and 0.1, and then finetuned once more in the chosen range. As we will later see, the most successful 3-layer neural network was one that had a hidden size of 21 units, which achieved a 71.23% accuracy.

As we expect, the 3-layer neural network was able to outperform the 2-layer neural network in just about every hidden size comparison.

5 Experiments/Results

We see that we have tried exactly three different approaches. First, we tried the logistic regression approach to provide a baseline model for comparison. Next, we attempted a 2-layer neural network with a hidden dimension size ranging from 16 to 22. Finally, we attempted a 3-layer neural network with a hidden dimension size ranging from 16 to 22 as well.

Table 1: Results of the Experiments

Approach	Hidden Layer Size	Accuracy
Logistic Regression	N/A	39.89
2-Layer Network	16	55.90
2-Layer Network	17	57.89
2-Layer Network	18	58.32
2-Layer Network	19	59.61
2-Layer Network	20	59.43
2-Layer Network	21	60.20
2-Layer Network	22	58.23
3-Layer Network	16	61.23
3-Layer Network	17	63.56
3-Layer Network	18	65.39
3-Layer Network	19	65.89
3-Layer Network	20	67.89
3-Layer Network	21	71.23
3-Layer Network	22	69.13

Now, from this table we can see that the best overall accuracy we achieved was 71.23%, which was achieved using a 3-layer neural network with a hidden dimension size of 21. Now, it is interesting to look at the general trends of these results.

As we expect, the more layers we add the better the network seems to have performed. However, there seemed to be an interesting trend in terms of how accuracy is affected by the hidden dimension size. Though we would expect higher hidden dimension to relate to higher accuracy. But we see that in both the 2-layer and 3-layer networks, the hidden size of 22 both times results in a lower accuracy. This is an odd anomaly that is hard to explain since we have done hyper-parameter searching for all of the networks. Thus, we cannot attribute it to the a difference in tuning.

Additionally, we see that we are able to match the general accuracy level shown in the Barron study on predicting soccer player's success, which is promising since we are able to achieve state of the art (or comparable) results with just 2 and 3 layers.

6 Conclusion and Future Work

The biggest conclusion we can draw is that we are able to perform at state of the art or comparable levels with our 3-layer neural network. The network performs far better than any regression method and is also significantly better at predicting success than a random guess amongst the categories, which is precisely what we want.

It is encouraging that we are able to achieve such success with deep learning since these methods can be incredibly useful for teams and organization, along with businesses. However, there are many improvements that can be made with these methods. First, we could always start with deeper networks, which would help us potentially achieve even higher accuracy.

References

[1] Ivankovic, Zdravko , Racković, Miloš , Branko, Markoski , Dragica, Radosav & Ivkovic, M.. (2010). Appliance of Neural Networks in Basketball Scouting. *Acta Polytechnica Hungarica*

[2] Barron D, Ball G, Robins M, & Sunderland C (2018) Artificial neural networks and player recruitment in professional soccer.