ECE 572 Neural Networks Term Project Report

Predicting current season's NBA match outcomes using Feedforward Neural Networks

Ourania Spantidi Vasileios Pentsos Ioannis Galanis

Introduction

Predicting a sport match outcome has been a serious topic of discussion since the birth of the betting industry. Being able to achieve a good accuracy in a match prediction would help set the betting coefficients that would later determine the wins or the losses of both the bookie and the clients that are participating in the corresponding bet. Hence, being able to predict any match outcome would be a highly important financial factor.

Of course this issue has been addressed multiple times in the past via numerous methods. The usage of plain statistics, simply counting win to lose ratio or combination of both have been applied to predict specifically NBA match outcomes. In addition, people have tried to solve this problem with many different techniques, such as hidden Markov models [2] or in our case, neural networks. [1]

Our approach involves simple feed-forward neural networks with input that is differentiated from other related work. At the first part of our project we researched data that were not used in any previous work, found the only website providing them and scrapped the data from it. At the second part of our project, we developed a database and a script that loads data from it to begin forming the inputs for our neural network. Finally, we did the training and tested our architectures on last week's games.

1. The Statistics

After studying previous work, we ended up using some *advanced* statistics. We further filtered these statistics to what we ended up calling *even more advanced* statistics. The final set that we concluded to, includes 11 total statistics, all subsequently described.

- TS% : True Shooting Percentage

  This is a measure of shooting efficiency that considers field goals, 3-point field goals and free throws. It is calculated as:

  $$TS\% = \frac{PTS}{2(FGA + (0.44 \times FTA))} \text{ x } 100$$

  where:

  PTS = points scored,

  FGA = field goal attempts,

  FTA = free throw attempts

- eFG% : Effective Field Goal Percentage

  This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal. For example, suppose Player A goes 4 for 10 with 2 threes, while Player B goes 5 for 10 with 0 threes. Each player would have 10 points from field goals, and thus would have the same effective field goal percentage (50%). It is calculated as:

  $$eFG\% = \frac{FG + 0.5 * 3P}{FGA}$$

  where:

  FG = field goals made,

  3P = 3 point field goals made,

  FGA = field goal attempts

- 3Par : 3-Point Attempt Rate

  This is a measure of percentage of field goal attempts from the 3-point range. It is calculated as $\frac{3\ Point\ Field\ Goal\ Attempts}{Field\ Goal\ Attempts}$.

- FTr : Free Throw Attempt Rate

  Calculated as the amount of Free Throws per Field Goal attempt.

- ORB% : Offensive Rebound Percentage
- DRB% : Defensive Rebound Percentage

- TRB% : Total Rebound Percentage

  These three statistics are calculated by the following formulas:

$$\text{Rebound Rate} = \frac{100 \times \text{Rebounds} \times \frac{\text{Team Minutes Played}}{5}}{\text{Minutes Played} \times (\text{Team Total Rebounds} + \text{Opposing Team Total Rebounds})}$$

$$\text{Offensive Rebound Rate} = \frac{100 \times \text{Offensive Rebounds} \times \frac{\text{Team Minutes Played}}{5}}{\text{Minutes Played} \times (\text{Team Offensive Rebounds} + \text{Opposing Team Defensive Rebounds})}$$

$$\text{Defensive Rebound Rate} = \frac{100 \times \text{Defensive Rebounds} \times \frac{\text{Team Minutes Played}}{5}}{\text{Minutes Played} \times (\text{Team Defensive Rebounds} + \text{Opposing Team Offensive Rebounds})}$$

  In basketball statistics, rebound rate or rebound percentage is a statistic to gauge how effective a player is at gaining possession of the basketball after a missed field goal or free throw. Rebound rate is an estimate of the percentage of missed shots a player rebounded while he was on the floor. Using raw rebound totals to evaluate rebounding fails to take into account external factors unrelated to a player's ability, such as the number of shots taken in games and the percentage of those shots that are made. Both factors affect the number of missed shots that are available to be rebounded. Rebound rate takes these factors into account.

- AST% : Assist Percentage

  An estimate of the percentage of teammate field goals a player assisted while he was on the floor. The formula is:

$$100 * \frac{\text{AST}}{\left(\left(\left(\frac{MP}{\frac{Tm\,MP}{5}}\right) * Tm\,FG\right) - FG\right)}$$

  where:

  AST = Assists,

  MP = Minutes Played (references the whole game),

  Tm FG = The monitored team’s Field Goals (includes both 2-point field goals and 3-point field goals),

  Tm MP = The monitored team’s Minutes Played,

  FG = Field Goals made by both teams in the match.

- STL% : Steal Percentage

  An estimate of the percentage of opponent possessions that end with a steal by the player while he was on the floor. The formula is:

$$100 * \frac{(STL) * \left(\frac{Tm\ MP}{5}\right)}{MP * Opp\ Poss}$$

where:

STL = Steals,

Tm MP = monitored teams Minutes Played,

MP = Minutes Played,

Opp Poss = Opponent team's possessions.

- BLK% : Block Percentage

  An estimate of the percentage of opponent two-point field goal attempts blocked by the player while he was on the floor. The formula is:

$$100 * \frac{BLK * \frac{Tm\ MP}{5}}{MP * (Opp\ FGA - Opp\ 3PA)}$$

where,

BLK = Blocks,

Tm MP = Monitored team's Minutes Played,

MP = Minutes Played,

Opp FGA = Opponent team's Field Goal Attempts,

Opp 3PA = Opponent team's 3-Point Field Goal Attempts.

- TOV% : Turnover Percentage

  An estimate of turnovers per 100 plays. The formula is:

$$100 * \frac{TOV}{FGA + 0.44 * FTA + TOV}$$

where,

TOV = Turnovers,

FGA = Field Goal Attempts,

FTA = Free Throw Attempts.

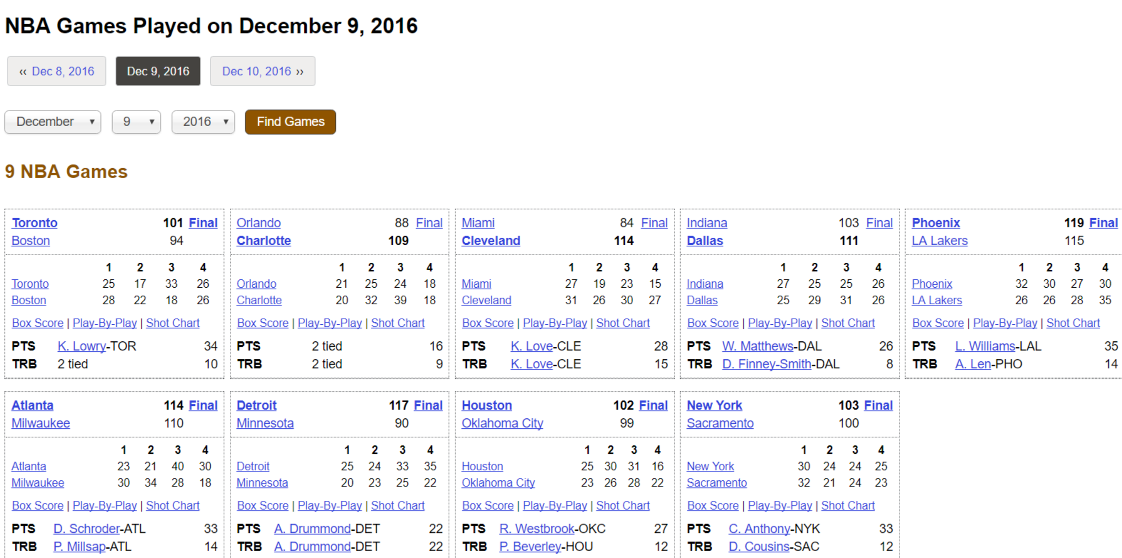
However, the statistics no matter how advanced they might be considered, are not sufficient in predicting a match outcome. There are numerous factors that affect the final result of an NBA match, and the number one factor which is also unpredictable, is the *human factor*. There are more than one cases where unexpected individual behavior can significantly affect the course of a match. If a team is on a winning or a losing streak, its psychology changes and might perform better or worse than expected accordingly. Also some games are considered "more important" than others, such as nationally broadcasted basketball games or the so called *rivalry games*. In every single NBA game there is the factor of personal motivation and performance. For example, a player that left his previous team and now plays with his current team against it, will have very negative attitude from his former team's fans in the court and therefore, might perform way worse or way better. Finally, there is also the *referee factor*, where a referee might do wrong calls or ignore turnovers that affect in a great way the final statistics.

2. Preparing the Data

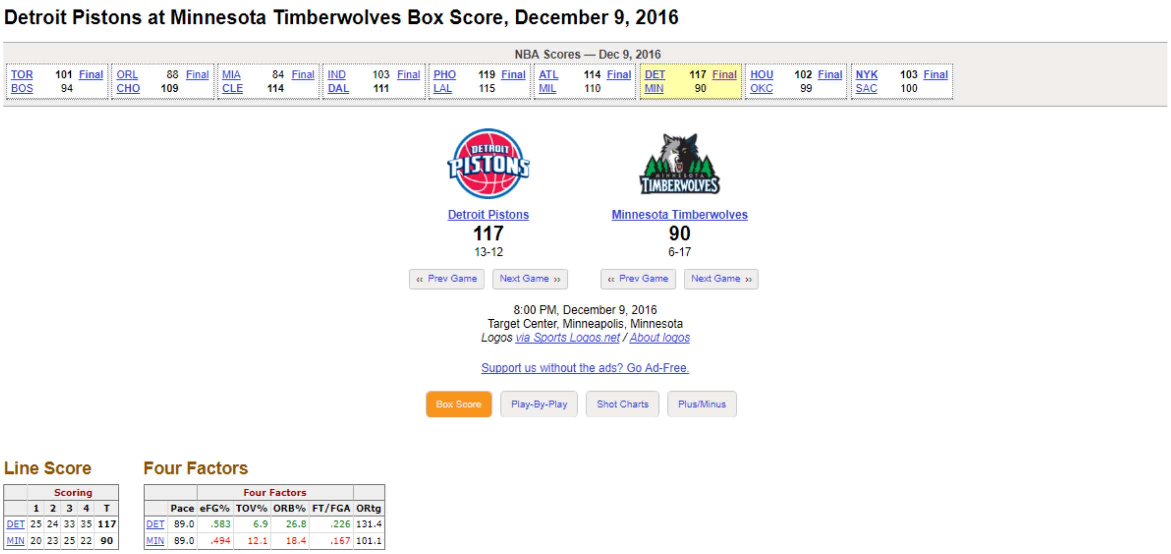
Since we ended up with the usage of these 11 specific statistics, we had to find a website to scrap this data from. The only website that could provide us with these statistics was basketball-reference.com. We agreed on gathering statistics for the season 2016 – 2017, i.e. last year's season. Hence, we had to navigate to each month, then to each day of the month and for every game that took place in this day we had to further go into the specific's game page and then get the data we wanted.

For example, for the 9th of December 2016, we have this following pipeline in order to get the advanced statistics for the match Detroit vs Minnesota:

a. Navigate to https://www.basketball-reference.com/boxscores/?month=12&day=9&year=2016



b. Program the script to click on the "Final" option on the top right of each match window so as to open the following page.



Detroit Pistons at Minnesota Timberwolves Box Score, December 9, 2016

NBA Scores — Dec 9, 2016

TOR 101 Final	ORL 88 Final	MIA 84 Final	IND 103 Final	PHO 119 Final	ATL 114 Final	DET 117 Final	HOU 102 Final	NYK 103 Final
BOS 94	CHO 109	CLE 114	DAL 111	LAL 115	MIL 110	MIN 90	OKC 99	SAC 100

Detroit Pistons
117
13-12
« Prev Game | Next Game »

Minnesota Timberwolves
90
6-17
« Prev Game | Next Game »

8:00 PM, December 9, 2016
Target Center, Minneapolis, Minnesota
Logos via Sports Logos.net / About logos

Support us without the ads? Go Ad-Free.

Box Score | Play-By-Play | Shot Charts | Plus/Minus

Line Score

	Scoring				
	1	2	3	4	T
DET	25	24	33	35	**117**
MIN	20	23	25	22	**90**

Four Factors

		Four Factors				
	Pace	eFG%	TOV%	ORB%	FT/FGA	ORtg
DET	89.0	.583	6.9	26.8	.226	131.4
MIN	89.0	.494	12.1	18.4	.167	101.1

c. By scrolling down to this page, we managed to find gathered all the statistics that we wanted.

Share & more ▾ Glossary

	Advanced Box Score Stats													
Starters	**MP**	**TS%**	**eFG%**	**3PAr**	**FTr**	**ORB%**	**DRB%**	**TRB%**	**AST%**	**STL%**	**BLK%**	**TOV%**	**USG%**	**ORtg**	**DRtg**
Marcus Morris	34:54	.625	.625	.500	.000	0.0	25.3	12.2	4.1	0.0	0.0	7.7	17.7	118	103
Kentavious Caldwell-Pope	31:25	.388	.333	.500	.167	0.0	4.0	1.9	28.6	3.4	0.0	13.4	22.5	97	100
Reggie Jackson	28:57	.477	.400	.200	.267	0.0	4.4	2.1	19.7	0.0	0.0	0.0	27.5	117	106
Andre Drummond	28:03	.745	.769	.000	.308	33.4	63.0	47.7	0.0	0.0	2.8	6.3	26.7	154	93
Tobias Harris	24:59	.805	.813	.375	.375	0.0	0.0	0.0	5.9	2.2	0.0	0.0	17.7	164	102
Reserves	**MP**	**TS%**	**eFG%**	**3PAr**	**FTr**	**ORB%**	**DRB%**	**TRB%**	**AST%**	**STL%**	**BLK%**	**TOV%**	**USG%**	**ORtg**	**DRtg**
Jon Leuer	28:03	.629	.500	.400	.800	4.2	18.0	10.8	5.1	1.9	0.0	0.0	22.9	142	100
Darrun Hilliard	16:35	.833	.833	.667	.000	0.0	0.0	0.0	8.0	0.0	0.0	25.0	11.4	122	107
Aron Baynes	15:55	.500	.500	.000	.000	7.4	7.9	7.6	7.7	0.0	0.0	0.0	6.0	143	105
Ish Smith	15:01	.800	.800	.400	.000	0.0	8.4	4.0	59.2	7.2	0.0	0.0	15.8	187	91
Beno Udrih	4:02	1.136				0.0	0.0	0.0	28.3	0.0	0.0	0.0	10.4	246	107
Henry Ellenson	4:02	.500	.500	1.000	.000	0.0	0.0	0.0	0.0	0.0	0.0	40.0	58.8	54	107
Stanley Johnson	4:02	1.000	1.000	.000	.000	29.0	62.6	45.2	0.0	0.0	0.0	0.0	11.8	212	97
Boban Marjanovic	4:02					0.0	0.0	0.0	0.0	0.0	0.0		0.0	0	106
Team Totals	**240**	**.622**	**.583**	**.345**	**.274**	**26.8**	**81.6**	**53.2**	**54.8**	**6.7**	**1.6**	**6.9**	**100.0**	**131.4**	**101.1**

d. We needed the "Team Totals". This page has the team totals for both the teams. So we scrapped the data in a vector, keeping also track of which team is the home and which the away, and we concatenated at the end of the vector 0 if the home team lost or 1 if the home team won. The final form of the vector is:

11 statistics for HOME team	11 statistics for AWAY team	Match outcome

So, the final size of the vector is 11 + 11 + 1 = 23.

e. We saved all the vectors for the 2016 – 2017 season in an excel sheet.

f. We developed a new script, that read the data from this excel sheet and formed a list of lists. That is, we had a list for every vector, and all the vectors gathered in a sort of "super" list. So each entry in this super list, is a vector that we want for our training.

3. The Neural Network Architecture

The way we decided to build out Neural Network (NN) was based on trial and error. Since there is no specific way on concluding to an optimized number of layers and number of neurons, we started experimenting with different numbers and architectures. We used the ReLU activation function for all hidden layers and the sigmoid function for the output layer. ReLU is described as $f(x) = \max(0, x)$ and sigmoid is described as $f(x) = \frac{1}{1+e^{-x}}$. The NN is a simple feed-forward architecture.

In order to begin defining out architecture, we had to first start training our data. For every experimental trial, we first trained the architecture we chose with the data we gathered from the 2016 – 2016 season as described in Chapter 2, and then we decided to do the testing on the games that took place on the 1st, 2nd and 3rd of December.

The issue that emerged from our tactic so far, was that the statistics were based on match outcomes. Therefore, in order to make a prediction we would need every team's statistics in matches so far. Hence, we decided to make another script that would scrap data from the same website using the same procedure, but this time it would form a database with the names of the teams. For every match, it would monitor the teams that played and the statistics for each team, so to find a specific team all we had to do was to query the database for the two teams that we wanted, get their statistics for all the games played in ***this*** season, i.e. 2017 – 2018 season, and then find the average values.

For example, the new script produces an excel sheet like this:

Boston Celtics	Cleveland Cavaliers	17/10/2017												
MP	TS%	eFG%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	ORtg	DRtg
240	.500	.455	.364	.284	18.0	80.4	47.9	66.7	11.1	6.6	9.2	100.0	99.7	102.7
240	.543	.488	.265	.301	19.6	82.0	52.1	50.0	3.0	7.1	15.3	100.0	102.7	99.7
Houston Rockets	Golden State Warriors	17/10/2017												
MP	TS%	eFG%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	ORtg	DRtg
240	.579	.562	.423	.196	22.2	84.6	51.2	59.6	8.8	10.0	10.2	100.0	119.6	118.6
240	.678	.638	.375	.263	15.4	77.8	48.8	79.1	4.9	16.1	16.0	100.0	118.6	119.6
Milwaukee Bucks	Boston Celtics	18/10/2017												
MP	TS%	eFG%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	ORtg	DRtg
240	.605	.546	.276	.395	15.8	78.0	51.1	50.0	6.1	6.3	14.4	100.0	110.4	102.2
240	.499	.489	.308	.231	22.0	84.2	48.9	59.0	12.3	3.6	10.7	100.0	102.2	110.4
Atlanta Hawks	Dallas Mavericks	18/10/2017												
MP	TS%	eFG%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	ORtg	DRtg
240	.582	.559	.191	.160	30.4	80.0	54.9	41.7	11.2	14.6	11.4	100.0	118.7	112.6
240	.583	.541	.523	.244	20.0	69.6	45.1	71.1	5.1	5.3	13.6	100.0	112.6	118.7
Charlotte Hornets	Detroit Pistons	18/10/2017												

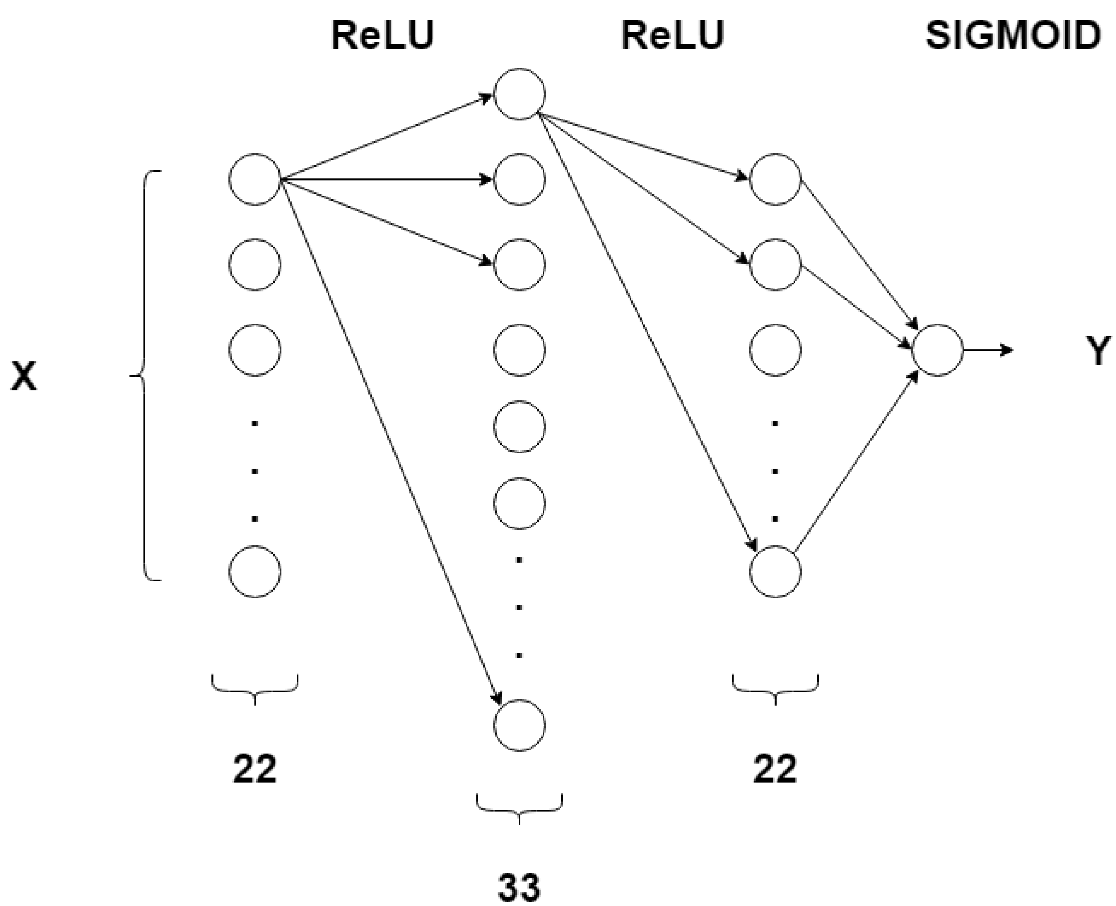
So, for the 2nd of December we wanted to see the match prediction for Cleveland Cavaliers and Memphis Grizzlies. The script will query the excel sheet for every single match the Cleveland Cavaliers have played, and keep a list of lists with all of them. Then it would merge into a single list with the average values for each statistic. Same with the Memphis Grizzlies.
We ran four experimental trials all described in the fourth chapter.

4. Experiments

We developed all of our work using Python and implemented our network using the Keras deep learning library with TensorFlow backend. We consider an acceptable accuracy rate a correct prediction of 2 out of 3 matches, which means we consider a successful experiment any architecture that produces over 66.7% successful predictions.

a. 1st Experiment

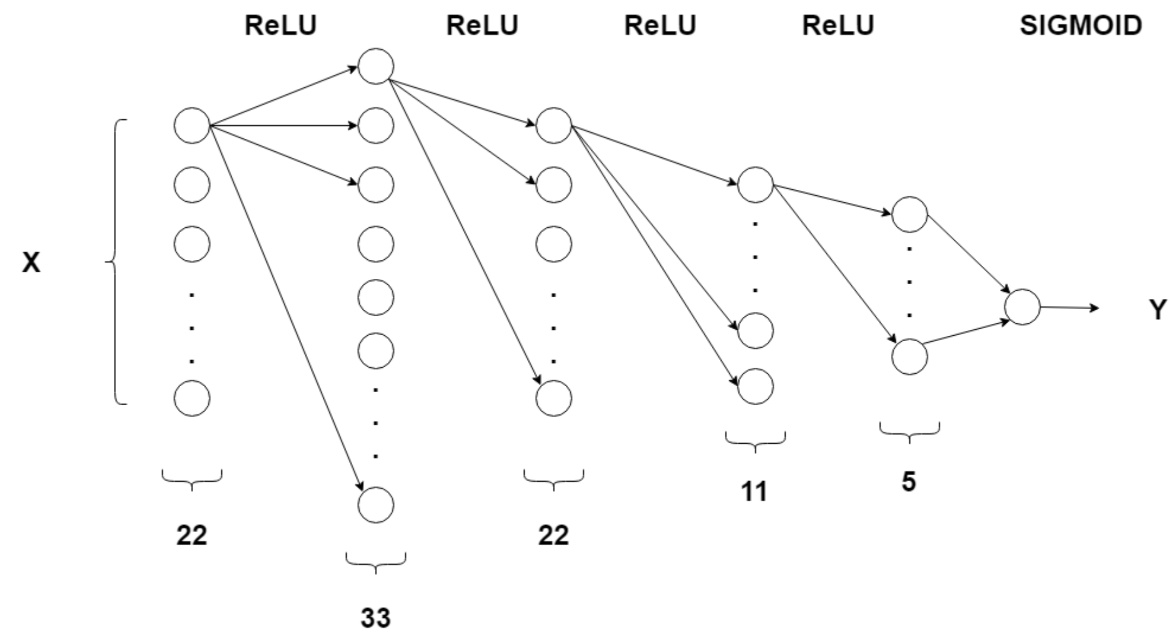
For the first experiment we implemented the following network architecture:



which achieved 47.6% accuracy. That is 10 out of 21 matches, which according to our initial assumptions is unacceptable. So we decided to increase our layers and toy with the number of neurons in each hidden layer.

b. 2nd Experiment

For this experiment we added more layers up to a total of 5 as shown below:



The results are shown below:

Sunday 12/3 games:

- New York Knicks vs Orlando Magic
- Miami Heat vs Golden State Warriors
- Minnesota Timberwolves vs LA Clippers
- Oklahoma City Thunder vs San Antonio Spurs
- Los Angeles Lakers vs Houston Rockets

Saturday 12/2 games:

- Boston Celtics vs Phoenix Suns
- Dallas Mavericks vs Los Angeles Clippers
- Brooklyn Nets vs Atlanta Hawks
- Cleveland Cavaliers vs Memphis Grizzlies
- Philadelphia 76ers vs Detroit Pistons
- Milwaukee Bucks vs Sacramento Kings
- Denver Nuggets vs Los Angeles Lakers
- Portland Trail Blazers vs New Orleans Pelicans

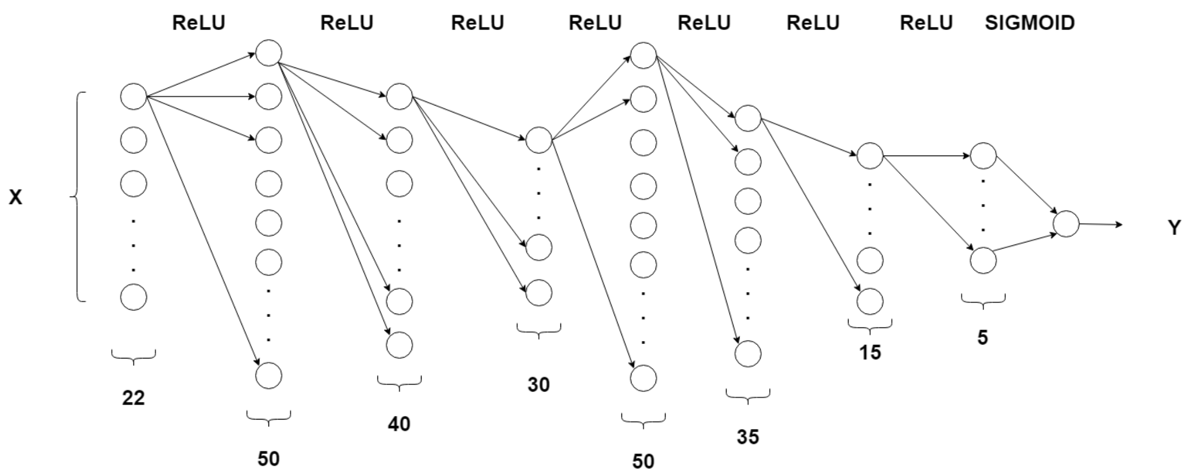
Friday 12/01 games:

- Orlando Magic vs Golden State Warriors
- Washington Wizards vs Detroit Pistons
- Toronto Raptors vs Indiana Pacers
- Miami Heat vs Charlotte Hornets
- Chicago Bulls vs Sacramento Kings
- Memphis Grizzlies vs San Antonio Spurs
- Oklahoma City Thunder vs Minnesota Timberwolves
- Utah Jazz vs New Orleans Pelicans

The highlighted teams are the teams that won the match, and the green color indicates a correct network prediction while the red indicates a wrong prediction. This architecture achieved a 71.4% accuracy which was more than sufficient for us to consider as successful.

c. 3rd Experiment

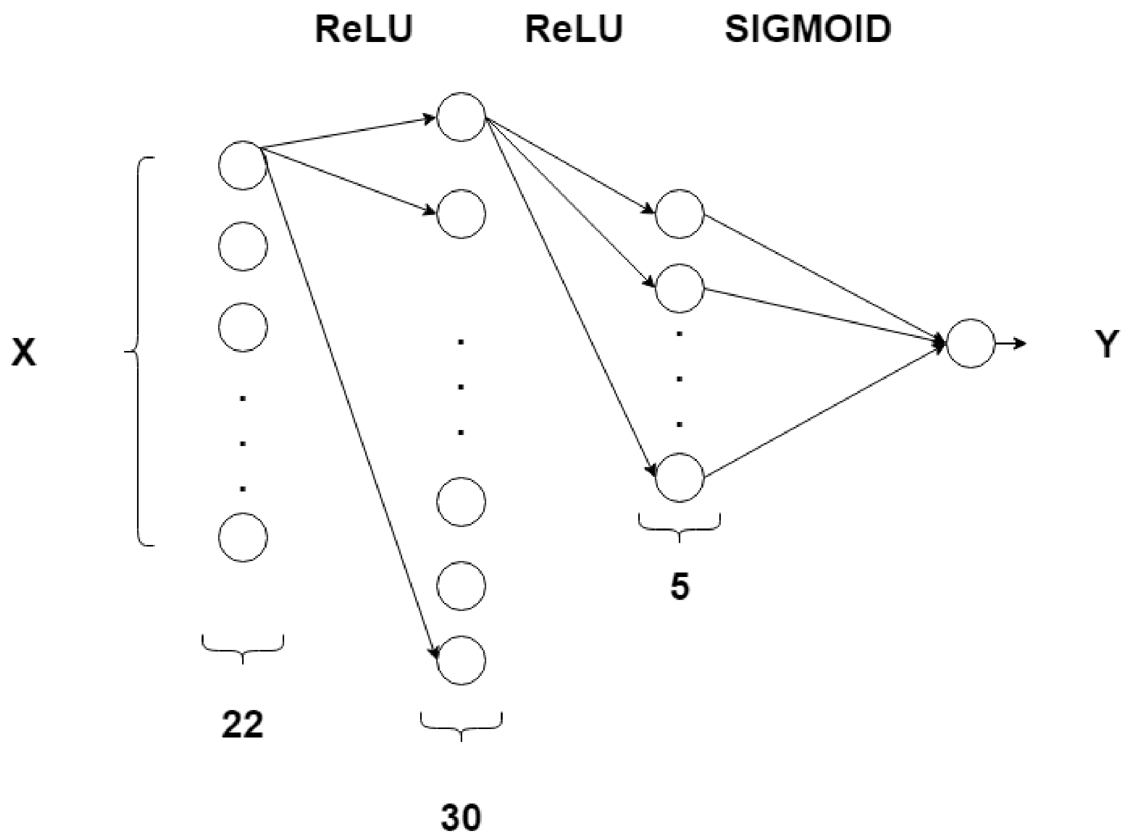
Since the additional hidden layers looked to provide additional accuracy and better predictions, we decided to add even more to a total of 8 hidden layers, with the NN architecture being the one shown below:



However, this architecture did not provide the expected results, since it only produced a 66.7% accuracy. Even though we considered anything above 66.7% a successful experiment, after achieving the percentage of 71.4% at the 2nd experiment, we figured that adding more layers doesn't necessarily mean improved results, hence we decided to run one last experimental trial

d. 4th Experiment

Since more layers resulted in less accuracy, we went back to the 1st experiment's architecture and changed the number of neurons of the hidden layers, with the NN architecture shown below:



Again, this architecture resulted in even smaller accuracy, 52.4%. We decided to stop experimenting at this point.

5. Conclusion

The number of layers and neurons in each layer is a decision to be taken at random. There is no way to specify the optimal architecture for a NN that is designed to predict. However, through random testing with the architecture of our network, we achieved over 70% accuracy in our 2nd experiment which exceeded our expectations.

For future work, we assumed that better predictions could be achieved by following a more player oriented approach and not a global team-oriented one, since rosters change throughout the seasons and from season to season.

References

[1] *Loeffelholz, Bernard & Bednar, Earl & W Bauer, Kenneth. (2009). Predicting NBA Games Using Neural Networks. Journal of Quantitative Analysis in Sports*

[2] http://vashishtm.net/pdf/hmm_nba.pdf

[3] https://keras.io/models/sequential/

[4] https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

[5] https://www.basketball-reference.com/about/glossary.html

[6] https://deeplearning4j.org/neuralnet-overview

[7] http://docs.python-guide.org/en/latest/scenarios/scrape/

[8] http://www.orbifold.net/default/2016/11/25/some-feedforward-neural-networks-using-keras/

[9] J. Heaton, S. McElwee, J. Fraley and J. Cannady, "Early stabilizing feature importance for TensorFlow deep neural networks," 2017 International Joint Conference on Neural Networks (IJCNN)