Ben Barnett - ben.barnett@stanford.edu

/ff @ 20: Competitive Gaming Analysis in League of Legends

Objective/Motivation

This project aims to investigate what minimum information is necessary to accurately predict the outcome of a ranked, competitive match of League of Legends ("LoL"). **LoL is an online, competitive, team-based game made by Riot Games with over 80 million monthly players from over 145 countries** [1]. As esports and competitive gaming continues to grow internationally, the demand for prediction algorithms and "realtime" match analytics (before a game has finished) will likely soon mirror those of basketball, football, and other popular sports. Furthermore, the results of this study could provide insight for professional coaches in terms of knowing what in-game objectives and priorities should be focused on most heavily, potentially altering strategies, player behavior, and the "meta" of League of Legends.

Data

All data used was downloaded from a Kaggle dataset published by Chuck Ephron, which contains comprehensive match information from approximately 7600 challenger-tier competitive matches from 2015 to 2018 [2].

The datapoint of a single match contains information including the gold difference between teams at every minute mark until the game is completed, as well as the outcome of the game, times that each structure was destroyed, champions selected, kills at each minute, and more.

Features

From the dataset above, I extracted the gold difference (scalar) between teams at 1 minute increments until 20 minutes, as well as the times that structures were destroyed rounded to the nearest minute mark (with the exception of the baseline model described in the following section).

To encode temporal structure elimination, a 28-vector of boolean values was assigned to each minute mark of the game, wherein each index corresponds to a single structure (see Fig. 1), and a value of "true" means that structure was destroyed at approximately that minute mark.

Models

Model 1: 2 layer NN "Baseline"
Fed 20-vector containing gold differences at each minute

Model 2: 2 layer NN with enhanced input features
Fed 580-vector containing gold differences and structural information (concatenated minute encodings)

Model 3: 1 layer RNN/LSTM with enhanced input features
Fed temporal structure gold/diff embeddings (see Fig. 2)

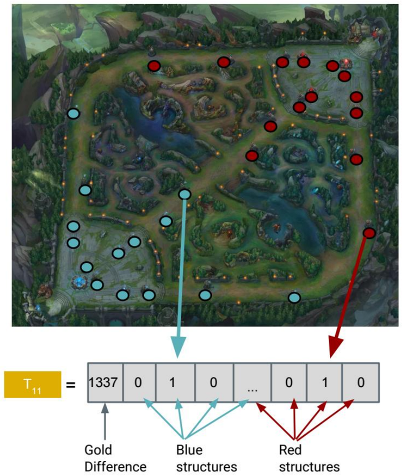

Figure 1: Map of Summoner's Rift (above) with example minute encoding (below)

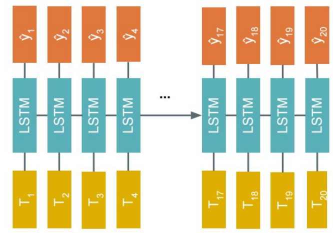

Figure 2: Final model architecture, using 128-hidden-unit LSTM cells and cross-entropy loss with a sigmoid activation function

Results

When provided all game state information until 20 minutes, models yielded the following result with a <u>train set of 7000</u> and a <u>test set of 600</u>, using <u>Adam optimization</u> with <u>grad. descent</u>.

Model 1: 98.2% train accuracy, <u>56.2% test</u> accuracy
Model 2: 89.0% train accuracy, <u>70.3% test</u> accuracy
Model 3: 81.3% train accuracy, <u>79.5% test</u> accuracy

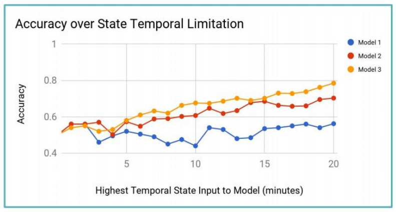

Figure 3: Accuracy for different models as the highest temporal game state is limited from 1 to 20 minutes

Conclusions

The results of the final model imply that challenger-tier matches can be predicted with at least 79% accuracy by looking at the two teams' gold difference and times that objectives were taken up to the game's 20 minute mark.

In other words, without even examining selected champions, kills, individual gold differences, or previous player history, an accuracy of over 80% was achieved. Furthermore, it is highly likely that an improved NN architecture or different hyperparameters could yield even better results. The simpler models may have also provided better results if some form of regularization were implemented, as overfitting is clear from the train/test accuracy discrepancy.

One other important fact to consider is that given how much variation in gameplay there could be in the following 5-25 minutes after the 20 minute mark, it may be impossible to make a prediction better than about 80% accuracy after 20 minutes; there is no definite outcome

Future

Given the success of this final model with such limited information, I would like to investigate the effects of adding other input features (apriori and temporal), such as kills, individual players' gold differences, player items at each time, and champions selected. I am also curious about predicting more than just the outcome of the game; for example, how accurately could a model predict the location of the next kill, or the next objective(s) to be taken?

References & Acknowledgements

Special thanks to my mentor, Zach Barnes, and the rest of the CS230 teaching staff.
Thanks to Riot for making a kickass game worth investigating.

[1]. "You'll Never Guess How Many People Play League of Legends." Unranked Smurfs. Oct. 4, 2017.

[2]. Ephron, Chuck. "League of Legends Competitive Matches, 2015 to 2018." Kaggle. Jan. 29, 2018.