IST 597 Project Proposal

Kenneth Hall
11/18/2017

Overview

League of Legends is a popular multiplayer online battle arena (MOBA) video game. Two teams of five players control characters called "champions" and compete to complete objectives. Teams begin the game on opposite corners of a sizeable map. To win the game, they must destroy a sequence of enemy turrets, invade the enemy base, and destroy the enemy nexus.

During the game, players engage in complex strategic actions to increase the strength of their champions, accomplish objectives, and eliminate players on the enemy team. Games tend to last between 30-45 minutes, although some may be much shorter or longer. League of Legends has over 100 million unique monthly players, and billions of League of Legends games have been played. Each game generates a tremendous amount of data, which is archived and made available by Riot (the company behind League of Legends) to enable players to track the progress of themselves and their friends.

I propose to use neural networks to predict the outcome of a League of Legends match based upon player performance in prior games. Due to the tremendous amount of data available, I anticipate that a network trained on a sufficiently large dataset can achieve a high accuracy at this prediction task.

Approach

Riot makes data from all League of Legends matches available to developers through the [Riot Games API](). A Python interface is available [here](). I will create a Python script to analyze League of Legends games and collect prior game data for each player on the team. This data will be used to train a neural network to predict whether a team will win or lose based upon their match history.

I will implement a multi-layer, fully-connected neural network using Google's Tensorflow library. At least 1,000 games will be included in the training set, and over 80,000 games will be represented in the prior game data from player histories.

Player performance data will be measured by 50 variables for each match. Some of these parameters will include:

- Kills
- Assists
- Deaths
- Damage to Objectives
- Damage to Turrets
- Damage that is Self-Mitigated
- Gold Earned
- Gold Spent
- Neutral minions killed
- Physical Damage Dealt
- Magical Damage Dealt
- Double, Triple, Quadra, and Pentakills
- Time applying Crowd Control to Others
- Wards Placed
- Victory or Defeat

Additional performance metrics will be drawn from match timelines, such as damage taken per minute, gold acquired per minute, and experience points gained per minute.

Expected Results

Randomly guessing which team will win a League of Legends match is like flipping a coin. The expected accuracy approaches 50% as the number of games in the dataset increases.

Clever League of Legends players have come up with heuristics to help them evaluate whether a game is likely to be won. Some of the features examined by such players include "team composition", or the set of champion characters that are selected by the team; player champion mastery, or how experienced players are using a particular champion, and which side of the battlefield their team is on (blue or red).

To establish a baseline performance metric, I decided to analyze several thousand League of Legends matches and attempt to predict a winner using these metrics. Team composition was too complex to analyze with only several thousand games, so I discarded this analysis. However, player champion mastery was sufficient to predict winrate with an accuracy of 60%. I also tested the truth of the player theory that the blue team wins more often than the red team. In League of Legends matches, the blue team starts at the bottom left of the screen and advances towards the top right, while the red team follows the opposite path. I discovered that indeed, the blue team wins 55% of League of Legends matches, despite rumors that Riot has attempted to rebalance the teams by having the matchmaking system select players with marginally higher skill levels overall for the red team.

Mention of the League of Legends matchmaking system brings me to an important point. Since there is an intelligent agent selecting the matchups for League of Legends (attempting to make them as fair as possible, presumably), why do I predict that a neural network can achieve better than 50% accuracy at classifying the winner to a matchup?

The answer is that the matchmaking system has to balance creating fair team matchups with doing so in a timely manner. Therefore, I infer that Riot's matchmaking algorithm is likely not optimized entirely for matchup fairness. This means that a prediction system with a sufficient amount of data may be able to achieve greater than 50% accuracy at win prediction. If Riot's matchmaking system is indeed optimized well enough that it can match the performance of a large neural network on win prediction, then I expect the neural network will not achieve much greater than 50% accuracy at win prediction.

Bibliography

Several attempts have been made to use data fitting models for League of Legends match prediction.

A project by Lucas Lin showed that champion mastery and summoner spells were insufficient for successful pre-match prediction with gradient boosted trees. He also showed that the inclusion of data from the match (unsurprisingly) contributed to a better prediction of win/loss outcomes.

This project by Thomas Huang, David Kim, and Leung claims to be able to predict the win probability with a success rate of over 90% using player-specific champion win rates as pre-match data. They compare the performance of Decision Trees, Nearest Neighbor, Multilayer Perceptron, and two flavors of Bayesian Networks on this task. One problem with their approach is that the champion specific win rate data they rely on is no longer available in the Riot API.

My approach uses dramatically more features to evaluate player skill and win potential than either of these precedents. Therefore, I predict that this new approach will outperform the results from the above references.

Image Sources

Champions Map Zac