Project Documentation:

Neural Networks and Genetic Algorithms

Ambrose M Malagon Cordero

Bellevue University

DSC 680 – Applied Data Science

Sunday May 30, 2021

Table of Contents

Abstract

We define Artificial Neural Networks as computing systems designed to replicate the way human brain analyze and work through problems. These algorithms work together under a single framework called the neural network. Neural networks are inspired by the structure of biological neural networks in a human brain. There is an input neuron which acts as an interface for all the other neurons to pass the input. Also, there is an output neuron which accepts all the outputs from different neurons.

Artificial Neural Networks fall under the purview of Deep Learning. Other methods which such as Linear and Logistic regression fall under the scope of Machine Learning, which focuses on using algorithms to parse, learn and make informed decisions on the data it is given. Both Deep Learning and Machine Learning aim at tackling the same prediction and classification problems, however both have distinct prerequisites that make it preferrable to use one method over the other. Similar to Machine Learning, the success of any Deep Learning model is based on the choice of basic parameters such as network topology, learning rate and the initial weights. While there are basic guidelines we can follow, these tend to be arbitrary at best.

Genetic algorithms are global search methods, that are based on principles like selection, crossover and mutation. Genetic Algorithms can be used to optimize the topology of a neural network. The purpose of this project is to explore the use of Neural Networks to predict Major League Baseball score results and use Genetic Algorithms to potentially optimize the topology of our Neural Network.

Introduction

FiveThirtyEight, which takes its name from the number of electors in the United States electoral college, is an American website that focuses on opinion poll analysis, politics, economics, and sports blogging. The site is known for its use of advanced statistics techniques to produce their predictions, while providing articles that give insights into their approach.

For their Major League Baseball predictions, FiveThirtyEight utilized data dating all the way back to 1871 to create an Elo-based rating system and predictive model for baseball that accounts for home-field advantage, margin of victory, park and era effects, travel, rest and starting pitchers. A quick glance at the data they provide, their methodology and their business case, this data and it's intended use are ripe for either regression or classification analysis.

The amount of data collected, transformed and utilized for this endeavor alone exceeds 200,000 observations, which this a perfect use case for Deep Learning Neural Networks. Using the data provided by FiveThirtyEight, our goal is to create several deep learning models, assess their overall performance and select the best one.

Our secondary goal will be the use of Genetic Algorithms to optimize the topology of the Neural Network developed with this data, assess its performance and compare it against our best performing base model.

Business Questions/Hypothesis

[1] Performance wise, which base neural network model is the best?

[2] Does a Genetic Algorithm enhanced Neural Network perform better than a base neural network model?

[3] Does deep learning make sense for this data?

[4] How do Linear Regression results compare against the Neural Network results?

[5] Why use Genetic Algorithms?

[6] Are there other methods aside from Genetic Algorithms we can use to optimize Neural Networks?

[7] Can we use the methods applied in this case study for future Neural Network development? Will this be our method going forward?

[8] Can we gain insights from FiveThirtyEight's methodology towards predicting MLB scores?

[9] What are the standards when defining a Neural Network? Are there any official guidelines?

[10] Can we explore this as a classification problem?

Methods

Step I – Data Acquisition and Preparation.

- MLB Elo – Provided by FiveThirtyEight. Retrieved from https://github.com/fivethirtyeight/data/tree/master/mlb-elo. The GitHub repository contains the following two csv files:
    - mlb_elo.csv – contains game-by-game Elo ratings and forecasts back to 1871.
    - mlb_elo_latest.csv – contains game-by-game Elo ratings and forecasts for only the latest season.

Both files were downloaded from the GitHub repository on May 22, 2021. For the purposes of this case study, we will use the mlb_elo.csv as it contains all of the pertinent observations.

MLB Elo data:

Contains 238,389 observations with the following 26 variables:

Column	Definition
date	Date of game
season	Year of season
neutral	Whether game was on a neutral site
playoff	Whether game was in playoffs, and the playoff round if so
team1	Abbreviation for home team
team2	Abbreviation for away team
elo1_pre	Home team's Elo rating before the game
elo2_pre	Away team's Elo rating before the game
elo_prob1	Home team's probability of winning according to Elo ratings
elo_prob2	Away team's probability of winning according to Elo ratings
elo1_post	Home team's Elo rating after the game
elo2_post	Away team's Elo rating after the game
rating1_pre	Home team's rating before the game
rating2_pre	Away team's rating before the game

pitcher1	Name of home starting pitcher
pitcher2	Name of away starting pitcher
pitcher1_rgs	Home starting pitcher's rolling game score before the game
pitcher2_rgs	Away starting pitcher's rolling game score before the game
pitcher1_adj	Home starting pitcher's adjustment to their team's rating
pitcher2_adj	Away starting pitcher's adjustment to their team's rating
rating_prob1	Home team's probability of winning according to team ratings and starting pitchers
rating_prob2	Away team's probability of winning according to team ratings and starting pitchers
rating1_post	Home team's rating after the game
rating2_post	Away team's rating after the game
score1	Home team's score
score2	Away team's score

A preliminary inspection provides the following insights:

- The preliminary variable contains the most null values. As such, we chose to drop it.
- We drop all observations/rows that contain null values. Our resulting dataset contains 187,200 observations with 25 available variables.
- Taking into account my computer's processing power and its inability to use GPUs to aid in computations for Neural Networks, we need to limit the amount of data in order to achieve a sensible computation time. As such, we select data for 8 teams out of the 89 teams present in the data set:
    - ATL – Atlanta Braves.
    - BAL – Baltimore Orioles.
    - CHC – Chicago Cubs.
    - CLE – Cleveland Indians/Naps/Broncos/Bluebirds/Lake Shores.
    - FLA – Florida Marlins.
    - HOU – Houston Astros/Colt .45s

- PIT – Cleveland Indians/Naps/Broncos/Bluebirds/Lake Shores
- TB – Tampa Bay Buccaneers.

This preselection reduces our data size down to 49,491 observations and 25 variables.

Selecting a Target Variable:

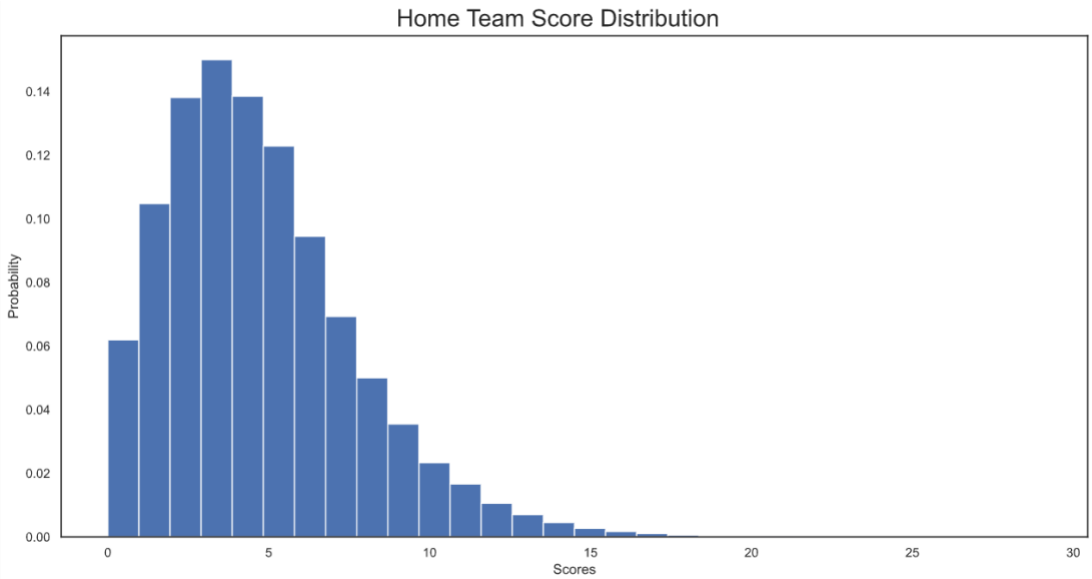
We select the score1 variable as our target variable. Since this is a numeric/float type variable, we can assess that our Neural Network has to be tuned towards regression/predicting the resulting number. With this in mind, we make the following adjustments:

- Remove date variable – Cannot use dates for regression.
- Remove score2 – By this same logic, we could use this as an alternative target variable.

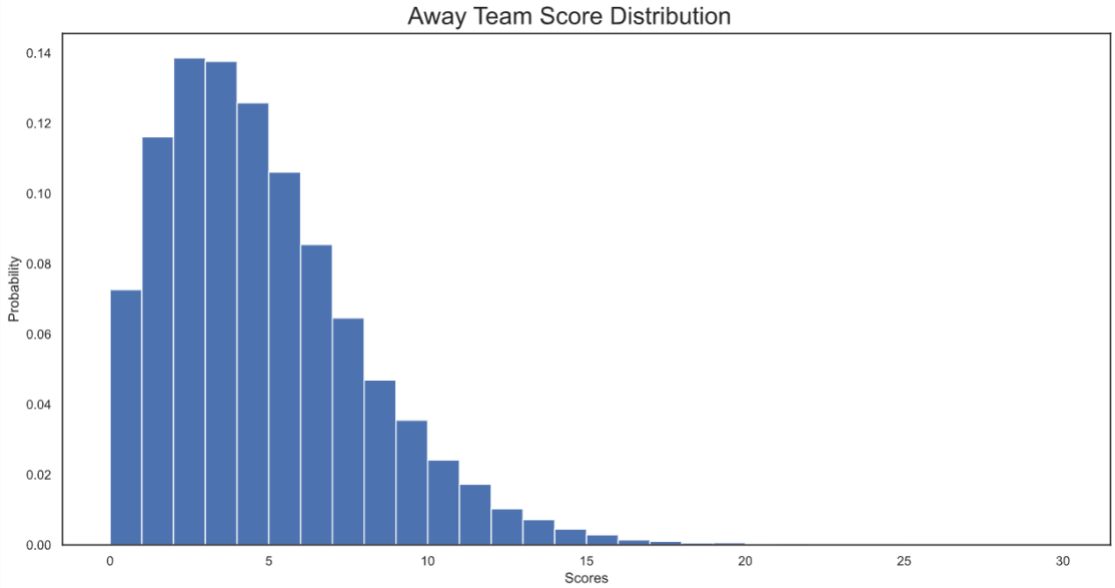
This reduces our total number of variables down to 22, with the same number of observations (49,491) as before.

Step II – Exploratory Data Analysis.

We start by evaluating the score1 variable, which represents the Home Team's score, for the selected teams. Notice how the data follows a normal distribution, with scores higher than 10 becoming scarcer.

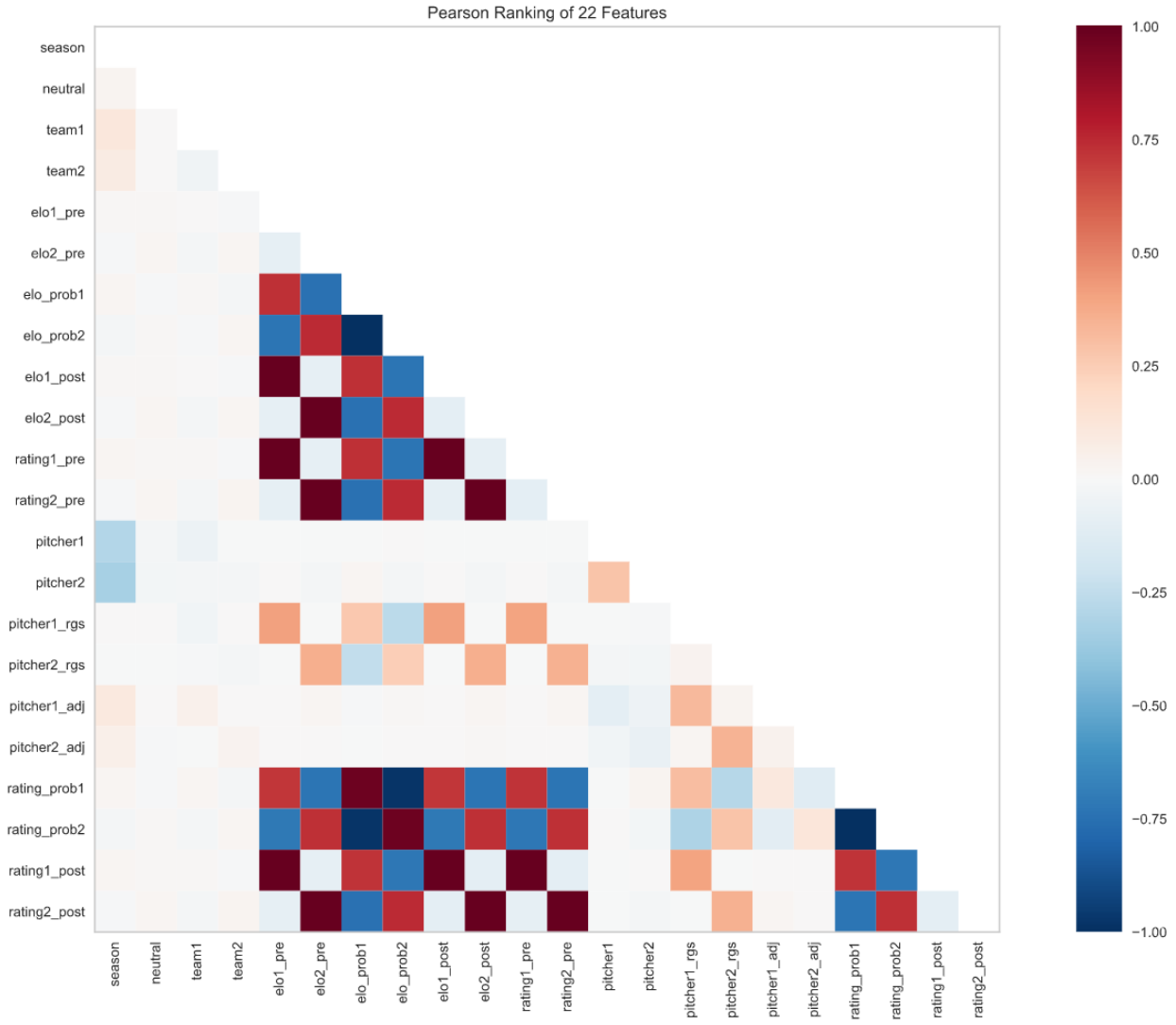


While we are omitting the score2 variable as a predictor variable, the Away Team's score distribution also appears to follow the same distribution as the score1 variable:

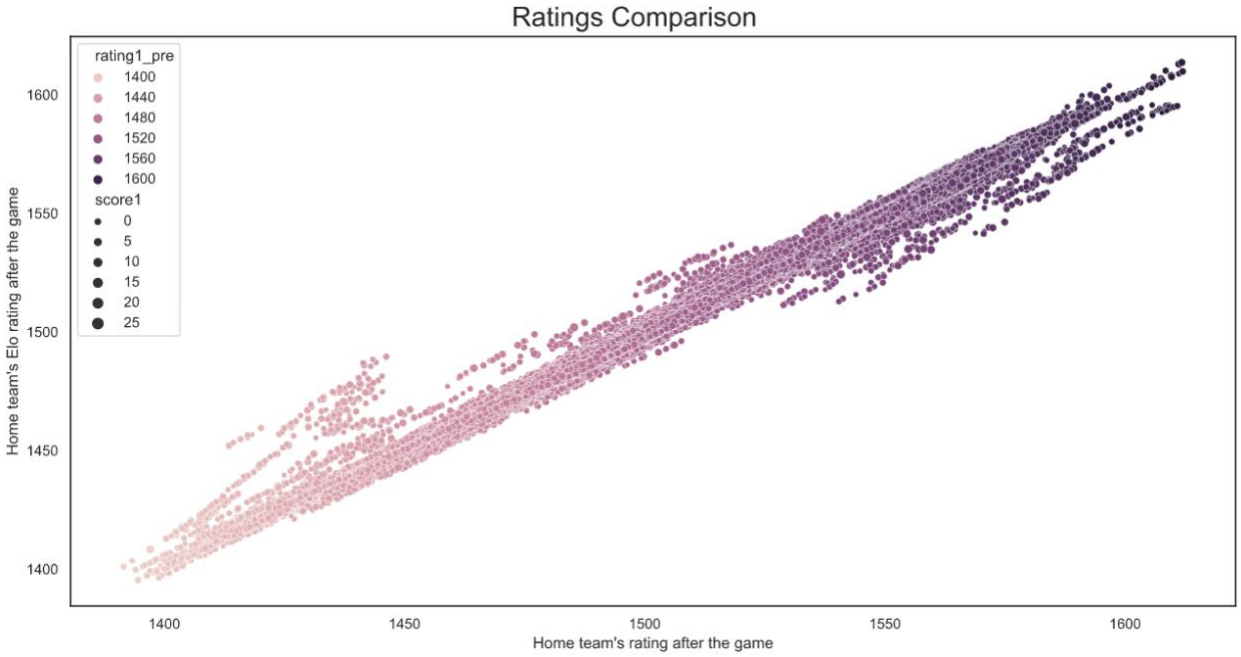


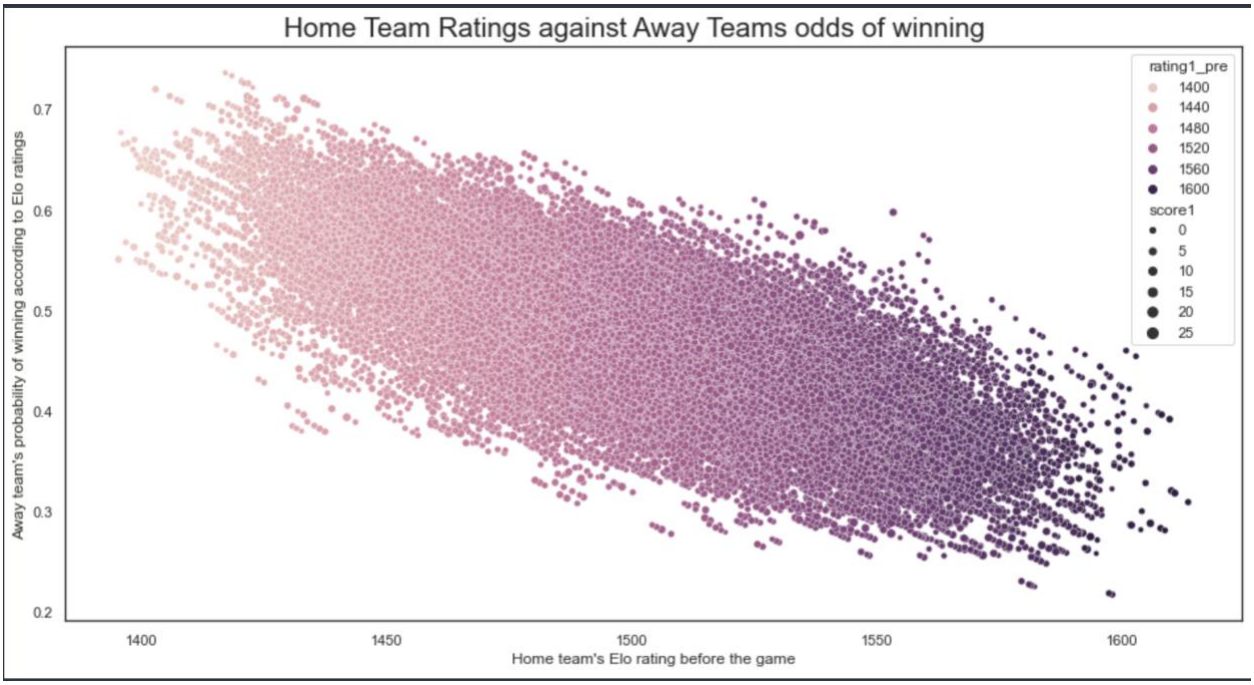
Explanatory Variables

Using Pearson's Correlation, we look at the degree of correlation between all 22 selected explanatory variables:



Notice how a good portion of the variables have strong correlation levels with other variables. That is because some of these values were actually derived/calculated as explained by the FiveThirtyEight team (https://fivethirtyeight.com/features/how-our-mlb-predictions-work/)


Ratings Comparison
rating1_pre
1400
1440
1480
1520
1560
1600
score1
0
5
10
15
20
25
Home team's Elo rating after the game
1400
1450
1500
1550
1600
Home team's rating after the game


Home Team Ratings against Away Teams odds of winning
rating1_pre
1400
1440
1480
1520
1560
1600
score1
0
5
10
15
20
25
Away team's probability of winning according to Elo ratings
0.2
0.3
0.4
0.5
0.6
0.7
1400
1450
1500
1550
1600
Home team's Elo rating before the game

Step III – Building the Neural Networks

We are going to build 3 Neural Network models using keras:

- A baseline mode with one layer.
- A single layer model with standardized data.
- A two-layer model with standardized data.

First Attempt – Building a Baseline Single-Layer model using Keras.

We pass all 22 variables, using a baseline function that runs for a total of 10 epochs and cross-validated a total of 5 times. We get the following results:

```
Results: -10.06 (1.85) MSE
```

It is important to note that these results could change based on the run instance, but will fall within the same range.

Second Attempt – Building a single layer Neural Network model with standardized data.

Here, we standardize the data for the model and pass it along to a single layer model for a total of 10 epochs and cross validated a total of 5 times for the following results:

```
Standardized: -5.43 (0.80) MSE
```

Third attempt – Building a 2-layer Neural Network with standardized data.

The first layer assumes the use of all 22 input (standardized) variables. Our second layer utilizes receives a total of 11 inputs, which is half of the entry. Running for a total of 10 epochs and cross validated 5 times we get the following results:

```
Larger Model: -5.27 (0.73) MSE
```

Step IV – Using Support Vector Machines to predict Home Team Scores

We compare the benchmark results of these Neural Network models against standard Machine Learning models such as Support Vector Machines. We start by evaluating the R^2 score for the Polynomial, Gaussian, Sigmoid and Linear Support Vector Regression methods:

```
SVR Polynomial R^2 Score: 0.21803816673268395
SVR Gaussian R^2 Score: 0.3617812081887404
SVR Sigmoid R^2 Score: 0.32811098161961394
SVR Linear R^2 Score: -36325.56523922383
```

We note that the R^2 score for the Gaussian Support Vector Regression provides the best predictive output. However, an R^2 score of 0.36 means that the results will not be a close match to the original values in the testing subset. We can assess this by calculating the Mean Squared Error:

```
SVR Polynomial MSE: 7.527771975927586
SVR Gaussian MSE: 6.091629457280394
SVR Sigmoid MSE: 6.610456816835408
SVR Linear MSE: 321750.6912321718
```

Thus, we can conclude that a basic Neural Network without standardized data will yield better results than any of the machine learning methods provided within the Support Vector Machine.

Step IV – Genetic Algorithm Neural Network Optimization

Genetic Algorithms are biological-inspired optimization algorithms that have gained importance over the years. Genetic Algorithms can solve for complex challenges such as route optimization, training machine learning algorithms and playing games. The process for using Genetic Algorithms is as follows:

1. Determine the problem and goal
2. Break down the solution to bite-sized properties (genomes)
3. Build a population by randomizing said properties
4. Evaluate each unit in the population
5. Selectively breed (pick genomes from each parent)
6. Rinse and repeat

Genetic Algorithms are best used with challenges where there is a clear way to evaluate fitness. If the problem is not well-constrained or if the process is computationally expensive, then Genetic Algorithms may not find the solution in a reasonable amount of time. This then prompts the following question – Is our intended goal/data optimized for a Genetic Algorithm? We answer this question by answering the previously posted questions:

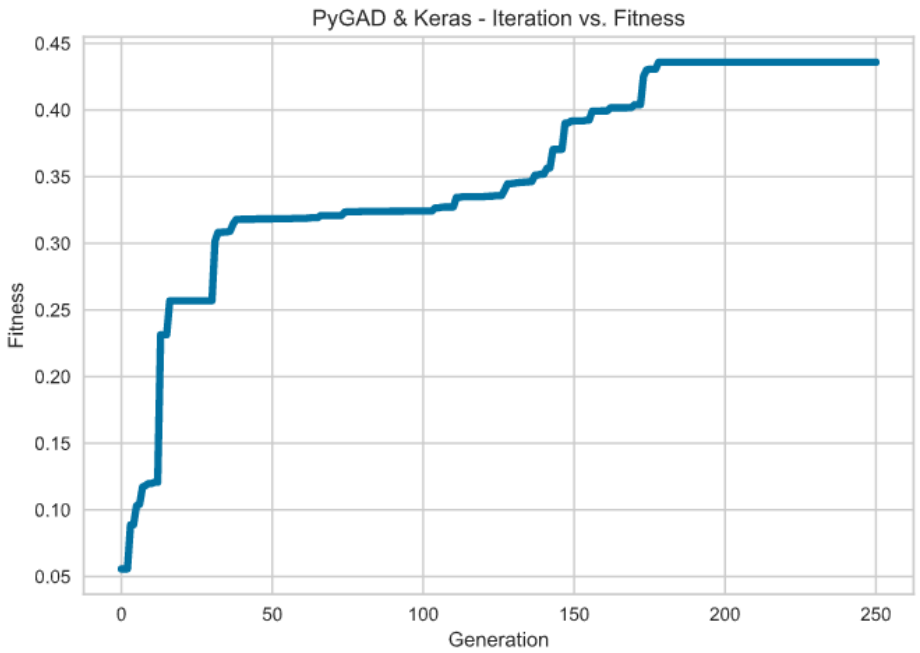
- Is the problem well-constrained? We have no clear way of delineating a fitness function.
- Is it computationally expensive? Yes.

For the purpose of this case study, we limit the amount of data to just one team: the Atlanta Braves. We also constrain our features from the selected 22 to the following 5:

- elo_prob1
- elo_prob2
- pitcher1_rgs
- rating1_pre
- rating2_pre

We set the parameters as follows:

- Number of Generations: 250
- Number of Parents: 2
- Number of solutions: 10



Our best Fitness results at 0.43758145181866953 with an absolute error 2.2941508. This implies that our choice of variables, our parameters were not optimal or computation limits.

Addressing the Research Questions

[1] Performance wise, which base neural network model is the best?

The two-layer model was the better performer out of the two standardized data models with an MSE of 0.73.

[2] Does a Genetic Algorithm enhanced Neural Network perform better than a base neural network model?

Not in this case. The Genetic Algorithm only reached a 0.43 Fitness Function with absolute error higher than 2. The resulting weights did little to optimize the results.

[3] Does deep learning make sense for this data?

Yes. One of the key tenets of Neural Networks is that they improve with the addition of more data. In this case, the FiveThirtyEight team provides all of their projected estimates for all Major League Baseball games since 1871. Not accounting for missing data, this is over 180,000 observations.

[4] How do Linear Regression results compare against the Neural Network results?

Poorly. Using Support Vector Regression, we note that the best performing Machine Learning model, we note that it is actually the Gaussian model with an MSE of 6.091 which is much higher than one produced by the Neural Networks.

[5] Why use Genetic Algorithms?

The goal was to see if we could further optimize the Neural Network to produce more accurate predictions. That was not the case with this data. The resulting Neural Network had the worst performance.

[6] Are there other methods aside from Genetic Algorithms we can use to optimize Neural Networks?

Genetic Algorithms are a subset of Evolutionary Algorithms such as the Swarm Algorithms, Gaussian Adaptation, Adaptive Dimension Search and Memetic Algorithms.

[7] Can we use the methods applied in this case study for future Neural Network development? Will this be our method going forward?

Not with this use case. If anything, I want to continue exploring the possibilities of Genetic Algorithms and Machine Learning with the right data set.

[8] Can we gain insights from FiveThirtyEight's methodology towards predicting MLB scores?

Yes, but it was not focus of this case study. The initial goal was to try and replicate the FiveThirtyEight process to general Elo scores but that was not achieved. This will be tackled as a side project.

[9] What are the standards when defining a Neural Network? Are there any official guidelines?

Not necessarily. While we understand some of the basic tenets and concepts of how Neural Networks are supposed to work, many elements such as the total number of layers and weights are complete guesswork making it a somewhat (if not completely) trial and error process.

[10] Can we explore this as a classification problem?

Potentially – If we were to create a binary win/lose variable that populates based on the Home Team's score total against the Away Team's score total. How this variable would correlate with the rest of the dataset was something we did not get to explore.

Discussion/Conclusion

The goal of this project was to see if it was possible to optimize a Neural Network using a Genetic Algorithm. While there's an inherent nature to try and discover insights within our data, our focus is more geared towards technique. In essence: Can we find ways to optimize our models that are more precise than simple trial and error?

We were successful in creating several Neural Networks with the MLB predictions dataset. We ran and measured their performance against the Support Vector Regression methods, noting that the Neural Networks had a better MSE scores. While Deep Learning Models benefit from copious amounts of data, we had to subset our data due to a smaller subset of data due to machine performance. We need to ensure we take the time to review the results of model using a different subset of the data to ensure the accuracy of our results.

However, we did not succeed in our second endeavor. The data set, or potentially, the problem in itself was not ripe to use Genetic Algorithms for Neural Network optimization. If anything, we need to take more time to study this material and potentially understand when and how we can use this technique effectively.

Appendix – List of Articles, Studies and Data Sources regarding Neural Networks and Genetic Algorithms.

[1] Xin-She Yang. Genetic Algorithm. Science Direct. Research Paper. Retrieved from https://www.sciencedirect.com/topics/engineering/genetic-algorithm

[2] Shubhaim Jain. Introduction to Genetic Algorithm & their application in data science. Analytics Vidhya. Article. Retrieved from https://www.analyticsvidhya.com/blog/2017/07/introduction-to-genetic-algorithm/

[3] Sourabh Katoch, Sumit Singh Chauhan & Vijay Kumar. A review on genetic algorithm: past, present, and future. Springer Link. Research Paper. Retrieved from https://link.springer.com/article/10.1007/s11042-020-10139-6

[4] Greg Sommerville. Using genetic algorithms on AWS for optimization problems. AWS Machine Learning Blog. Article. Retrieved from https://aws.amazon.com/blogs/machine-learning/using-genetic-algorithms-on-aws-for-optimization-problems/

[5] Peter G. Zhang. Neural Networks for Classification: A Survey. ResearchGate. Article. Retrieved from https://www.researchgate.net/publication/3421357_Neural_Networks_for_Classification_A_Survey

[6] Robert Keim. How to Perform Classification Using a Neural Network: What Is the Perceptron? All About Circuits. Article. Retrieved from https://www.allaboutcircuits.com/technical-articles/how-to-perform-classification-using-a-neural-network-introducing-the-perceptron/

[7] Oliver Knocklein. Classification Using Neural Networks. Towards Data Science. Article. Retrieved from https://towardsdatascience.com/classification-using-neural-networks-b8e98f3a904f

[8] Pranoy Radhakrishnan. What are Hyperparameters? and how to tune the Hyperparameters in a Deep Neural Network? Towards Data Science. Article. Retrieved from https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

[9] Jason Brownlee. Recommendations for Deep Learning Neural Network Practitioners. Machine Learning Mastery. Article. Retrieved from https://machinelearningmastery.com/recommendations-for-deep-learning-neural-network-practitioners/

[10] Pulkit Sharma. Improving Neural Networks – Hyperparameter Tuning, Regularization, and More (deeplearning.ai Course #2). Analytics Vidhya. Article. Retrieved from

https://www.analyticsvidhya.com/blog/2018/11/neural-networks-hyperparameter-tuning-regularization-deeplearning/