LeagueNet

Problem

League Of Legends (LoL) is a game with 100 million monthly active players. Many players craft ever-evolving strategies on how to win and climb the ranked ladder. A tool to evaluate the likelihood of winning a match based on certain variables would be invaluable to both professional organizations who compete for multi-million dollar prize pools and casual players who want an edge. We aimed to create a model that intakes variables such as player ranks and champions in game to output whether a team would win or lose.

Data

The data collection was perhaps the hardest part of this project. By manually scraping usernames and ranks off of public match records hosted on op.gg, we were able to then query Riot APIs for further match history. However, there were a LOT of complications you could read more about in our paper. Eventually, we had a bunch of match data JSON from Riot and other data in a few text files from op.gg.

Features

We settled with 20 input features after tweaking with hundreds of potential features. We found that the ranks of in-game players and the champions they are playing (software patch dependent) are the best indicators of success. This makes sense - rank correlates to a player's skill and certain champions tend to have better synergy with others.

Models

We're using a three layer, batch normalized, fully connected neural net with the output layer using a ReLu activation function.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

Softmax function

$$H(p,q) = -\sum_x p(x) \log q(x).$$

Cross entropy

Results

Model	Training Acc	Test Acc	Train/Test
1	0.57	0.62	15999/883
2	0.56	0.58	15999/883

Model 1 = Three layer, batch-norm, relu output

Model 2 = Restricted Boltzmann Machine

Discussion

Since no one has ever tried to conduct a study like this on League of Legends, with the most similar study being a paper on predicting soccer results using recurrent neural networks, we weren't sure what to expect. A lot of the architectures we studied in class, such as RNNs and CNNs, didn't really make sense because of the nature of the data and desired output, so a lot of research and experimentation was needed to find an appropriate neural net. We're sure the results of the net could be better. With the comprehensiveness of the data we gathered from Riot and op.gg, there are so many more input features we could add. However, we would need to spend a great amount of time in collecting more data and tweaking the architecture accordingly. Had we come into this project with a lot more data, we are positive the results would have been substantially better.

Future

If we had time, we would definitely collect more data and increase the number of input features.

Segrio Sardar(ssardar@stanford.edu), Brandon Walker(bmwalk@stanford.edu)